

APPLIED DIGITAL IMAGING

Image Processing for Embedded Devices



Editors:

Sebastiano Battiato
Arcangelo Ranieri Bruna
Giuseppe Messina
Giovanni Puglisi
University of Catania
Italy

Image Processing for Embedded Devices

From CFA data to image/video coding

Editors:

Sebastiano Battiato
Arcangelo Ranieri Bruna
Giuseppe Messina
Giovanni Puglisi

eBooks End User License Agreement

Please read this license agreement carefully before using this eBook. Your use of this eBook/chapter constitutes your agreement to the terms and conditions set forth in this License Agreement. Bentham Science Publishers agrees to grant the user of this eBook/chapter, a non-exclusive, nontransferable license to download and use this eBook/chapter under the following terms and conditions:

1. This eBook/chapter may be downloaded and used by one user on one computer. The user may make one back-up copy of this publication to avoid losing it. The user may not give copies of this publication to others, or make it available for others to copy or download. For a multi-user license contact permission@bentham.org
2. All rights reserved: All content in this publication is copyrighted and Bentham Science Publishers own the copyright. You may not copy, reproduce, modify, remove, delete, augment, add to, publish, transmit, sell, resell, create derivative works from, or in any way exploit any of this publication's content, in any form by any means, in whole or in part, without the prior written permission from Bentham Science Publishers.
3. The user may print one or more copies/pages of this eBook/chapter for their personal use. The user may not print pages from this eBook/chapter or the entire printed eBook/chapter for general distribution, for promotion, for creating new works, or for resale. Specific permission must be obtained from the publisher for such requirements. Requests must be sent to the permissions department at E-mail: permission@bentham.org
4. The unauthorized use or distribution of copyrighted or other proprietary content is illegal and could subject the purchaser to substantial money damages. The purchaser will be liable for any damage resulting from misuse of this publication or any violation of this License Agreement, including any infringement of copyrights or proprietary rights.

Warranty Disclaimer: The publisher does not guarantee that the information in this publication is error-free, or warrants that it will meet the users' requirements or that the operation of the publication will be uninterrupted or error-free. This publication is provided "as is" without warranty of any kind, either express or implied or statutory, including, without limitation, implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the results and performance of this publication is assumed by the user. In no event will the publisher be liable for any damages, including, without limitation, incidental and consequential damages and damages for lost data or profits arising out of the use or inability to use the publication. The entire liability of the publisher shall be limited to the amount actually paid by the user for the eBook or eBook license agreement.

Limitation of Liability: Under no circumstances shall Bentham Science Publishers, its staff, editors and authors, be liable for any special or consequential damages that result from the use of, or the inability to use, the materials in this site.

eBook Product Disclaimer: No responsibility is assumed by Bentham Science Publishers, its staff or members of the editorial board for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products instruction, advertisements or ideas contained in the publication purchased or read by the user(s). Any dispute will be governed exclusively by the laws of the U.A.E. and will be settled exclusively by the competent Court at the city of Dubai, U.A.E.

You (the user) acknowledge that you have read this Agreement, and agree to be bound by its terms and conditions.

Permission for Use of Material and Reproduction

Photocopying Information for Users Outside the USA: Bentham Science Publishers Ltd. grants authorization for individuals to photocopy copyright material for private research use, on the sole basis that requests for such use are referred directly to the requestor's local Reproduction Rights Organization (RRO). The copyright fee is US \$25.00 per copy per article exclusive of any charge or fee levied. In order to contact your local RRO, please contact the International Federation of Reproduction Rights Organisations (IFRRO), Rue du Prince Royal 87, B-1050 Brussels, Belgium; Tel: +32 2 551 08 99; Fax: +32 2 551 08 95; E-mail: secretariat@ifrro.org; url: www.ifrro.org This authorization does not extend to any other kind of copying by any means, in any form, and for any purpose other than private research use.

Photocopying Information for Users in the USA: Authorization to photocopy items for internal or personal use, or the internal or personal use of specific clients, is granted by Bentham Science Publishers Ltd. for libraries and other users registered with the Copyright Clearance Center (CCC) Transactional Reporting Services, provided that the appropriate fee of US \$25.00 per copy per chapter is paid directly to Copyright Clearance Center, 222 Rosewood Drive, Danvers MA 01923, USA. Refer also to www.copyright.com

CONTENT

<i>Foreword</i>	<i>i</i>
<i>Preface</i>	<i>ii</i>
<i>Biographies</i>	<i>iv</i>
<i>Contributoru</i>	<i>vii</i>
<i>Acknowledgement</i>	<i>viii</i>
CHAPTER	
1. Fundamentals and HW/SW Partitioning	01
<i>S. Battiato, G. Puglisi, A. Bruna, A. Capra and M. Guarnera</i>	
2. Notions about Optics and Sensors	10
<i>A. Bruna, A. Capra, M. Guarnera and G. Messina</i>	
3. Exposure Correction	34
<i>A. Castorina and G. Messina</i>	
4. Pre-acquisition: Auto-focus	54
<i>A. Capra and S. Curti</i>	
5. Color Rendition	92
<i>A. Bruna and F. Naccari</i>	
6. Noise Reduction	117
<i>A. Bosco and R. Rizzo</i>	
7. Demosaicing and Aliasing Correction	149
<i>M. Guarnera, G. Messina and V. Tomaselli</i>	
8. Red Eyes Removal	191
<i>G. Messina and T. Meccio</i>	
9. Video Stabilization	217
<i>T. Meccio, G. Puglisi and G. Spampinato</i>	
10. Image Categorization	237
<i>G. M. Farinella and D. Ravi</i>	
11. Image and Video Coding and Formatting	270
<i>A. Bruna, A. Buemi, G. Spampinato</i>	
12. Quality Metrics	310
<i>Ivana Guarneri</i>	
13. Beyond Embedded Devices	343
<i>S. Battiato, A. Castorina and G. Messina</i>	
Index	374

Foreword

Image Processing in embedded devices has been an area of growing interest with the revolution of digital imaging devices since the last decade of the 20th century and it will continue to expand to new frontiers in this century. Despite its relevance, there is not, as far as I know, a comprehensive publication that address this topic encompassing practical aspects of image processing design.

With chapters contributed by both experienced researchers from academia as well as researchers and engineers from industry, the present publication covers fundamental aspects of image processing in embedded devices such as exposure correction, auto-focus, color rendition, noise reduction, demosaicing, encoding, red-eye removal, image categorization and presents relevant quality metrics and also recent trends in imaging.

The editors have done an excellent job of bringing out contributors that work with the challenges of finding solutions and also implementing image processing solutions for embedded imaging devices in a daily basis with continuous spread across all relevant operational aspects for an imaging system.

I believe, the present publication is going to be beneficial not only to imaging and engineering students but also be a reference for academic researchers and engineers working in imaging industry.

This publication is also unique because it moves away from the traditional paper book for technical publications and follows the trend of electronic book. This makes the publication more accessible, more portable with current e-readers in the market, potentially more environment friendly without ever going out of print. Electronic publications have also attribute such language accessibility by electronic translations and text-to-speech software capabilities.

It is a great pleasure for me to write a foreword for this prestigious, multi-authored, international publication on a topic that I believe is very relevant to the imaging industry. Finally, I would like to compliment the editors and contributors for their effort in making this publication a great success.

Francisco Imai, Ph.D.

Principal Scientist

Canon Development Americas, Inc.

3300 North First Street

San Jose, CA, 95134 USA

Preface

Embedded imaging devices, such as digital still and video cameras, mobile phones, personal digital assistants, and visual sensors for surveillance and automotive applications, make use of the single-sensor technology approach. An electronic sensor (Charge Coupled Device - CCD or Complementary Metal-Oxide-Semiconductor - CMOS) is used to acquire the spatial variations in light intensity and then uses image processing algorithms to reconstruct a color picture from the data provided by the sensor. Acquisition of color images requires the presence of different sensors for different color channels. Manufacturers reduce the cost and complexity by placing a color filter array (CFA) on top of a single sensor, which is basically a monochromatic device, to acquire color information of the true visual scene.

The overall performance of any device are the result of a mixture of different components including hardware and software capabilities and, not ultimately, overall design (i.e., shape, weight, style, etc.).

This book is devoted to cover algorithms and methods for the processing of digital images acquired by single-sensor imaging devices. Typical imaging pipelines implemented in single-sensor cameras are usually designed to find a trade-off between sub-optimal solutions (devoted to solve imaging acquisition) and technological problems (e.g., color balancing, thermal noise, etc.) in a context of limited hardware resources. State of the art techniques to process multichannel pictures, obtained through color interpolation from CFA are very advanced. On the other hand, not too much is known and published about the application of image processing techniques directly on CFA images, i.e. before the color interpolation phase.

The various chapters of the book cover all aspects of algorithms and methods for the processing of digital images acquired by imaging consumer devices. More specifically, we will introduce the fundamental basis of specific processing into CFA domain (demosaicing, enhancement, denoising, compression). Also ad-hoc matrixing and color balancing techniques devoted to preprocess input data coming from the sensor will be treated. In almost all cases various arguments have been presented in a tutorial way in order to provide to the readers a comprehensive overview of the main basis of each involved topics. All contributors are well renowned experts in the field as demonstrated by the number of related patents and scientific publications.

The main part of the book analyzes the various aspects of the imaging pipeline from the CFA data to image and video coding. A typical imaging pipeline is composed by two functional modules (pre-acquisition and post-acquisition) where the data coming from the sensor in the CFA format are properly processed. The term pre-acquisition is referred to the stage in which the current input data coming from the sensor are analyzed just to collect statistics useful to set parameters for correct acquisition.

The book also contains a number of chapters that provide solution and methods to address some undesired drawbacks of acquired images (e.g., red-eye, jerkiness, etc.); an overview of the current technologies to measure the quality of an image is also given. Just considering the impressive (and fast) growth in terms of innovation and available technology we conclude the book just presenting some example of solution that makes

use of machine learning for image categorization and a brief overview of recent trends and evolution in the field.

Catania (Italy), June 2010.

Sebastiano Battiato
Arcangelo Ranieri Bruna
Giuseppe Messina
Giovanni Puglisi

Biographies

Sebastiano Battiato

Sebastiano Battiato was born in Catania, Italy, in 1972. He received the degree in Computer Science (summa cum laude) in 1995 and his Ph.D. in Computer Science and Applied Mathematics in 1999. From 1999 to 2003 he has led the "Imaging" team c/o STMicroelectronics in Catania. Since 2004 he has been a researcher at Department of Mathematics and Computer Science of the University of Catania. His research interests include image enhancement and processing, image coding and camera imaging technology. He published more than 90 papers in international journals, conference proceedings and book chapters. He has authored 2 books and is a co-inventor of about 15 international patents. He is a reviewer for several international journals and he has been regularly a member of numerous international conference committees. He has participated in many international and national research projects. He is an Associate Editor of the SPIE Journal of Electronic Imaging (Specialty: digital photography and image compression). He is a director (and cofounder) of the International Computer Vision Summer School. He is a Senior Member of the IEEE. For more details see (<http://www.dmi.unict.it/battiato>)

Arcangelo R. Bruna

Arcangelo R. Bruna received the degree in Electronic Engineering (summa cum laude) in 1998 at the University of Palermo. First he worked in a telecommunication company in Rome. He joined STMicroelectronics in 1999 where he works in the Advanced System Technology (AST) Catania Lab - Italy. Today he leads the Image Generation Pipeline and Codecs group and his research interests are in the field of image acquisition, processing and enhancement. He published several patents and papers in international conferences and journals.

Giuseppe Messina

Giuseppe Messina was born in Crhange, France, in 1972. He received his MS degree in Computer Science in 2000 at the University of Catania doing a thesis about Statistical Methods for Textures Discrimination. Since March 2001 he has been working at STMicroelectronics in the Advanced System Technology (AST) Imaging Group as Software Design Senior Engineer II / PL. Since 2007 he is Ph.D. student in Computer Science at the University of Catania accomplishing a research in Information Forensic by Image/Video Analysis. He is member of the Image Processing Laboratory, at the University of Catania. His research interests are in the field of Image Analysis e Image Quality Enhancement. He is author of about several papers and patents in Image Processing field. He is a reviewer for several international journals and international conferences. He is an IEEE member.

Giovanni Puglisi

Giovanni Puglisi was born in Acireale, Italy, in 1980. He received his degree in Computer Science Engineering (summa cum laude) from Catania University in 2005 and his Ph.D. in Computer Science in 2009. He is currently contract researcher at the Department of Mathematics and Computer Science and member of IPLab (Image Processing Laboratory) at the University of Catania. His research interests include video stabilization, artificial mosaic generation, animal behavior and raster-to-vector conversion techniques. He is the author of several papers on these activities.

Image Processing Lab (<http://iplab.dmi.unict.it>)

IPLab research group is located at Dipartimento di Matematica ed Informatica in Catania. The scientific knowledge of the group is on Computer Graphics, Multimedia, Image processing, Pattern Recognition and Computer Vision. The group has a good expertise in the overall digital camera pipeline (e.g., acquisition and post acquisition processing) as well as a good and in-depth knowledge of the recognition of scene categorization field. This is confirmed by the numerous research paper, within the area of image processing in single sensor domain (in acquisition and post acquisition time) as well as different works relatively the semantic analysis of images content, to drive some image processing tasks such as image enhancement. Moreover, the collaboration between members of the Catania unit and industrial company leaders in single sensor imaging (e.g., STMicroelectronics) has already done the possibility of transferring to the industry (pre-competitive research) the knowledge acquired in academic research facilitating the industry in producing new advanced products and patents. A joint research lab IPLab-STMicroelectronics, has been recently created where researchers coming from both partners work together on imaging research topics. More specifically, 2 Ph.D. students in Computer Science (XXIII Ciclo Dottorato in Informatica - Università di Catania) have received financial support by STMicroelectronics to investigate about "Methodologies and Algorithms for Image Quality Enhancement for Embedded Systems". The group published more than 100 papers on topics related to the previous mentioned disciplines. Moreover the IPLab group established a number of international relationships with academic/industrial partners for research purpose. In the last years the group organized the "Fourth Conference Eurographics Italian Chapter 2006" and the "International Computer Vision Summer School 2007, 2008, 2009, 2010" (<http://www.dmi.unict.it/icvss>).

Advanced System Technology - Catania Lab - STMicroelectronics (<http://www.st.com>)

Advanced System Technology (AST) is the STMicroelectronics organization in charge of system level research and innovation. Active since 1998, AST responds to the need to strengthen the position of STMicroelectronics as a leading-edge system on chip company. The AST Catania Lab and, in particular, the Imaging Group, works on research and innovation in the field of imaging processing. Its mission is to acquire digital pictures with superior Performance/Cost using advanced image processing methodologies, to extend the acquisition capability of imaging devices through the development of new applications and to determine the computational power, the required bandwidth, the flexibility and the whole imaging engine. Its members have long experience in image algorithms, documented also by many patents and scientific publications. Primarily, through active contacts and collaborations with several universities and a dedicated joint lab with the IPLab of Catania University, they have concretized and made effective the link between academic and industrial R&D.

List of Contributors

Image Processing Lab, University of Catania, Italy.

Sebastiano Battiato	battiato@dmi.unict.it
Giovanni Maria Farinella	gfarinella@dmi.unict.it
Tony Meccio	meccio@dmi.unict.it
Giovanni Puglisi	puglisi@dmi.unict.it
Rosetta Rizzo	rosetta.rizzo@dmi.unict.it

Advanced System Technology - Catania Lab, STMicroelectronics, Italy.

Angelo Bosco	angelo.bosco@st.com
Arcangelo Ranieri Bruna	arcangelo.bruna@st.com
Antonio Buemi	antonio.buemi@st.com
Alessandro Capra	alessandro.capra@st.com
Alfio Castorina	alfio.castorina@st.com
Salvatore Curti	salvatore.curti@st.com
Mirko Guarnera	mirko.guarnera@st.com
Ivana Guarneri	ivana.guarneri@st.com
Giuseppe Messina	giuseppe.messina@st.com
Filippo Naccari	filippo.naccari@st.com
Daniele Ravi'	daniele.ravi-uni@st.com
Giuseppe Spampinato	giuseppe.spampinato@st.com
Valeria Tomaselli	valeria.tomaselli@st.com

Acknowledgements

We would like to take this opportunity to thank all contributors of this book, and all people working into the two involved research groups: Image Processing Lab (Catania University) and Advanced System Technology, Catania Lab (STMicroelectronics). A special thanks to Massimo Mancuso for having contributed with his extraordinary tenacity and competence to establish an imaging R&D group in Catania. We also thank prof. Giovanni Gallo for his invaluable help and support.

Fundamentals and HW/SW Partitioning

S. Battiato, G. Puglisi

Image Processing Lab, University of Catania, Italy.

A. Bruna, A. Capra, M. Guarnera

Advanced System Technology - Catania Lab, STMicroelectronics, Italy.

Abstract: The main goal of this Chapter is devoted to provide all the fundamental basis related to the involved technological issues relative to the single-sensor imaging devices. A rough understanding of the overall ingredients of a typical imaging pipeline is important also to consider the performance of any imaging devices, from low to high level, as the result of several components that run together to compose a complex system. The final image/video quality is the result of a certain number of design choices, that involve, in almost all cases, all aspects of the hardware and software technology. As briefly stated in the preface, the book aims to cover all aspects of algorithms and methods for the processing of digital images acquired by imaging consumer devices. More specifically, we will introduce the fundamental basis of specific processing into CFA (Color Filter Array) domain such as demosaicing, enhancement, denoising, compression together with ad-hoc matrixing, color balancing and exposure correction techniques devoted to preprocess input data coming from the sensor. We conclude the Chapter just including some related issues related to the intrinsic modularity of the pipeline together with a brief description of the hardware/software partitioning design phase.

1.1 The Simplest Imaging Pipeline

A typical imaging pipeline (see Fig.(1.1)) is composed by two functional modules (pre-acquisition and post-acquisition) where the data coming from the sensor in the CFA format are properly processed. The term pre-acquisition is referred to the stage in which the current input data coming from the sensor are analyzed just to collect statistics useful to set parameters for correct acquisition. In some cases several application can be present.

The initial data is composed by a matrix of data, coming from the sensor. For each pixel only a single chromatic value is acquired just using suitable CFA, typically arranged in the classic Bayer format. We omit all the details about optics and sensor capabilities that will be deeply treated in the next Chapter. Starting from the CFA data ad-hoc algorithms and methods can be used to obtain, at the end of the process, a compressed RGB version of the acquired scene. Some high-end devices allow the saving of the input data without applying any kind of processing, including compression, just providing as output an intermediate format, called "raw" format, where each pixel contains values very similar to those acquired by the sensor in the corresponding photosite. In the remaining cases, an imaging pipeline is needed to reconstruct (or recover) the missing data, maximizing whenever is possible, the related image quality. In the following Subsections we briefly summarize, with some examples, the typical (and mandatory) processing steps, just providing some initial overview of the relative algorithms that will be treated in more details in the rest of the book.

As depicted in Fig.(1.1) there could be a series of functional blocks devoted to im-

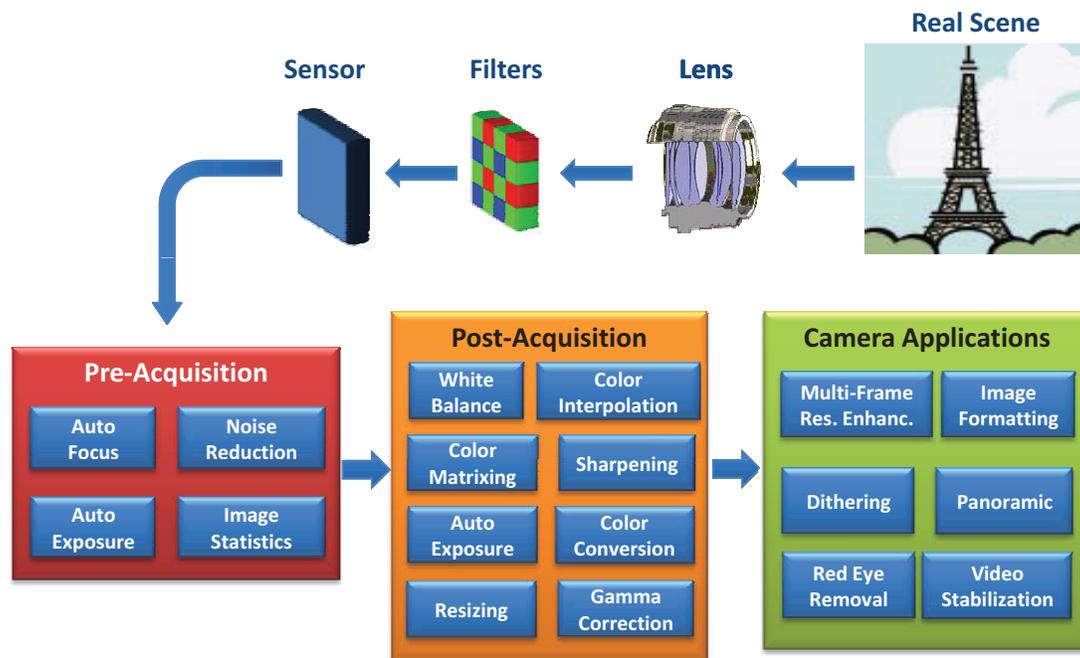


Figure 1.1 : Typical imaging pipeline. Data coming from the sensor (typically in Bayer format) are first analyzed to collect useful statistics for parameters setting (pre-acquisition) and then properly processed in order to obtain, at the end of the process, a compressed RGB image of the acquired scene (post-acquisition and camera applications).

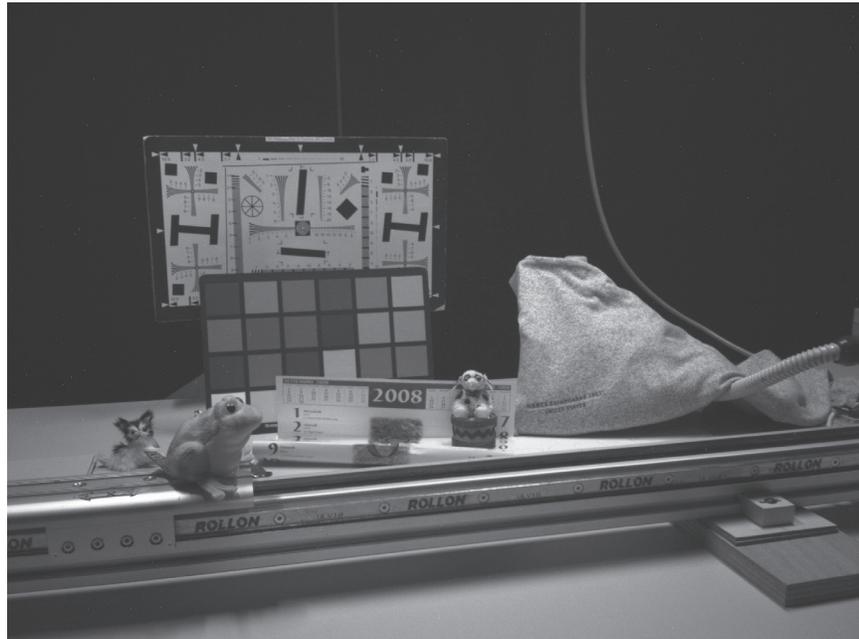
plements some specific camera applications: These functionalities are not mandatory and usually include solutions for panoramic, resizing, red-eye removal, etc. Some of them could also require the multiple acquisition of the input scene at different exposure and/or focus settings (e.g., bracketing). An example of Bayer image, acquired by the monochromatic sensor, and the corresponding RGB image, obtained at the end of the pipeline, is shown in Fig.(1.2) and in Fig.(1.3).

Other related info can be found on [1], that is mainly devoted to cover aspects relative to optics and sensors, and [2] that addresses specific research challenges and recent trends.

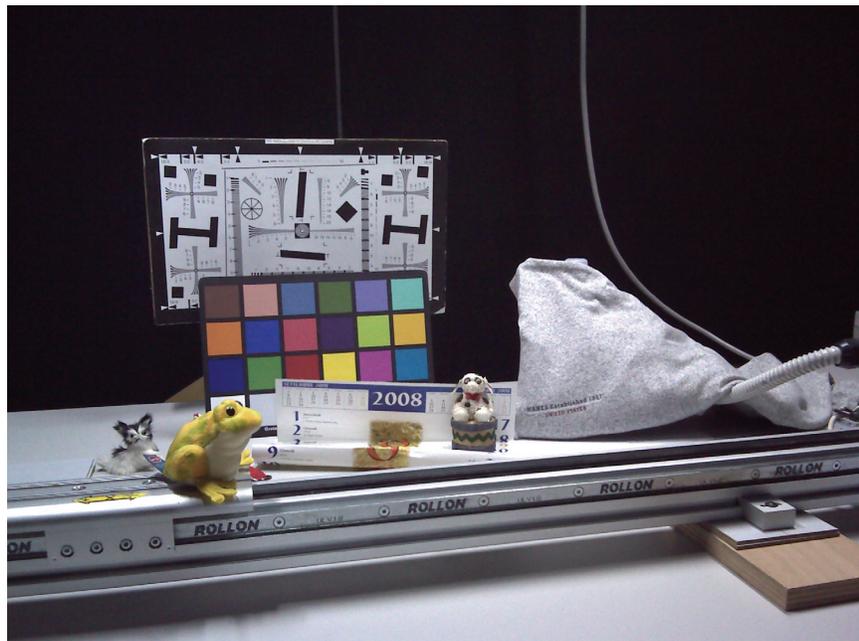
1.1.1 Exposure Setting

Like in old-fashioned film cameras, digital sensors need to be correctly exposed during acquisition. The pixel (picture element) is composed of an electronic device sensitive to the light (photo-diode or photo-transistor) which collects and translates incident photons (the electromagnetic element of the light) to electric signal. This signal is stored into an accumulation cell and, after an analog to digital conversion, represents the final pixel value (for detailed explanation see Chapter 2).

This basic light acquisition device has a few constraints: light sensitivity is fixed and



(a)



(b)

Figure 1.2 : An example of Bayer image (a) acquired by the monochromatic sensor and the corresponding RGB image (b) obtained at the end of the pipeline.

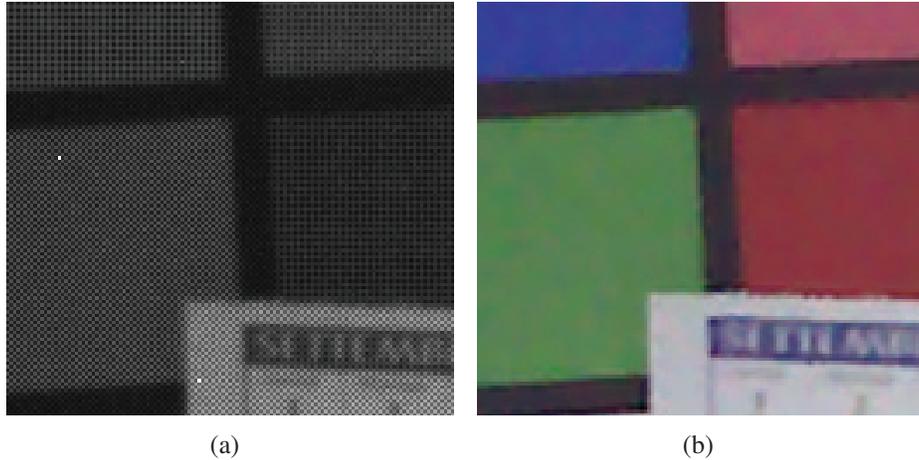


Figure 1.3 : An enlarged detail of Fig.(1.2) (a) acquired by the monochromatic sensor and the corresponding RGB image (b) obtained at the end of the pipeline.

it may be affected by noise (i.e., any kind of not actual information wrongly converted as useful information). Usually noise level is limited and not dangerous until the actual signal is adequate and significantly greater, i.e., high level of Signal to Noise Ratio (SNR).

To guarantee this fundamental principle each photosite (pixel) must be configured so that it acquires the correct level of light and thus the correct level of signal: varying the light intensity of the scene there must be a way to change the capability of the sensor to correctly and properly store in its cell the right level of light. This control is performed by the integration time. It represents the time during which the photo-element is acquiring and converting light into electrical charge. The lower the light intensity of the scene the higher the integration time. By changing this integration time a given scene digital acquisition can be under-exposed (too dark, too short integration time), over-exposed (too lit, too long integration time) or correctly exposed.

Two cases must be avoided or considered extreme cases: no accumulation, which corresponds to black, and over-accumulation (also known as saturation) which corresponds to extreme light or white. For actual black or white it is correct that the pixel assumes these values but they can also come out from a bad exposure (black from under-exposition and white from over-exposition). Also, there is no way to control integration time separately for each pixel of the sensor and this means that all the pixels of the sensor are exposed with the same integration time, although frame by frame it may change to adapt to variations of light which may occur in real scenes. Usually the integration time value is chosen so that the mean brightness of a picture is around the mid-range of the possible values (e.g., for a 8 bit per pixel image there are 256 different light levels and a correct exposed image has a mean brightness of 128).

Finally, it is not always possible to select an appropriate integration time for each scene. Too long or short integration time are not feasible because other problems may occur and affect the SNR (for details see Chapters 2, 3 and 6). Also, integration time may be lower-limited by the framerate and/or by a safe value which aims to reduce motion

blur effects. Motion blur is caused by long integration time and moving objects in the acquisition scene or hand-shaking. It is typical in low-light acquisitions and for this reason often a flash is used in such a situation.

Each time a lower threshold limits the integration time the only way to properly read the minimum accumulated information of the cell is to use a multiplicative gain to amplify the information to a usable value.

In summary, a good exposure control module is compound of:

- an appropriate module which estimates the light intensity of the scene and properly settles the correspondent integration time, avoiding under-exposition or over-exposition;
- an appropriate gain control which furnishes support and compensates the limits of the integration time; when selecting a proper balance between integration time and gains priority goes to the former;
- a method to identify actual black and white regions, assumed that the rest of the identification and proper compensation is demanded to following modules in the image generation pipeline (see Chapter 4 for additional details);
- a loop with other modules which apply additional gains to the signals (like AWB, see Chapter 5);
- and additional optional module to control and avoid motion blur; usually in literature this methodology goes with the name of AutoISO.

1.1.2 White Balance

One of the most challenging processes that affects perceived image quality in a digital camera is the correct color reproduction. Human visual system is able to remove color casts: an object appears to our eyes with the same color under different illuminant conditions. On the contrary the sensor simply acquires raw data and is not able to cope with real scene illumination variability. For instance a white paper in outdoor or indoor environment can be recorded by the sensor with bluish or reddish colors.

In order to cope with these problems a lot of techniques have been developed. High end cameras typically provide a variety of presets related to the most common light sources (tungsten, fluorescent, daylight, flash, etc.). Moreover white balance parameters can be set, for future photos, taking a picture of a known gray reference under the same illumination source (custom white balance).

All the techniques above described need a close interaction with the user in order to properly work. On the contrary auto white balance techniques try to guess the correct illumination properties and remove color casts without user interaction. These techniques, based on strong assumptions on scene reflectance distribution, have been also implemented in low cost devices (e.g., smart phone) and will be in depth described later in Chapter 5.

1.1.3 Noise Reduction

The perceived image quality is deeply influenced by image noise (named by analogy with unwanted sound). These unwanted fluctuations, if not properly managed, heavily degrade image quality. Different noise sources, with different characteristics, are superimposed to the image signal: photon shot noise, dark current noise, readout noise, reset noise, quantization noise, etc.

Although many efforts have been done by manufacturers to reduce the presence of noise in imaging devices it is still present and can be considered unavoidable in critical situations. For instance, low light conditions together with low integration time, produce very low SNR (signal to noise ratio), very few photon were captured, making really difficult obtaining pleasing photos. This physical limit does not depend only on the sensor characteristics but it is strictly related to the nature of light. Moreover the increasing of the number of pixels and the limited size of the embedded devices, implying the decreasing of the pixel size, produces further problems. Small pixels, acquires less photons with respect to larger pixels. Less useful signal implies then noisier picture.

In order to cope with these problems smart filters must be designed. These filters must be able to estimate image noise characteristics (e.g., mean and standard deviation if a Gaussian model is used), and then remove unwanted noise without affecting image details.

Finally, it should be noted that noise reduction can be performed during the various stages of the pipeline. Some approaches works on RGB images, others directly on Bayer data. The latter typically provides some advantages (demosaicing step typically introduces nonlinearities that make difficult noise reduction). Further details about noise reduction algorithms will be provided in Chapter 6.

1.1.4 Demosaicing

Digital cameras, in order to reduce costs and complexity, acquire images by means of a monochromatic sensor covered by a CFA (color filter array). A lot of CFA have been developed but the most common is the Bayer pattern. This simple CFA, taking into account human visual system characteristics (human eyes are more sensitive to green with respect to the other primary colors), contains twice as many green as red or blue sensors. Some spatially undersampled color channels (three in the Bayer pattern) are then provided by the sensor and the full color information is reconstructed by color interpolation algorithms (demosaicing). Demosaicing is a very critical task. A lot of annoying artifacts that heavily degrade picture quality can be generated in this step: zipper effect, false color, moiré effect, etc. Simple intra-channel interpolation algorithms (e.g., bilinear, bicubic) cannot be then applied and more advanced solutions (inter-channel), both spatial and frequency domain based, have been developed. In embedded devices the complexity of these algorithms must be pretty low. Demosaicing approaches are not always able to completely eliminate false colors and zipper effects, thus imaging pipelines often include a post-processing module, with the aim of removing residual artifacts. Further details about demosaicing algorithms will be provided in Chapter 7.

1.1.5 Color Matrixing

The Color Matrix sub-system, also known as Color Calibration, aims to convert the color response of the acquisition device to a standard color space. Usually the standard RGB (sRGB) color space is used, according to the ITU-R BT.709 directive. This transformation is needed since the spectral sensitivity function of the sensors does not match with the desired color space. The correction is performed usually according to the formula:

$$\mathbf{RGB}_{out} = \mathbf{A} \cdot \mathbf{RGB}_{in} \quad (1.1)$$

where \mathbf{A} is a 3-by-3 matrix, \mathbf{RGB}_{in} and \mathbf{RGB}_{out} the image before and after color matrixing. The matrix coefficients are not obtained using the effective response. Usually they are retrieved using optimization methods with real acquisitions. Moreover the constraint of the white point preservation is usually used. It corresponds to the following constraint (as better detailed in Chapter 5):

$$\sum_{j=1}^3 A_{(i,j)} = 1, \forall i \in \{1, 2, 3\} \quad (1.2)$$

1.1.6 Image Formatting

The data acquired by the sensor have to be processed by the coprocessor or the host microprocessor, so both the systems must share the same communication protocol and data format. Moreover, at the end of the image generation pipeline the image must be coded in a standard format in order to be read by any external device. Usually the sensor provides the acquired image in the Bayer format. In the past the Bayer data were stored and transmitted using proprietary format and protocol. Such solution has the drawback that every customer had to design the same proprietary interface to manage the sensor data. In the latest years the main companies making, buying or specifying camera modules proposed a new standard called Standard Mobile Imaging Architecture (SMIA). It allows interconnecting sensors and hosts of different vendors.

Concerning the output of the coprocessor, several standard formats are available. For the still images the most frequently used are the Joint Picture Expert Group (JPEG) with a lossy compression, the Targa Interchange Format (TIF) with a lossless compression. In the top level cameras the output of the sensors can also be stored directly. In this case usually a proprietary file format is used (e.g., the Nikon Electronic Image Format (NEF), the Canon RAW File Format (CRW), etc.). For videos the most used are Motion JPEG, MPEG-4, H.263 and H.264 standards.

In Chapter 11 the main data formats will be presented. Moreover some techniques concerning the compression factor control and the error concealment will be introduced. The compression factor control aims to obtain the file size as close as possible to a target value whereas the error concealment aims to handle errors in the bit-stream trying to recover the missing information.

1.2 HW/SW Partitioning

Cameras embedded in mobile phones are now becoming a commodity supporting applications like capturing and transmission of still images as well as video clips (Multimedia Messaging Services). With the increase of network bandwidth (e.g., 3G UMTS) real time mobile video links will become feasible, enabling new applications like mobile video telephony and video chat. It has to be noted, that the ease of use of these applications is of high importance as this is expected to be a crucial requirement for market acceptance of such new services. Thereby not only quality issues like frame and image stabilization are to be focused but also the user comfort. The automatic detection and tracking of the user's head is such an example, which helps to keep one's face in view of the camera during a mobile video telephone conference. But the processing units in imaging devices should be low-cost, low-power and, at the same time, suitable of supporting the above mentioned mobile communication applications. In order to satisfy cost and performance requirements, imaging device systems are generally implemented with a combination of different components, from custom designed accelerators to standard processors. These components can vary in their area, speed, methodology to program, and the system functionality is partitioned amongst the components to best utilize this tradeoff. However, for performance critical designs, it is not sufficient to only implement the critical sections as custom-designed high-performance hardware, but it is also necessary to pipeline the system at several levels of granularity. The custom designed accelerators can be implemented by using Reconfigurable hardware devices, such as Field Programmable Gate Arrays (FPGAs). The HW/SW partitioning (i.e., the definition of an architecture where the algorithms are smartly split as hardware accelerators and software modules) is not as straightforward as designing either software or hardware, since the application is intrinsically a hardware/software co-design. For instance, while an application implemented on an FPGA can be one to two orders of magnitude faster than the application implemented in software, processing in hardware incurs additional costs that are not required for software. Some of these costs are hardware initialization costs, extra processing steps for easy processing of the border cases, and communication of the image to and from the reconfigurable device. The runtime of image processing applications varies with image size, so processing small images on an FPGA might not be efficient due to the additional overhead. The imaging accelerators are often designed to create data-paths that are capable to process several image pixels concurrently. For the definition of these data path can be used some well-known design approaches like:

- SIMD parallelism. Typically the data-path processes N pixels in parallel or, for some binary operation, $8 \times N$ pixels. This type of processing is well known from multi-media extensions used in general-purpose CPUs.
- Deeper arithmetic pipelines. These enable the encoding and execution of complex arithmetic operations with a single microinstruction.

Bibliography

- [1] J. Nakamura, *Image Sensors and Signal Processing for Digital Still Cameras*. CRC Tailor & Francis, 2006.
- [2] R. Lukac, *Single-Sensor Imaging: Methods and Applications for Digital Cameras*. CRC Press, 2008.

Notions about Optics and Sensors

A. Bruna, A. Capra, M. Guarnera, G. Messina

Advanced System Technology - Catania Lab, STMicroelectronics, Italy.

Abstract: This Chapter gives information about the optics and the sensor in an image acquisition system. Optic is the first stage of the image acquisition system and is composed by one or more lenses aiming to concentrate the light in the physical sensor obtaining an in focus image. It is one of the most expensive parts of an imaging system. In mobile cameras usually a lens is compound of a system of plastic and glass lenses stacked together, while in single lens reflex cameras (SLR) a group of several glass lenses system is employed to reduce image artifacts. In this Chapter an overview of the lenses will be discussed. Moreover, some typical artifacts will be introduced (e.g., cross talk and chromatic aberration). The sensor is the part that converts the optical image (light) to an electric signal. There are several kinds of sensors depending on the technology (CCD - Charge Coupled Device and CMOS - Complementary MetalOxideSemiconductor), on the color filter array (Bayer , Foveon, 3CCD, Panchromatic), on the transducer function (LDR - low dynamic range, WDR - wide dynamic range and HDR - high dynamic range sensors).

2.1 Introduction

In Fig.(2.1) is reported the classical "yellow duck" system that describes the entire process from the real world, the system acquisition, the digital processing up to the final digitally compressed image. The first element, the lens and its working within the complete system, will be deeply studied in Chapter 4. This Chapter is mainly devoted to the explanation of the parts inside the box *Image Sensor*, the silicon sensor, its surface, its technology and its basic element: the pixel (picture element).

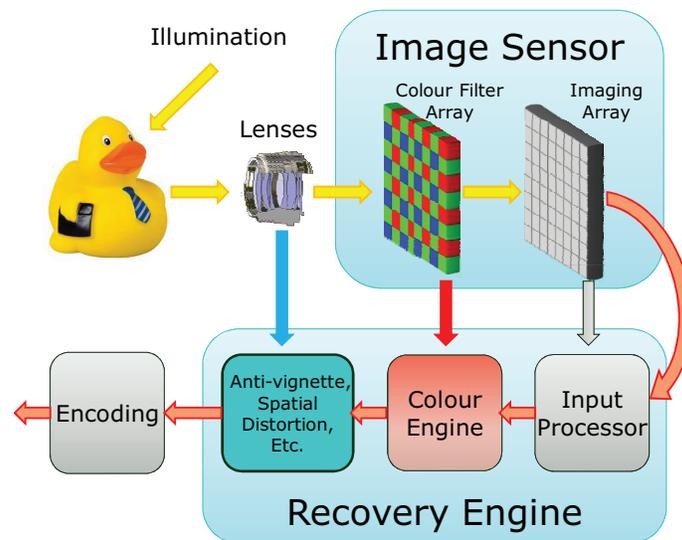


Figure 2.1 : The yellow duck system: from the real world to the final digital picture.

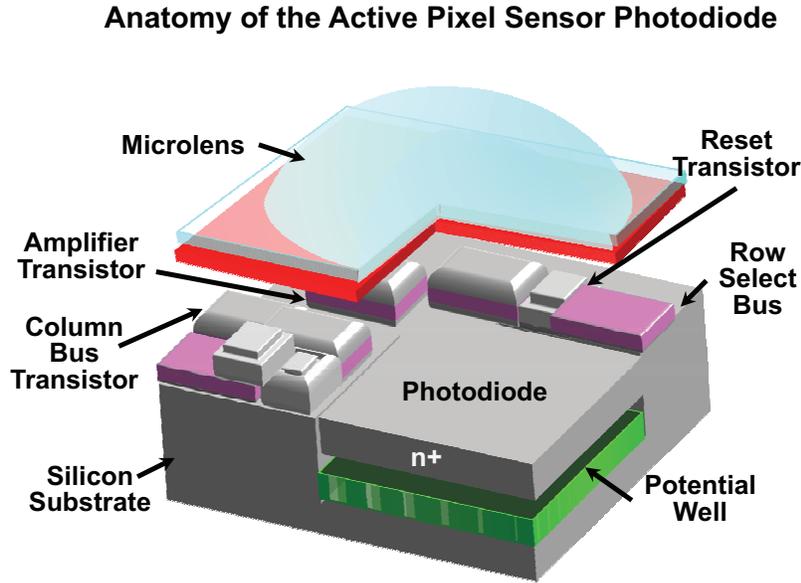


Figure 2.2 : Example of a pixel structure.

In the following Subsections will be explained the overall description of the sensor system, the different sensor technologies and the color patterns used in the real devices.

2.2 The Sensor System

In Fig.(2.2)¹ is reported a classical structure of a pixel inside a sensor. All the related compounding elements are visible: micro-lens, electronic part and photo-sensitive area. In the following Subsections all these elements are analyzed, describing their strongness and weakness and their contribute in the resulting final image.

2.2.1 Microlenses

One of the most important aspect of a imaging sensor is the sensitivity to capture the light (light sensitivity). As we will see in detail in Sec.2.3 it mainly depends by the design of the photo-sensitive area, Fig.(2.2), and, for equal design and technology, by the size of this area. The ratio between the photosensitive area and the pixel size is indicated as fill factor (it may vary from 30% to 100%). When this fill factor is too low and needs to be improved the microlenses are used. They are individual lenses deposited on the surface of each pixel to focus light on the photosensitive area. Microlenses can boost effective fill factor (i.e., the ratio of the active refracting area to the total area occupied by the lens array) to approximately 70%, improving sensitivity (but not charge capacity)

¹Image source: http://digitalcontentproducer.com/hdhdv/depth/cmos_tech_hdv_10092006

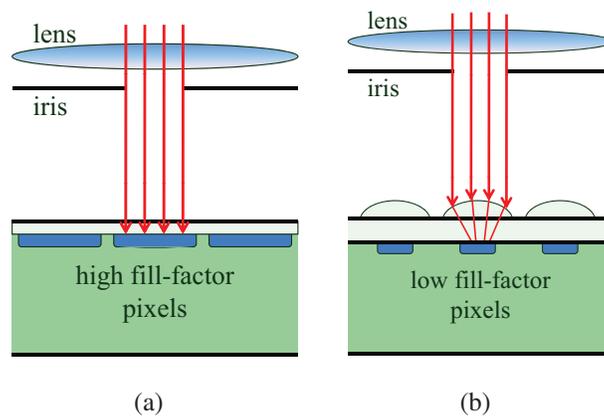


Figure 2.3 : Strategies to compensate different pixels fill factor: **(a)** Pixel with good fill factor, all the entering light is captured; **(b)** Pixel with poor fill factor, a micro-lens is used to converge the entering light to the useful region.

considerably. In Fig.(2.3)² is illustrated the difference of a pixel with and without micro-lens.

2.2.2 Lenses and Microlenses Aberrations

Crosstalk

In an ideal pixel charging [2], all the photons intersecting at any angle a filter element in the CFA are color filtered and accumulated in the photodetector under the filter element as shown in Fig.(2.4(a)). Optical crosstalk results when a photon, intersecting at an angle with a filter element in the CFA, enters the adjacent pixel's photodetector (photodiode) rather than the photodetector under the filter element. This can contaminate the adjacent pixel's charge packet, Fig.(2.4(b)). Moreover there may be also electrical crosstalk, which happens when photons passing through the red filter travel further into the silicon before generating electrons. This leads to a non-uniform response to the different colors, a loss of charge into the substrate and electrons wandering into the wrong pixel well as shown in Fig.(2.4(c)).

Chromatic Aberration

Chromatic Aberration (CA) is the term for an imaging system placing incorrect colors in at least some locations within the image. CA is caused by a lens having different refractive indices to different wavelengths of light. This implies the color channels to focus differently, and hence CA creates visible color fringes or colored blurs. An example of CA is shown in Fig.(2.5). Such aberration is typically diffused in the entire image, but

²Image source: [1]

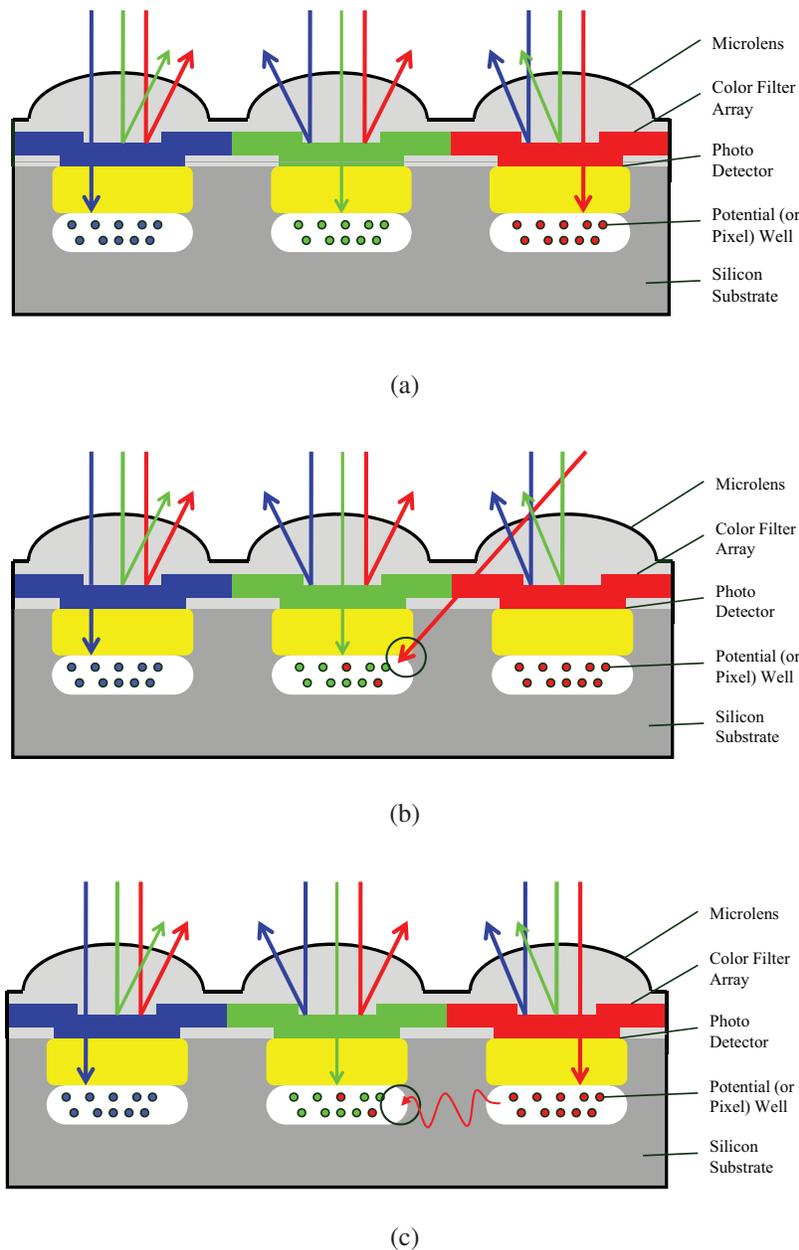


Figure 2.4 : Different types of crosstalk: **(a)** In this case all the entering light is correctly captured by the detector, there is not crosstalk effect; **(b)** Optical crosstalk: some filtered light is not captured by the respective detector but goes to an adjacent detector, in this case red light is collected in the green collector; **(c)** Electrical crosstalk: some photons are dispersed in the silicon and collected by a different detector, in this case a red photon travelling in the silicon is collected by the adjacent green detector.



Figure 2.5 : Particular of an image affected by CA.

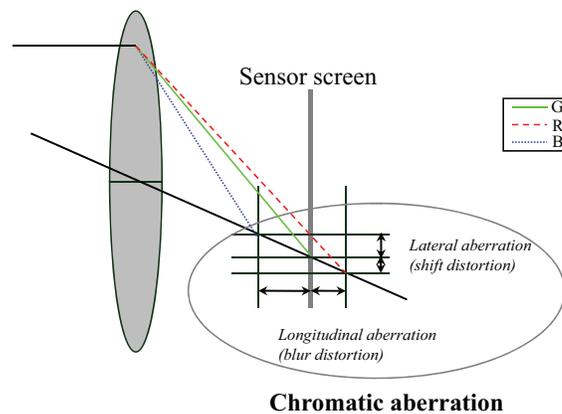


Figure 2.6 : Longitudinal and lateral aberrations.

it is mostly observed near high contrast edges. CA is usually classified in two types (see Fig.(2.6)):

- Longitudinal aberration;
- Lateral aberration (also known as aberration of magnification).

Longitudinal aberration causes loss of focus in the image plane according to the wavelength, and lateral aberration occurs due to the shift in the focus. The former causes blur distortion and the latter causes shift distortion of the color channels. Imaging devices usually focus the image plane based on the Green channel, as shown in Fig.(2.7), because it is a dominant factor for brightness. Therefore the Green channel exhibits little CA as compared to the other two color channels.

The two main methods for correcting chromatic aberration are:

- Lens design for improved optics;

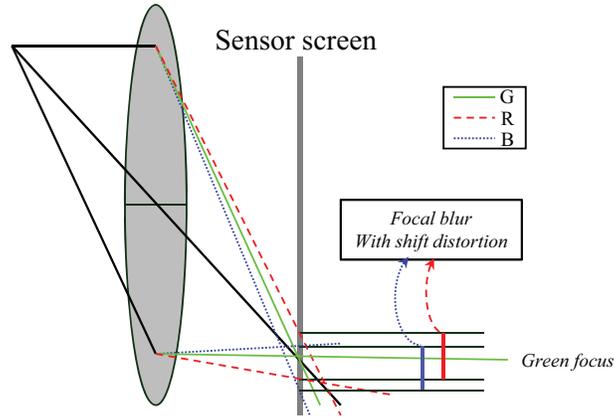


Figure 2.7 : Distortions of the R and B channels.

- Application of a signal processing method to the acquired image [3,4].

The first approach is generally expensive and difficult to apply. On the other hand, signal processing algorithms can correct this defect in a cheap and effective manner.

Simplest techniques measure the chromatic aberration of magnification on standard patterns, such as chessboard patterns. An example of pattern used for calibration is shown in Fig.(2.8) (left side). A second lower density chessboard pattern (test image), shown in Fig.(2.8) (right side), is used for independent validation of the results. In particular as discussed in [3], the reference information on the standard pattern without chromatic aberration is prepared in advance, then the standard pattern is actually photographed and the color shift is detected by comparing this data with the reference information .

Conventional image processing algorithms for correcting chromatic aberration usually make use of supplementary information about the optical system to calculate the color shift due to the chromatic aberration of magnification. The technique proposed in [4], tries to perform the correction of the lateral chromatic aberration by introducing differential geometric distortion into the color components of the image object. This distortion is used to substantially cancel the differences in image heights of the red, green and blue color components of the virtual image formed by the lens element. This is accomplished by determining one or more color-specific distortion functions (taking into account the different lateral magnification parameters of lens element for red, green and blue image planes) which are applied to the geometry of one or more color components. Differential distortion functions may be applied to the red and blue color components to align them with the green color component.

Purple Fringing

Although chromatic aberrations can be purple in color under certain circumstances, "Purple Fringing" usually refers to a typical imaging device phenomenon that is caused by the

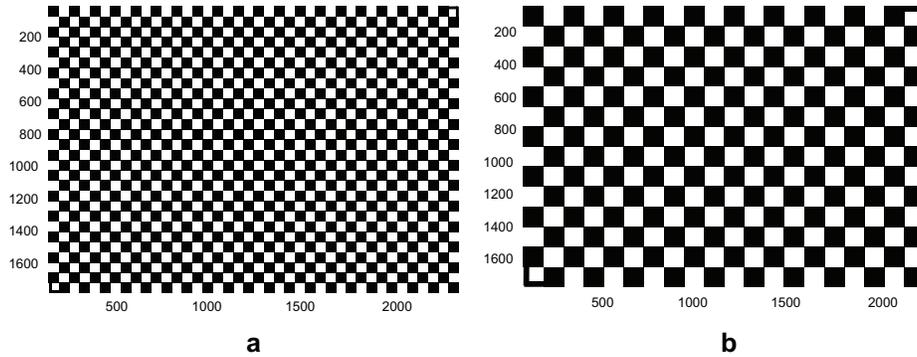


Figure 2.8 : Chessboard patterns used for calibration (left) and testing (right).



Figure 2.9 : Particular of an image affected by PF.

microlenses. It can be considered as "chromatic aberration at microlens level". As a consequence, purple fringing is visible throughout the image frame. An example of image affected by purple fringing is shown in Fig.(2.9). Edges of contrasty subjects suffer most, especially if the light comes from behind them. Blooming tends to increase the visibility of purple fringing. In fact imaging devices usually meter the exposure of the scene so that in the brightest regions as much charge is accumulated as possible without overflowing the potential well, so giving the best dynamic range and a response that depends directly on the amount of light striking the photodiode. The exposure control normally works well but the metering of the scene is done in some average sense. If the scene contains small regions where the light is extremely bright compared to the scene average, these bright regions lead to a very large amount of light being incident on the sensor. This light causes charge to build up in the sensor until a sufficiently large amount accumulates that it begins to leak out of the sensor well into the surrounding sensors (blooming effect).

When charge leaks from one photodiode well into surrounding photodiode wells, the result is a spuriously higher signal in the surroundings. That spurious signal will be particularly noticeable if the surrounding sensors should be producing no signal because

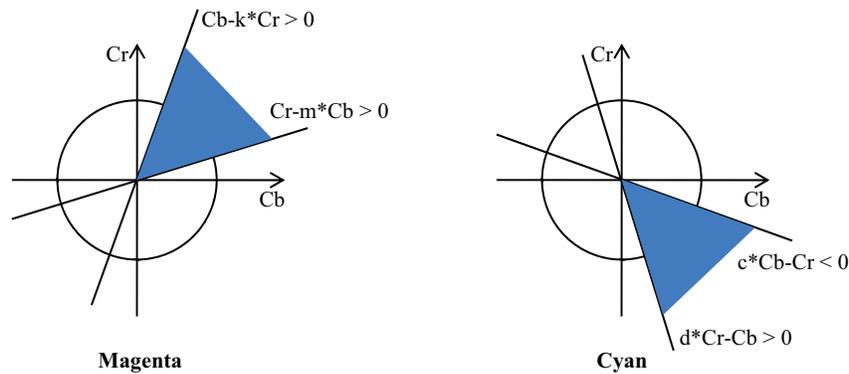


Figure 2.10 : Locus of purple in CMY color space.

the scene is dark in these regions. Is thus easier to see the effect of blooming most strongly at sharp transitions from light to dark. We have already seen that lens aberration will cause the blue and red components of bright white light to appear at incorrect sensor positions. Charge leakage magnifies this effect by spreading the sensor response further from the true position, and the positional error is additionally potentiated by demosaicing

Most of the state of the art algorithms devoted to correct the purple fringe effect construct a binary map, discriminating overexposed regions from non-overexposed ones. They detect purple regions close to overexposed areas and desaturate them. Some of these techniques try to prevent an excessive desaturation, according to the saturation of the surrounding pixels, to provide high quality images. Some other techniques try to identify the occurrence of purple fringing by detecting magenta and/or cyan color ranges in the $Cr Cb$ domain, as shown in Fig.(2.10).

An interesting technique [5] for removing purple fringing assumes purple fringe appearing principally around an overexposed highlight, then the false color degree of each pixel is calculated, and appropriate correction is performed corresponding to these false color degrees. A pixel which satisfies the following conditions:

- an overexposed pixel exists around the noticed pixel;
- the noticed pixel exhibits a purple color;
- the noticed pixel has a high saturation.

is determined as a purple fringe pixel. The correction process can be executed by averaging neighboring pixels which do not exhibit a false color.

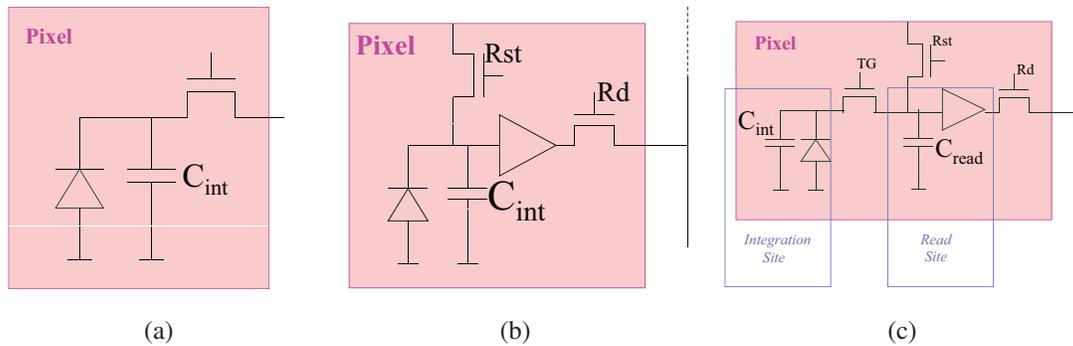


Figure 2.11 : Structure of a pixel and its evolution; **(a)** pixel in its basic structure: photodiode and accumulation cell; **(b)** active pixel, an on site amplification has been added; **(c)** active pixel in 4-T configuration: accumulation and reading phases are separated.

2.3 The Sensor Technology

Imaging sensor functionality for digital camera is the same of a film for old camera. It is used to convert light entering from the lens system to electrical signal which is properly processed and stored at the end of the pipeline in a standard image format file. The basic structure of a pixel is reported in Fig.(2.11(a)): it is compound of a photodiode to convert light to an electrical signal and an accumulation cell (capacitor) and a switch. This configuration is known as passive pixel. The advantage of this configuration is the maximization of the fill factor.

Adding a signal amplification to the pixel, it is possible to have an active pixel, as shown in Fig.(2.11(b)). The amplification system allows to improve the signal level during the reading process, thus improving the SNR. It is also known as 3-T (3 transistors) configuration. The charge to voltage conversion within the pixel improves output impedance but it causes a lower fill factor, because now part of the pixel surface is dedicated to two switches and an amplifier. A sensor compound of such a kind of pixels goes by the name of Active Pixel Sensor (APS) [6].

Nevertheless this configuration has the accumulation and the reading stages not separated and thus, during the reading process, noise coming from the accumulation part is amplified and transmitted. For this reason a widely used configuration is the 4-T configuration, as shown in Fig.(2.11(c)). In this case accumulation and reading are separated, improving the final SNR.

2.3.1 CCD and CMOS Sensors

First imaging sensors were introduced around 1970 and were based on CCD (Charge Coupled Device) technology. As for CMOS sensor they are silicon based and share similar materials. However CCD sensors have been specifically designed to capture images and have 40 years of development. Until a few years ago they were the state of the art as

imaging sensors (Table 2.1) having excellent performances in terms of image quality, low noise, high fill factor, high light sensitivity and low dark current³ [1, 7]. Differently from CMOS devices, CCD requires dedicated clock drivers with high voltage amplitude (10V), high bias voltages (up to 15V) and has the disadvantage that pixels are read sequentially, as shown in Fig.(2.12). This also forced developers to split sensor from the image coprocessor (Fig.(2.13)) with high cost due to a different fabrication technology (often a sensor maker is not a coprocessor maker) and bigger dimension of the CCD sensor (with the consequence that it requires more space into the device).

For this reason at the end of 1990 decade chip makers started to realize first CMOS image sensors, lower in quality than a CCD sensor and thus mainly dedicated to low cost market segments [1, 7, 8]. However in the last years this trend is changed and now CMOS sensor have a quality vs cost ratio greater than a CCD and many professional SLR use CMOS sensors.

Table 2.1 : CCD and CMOS performances.

CCD	CMOS
Low dark current	Low power dissipation
Uniform conversion gain	Scaling to high resolution formats
Low temporal noise	Compatibility with camera system integration
Low Fixed Pattern Noise	Low cost
Quantum efficiency	
Low Cross Talk	
Colorimetric accuracy	
Signal to Noise Ratio	

Let's have a quick difference between the CCD and the CMOS image sensor.
Charge-Coupled Device:

- The charge is actually transported across the chip and read at one corner of the array;
- Use of a special manufacturing process to create the ability to transport charge across the chip without distortion;
- Higher Fill Factor.

Complimentary Metal-Oxide Semiconductor:

- Several transistors at each pixel amplify and move the charge using more traditional wires;

³The term dark current indicates a residual current that pass through the photo-sensitive area also in absence of light. This is stored into the charge cell and produces false signal, i.e., noise, highly visible in low light conditions.

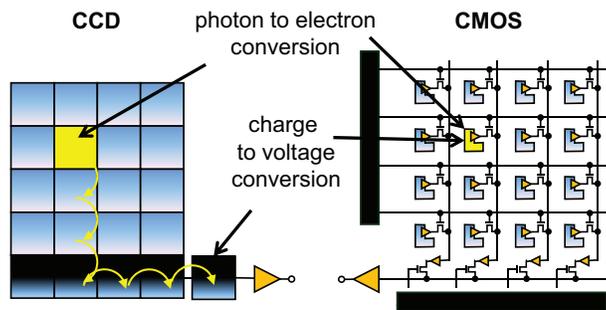


Figure 2.12 : Comparison of the reading mode: by charge transfer (CCD) and by address direct access (CMOS).

- Is more flexible because each pixel can be read individually;
- Use of the same traditional manufacturing processes to make most microprocessors;
- Easy integration;
- Lower Fill Factor.

Fig.(2.12)⁴ shows the different reading mode between a CCD and a CMOS. In the former data is read by transferring the pixel value to adjacent pixels in column wise order until the end of the column is reached, in the latter pixel value is directly read by addressing actual pixel location. This means that a broken pixel in CCD sensor may affect the readout process of an entire column while in a CMOS sensor it affects only its value leaving safe the rest of the column. Also, the reading process in a CCD is "sequential" and thus slower than the correspondent "parallel" process of the CMOS. Also, CCD sensor technology is different from the technology used in the image signal processor (usually CMOS) and the sensor itself is larger than a CMOS sensor, while a CMOS sensor uses the same technology on the processor. As a result, as shown in Fig.(2.13)⁵, CMOS sensor and processor can be placed together saving space, transmission problems and circuits.

3CCD

To take advantage of a full color acquisition some systems use three sensor, one for each color component so that for each pixel all color components are acquired. A prism is used to send the input image from the optical system to the sensors. For best image quality and ease of use, separation prisms should have a few simple characteristics:

- All output images should be oriented in the same direction as the input image;

⁴Image source: [1]

⁵Image source: [1]

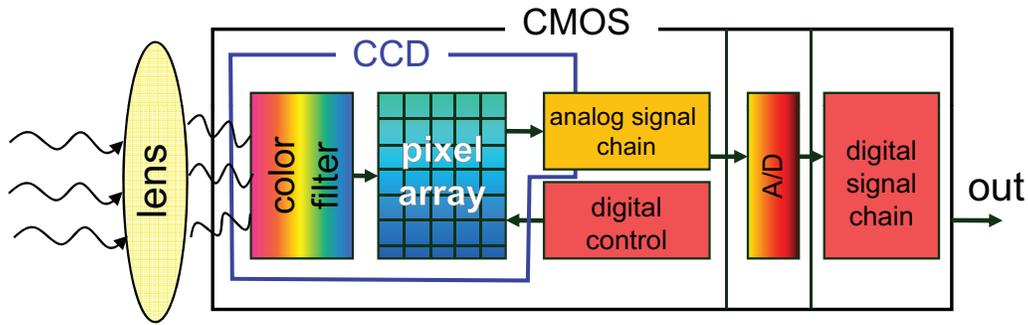


Figure 2.13 : Sharing the same technology, CMOS sensor and processor can be realized in a single chip while CCD sensor must be kept separate from the processor.

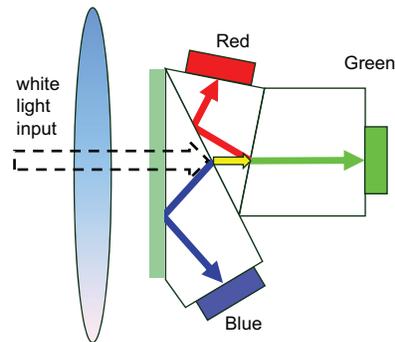
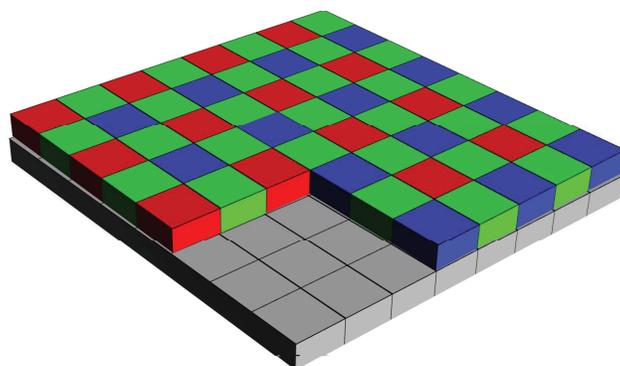


Figure 2.14 : Structure of a 3CCD device: input image is sent to three sensors, one for each color sensor.

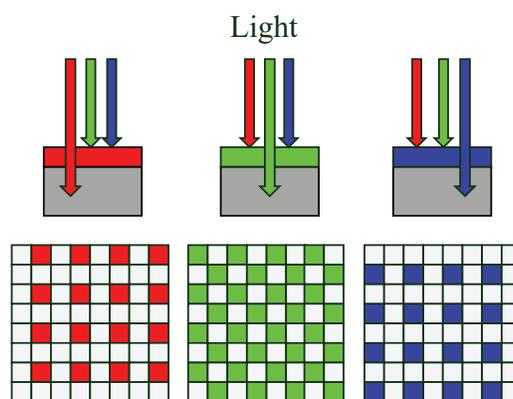
- All channels must have the same optical path length;
- The prism transmission should handle all polarizations with good uniformity;
- All coatings should be protected from the environment;
- Ample space should be available for mounting of filters and sensors.

This kind of system is often used in professional video recorder. In Fig.(2.14)⁶ is reported a classical structure of a 3CCD acquisition system. Note how the prism is designed so that each sensor is virtually placed at the same distance from the lens.

⁶Image source: [1]



(a)



(b)

Figure 2.15 : Structure of the Bayer Colour Filter Array: (a) CFA on the sensor surface; (b) Light split in colour components and affecting different pixels of the sensor.

2.3.2 The CFA and the Bayer Pattern

Each "pixel" on an imaging device sensor contains a light sensitive photo diode which measures the brightness of light. Because photodiodes are monochrome devices, they are unable to select the difference between different wavelengths of light. Therefore, a "mosaic" pattern of color filters, a color filter array (CFA), is positioned on top of the sensor to filter out the red, green, and blue components of light falling onto it. Although GRGB Bayer Pattern is the most common CFA used since 1976, many investigation have been done to define more performing CFAs [9]. Fig.(2.15)⁷ represents a classical Bayer CFA [10] and how color components are acquired by the sensor.

⁷Image source: http://www.dpreview.com/learn/?/Glossary/Camera_System/sensors_01.htm

The Panchromatic Sensor

Recently Kodak, already inventor of the Bayer format [10], has proposed a new format, the panchromatic sensor [11]. The advantage of this sensor is that 50% of the pixels are white (there is not a color filter on top of it) making them sensitive to luminance and useful in low light. Another important benefit of this structure is that a 2x2 macroblock of Bayer corresponds to a 4x4 block (Fig.(2.16))⁸, thus in each block there are more green, blue and red pixels and, in addition, white pixels which are useful to reconstruct light intensity.

To understand the improvement of the sensitivity in low light of the panchromatic sensor we can express light energy, split from RGB components according to ITU-R BT.601 by the equation:

$$Y = 0.299R + 0.587G + 0.114B \quad (2.1)$$

Applying the above equation to a CFA Bayer macroblock:

$$(0.299 + 2 \times 0.587 + 0.114)/4 = 0.396 \quad (2.2)$$

which is less than 40% of efficiency; about 60% of photons are lost and absorbed in filters.

Applying the same equation to a panchromatic macroblock:

$$(8 \times 1 + 2 \times 0.299 + 4 \times 0.587 + 2 \times 0.114)/16 = 0.698 \quad (2.3)$$

which means that about 70% of light is captured by the sensor. As a consequence a panchromatic sensor is 1.76 times more efficient than a Bayer sensor in terms of light efficiency. Of course light efficiency is not the only point to be analyzed when comparing different sensor patterns. Another interesting point is the image processing pipeline. Most of the camera sensors in the market adopt the Bayer format. This means that it has become a standard *de facto*. New formats, even if have some advantages, have the inconvenient of requiring a proper image processing which differs from the classic one. In Fig.(2.17) is reported the pipeline patented by Kodak in [11]. It differs from a common pipeline, for example, for the automatic exposure control, noise reduction, demosaicing (even if some methodologies from Bayer may be reused), color accuracy, aliasing and sharpening.

The Foveon Sensor

Foveon X3 [12] image sensors have three layers of pixels. The layers of pixels are embedded in silicon to take advantage of the fact that red, green, and blue light penetrate

⁸Image source: <http://pluggedin.kodak.com/default.asp?item=624876>

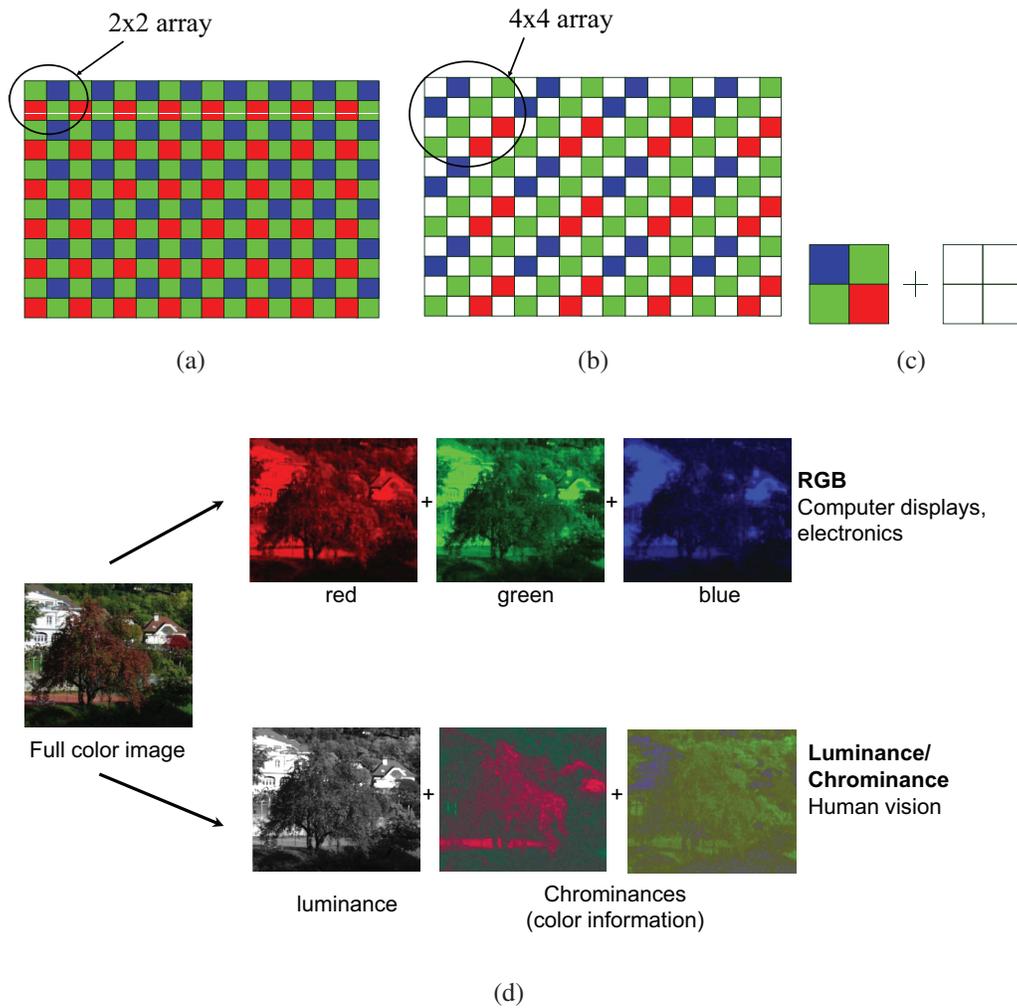


Figure 2.16 : Panchromatic sensor structure: **(a)** classical Bayer CFA sensor surface; **(b)** new Panchromatic surface, 50% of the surface is dedicated to white pixels; **(c)** equivalent panchromatic macroblock is compound of a classical RG_rG_bB macroblock plus a white macroblock; **(d)** color image decomposition, panchromatic sensor produces an image similar to human vision.

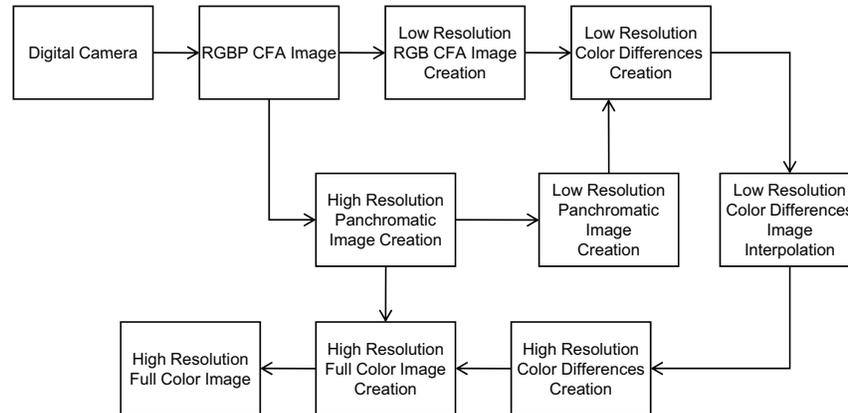


Figure 2.17 : Panchromatic image generation pipeline as patented by Kodak [11].

silicon to different depths forming the first and only image sensor that captures full color at every point in the captured image. Fig.(2.18)⁹ shows the different penetration of the light through the sensor and the relative displacement of the three color planes on top of the sensor surface. As a consequence this sensor does not require color demosaicing and, compared to a Bayer sensor, the number of pixels is sensibly reduced for the same image quality (1 Foveon = 4 Bayer pixels).

This triple vertical junction 'Foveon type' allows pixels to be bigger and so collect more light. On the other hand this stack can easily generate color crosstalk which can limit SNR performance. In fact blue channel has large green and red signals penetration, green channel has large red signal penetration while red channel is the only channel with low crosstalk. Image processing for Foveon sensors needs a strong crosstalk digitally removing and dedicated color processing to restore color accuracy. As a consequence the process control needs particular attention, as junction depth and doping highly affects color accuracy.

2.3.3 Different Dynamic Range Sensors

Real world scenes can contain a very wide dynamic range, as briefly summarized in Table 2.2 (note that the lux is the standard unit of luminous emittance). Usually normal sensors are not able to acquire a real world scene, especially if the dynamic range is very high, as can be an outdoor scene with mixed sunny and shadow zones. Measuring the dynamic range (DR) in dB with the formula shown below, these values can easily go above 60 dB in the real scenes.

$$DR(dB) = 20 \cdot \log_{10} \frac{\max Lux}{\min Lux} \quad (2.4)$$

⁹Images source: <http://www.foveon.com>

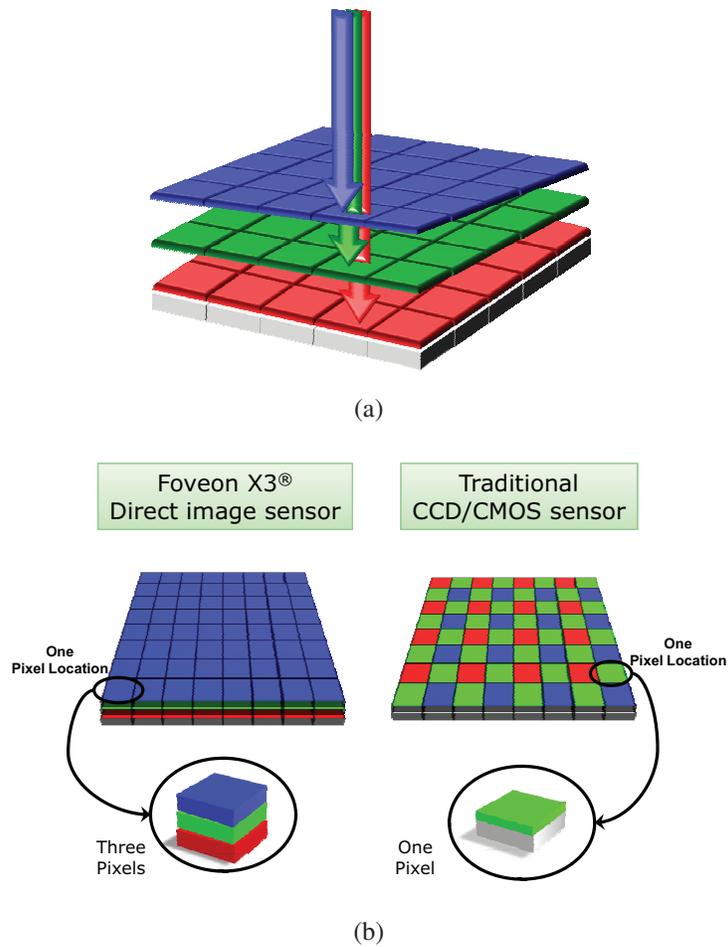


Figure 2.18 : Structure of the Foveon sensor. (a) color planes are stacked together; (b) comparison between a Foveon sensor and a CCD/CMOS CFA Sensor.

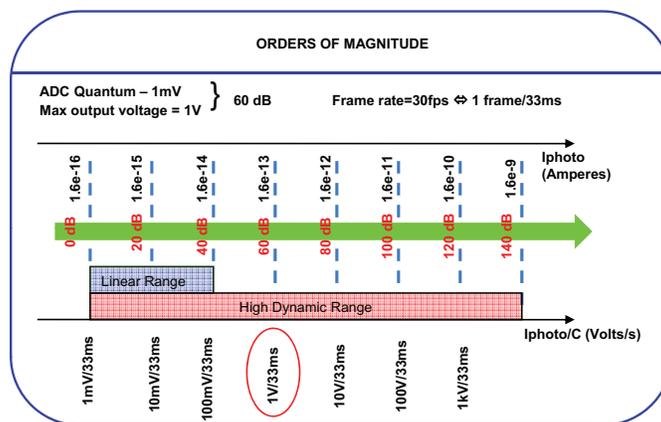


Figure 2.19 : Real world dynamic ranges expressed in dB and LDR and HDR sensor families range.

Table 2.2 : Different light intensity in real world scenes.

New Moon Night	10-5 Lux
Full Moon Night	0.2 Lux
Street Lighting	20 Lux
Bedroom	70 Lux
Classroom, laboratory	400 Lux
Offices	500 Lux
Drawing Table	1000 Lux
Television broadcasting	800 to 1100 Lux
Sun at zenith, summer sunny day	105 Lux

where DR is the dynamic range (in dB), $maxLux$ and $minLux$ are the highest and the lowest light intensity values in the scene measured in Lux (i.e., the SI unit of illuminance). Normal sensors are commonly designed to have a linear response and a useful dynamic range (the maximum dynamic range able to acquire) of around 60 dB. To design sensors capable of acquiring a wider dynamic range, designers have been forced to change the response function of the sensor from linear to non-linear (usually logarithmic), thus implementing a compression of the dynamic range directly during the storing phase in the sensor. In Fig.(2.19) is represented a comparison between a normal sensor, also known as Low Dynamic Range (LDR) sensor, and a High Dynamic Range (HDR) sensor. In this figure dB values represent the ratio between the minimum eV charge which can be stored per second in the accumulation cell of the pixel and the maximum value. The dynamic range of the HDR is 80dB higher than the LDR. Between these two families there is another one, named Wide Dynamic Range (WDR), characterized by the fact that the sensor response function is still linear but wider than a LDR sensor.

2.4 Pixel Size Reduction

Forty years ago, the co-founder of Intel, Gordon Moore, predicted that the speed of processors would double every eighteen months. Although the statement is always at the center of heated discussions among computer scientists, it has been proved to be true and for this reason it has been named the "Moore's Law". As the technology has improved, costs have decreased dramatically. Counting the "pixels per dollar" as a basic measure of value for a digital camera, there has been a continuous and steady increase in the number of pixels each dollar buys in a new camera, in accord with the principles of Moore's Law. This predictability of camera prices was first presented in 1998 at the Australian PMA DIMA conference by Barry Hendy and since referred to as "Hendy's Law". In Fig.(2.20) an overview of the evolution of CMOS imaging sensors data, published at IEDM and ISSCC [13], is shown¹⁰. The bottom curve illustrates the CMOS scaling effects over the years, as described by ITRS roadmap [15]. The second curve shows the technology node used to fabricate the reported CMOS image sensors, and the third curve illustrates the pixel size of the same devices. It should be clear that:

- CMOS image sensors use a technology node that is lagging behind the technology nodes of the ITRS. The reason for this is quite simple: very advanced CMOS processes, used to fabricate digital circuits, are not imaging friendly (issues with large leakage current, low light sensitivity, noise performance, etc.).
- CMOS image sensor technology scales almost at the same pace as standard digital CMOS processes do.
- Pixel dimension scales with the technology node used, and the ratio is about a factor of 20 [14].

The CMOS imager replaced the CCD by scaling to a lower cost and higher performances. The reasons for the continued scaling the CMOS devices are as follows: increase customers perceived value by increased number of pixels; reduce cost; survive by continued innovation; crush competitors by rendering their products obsolete; increase imager spatial resolution.

The present state of the art of CMOS imaging sensor has pixel size of $1.75 \mu\text{m} \times 1.75 \mu\text{m}$ for Front-side pixel illumination (FSI) technology and $1.4 \mu\text{m} \times 1.4 \mu\text{m}$ for Back-side pixel illumination (BSI) technology¹¹. In Fig.(2.21) is shown the difference between such technologies. Recently Sony has declared the prototyping of a new CMOS sensor 10 million pixels with $0.9 \mu\text{m}$ pixel pitch [17].

The reduction of pixel pitch has unfortunately increased the problem of crosstalk (see Chapter 2.2.2). Crosstalk occurs because in digital photography separation of the colors of light and the actual sensing occur in different points of the sensor and at significant distances from each other. In CMOS image sensors, incoming photons go through color filters, then through glass layers supporting the metal interconnect layers, and then into

¹⁰This figure has been taken from [14] which is full property owner of its copyrights.

¹¹The OmniBSI technology has been introduced by first by OmniVision [16].

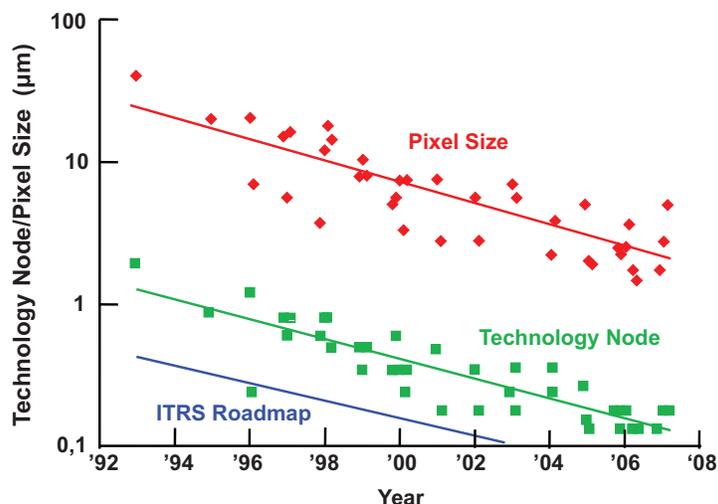


Figure 2.20 : Evolution of CMOS imaging sensors. The ITRS curve: roadmap for CMOS scaling effects [15]. The Technology Node curve: size of the technology used to fabricate CMOS image sensors. The Pixel size curve: Pixel pitch size of the CMOS imaging devices.

silicon, where they are absorbed, exciting electrons that then travel to photodiode structures to be stored as signal. Along the way, the photons and electrons have many opportunities to be diverted and end up at the wrong photodiodes, contributing to the wrong color signal. As market pressure drives down the pixel size in cell phone cameras, the optical crosstalk problem becomes more severe because the decrease in pixel pitch separates light of different colors on an ever smaller scale. A CMOS image sensor photodiode typically occupies 20 to 40 percent of the pixel silicon surface area because of the need to accommodate pixel transistors.

The microlens was introduced originally to improve the sensitivity of a sensor by directing more light from the pixel periphery toward the photodiode, thus increasing the "effective fill factor" (i.e., the fraction of light that hits the silicon on or close to the photodiode and thus has a chance to contribute to the signal). By the same token, a microlens may reduce the amount of light that hits a photodiode after crossing a filter associated with its neighbor. One problem with microlenses is that some of the light can enter gaps between the microlenses and be scattered by the irregular concave surfaces. This light contributes a lot to color crosstalk and almost nothing to sensitivity.

Other than microlenses, other factors can contribute to crosstalk. For example, crosstalk can be increased (and sensitivity lowered) because the light passage in the interconnect glass is curved by diffraction and by scattering around the absorbing-reflecting metal lines and vias. Traditional pixel design pushes the lines as far as possible outside the photodiode into the pixel periphery to minimize their optical influence. However, increasing some line widths can help reduce crosstalk by creating a shielding effect around the photodiode that lets light from the correct color filter through but blocks light crossing from a neighboring filter. If this metal shielding design is used, it should balance the crosstalk

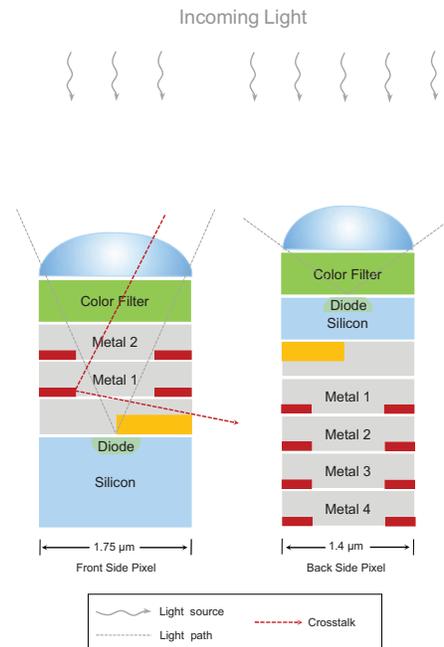


Figure 2.21 : Front-side pixel illumination (FSI) technology versus Back-side pixel illumination (BSI) technology.

decrease with the inevitable sensitivity decrease.

The newest CMOS image sensors have a pixel pitch of $1.75\ \mu\text{m}$, are designed with two or three metal interconnect layers (versus three to five layers in the older 3.5 to $6.0\ \mu\text{m}$ pixel sensors), and have an aspect ratio of approximately two. These characteristics produce an acceptable final image color quality. However, it must be reduced further for even smaller pixels to become viable. The reduction in pixel size with no image color quality deterioration is becoming ever more difficult as the light is being pushed into photodiodes that are smaller and spaced closer than the wavelength itself.

A $1.75\ \mu\text{m}$ pixel in a modern image sensor might get only 30 to 100 electrons of photo-signal for images taken with 100 lux illumination. If the charge transfer is incomplete and leaves just 30 electrons on the photodiode, a major image lag in the sensor results. Furthermore, the 30 electrons left are usually noisy and change from pixel to pixel.

Measuring $1.75\ \mu\text{m}$, FSI pixels are larger than BSI pixels. Consequently, they require certain camera components, particularly the length of the lens, to be larger. In the race for less space, every millimeter and micron is precious. The OmniBSI architecture takes the FSI topology and reverses the arrangement of layers, situating the metal and dielectric layers under the sensor array (Fig.(2.21)). Instead of passing through the metal layers, light hits the silicon layer directly without interference. The first advantage of this approach is that light entering the sensor takes the shortest path to the detector, through the color filter only. There are no metal layers or transistors to block or reflect light. Since light strikes the silicon directly, the sensors fill factor significantly improves, which in turn boosts low-light sensitivity dramatically. As this arrangement optimizes light absorption,

it most notably creates a 1.4- μm BSI pixel, which OmniVision claims surpasses all the performance metrics of 1.4- μm and most 1.75 μm FSI pixels.

The BSI approach has been around for a while. But due to cost issues, its application has been predominantly in the military and avionics fields. Although backside illumination concepts have been studied for over 20 years, up until now nobody has been able to successfully develop the process for commercial, high-volume CMOS sensor manufacturing. Tessera has estimated that even allowing for economics of scale (more die per wafer), mass-produced back-illuminated die will cost twice to four times as much as front-illuminated die with similar resolution. Yet the payoff is what is estimated as five times improvement in performance overall.

Bibliography

- [1] DALSA, “Image sensor architectures for digital cinematography.” DALSA Digital Cinema, White Paper.
- [2] K. Hirakawa, “Cross-talk explained,” in *Proceedings of IEEE International Conference on Image Processing (ICIP08)*, pp. 677–680, 2-15 Oct. 2008.
- [3] J. Mallon and P. F. Whelan, “Calibration and removal of lateral chromatic aberration in images,” *Pattern Recognition Letters*, vol. 28, no. 1, pp. 125–135, 2007.
- [4] E. C. Haseltine and W. G. Redmann, “Electronic and computational correction of chromatic aberration associated with an optical system used to view a color video display,” *U.S. Patent 5369450*, November 1994.
- [5] T. Masuno, M. Ohki, and R. Yamada, “Image processing apparatus and image processing method as well as computer program,” *U.S. Patent 7529405*, May 2009.
- [6] O. Yadid-Pecht, “Active pixel sensor (APS) design from pixels to systems.” Lecture, 2008.
- [7] N. B. Zurich, “CCD versus CMOS – has CCD imaging come to an end?,” 2001.
- [8] G. J. Barlow, “CMOS active pixel sensor for a polarization-difference camera,” in *NSF Technical Report TR-CST18SEPT01*, pp. 127–145, University of Pennsylvania, 2001.
- [9] K. Hirakawa and P. J. Wolfe, “Spatio-spectral color filter array design for optimal image recovery,” *IEEE Transactions on Image Processing*, vol. 17, pp. 1876 – 1890, Oct. 2008.
- [10] B. Bayer, “Color imaging array,” *U.S. Patent 3,971,065*, 1976. Eastman Kodak Company.
- [11] J. F. J. Hamilton and J. T. Compton, “Processing color and panchromatic pixels,” *U.S. Patent Application 20070024879*, July 28 2005. Eastman Kodak Company.
- [12] P. M. Hubel, J. Liu, and R. J. Guttosch, “Spatial frequency response of color image sensors: Bayer color filters and foveon X3.” Foveon, Inc. White Paper.
- [13] “Digest of technical papers from 1990 till 2007.” International electron devices meeting and international solid-state circuits conference.
- [14] A. J. P. Theuwissen, “CMOS image sensors: State-of-the-art.,” *Elsevier - Solid-State Electronics*, no. 52, pp. 1401 – 1406, 2008.
- [15] “Executive summary - 2007 edition,” 2007.

- [16] M. Okincha, "OmniBSI TM technology backgrounder," tech. rep., OmniVision, June 2009. http://www.ovt.com/technologies/bsi_technology_backgrounder.pdf.
- [17] K. Itonaga, K. Mizuta, T. Kataoka, M. Yanagita, S. Yamauchi, H. Ikeda, T. Haruta, S. Matsumoto, M. Harasawa, T. Matsuda, A. Matsumoto, I. Mizuno, T. Kameshima, I. Sugiura, T. Umebayashi, K. Ohno, and T. Hirayama, "0.9um pitch pixel cmos image sensor design methodology," in *International Electron Devices Meeting*, (Baltimore), IEEE/IEDM, 2009.

Exposure Correction

A. Castorina, G. Messina

Advanced System Technology - Catania Lab, STMicroelectronics, Italy.

Abstract: The problem of the proper exposure settings for image acquisition is of course strictly related with the dynamic range of the real scene. In many cases some useful insights can be achieved by implementing ad-hoc metering strategies. Alternatively, it is possible to apply some tone correction methods that enhance the overall contrast of the most salient regions of the picture. The limited dynamic range of the imaging sensors doesn't allow to recover the dynamic of the real world. In this Chapter we present a brief review of automatic digital exposure correction methods trying to report the specific peculiarities of each solution. Starting from exposure metering techniques, which are used to establish the correct exposition settings, we describe automatic methods to extract relevant features and perform corrections.

3.1 Introduction

One of the main problems affecting image quality, leading to unpleasant pictures, comes from improper exposure to light. Beside the sophisticated features incorporated in today's cameras (i.e., automatic gain control algorithms), failures are not unlikely to occur. Digital consumer devices make use of ad-hoc strategies and heuristics to derive exposure setting parameters. Typically such techniques are completely blind with respect to the specific content of the involved scene. Some techniques are completely automatic, cases in point being represented by those based on average/automatic exposure metering or the more complex matrix/intelligent exposure metering. Others, again, accord to the photographer a certain control over the selection of the exposure, thus allowing space for personal taste or enabling him to satisfy particular needs. In spite of the great variety of methods for regulating the exposure and the complexity of some of them, it is not rare for images to be acquired with a nonoptimal or incorrect exposure. This is particularly true for handset devices (e.g., mobile phones) where several factors contribute to acquire bad-exposed pictures: poor optics, absence of flashgun, not to talk about difficult input scene lighting conditions, and so forth.

There is no exact definition of what a correct exposure should be. It is possible to abstract a generalization and to define the best exposure that enables one to reproduce the most important regions (according to contextual or perceptive criteria) with a level of gray or brightness, more or less in the middle of the possible range. In any case if the dynamic range of the scene is sensibly "high" there is no way to acquire the overall involved details. One of the main issues of this Chapter is devoted to give an effective overview of the technical details involved in:

- Exposure settings of imaging devices just before acquisition phase (i.e., preprocessing phase) [1];
- Content-dependent enhancement strategies applied as post-processing [2];

- Advanced solution where multi-picture acquisition of the same scene with different exposure time allows to reproduce the Radiance map of the real world [3].

The rest of the Chapter is organized as follows. The initial Section will discuss in details traditional and advanced approaches related to the pre-processing phase (i.e., the sensor is read continuously and the output is analyzed in order to determine a set of parameters strictly related with the quality of the final picture [1]). The role of exposure setting will be analyzed also considering some case studies where, by making use of some assumptions about the dynamic range of the real scene, it is possible to derive effective strategies. Section 3.3 will describe the work presented in [2] where by using post-processing techniques, an effective enhancement has been obtained just analyzing some content dependent features of the picture. The presentation follows in part the structure of our recently published work on the same subject [4].

3.2 Exposure Metering Techniques

Metering techniques built into the camera are getting much better with the introduction of computer technology but limitations still remain. For example taking a picture on a snow scene or trying to photograph a black locomotive without overriding the camera calculated metering is very difficult. The most important aspect of the exposure duration is to guarantee that the acquired image falls in a good region of the sensors sensitivity range. In many devices, the selected exposure value is the main processing step for adjusting the overall image intensity that the user will see. Many of the first digital cameras used a separate metering system to set exposure duration, rather than using data acquired from the sensor chip. Integrating exposure-metering function into the main sensor (through-the-lens, or TTL, metering) may reduce system cost. The imaging community uses a measure called *Exposure Value* (EV) to specify the relationship between the f-number¹, F , and exposure duration, T :

$$EV = \log_2\left(\frac{F^2}{T}\right) = 2\log_2(F) - \log_2(T) \quad (3.1)$$

The exposure value (3.1) becomes smaller as the exposure duration increases, and it becomes larger as the f-number grows. Most auto-exposure algorithms work in this way:

1. Take a picture with a pre-determined exposure value (EV_{pre});
2. Convert the RGB values to luminance, L ;
3. Derive a single value L_{pre} (like center-weighted mean, median, or more complicated weighted method as in matrix-metering) from the luminance picture;

¹f-numbers, or aperture values, are measurement of the size of the hole that the light passes through the rear of the lens, relative to the focal length. The smaller the f-number, the more light gets through the lens.

EV	f-number												
	1.0	1.4	2.0	2.8	4.0	5.6	8.0	11	16	22	32	45	64
-6	60	2 m	4 m	8 m	16 m	32 m	64 m	128 m	256 m	512 m	1024 m	2048 m	4096 m
-5	30	60	2 m	4 m	8 m	16 m	32 m	64 m	128 m	256 m	512 m	1024 m	2048 m
-4	15	30	60	2 m	4 m	8 m	16 m	32 m	64 m	128 m	256 m	512 m	1024 m
-3	8	15	30	60	2 m	4 m	8 m	16 m	32 m	64 m	128 m	256 m	512 m
-2	4	8	15	30	60	2 m	4 m	8 m	16 m	32 m	64 m	128 m	256 m
-1	2	4	8	15	30	60	2 m	4 m	8 m	16 m	32 m	64 m	128 m
0	1	2	4	8	15	30	60	2 m	4 m	8 m	16 m	32 m	64 m
1	1/2	1	2	4	8	15	30	60	2 m	4 m	8 m	16 m	32 m
2	1/4	1/2	1	2	4	8	15	30	60	2 m	4 m	8 m	16 m
3	1/8	1/4	1/2	1	2	4	8	15	30	60	2 m	4 m	8 m
4	1/15	1/8	1/4	1/2	1	2	4	8	15	30	60	2 m	4 m
5	1/30	1/15	1/8	1/4	1/2	1	2	4	8	15	30	60	2 m
6	1/60	1/30	1/15	1/8	1/4	1/2	1	2	4	8	15	30	60
7	1/125	1/60	1/30	1/15	1/8	1/4	1/2	1	2	4	8	15	30
8	1/250	1/125	1/60	1/30	1/15	1/8	1/4	1/2	1	2	4	8	15
9	1/500	1/250	1/125	1/60	1/30	1/15	1/8	1/4	1/2	1	2	4	8
10	1/1000	1/500	1/250	1/125	1/60	1/30	1/15	1/8	1/4	1/2	1	2	4
11	1/2000	1/1000	1/500	1/250	1/125	1/60	1/30	1/15	1/8	1/4	1/2	1	2
12	1/4000	1/2000	1/1000	1/500	1/250	1/125	1/60	1/30	1/15	1/8	1/4	1/2	1
13	1/8000	1/4000	1/2000	1/1000	1/500	1/250	1/125	1/60	1/30	1/15	1/8	1/4	1/2
14		1/8000	1/4000	1/2000	1/1000	1/500	1/250	1/125	1/60	1/30	1/15	1/8	1/4
15			1/8000	1/4000	1/2000	1/1000	1/500	1/250	1/125	1/60	1/30	1/15	1/8
16				1/8000	1/4000	1/2000	1/1000	1/500	1/250	1/125	1/60	1/30	1/15

Figure 3.1 : Example of fixed exposure times.

- Based on linearity assumption and equation (3.1), the optimum exposure value EV_{opt} should be the one that permits a correct exposure. The picture taken at this EV_{opt} should give a number close to a pre-defined ideal value L_{opt} , thus:

$$EV_{opt} = EV_{pre} + \log_2(L_{pre}) - \log_2(L_{opt}) \quad (3.2)$$

The ideal value L_{opt} for each algorithm is typically selected empirically. Different algorithms mainly differ in how the single number L_{pre} is derived from the picture. In Fig.(3.1) an example Table of EVs, which take into consideration different exposure times and f-numbers, is reported.

3.2.1 Classical Approaches

The metering system in typical imaging device measures the amount of light in the scene and calculates the best-fit exposure value based on the metering mode explained below. Automatic exposure is a standard feature in all digital cameras. After having selected the metering mode, it is requested just pointing the camera and pressing the shutter release. The metering method defines which information of the scene is used to calculate the exposure value and how it is determined. Cameras generally allow the user to select between spot, center-weighted average, or multi-zone metering modes.

Spot Metering

Spot metering allows user to meter the subject in the center of the frame (or on some cameras at the selected AutoFocus (AF) point). Only a small area of the whole frame (between 1-5% of the viewfinder area) is metered while the rest of the frame is ignored. In this case L_{pre} (3.2) is the mean of the center area (see Fig.(3.2(a))). This will typically be the effective center of the scene, but some cameras allow the user to select a different off-center spot, or to recompose by moving the camera after metering. A few models support a Multi-Spot mode which allows multiple spot meter readings to be taken of a scene that are averaged. Both of those cameras and others also support metering of highlight and shadow areas. Spot metering is very accurate and is not influenced by other areas in the frame. It is commonly used to shoot very high contrast scenes. For example (see Fig.(3.2(d))), if the subject's back is being hit by the rising sun and the face is a lot darker than the bright halo around the subject's back and hairline (the subject is "backlit"), spot metering allows the photographer to measure the light bouncing off the subject's face and expose properly for that, instead of the much brighter light around the hairline. The area around the back and hairline will then become over-exposed. Spot metering is a method upon which the Zone System depends².

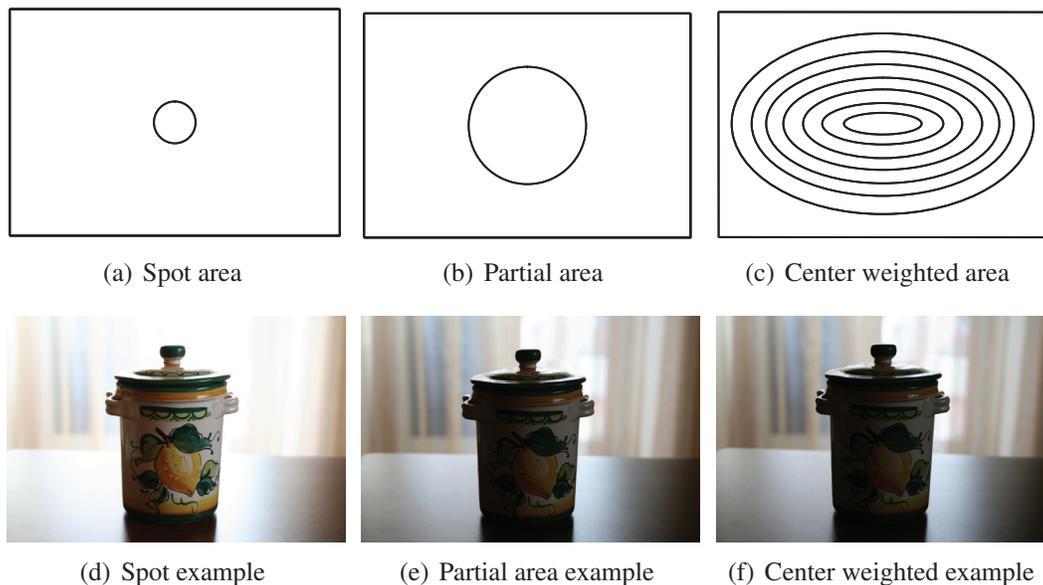


Figure 3.2 : Metering examples.

²The Zone System is a photographic technique for determining optimal film exposure and development, formulated by Ansel Adams and Fred Archer in 1941. The Zone System provides photographers with a systematic method of precisely defining the relationship between the way they visualize the photographic subject and the final results. Although it originated with black and white sheet film, the Zone System is also applicable to roll film, both black and white and color, negative and reversal, and to digital photography.

Partial Area Metering

This mode meters a larger area than spot metering (around 10-15% of the entire frame), and is generally used when very bright or very dark areas on the edges of the frame would otherwise influence the metering unduly. Like spot metering, some cameras can use variable points to take readings from (in general autofocus points), or have a fixed point in the center of the viewfinder. In Fig.(3.2(e)) an example of partial metering on a backlight scene is shown; this method permits to equalize much more the global exposure.

Center-weighted Average Metering

This method is probably the most common metering method implemented in nearly every digital camera: it is also the default for those digital cameras which don't offer metering mode selection. In this system, as described in Fig.(3.2(c)), the meter concentrates between 60 to 80 percent of the sensitivity towards the central part of the viewfinder. The balance is then "feathered" out towards the edges. Some cameras allow the user to adjust the weight/balance of the central portion to the peripheral one. One advantage of this method is that it is less influenced by small areas that vary greatly in brightness at the edges of the viewfinder; as many subjects are in the central part of the frame, consistent results can be obtained. Unfortunately, if a backlight is present into the scene the central part results darker than the rest of the scene (Fig.(3.2(f))), and unpleasant underexposed foreground is produced.

Average Metering

In this mode the camera will use the light information coming from the entire scene and averages for the final exposure setting, giving no weighting to any particular portion of the metered area. This metering technique has been replaced by Center-Weighted metering, thus is only obsolete and present in older cameras only.

3.2.2 Advanced Approaches

Matrix or Multi-zone Metering

This mode is also called matrix, evaluative, honeycomb, segment metering, or esp (electro selective pattern) metering on some cameras. It was first introduced by the Nikon FA, where it was called Automatic Multi-Pattern metering. On a number of cameras, this is the default/standard metering setting. The camera measures the light intensity in several points of the scene, and then combines the results to find the settings for the best exposure. How they are combined/calculated deviates from device to device. The actual number of zones used varies wildly, from several to over a thousand. However performance should not be concluded on the number of zones alone, or the layout. As shown in Fig.(3.3) the layout can change drastically from a manufacturer to another, also within the same company the use of different multi-zone metering can change due to several reasons (e.g., the dimension of the final pixel matrix).

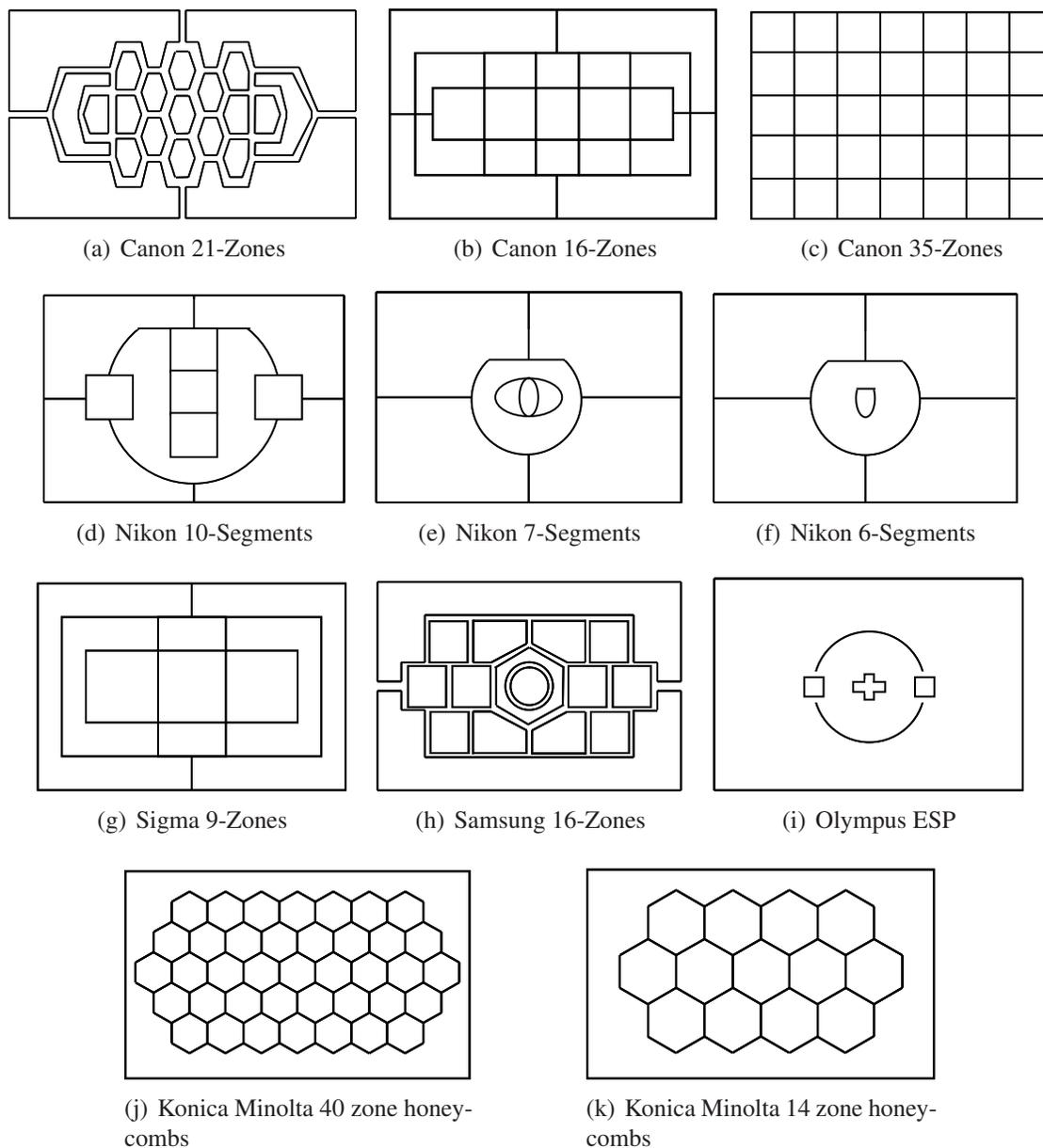


Figure 3.3 : Examples of different kind of Multi-zone Metering mode used by several cameras manufacturers.

Many manufacturers are less than open about the exact calculations used to determine the exposure. A number of factors are taken into consideration, these include: AF point, distance to subject, areas in-focus or out-of-focus, colors/hues of the scene, and back-lighting. Multi-zone tends to bias its exposure towards the autofocus point being used (while taking into account other areas of the frame too), thus ensuring that the point of interest has been properly exposed (it is also designed to avoid the need to use exposure compensation in most situations). A database of many thousands of exposures is pre-stored in the camera, and the processor can use a selective pattern to determine what is being photographed. Some cameras allow the user to link (or unlink) the autofocus and metering, giving the possibility to lock exposure once AF confirmation is achieved, AEL (Auto Exposure Lock). Using manual focus, and on many compact cameras, the AF point is not used as part of the exposure calculation, in such instances it is common for the metering to default to a central point in the viewfinder, using a pattern based off of that area. Some users have problems with wide angle shots in high contrast, due to the large area which can vary greatly in brightness, it is important to understand that even in this situation, the focus point can be critical to the overall exposure.

3.2.3 Exposure Control-System

In many conventional digital cameras, such as Digital Single Lens Reflex (DSLR), the exposure control systems are implemented using mechanical devices. Such mechanical devices include a mechanical iris and/or a mechanical shutter wheel. The most common implementation, the mechanical iris, varies the rate at which the sensor receives photons. The mechanical shutter varies the amount of time during which the sensor receives photons.

Since mechanical devices have a relatively low reliability, slow response time, and increase the size and the cost of lenses, mobile devices have been fitted up with exposure control systems which take into account integration time and multiplication gain factors (see Fig.(3.4)).

The exposure control system performs a first lecture of pixels values, directly on Bayer data coming from the sensor and analyzes the pixel values through a statistic processing block. The algorithm involved into this statistical analysis can be a simple statistical (weighted) mean brightness (see Section 3.2.1) or a more sophisticated metering (see Section 3.2.2).

The viewfinder pipeline is used to estimate the correct exposure. This pipeline is a simplified version of the image generation pipeline, and is used to show a preview of captured image on the embedded display. The exposure control system, described in Fig.(3.4), performs the following steps:

1. Initial integration time consideration: a first image is captured and is given to the system, which perform the statistical analysis.
2. Calculate ideal gain: to achieve a first correction an ideal multiplication gain is is used to accordingly set both analog and digital gains.

3. Determine analog gain: the analog gain is estimated through statistical measures.
4. Finalize integration time: the integration time is then fixed.
5. Determine digital gain: the digital gain is estimated through statistical measures.
6. Determine final exposure: the final exposure is thus estimated and the image capture is achieved.

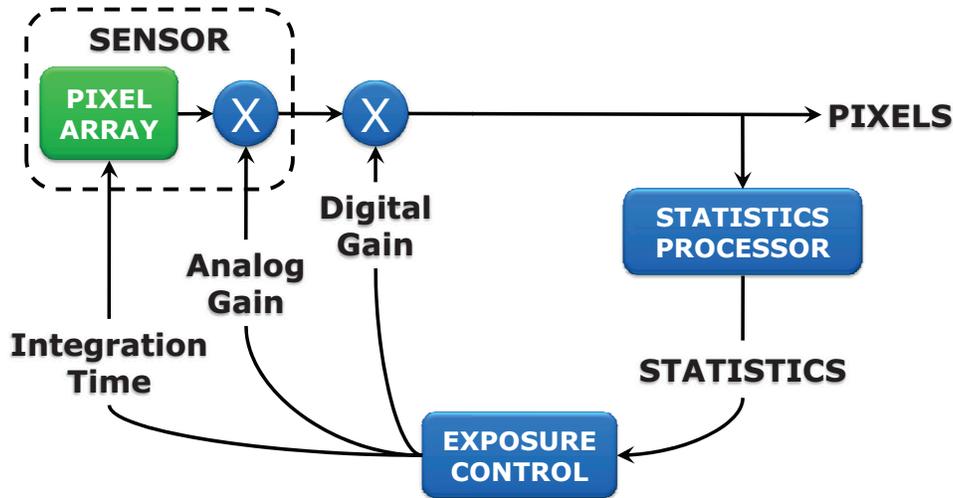


Figure 3.4 : Exposure Control-System pipeline.

3.3 Exposure Correction

As explained in Section 3.2, a good exposure should be able to reproduce the most important regions (according to contextual or perceptive criteria) with a level of gray or brightness, more or less in the middle of the possible range. After acquisition phase typical post-processing techniques try to realize an effective enhancement by using global approaches: histogram specification, histogram equalization and gamma correction to improve global contrast appearance [5, 6] only by stretching the global distribution of the intensity. More adaptive criterions are needed to overcome such drawback. The exposure correction technique [2] described in this Section is designed essentially for mobile sensors applications. This new element, present in newest mobile devices, is particularly harmed by “backlight” when the user utilizes a mobile device for video phoning. The detection of skin characteristics in captured images allows selection and properly enhancement and/or tracking of regions of interest (e.g., faces). If no skin is present in the scene the algorithm switches automatically to other features (such as contrast and focus or indoor/outdoor) tracking for visually relevant regions. This implementation differs from

the algorithm described in [7] because the whole processing can also be performed directly on Bayer pattern images [8], and simpler statistical measures were used to identify *information carrying* regions. The algorithm is defined as follows:

1. Luminance extraction. If the algorithm is applied on Bayer data, in place of the three full color planes, a sub-sampled (quarter size) approximated input data (Fig.(3.8)) is used.
2. Using a suitable features extraction technique the algorithm fixes a value to each region. This operation permits to seek visually relevant regions (for contrast and focus, or indoor/outdoor, the regions are block based, for skin recognition the regions are associated to each pixel).
3. Once the ‘visually important’ pixels are identified (e.g., the pixels belonging to skin features) a global tone correction technique is applied using as main parameter the mean gray level of the relevant regions.

3.3.1 Feature Extraction

As aforementioned, in the following we will briefly describe three techniques able to extract useful information to correct the image exposure.

Focus and Contrast

To be able to identify regions of the image that contain more information, the luminance plane is subdivided in N blocks of equal dimensions. For each block, statistical measures of “contrast” and “focus” are computed. Contrast refers to the range of tones present in the image. A high contrast leads to a higher number of perceptual significance regions inside a block. Focus characterizes the sharpness or edgeness of the block and is useful in identifying regions where high frequency components (i.e., details) are present.

The contrast measure is computed by simply building a histogram H_x for each block x of the N blocks and then calculating its deviation (3.3) from the mean value (3.4):

$$C_x = \frac{\sum_{i=0}^{255} |i - M_x| \cdot H_x[i]}{\sum_{i=0}^{255} H_x[i]} \quad (3.3)$$

where M is the mean value:

$$M_x = \frac{\sum_{i=0}^{255} i \cdot H_x[i]}{\sum_{i=0}^{255} H_x[i]} \quad (3.4)$$

A high deviation value denotes good contrast and vice versa.

The focus measure is computed by convolving each block with a simple 3x3 Laplacian filter. In order to discard irrelevant high frequency pixels (mostly noise), the outputs of

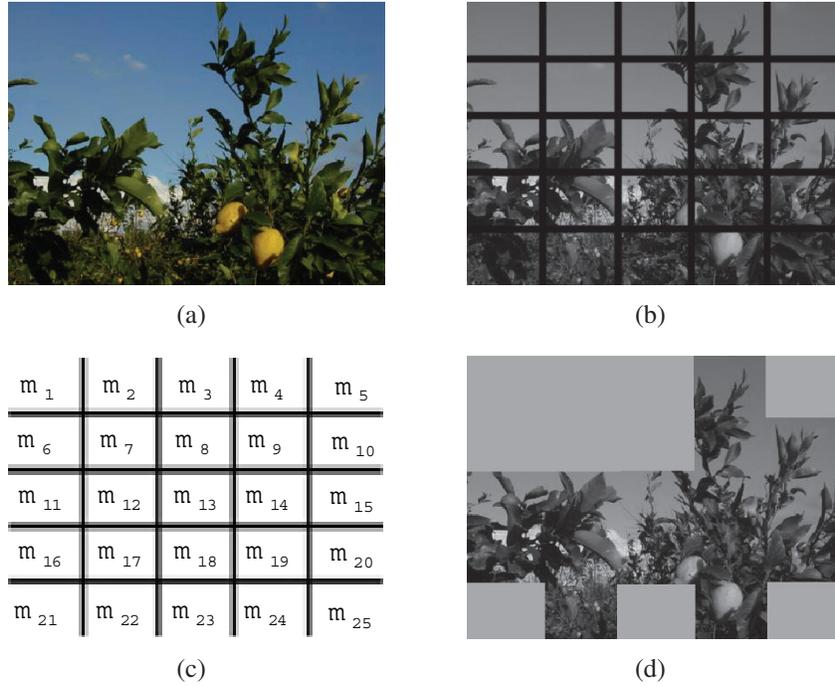


Figure 3.5 : Features extraction pipeline (for focus and contrast with $N = 25$). Visual relevance of each luminance block **(b)** of the input image **(a)** is based on relevance measures **(c)** able to obtain a list of relevant blocks **(d)**.

the convolution at each pixel are thresholded. The mean focus value of each block is computed as:

$$F_x = \frac{\sum_{i=1}^M \text{thresh}[\text{lapl}(i), \text{Noise}]}{M} \quad (3.5)$$

where M is the number of pixels and the $\text{thresh}()$ operator discards values lower than a fixed threshold Noise . Once the values F_x and C_x are computed for all blocks, relevant regions will be classified using a linear combination of both values. Features extraction pipeline is illustrated in Fig.(3.5).

Therefore it is assumed that well focused or high-contrast blocks are more relevant compared to the others.

- Contrast refers to the range of tones present in the image. A high contrast leads to a higher number of perceptual significance regions inside a block.
- Focus characterizes the sharpness or edginess of the block and is useful in identifying regions where high frequency components (i.e., details) are present.

If the aforementioned measures were simply computed on highly underexposed images, then regions having better exposure would always have higher contrast and edginess compared to those that are obscured.

In order to perform a visual analysis revealing the most important features regardless to lighting conditions, a new “visibility image” is constructed by pushing the mean gray level of the input green Bayer pattern plane (or the luminance channel for color images) to 128. The push operation is performed using the same function that is used to adjust the exposure level and it will be described later. Furthermore to remove irrelevant peaks, the histogram is slightly smoothed by replacing each entry with its mean in a ray 2 neighborhood. Thus, the original histogram entry is replaced with the gray-level $\tilde{H}_x[i]$:

$$\tilde{H}_x[i] = \frac{(H_x[i-2] + H_x[i-1] + H_x[i] + H_x[i+1] + H_x[i+2])}{5} \quad (3.6)$$

Once the values F_x and C_x , for each block x of the N blocks, are computed, relevant regions will be classified using a linear combination of both values.

Skin Recognition

As before a “visibility image” obtained forcing the mean gray level of the luminance channel to be about 128 is built. Most existing methods for skin color detection usually threshold some sort of measure of the likelihood of skin colors for each pixel and treat them independently. Human skin colors form a special category of colors, distinctive from the colors of the most other natural objects. It has been found that human skin colors are clustered in various color spaces ([9], [10]). The skin color variations between people are mostly due to intensity differences. These variations can therefore be reduced by using chrominance components only. Yang *et al.* [11] have demonstrated that the distribution of human skin colors can be represented by a two-dimensional Gaussian function on the chrominance plane. The center of this distribution is determined by the mean vector $\vec{\mu}$ and its shape is determined by the covariance matrix Σ ; both values can be estimated from an appropriate training data set. The conditional probability $p(\vec{x}|s)$ of a block belonging to the skin color class, given its chrominance vector \vec{x} is then represented by:

$$p(\vec{x}|s) = \frac{1}{2\pi} |\Sigma|^{-\frac{1}{2}} \exp \left\{ \frac{-[d(\vec{x})]^2}{2} \right\} \quad (3.7)$$

where $d(\vec{x})$ is the so-called Mahalanobis distance from the vector \vec{x} to the mean vector $\vec{\mu}$ and defined as:

$$[d(\vec{x})]^2 = (\vec{x} - \vec{\mu})' \Sigma^{-1} (\vec{x} - \vec{\mu}) \quad (3.8)$$

The value $d(\vec{x})$ determines the probability that a given block belongs to the skin color class. The larger the distance $d(\vec{x})$, the lower the probability that the block belongs to the skin color class s .

Due to the large quantity of color spaces, distance measures, and two-dimensional distributions, many skin recognition algorithms can be used. The skin color algorithm can be applied in different ways (as shown in Fig.(3.6)):

1. By using the input YCbCr image and the conditional probability (3.7), each pixel is classified as belonging to a skin region or not. Then a new image with normalized

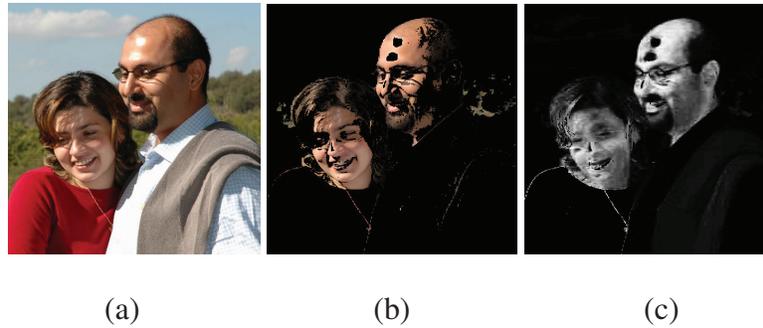


Figure 3.6 : Skin recognition examples on RGB images: **(a)** original images compressed in JPEG format; **(b)** simplest threshold method output; and **(c)** probabilistic threshold output.

grayscale values is derived, where skin areas are properly highlighted (Fig.(3.6(c))). The higher the gray value the bigger the probability to compute a reliable identification.

2. By processing an input RGB image, a 2D chrominance distribution histogram (r, g) is computed, where $r=R/(R+G+B)$ and $g=G/(R+G+B)$. Chrominance values representing skin are clustered in a specific area of the (r, g) plane, called “*skin locus*” (Fig.(3.7(c))), as defined in [12]. Pixels having a chrominance value belonging to the skin locus will be selected to correct exposure.
3. For Bayer data, the skin recognition algorithm works on the RGB image created by sub-sampling the original picture, as described in Fig.(3.8).

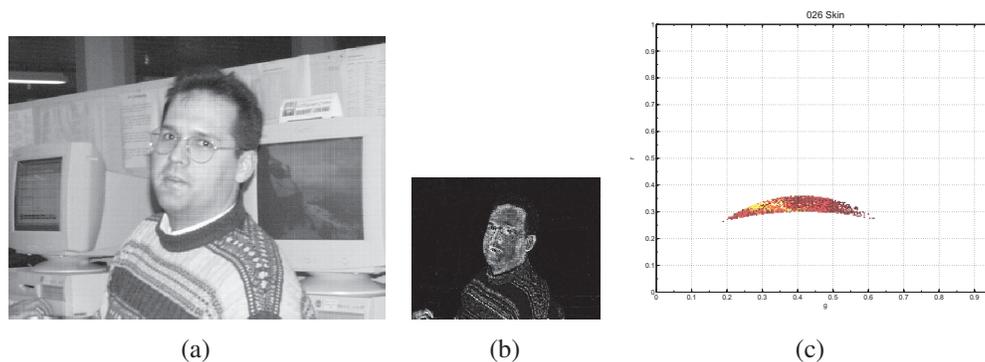


Figure 3.7 : Skin recognition examples on Bayer pattern image: **(a)** original image in Bayer data; **(b)** recognized skin with probabilistic approach; and **(c)** threshold skin values on $r - g$ bidirectional histogram (*skinlocus*).

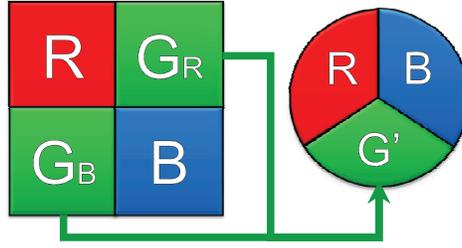


Figure 3.8 : Bayer data sub-sampling generation, where $G' = \frac{G_R+G_B}{2}$.

Indoor/Outdoor

Another technique reported in [13] uses a two-stage classifier exploiting two features: colors and textures. Then through a Support Vector Machine (SVM) classifier [14], the algorithm independently classifies image blocks according to features using wavelets coefficients. The classified blocks are then used by the second stage SVM classifier to determine a final indoor/outdoor decision.

To be able to extract color information the image is by first converted into the LST color space. This color space is based on Otha Color Space [15] except for the normalization factors:

$$\begin{aligned}
 L &= \frac{k}{\sqrt{3}}(R + G + B) \\
 S &= \frac{k}{\sqrt{2}}(R - B) \\
 T &= \frac{k}{\sqrt{6}}(R - 2G + B)
 \end{aligned}
 \tag{3.9}$$

where $k = 225/\max(R, G, B)$. Thus L represents the luminance channel, S and T the chrominances one. This color space is used to de-correlate color channels in the original RGB image. Using a 16 bin histogram for each channel a 48 dimensions feature vector is computed for each block [13] and classified independently.

The texture features are extracted, from the L channel, through a two level wavelets decomposition, using Daubechies' 4-taps filters [16].

Let $c_2, c_3, c_4, c_5, c_6, c_7$ and c_8 represents the subband coefficients of the two level wavelets decomposition, as described in Fig.(3.9). The texture features are obtained by first filtering the low-frequency coefficients c_5 using the Laplacian filter and then by tak-

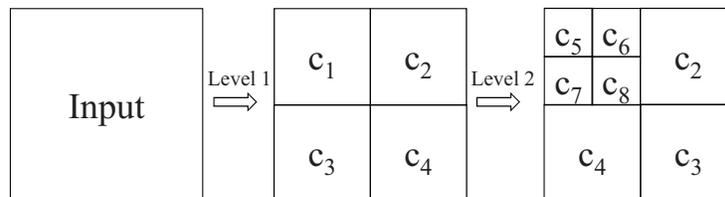


Figure 3.9 : Coefficients of the two-level wavelets decomposition.

ing into consideration the subband energy:

$$e_x = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |c_x(i, j)|^2, \quad x = 2, 3, \dots, 8 \quad (3.10)$$

where M and N are the image dimensions.

For classifying each block, using calculated features, the SVM is applied with a radial basis kernel function, as described by Efimov *et al.* in [17].

By taking into account the information, collected on block basis, a similar approach as described in 3.3.1 can be achieved. In this case the correction is adaptively performed considering the number of indoor versus the number of outdoor blocks. In particular:

1. if the number of indoor blocks is larger than the number of outdoor blocks then the scene has probably been taken into a room and the exposure must be correct in function of such blocks.
2. if the number of outdoor blocks is larger than the number of indoor blocks then the scene has probably been taken outside the exposure must be correct in function of such blocks.
3. finally if the number of blocks of each class is equal or more or less equal then the exposure must be corrected taking into consideration the global statistics.

3.3.2 Exposure Correction

Once the visually relevant regions are identified, the exposure correction is carried out using the mean gray value of those regions as reference point. A simulated camera response curve is used for this purpose. This function can be expressed by using a simple parametric closed form representation:

$$f(q) = \frac{255}{(1 + e^{-(Aq)})^C} \quad (3.11)$$

where q represents the ‘light’ quantity and the final pixel value is obtained by means of the parameters A , and C used to control the shape of the curve. q is supposed to be expressed in 2-based logarithmic unit (usually referred as “stops”). These parameters could be estimated, depending on the specific image acquisition device or chosen experimentally, as better specified below (see Chapter 13). The offset from the ideal exposure is computed using the f curve and the average gray level of visually relevant regions avg , as:

$$\Delta = f^{-1}(Trg) - f^{-1}(avg) \quad (3.12)$$

where Trg is the desired target gray level. Trg should be around 128 but its value could be slightly changed especially when dealing with Bayer Pattern data where some post processing is often applied.



Figure 3.10 : Exposure Correction results by real-time and post processing: **(a)** original Bayer input image; **(b)** Bayer skin detected in real-time; **(c)** color interpolated image from Bayer input; **(d)** RGB skin detected in post processing; **(e)** exposure corrected image obtained from RGB image.

The luminance value $Y(x,y)$ of a pixel (x,y) is modified as follows:

$$Y'(x,y) = f(f^{-1}(Y(x,y)) + \Delta) \quad (3.13)$$

Note that all pixels are corrected. Basically all processing steps could be implemented through a LUT (Lookup Table) transform.

3.3.3 Results

The described technique has been tested using a large database of images acquired at different resolutions, with different acquisition devices, both in Bayer and RGB format. In the Bayer case the algorithm was inserted in a real-time framework, using a CMOS-VGA sensor on “STV6500 - EOI” Evaluation Kit equipped with “502 VGA sensor” [18]. Examples of skin detection by using real time processing are reported in Fig.(3.10). In the RGB case the algorithm could be implemented as post-processing step. Examples of contrast/focus, indoor/outdoor and skin exposure correction are respectively shown in Fig.(3.11), Fig.(3.12) and Fig.(3.13). Results show how the features analysis capability

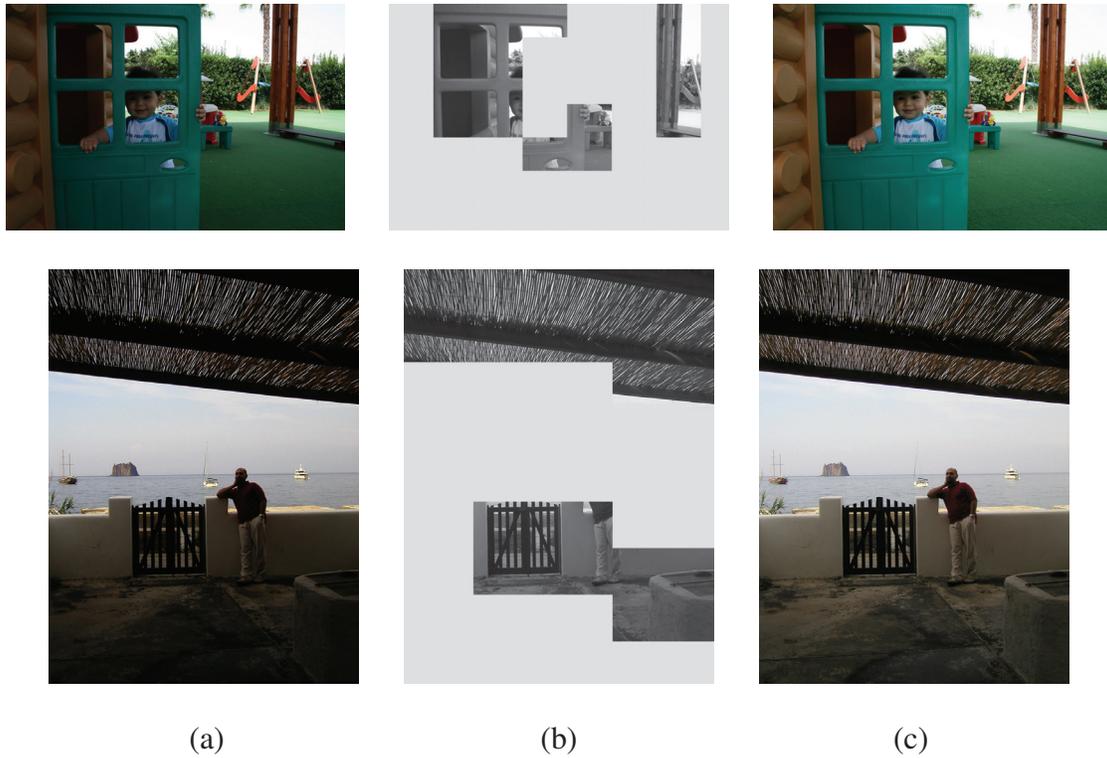


Figure 3.12 : Exposure correction results by post processing: (a) original color input image; (b) indoor blocks detected; (c) exposure corrected image obtained from RGB image.

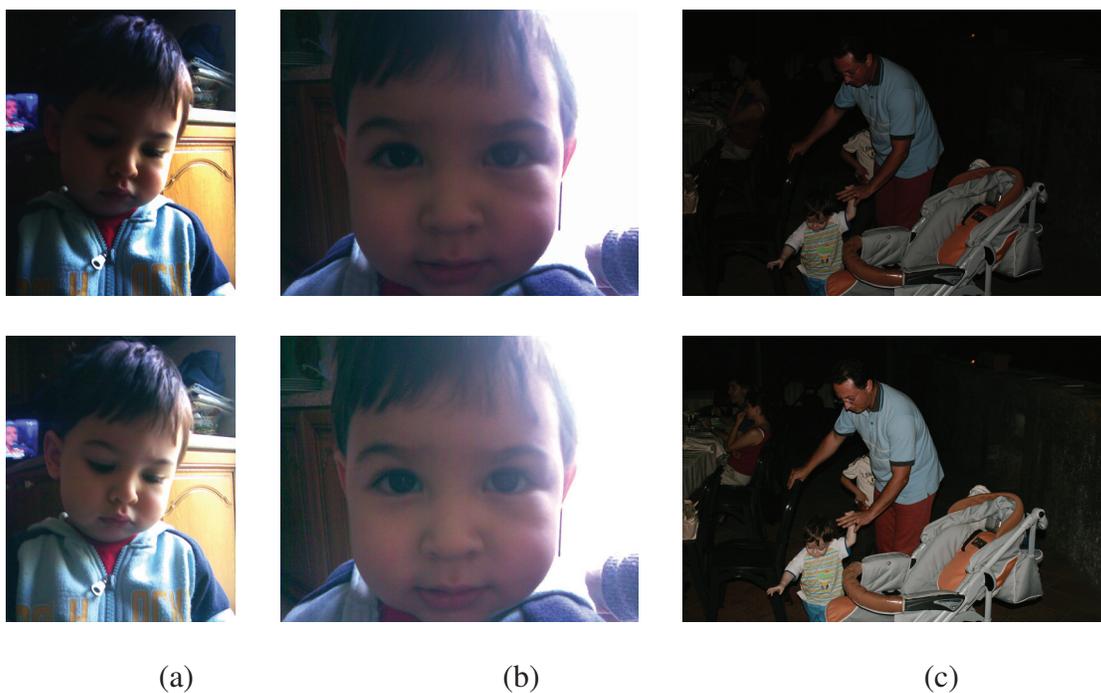


Figure 3.13 : Exposure correction results: in the first row the original images (a) and (b) acquired by Nokia 6125, 1.3Mpixels CMOS sensor; (c) picture acquired with CMOS sensor 10 Mpixels, Canon 400D camera; in the second row the corrected output.

Bibliography

- [1] M. Mancuso and S. Battiato, "An introduction to the digital still camera technology," *ST Journal of System Research*, vol. 2, no. 2, pp. 1–9, 2001.
- [2] S. Battiato, A. Bosco, A. Castorina, and G. Messina, "Automatic image enhancement by content dependent exposure correction," *EURASIP Journal on Applied Signal Processing*, vol. 12, pp. 1849–1860, 2004.
- [3] S. Battiato, A. Castorina, and M. Mancuso, "High dynamic range imaging for digital still camera: an overview," *Journal of Electronic Imaging*, vol. 12, pp. 459–469, July 2003.
- [4] S. Battiato, G. Messina, and A. Castorina, *Single-Sensor Imaging, Methods and Applications for Digital Cameras*, ch. Exposure Correction for Imaging Devices: an Overview. Image Processing Series, R. Lukac, CRC Press, 2008.
- [5] R. Gonzalez and R. Woods, *Digital Image Processing 3rd Edition*. Prentice Hall, 2008.
- [6] T. Arici, S. Dikbas, and Y. Altunbasak, "A histogram modification framework and its application for image contrast enhancement," *IEEE Transaction on Image Processing*, vol. 18, no. 9, pp. 1921–1935, 2009.
- [7] S. Bhukhanwala and T. Ramabadran, "Automated global enhancement of digitized photographs," *IEEE Transactions on Consumer Electronics*, vol. 40, no. 1, 1994.
- [8] B. Bayer, "Color imaging array," *U.S. Patent 3,971,065*, 1976.
- [9] S. Phung, A. Bouzerdoun, and D. Chai, "A novel skin color model in YCbCr color space and its application to human face detection," in *International Conference on Image Processing*, vol. 1, pp. 289–292, 2002.
- [10] B. Zarit, B. Super, and F. Quek, "Comparison of five colour models in skin pixel classification," in *International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, IEEE Computer Society*, pp. 58–63, 1999.
- [11] J. Yang, W. Lu, and A. Waibel, "Skin-colour modeling and adaptation," *Technical Report CMU-CS-97-146 School of Computer Science, Carnegie Mellon University*, 1997.
- [12] M. Soriano, B. Martinkauppi, and M. L. S. Huovinen, "Skin color modeling under varying illumination conditions using the skin locus for selecting training pixels," *Real-time Image Sequence Analysis (RISA 2000)*, 2000.

- [13] N. Serrano, A. Savakis, and J. Luo, "A computationally efficient approach to indoor/outdoor scene classification," in *International Conference on Pattern Recognition (ICPR 2002)*, pp. 146–149, 2002.
- [14] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.
- [15] Y. Otha, T. Kanade, and T. Sakai, "Color information for region segmentation," *Computer Graphic Image Processing*, vol. 13, pp. 222 – 241, 1980.
- [16] I. Daubechies, *Ten lectures on wavelets*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1992.
- [17] S. Efimov, A. Nefyodov, and M. Rychagov, "Block-based image exposure assessment and indoor/outdoor classification," in *International conference on the Computer Graphics and Vision (GraphiCon'2007)*, (Moscow, Russia), June 2007.
- [18] STMicroelectronics, "Colour sensor evaluation kit vv6501." Edinburgh - www.edb.st.com/products/image/sensors/501/6501evk.htm.
- [19] R. Raskar, A. Agrawal, and J. Tumblin, "Coded exposure photography:motion deblurring using flattered shutter," in *SIGGRAPH - International Conference on Computer Graphics and Interactive Techniques*, (Boston), July-August 2006.

Pre-acquisition: Auto-focus

A. Capra, S. Curti

Advanced System Technology - Catania Lab, STMicroelectronics, Italy.

Abstract: In this Chapter different auto focusing techniques are analyzed. First Auto-focus (AF) techniques were implemented into Single Lens Reflex (SLR) cameras. They use a dedicated system to focus a scene which is independent from the acquisition part mainly based on a phase detection system. Digital still cameras (DSC) instead use the acquisition sensor also to focus the scene. An image processing system based on contrast analysis finds the in-focus position and is employed in DSC due to its compactness and cheapness. Both the SLR and the DSC optics need a moving lens to focus the scene. Very low cost and ultra small cameras, such as those integrated into mid-low cost Personal Digital Assistant (PDA) (i.e., smart-phones), don't have any moving part. In this case to accomplish a further extension of the Depth of Field (DoF) these modules implement a digital auto-focus technique known as Extended Depth of Field (EDoF). Nowadays most of the camera systems implement sophisticated content dependent AF models: they are capable to optimize their behavior for still and video acquisition, to detect and prioritize focus on faces and to predict new lens position when moving object are being focused.

4.1 Principles of Auto-focusing

Before analyzing different methods used to get an in-focus image it is important to give an overview on fundamentals of optics (more details can be found in [1]) and the correspondence between the amount of blur and the lens position. In the following Subsections the main components of a focusing system are introduced as well as their contribute to the final picture. Starting from the optical components a few useful methods to estimate level of focus and a high level description of a control system used to get an in-focus picture are described.

4.1.1 Lens System

Actually, a lens system is compound of many single lenses properly stacked together (for example, low cost mobile cameras use four lenses). Nevertheless in the following it is useful to refer to an acquisition system as compound of very basic and schematic elements: a single tiny lens, an object represented by a single point far away from the lens (the following relationships and equations are not valid for macro photography) and a sensor surface (represented by a segment) where the image from the real world is projected. Fig.(4.1(a)) is a schematic representation of this basic system.

Table 4.1 summarizes all the symbols used in the remaining of this Section.

Fundamentals on Optics Rules

An illuminated object reflects the incident light and radiates in almost all directions. The part of light of interest for the camera is only the one which produces to the sensor an image of the object. If we draw two lines from the object to the outer edges of the lens

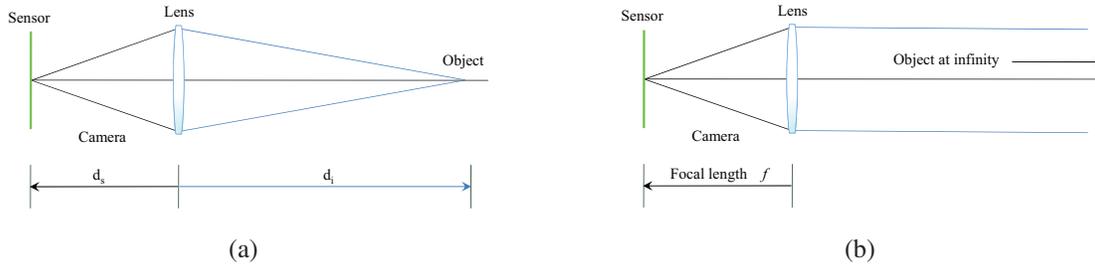


Figure 4.1 : Basic imaging system: **(a)** focusing a point away from the lens; **(b)** focusing a point positioned at infinity distance.

Table 4.1 : Symbols used to represent optical fundamentals hereby reported.

f	= focal length
d_i	= lens-to-object distance
d_s	= lens-to-sensor distance (back focal length)
M	= image magnification (the size of the image expressed as a fraction of the actual size of the real object)
c	= diameter of the circle of confusion
C	= max allowable diameter of the circle of confusion
R	= Radius of the circle of confusion
A_p	= diameter of the lens
e	= focus error distance
N_A	= f -number, diameter of the diaphragm
p	= dimension of the side of a square pixel, or pixel pitch
E	= max allowable focus error distance
D_1	= Inner limit of the Depth of Field
D_2	= Outer limit of the Depth of Field
H	= Hyperfocal Distance

and from there to the sensor, we obtain the basic lines of light (two cones actually) which affect the imaging process. This schema correctly represents a system which projects a point in the real world to a point in the sensor and thus a correctly in-focus point. One cone has its apex at the object and its base on the front of the lens, the other cone has its apex at the sharply focused image on the sensor and its base on the back of the lens.

There is a mathematical relationship between the lens-to-sensor distance and the focal length of the lens when the image is perfectly sharp, which can be obtained applying some geometric rules:

$$\frac{1}{f} = \frac{1}{d_s} + \frac{1}{d_i} \quad (4.1)$$

The focal length f of a lens is simply defined as the lens-to-sensor distance which

gives a perfect in-focus image when the object is at infinity, Fig.(4.1(b)). From (4.1), with some derivation, the following equations can be obtained:

$$d_i = \frac{d_s f}{d_s - f} \quad (4.2)$$

$$d_s = \frac{d_i f}{d_i - f} \quad (4.3)$$

$$f = \frac{d_s d_i}{d_i + d_s} \quad (4.4)$$

An object projected on the sensor is represented by a figure with a size which is a fraction of its real size. It is known as image magnification and its equation is:

$$M = \frac{d_s}{d_i} \quad (4.5)$$

Fundamentals on Circle of Confusion

Until now it has been supposed that a point in the real world is correctly projected on the sensor and thus the image is in-focus. As Fig.(4.2) shows, if the object is a single point source of light and the sensor is away from where this point is normally projected (at a distance e from the correct d_s), the image of that point on the sensor becomes a small disk of light, not a single point. In this case the image of the point is blurred and the system is out-of-focus. This small disk of light is called circle-of-confusion of diameter c . The greater is c the more the system is out of focus. The distance e is known as focus error and measures the distance of the sensor from the position where the image of the object is actually projected and thus the actual in-focus position. In an ideal system when $e = 0$ the system is in-focus and $c = 0$.

Applying geometric equations about similar triangles, the diameter of circle of confusion c is proportional to the diameter of the lens A_p and to the focus error e :

$$c = \frac{e}{d_s} A_p \quad (4.6)$$

In literature there are other relationships to estimate the circle of confusion. In [2] being R the radius of the circle of confusion, a scaling factor q is defined as:

$$q = \frac{c}{A_p} = 2 \frac{R}{A_p} \quad \text{and} \quad R = \frac{c}{2} \quad (4.7)$$

From similar triangles equations it is possible to obtain:

$$q = \frac{d_s - d'_s}{d_s} = d'_s \left(\frac{1}{d'_s} - \frac{1}{d_s} \right) \quad (4.8)$$

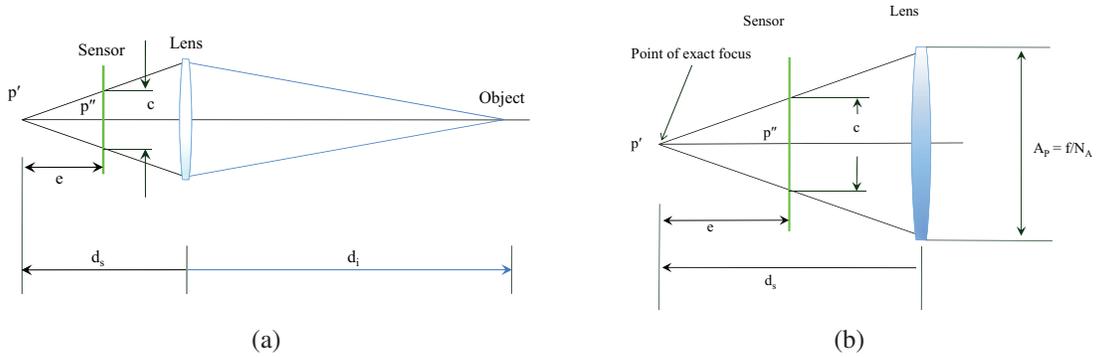


Figure 4.2 : Out-of-Focus projection of a point on the sensor surface: **(a)** global view of the system; **(b)** detail between the lens and the sensor of an Out-of-Focus.

where:

$$d'_s = d_s - e \quad (4.9)$$

Combining (4.1) into (4.8):

$$q = d'_s \left(\frac{1}{d'_s} - \frac{1}{f} + \frac{1}{d_i} \right) \quad (4.10)$$

Therefore, using (4.7) and (4.10) R can be expressed as follows:

$$R = \frac{A_p}{2} d'_s \left(\frac{1}{d'_s} - \frac{1}{f} + \frac{1}{d_i} \right) \quad (4.11)$$

Note that q and R can be either positive or negative, depending on whether $d_s > d'_s$ or $d_s < d'_s$ and represent the degree of de-focus of a lens system.

The circle of confusion is important to understand if, for a given scene and lens position, the image is in-focus or not. Fundamentally in digital photography a point source is in-focus if its projection on the sensor is still a point and thus if it falls inside a pixel. This means that until the circle of confusion diameter is included inside a pixel the image is sharp. Supposing to have a square pixel of side p it is possible to express the maximum allowed circle of confusion for a sharp image just considering the following inequality:

$$c \leq p\sqrt{2} \quad (4.12)$$

where $p\sqrt{2}$ represents the diagonal of the pixel.

As described in Section 2.2 not all the surface of a pixel is dedicated to the photodetector but only a part. In this case it is better to refer to the distance between the centre of two pixels instead of the side of a single pixel. For this reason p may also represent the pixel pitch.

From the (4.12) it is clear the smaller is the pixel size the more accurate (and with less tolerance) has to be a lens system to keep in focus an image. For example for a sharp

image a sensor with a pixel size of 1.8μ requires a circle of confusion smaller than 2.55μ . The same system with a pixel size of 2.2μ has the same sharpness level when a circle of confusion is 3.11μ , which is 22% greater.

Fundamentals on Diaphragm

A diaphragm is a device with a circular opening in the center put on top of the lens, thus limiting the entering light passing through the lens itself. Usually, the diameter of this opening is variable and its effect is equal to change the lens diameter from its max to a lower value. The diameter of this circular opening is usually known as *f-number* and represents the effective diameter of the lens (i.e., the useful diameter of the lens). It is represented by numbers (like 1.4, 2, 2.8, ..., 22) which represent the diameter as a fraction of the lens focal length. For example a *f-number* = 8 means the diameter of the circular hole is 1/8 of the focal length of the lens. There are different notations in literature, for example in [1] a *f-number* = 8 is denoted as $f/8$.

Diaphragm is mainly used to reduce the amount of light entering through the lens thus controlling the brightness of the image but it has effect on image sharpness. Hence by using the diaphragm the effective diameter of the lens can be expressed as:

$$A_p = \frac{f}{N_A} \quad \text{where } N_A = f\text{-number} \quad (4.13)$$

It is clear that N_A variations have effect on (4.6) and as consequence on the focusing performance of the lens, as described in the next two paragraphs.

Fundamentals on Depth of Field

Assumption summarized by (4.12) states that an object positioned at a distance d_i from the lens is focused in accordance with the (4.1) until the lens to sensor distance d'_s is comprised in the range represented by the following disequality:

$$(d_s + e) \leq d'_s \leq (d_s - e) \quad (4.14)$$

The zone which goes from $(d_s + e)$ to $(d_s - e)$ is known as depth of focus (dof). In addition, there is a range of space in which the objects are in-focus if their distance d'_i from the lens, in accordance with (4.1) has a respective d_s distance with satisfies (4.14). In other words, when the system is perfectly focusing an object at distance d_i , if D_1 represent the lens to object distance when the sensor to lens distance is $(d_s - e)$ and D_2 the respective for $(d_s + e)$ all the objects in the range $[D_1, D_2]$ are in-focus. This range is known as Depth of Field (DoF).

Combining (4.13) into (4.6) under the assumption that d_s is close to f it is obtained:

$$c \approx \frac{e f}{f N_A} = \frac{e}{N_A} \quad (4.15)$$

Denoting with C the maximum allowable diameter of the circle of confusion, the maximum allowable distance from the exact point of focus d_s can be expressed by

$$E = N_A C \quad (4.16)$$

Here again the equation states that the lens aperture can be used to control the maximum allowable focus error and, as consequence, the Depth of Field.

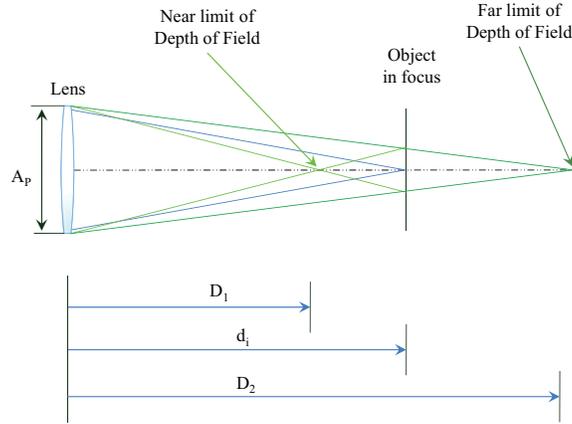


Figure 4.3 : Depth of Field.

By referring to Fig.(4.3) and by using simple geometrical algebra (4.1) can be used to calculate D_1 and D_2 as follows:

$$D_1 = \frac{d_i f^2 + E f d_i - E f^2}{f^2 - E f + E d_i} \quad (4.17)$$

and

$$D_2 = \frac{d_i f^2 - E f d_i + E f^2}{f^2 + E f - E d_i} \quad (4.18)$$

Fundamentals on Hyperfocal Distance

When the system focuses an object at infinity (4.18) becomes:

$$\lim_{d_i \rightarrow \infty} D_1 = f + \frac{f^2}{E} = H \quad (4.19)$$

The Hyperfocal distance represents the inner limit of the Depth of Field when the system focuses an object at infinity. By replacing (4.19) into (4.17) and (4.18):

$$D_1 = \frac{d_i H - f^2}{H + d_i - 2f} \approx \frac{d_i H}{H + d_i} \quad (4.20)$$

and

$$D_2 = \frac{d_i H - 2f d_i + f^2}{H - d_i} \approx \frac{d_i H}{H - d_i} \quad (4.21)$$

These approximations are valid until d_i is many times greater than the focal length and thus are not valid in macro photography.

Special importance has the hyperfocal distance. Equations (4.20) and (4.21) state that if the lens is set to focus an object at distance H then D_2 tends to infinity and ($D_1 = H/2$). In this case all the objects distant from D_1 to infinity are in focus and this is the maximum Depth of Field a lens system can achieve. Equivalently if d_i is at infinity then ($D_1 = H$) and ($D_2 = -H$) which means D_2 is far away beyond the infinity. It is easy to conclude that ($D_2 \leq 0$) every time ($d_i \geq H$).

Fundamentals on Point Spread Function

Once the circle-of-confusion and its relationship with optical model has been defined let put into equations its effect on a sharp image.

Since the lens aperture is circular, the blurred image of a single point is also a circle with uniform brightness inside the circle and zero outside. This is usually called blur circle. If the light energy incident on the lens from a single point during one exposure period of camera is one unit then the blurred image of this single point is the response of the camera to a unit point source or in mathematics the point spread function (PSF) of the camera system. Assuming the system to be a lossless system (i.e., no light energy is absorbed) and denoting the PSF by $h(x, y)$, subsists the following equation:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x, y) dx dy = 1 \quad (4.22)$$

As sees the blur circle of a single point (4.7 and subsequent) has uniform brightness inside the circle of radius R and zero outside. Thus:

$$h(x, y) = \begin{cases} 1/\pi R^2 & \text{if } x^2 + y^2 \leq R^2 \\ 0 & \text{otherwise} \end{cases} \quad (4.23)$$

In a real system [2] diffraction, polychromatic illumination, lens aberrations, etc. make the brightness inside the circle not constant but falling off gradually. For this reason the (4.23) is usually represented (Fig.(4.4(a))) by a two dimensional Gaussian filter:

$$h(x, y) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (4.24)$$

where σ represents the standard deviation of the distribution of a PSF and in practice proportional to R by the relationship:

$$\sigma = k \cdot R \quad (4.25)$$

with k a constant approximately equal to:

$$k = \frac{1}{\sqrt{2}} \quad (4.26)$$

Known the PSF it is possible to compute the blurred image from a sharp image. Being $f(x, y)$ the sharp image, the blur image $g(x, y)$ is obtained by the convolution of the sharp image with the PSF:

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(u, v) h(x - u, y - v) du dv \quad (4.27)$$

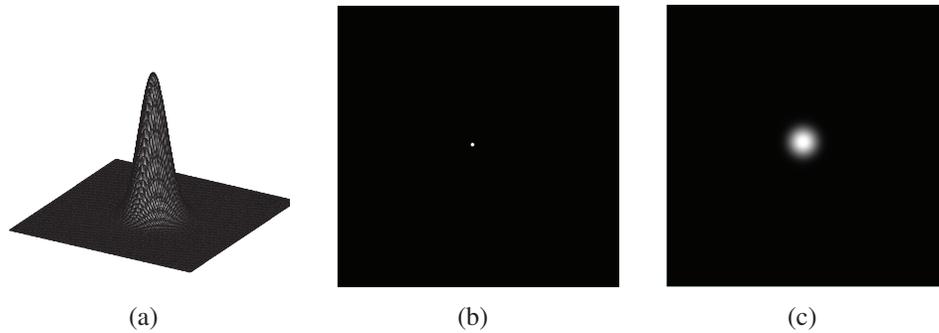


Figure 4.4 : Point Spread Function applied to a point image: (a) two dimensional Gaussian Point Spread Function; (b) single point input image; (c) blurred image after the convolution of the single point with the PSF.

In Fig.(4.4) is represented the convolution result. It is easy to verify that a PSF, as described here, corresponds to a low pass filter. As a conclusion a blur image is the resulting of applying a low pass filter to a sharp image.

4.1.2 Actuator Models

At this point of the discussion it is clear that to get an in-focus image the lens system must satisfy the above mentioned rules. In some cases a system with a small focal length and high f -number (or, equivalently, small lens diameter) has an adequate DoF to focus all the object in the scene without the need to move the lens. This often happens in low cost mobile cameras where the low resolution (up to 1-2M pixels) and the small sensor size support this solution.

When the DoF is not wide enough it is required a moving lens that, in accordance with (4.1), properly focuses the main object in the scene. Of course, other objects may be focused if they fall inside the DoF region, as described by (4.20) and (4.21). To move the lens to the appropriate position a motor is required. In the past years many different types of actuator have been developed (some of which are illustrated in Fig.(4.5)) and a few of them have been implemented in commercial products. In the following Sections some of the most used or promising micro-actuators and solutions for mobile sensors are described. Table 4.2 summarizes the main differences.

Stepper Motor

Stepper motors have been widely used in AF systems for DSC and SLR cameras. There are different implementations which aim to improve speed and reduce noise during the movement.

The advantage of this motor (basic schema illustrated in Fig.(4.6)) is that it is well known and at any time it is possible to exactly know the position of the lens. Also, the

Electromagnetic
Stepper Motors
Simple Solenoids
Voice Coil Solenoids
Piezoelectric
Stacked Piezo Devices
Bimorphs
Disk Translators
Moonie Motors
Helimorphs
Oscillating Bimorphs
Inch Worms
Ultrasonic disk motors
Electrostatic
Electrostrictive
Magnetostrictive
Shape memory alloys
MEMS



Figure 4.5 : List of different types of actuators and graphical representation. Many technologies have been investigated as well as different shapes and dimensions.

Table 4.2 : Comparison between different types of actuators.

Stepper Motor	Voice Coil	Piezo Electric	Liquid Lens
Thin and long, large volume. Many components. Complex to produce	Multiple windings needed around the lens barrel	Wide and flat, compact volume. Few components, only three	Need to avoid water freezes
Low torque at high rotational speeds	Not feasible	High torque at low rotational speeds	Not feasible
Gearing often necessary	No gearing necessary	No gearing necessary	No gearing necessary
Electromagnetic interference	Electromagnetic interference	No magnets. No electromagnetic interference	Temperature may affects performances
Multiple phase current switching necessary to command the motor	Separate driver required	Simplest motor control system by means of two drive signals	Simple motor control system
No static holding power drain	Static power to hold position	No static holding power drain	Static power to hold position but very low current required

new position is accurately and quickly reached by the motor without drift or hysteresis problems.

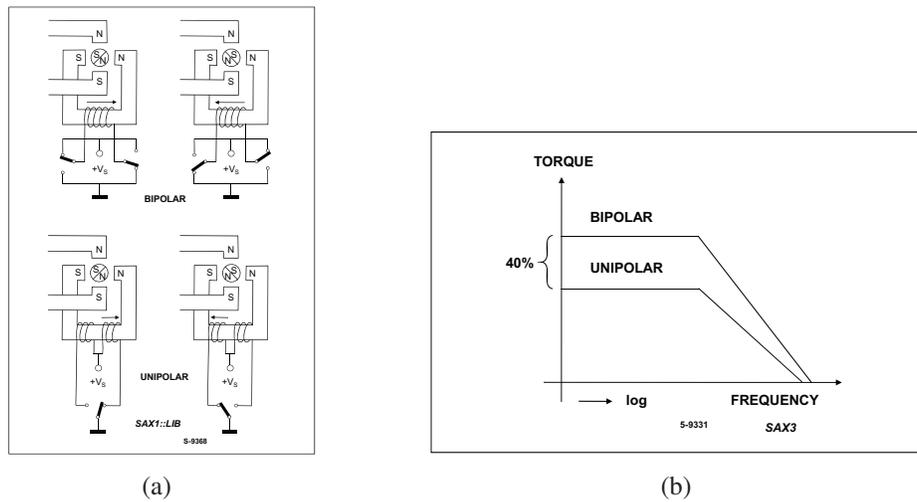


Figure 4.6 : Stepper motor electro-mechanical schematics: (a) Basic schema of a stepper motor. A shaft is driven by a discrete rotation; (b) Torque force of the stepper.

In mobile imaging phones this technology was the first to be implemented but has been replaced by new solutions mainly due to miniaturization problems (thickness is imperative in these devices) and the audible residual noise which is considered not acceptable by the user. Noise generated by an AF lens is usually accepted if generated by devices specifically dedicated to photography (SLR, DSC, camcorders) but not from devices where imaging functionalities are considered extension to a main feature; and mobile phones are good representative of this category.

Voice Coil

Voice coil actuators are the standard actuators used in speakers for Hi-Fi. As illustrated in Fig.(4.7) they are composed of a magnet which is moved inside a sheath by applying a voltage to the coil. By varying polarity of the voltage and current amplitude it is possible to control the position of the cylindrical magnet inside the sheath. Although this kind of actuator is a well know technology and it is easy to control the position, it has the drawback of requiring a current to hold the lens also in static position. This causes a high power consumption and for this reason it is rarely used in battery powered devices as mobile phones. In the last years some manufactures have developed an advanced voice coil actuator which maintains the static position by using a static holding magnet.

Piezo-Electric

This actuator is based on the principle that piezoelectric materials have the ability to generate a mechanical movement (or change in their shape) if subjected to an electrical

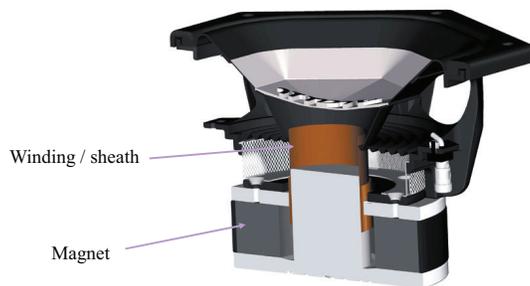


Figure 4.7 : Basic principle of a voice coil system.

field. Recently new devices based on piezoelectric materials in 2 or more layers have been presented as promising for micro-actuators solutions [3, 4]. By putting a voltage on the layers the different expansion of the layers can cause a bend. 1Limited [5] uses a patented shape, the helimorph, to magnify the deflection to make it larger than a simple bender. In Fig.(4.8) are represented different methods to curve this material to generate piezo-based micro-actuators. One of the problems of which may suffer this actuators is the hysteresis. It may happen that the bend obtained when removing an incremental voltage is not the same that the material had before applying this voltage. This can be a problem because if the driver moves the actuator from a starting position to a new position and then goes back it may happen that the final position is not the starting position. To override this problem it is required a position sensor which communicates the correct position to the driver (Fig.(4.9)).

Another problem which affected this technology is the speed. Depending on the selected shape the maximum speed sometimes is not fast enough.

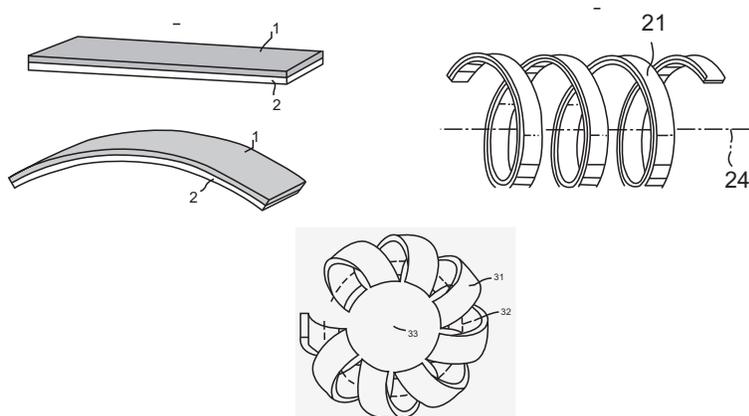


Figure 4.8 : Example of bend of the helimorph piezoelectric materials used to generate micro-actuators.

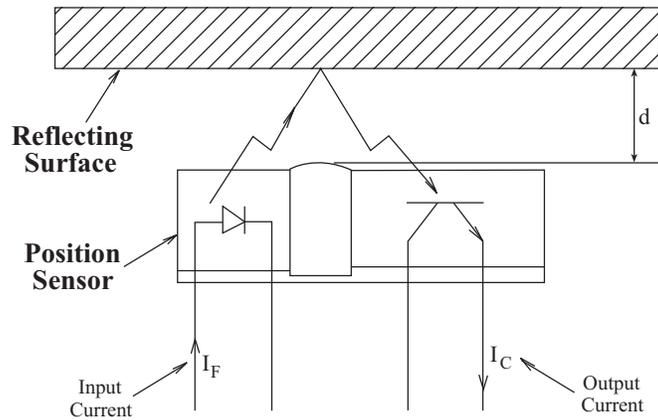


Figure 4.9 : Example of position sensor. A photodiode emits infra-red light which is reflected off a reflecting surface and received by a photo transistor whose output current depends on the amount of light received. Amount of reflected light depends on the distance d .

Liquid

This kind of actuator has been developed in recent years [6–8], specifically designed to target small camera systems like mobile phones, is already in commercial products. It uses the principle of electro-wetting phenomenon: a drop of two non miscible liquids (an insulator and a conductor) is deposited on a substrate made of metal, covered by a thin insulating layer. The lens curvature is controlled by applying a voltage to the substrate. The focal length changes by varying the applied voltage. In Fig.(4.10)¹ is represented an example of this micro-actuator. In the past this technology suffered from problem of freezing at low temperatures. This was caused by the water when temperature goes below 0° C. This problem has been now solved but there may still be residual problems of converging speed and hysteresis.

4.1.3 The Entire Auto-Focus Environment

To complete this overview on the aspects affecting an auto-focus system it is important to define how are used the concepts defined in Subsections 4.1.1 and 4.1.2.

Referring to Fig.(4.11) a basic scheme on an auto-focus system is reported. A control system receives information from the lens about current scene. To get an in-focus image, in accordance to (4.1) and with the tolerances resumed by (4.20) and (4.21), this control system selects the correct lens position which is communicated to the actuator which moves the lens to its new position.

All this basic explanation is a simplification of real systems. Usually a sharp in-focus image is achieved after a subsequent approximation loop as better explained in the next

¹Source: <http://www.varioptic.com/en/tech/technology-autofocus-optical-image-stabilization.php>

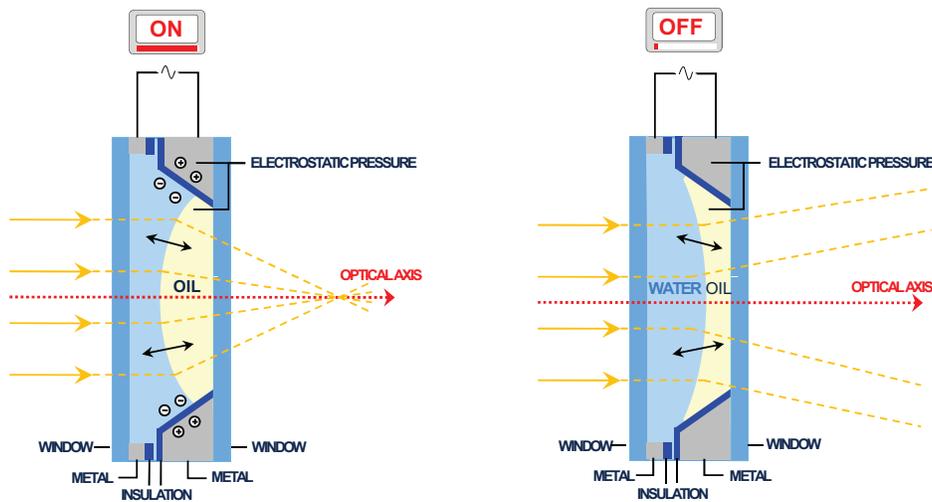


Figure 4.10 : Liquid lens basic scheme and working principles as claimed by Varioptic.

Section.

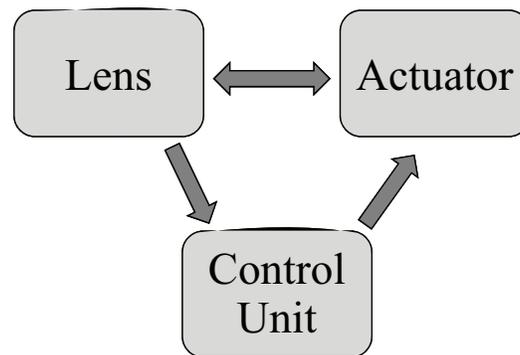


Figure 4.11 : Basic scheme of an Auto-Focus system. Lens is driven by an actuator. An optional position sensor is used to read current lens position and send information to the actuator. A control system analyzing current information moves the lens to the correct position to focus the current scene.

4.2 Auto-focus System

This Section explains in detail the block *Control Unit* of Fig.(4.11). In particular it will be explained how this block understands if a scene under acquisition is in-focus and otherwise how opportunely drives the actuator to get an in-focus image.

4.2.1 Optical Auto-focus

Optical auto-focus is mainly used in SLR cameras where size is not relevant while speed and accuracy are considered top features. In these devices there is a dedicated processor which, through a sophisticated algorithm, detects changes by implementing scene understanding.

Image content and validity of the (4.1) are verified by a dedicated and separated sensor incorporated into the lens system. The principle used to detect amount of de-focus is based on the *phase detection system* [9–12]. Image of the scene passing through the optical system is sent to a separator lens which generates two images. These images are processed by a line sensor. This sensor measures phase differences of similar light intensity peaks and valleys. If these peaks are wider than usual, the system is in rear focus (which means that the focal point is behind the focal plane). If these peaks are narrower than usual, the system is in front focus (the focal point is in front of the focal plane). An illustration of these different working conditions is reported in Fig.(4.12)². This information can be immediately used to understand the direction towards move the lens. Also, the amount of distance of the peaks can be used to estimate the amount of movement the lens should have. In addition in [13, 14] is claimed a method to speed up the convergence time by acquiring focus statistics whilst the lens is moving. Many manufacturers [12, 15] have invented methods capable to track movements of the object and understand if it is moving closer or farther from the lens thus anticipating the new if-focus lens position.

SLR cameras auto-focus has reached very high performances nowadays. It is able to check sharpness in different regions of the image, combine this information with the DoF characteristic of the lens system, prioritize foreground subjects, track movements and compensate for motion blur. A few of these features are being implemented also in compact cameras and are described in Section 4.3.

All these features and benefits have a drawback: they require dedicated sensor, for the phase detection, and processor, to implement all the logic. All this requires additional space and cost which is easily justifiable in SLR market segment.

4.2.2 Digital Auto-focus

In embedded devices (like DSC and phone cameras) it is not possible to use the fast and accurate auto-focus system based on the principle of the phase detection (Section 4.2.1). Neither there is enough space to host the required components nor the extra cost can be justified. For this reason these devices implement a different auto-focus system based on the direct processing of data from the image sensor. This kind of auto-focus is known as digital auto-focus and its high level representation is shown in Fig.(4.13). In this configuration the *control unit* receives information about the focus conditions from a *focus measure* block which, by analyzing in a proper way the image extracted from the sensor, computes a focus measure. This measure is used to determine if the system is in-focus or not and if it is the case to drive the lens to the new correct position.

²Source: <http://www.nikon.com/about/technology/core/software/caf/index.htm>

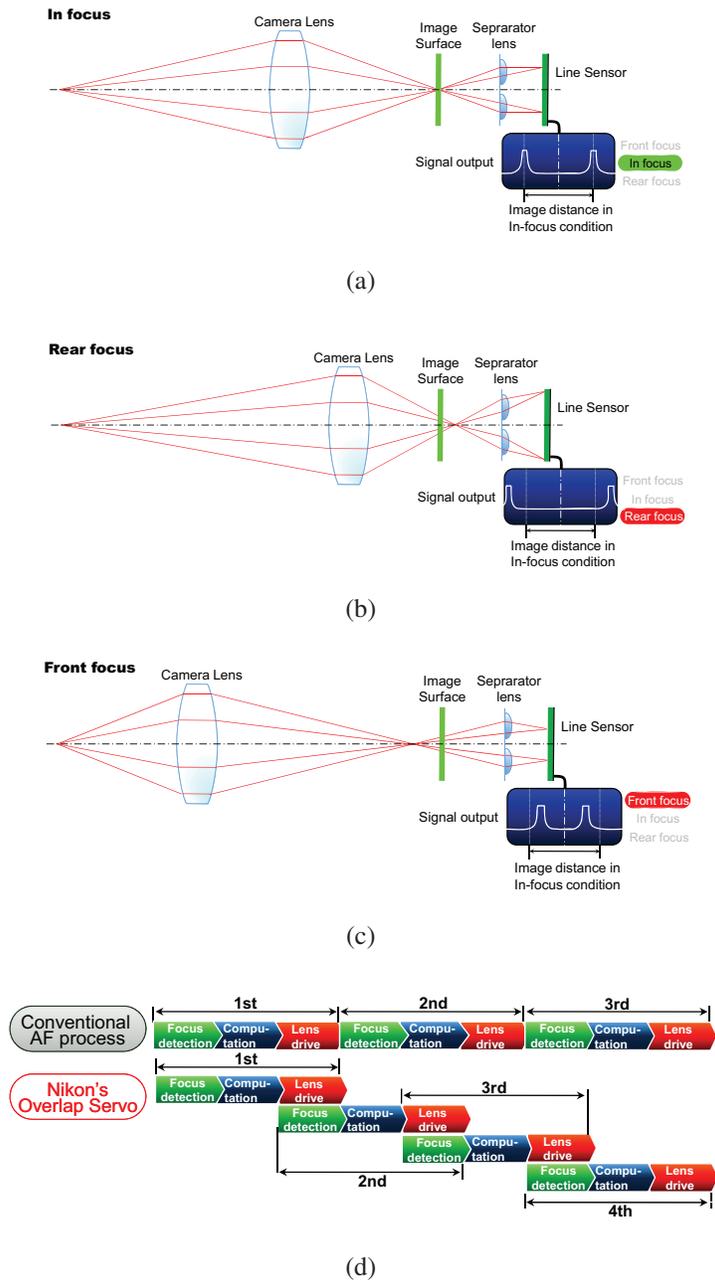


Figure 4.12 : Phase detection based auto-focus system as described by Nikon [12]: **(a)** Lens in the correct position: system in focus; **(b)** Focal point behind the focal plane: rear focus; **(c)** Focal point in front of the focal plane: front focus; **(d)** Nikon patented overlap servo: focus estimation is performed whilst the lens is moving, thus accelerating the convergence time.

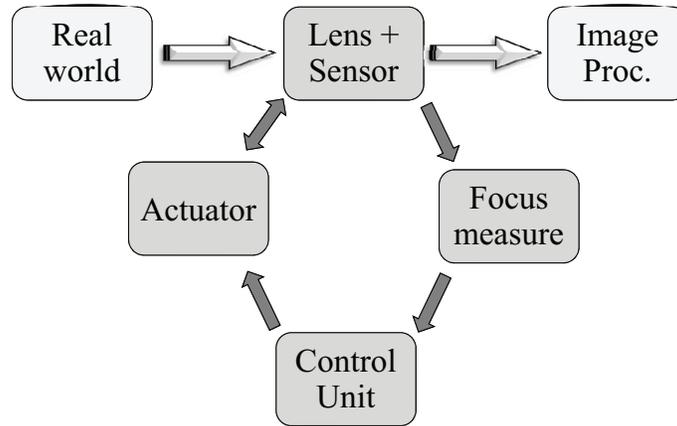


Figure 4.13 : Digital AF general scheme: a focus measure is extracted from the image sensor and sent to the control unit which selects the new lens position to be driven to by the actuator.

Statistics Acquisition

The first step of this process is to define a statistic measure able to represent when the acquired image of a given scene under acquisition is in-focus. In particular this statistic is usually designed to have a value which is maximum for the sharp in-focus image and gradually decreases as the image blur increases. This kind of statistic is usually related to the amount of high frequency contained in the input Bayer Pattern image. In fact it is straightforward from the concepts defined in the previous Sections of this Chapter (Section 4.1.1 on page 57 and subsequent and formally defined by equations in Section 4.4.1) that the more the image is out of focus the more the high frequencies, corresponding to sharp edges, are lost. Several focus measures have been proposed and compared in literature in the past years [2, 16, 17]. The more representative hereby described are based on the following assumptions:

- A grayscale image of $N \times M$ spatial dimension.
- $f(x, y)$ is the focused image of a planar object in accordance with (4.1), where (x, y) corresponds to a point (x, y) of the scene.
- $g_i(x, y)$ is a sequence of images of an object acquired by changing camera parameters (focal length, *f-number*, focusing position); in detail $g(x, y)$ corresponds to (4.27) and it has been obtained by changing only the focusing position.

Variance

Energy of a discrete image can be computed as:

$$E_i = \frac{1}{MN} \sum_x \sum_y g_i^2(x, y) \quad (4.28)$$

The energy variance is a linear and monotonic function computed as:

$$\text{Variance}_i = \frac{1}{MN} \sum_x \sum_y [g_i^2(x,y) - \mu_i]^2 \quad (4.29)$$

where:

$$\mu_i = \frac{1}{MN} \sum_x \sum_y g_i^2(x,y) \quad (4.30)$$

Energy of Image Gradient

This focus measure is represented by a first order derivative of a bi-dimensional digital image and it is expressed as:

$$\sum_x \sum_y [\nabla(g_i(x,y))]^2 \quad (4.31)$$

and the gradient operator ∇ applied to a function g_i at coordinates (x,y) is defined as a vector:

$$\nabla \mathbf{g}_i = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\delta g_i}{\delta x} \\ \frac{\delta g_i}{\delta y} \end{bmatrix} \quad (4.32)$$

Its magnitude is given by:

$$\nabla g_i = \text{mag}(\nabla \mathbf{g}_i) = [G_x^2 + G_y^2]^{1/2} = \left[\left(\frac{\delta g_i}{\delta x} \right)^2 + \left(\frac{\delta g_i}{\delta y} \right)^2 \right]^{1/2} \quad (4.33)$$

Due to the high computational cost of this operation it is common to approximate the magnitude of the gradient by using absolute values:

$$\nabla g_i \approx |G_x| + |G_y| \quad (4.34)$$

or by avoiding the square root operation:

$$\nabla g_i \approx G_x^2 + G_y^2 \quad (4.35)$$

Combining (4.31) and (4.33) and using one of the above simplifications the measure can be expressed as:

$$\text{Sobel}_i = \sum_x \sum_y [G_x(x,y)^2 + G_y(x,y)^2] \quad (4.36)$$

G_x and G_y can be calculated by applying a spatial mask [17] centered in $g_i(x, y)$. The smallest and more used mask is a 3×3 mask and it is known as Sobel operator:

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (4.37)$$

Exploiting (4.34) with the use of the masks (4.37) it is obtained:

$$\begin{aligned} \nabla g_i \approx & |(g_i(x-1, y+1) + 2g_i(x, y+1) + g_i(x+1, y+1)) \\ & - (g_i(x-1, y-1) + 2g_i(x, y-1) + g_i(x+1, y-1))| \\ & + |(g_i(x+1, y-1) + 2g_i(x+1, y) + g_i(x+1, y+1)) \\ & - (g_i(x-1, y-1) + 2g_i(x-1, y) + g_i(x+1, y-1))| \end{aligned} \quad (4.38)$$

Energy of Image Laplacian

It represents a second order derivative of an image by the equation:

$$Laplace_i = \sum_x \sum_y [\nabla^2(g_i(x, y))]^2 \quad (4.39)$$

which is known as Laplace operator. From (4.33) it results that:

$$\nabla^2 g_i = \frac{\delta^2 g_i}{\delta x^2} + \frac{\delta^2 g_i}{\delta y^2} \quad (4.40)$$

To be applied to a digital image this equation needs to be expressed in discrete form. Considering that a first order derivative of a one-dimensional function in the discrete domain can be expressed as:

$$\frac{\delta g_i}{\delta x} = g_i(x+1) - g_i(x) \quad (4.41)$$

and its second order derivative as:

$$\frac{\delta^2 g_i}{\delta x^2} = g_i(x+1) + g_i(x-1) - 2g_i(x) \quad (4.42)$$

For a bi-dimensional function it becomes:

$$\begin{cases} \frac{\delta^2 g_i}{\delta x^2} = g_i(x+1, y) + g_i(x-1, y) - 2g_i(x, y) \\ \frac{\delta^2 g_i}{\delta y^2} = g_i(x, y+1) + g_i(x, y-1) - 2g_i(x, y) \end{cases} \quad (4.43)$$

As for (4.37) the operator (4.40) can be obtained by a 3×3 spatial mask operation performed pixel by pixel:

$$\nabla^2 g_i = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (4.44)$$

or alternatively by:

$$\nabla^2 g_i = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (4.45)$$

Frequency Selective Weighted Median Filter

If we have an image $g_i(x, y)$ where $p(x, y)$ is its pixel of coordinates (x, y) , the frequency selective median filter (FSWM) [16] can be calculated by considering the following spatial mask:

$$FSWM_{Mask} = \begin{bmatrix} 0 & 0 & p(x, y-2) & 0 & 0 \\ 0 & 0 & p(x, y-1) & 0 & 0 \\ p(x-2, y) & p(x-1, y) & p(x, y) & p(x+1, y) & p(x+2, y) \\ 0 & 0 & p(x, y+1) & 0 & 0 \\ 0 & 0 & p(x, y+2) & 0 & 0 \end{bmatrix} \quad (4.46)$$

From this mask it is possible to compute the following median operations valid for a vertical frequency analysis:

$$M_1(x, y) = medv_1 - medv_2 \quad (4.47)$$

where:

$$\begin{aligned} medv_1 &= median \{ p(x, y-2), p(x, y-1), p(x, y) \} \\ medv_2 &= median \{ p(x, y), p(x, y+1), p(x, y+2) \} \end{aligned} \quad (4.48)$$

Equivalently the same can be computed for horizontal frequency analysis:

$$M_2(i, j) = medh_1 - medh_2 \quad (4.49)$$

where:

$$\begin{aligned} medh_1 &= median \{ p(x-2,y), p(x-1,y), p(x,y) \} \\ medh_2 &= median \{ p(x,y), p(x+1,y), p(x+2,y) \} \end{aligned} \quad (4.50)$$

Final FSWM computation on the whole image is performed by the equation:

$$FSWM = \sum_x \sum_y [M_1(x,y)^2 + M_2(x,y)^2] \quad (4.51)$$

Being this statistic dominated by median operators it is almost immune to noise in the very high frequencies. In fact this specially designed median mask is able to adapt its filter response to the real content of the image.

Other Focus Measures

Among the different focus measures which have been proposed in literature [2, 16] one of the most used in the past was Tenengrad [18, 19]. It is a gradient based measure similar to (4.36) but it is threshold based (only gradient magnitude values greater than a given threshold are accumulated). The thresholding operation introduces a few limitations to this measure. The most relevant is that the final in-focus position depends on the value of the threshold and as consequence by changing its value the global maximum might not occur for the best focused image. For this reason nowadays this statistic is no more used.

Nayar [20] has proposed a focus measure based on a new operator called sum-modified Laplacian (SML). It is defined as:

$$SML = \nabla_M^2 g_i(x,y) = \left| \frac{\delta^2 g_i}{\delta x^2} \right| + \left| \frac{\delta^2 g_i}{\delta y^2} \right| \quad (4.52)$$

By summing (4.52) to the entire image it is obtained the SML-based focus measure (SMLF) defined as:

$$SMLF = \sum_x \sum_y \left(\left| \frac{\delta^2 g_i}{\delta x^2} \right| + \left| \frac{\delta^2 g_i}{\delta y^2} \right| \right) \quad (4.53)$$

where the second order derivatives in discrete domains are expressed in (4.43).

Compared with Laplace (4.39) SMLF operator (4.53) sums absolute instead of square values. Usually sum of square values tends to accentuate peak values thus this statistic could suffer lack of precision for not enough sharp peak. As the Laplacian this statistic has the disadvantage that second order derivatives accentuate noise effects. This may causes that in certain conditions noise tends to dominate real information by producing the biggest peak which is misinterpreted as actual in-focus position.

Analysis of Sidelobe Effect

All the statistics discussed in the previous Subsections may be affected by sidelobes in the frequency domain. If the energy content in these frequencies is high it can create local maxima which affect the detection of the actual in-focus position. To reduce influence of sidelobes in the monotonicity of these functions it should be suggested to apply a low pass filter to the image.

Anyway this filter might be considered an unjustifiable additional computational cost and thus usually not implementable. Thus by simply acquiring statistics after a low pass filter there is the double advantage of reducing sidelobes and noise influence in the high frequencies.

If the system has not been correctly designed, frequencies above the sensor Nyquist frequency can introduce aliasing effects which may contribute to a defective in-focus estimation. Noise reduction filter can help to limit this effect as well. As a result detection of the in-focus position is more accurate when statistics are acquired after a noise reduction filter.

Comparing the Focus Measures

To achieve an effective in-focus search, employed statistics should generate focus curves which present:

- **Monotonicity:** it is mandatory that the peak corresponds to the in-focus position and that the curve has only a peak; the search algorithm must be able to understand if the image is blurred or not and eventually decide the direction towards move the lens to reach the in-focus position.
- **Magnitude of slope:** also far from the peak the curve must present a slope so that the searching algorithm can opportunely move the lens; local peak (sidelobe effects) and flatness in highly blurred regions can affect the final result.
- **Smoothness:** the search algorithm must move among the whole curve understanding, step by step, current focus measure and direction to reach the in-focus position; local peaks and not smooth curves tend to falsify the search.

The study reported in [2] highlights that (4.29) is smooth but lacks of slopeness in highly blurred and almost focused portion of the curve.

The focus measures (4.36) and (4.39) are high pass filtering based. This accentuates the sidelobes effects and residual noise influence in the monotonicity and smoothness of the curve.

FSWM (4.51) is a band pass filter operator. It is generally smooth and monotonic, having moderated slopes both in blurred and focused part of the curve and almost sharp peak at the in-focus position. For this reason it is usually referred as a valid focus measure in digital auto focus techniques. An example of effective curve is reported in Fig.(4.14(c)).

4.2.3 Control Unit

Referring to Fig.(4.13) the *control unit* has the task to analyze focus measures and move the lens through the actuator to the correct in-focus position. Although both optical and digital auto-focus systems have a control unit in the following of this Subsection it is referred to a digital auto-focus system (Subsection 4.2.2).

For a given static scene the best and more accurate method to find this position is to perform a full search. It consists in evaluating the focus measure for any lens position and, at the end of this range spanning, selecting the highest position. In Fig.(4.14(c)) is reported an example of this search.

Although this method is actually accurate to find the sharpest position it is time expensive and visually annoying: the former because it is required a spanning of all the lens position; the latter because this spanning displays to the user images which go from blurred to sharp to blurred again before selecting the definitive in-focus position.

To optimize the search many embedded devices use the hill climbing search (HCS). It is based on a method which identifies the slope of the curve and its direction and follows this slope until the top is reached (hill climbing). This top of the hill represents the in-focus position.

By referring to Fig.(4.14(d)) a basic HCS algorithm is compound of three different phases:

- Start (*step 1*): the focus measure for a given lens position is acquired. At this point the system does not know the direction towards which climb the hill so it moves the lens on one step and acquires the new focus measure.
- Direction decision (*step 2*): by comparing current with previous focus measures it is possible to understand slope and direction of the curve. If the previously selected direction is not correct (as in the case of this example) it is inverted and lens moved back of two positions and new focus measure acquired.
- Peak searching (*steps 3-5*): until the focus measure for the current lens position is higher than the previous one the search algorithm goes on. When there is an inversion in the focus values (*step 4*) the system classifies the previous position as in-focus position and goes back there (*step 5*).

The main disadvantage of a search is the time spent to find the in-focus position. A big step size may reduce the converging time with the cost of a not accurate final position; a small step size produces an accurate estimation of the peak position with the cost of the extra time spent to span more positions. To increase the speed of the search and guarantee the final accurateness, HCS may be compound of two consecutive sub-searches:

- Coarse HCS: it allows to roughly estimate the in-focus position by using a coarse search step but it is not able to accurately identify the lens position which produces the sharpest image. The advantage of this search is that the rough in-focus position is quickly identified by fast climbing the hill even if the starting position is far away.

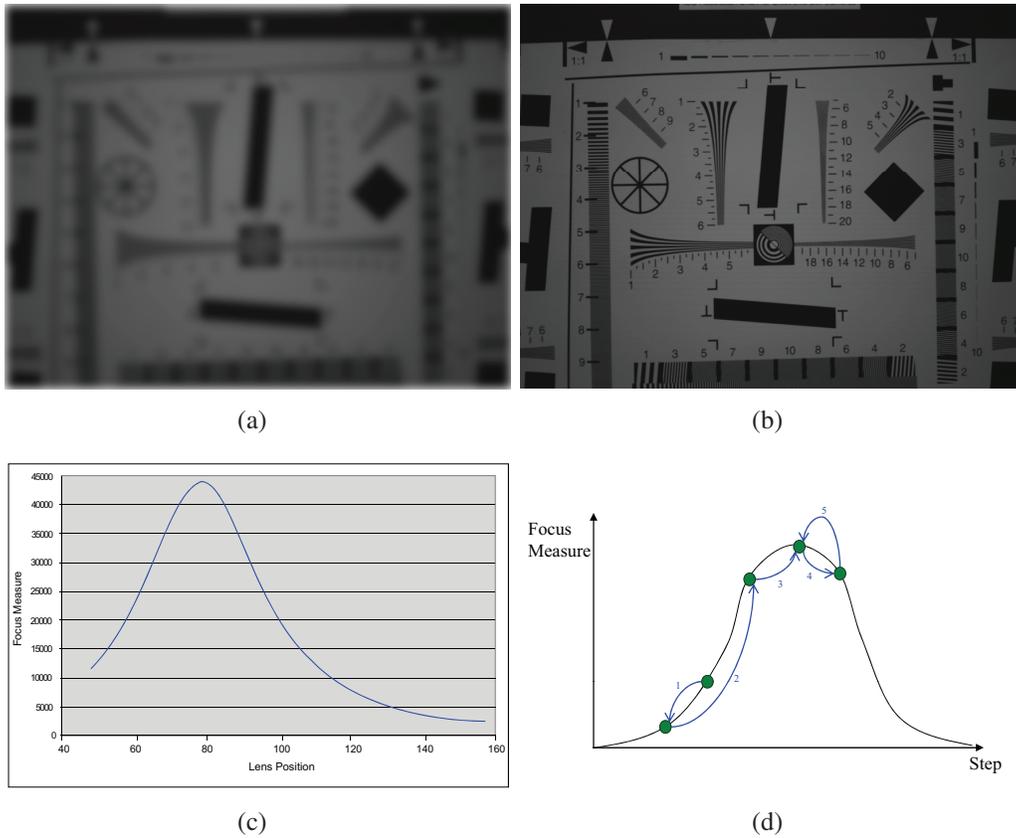


Figure 4.14 : Auto focus search example: **(a)** image of a resolution chart in out-of-focus position; **(b)** image of a resolution chart in in-focus position; **(c)** full search curve plotting values of the focus measures of the resolution chart acquired at each lens position; the in-focus position is selected at the end of the full range spanning; **(d)** hill climbing search, lens positions are selected for increasing focus measures; the search stops when an inversion is detected.

- Fine HCS: starting from the previously approximated in-focus position, a new HCS is performed by using a smaller step size. This fine search allows to identify the sharpest in-focus position and to limit the extra search time by a local fine climbing in the previously identified neighborhood.

Another critical problem which can generate a bad estimation of the in-focus position is a change in the scene during the execution of a search. Unfortunately this exception must be seen as a general scene change: moving objects, light changes and even hand-shaking of camera device can affect the in-focus estimation.

Moving objects in the scene can change frequency distribution and this by consequence affects the curve. Also, an object can move closer or farther from the lens causing a shift in the peak position. If those changes happen while the *control unit* is estimating the curve (slope and peak position) any full search or HCS searching algorithm fails.

To avoid such problems the *control unit* adopts a few control procedures to estimate if a scene or light change is happening. In this event the search is usually stopped and resumed when the perturbation is finished. In this way the search is robust even in presence of external perturbations.

4.3 Working Modes

The working modes hereby described are valid for any auto-focus system. Anyway for the purposes of this book they are specifically related to a digital auto-focus system as described in Subsections 4.2.2 and 4.2.3.

The evaluation of the frequency content of a scene is performed by computing a statistic measure in a delimited area of the image. This is done either to reduce complexity and execution time and to prioritize some portions of the image to others. Statistics are thus collected in a limited number of regions geometrically displaced in the image. In Fig.(4.15) are illustrated some possible examples of displacement.

By changing number and displacement of those regions it is possible to change the sensitivity to focus different portions of the image allowing the *control unit* to adapt the focus strategy to the scene content. In addition, each collected statistic can be weighted so that its contribution to the final evaluation can vary. For example an extreme and widely used weights distribution gives a null value to all the regions except the central one (spot mode); it is used to focus small portion of the image with high precision.

Intelligent auto-focus systems are able to understand the content of the scene and consequently adapt their strategy to automatically focus the main subject. Former systems, even if had not such intelligence, provided the user the possibility to manually select a working mode.

In the next Subsections are described the most common working modes.

4.3.1 Still Auto-focus

Still auto-focus refers to a single (or multiple) capture of a static scene (where static means no moving objects and light changes). Main AF priority goes to achieve the sharpest image. Focusing converging time and computational cost are less important. It is possible to perform a full search (Section 4.2.3) to guarantee an exact localization of the maximum peak of the curve. The searching zones are usually displaced to cover as more as possible the entire image. In the examples illustrated in Fig.(4.15) the more representative configurations are (b), (c) and (f). A HCS can be executed instead of a full search if converging time becomes critical.

Search can be optimized to focus the foreground, the background or specific objects in the scene.

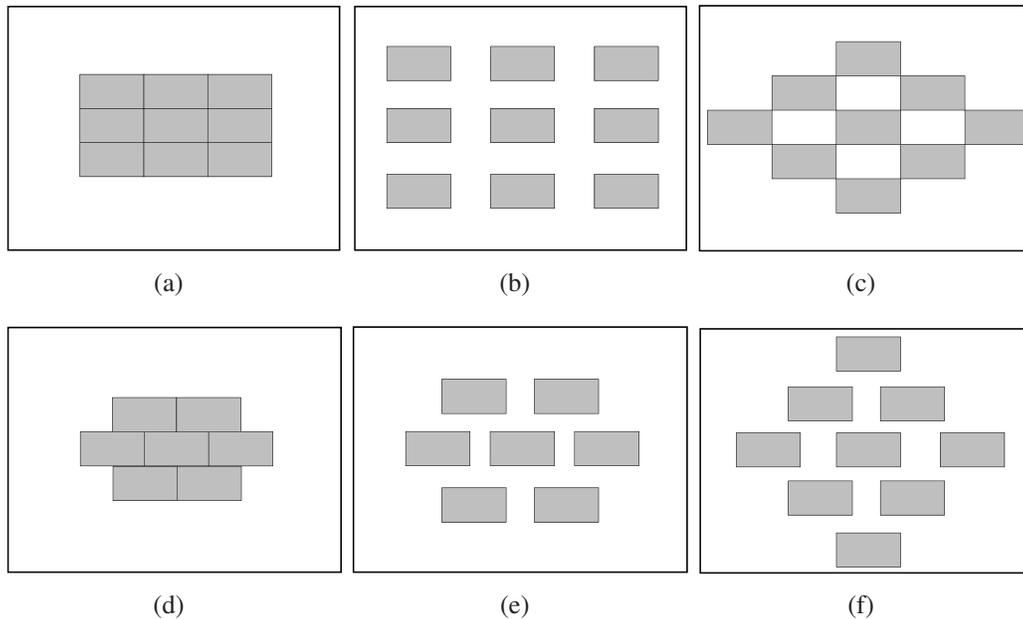


Figure 4.15 : Examples of different displacement of the zones used to estimate frequency content of the scene. Some are displaced in rectangular shape other in diamond shape. Number of zones and displacement can vary from device to device and upon working mode.

4.3.2 Video Auto-focus

In case of video auto-focus (also known as continuous auto-focus) scene and light changes are part of the environment. In opposition to still auto-focus main features of this search are focus speed and capability to maintain the scene in-focus even in presence of movements. For this reason a full search cannot be used (it requires a full spanning of the whole lens range every time a re-focus is required) while a HCS is normally implemented. Differently from a pure HCS, in this case it is mandatory to have a few additional controls to properly manage perturbation events:

- *Light change.* Statistics measure the contrast of a scene and this is influenced by the light level. For a given static scene change of the light conditions causes a change in the exposure settings (Chapter 3) and as consequence a change in the statistic value, Fig.(4.16). This variation must be properly managed or the control unit tends to evaluate it as real scene change and starts to search the new in-focus position. As a result during the video acquisition there is a useless search (the system goes from in-focus to out-focus and then goes back to the same in-focus position without any change in the scene) which is interpreted as a system defect by the user. Usually a light change is detected either with a direct link with the exposure control (through a flag the exposure control informs the auto-focus control system that the light is changing) or with a dedicated internal module which detects light change by

measuring average brightness in the scene or by a combination of both methods for a very accurate light detector based auto-focus.

- *Scene change.* There are different kinds of scene change which may affect the results of a search: intra motion, which consists in single or multiple objects moving inside the scene; inter motion, which is generally caused by a movement of the acquisition system like panning or zooming; both. Any of those movements causes a change in the shape of the focus curve; for example an object moving closer to the lens causes a shift of the peak position toward the macro region. A HCS performed during these changes fails if not properly managed. Here again to avoid visual annoying effects to the user, or bad final focus estimation, the search is stopped and re-executed if a movement is detected. To detect a moving object usually a local motion estimation is computed in the same regions where the statistics are collected. In alternative, the control system can receive a flag from the video stabilization if a global motion, like panning, is occurring during the search. As for light change a more accurate motion detection uses both information.

In Fig.(4.17) is illustrated a block diagram of an auto-focus control system that implements an additional module to detect light and motion changes and properly manage such situations. If a light or motion change is detected the search is not started because the change in the statistic value is not real but caused by it. If this change happens during a search this is stopped until the change finishes and then a new search is started again with the new scene conditions.

Also in case of video auto-focus the search can be optimized to focus the foreground, the background or specific objects in the scene.

4.3.3 Object Priority

Either in case of still or continuous auto-focus the search can be optimized to focus specific content of the scene. The three main classes are portrait, landscape and text. Landscape priority occurs every time the control unit gives priority to the infinity position. At the opposite condition every time the system prioritizes objects in macro positions it is settled in text mode (typical application is to take a picture of a paper or a business card). In the midrange between macro and infinity there is the portrait mode, in which the system tends to prioritize the foreground.

In addition to these very generic working modes during the search statistic information collected from each zone can be separately analyzed. By using this local information the control unit can be targeted to focus specific portions of the image having a specific content. With reference to Fig.(4.18) most common cases of object priority modes are:

- *Biggest object in the scene.* In this case priority goes to the maximum number of zones having the same behaviour. In the case illustrated zones 3 and 4 are selected while the others are discarded. Having very similar curves, in fact, both regions are covered either by the same object or by two different objects at the same distance from the lens. In both cases the target of this working mode is to give priority to the

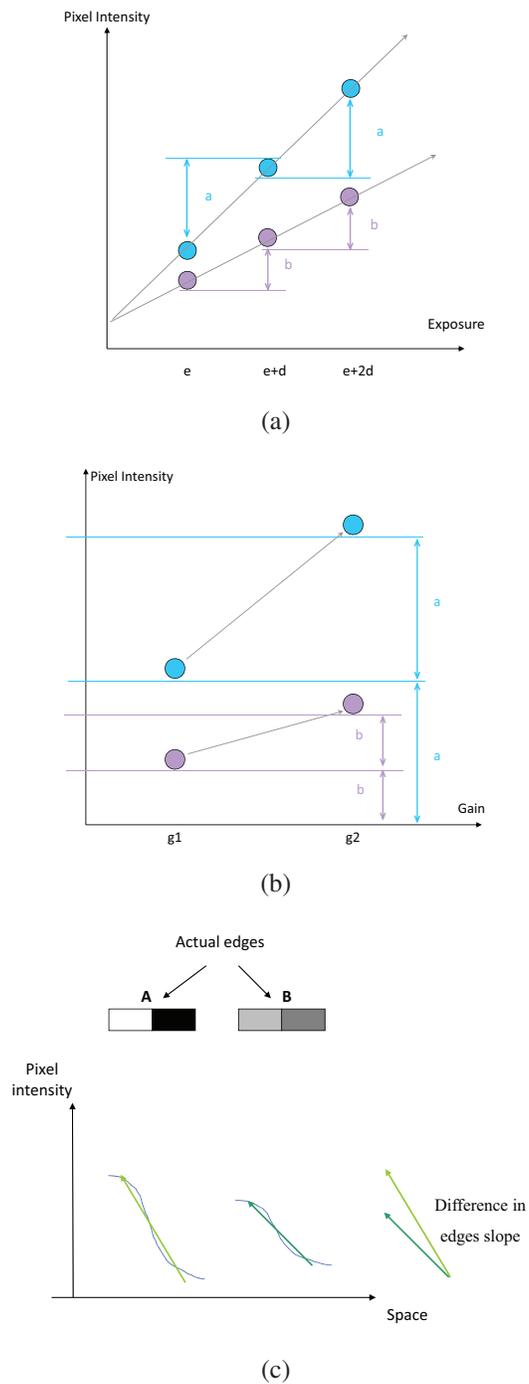


Figure 4.16 : Examples of how changes the edge appearance by changing light conditions and related exposure settings: **(a)** changes in the exposure time; **(b)** changes in the analog gain; **(c)** visual changes in the slope of an edge when exposure settings change. In this case the contrast from **A** to **B** decreases.

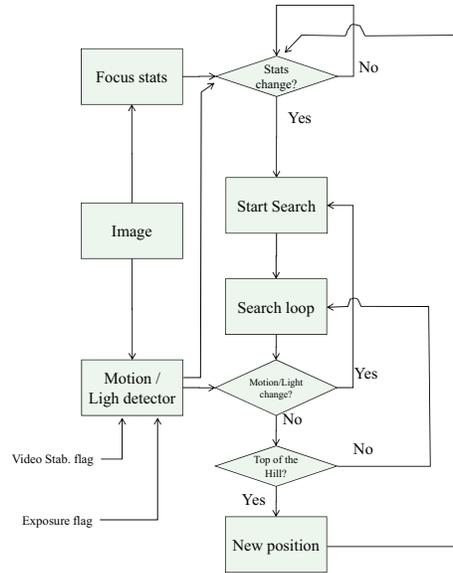


Figure 4.17 : AF control unit block diagram with light and motion detector. If any of them is detected either the search does not start or if already started it is blocked until the change finishes.

most relevant portion of the scene and both regions appear to have a similar content. At the end of the search the system selects the lens position which correspond to the peaks of zones 3 and 4.

- *Closer object to the lens.* In this case the search is optimized to focus the closest object to the lens. By separately analyzing the zones (supposing to move from infinity to macro) the control system skips the zones which present a peak position if there is at least a zone where the peak is not yet reached (the climbing is on the growing phase). The search stops then the latest peak is identified. In the case of the image it corresponds to the zone 1. Opposite behaviour when moving from macro to infinity: the first zone which presents a peak is selected and the others discarded; the search ends at this point. Example of this working mode is macro photography as business card acquisition. It may be useful also as alternative to the above case when foreground is more relevant than the background.
- *Farther object from the lens.* This search is the opposite of the above case. Peaks in the infinity zone are prioritized. In the case of the picture considered as example it corresponds to the curve 5. Target of the working mode is panorama acquisition, where background has the priority on foreground.
- *Central object (or in any specific region).* In this working mode the selection criterion is manually imposed by the user by activating a single (or many over all) region. Typical application is the spot mode, where only the central zone is acti-

vated. By settling this mode it is possible to focus specific portions of the image. In the case under examination any of the zones can be selected and as consequence any of the corresponding curves will produce the final in-focus selection.

- *As more objects as possible.* In this case the selection of the in-focus position prioritizes the maximum number of objects in accordance to the Depth of Field, (4.20) and (4.21). The system selects an appropriate position sensor so that as many zones as possible are in-focus and inside the Depth of Field of the lens system. In the case of Fig.(4.18) zones 3 and 4 are selected and lens position selected so that zone 2 falls inside the DoF. If the system has a wider DoF more zones can fall inside the focused area (for example zone 1 is a good candidate).

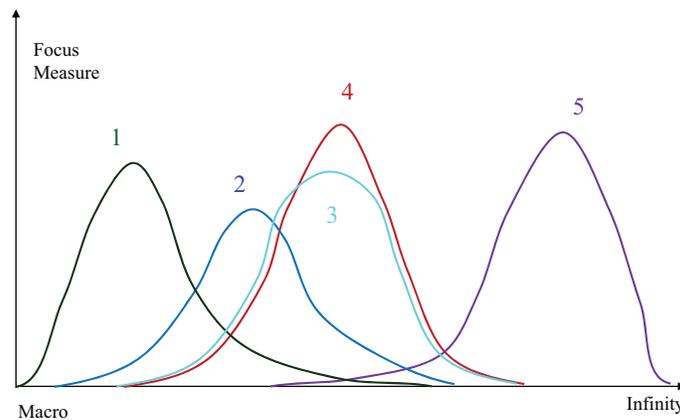
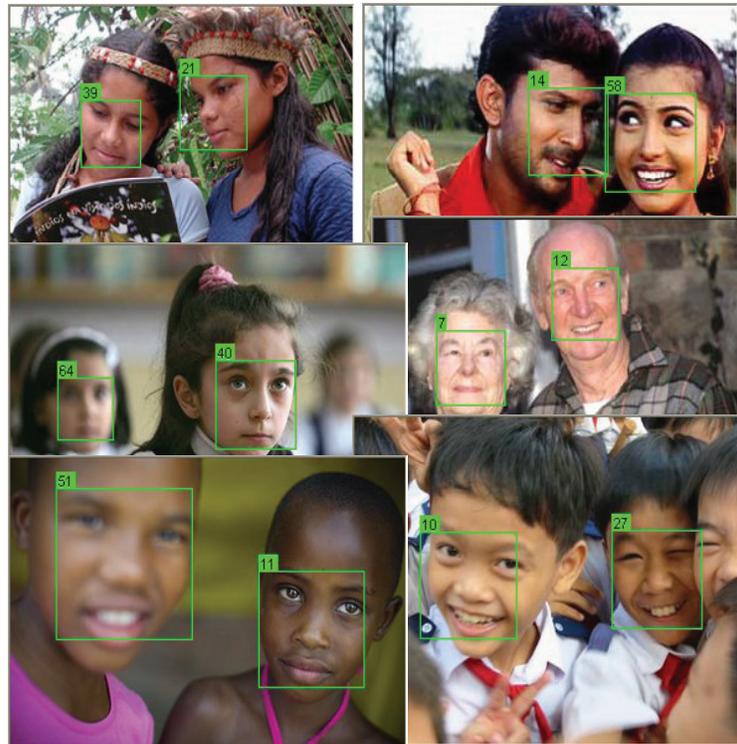


Figure 4.18 : Object priority search: statistics collected through different zones (in the example 5) have a different shape and peak position because they are related to objects displaced at different distances from the lens. Object 1 is closest to the lens, object 5 farthest.

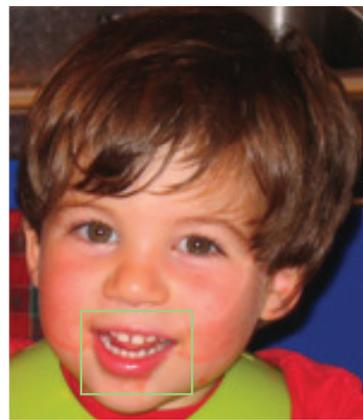
4.3.4 Face Priority

A special class for object detection is the face priority working mode. This priority mode has been widely implemented in the latest years either in professional and commercial acquisition devices. In recent years also embedded devices offer this feature. It is one of the most wanted and used priority not only for its specificity (of course is one of the most frequent subject of a picture) but also because from this many additional features can be derived: from smile and blink eye detection to face priority driven exposure, and auto white balance, and also to support red eye removal.

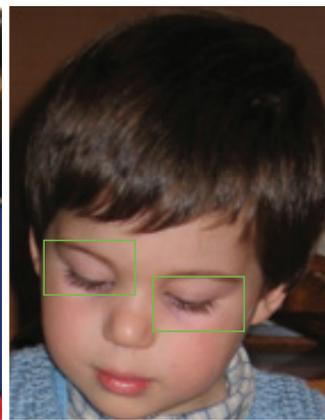
In Fig.(4.19(a)) detected faces are boxed in green. Depending on its orientation, it can face the lens system but rotated, in this case it is referred as Rotation In Plane (RIP), or can not face the acquisition system, in which case it is referred as Rotation Out of Plane (ROP). In the case of the figure boxes numbered 7, 11, 12 and 58 are examples of RIP,



(a)



(b)



(c)

Figure 4.19 : Auto-focus in face priority: **(a)** examples of detection of faces in different rotations (RIP and ROP); **(b)** smile detection; **(c)** blink eye detection: in this case two separate eyes are localized.

the remaining are ROP with different degrees of rotation.

Once the face is identified (in the same figure box 51 validates that also an out of focus face is identified) it is possible to prioritize the control system of the auto-focus to position the regions where statistics are collected on top of the face (or many faces if more than

one are detected simultaneously). The result is a picture with the face in focus. In case of many faces the system tries to optimize the DoF so that as many faces as possible are in focus. Otherwise bigger faces (closer to the lens) have priority on smaller (farther).

Region where the face is detected can be as well used to perform additional controls on the face. Most used are smile (Fig.(4.19(b))) and blink eye (Fig.(4.19(c))) detection. If activated the former is used to prevent a snapshot when the localized person's face is not smiling by delaying the acquisition time. Usually, system implementing this feature let the user the possibility to select among different values of smiling (from faint to big). Similarly the latter delays the acquisition time if a blink eye is detected. It is important to note that in the referred figure the two eyes of a face are detected independently. This help to tune the model to avoid the acquisition also if a single eye is shut.

4.4 Alternative Auto-focus Techniques

In the previous Sections different methods and methodologies to identify the in-focus position and opportunely drive the lens to its correct position have been described. Mandatory requirement for those implementations is a moving system compound of a set of lenses (Subsection 4.1.1), a motor actuator (Subsection 4.1.2), and a control system (Subsection 4.2.3). All these elements require additional space and extra cost, compared to a fixed lens system. On the contrary fixed lens systems suffers of a fixed Depth of Field (Subsection 4.1.1, page 59) with the consequence that, under the market trends nowadays oriented to increase sensor resolution and to decrease the focal length (for compactness of the camera), their useful DoF decreases becoming no more adequate to guarantee the capability to focus from 20cm to infinity (which is a widely used focus range requirement in embedded devices).

Nevertheless low cost imaging systems for mobile applications are still mainly dominated by fixed lens devices. These cameras are implementing new focus strategies to get rid of the above described limitation: the digital auto-focus and the depth of field extension. Both approaches rely on the principle that replacing auto-focus methodologies with dedicated image processing it is possible to achieve a final image with almost the same benefits of a traditional focusing system, at a minor cost and dimensions.

4.4.1 Extended Depth of Field Auto-Focus

This technique is based on the fact that knowing the PSF of a lens even a blurred image can become sharp by a convolution with the inverse of the PSF (Section 4.1.1). Starting from these concepts many techniques have been developed as proprietary solution to extend the original Depth of Field in embedded devices [21–24]. The basic principles on which rely these techniques are illustrated in Fig.(4.20)³: passing through a proper filter positioned on top of the lens system the amount of blur is controlled and limited. A post processing software based on inverse filtering allows to reconstruct the original sharp image.

³Source: <http://www.eece.hw.ac.uk/ceearh2/Optical%20designers%27%20meet%20Sept06.6.ppt>

In Section 4.1.1 it has been seen as for any distance of the object from the lens there is only a valid lens position, Fig.(4.21(a)), to get an in-focus image (4.1) and that, by using the circle of confusion, the aperture of the lens and the diaphragm, it is possible to calculate a lens position range, the dof (4.14), within which the object is in focus. Estimation of the PSF for any level of blur and computation of its inverse to reconstruct the in-focus image is a too long and expensive process. It requires a computation on the fly for any lens position or, alternatively, many PSF functions to be stored in memory. To avoid that, it is possible to control the level of blur and make possible to extend the same level of blur in a wider range than a normal dof allows, as illustrated in Fig.(4.21(b)). It is usually done by a properly designed tiny film to be placed on top of the lens system surface. Unfortunately this process is not as straightforward as can appear. As illustrated in Fig.(4.4(a)) a PSF has many zeros which imply a not invertible function. The PSF is thus adapted to make it invertible and this process generates a lossy reconstruction of a sharp image from a blurred one. These defects are visible as overshoots or halo effects along sharp edges. In Fig.(4.22) is reported a comparison between an acquisition with a classical lens system and the same scene acquired with a wavefront coding provided system resulting in an extended depth of field (EDoF).

4.4.2 Picture Merging

To avoid the intrinsic problems of EDoF techniques an alternative approach consists in taking two or more pictures of the same scene with different lens positions [25, 26]. In this way each picture has a part of the image in focus and the entire dataset is compound so that each part of the scene is in focus at least in an image.

The images so acquired can be properly merged together by taking from each image only

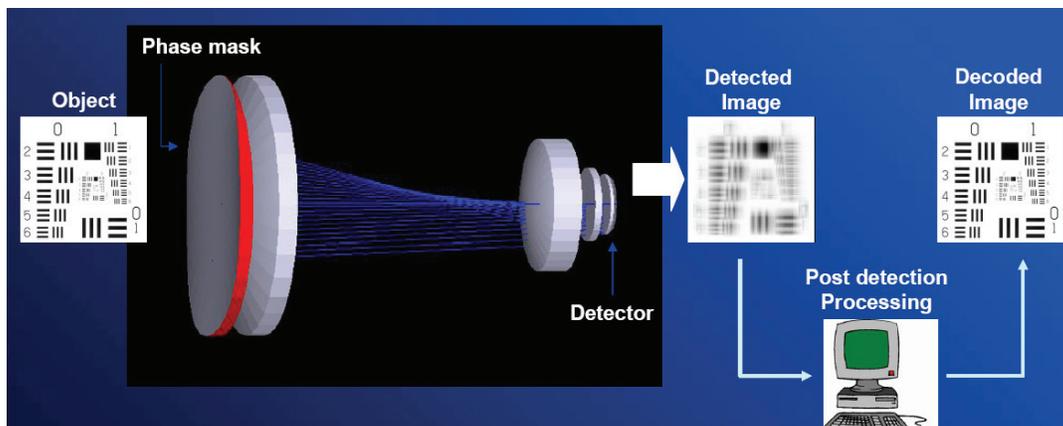


Figure 4.20 : Working scheme of principles used in wavefront coding, as illustrated by Heriot Watt University: passing through a proper filter positioned on top of the lens system the amount of blur is controlled and limited. A post processing software based on inverse filtering allows to reconstruct the original sharp image.

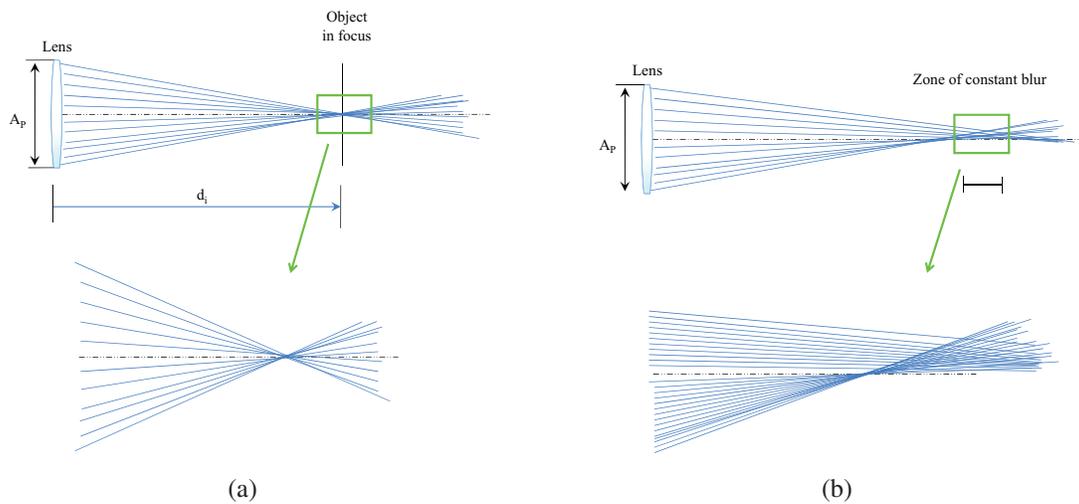


Figure 4.21 : Comparison of the ray height in the focal plane between a classical optical system (a) and wavefront coded system (b) in which the height is controlled and almost constant in a wider plane range.

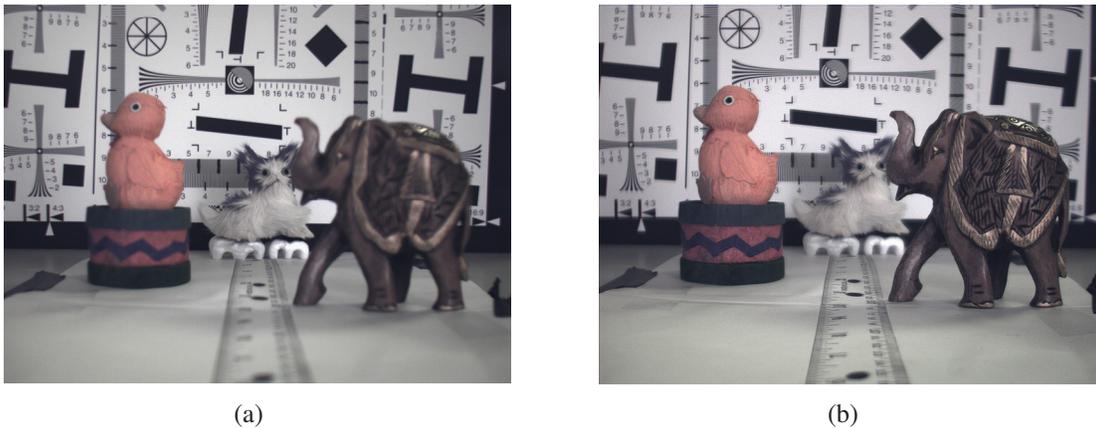


Figure 4.22 : Acquisition of a scene where objects are disposed at different distances: (a) classical lens system; not all the objects can be focused in a single shot due to a limited DoF; (b) by using a wavefront coding system the original DoF is extended and all the objects and the background are almost in focus.

the in-focus portion; the resulting is an image where all the scene is in focus. In Fig.(4.23) are illustrated a few examples of this technique.

Although this method requires neither computational cost nor additional lens as in EDoF techniques it can suffers of other problems. The scene must not change during the multiple-acquisition, which implies that either the scene is still or the framerate of the acquisition system is so high that during the burst the movement is not relevant. As

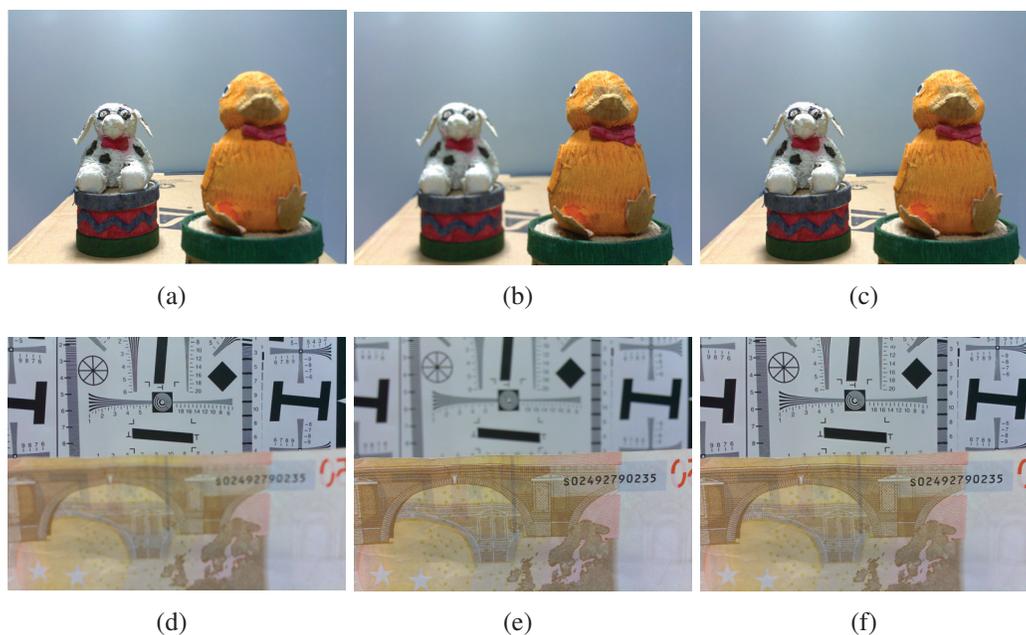


Figure 4.23 : Merging of images having the same scene content acquired at different lens positions: **(a)(d)** back object in focus; **(b)(e)** front object in focus; **(c)(f)** resulting image after the merging process (both foreground and background are in focus).

a consequence this approach is of difficult application in video mode. Also, jagged and fuzzy edges make difficult the segmentation process, required to select from each image only the portion in focus. Finally, embedded devices lens system is affected by parallax when changing the lens position. It is visible as a sort of magnification of close to the lens objects when changing the focus position from macro to infinity. This causes an overlapping of the foreground with the background and if not properly managed halo artifacts may appear around the foreground object as illustrated in Fig.(4.24(c)). In these cases a more accurate edge detection and segmentation may help to reduce the visual annoying effects, specially if coupled with an opportunely tuned blending algorithm during the merging process, as visible in Fig.(4.24(d)).

Conclusions

In this Chapter the principal characteristics of an auto-focus system for embedded devices have been outlined, describing fundamental components, main requirements and features. In the next product generation market trends highlight a continuous improvements of the performances so far implemented with special regards to faster actuators coupled with high accuracy to focus in still and video mode, either in macro and panoramic, more functionalities as intelligent face detection, object tracking and predictive auto-focus and, of course, more thinness and less cost of the whole device.

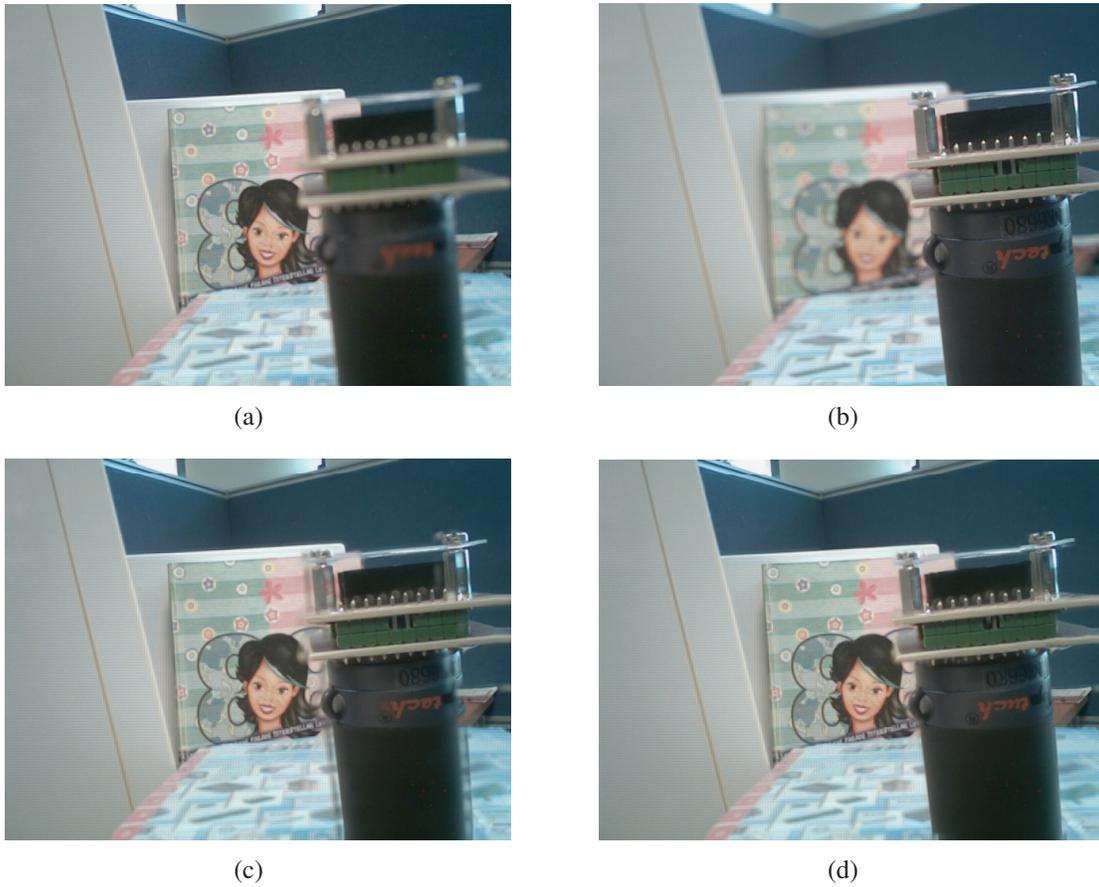


Figure 4.24 : Effects of parallax and magnification in picture merging: (a) background in focus; (b) foreground in focus; (c) picture merging with visible defects around the foreground object; (d) an appropriate processing picture merging is able to remove those defects.

Bibliography

- [1] H. M. Merklinger, *The INs and OUTs of FOCUS*. Seaboard Printing Limited, 1992.
- [2] M. Subbarao, T. Choi, and A. Nikzad, “Focusing techniques,” *Journal of Optical Engineering*, vol. 32, pp. 2824–2836, 1993.
- [3] R. Imai, “Piezoelectric actuator, lens driving device, and image taking device.” *United States Patent Application* 20070035202, <http://www.freepatentsonline.com/20070035202.html>, February 2007. Fuji Photo Film Co., LTD.
- [4] J. Allan, G. Mckevitt, M. R. Shepherd, and R. Topliss, “Piezoelectric devices.” *United States Patent* 7486004, <http://www.freepatentsonline.com/7486004.html>, February 2009. 1 Limited.
- [5] J. Allan, D. H. Pearce, and R. Topliss, “Electro-active actuator.” *United States Patent Application* 20050269906, <http://www.freepatentsonline.com/20050269906.html>, December 2005. 1 Limited.
- [6] A. Obrebski and F. Strahle, “Imaging optics with adjustable optical power and method of adjusting an optical power of an optics.” *United States Patent* 7411739, <http://www.freepatentsonline.com/7411739.html>, August 2008.
- [7] B. Berge, J. Peseux, B. Boutaud, and P. Craen, “Method of manufacturing an electrowetting-based variable-focus lens.” *United States Patent Application* 20090021842, <http://www.freepatentsonline.com/20090021842.html>, January 2009. Varioptic S.A.
- [8] B. Berge and J. Peseux, “Lens with variable focus.” *United States Patent* RE39874, <http://www.freepatentsonline.com/RE39874.html>, October 2007. Varioptic.
- [9] D. A. Kerr, “Principle of the split image focusing aid and the phase comparison autofocus detector in single lens reflex cameras.” http://doug.kerr.home.att.net/pumpkin/Split_Prism.pdf, August 2005. Issue 5.
- [10] H. Yasuda and M. Hirasawa, “Lens controlling apparatus.” *United States Patent* 5594311, <http://www.freepatentsonline.com/5594311.html>, January 1997.
- [11] M. Shirai, R. Ogawa, S. Suzuki, and H. Takayama, “Autofocus distance-measuring optical system.” *United States Patent* 6480266, <http://www.freepatentsonline.com/6480266.html>, November 2002.
- [12] Nikon Corporation, “Predictive focus tracking system.” <http://www.nikon.com/about/technology/core/software/caf/index.htm>, March 2008. Nikon.

- [13] A. Ogasawara, "Control method for an auto focus apparatus." *United States Patent* 5177525, <http://www.freepatentsonline.com/5177525.html>, January 1993. Nikon Corporation.
- [14] S. Uchiyama, Y. Kusaka, and S. Yamano, "Autofocus camera." *United States Patent* 5469240, <http://www.freepatentsonline.com/5469240.html>, November 1995. Nikon Corporation.
- [15] M. Higashihara, I. Ohnuki, A. Akashi, and T. Kadohara, "Auto focus device with predictive focussing." *United States Patent* 5089843, <http://www.freepatentsonline.com/5089843.html>, February 1992. Canon Kabushiki Kaisha.
- [16] K. S. Choi, J. S. Lee, and S. J. Ko, "New autofocusing technique using the frequency selective weighted median filter for video camera," *IEEE Transactions on Consumer Electronics*, vol. 45, pp. 418–420, August 1999.
- [17] R. C. Gonzales and R. E. Woods, *Digital Image Processing*. Addison Wesley, 1st edition ed., 1992.
- [18] J. M. Tenenbaum, *Accommodation in computer vision*. PhD thesis, Stanford University, Stanford, CA, 1970.
- [19] J. F. Schlag, A. C. Sanderson, C. P. Neuman, and F. C. Wimberly, "Implementation of automatic focusing algorithms for a computer vision system with camera control," Technical report CMU-RI-TR-83-14, Carnegie Mellon University, August 1983. Robotics Institute.
- [20] S. K. Nayar and Y. Nakagawa, "Shape from focus," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 824–831, August 1994.
- [21] E. R. Dowski and W. T. Cathey, "Extended depth of field through wavefront coding," technical report, Imaging Systems Laboratory, Department of Electrical Engineering, University of Colorado, Boulder, Colorado 80309, 1995. <http://www.colorado.edu/isl/papers/edf/paper.html>.
- [22] R. Narayanswamy, A. E. Baron, V. Chumachenko, and A. Greengard, "Applications of wavefront coded imaging," in *Computational Imaging II Proceedings volume #5299*, CDM Optics Inc, Boulder, Colorado 80303, IS&T/SPIE Electronic Imaging 2004, San Jose, CA, 2004.
- [23] Z. Zalevsky and J. Solomon, "Optical method and system for enhancing image resolution." *United States Patent Application* 20070040828, <http://www.freepatentsonline.com/20070040828.html>, February 2007. Xceed Imaging LTD.

- [24] G. Muyo and A. R. Harvey, "Decomposition of the optical transfer function: wavefront coding imaging systems," *OPTICS LETTERS*, vol. 30, pp. 2715–2717, October 2005. School of Engineering and Physical Sciences, Heriot-Watt University, Riccarton, Edinburgh EH14 4AS, United Kingdom.
- [25] A. Castorina, A. Capra, S. Curti, M. La Cascia, and V. Lo Verde, "Extension of the depth of field using multi-focus input images," in *IEEE International Symposium on Consumer Electronics (ISCE-04)*, (Reading, UK), pp. 146–150, 2004.
- [26] A. Capra, A. Castorina, P. Vivirito, and S. Battiato, "Adaptive image data fusion for consumer devices application," in *IEEE 6th Workshop on Multimedia Signal Processing (MMSP-04)*, (Siena, Italy), pp. 243–246, September 2004.

Color Rendition

A. Bruna, F. Naccari

Advanced System Technology - Catania Lab, STMicroelectronics, Italy.

Abstract: The color reproduction accuracy of digital imaging embedded devices is a key factor of the overall perceived image quality. The human visual system, under most conditions, is able to compensate for the effects of the scene illuminant on the perceived image. Therefore on digital imaging devices, some processes have to be performed across the image generation pipeline in order to obtain an effective color accuracy regardless of the scene illuminant and the sensor response features. In this chapter we describe the most common color processing algorithms performed across the image generation pipeline: white balancing algorithm, which is aimed to compensate the effects of the illuminant power spectral distribution and the color correction process, which compensates the mismatch between the color filters array transmittance and the color response of the human visual system to different wavelengths. The first one, on embedded devices, is usually performed through an image statistical analysis to obtain an estimation of the scene illuminant and is often based on strong assumptions on scene spectral reflectance distribution. The second relies on the characterization of the sensor color filters spectral transmittance. We illustrate the role of such algorithms on the overall perceived color image quality and describe typical methods for white balancing performance and sensor characterization benchmarking. We describe also some additional algorithms which, frequently on consumer devices, can be used to improve the visual appearance of common colored objects (e.g., skin tones, vegetation and sky).

5.1 Introduction

An image acquired by a digital camera can be seen as a function ρ mainly dependent on three physical factors: the illuminant spectral power distribution $I(\lambda)$, the surface spectral reflectance $S(x, y, \lambda)$ and the sensor spectral sensitivities $\mathbf{C}(\lambda)$. Using this notation, the sensor responses at the pixel with coordinates (x, y) can be thus described as:

$$\rho(x, y) = \int_{\omega} I(\lambda)S(x, y, \lambda)\mathbf{C}(\lambda)d\lambda, \quad (5.1)$$

where ω is the wavelength range of the visible light spectrum, ρ and $\mathbf{C}(\lambda)$ are three-component vectors. Since the three sensor spectral sensitivities are more sensitive respectively to the low, medium and high wavelengths, then the three-component vector $\rho = (\rho_1, \rho_2, \rho_3)$ represents the sensor response to the scene (see Fig.(5.1)). The digital conversion of the sensor response, which is performed by an analogue to digital converter, leads to a numerical representation of the image, which is referred to as the sensor or camera raw response $RGB = (R, G, B)$ triplet. Unlike human vision, imaging devices (such as digital cameras) cannot adapt their spectral responses to different lighting conditions. To recover the original appearance of the scene under different lighting, the camera raw response must be transformed. Such transformations are called chromatic adaptation models, and are the basis of several color balancing methods available in the literature [1–3]. A chromatic adaptation model [4], does not include appearance attributes, such as lightness, chroma and hue, but simply provides a transformation from tristimulus

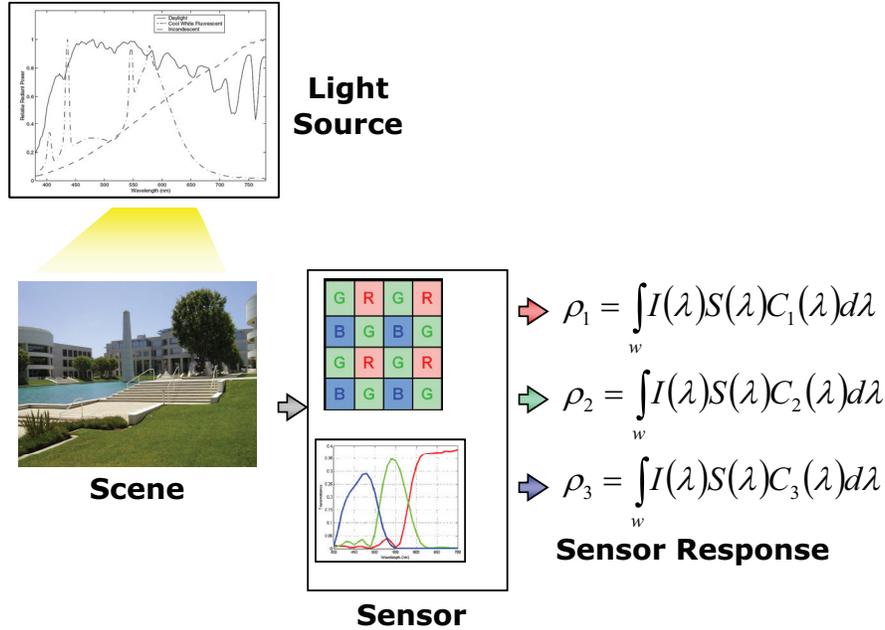


Figure 5.1 : Channels response of a digital color image sensor.

values in one viewing condition to matching tristimulus values in a second set of viewing conditions. Most of these models are based on the Von Kries [5, 6] hypothesis, which states that chromatic adaptation is an independent gain regulation of the three cone signals L , M and S , through three different gain coefficients. In different models the particular values of the coefficients are obtained with different procedures. In these models the R , G and B channels are usually considered an approximation of the L , M and S retinal wavebands [7], so that the post-adaptation values, $(R, G, B)_{awb}$ can be obtained by a Von Kries diagonal transform. Moreover, since the primaries of the target color space do not correspond with the primaries of the sensor chromatic response, thus the chromatic adapted response has to be modified by applying color processing that tries to match the sensor response to output color space. The commonly adopted solution is a color matrix and a tonal curve [8]. The chromatic sensor spectral response distribution affects the quality of the color reproduction. In particular, if the spectral response is too wide, then some subtle differences between two colored objects will not be reproduced, otherwise if the sensor spectral response is too narrow, the camera sensitivity will be reduced, thus leading to noisy images. Manufacturers generally design the color filter array spectral response in order to obtain the best trade-off between signal to noise performance and color gamut, which is the subset of colors which can be accurately represented in a given circumstance by a certain device.

5.2 Color Processing Pipeline Model

According to the chromatic adaption models described in the previous Section, we can represent a simplified color processing pipeline by the following equation:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix}_{out} = \left(\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} r_{wb} & 0 & 0 \\ 0 & g_{wb} & 0 \\ 0 & 0 & b_{wb} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}_{in} \right)^\gamma \quad (5.2)$$

where RGB_{in} are the camera raw RGB values, the diagonal matrix $diag(r_{awb}, g_{awb}, b_{awb})$ is the independent channels gain compensation of the illuminant, the full 3-by-3 matrix $a_{(i,j)}$, $i, j = \{1, 2, 3\}$ is the color space conversion transform from the device-dependent color filter array representation of RGB to the target color space, γ is the gamma classic correction to map the sensor linear response to a non linear response of the human visual system, typically defined for each color space where the final image will be represented. Commonly, for digital imaging consumer devices, the target color space is the $sRGB$ (e.g., ITU-R BT.709). The color response of a digital camera is then mainly performed by these two matrix operation: a diagonal transform and 3x3 color space conversion. As described in different Chapters, numerous algorithms across the image reconstruction pipeline modify the sensor camera raw response in order to correct and improve the quality appearance of the final image. In this Chapter we focus our attention on those processes which mainly determine the color appearance of the final image, avoiding to further describe the possible interaction between the color processing algorithms and other algorithms of the image reconstruction pipeline, which should be designed to not interact with the color rendition in most conditions.

5.3 White Balance

In order to render the acquired image as close as possible to what a human observer would have perceived if placed in the original scene, the first stage of the color correction pipeline aims to emulate the color constancy feature of the human visual system (HVS) (i.e., the ability of perceive relatively constant colors when objects are lit by different illuminants). As described before, on digital devices this process is carried out by the white balancing process usually through a spatial uniform diagonal transform as described by (5.3). The process can be performed in a fixed way, by associating a set of channel gains to the most common scene illumination situations. Typically the following situations are considered: daylight, direct sunlight, incandescent light, fluorescent light, flash gun light, overcast, sunlight shadows. For each illuminant the set of corrective channel gains is calculated off-line and depends mainly to the sensor spectral response to the light. The set of corrective gains can be otherwise estimated, during the image processing pipeline execution, by an algorithm which is referred as automatic white balancing (AWB). Since the gains estimation is an under-constrained problem [1, 9, 10], strong assumptions are usually adopted in order to solve for it. Such assumptions are generally related to the spectral distribution of the scene reflectance. In this case the set of corrective gains mainly depends

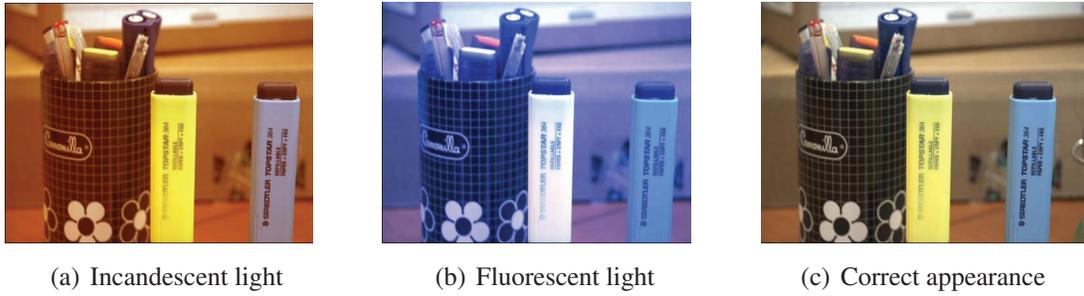


Figure 5.2 : Scene appearance under different illuminants and their correct white balanced version.

on statistical distribution of pixels levels, other system metadata and on the sensor chromatic response. Some professional cameras can use other specific sensors to estimate the illuminant; consumer devices perform the estimation of corrective gains just on the digital raw image features. The following equation describes the white balancing application process:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix}_{wb} = \begin{bmatrix} r_g & 0 & 0 \\ 0 & g_g & 0 \\ 0 & 0 & b_g \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}_{in} \quad (5.3)$$

5.3.1 White Balancing Estimation Techniques

All embedded digital imaging systems implement an automatic estimation of the white balancing coefficients (r_g, g_g, b_g) of the diagonal matrix of (5.3). Several methods exist in the literature with some excellent comparisons and reviews [1, 2, 9]. Since illuminant estimation on the sensor raw image is in a strictly way an under constrained problem, some strong assumptions have to be made on the scene reflectance distribution. In Fig.(5.2) are reported two captured images of the same scene under different illuminants. They are raw sensor images represented as color images, where no color interpolation has been performed. In Fig.(5.2(c)) is reported the correct white balanced image as it should appear if the corrective estimated gains were perfect, starting from both the different conditions.

One of the most common assumption is widely known as the gray-world assumption. It assumes that, given an image of sufficiently varied colors, the average surface color in a scene is gray [11, 12]. Hence, the shift from gray of the measured averages on the different chromatic channels is due to the color of the illuminant. In most conditions, where complex scenes are represented, it has a good match with the scene features. The three scaling coefficients in (5.3) are therefore set to compensate the shift from gray. Assuming that the channels averages are $(R_{avg}, G_{avg}, B_{avg})$, the corrective gains can be

easily calculated as follows:

$$\begin{bmatrix} r_g \\ g_g \\ b_g \end{bmatrix} = \begin{bmatrix} \frac{R_{gray}}{R_{avg}} \\ \frac{G_{gray}}{G_{avg}} \\ \frac{B_{gray}}{B_{avg}} \end{bmatrix} \quad (5.4)$$

where $R_{gray} = G_{gray} = B_{gray}$ are typically the coordinates of middle tone gray (e.g., 128 for a 8-bit representation). Another common assumption is based on the hypothesis that the brightest object on the scene is white, thus it reflects the spectrum of the light. In this case the white balancing correction gains can be calculated:

$$\begin{bmatrix} r_g \\ g_g \\ b_g \end{bmatrix} = \begin{bmatrix} \frac{R_{white}}{R_{max}} \\ \frac{G_{white}}{G_{max}} \\ \frac{B_{white}}{B_{max}} \end{bmatrix} \quad (5.5)$$

where $R_{white} = G_{white} = B_{white}$ are the reference signal levels for white representation and $R_{max} = G_{max} = B_{max}$ are generally the channels average levels of the brightest area of the scene. On complex scenes both these assumptions have often a good match. When few objects and colors are represented, these often fail, thus producing poor white balancing result and evident color cast. In practice, scenes with few highly chromatic objects, such as closeup or macro images, have a poor match with the gray world assumption. In such situations, assuming that the average reflectance of the entire scene is gray could be risky and leads to a poor white balancing estimation. Moreover, the system acquisition limits, like noise, vignetting and limited dynamic range can affect the white balancing estimation effectiveness. In particular the white patch assumption could be broken in the case of noisy low light situations, where signal spikes could be erroneously considered as coming from a white reflectance object. Moreover, the limited dynamic range of the sensors used on embedded devices, produces in most situation clipped signals, thus providing a distorted representation of the white objects. Usually, in order to reinforce the assumptions and limit the interaction between system distortions and white balancing estimation, more sophisticated algorithms can be implemented. Image color based segmentation can be used to limit the image statistical analysis to the regions where the matching with the underlying assumption is higher. Authors in [13–15] investigate the role of low and high chromatic objects on the white balancing estimation. These approaches increase the weight of the neutral objects during the image statistical analysis, thus providing better results when the whole actual image is not neutral. A statistical investigation on the chromaticity distribution of the image has been proposed by the authors in [16], relying mainly on the observation [17] that not all possible chromaticities can be reproduced by a sensor under all possible source lights. This approach requires a sensor chromatic response characterization by analyzing the chromaticity distribution that a sensor produces under specific illuminants on a set of several reflectances [18]. This information and the observed chromaticity of an image, on which the illuminant is to be estimated, are statistically processed. A correlation matrix between the image data and each possible illuminants is calculated, thus producing a likelihood score for each illuminant. All con-

sidered illuminants can be assumed with the same probability, but further assumptions can be done on the statistical distribution of common light sources. The relationship between the camera color response and the signal level has been also investigated [19], demonstrating that brighter pixels have a greater impact on the illuminant estimation process than the darker areas of the image. The color response of the sensor is strictly related to the color filter array spectral response. Highly correlated spectra tend to produce quite similar chromaticity gamuts under different source lights. In order to improve the separation of chromatic response under different source lights, some authors [20, 21] have proposed a pre-processing step aimed to reduce the sensor spectral correlation before the color constancy image processing step, showing that evident benefits could be obtained. Moreover, the color representation domain has also a role in the illuminant estimation accuracy as demonstrated in [22]. As mentioned above, most of these techniques produce an effective source light estimation when the image to be classified contains a lot of different colors. Other color representation models have been investigated, which could be able to produce an estimation on image with only two colors [23–25]. They are based on the dichromatic reflection model (DRM), which can be described as follows:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \alpha \begin{bmatrix} R_o \\ G_o \\ B_o \end{bmatrix} + \beta \begin{bmatrix} R_w \\ G_w \\ B_w \end{bmatrix} \quad (5.6)$$

where $(R_o, G_o, B_o)^t$ is the color of an object diffused reflection and $(R_w, G_w, B_w)^t$ is the color of the specular reflection (i.e., the color of the illuminant). The model states the observed colors of an object with homogeneous color $(R, G, B)^t$ are then distributed on a plane in the RGB color space. Hence, if another homogeneous object is present in the image with colors $(R', G', B')^t$, will be distributed on another plane of the color space. Since the illuminant is the same for both objects, then the two planes intersect at the illuminant color vector. The illuminant estimation process, which in theory could be performed on images with just two colored objects, relies then on the segmentation of colored objects and the estimation of the intersection line (parametrized by α and β) which describes the illuminant color. The object segmentation and the identification of color planes is the critical factor for the effectiveness of such approach. A two-dimensional dichromatic model can also be adopted. The chromaticity diagram can be calculated by the following equation on the normalized CIE-xy normalized plane:

$$\begin{aligned} x &= \frac{X}{X+Y+Z} \\ y &= \frac{Y}{X+Y+Z} \end{aligned} \quad (5.7)$$

where X, Y, Z are the tri-stimulus values of CIE 1931 XYZ color space [26]. If many objects are present on the scene, the chromaticity of the light can be calculated as the intersection of the multiple lines that are related to different objects on the chromaticity plane. The object identification is a key factor for the effectiveness of the source light

color estimation on both three and two-dimensional dichromatic based models. Dichromatic model have been proposed also for the general problem of the mixed light situations, where not just one dominant illuminant is present on the scene. In fact in color science, it is known that the chromaticity of the mixture of two colored lights is on the straight line between the chromaticities of the two colors on a chromaticity diagram. A common case of mixed light situation is related to the separation of shadows on the images [27], which could lead to a basic scheme of image segmentation for the detection of the dominant light of the image and then to possible spatial adaptive correction [28] for illuminant effects.

5.3.2 Pixels vs. Colors

As described previously, different domains can be used in order to analyze the image features to the aim of solving the color constancy problem. Most of classic approaches, like gray world typically adopt a pixel based analysis, thus giving more relevance to objects that fill most of the scene. Accurate segmentation process can be required in order to minimize the effects of dominant objects on the estimation of light source color. The wide diffusion of digital photography and mobile camera phones has leaded users to take photos that probably would have not been taken with classical film cameras. Most of these unusual pictures can easily be critical for light source estimation. In this Section we want to describe what happens when few colors are present on the scene if we adopt either a pixel based analysis or a color based analysis of the image. Commonly, close up situations belong to this risky category, since objects tend to cover most of the scene with few colors, thus producing a mismatch with widely used gray world assumption to solve for color constancy. Fig.(5.3) reports a scheme of a critical situation with few colored objects present on the scene, where the black frame represents the viewfinder position. It is easy to observe that a small movement on hand held cameras, as described in Fig.(5.3) can easily modify the scene observed by the device Fig.(5.3(b)). If we look at the image statistics in terms of pixels, the color histograms of the image change with such small shift, but if we look just at the image colors, these will be unchanged by such small movements.

In order to clarify the role of the colors and pixels it is possible to observe on a real scene what happens when few colored objects are introduced on the scene. Fig.(5.4) reports two different fields of view of the same scene and the effects that the addition of few colored objects can produce on the distribution of the chromaticity, reported in the normalized rb channels plane. On the left just one dominant color is present on the scene, thus producing poor information for the source light color estimation. On Fig.(5.4(b)) it is evident that few colored objects added to the field of view add more information that can be used to estimate the source light color.

5.3.3 Source Light Classification

The illuminant estimation relies on the concept of CCT (correlated color temperature), which describes a light source as the correlated absolute temperature (Kelvin) of the

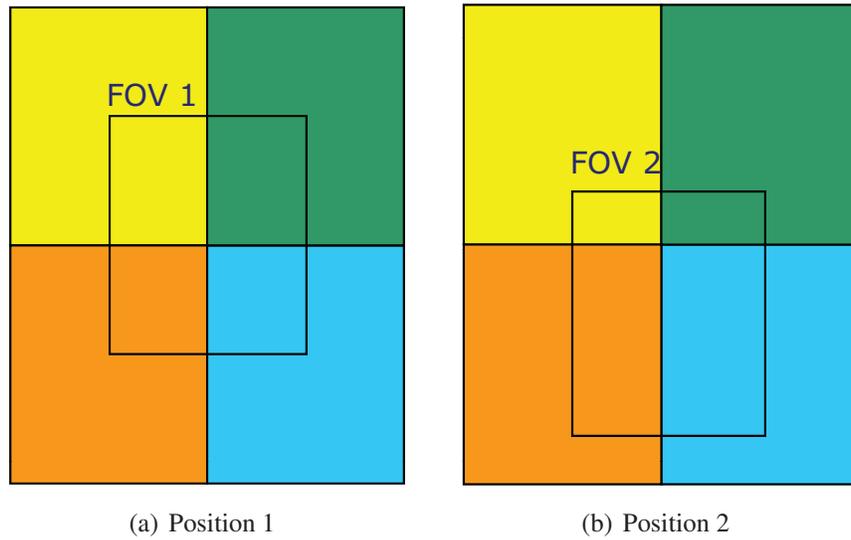
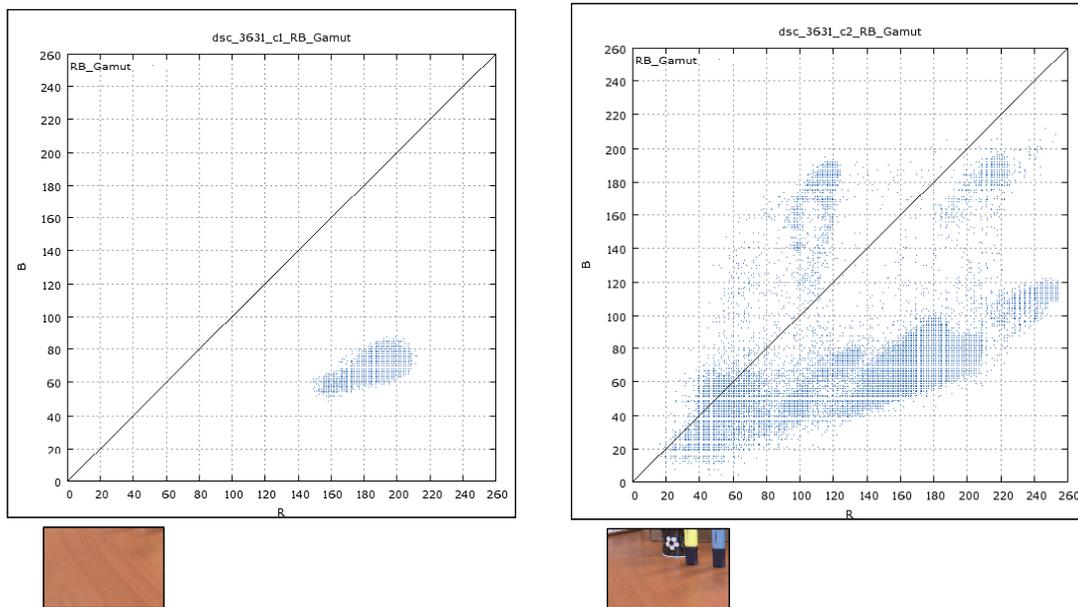


Figure 5.3 : Simulation of a close-up scene with two different fields of view.



(a) A single dominant color scene and related chromaticity distribution on the rb normalized plane

(b) The same scene with few more colored objects and the related chromaticity distribution on the rb normalized color plane

Figure 5.4 : Chromaticity distribution of a scene with a single dominant color and the effects of the addition of few colored objects on the chromaticity distribution.

blackbody radiator. The equation of the spectral radiant power of the blackbody radiators [29], as a function of temperature T (Kelvin) is given by the following:

$$M(\lambda) = \frac{c_1 \lambda^{-5}}{e^{\frac{c_2}{\lambda T}} - 1} \quad (5.8)$$

where:

$$c_1 = 3.7418 \times 10^{-16} \frac{W}{m^2} \quad (5.9)$$

$$c_2 = 1.4388 \times 10^{-2} \frac{W}{K} \quad (5.10)$$

and λ is the wavelength in m .

For an arbitrary light source, the correlated color temperature is defined as the color temperature of the blackbody radiator that is closest to the illuminant. Common light sources, like incandescent light and daylight have a correlated color temperature of 2900K and 6500K respectively. Fig.(5.5) describes the white point distribution of different blackbody radiator light sources on the CIE (Commission International de l'Eclairage) [26] xy chromaticity diagram. Differences in color temperature do not correspond to equal perceptual color differences. In particular, color differences at low color temperatures generate a greater perceived difference than the same color temperature difference at higher color temperatures. Some empirical metrics have also been developed to deal with this perceptive non uniformity [30, 31].

5.3.4 Color Errors

The color space XYZ is not perceptually uniform. This means that at equal distances in the XYZ space do not correspond equal perceived distances. The XYZ space can be transformed to a more nearly uniform CIE 1976 $L^*a^*b^*$ (CIELab) color space. Here are reported the equations to transform the XYZ coordinates to the $L^*a^*b^*$ space:

$$L^* = 116 \left[f \left(\frac{Y}{Y_n} \right) - \frac{16}{116} \right] \quad (5.11)$$

$$a^* = 500 \left[f \left(\frac{X}{X_n} \right) - \left(\frac{Y}{Y_n} \right) \right] \quad (5.12)$$

$$b^* = 200 \left[f \left(\frac{Y}{Y_n} \right) - \left(\frac{Z}{Z_n} \right) \right] \quad (5.13)$$

$$f(x) = \begin{cases} x^{\frac{1}{3}} & \text{if } x > 0.008856 \\ 7.787x + \frac{16}{116} & \text{otherwise} \end{cases} \quad (5.14)$$

$$C_{ab}^* = \sqrt{a^{*2} + b^{*2}} \quad (5.15)$$

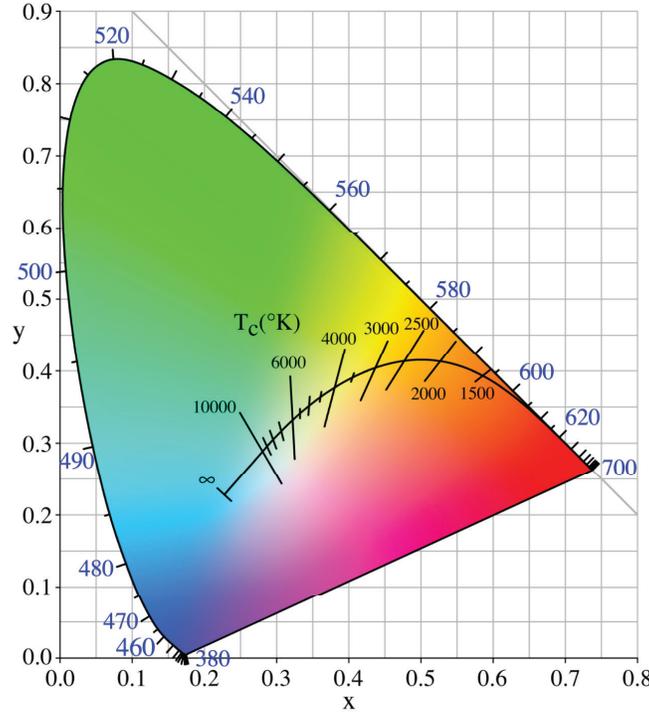


Figure 5.5 : White point position of different light sources on the CIE chromaticity diagram.

$$h_{ab}^* = \tan^{-1} \left(\frac{b^*}{a^*} \right) \quad (5.16)$$

where X_n , Y_n and Z_n are the tristimulus values of the reference white, L^* denotes the lightness, a^* and b^* denote chromaticity, C_{ab}^* denotes chroma, and h_{ab}^* denotes hue. The CIELab equations were derived in such a way that the illuminant is always at $L^* = 100$, $a^* = 0$, $b^* = 0$, thus defining the illuminant as the reference white. The Euclidean color difference ΔE_{ab} in the $L^*a^*b^*$ color space of the L_1^*, a_1^*, b_1^* point from a reference L_0^*, a_0^*, b_0^* point can be then calculated as:

$$\Delta E_{ab} = \sqrt{(\Delta L)^2 + (\Delta a)^2 + (\Delta b)^2} \quad (5.17)$$

where:

$$\Delta L = L_1^* - L_0^* \quad (5.18)$$

$$\Delta a = a_1^* - a_0^* \quad (5.19)$$

$$\Delta b = b_1^* - b_0^* \quad (5.20)$$

In 1994 the CIE introduced a new formula to calculate color difference aimed to be more perceptually uniform than the ΔE_{ab} metric [32]. This color difference, known as ΔE_{94} is widely used and reported below:

$$\Delta E_{94} = \sqrt{\left(\frac{\Delta L^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C^*}{k_C S_C}\right)^2 + \left(\frac{\Delta H^*}{k_H S_H}\right)^2} \quad (5.21)$$

where:

$$\Delta H^* = \sqrt{(\Delta a^*)^2 + (\Delta b^*)^2 + (\Delta C^*)^2} \quad (5.22)$$

$$S_L = 1 \quad (5.23)$$

$$S_C = 1 + K_1 C_1 \quad (5.24)$$

$$S_H = 1 + K_2 C_2 \quad (5.25)$$

and $k_L = k_C = k_H = 1$ for reference conditions; $K_1 = 0.045$ and $K_2 = 0.015$. Other metrics, aimed to compare whole images rather than single colors, have been proposed and developed [32–34]. They take into account also spatial aspects of the color perception according to human visual system features.

5.3.5 Automatic White Balancing Error

Most of the automatic white balancing estimation techniques rely on a statistic analysis of the image features. Hence it is quite important evaluating how the different algorithms are sensitive to some specific combinations of ambient illuminants and image features. It is possible using lab techniques to evaluate the estimation performance of white balancing algorithms. Light booths and color checkers can be used to do it, but scene composition in this case is quite different from typical field test images, thus leading to a difficult evaluation of the reliability of the automatic estimation. Authors in [35] describe a method aiming to evaluate the estimation error of white balancing algorithms by using neutral reference present on common scenes to retrieve the target white balancing correction parameters. In this Section we describe a method for a comparative analysis of different approaches to white balancing estimation on field test images, aimed to evaluate also the effects of the automatic white balancing estimation on the final color rendition of the image. The method consists of a data set of calibrated raw images and a framework aimed to perform a quantitative evaluation of the different automatic white balancing estimation techniques. A neutral reflectance target, inserted on the scene near the subject, allows to read the color of the light and the related target correction parameters. This provides for the entire data set the color channel gains that should be applied for a perfect illuminant compensation. The set of calibrated images is then processed by the other algorithms of the image processing pipeline and the images coming out from the estimation process are then compared with the ones coming from the target channel gains application used as

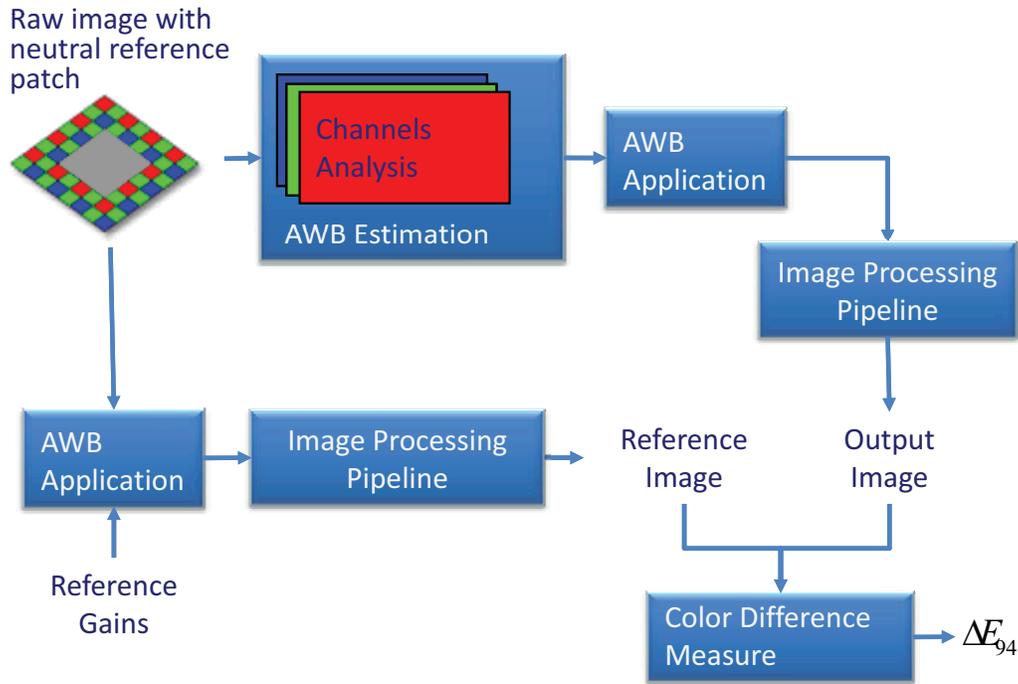


Figure 5.6 : Block scheme of an auto white balancing evaluation framework.

reference, thus providing a color error measure. Fig.(5.6) report the block scheme of the described framework.

Fig.(5.7) reports a section of a test report obtained by the evaluation framework. On first column are reported the colored and brightness adjusted versions of the input raw images, on the second column are reported the target output reference images, whereas on the remaining columns the output images, coming out from six different auto white balancing estimation algorithms, are reported. It is widely accepted that the perfect estimation algorithm for all situation does not exist, hence a potential improvement to the estimation confidence can be achieved by combining multiple estimation techniques according either to specific algorithm or scene features [36,37].

5.4 Color Correction

The second stage of the color processing pipeline, as described by (5.2), transforms the image data into a standard RGB (e.g., ITU-R BT.709) color space. This transformation, commonly referred as color matrixing, is required because the spectral sensitivity of the sensor color channels rarely match those of the desired output color space. Typically on three channels sensors this transform is carried out by a 3-by-3 matrix with 9 variables to be optimally determined either by algebraic [38] or optimization-based methods [39]. In

	Original	Target Balance	Algorithm 1	Algorithm 2	Algorithm 3	Algorithm 4	Algorithm 5	Algorithm 6
dsc_0254	 R 199 G 212 B 227 Ref. Patch Levels	 R 1.14 G 1.07 B 1.00 Calibrated Gains	 R 1.66 G 1.29 B 1.00 Error: 0.065	 R 1.12 G 1.06 B 1.00 Error: 0.004	 R 1.71 G 1.31 B 1.00 Error: 0.070	 R 1.72 G 1.31 B 1.00 Error: 0.071	 R 1.59 G 1.26 B 1.00 Error: 0.057	 R 1.45 G 1.20 B 1.00 Error: 0.041
dsc_0291	 R 133 G 143 B 148 Ref. Patch Levels	 R 1.11 G 1.03 B 1.00 Calibrated Gains	 R 1.47 G 1.18 B 1.00 Error: 0.049	 R 1.03 G 1.02 B 1.00 Error: 0.016	 R 1.56 G 1.23 B 1.00 Error: 0.057	 R 1.68 G 1.28 B 1.00 Error: 0.069	 R 1.39 G 1.14 B 1.00 Error: 0.040	 R 1.23 G 1.05 B 1.00 Error: 0.022
dsc_0291_c1	 R 133 G 143 B 148 Ref. Patch Levels	 R 1.11 G 1.03 B 1.00 Calibrated Gains	 R 1.40 G 1.13 B 1.00 Error: 0.043	 R 1.00 G 1.00 B 1.04 Error: 0.023	 R 1.45 G 1.16 B 1.00 Error: 0.048	 R 1.59 G 1.24 B 1.00 Error: 0.061	 R 1.31 G 1.08 B 1.00 Error: 0.034	 R 1.21 G 1.02 B 1.00 Error: 0.024
dsc_0291_c2	 R 133 G 143 B 148 Ref. Patch Levels	 R 1.11 G 1.03 B 1.00 Calibrated Gains	 R 1.09 G 1.00 B 1.01 Error: 0.007	 R 1.05 G 1.03 B 1.00 Error: 0.013	 R 1.18 G 1.05 B 1.00 Error: 0.011	 R 1.27 G 1.10 B 1.00 Error: 0.023	 R 1.09 G 1.00 B 1.01 Error: 0.006	 R 1.09 G 1.00 B 1.04 Error: 0.011
dsc_0291_c3	 R 133 G 143 B 148 Ref. Patch Levels	 R 1.11 G 1.03 B 1.00 Calibrated Gains	 R 1.01 G 1.00 B 1.14 Error: 0.033	 R 1.01 G 1.00 B 1.00 Error: 0.018	 R 1.01 G 1.00 B 1.08 Error: 0.027	 R 1.00 G 1.02 B 1.06 Error: 0.027	 R 1.02 G 1.00 B 1.12 Error: 0.030	 R 1.01 G 1.00 B 1.15 Error: 0.036
dsc_0292	 R 133 G 143 B 148 Ref. Patch Levels	 R 1.11 G 1.03 B 1.00 Calibrated Gains	 R 1.55 G 1.23 B 1.00 Error: 0.060	 R 1.03 G 1.02 B 1.00 Error: 0.016	 R 1.65 G 1.27 B 1.00 Error: 0.068	 R 1.76 G 1.33 B 1.00 Error: 0.078	 R 1.50 G 1.19 B 1.00 Error: 0.052	 R 1.28 G 1.07 B 1.00 Error: 0.029

Figure 5.7 : A test report for the evaluation of different white balancing estimation techniques.

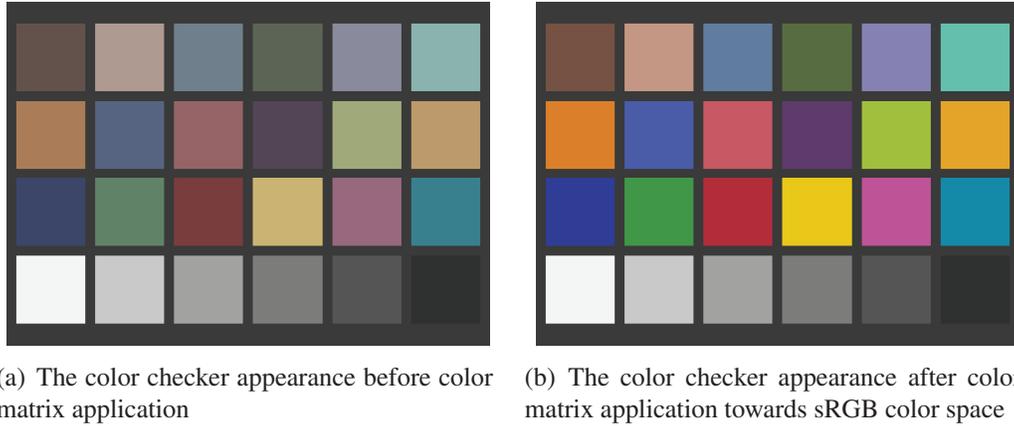


Figure 5.8 : A color checker as it appears before and after the color matrix application process.

the following a compact version of (5.2) is reported:

$$\mathbf{RGB}_{out} = (\mathbf{A} \cdot \mathbf{I}_w \cdot \mathbf{RGB}_{in})^\gamma \quad (5.26)$$

where \mathbf{I}_w and \mathbf{A} represent the diagonal matrix for the illuminant compensation and the color matrixing transformation respectively. Given a set of n different patches whose sRGB target values \mathbf{r} are known, and the corresponding camera raw values \mathbf{c} measured by the sensor when the patches are lit by a particular illuminant, what is usually done is to find the matrix \mathbf{M} that satisfies:

$$\mathbf{M} = \arg \left(\min_{\mathbf{A} \in \mathbb{R}^{3 \times 3}} \sum_{k=1}^n \mathcal{E}(\mathbf{r}_k, (\mathbf{A} \mathbf{I}_w \mathbf{c}_k)^\gamma) \right) \quad (5.27)$$

where \mathcal{E} is the chosen error metric, and the subscript k indicates the k^{th} patch. Usually the error metric \mathcal{E} is performed by the computation of the average ΔE_{94} colorimetric error between the reference and calculated sRGB values mapped in the CIELab color space. The coefficients of the diagonal illuminant compensation \mathbf{I}_w are previously computed in order to perfectly compensate for the chosen illuminant. In order to avoid that the color transform affect the compensation for the source light, some constraints are introduced. This is usually carried out by constraining the coefficients of the matrix \mathbf{M} to generate an illuminant preserving transform. Typically the nine degrees of freedom of the matrixing transformation are then reduced to six, in order to have a white point preserving transformation, (i.e., a neutral color in the device dependent color space is mapped to a neutral color in the device independent color space). This can be easily obtained constraining the coefficients of each row of \mathbf{M} to sum to one. Fig.(5.8) reports a color checker as it appears before color matrix application Fig.(5.8(a)) and after the color matrix transform from color filter array response to sRGB color space Fig.(5.8(b)). The color matrix applied has been derived through a minimization process as described by (5.27).

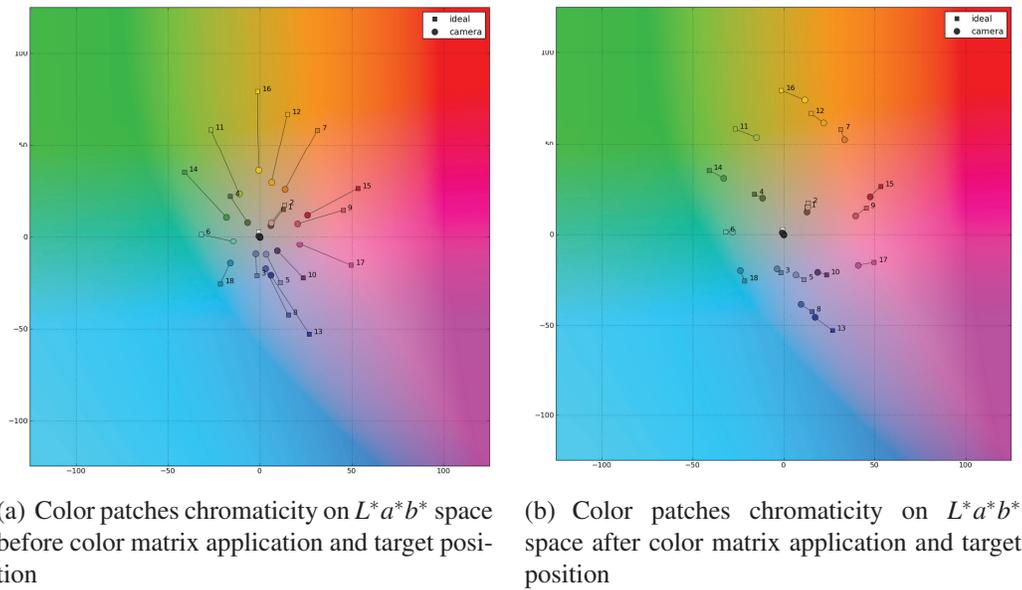


Figure 5.9 : Representation of the camera chromatic response for a color checker on $L^*a^*b^*$ space before and after color matrix application.

A useful representation of the sensor chromatic response can be performed on the $L^*a^*b^*$. This can give information about the colorimetric error performed on a single color patch by the application of the color matrix resulting from the minimization process. Fig.(5.9) reports the distribution of the color patches a^* and b^* components. These are reported before and after the color matrix application, together with the target chromaticities position that they should have through a perfect color matrix transform toward sRGB color space.

5.4.1 Constrained Color Correction

Usually the color matrix transform is optimized for a single scene condition and for all the color patches of the reference color checker at the same time. According to wide diffused user modes on digital imaging devices it is also possible to constrain the color matrix characterization to produce better colorimetric results in some specific situations. A possible solution can be obtained by performing different minimizations according to some typical scene preset modes, (i.e., landscape, portrait), thus leading to different color matrices for different scene modes. By using a not uniform weighting function for the reference patches rather than giving to all the patches the same importance, it is possible to generate color matrices with a less colorimetric error on specific patches. For example a portrait preset requires to give more importance to the skin tone patches, whereas for landscape situations the vegetation and sky tone patches have more importance than other patches. Equation (5.27) can be then easily modified to take into account a weighting

function for the different color patches:

$$\mathbf{M} = \arg \left(\min_{\mathbf{A} \in \mathbb{R}^{3 \times 3}} \sum_{k=1}^n \omega_k \mathcal{E}(\mathbf{r}_k, (\mathbf{A}\mathbf{I}_w \mathbf{c}_k)^\gamma) \right) \quad (5.28)$$

where ω_k indicates the relative weight of the k_{th} color reference patch.

Another possibility is also to constraint the color matrix coefficients in order to take into account also different exposure or illuminant situations. As mentioned above, usually the color matrix coefficients are retrieved starting from a single ambient light situation, which is commonly chosen according to statistical analysis of photo scene type distribution. If multiple data sets are collected under different light sources, the minimization process can be also extended to take into account different situations. The light sources distribution can be either uniform or related to a-priori probabilistic distribution analysis, usually coming out from a statistical analysis of the most common scenes grabbed on consumer devices. The (5.28), in the case of m different data set, can be further extended as follows:

$$\mathbf{M} = \arg \left(\min_{\mathbf{A} \in \mathbb{R}^{3 \times 3}} \sum_{j=1}^m \alpha_j \left(\sum_{k=1}^n \omega_k \mathcal{E}(\mathbf{r}_k, (\mathbf{A}\mathbf{I}_{w_j} \mathbf{c}_{k_j})^\gamma) \right) \right) \quad (5.29)$$

Other ambient light situations can be also taken into account in the minimization process in order to derive an overall optimal color matrix, but this could lead to sub optimal color accuracy under common situations. Adaptive solutions with multiple sensor chromatic response characterization and the use of different color matrices in different situations can also rely on the estimation algorithms of the image pipeline, such as auto-focus, auto-exposure and auto white-balancing, which can produce useful metadata for the choice of the optimal color correction to be applied.

5.4.2 Trade Offs on Color Correction

The color filter array spectral distribution rarely matches those of the color space where the final image will be represented. As previously reported, this requires always the color matrix correction operation. In Section 5.4 we described how this color matrix can be derived through a sensor chromatic characterization. The color matrix coefficients are strictly related to the spectral transmittance of the color filter arrays, the way the sensor responds to the different wavelengths of the incident light. If the filters color spectral responses are very wide, then many subtle difference of color can be represented, thus producing a wide gamut color response (i.e., the number of different color that the device is capable to represent). Moreover, if the color filters have a wide spectral response, the greater channels spectral correlation requires a stronger color transform to correct the mismatch between the sensor primaries and the sRGB primaries, producing color matrices with bigger coefficients. This can lead to a consistent signal degradation as will be described later. On the contrary if the sensor responses are not wide, then less color spectral correlation leads to a color matrix with smaller coefficient, since the mismatch

between sensor primaries and sRGB primaries can be easily corrected. Narrow spectral responses though produce smaller channels response levels with lower signal to noise ratio. Therefore, many trade-offs must to be considered in the sensor design process to deal with color accuracy, color gamut extension, and color processing signal to noise ratio degradation [40–42]. In order to explicit the signal degradation of the color matrix application let consider the 5.2. For a neutral image let consider the red channel output:

$$R_o = a_{11}R_i + a_{12}G_i + a_{13}B_i \quad (5.30)$$

Each channel has a certain amount of noise, then we can calculate the variance of the output channels as follows:

$$\sigma_{R_o}^2 = a_{11}^2 \sigma_{R_i}^2 + a_{12}^2 \sigma_{G_i}^2 + a_{13}^2 \sigma_{B_i}^2 \quad (5.31)$$

If we assume that the noise is equal for all channels, then:

$$\sigma_{R_i} = \sigma_{G_i} = \sigma_{B_i} = \sigma_i \quad (5.32)$$

and:

$$\sigma_{R_o}^2 = (a_{11}^2 + a_{12}^2 + a_{13}^2) \sigma_i^2 \quad (5.33)$$

then, the ratio between the output and input standard deviation of the red channel can be calculated

$$\frac{\sigma_{R_o}}{\sigma_{R_i}} = \sqrt{(a_{11}^2 + a_{12}^2 + a_{13}^2)} \quad (5.34)$$

Similar considerations can be done for all other channels. If we consider the neutral preserving constraint applied to the rows of the color matrix, this ratio is greater than one, thus producing a signal to noise ratio degradation after the color matrix application. Therefore, the ideal color matrix, which does not produce noise amplification, should be the 3 by 3 identity matrix and it is possible just when a perfect match between sensor primaries and sRGB primaries is obtained. The trade-off between color signal degradation, color gamut extension other than technological constraints drive the color filters response design.

5.5 Color Rendition Enhancements

Due to physical and computational constraints often on consumer embedded devices, the color reproduction is a trade-off between color accuracy and the devices capabilities [43]. The large diffusion on consumer market of digital imaging devices, suggests solutions aimed to enhance the color appearance especially for such chromatic classes which have the most perceptive impact on the human visual system [44–46]. In this Section we describe a possible application for embedded devices aimed to automatically improves such chromatic classes, in particular skin, vegetation, and sky tones in a selective way

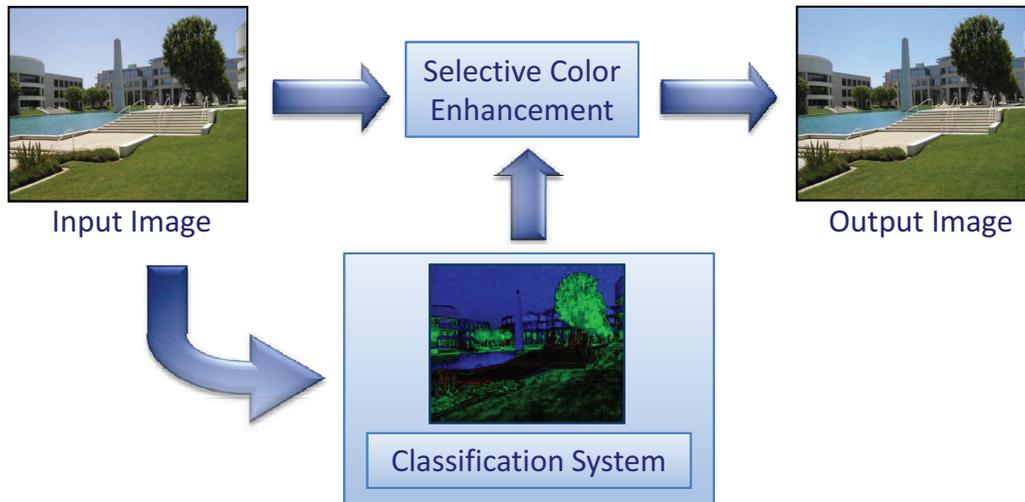


Figure 5.10 : Block scheme of the selective color enhancement process.

[47,48]. This solution improves the color appearance for the mentioned chromatic classes, and has been designed to be performed in a cost effective way on embedded devices, by an automatic chromatic classifier driven by a statistical characterization of a large image database of natural scenes, followed by an adaptive color enhancement process. In Fig.(5.10) a scheme of the application is reported.

The algorithm recognizes image areas (e.g., skin, vegetation, sky) and corrects them towards a set of programmable chromatic targets. It is mainly a post processing solution and it can be either implemented in a full image processing pipeline architecture or as a stand-alone algorithm. Generally, output devices are calibrated in order to reproduce with a controlled error most colors without any consideration to the preferred chromatic classes. This approach tries to minimize the amount of color reproduction errors on a limited set of preferred classes without affecting the color accuracy of other chromatic classes, assuming that the human visual system does not perceive in the same way different chromatic objects, thus focusing on which have the most perceptive impact. A large image database of high quality natural sRGB images has been collected in order to extract and characterize chromatic information related to the classes under investigation. All the images were chosen according to a perceived naturalness principle. Images affected by severe color cast and/or anomalous color distortions were not considered. An automatic segmentation algorithm [49], aimed to extract homogeneous chromatic regions, was also used to avoid collecting statistics on excessively scattered color samples. Fig.(5.11(a)) reports the chromaticity of different classes where each point represents not just a pixel but a color segment belonging to each chromatic class. The chromaticity coordinates are represented by the normalized red and green channels. For a pixel of RGB coordinates, the red and green chromaticity coordinates are calculated as follows:

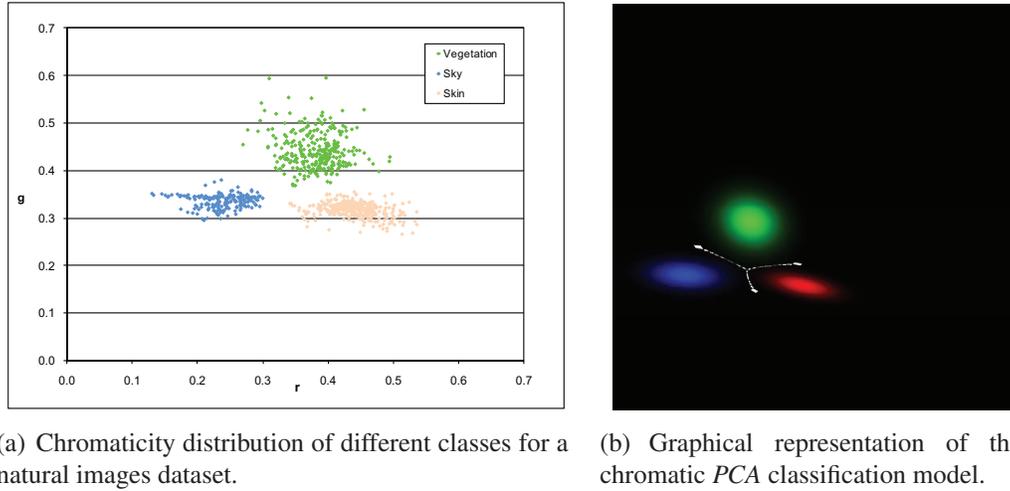


Figure 5.11 : Distribution and model of natural objects chromatic classes.

$$r = \frac{R}{R + G + B} \quad (5.35)$$

$$g = \frac{G}{R + G + B}$$

Starting from the collection, a statistical model has been obtained through a principal component analysis of each chromatic class. This model drives the automatic pixel classification process by retrieving the membership class and the probability level of each pixel. A global distance to the chromatic targets is then calculated for each chromatic cluster. A graphical representation of the chromatic classification model is reported in Fig.(5.11(b)).

This model can be either implemented as a functional model through a two dimensional Gaussian function or by a look up table which entries are the chromaticity coordinates, whereas the content is the belonging class and the degree of the classification estimation process. Formula (5.36) implements a two dimensional Gaussian function of a single chromatic class, where parameters come from the principal component analysis step of data characterization:

$$G_c(r, g) = e^{\left\{ \frac{[-(r-r_t)\cos\alpha - (g-g_t)\sin\alpha]^2}{2\sigma_r^2} \right\}} \left\{ \frac{[-(r-r_t)\sin\alpha + (g-g_t)\cos\alpha]^2}{2\sigma_g^2} \right\} \quad (5.36)$$

The enhancement process takes into account both the classification results and the programmable targets position in order to move each chromatic cluster toward the expected position. For a pixel classified as belonging to chromatic class c , an intensity preserving chromatic shift is carried out, according to:

$$\begin{aligned}
R' &= 3I(r + \Delta r_c) \\
G' &= 3I(g + \Delta g_c) \\
B' &= 3I[1 - (r + \Delta r_c) - (g + \Delta g_c)]
\end{aligned} \tag{5.37}$$

where Δr_c and Δg_c represent the distance of the chromatic cluster c to its selected chromatic targets, and I is the average intensity of the input RGB pixel. The RGB output coordinates are then derived by a linear combination of the input values and the target levels:

$$\begin{aligned}
R_o &= w_c R' + (1 - w_c)R \\
G_o &= w_c G' + (1 - w_c)G \\
B_o &= w_c B' + (1 - w_c)B
\end{aligned} \tag{5.38}$$

where, w_c is a correction weight resulting from the classification process. In Fig.(5.12) two examples are reported with input images, their relative chromatic classification masks, and output images after processing. The images reported in the central column of the figure come out from the pixel based chromatic classification (red for skin tones, green for vegetation, blue for sky). Since this classification drives the color enhancement process, not classified pixels (black pixels) will be not affected by the processing, thus leading to a selective enhancement of specific chromatic classes, which could be not performed by a typical global saturation-hue enhancement.

5.6 Conclusions

This Chapter has surveyed the main color processing techniques across the image generation pipeline. A simplified model has been proposed in order to highlight the main problems related to the ambient light source estimation and sensor chromatic response characterization and correction. We have provided an overview of most common approaches to light sources estimation pointing out also to most recent research fields about the estimation of multiple light sources present on the scene. We have described also different strategies that can be followed on embedded devices in order to characterize and correct the chromatic sensor response, describing also a potential adaptive color enhancement technique for most common natural scene images. The computational resources limitation, usually present on imaging embedded devices, heavily limits the complexity of the color processing algorithms. The development of more sophisticated color processing models for both complex image scenes and the human visual system features, is though a promising challenge for color imaging applications of the next future devices, when greater computational resource will be also common on embedded imaging systems.

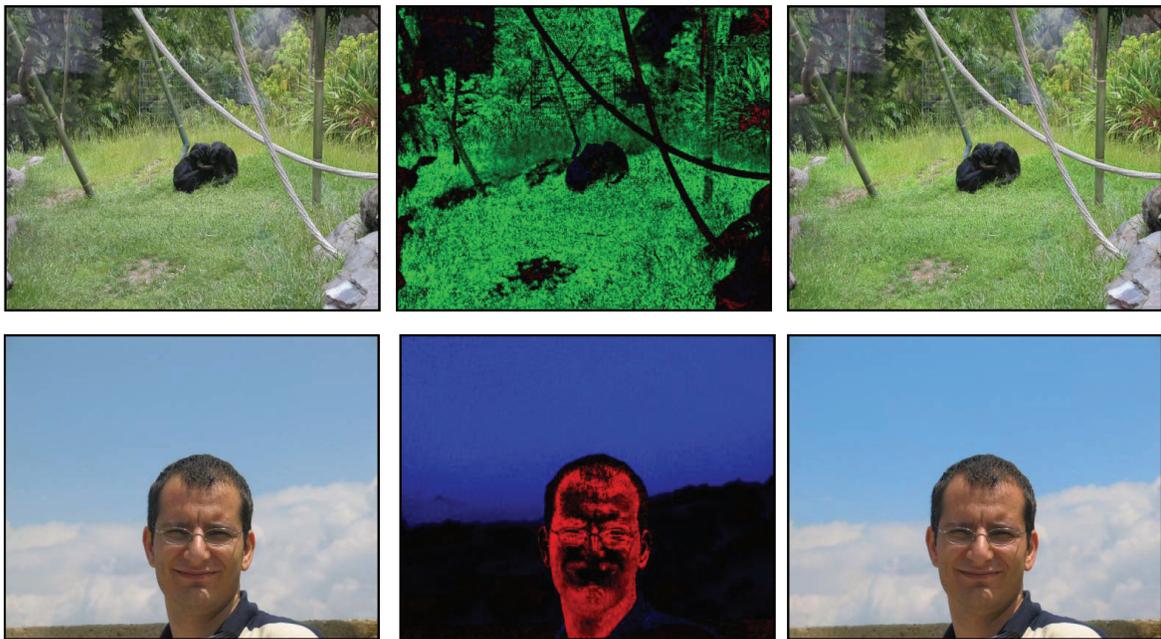


Figure 5.12 : Two natural scene images (on the left), their chromatic classification map (on the center) and the output color enhanced images (on the right).

Bibliography

- [1] S. D. Hordely, "Scene illuminant estimation: Past, present, and future," *Color Research & Application*, vol. 31, no. 4, pp. 303–314, 2006.
- [2] K. Barnard, V. Cardei, and B. Funt, "A comparison of computational color constancy algorithms—part i: Methodology and experiments with synthesized data," *IEEE Transaction on Image Processing*, vol. 11, pp. 972–984, Sept 2002.
- [3] K. Barnard, L. Martin, A. Coath, and B. Funt, "A comparison of computational color constancy algorithms—part ii: Experiments with image data," *IEEE Transaction on Image Processing*, vol. 11, pp. 985–996, Sept 2002.
- [4] M. D. Fairchild, "Color appearance models," *Addison Wesley*, 1997.
- [5] J. von Kries, "Chromatic adaptation - originally published in festschrift der albrecht-ludwigs-universitat (1902)," *D.L. Ed. Sources of Color Vision*, MIT Press, Cambridge, 1970.
- [6] J. A. Worthey and M. H. Brill, "Heuristic analysis of von kries color constancy," *Journal of the Optical Society of America*, vol. 3, pp. 1708–1712, 1986.
- [7] E. H. Land and J. McCann, "Lightness and retinex theory," *Journal of the Optical Society of America*, vol. 61, no. 1, pp. 1–11, 1971.
- [8] R. Gonzalez and R. Woods, *Digital Image Processing 2nd Edition*. Prentice Hall, 2002.
- [9] K. Barnard, "Practical colour constancy," *PhD thesis, Simon Fraser Univ., School of Computing Science*, 2000.
- [10] Y. Yang and A. Yuille, "Sources from shading," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR91)*, pp. 534–539, 1991.
- [11] S. Tominaga and B. A. Wandell, "Standard surface reflectance model and illuminant estimation," *Journal of the Optical Society of America*, vol. A, no. 6, pp. 576–584, 1989.
- [12] G. Buchsbaum, "A spatial processor model for object colour perception," *J. Franklin Inst.*, vol. 310, pp. 1–26, 1980.
- [13] T. J. Cooper, "A novel approach to color cast detection and removal in digital images," *Proceedings of SPIE*, vol. 3963, pp. 167–175, 1999.
- [14] T. J. Cooper, "Color segmentation as an aid to white balancing for digital still cameras," *Proceedings of SPIE*, vol. 4300, pp. 164–171, 2000.
- [15] F. Gasparini and R. Schettini, "Color balancing of digital photos using simple image statistics," *Pattern Recognition*, vol. 37, pp. 1201–1217, 2004.

- [16] G. D. Finlayson, S. D. Hordley, and P. M. Hubel, "Color by correlation: A simple, unifying framework for color constancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 1209–1221, November 2001.
- [17] D. Forsyth, "A novel algorithm for color constancy," *International Journal of Computer Vision*, vol. 5, pp. 5–36, 1990.
- [18] M. Vrhel, R. Gershon, and L. Iwan, "Measurement and analysis of object reflectance spectra," *Color Research and Application*, vol. 19, no. 1, pp. 4–9, 1994.
- [19] S. Tominaga, S. Ebisui, and B. Wandell, "Scene illuminant classification - brighter is better," *Journal of the Optical Society of America*, vol. 18, pp. 55–64, 2001.
- [20] K. Barnard and B. Funt, "Experiments in sensor sharpening for color constancy," *Proceedings of IS&T sixth Color Imaging Conference: Color Science, Systems and Applications*, pp. 43–46, 1998.
- [21] G. Finlayson, M. Drew, and B. Funt, "Spectral sharpening: sensor transformations for improved color constancy," *Journal of the Optical Society of America*, vol. 11, pp. 1553–1563, 1994.
- [22] F. Xiao, J. Farrell, J. D. Carlo, and B. Wandell, "Preferred color spaces for white balancing," *Proceedings of SPIE-IS&T Electronic Imaging*, vol. 5017, pp. 342–350, 2003.
- [23] J. Tajima, "Illumination chromaticity estimation based on dichromatic reflection model and imperfect segmentation," *Proceedings of IAPR Computational Color Imaging Workshop (CCIW09)*, vol. 5646, pp. 51–61, March 2009.
- [24] J. Toro, "Dichromatic illumination estimation without pre-segmentation," *Pattern Recognition Letters*, vol. 29, pp. 871–877, May 2008.
- [25] M. Ebner and C. Herrmann, "On determining the color of the illuminant using the dichromatic reflection model," *Pattern Recognition, Lecture Notes on Computer Science*, vol. 3663, pp. 1–8, 2005.
- [26] "Cie," 1931.
- [27] G. Finlayson, S. Hordley, M. Drew, and C. Lu, "On the removal of shadows on images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 59–68, 2006.
- [28] E. Hsu, T. Mertens, S. Paris, S. Avidan, and F. Durand, "Light mixture estimation for spatially varying white balance," *Proceedings of ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH08)*, pp. 1–7, 2008.
- [29] G. Wyszecki and W. S. Stiles, "Color science: Concepts and methods, quantitative data and formulae," *Wiley, New York*, 2000.

- [30] D. B. Judd, "Sensibility to color-temperature change as a function of temperature," *Journal of the Optical Society of America*, vol. 23, pp. 127–134, 1933.
- [31] S. Tominaga and B. A. Wandell, "Natural scene-illuminant estimation using the sensor correlation," *Proceedings of the IEEE*, vol. 90, pp. 42–56, January 2002.
- [32] G. Sharma, W. Wu, and E. N. Dalal, "The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations," *Color Research & Applications*, vol. 30, no. 1, pp. 21–30, 2005.
- [33] G. Johnson and M. Fairchild, "A top down description of S-CIELAB and CIEDE2000," *Color Research and Application*, vol. 28, pp. 425–435, 2009.
- [34] X. Zhang and B. Wandell, "A spatial extension of CIELAB for digital color-image reproduction," *Journal of the Society for Information Display*, vol. 5, no. 1, pp. 61–63, 1997.
- [35] F. Ciurea and B. Funt, "A large image database for color constancy research," *Proceedings of IS&T Eleventh Color Imaging Conference*, pp. 160–164, 2003.
- [36] A. Gijsenij and T. Gevers, "Color constancy using natural image statistics," *Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR07)*, pp. 1–8, 2007.
- [37] S. Bianco, F. Gasparini, and R. Schettini, "A consensus based framework for illuminant chromaticity estimation," *Journal of Electronic Imaging*, vol. 17, no. 2, pp. 1–9, 2008.
- [38] P. M. Hubel, J. Holm, G. D. Finlayson, and M. S. Drew, "Matrix calculations for digital photography," in *Proceedings of the IS&T/SID Fifth Color Imaging Conference*, pp. 105–111, 1997.
- [39] S. Bianco, F. Gasparini, A. Russo, and R. Schettini, "A new method for RGB to XYZ transformation based on pattern search optimization," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 3, pp. 1020–1028, 2007.
- [40] J. Buzzi, F. Guichard, and H. Hornung, "From spectral sensitivities to noise characteristics," *Proceedings of SPIE-IS&T Electronic Imaging*, vol. 6502, 2007.
- [41] S. Quan, "Evaluation and optimal design of spectral sensitivities for digital imaging," *PhD thesis - Rochester Institute of Technology*, 2002.
- [42] G. Sharma, "Digital color imaging handbook," *CRC Press*, 2003.
- [43] U. Barnhfer, J. M. D. Carlo, B. Olding, and B. A. Wandell, "Color estimation error trade-offs," *Proceedings of SPIE-IS&T Electronic Imaging*, vol. 5017, 2003.

- [44] S. Yendrikhovskij, F. Blommaert, and H. de Ridder, "Optimizing color reproduction of natural images," *Proceedings of Sixth Color Imaging Conference: Color Science, Systems, and Applications*, pp. 140–145, November 1998.
- [45] E. Lee and Y. Ha, "Favorite color correction for favorite colors," *IEEE Transaction on Consumer Electronics*, vol. 44, pp. 10–15, Feb. 1998.
- [46] D.-H. Kim, H.-C. Do, and S.-I. Chien, "Preferred skin color reproduction based on adaptive affine transform," *IEEE Transactions on Consumer Electronics*, vol. 51, pp. 191–197, Feb. 2005.
- [47] F. Naccari, S. Battiato, A. Bruna, A. Capra, and A. Castorina, "Natural scenes classification for color enhancement," *IEEE Transactions on Consumer Electronics*, vol. 51, pp. 234–239, Feb. 2005.
- [48] F. Naccari, A. Bruna, A. Castorina, and S. Curti, "Color refinement for natural scene images," *Proceedings of IEEE International Conference on Consumer Electronics (ICCE07)*, Jan. 2007.
- [49] D. Comaniciu and D. Meer, "Robust analysis of feature spaces: Color image segmentation," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR97)*, pp. 750–755, 1997.

Noise Reduction

A. Bosco

Advanced System Technology - Catania Lab, STMicroelectronics, Italy.

R. Rizzo

Image Processing Lab, University of Catania, Italy.

Abstract: Among the many factors contributing to image quality degradation, noise is one of the most recurrent and difficult elements to deal with. Smart filters capable to remove noise without affecting the tiny details of a digital image are of primary importance to produce pleasant pictures. Different noise sources, having different characteristics, are superimposed to the image signal; consequently, the design of effective filters capable to discriminate and remove unwanted signal from useful data, requires analysis and understanding of the whole image formation process. This Chapter is devoted to the analysis of the main noise sources that contaminate the ideal image signal, providing an overview of noise estimation and filtering techniques.

6.1 Introduction

Noise signal has a negative impact on the image processing pipeline as a whole and lowers the perceived quality of the acquired data. Hence, a smart filter enabling successful noise reduction without affecting the tiny details of an image is of paramount importance in an imaging system. Camera phones and low-end digital still cameras are particularly subject to noise degradation, especially when images are acquired in low light. In case of low light environments, such as indoor scenes, the image signal must be amplified so as to obtain an acceptable picture. The possible absence of a flashgun, typically in camera phones, further worsens the shooting conditions. Unfortunately, when boosting an already degraded signal, noise is also amplified (see Chapter 3). Noise reduction must be adaptive with respect to the noise and signal levels, such that the filter strength can be modulated according to the amount of noise and texture information present in the image. Noise contaminating raw images is usually assumed additive white and Gaussian (AWGN), because of the simple mathematical tractability of the Gaussian distribution (6.4.4). Noise intensity is provided by the standard deviation σ of the underlying distribution [1]; many filters rely on σ to adaptively change their smoothing effects (see Section 6.9.3).

The noise levels change depending on the particular image pipeline stage in which they are measured; in fact post-processing algorithms introduce correlation in data and noise, amplifying or reducing the noise intensities.

Fig.(6.1) shows a typical image processing pipeline and the most important noise sources contaminating images; the colors of the blocks denote the capability of each algorithm to increase (red) or reduce (blue) the noise amount of the image.

This Chapter provides an excursus on the principal noise sources and analyzes in detail the most important techniques of noise estimation and filtering.

The Chapter is organized as follows. In Section 6.2 some noise metrics are reported. Section 6.3 describes the intrinsic noise generated by the light itself (Photon Shot Noise).

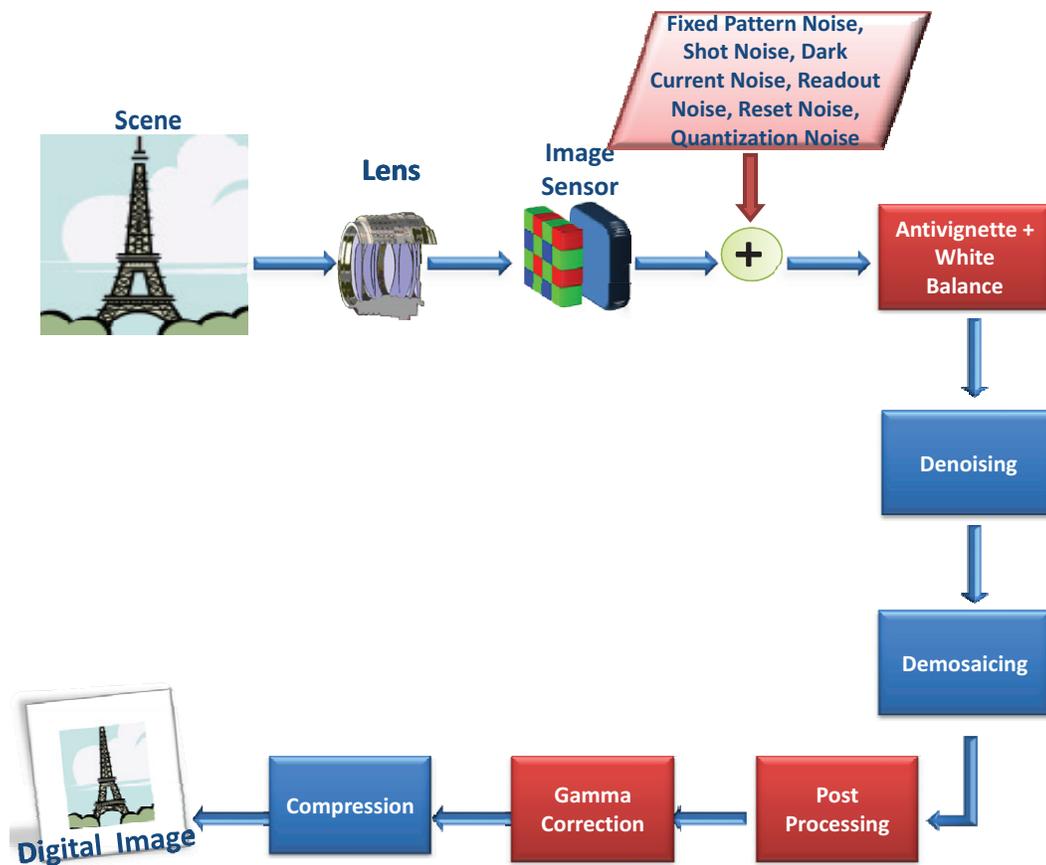


Figure 6.1 : Image processing pipeline and noise sources. Pipeline stages in red indicate the algorithms contributing to increase image noise. Blue stages represent the algorithms that cause a reduction of noise levels.

Section 6.4 defines image defects and the Gaussian noise model introducing the main noise sources at sensor level. Section 6.5 illustrates the image pipeline stages in which noise has a key impact on image quality, while Sections 6.6 and 6.7 describe luminance and chrominance noise respectively. Sections 6.8 and 6.9 report the main techniques used to estimate and filter image noise. Other references on image noise perception can be found in Chapter 12 and in [1–3].

6.2 Noise metrics

Two important measures dealing with noise are *SNR* (Signal to Noise Ratio) and *Dynamic Range*, useful to numerically define image quality and sensor acquired luminance range respectively.

6.2.1 Signal to Noise Ratio

SNR is usually adopted as a simple reference measure of image quality. It is computed as the ratio between the signal and the underlying noise and is expressed in decibel [1].

$$SNR(S) = 20 \log_{10} \left(\frac{S}{\sigma_N} \right) (dB) \quad (6.1)$$

All quantities are measured in electrons. The term S represents the signal level while σ_N represents the noise standard deviation. More specifically σ_N is defined as sum of different kinds of noise: $\sigma_N = \sigma_S + \sigma_R + \sigma_{DSNU} + \sigma_{PRNU}$ where σ_S , σ_R , σ_{DSNU} , σ_{PRNU} are shot noise, read noise, dark signal non uniformity and photon response non uniformity respectively. These types of noise will be introduced in Sections 6.3 and 6.4. After acquisition and digital conversion, the image is coded into L levels, where L depends on the bit depth of the Analog to Digital (*A/D*) conversion process. Hence, the *SNR* of an image $I(x, y)$ in this case is defined as:

$$SNR(I) = 20 \log_{10} \left(\frac{E(I)}{\sigma(I)} \right) (dB) \quad (6.2)$$

where $E(I)$ and $\sigma(I)$ are the average value and the standard deviation of the image I respectively. The higher the *SNR*, the better the image.

6.2.2 Dynamic Range

The sensor dynamic range *DR* is expressed in decibels and is defined as ([1]):

$$DR = 20 \log_{10} \left(\frac{S_{max}}{S_{min}} \right) (dB) \quad (6.3)$$

where S_{max} represents the photodiode charge capacity (*full well*) in electrons and S_{min} represents the minimum temporal noise value (*noise floor*) at zero exposure. The well capacity decreases as pixel size shrinks, reducing the dynamic range. However, with

improvements in sensor design it is possible to reduce the noise levels associated with small pixels, allowing increasing the dynamic range.

6.3 Photon Shot Noise

Image sensors record light by capturing photons into the photodiodes, eventually converting them into numbers. During the integration time, the arrival rate of photons at each photosite is not constant; rather, there is an intrinsic uncertainty caused by the oscillations of the number of photons that reach the imager. Consequently, even in the ideal case of constant and uniform light intensity, each photosite can receive a different number of photons; yet, on average, the number of photons per pixel is the same. These oscillations can be modeled by Poisson distribution.

6.3.1 Poisson Distribution

The Poisson distribution approximates the probability of a number of independent events occurring in a given timeframe at a known average rate, and has only one parameter λ , which represents both its mean and variance. A random variable X taking values $0, 1, \dots$ is Poisson distributed if:

$$P\{X=m\} = \frac{e^{-\lambda} \cdot \lambda^m}{m!}, \quad m=0, 1, \dots \quad \lambda > 0 \quad (6.4)$$

Photons hit independently each photosite at a certain average rate; it follows that *Photon Shot Noise* (*PSN*) has Poisson statistics. The variance of *PSN* is given by the square root of the number of photons hitting the sensor in the integration time [4]:

$$\sigma_{PSN} = \lambda_{PSN} = \sqrt{N} \quad N = \text{number of photons in the integration time} \quad (6.5)$$

Therefore, oscillations in the photons arrival rate are of the order \sqrt{N} . According to (6.5) it is evident that *PSN* increases as the amount of light (i.e., photons N) augments. Despite this, a larger number of photons also yields higher *SNR*, consequently *PSN* can be regarded as a problem only in case of low light. If the signal is strong, not only *SNR* is higher, minimizing the impact of *PSN*, but also Poisson distribution can be approximated by the Gaussian distribution allowing simpler denoising treatment (see Fig.(6.2)).

PSN is intrinsic in the quantity (i.e., light) recorded by the sensor and cannot be completely eliminated. However it can be reduced if larger pixels are adopted, maximizing the number of integrated photons even in very low light conditions. This, of course, increases the size and cost of the sensor, hence suitable trade-offs have to be found. In fact, given two sensors having the same resolution but pixels of the different sizes, the sensor with larger pixels is, in principle, capable of collecting more photons yielding higher *SNR*. We can conclude that the impact of *PSN* is greater with: fast shutter speeds, low light condition.

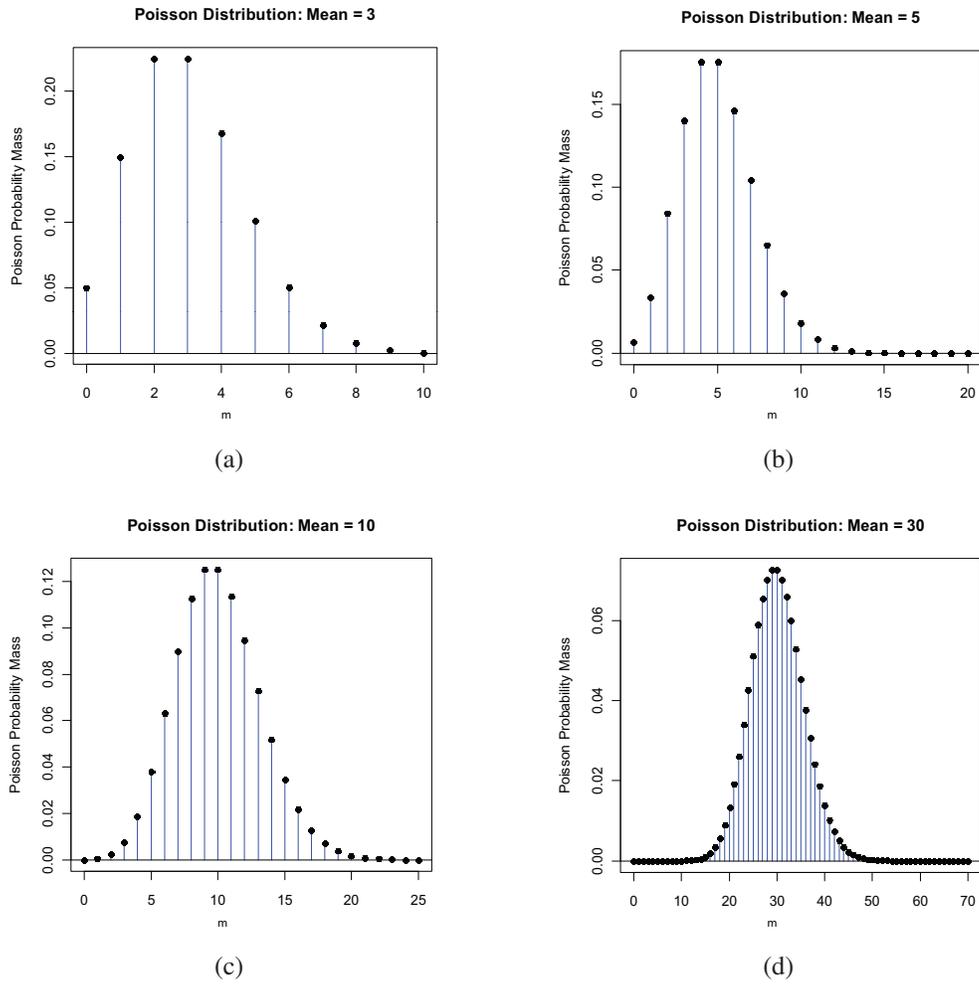


Figure 6.2 : Poisson distribution approaches Gaussian distribution as the mean increases.

6.4 Noise Types and Models

There is not just one single noise source, rather, many factors contribute to signal degradation. Each noise source injects extra-information in the ideal noise-free image signal; eventually unpleasant images are generated if noise is not properly treated.

6.4.1 Fixed Pattern Noise and Temporal Noise

Noise in a digital raw image can be classified into two main categories:

1. Fixed Pattern Noise (*FPN*);
2. Temporal (Random) Noise;

In *FPN*, the term *fixed* refers to the fact that this noise has a pattern which is invariant with respect to time. *FPN* has two main components, one at dark and one under illumination. The dark component is known as *dark-FPN* and it is present even in absence of illumination. The *FPN* under illumination is called *PRNU* (Pixel Response Non Uniformity) and is caused by different sensitivity of the pixels to light. If the image sensor contains column amplifiers *dark-FPN* may appear as vertical stripes in the image (*column-FPN*) that are very annoying and easily detected by the human eye. A technique for removing *column-FPN* is described in Section 6.9.1.

Temporal (random) noise, on the other hand, fluctuates over time, hence it appears different in intensity and spatial position in different frames. Temporal noise is the sum of different noise sources generated on the imager during the acquisition process (e.g., photon shot noise). Other sources of temporal noise are:

- **Dark current noise:** represents the temperature dependent noise generated on the surface of the image sensor. Noise is introduced by the sum of electrons freed by the thermal energy plus electrons generated by the photons hitting the imager.
- **Readout noise:** is the electronic noise generated during the sensor readout process.
- **Reset noise:** is generated by residual electrons left in sensors capacitor after the reset operation, which is performed before a new scene acquisition occurs.
- **Quantization noise:** is due to conversion of photons into a digital number performed by an A/D converter. The errors introduced in the conversion of an analog signal to a set of discrete digital values are known as quantization errors. In particular, quantization noise significantly affects image quality when the bit-depth of the digital conversion process is small.

6.4.2 Additive Noise Model

Consider an ideal image I with size $M_1 \times M_2$, denoted as $I = [i(x, y)]_{M_1 \times M_2}$, such that $i(x, y) \in \{0, \dots, L-1\}$, $0 \leq x \leq M_1 - 1$, $0 \leq y \leq M_2 - 1$. Ideal image I contains no noise,

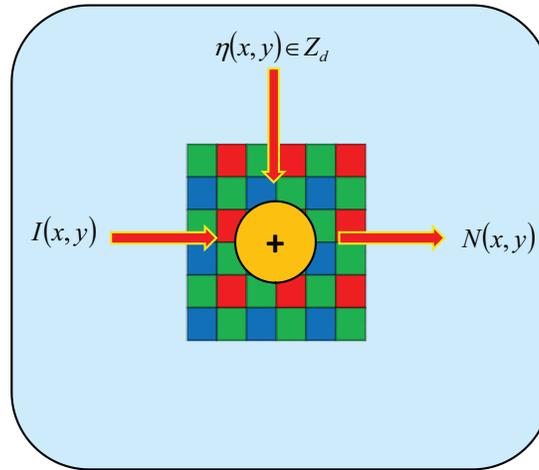


Figure 6.3 : Additive noise model. The ideal image signal is contaminated by a noisy signal $\eta(x,y)$ whose intensities are drawn from an underlying noise distribution Z_d .

every pixel being the exact representation of the light intensity perfectly recorded and converted by the sensor. In the additive noise model, each pixel of the ideal image is contaminated by a random value drawn from a certain underlying noise distribution Z_d ; this random quantity adds to the original ideal signal, generating the noisy observed image $N(x,y)$:

$$N(x,y) = I(x,y) + \eta(x,y) \quad (6.6)$$

The term $\eta(x,y)$ which is added to the ideal value $I(x,y)$ is generated by the contribution of many overlapping noise sources. Because of the central limit theorem, a common assumption is to model the contribution of all noise sources as zero mean *Additive White Gaussian Noise (AWGN)*. Eventually, the noisy term $N(x,y)$ is then observed and recorded.

6.4.3 Central Limit Theorem

Before proceeding, we recall the Central Limit Theorem (CLT). Consider n independent and identically distributed (i.i.d.) random variables X_1, X_2, \dots, X_n , each one having a certain mean μ and variance $\sigma^2 > 0$. Let S_n be the sum of each X_i , $i = 1, \dots, n$:

$$S_n = \sum_{i=1}^n X_i \quad (6.7)$$

Consider the new variable:

$$Z_n = \frac{S_n - n\mu}{\sigma\sqrt{n}} \quad (6.8)$$

The CLT states that the distribution of the sample average of the random variables converges to the normal distribution with mean μ and variance σ^2/n even if the X_i have

different distributions. In other words the distribution of Z_n converges in distribution to the normal standard distribution $\mathcal{N}(0, 1)$ as the number of added i.i.d. X_i approaches infinity:

$$Z_d \xrightarrow{D} \mathcal{N}(0, 1) \quad (6.9)$$

6.4.4 Additive White Gaussian Noise Model

AWGN is the most widely adopted noise model; this assumption arises from the central limit theorem: all noise sources overlap, finally producing a zero mean Gaussian distributed noise. More specifically, the theorem states that the sum of a large number of independent random variables is Gaussian distributed. In order to correctly apply the CLT, the following properties must be satisfied:

- Each single random variable must be independent;
- Each term in the sum must be small compared to the overall sum;
- There must be a large number of random variables contributing to the sum.

These assumptions fit well with the fact that not all noise sources have Gaussian distribution. Probability Density Function (PDF) of the Gaussian distribution is shown in Fig.(6.4) and is modeled as:

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (6.10)$$

where x is the signal intensity, μ and σ are respectively the mean and standard deviation of the signal x (see Fig.(6.4)) [1].

Some key properties of the normal distribution often used in noise reduction algorithms are given below. The probability that a randomly selected value of a variable x falls between the values a and b is defined as:

$$P(a \leq x \leq b) = \int_a^b f(x) dx \quad (6.11)$$

Let z be the z -score defined as:

$$z = \frac{(x - \mu)}{\sigma} \quad (6.12)$$

The Chebychev theorem states that for any population or sample, the proportion of observations, whose z -score has an absolute value less than or equal to k , is no less than $(1 - (1/k^2))$:

$$P(x \leq k) \geq 1 - \frac{1}{k^2} \quad (6.13)$$

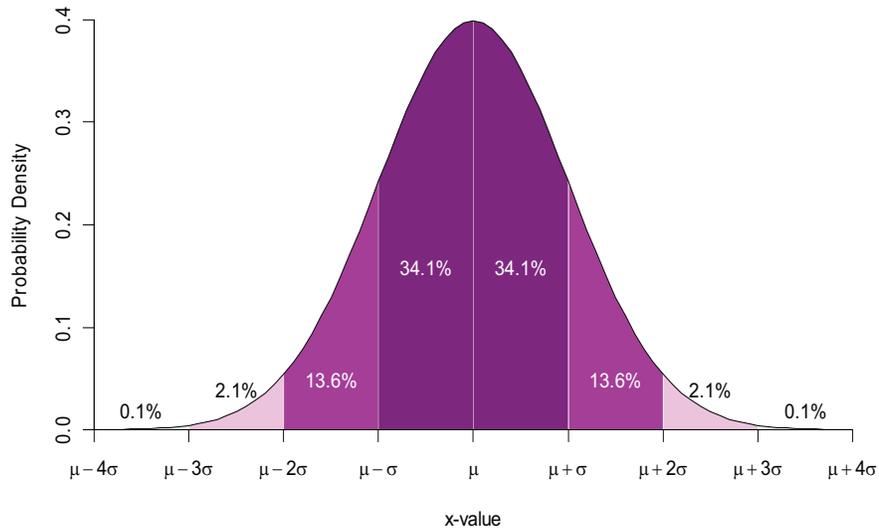


Figure 6.4 : Probability density function of the Gaussian (normal) distribution.

In case of Gaussian distribution, the Chebychev theorem can be further refined. In particular the following properties hold:

$$\begin{aligned}
 P(-1 \leq z \leq 1) &= \int_{-1}^1 f(z) dz = 68.27\% \\
 P(-2 \leq z \leq 2) &= \int_{-2}^2 f(z) dz = 95.45\% \\
 P(-3 \leq z \leq 3) &= \int_{-3}^3 f(z) dz = 99.73\%
 \end{aligned}
 \tag{6.14}$$

In other words:

- 68% of the samples fall within -1 and +1 standard deviations from the mean;
- 95% of the samples fall within -2 and +2 standard deviations from the mean;
- 99% of the samples fall within -3 and +3 standard deviations from the mean.

This implies that there is a small probability that a normally distributed variable falls more than 2 times standard deviations away from its mean.

This noise model is representative of the *small* oscillations that are observed in the pixels values. It must be observed however, that for high levels of noise, the Gaussian distribution bell becomes significantly wide, eventually degenerating to a fat tailed distribution which causes increase of color noise and leaky pixels.

6.4.5 Impulse Noise Model

Image sensors are composed of millions of photodiodes collecting photons. Faulty elements in the sensor array may occur, generating pixels that do not record correct information. The single isolated defective pixels located in random spatial positions of the imager



Figure 6.5 : Impulse noise.

are referred to as impulse noise. The defective nature of a pixel can be classified into two main classes: *fixed-valued* and *random-valued* impulse noise. The following definition shows the *fixed-valued* impulse noise pdf:

$$f(x) = \begin{cases} f_a & \text{if } x = a \\ f_b & \text{if } x = b \\ 0 & \text{otherwise} \end{cases} \quad (6.15)$$

For a 8-bit image, $a = 0$ yields black pixels in the image (*dead pixels*), and $b = 255$, produces clipped values (*spikes*). Pixels affected by fixed-valued impulse noise always appear defective unless they are masked by texture, and they can be corrected using a defect map, which stores the position of the faulty elements. The correction stage uses information from the neighboring pixels. Fig.(6.5) shows an image contaminated by fixed-valued impulse noise.

Leaky pixels do not respond well to light, rather, their response is uncertain, causing *random-valued* impulse noise (i.e., impulse noise with variable amplitude). The behavior of leaky pixels is not constant and varies according to external factors such as temperature; this extra uncertainty makes leaky pixels position almost unpredictable. For an image contaminated with impulse noise, the impulse noise ratio Q can be defined as:

$$Q = \text{Number of impulse defective pixels} / \text{Total number of pixels} \quad (6.16)$$

The position of the defects and their amplitude are two independent quantities, hence, the map of defects D is defined as the point by point multiplication between D_{POS} and D_{AMP} [5]:

$$D = D_{POS} \cdot D_{AMP} \quad (6.17)$$

where:

- D_{POS} is $M \times N$ binary matrix mapping the positions of the impulse noise;
- D_{AMP} is $M \times N$ representing the amplitudes of the impulse noise at each pixel position;

The following probabilities can be then defined:

$$\begin{aligned} P\{D_{POS}(x,y) = 1\} &= n \\ P\{D_{POS}(x,y) = 0\} &= 1 - n \end{aligned} \quad (6.18)$$

where $x = 1, \dots, M$, $y = 1, \dots, N$ and $0 \leq n \leq 1$. Binary distribution (6.18) indicates that position (x,y) is faulty with probability n and correct with probability $1 - n$. The correction of impulse can incur into three classes of errors:

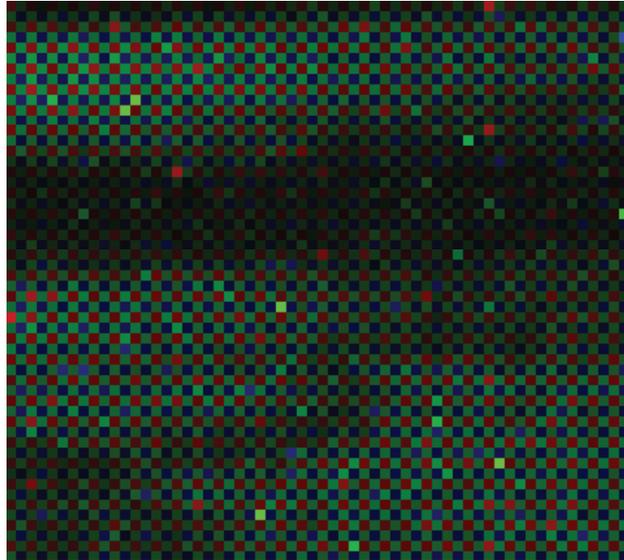
- Type I: this type of errors simply refer to the case in which a defective element is not detected (false negative); this error causes a visible, not corrected, defect in the final image unless it is masked by texture.
- Type II: a pixel not affected by impulse noise is erroneously classified as defective and corrected (false positive). Type II errors occurring in textured areas of the image cause loss of detail because important information related to sharpness is lost after correction. False positives in homogeneous areas are not a problem because overcorrecting a homogeneous area does not produce visible artifacts.
- Type III: a defective pixel is correctly classified and corrected, nonetheless its correction augments defectivity (overcorrection problem). This category of errors is more subtle and refers to the case in which the correction of the defect produces a new value which is more defective and visible than the previous one.

As the pixel size decreases and the operating conditions of the imager become critical (e.g., high temperature, low light, etc.) the probability of occurrence of adjacent defective pixels augments. For example, adjacent leaky pixels, in certain conditions behave as couplets of defective pixels, particularly visible and annoying in uniform and dark backgrounds (heavy tailed noise). Couplets are difficult to remove because two adjacent defective elements may be considered as part of an edge and not corrected. To cope with this problem, ad-hoc defect correction algorithms must be used or properly tuned defect maps have to be built [6]. Fig.(6.6) shows the defective and filtered version of a Bayer image in false color.

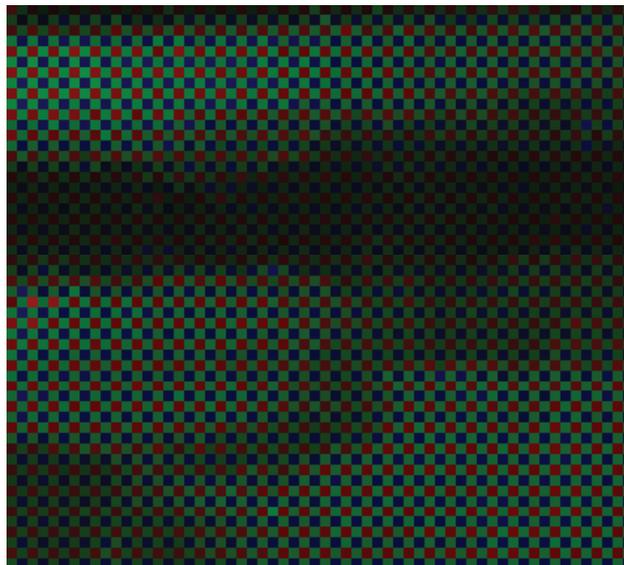
Strong defect correction, such as heavy median filtering, can cause significant resolution loss generating unpleasant blurred images.

6.4.6 Heavy Tailed Noise

In some cases the hypotheses of the central limit theorem are not completely satisfied: the sum of the underlying random variables may include not independent addends, or the contribution of some variables may be too large. In these cases, the simple Gaussian noise



(a) Colorized defective Bayer image.



(b) Colorized Filtered Bayer image.

Figure 6.6 : Defective Bayer image.

model does not fit well the actual noise distribution. In particular, although the central part of the noise distribution fits the Gaussian model, the tails may not. More specifically, it is frequent the case in which the noise distribution tails do not tend to zero as fast as the Gaussian distribution. Statistical distributions exhibiting this property are known as heavy-tailed distributions (see Fig.(6.7)). Consequently, noise modeled by heavy tailed distributions is called heavy tailed noise. It is useful to remind some key properties that define the shape of a distribution.

6.4.7 Shape Measures of a Distribution

There are two quantities that describe the shape of a statistical distribution:

- **Skewness**: indicates the degree of distortion (asymmetry) along the horizontal axis and around the mean of the distribution.
- **Kurtosis**: indicates the degree of peakness of the distribution.

To define the above quantities, the definition of *central moments* of a distribution must be provided. The n -th central moment \mathcal{C}_n of a random variable X is defined as:

$$\mathcal{C}_n = E(X - E(X))^n \quad (6.19)$$

where $E(\cdot)$ is the expectation operator. According to (6.19) the first three central moments are computed as follow:

- $n=0$: $\mathcal{C}_0 = E(X - E(X))^0 = 1$;
- $n=1$: $\mathcal{C}_1 = E(X - E(X))^1 = 0$;
- $n=2$: $\mathcal{C}_2 = E(X - E(X))^2 = \sigma^2$;

The next central moments ($n = 3, 4$) are useful to measure distribution shape:

- $n=3$: \mathcal{C}_3 defines the *skewness* of the distribution;
- $n=4$: \mathcal{C}_4 defines the *kurtosis* of the distribution.

A distribution having skewness close to zero is symmetric around its mean (Gaussian distribution). If skewness is greater than zero, the distribution exhibits a right tail, whereas negative skewness indicates a left tail. Depending on the kurtosis value, statistical distributions can be classified in the following categories:

- *leptokurtic (highly peaked distribution)*;
- *mesokurtic (normally peaked distribution)*;
- *platycurtic (flat peaked distribution)*.

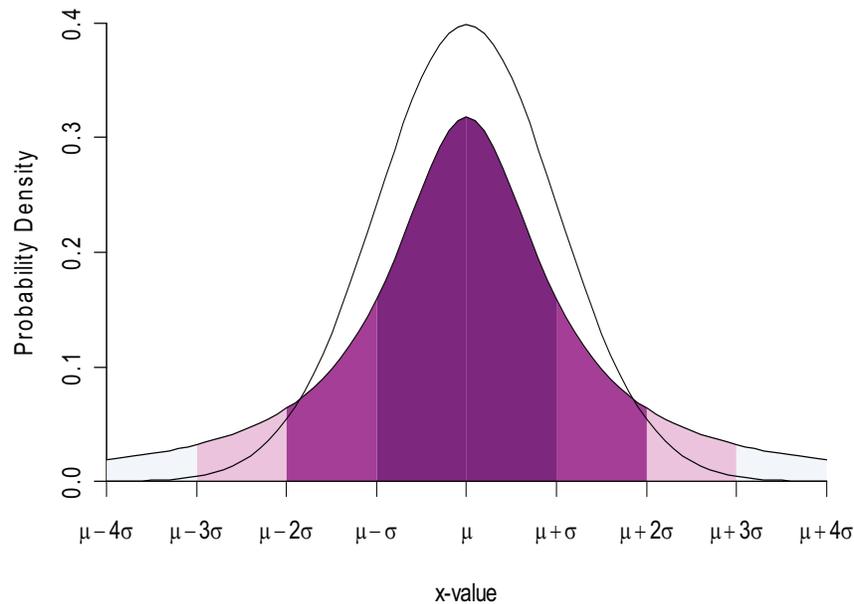


Figure 6.7 : Normal distribution (outlined) and Fat Tailed Distribution.

Leptokurtic distributions have a higher and sharper *peak and present fatter tails*. In case of low light, the shape of the Gaussian bell becomes particularly wide, eventually degenerating to a fat tailed distribution. In case of noise with an underlying fat tailed distribution there is a high probability of relatively high deviations from the correct pixel value.

Fat tailed noise mainly appears in low light images and depends on gains and temperature. It manifests as high amplitude random *impulse noise* in non predictable locations at "shot time" (i.e., when the image is taken). An algorithm capable to deal with unpredictable defects must be devised (on the fly defect-correction); simple defect-maps cannot be effectively used because the *fat* defects position is not predictable.

6.5 Noise in Image Pipeline

Despite the efforts in reducing noise introduced during the acquisition process, the residual unfiltered noise may be amplified in the subsequent image pipeline processing steps. This is a problem especially in low light conditions because analog and digital gains are set high in order to produce an acceptable picture. This Section describes the main noise amplification effects of a typical image generation pipeline.

Noise reduction can take place in different stages of the image processing pipeline. In order to keep noise levels low, it may be necessary to perform more than a single noise reduction stage.

Unfiltered sensor noise can also introduce artifacts in colors that are difficult to recover

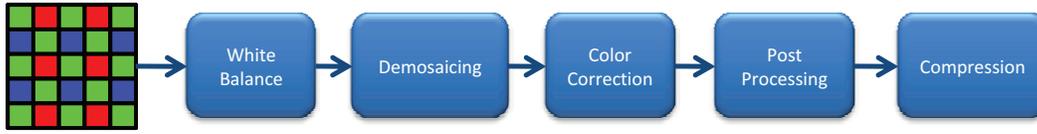


Figure 6.8 : Image pipeline.

after demosaicing, because color interpolation mixes noises of different color channels, increasing signal and noise correlation.

Noise can change significantly its intensity and statistics in the different stages of the pipeline. Before the application of the pipeline algorithms, the acquired raw image contains noise which is the juxtaposition of many different sources, overlapping and contaminating the ideal signal. The amplification necessary to produce a correctly exposed image, impacts the amount of noise in the acquired picture. Temporal random noise describes the 2D spatial distribution inside the image changes from frame to frame.

In addition, as shown in Fig.(6.1), each block affects noise statistics [7]. Algorithms in Fig.(6.8) and their side-effects on the noise distribution are described in detail in the following subsections.

6.5.1 White Balance effects

The first block having a high impact in noise amplification is the White Balance (*WB*). The *WB* algorithm typically applies three different global gains (one for each CFA channel) in order to compensate the amounts of red, green and blue such that the neutral colors are represented correctly (see Chapter 5). Let I_R , I_G , I_B be the red, green and blue pixels of the CFA image respectively. Let g_R^{WB} , g_G^{WB} , g_B^{WB} be the gains applied to each CFA color channel according to:

$$\begin{aligned} I_R^{WB} &= g_R^{WB} I_R \\ I_G^{WB} &= g_G^{WB} I_G \\ I_B^{WB} &= g_B^{WB} I_B \end{aligned} \quad (6.20)$$

Hence, the noise level (*NL*) in each CFA plane is modified in the following ways:

$$\begin{aligned} NL(I_R^{WB}) &= g_R^{WB} NL(I_R) \\ NL(I_G^{WB}) &= g_G^{WB} NL(I_G) \\ NL(I_B^{WB}) &= g_B^{WB} NL(I_B) \end{aligned} \quad (6.21)$$

6.5.2 Demosaicing Effects

The demosaicing process allows recovering the color image from the interspersed samples of the Bayer pattern. The algorithm chosen to reconstruct the color image impacts the noise levels because of changes in the spatial correlation of data (see Chapter 7 for more

details). To show the effects of demosaicing on noise level a simple algorithm which recovers the color component at location (x, y) by averaging the available color components in the neighborhood is employed. For example, if the current (x, y) is the site of a green sample, the missing red and blue components are recovered by using:

$$\begin{aligned} I_R^*(x, y) &= \frac{I_R^{WB}(x-1, y) + I_R^{WB}(x+1, y)}{2} \\ I_B^*(x, y) &= \frac{I_B^{WB}(x-1, y) + I_B^{WB}(x+1, y)}{2} \end{aligned} \quad (6.22)$$

The NL related to the red and blue components interpolated at pixel (x, y) are defined as ([7]):

$$\begin{aligned} NL(I_R^*(x, y)) &\cong \frac{NL(I_R^{WB}(x-1, y))}{2} \\ NL(I_B^*(x, y)) &\cong \frac{NL(I_B^{WB}(x-1, y))}{2} \end{aligned} \quad (6.23)$$

NL is scaled by a factor of two because of the average operation on data. The spatial correlation of noise also increases.

6.5.3 Color Correction Effects

Color correction is necessary because the response of the color filters placed on top of the imager do not match the one of the human eye; consequently, the RGB values must be corrected using a proper 3×3 matrix that adjusts the values accordingly (see Chapter 5). This multiplication changes the pixel values but, meanwhile, increases noise and reduces the SNR, especially in the blue channel:

$$\begin{bmatrix} I_R^c \\ I_G^c \\ I_B^c \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} I_R^* \\ I_G^* \\ I_B^* \end{bmatrix} \quad (6.24)$$

(NL) for each color channel change in the following way ([7]):

$$\begin{aligned} NL(I_R^c) &= c_{11}NL(I_R^*) + c_{12}NL(I_G^*) + c_{13}NL(I_B^*) \\ NL(I_G^c) &= c_{21}NL(I_R^*) + c_{22}NL(I_G^*) + c_{23}NL(I_B^*) \\ NL(I_B^c) &= c_{31}NL(I_R^*) + c_{32}NL(I_G^*) + c_{33}NL(I_B^*) \end{aligned} \quad (6.25)$$

6.5.4 Sharpening, Gamma Correction and Compression Effects

The demosaicing process reconstructs the full RGB color image starting from the Bayer samples; this process is essentially a low-pass operation, hence the output image is weak in terms of sharpness and it looks blurred; therefore a sharpening algorithm is mandatory in order to obtain an acceptable image. Subsequently a gamma correction algorithm is also applied, to align the linear response to light intensity of the imager to the nonlinear response of the human visual system.

Sharpening and gamma correction algorithms improve image quality, but increase noise as well. The sharpening algorithm amplifies the high frequencies, consequently increasing the image noise. Gamma correction modifies luminance values to enhance contrast in the dark regions; due to its nonlinearity, it makes the noise distribution for each signal level even more complex to describe. Compression is used to reduce the size of image files. There are two types of compression: *lossless*, where the amount of image data is reduced without loss of information, and *lossy* (e.g., *JPEG*) where image file is reduced allowing the loss of a part of data.

JPEG converts the *RGB* image in the *YCbCr* color space; the luminance plane (*Y*) is used for recognizing structures and details while the chrominance planes (*CbCr*) are subsampled without significant loss of image information for the observer. The *Y* plane is divided in 8×8 blocks that are compressed separately and transformed in the frequency domain using *DCT* (*Discrete Cosine Transform*). *DCT* coefficients are then quantized using a quantization table. Compression rate depends on the used quantization table; the higher the compression rate the lower the image quality, due to presence of artifacts. Lossy image compression is similar to a noise reduction algorithm, that maintains the main image structures and suppresses fine textures and details. Anyway, artifacts introduced by compression reduce global image quality.

6.5.5 Noise Reduction Block Position

According to the previous considerations about the image pipeline, the position of the noise reduction stage strongly affects the quality of the final image. Basically, in order to avoid false colors and increment of noise spatial correlation, it is important to implement noise reduction before demosaicing. Nonetheless, as discussed above, not all noise can be removed before (or jointly to [8]) demosaicing; the residual noise is further amplified by the color correction and sharpening algorithms, hence a new application of noise reduction is generally required at the end of the pipeline. Fig.(6.9) shows a possible image processing pipeline with two noise reduction stages. The first denoising stage is applied in the *CFA* domain, before demosaicing; the second noise filtering stage works in the luminance domain and is positioned at the end of pipeline before compression.

6.6 Luminance Noise

Luminance noise appears as neutral grain in the final *RGB* image. If not excessive, the image may even benefit from this noise because it contributes to the overall sense of sharpness, avoiding cartoon-like looking pictures. In other cases, especially when pixel size is very small, high levels of noise are present, even in good light conditions; residual luminance noise at the end of the pipeline cannot be left untreated because it is particularly annoying especially in homogeneous areas where the eye easily detects it.

In the *RGB* domain, luminance information is interspersed in the red, green and blue channels; in order to separate luminance from chrominance information, a conversion to the *YCbCr* color space is needed. Other color spaces separating luminance from chromi-

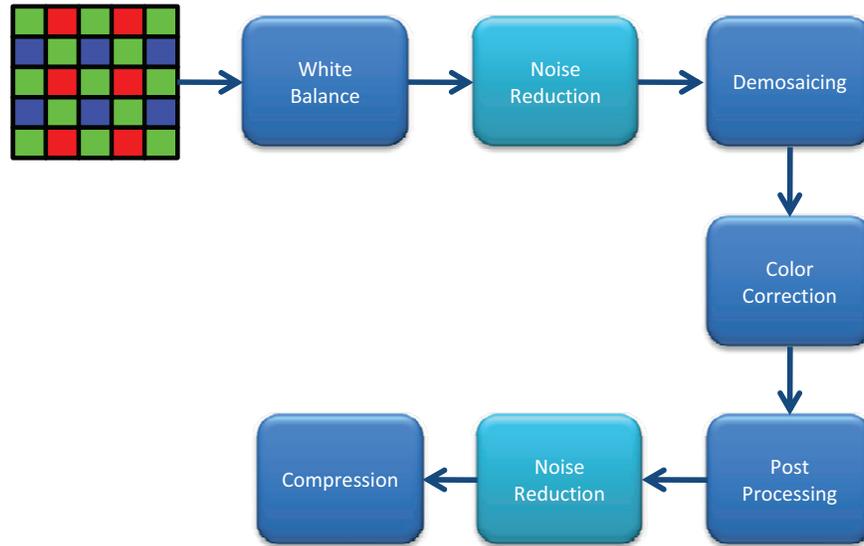


Figure 6.9 : Noise reduction blocks in Image pipeline.

nance exist (e.g., $L^*a^*b^*$); despite this, $YCbCr$ conversion comes for free because it is already implemented inside the *JPEG* engine. The luminance channel is given by a weighted average of the RGB values at any given pixel. Luminance is defined as (ITU-R BT.709):

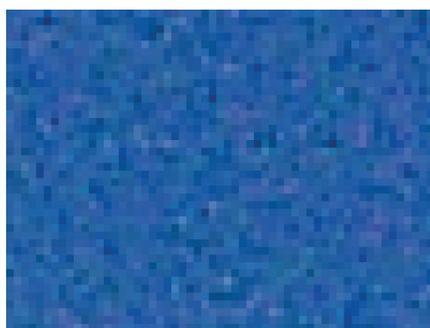
$$Y = 0.2125 * R + 0.7154 * G + 0.0721 * B \quad (6.26)$$

In contrast with raw CFA in which noise has a specific characterization curves, in $YCbCr$ noise depends also on how well the algorithms in the pipeline deal with corrupted signals. Typically, a luminance filter can be placed at the end of the pipeline, before the compression step, in order to reduce the possible excessive residual noise in the Y plane (Fig.(6.9)). Fig.(6.10) shows a noisy flat area and corresponding denoised images with and without the luminance noise filtering.

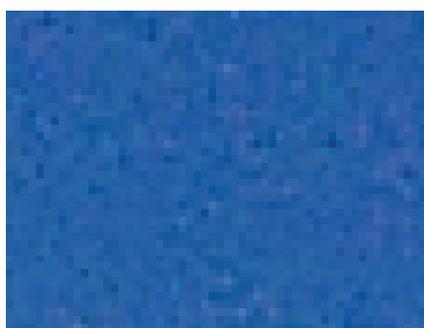
6.7 Chrominance Noise

Differently from luminance noise which is neutral and visible as brightness variations from pixel to pixel, chrominance noise is a low frequency distortion appearing as large colored blobs in the final image. These blobs are usually reddish or bluish and appear more visible especially in dark regions of low quality images (i.e., acquired in low light or underexposed). A chroma noise example is show in Fig.(6.11).

The low frequency nature of chroma noise does not allow efficient filtering under the assumption of the Gaussian noise model. Frequencies of chroma noise are very low, producing large color blobs, hence simple weighted averages do not reduce color noise



(a) No noise treatment.



(b) CFA noise treatment without final luminance noise filter.



(c) CFA noise treatment + residual noise Y filter.

Figure 6.10 : Denoising on CFA and on Luminance plane.

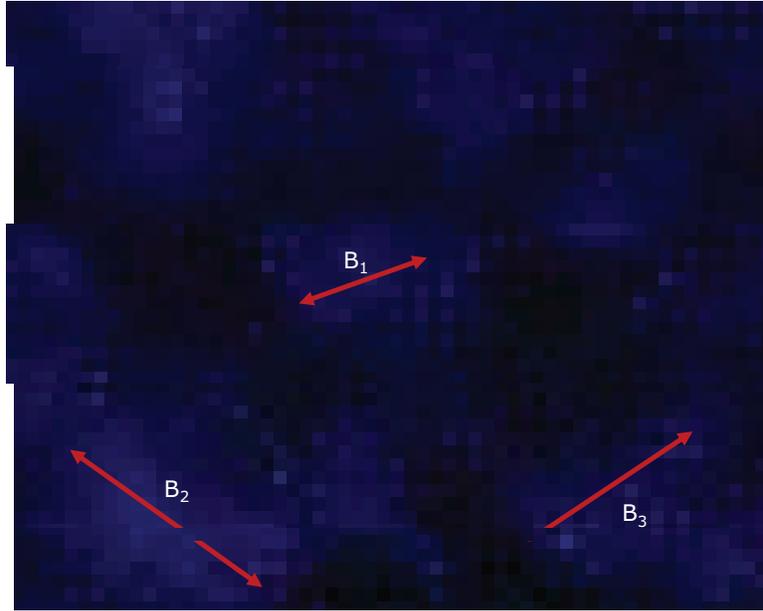


Figure 6.11 : In case of very high noise, colored blobs appear. Colored patches are visible in areas that should appear as uniformly dark and neutral. In the figure, three blobs B_1, B_2, B_3 are highlighted. Their extension can be of several pixels.

effectively. Ad-hoc processing must be performed in order to reduce this noise effectively. Due to the large size of the color blobs, large filter masks must be used; typically, undersampling and large masks are used to create very large virtual filter masks.

6.8 Noise Estimation

As discussed in Section 6.4, the zero mean *AWGN* noise model requires the estimation of the standard deviation of the underlying Gaussian noise distribution. Pixels deviate from their correct value by some value which is drawn from a Gaussian distribution; usually pixel fluctuations are small, but greater fluctuations are also possible. Nonetheless, in 99% of the cases, the deviations do not exceed 3 times sigma in absolute value. Large noise amplitudes generated by the distribution tails are possible; in this case the pixel of interest might appear as a spike or dead element. The knowledge of a good σ estimation allows filtering the image data properly, significantly reducing the unpleasant effects of Gaussian noise. Furthermore, σ can also be a reference value for detecting outliers.

Olsen [9] analyzed six methods for noise standard estimation and showed that the best was the average method, which is also the simplest. This method consists of filtering the data I with the average filter (a simple box function) and subtracting the filtered image from I . Then a measure of the noise at each pixel is computed. To avoid contribution of image edges to the estimate, the noise measure is rejected if the magnitude of the intensity gradient is greater than a fixed threshold, T .

Estimation of noise standard deviation is based on the following general ideas:

- Locate homogeneous areas in the image, because in flat areas pixel fluctuations are supposed to be caused exclusively by random noise.
- Compute the local variance in the detected flat areas.
- Repeat the previous two steps until the whole image has been processed.
- Estimate the signal variance using the accumulated noise samples.

Another alternative solution consists in filtering the noisy image to suppress image structure and then estimate the noise level on the filtered data.

6.8.1 Fast Noise Estimation

A rough approximation of the noise level in an image can be obtained by exploiting the statistical properties of the Gaussian noise distribution (6.14). It is reasonable to suppose that the image cannot contain an arbitrary high noise level [6]; hence we initially start with the assumption that the maximum noise level is σ_{max} ; this value is obtained using a tuning phase in which the behavior of the image sensor is characterized under different illumination conditions.

Assuming a 3x3 filter support, the absolute differences $\delta_0, \delta_1, \dots, \delta_7$ between the central pixel P_c and its neighborhood are computed:

$$\delta_i = |P_c - P_i| \quad i = 0, \dots, 7 \quad (6.27)$$

If $\delta_i \in [0, \sigma_{max}], i = 0, \dots, 7$ then the assumption of having localized a homogeneous area can be made. The idea is to build a noise histogram Ψ that accumulates the collected noise samples in its bins. Let γ_j be the value of the j -th absolute difference $\delta_j, j \in [0, \dots, 7]$ when $\delta_i \in [0, \sigma_{max}], i = 0, \dots, 7$; in this case the bin γ_i in Ψ is incremented:

$$\Psi(\gamma_j) = \Psi(\gamma_j) + 1 \quad (6.28)$$

After processing the entire frame, the absolute differences accumulated in the histogram will be Gaussian-like shaped. Because of the absolute values, only the positive side of the x -axis is filled; this is not a problem because the normal distribution is symmetric around its mean value, which is zero in our case. To determine the noise standard deviation we consider the property of the Gaussian distribution stating that 68% of its samples fall in the interval $[\mu - \sigma, \mu + \sigma]$. The histogram of the absolute differences is integrated until the 68% of the total samples has been accumulated. As soon as the histogram integrations stops, the value on the x -axis allowing reaching the 68% of the total samples represents the estimated noise standard deviation (Fig.(6.12)):

$$\sigma_{est} = \{ \max k | \sum_{i=1}^k \Psi(i) \leq \lceil 0.68 \cdot \Sigma_{samples} \rceil \} \quad (6.29)$$

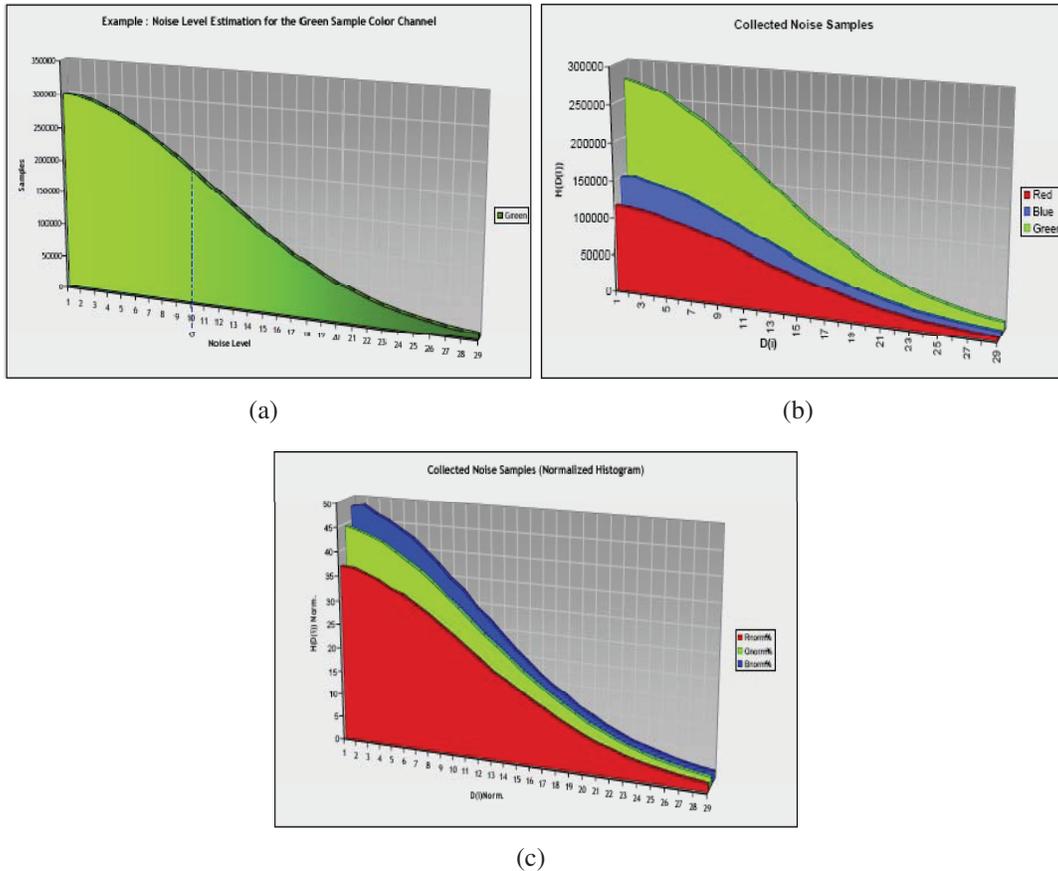


Figure 6.12 : Noise level estimation using noise histogram Gaussian-like distributed.

This solution is strongly based on the value originally chosen for σ_{max} . This number has to be carefully generated by performing a tuning phase which consists in testing the image sensor under different light conditions and determining the typical worst case noise situations. These noise levels will set an upper bound for $3\sigma_{max}$. Nonetheless, the gathered sample of the noise population could be contaminated by the real signal. A possible solution that can minimize the bias problem is to allow σ_{max} to change over time; if σ_{max} is initially overestimated, then σ_{max} can be decreased for the next iteration. This allows progressively reducing the sample bias and converging to the optimal estimation. The method can be further refined using a more sophisticated texture detector, like the one described in [10].

Though the estimation is not perfect and may be biased, it is anyway an approximation, indicating the overall noise level. On a *CFA* image this method will generate a single σ value for each color channel.

6.9 Noise Filtering

The noise filtering problem can be described as the process of effectively removing the unwanted noisy component from the acquired signal, restoring the original ideal data, without sacrificing the image sharpness and features (i.e., color component distances, edges, sharpness, etc.). The following sections describe some techniques for the removal of the fixed pattern noise [11], random noise [11, 12] and temporal noise for video acquisitions [13].

6.9.1 Column-FPN Filtering

The column-Fixed Pattern Noise (*FPN*) is caused by column amplifiers, and appears as vertical stripes in the image (see Section 6.4). Since *FPN* is equal in all acquisitions, for its effective cancellation, it is necessary to estimate its signature. Once the signature is learned, it can be subtracted from the image data. *FPN* estimation can be performed using supplementary data provided by the image sensor. As Fig.(6.13) depicts, a series of black and dark lines are placed at the top of the imager, that are not shown in the final color pictures. Black lines have zero integration time while dark lines have the same exposure time as the image lines but they are shielded from the incident light. These considerations imply that:

- black lines contain very little noise (specifically, *FPN* noise only);
- dark lines accumulate almost the same temporal noise as the image, because they have the same integration time of the image lines.

The *FPN* cancellation is achieved by continuously averaging the black sampled data, according to the following equation:

$$FPN_Est = FPN_Est(FPN_Est/LeakC) + (FPN_CurSample/LeakC) \quad (6.30)$$

where:

- *LeakC*: is a constant to weight the previous estimation.
- *FPN_Est*: is the estimation of the *FPN* signature.
- *FPN_CurSample*: is the *FPN* signature, extracted from the current frame.

Denoting with *nb* the number of black lines and with *W* the image width, the current estimation, *FPN_CurSample*, for the *FPN* of image *I* is obtained by averaging each column *j* of the black lines:

$$M_j = \frac{\sum_{i=0,1,\dots,nb} I(i,j)}{nb} \quad j = 1, \dots, W \quad (6.31)$$

FPN_Est is initialized to zero and is updated by means of equation (6.30), each time a new frame arrives. The first estimation, computed on the first frame, is merely a coarse

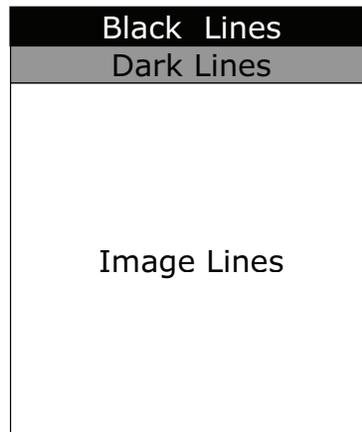


Figure 6.13 : Black lines are used for *FPN* estimation, Dark lines for random noise estimation.

approximation of the real *FPN* signature. After some iterations the estimation converges towards the correct signature that must be row-wise subtracted from the image data in order to get rid of the *FPN*. The *LeakC* value defines how much weight is attributed to the previous estimations; by changing this value, the speed of convergence can be modulated.

Also, the number of black lines used to learn the signature is a key element of the algorithm. If a low number of black lines is used, the estimation would be not reliable, as noise would generate uncertain approximations. On the other hand, using more lines than necessary is a useless waste of resources, both on the sensor and from a computational point of view. Thus, a trade-off between the number of black lines and the leak factor value must be found.

6.9.2 Spatial Filtering

Spatial filters are based on low pass filtering of neighboring pixels under the assumption that the noisy part of the signal is located in its high frequencies. Spatial filters can be partitioned into two main classes: linear and non-linear filters. Linear filters, such as the mean filter, are weak in terms of image details preservation and cannot be successfully adopted for removing noise without blurring the image. Other simple non-linear filters such as the median filter are also weak in terms of detail preservation, basically because this filter applies the same processing without explicitly identifying noise. Nonetheless a median filter has good response in cases in which the noise distribution has long tails.

A vast variety of spatial filters exist and covering them is out of the scope of this chapter [14]. Rather, in the following, two widely used filtering methods known as sigma filter and bilateral filter are analyzed.

6.9.3 Sigma-Filter

If a reliable noise estimator is available, the *Sigma-Filter* [6] represents a fast solution for reducing noise. The filtering process is based on the assumption that the observed pixel value $N(x,y)$ is a good estimate of the local signal mean. The observed pixel value $N(x,y)$ can be expressed as the sum of its representative mean η plus a Gaussian noise term Γ :

$$N(x,y) = \eta(x,y) + \Gamma(x,y) \quad (6.32)$$

We then consider a value $\delta = 3\sigma$ and consider all the pixels in the range delimited by the central pixel value $\pm\delta$. Under the assumption of zero mean *AWGN*, this range includes ~99% of the distribution from the same class as the central pixel. Let M be a $m_1 \times m_2$ filter mask and P_c the value of its central pixel. The final output is a weighted average of the pixels having value close to one of the mask central pixel. Weights decrease as the distance in intensity between the central pixel and the neighborhood augments. Under the assumption of Gaussian noise model, the Sigma Filter averages all the pixels whose values fall in the range $[P_c - 3\sigma, P_c + 3\sigma]$. In particular, pixels whose distance falls in the range $[P_c - \sigma, P_c + \sigma]$ receive maximum weight W_{max} . Pixels whose value falls in the range $[(P_c - \sigma) - \sigma, (P_c + \sigma) + \sigma]$ are weighted with medium weight W_{mid} . Finally, pixels whose intensity falls in the range $[(P_c - 2\sigma) - \sigma, (P_c + 2\sigma) + \sigma]$ are weighted with minimum weight W_{min} . Pixels outside of the range $[(P_c - 3\sigma), (P_c + 3\sigma)]$ are considered outliers having zero weight in the weighted average. Clearly, a reliable noise estimate is necessary, otherwise blurring or lack of noise reduction effectiveness can occur, depending on sigma over- or under estimation respectively.

The final weighted average P_f can be expressed as the sum of the mask pixels multiplied by their respective weights and divided by the sum of the weights:

$$P_f = \frac{\sum_{i=0}^{i \leq (m_1 \times m_2) - 1} W_i \cdot P_i}{\sum_{i=0}^{i \leq (m_1 \times m_2) - 1} W_i} \quad (6.33)$$

The selection of the range $[(P_c - 3\sigma), (P_c + 3\sigma)]$ excludes shot noise pixels and pixels outside a local edge, maintaining sharp edges and allowing effective noise suppression in homogeneous areas. On the other hand, the preservation of sharp edges and strong filtering strength in flat areas also becomes a weakness of this filter. The main problem is that the *Sigma-Filter* has a strong inclusion/exclusion rule in the average process; this, if not well controlled, adds a cartoon-like appearance to the filtered image because transitions become too abrupt [15]. In fact, if a strong edge separating two regions is present and if the grey level difference between both regions is larger than a threshold, the algorithm computes averages of pixels belonging to the same region as the reference pixel creating artificial shocks. In conclusion, the *Sigma-Filter* can create large flat zones and spurious contours inside smooth regions [15]. Not only noise must be reduced but, at the same time, it is necessary to retain a sense of sharpness, depth and focus which manifests through gradual and smooth edge transitions. The bilateral filter satisfies this requirement by applying a smooth weighting scheme in both spatial and intensity domains.

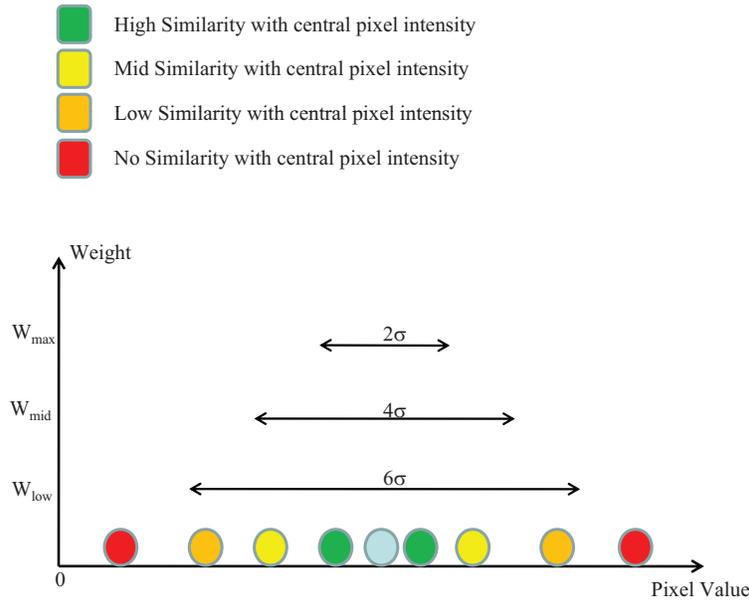


Figure 6.14 : Sigma filter.

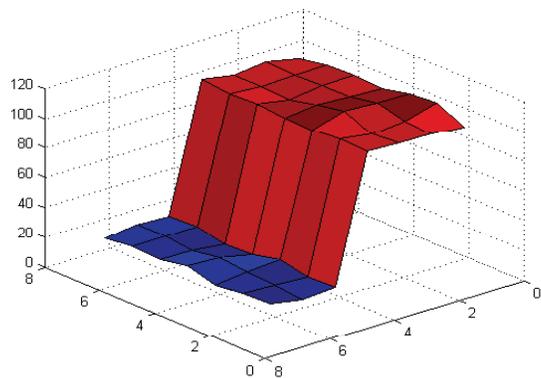
6.9.4 Bilateral Filtering

Bilateral filtering [12, 16] can be seen as an extension of the *Sigma-Filter*. Again, the noise reduction process is based on a weighted average of local samples, but in this case the filter is driven by two different standard deviation values: the intensity related σ_i and the spatial related σ_s . In analogy with the sigma filter, σ_i represents the effective noise level which depends on the pixel intensity values. The additional spatial σ_s is used to weight the pixels in the mask depending on their distance from the center of the mask. Hence, if a low σ_s is used, the pixels far from the central pixel are assigned a low weight and have less importance in the final weighted average. An example of bilateral filtering on a 7×7 mask is shown in Figure 6.15. At a pixel location \vec{x} the output of the filter is given by:

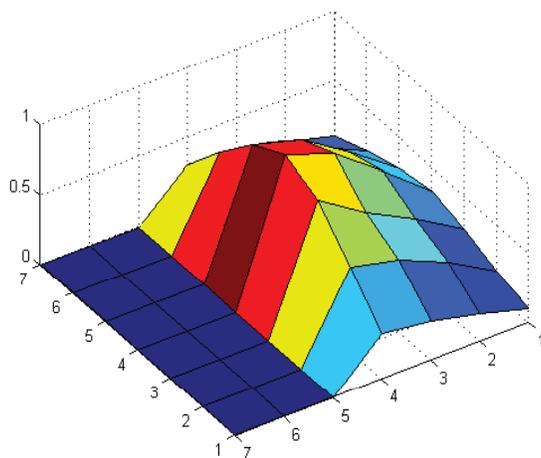
$$I(\vec{x}) = \frac{1}{C} \sum_{y \in N(\vec{x})} e^{\frac{-\|\vec{y}-\vec{x}\|}{2 \sigma_s^2}} e^{\frac{-|I(\vec{y})-I(\vec{x})|}{2 \sigma_i^2}} I(\vec{y}) \quad (6.34)$$

where $N(\vec{x})$ is a spatial neighborhood of pixel $I(\vec{x})$ and C represents the normalization constant:

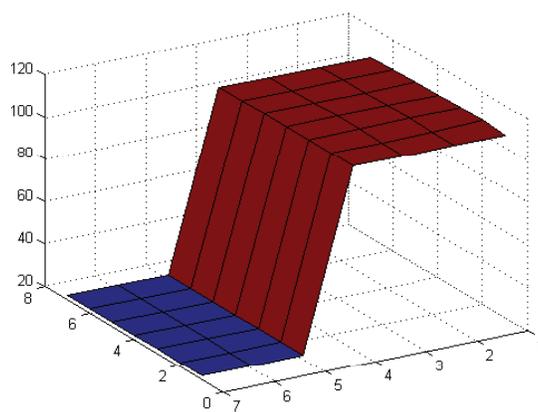
$$C = \sum_{y \in N(\vec{x})} e^{\frac{-\|\vec{y}-\vec{x}\|}{2 \sigma_s^2}} e^{\frac{-|I(\vec{y})-I(\vec{x})|}{2 \sigma_i^2}} \quad (6.35)$$



(a) Noisy input



(b) Filter



(c) Filtered output

Figure 6.15 : Bilateral filter.

6.9.5 Temporal Filtering

Noise types in an image sensor can be partitioned in two main classes: fixed pattern noise and random/temporal noise. When a picture is taken, the acquired signal represents a snapshot of the ideal signal *plus* noise fluctuations. Fixed pattern noise is removed in the sensor by using ad-hoc signal processing, hence the acquired picture is basically contaminated by random noise. This random noise is a bi-dimensional snapshot of the temporal noise at each pixel location, that can be defined as spatial noise.

If the imager is used to acquire a bi-dimensional signal, then spatial techniques to reduce the random noise can be applied. On the other hand, if the application requires the acquisition of a temporal signal (i.e., video) then the application of a spatial filter will not suffice. This is due to the fact that residual noise in each frame will be independent from the other frames. As a result, the same pixel of adjacent frames will be perceived, again, with small annoying fluctuations around its average (flickering). To perform effective noise video filtering, it is necessary to stabilize the signal also in the temporal direction; this can be implemented using a frame buffer to exploit the temporal redundancies of the image signal.

In temporal filtering, noise reduction is performed by using information from adjacent frames (Fig.(6.16)). To take advantage of the temporal redundancies between frames, however, simply adding a frame buffer for storing the previous frame is not very efficient. Clearly, motion may occur between two consecutive frames; hence, a motion estimation algorithm should be performed in order to compensate for movement between frames. If temporal noise reduction is performed in the loop of an MPEG encoder, the motion estimation engine can be shared between the encoder and the filter. Filters using a motion estimator are known as motion compensated temporal filters, whereas non motion compensated filters do not have an underlying motion estimator. It is clear that motion compensated temporal filters are much more efficient in exploiting the spatio-temporal redundancies.

In the simpler scheme, noise from the current frame is removed by considering also data from the previous filtered frame, as in Fig.(6.17).

In more complex schemes, data may be provided by more than one past frames. If video has already been acquired and noise reduction is performed offline, information from "future" frames can also be used.

6.10 Conclusions

This Chapter has provided the fundamentals about the main noise sources, illustrating techniques for noise modeling, estimation and reduction. Even if very complex noise estimation and reduction techniques exist, only fast solutions that can be realistically implemented in electronic devices have been treated; these techniques represent a trade-off between achievable image quality and computational cost. The Chapter has also illus-

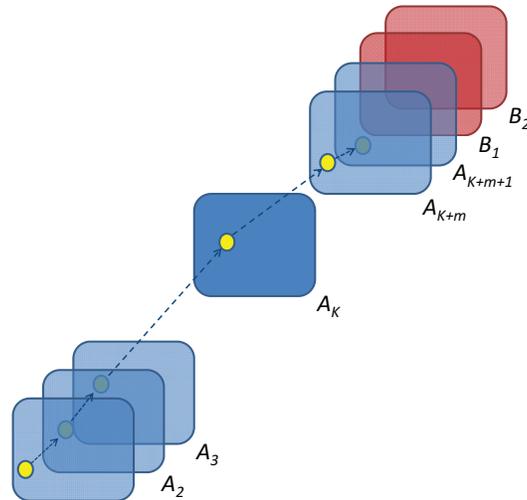


Figure 6.16 : The motion of a pixel (usually an entire block of pixels) is tracked by the motion estimator between temporally adjacent frames. Tracking allows temporal filtering even if the block is located in different spatial positions in the temporal sequence. Non motion compensated temporal filters cannot take advantage of this and can only average temporally if no apparent motion occurred between frames.

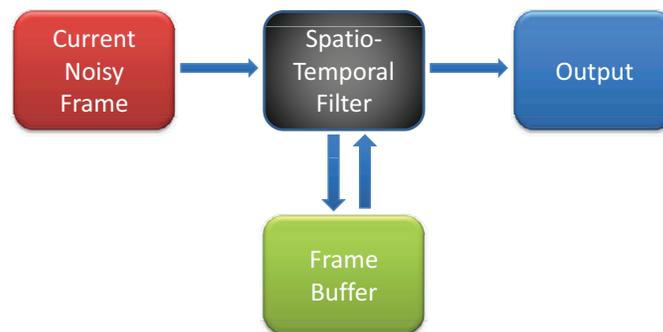


Figure 6.17 : Scheme of a temporal filter.

trated how noise levels change along the image processing pipeline; this theory is fundamental to devise an imaging pipeline that is capable to reduce noise and keep its level low from the Bayer domain to the final color image.

Bibliography

- [1] R. Gonzalez and R. Woods, *Digital Image Processing 2nd Edition*. Prentice Hall, 2002.
- [2] J. Nakamura, *Image Sensors and Signal Processing for Digital Still Cameras*. Boca Raton, FL, USA: CRC Press, Inc., 2005.
- [3] A. C. Bovik, *Handbook of Image and Video Processing*. Orlando, FL, USA: Academic Press, Inc., 2000.
- [4] Hewlett-Packard Components Group, “Noise sources in CMOS image sensors,” tech. rep., Hewlett-Packard Company, January 1998.
- [5] X. Xu, E. Miller, D. Chen, and M. Sarhadi, “Adaptive two-pass rank order filter to remove impulse noise in highly corrupted images,” *IEEE Transactions on Image Processing*, vol. 13, no. 2, pp. 238–247, 2004.
- [6] A. Bosco, A. Bruna, G. Spampinato, and G. Messina, “Fast method for noise level estimation and integrated noise reduction,” *IEEE Transactions on Consumer Electronics (ICCE-05)*, vol. 51, no. 3, pp. 1028–1033, 2005.
- [7] N. Azzabou, N. Paragios, F. Guichard, and F. Cao, “Variable bandwidth image denoising using image-based noise models,” in *Computer Vision and Pattern Recognition (CVPR-07)*, pp. 1–7, 2007.
- [8] T. Parks and K. Hirakawa, “Joint demosaicing and denoising,” in *International Conference on Image Processing (ICIP-05)*, pp. 309–312, 2005.
- [9] S. I. Olsen, “Noise variance estimation in images,” in *Proceeding 8th Scandinavian Conference on Image Analysis (SCIA-93)*, pp. 25–28, 1993.
- [10] A. Amer and E. Dubois, “Fast and reliable structure-oriented video noise estimation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, pp. 113–118, January 2005.
- [11] A. Bosco, K. Findlater, S. Battiato, and A. Castorina, “A noise reduction filter for full-frame data imaging devices,” *IEEE Transactions on Consumer Electronics*, vol. 49, no. 3, pp. 676–682, 2003.
- [12] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *International Conference on Computer Vision (ICCV-98)*, pp. 839–846, 1998.
- [13] A. Bosco, S. Battiato, M. Mancuso, and G. Spampinato, “Adaptive temporal filtering for CFA video sequences,” in *IEEE Advanced Concepts for Intelligent Vision Systems (ACIVS-02)*, pp. 19–24, 2002.
- [14] A. Bosco, S. Battiato, A. Bruna, and R. Rizzo, “Noise reduction for CFA image sensors exploiting HVS behaviour,” *Sensors*, vol. 9, no. 3, pp. 1692–1713, 2009.

- [15] A. Buades, B. Coll, and J. Morel, “The staircasing effect in neighborhood filters and its solution,” *IEEE Transaction on Image Processing*, vol. 15, no. 6, pp. 1499–1505, 2006.
- [16] M. Zhang and B. Gunturk, “A new image denoising method based on the bilateral filter,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-08)*, pp. 929–932, 2008.

Demosaicing and Aliasing Correction

M. Guarnera, G. Messina, V. Tomaselli

Advanced System Technology - Catania Lab, STMicroelectronics, Italy.

Abstract: Acquisition of color images requires the presence of different sensors for different color channels. Manufacturers reduce the cost and complexity by placing a color filter array (CFA) on top of a single image sensor, which is basically a monochromatic device, to acquire color information of the true visual scene. Since each image sensing element can detect only one color of illumination, the missing information must be filled in. The color interpolation process (also called demosaicing) aims to reconstruct the full resolution image acquired by the sensor, by calculating the missing components. Picture quality is strictly related to the peculiarity of demosaicing process. Due to the aliasing phenomenon, such as false colors and zipper effects, the color interpolation has to guarantee the rendering of high quality pictures avoiding artifacts. In this chapter we review some solutions devoted to demosaicing and antialiasing. Demosaicing algorithms can be basically divided into two main categories: spatial-domain and frequency-domain. Demosaicing solutions are not always able to completely eliminate false colors and zipper effects, thus imaging pipelines often include a post-processing module, with the aim of removing residual artifacts. Some of these techniques are also described.

7.1 Introduction

The most common arrangement of spectrally selective filters is known as Bayer pattern [1]. Since each image sensing element can only detect one color of illumination, the sensor provides a grayscale image, which then undergoes a color interpolation process to reconstruct the full resolution image. The demosaicing is by far the most important step of the whole image processing pipeline.

The simplest demosaicing method is the bilinear interpolation, a proper average on each pixel depending on its position in the Bayer Pattern is calculated. For a pixel, we consider its eight direct neighbors and then we determine the two missing colors of this pixel by averaging the colors of the neighboring ones. We actually have 4 different cases

R ₁₁	G ₁₂	R ₁₃	G ₁₄	R ₁₅	G ₁₆
G ₂₁	B ₂₂	G ₂₃	B ₂₄	G ₂₅	B ₂₆
R ₃₁	G ₃₂	R ₃₃	G ₃₄	R ₃₅	G ₃₆
G ₄₁	B ₄₂	G ₄₃	B ₄₄	G ₄₅	B ₄₆
R ₅₁	G ₅₂	R ₅₃	G ₅₄	R ₅₅	G ₅₆
G ₆₁	B ₆₂	G ₆₃	B ₆₄	G ₆₅	B ₆₆

Figure 7.1 : Example of Bayer pattern.

of averaging which correspond to the red pixel, the blue pixel, the green pixel on a red row and the green pixel on the blue row. On each of them, the averaging will be slightly different. Assuming the notation used in Fig.(7.1) and considering, for example, the pixels R_{33} , B_{44} , G_{43} and G_{34} , the bilinear interpolation proceeds as follows:

The interpolation on a red pixel (R_{33}) produces the *RGB* triplet as:

$$\begin{aligned} Red &= R_{33} \\ Green &= \frac{G_{23} + G_{34} + G_{32} + G_{43}}{4} \\ Blue &= \frac{B_{22} + B_{24} + B_{42} + B_{44}}{4} \end{aligned} \quad (7.1)$$

The interpolation on a blue pixel (B_{44}):

$$\begin{aligned} Red &= \frac{R_{33} + R_{35} + R_{53} + R_{55}}{4} \\ Green &= \frac{R_{33} + R_{35} + R_{53} + R_{55}}{4} \\ Blue &= B_{44} \end{aligned} \quad (7.2)$$

The interpolation on a green pixel in a blue row (G_{43}):

$$\begin{aligned} Red &= \frac{R_{33} + R_{53}}{2} \\ Green &= G_{43} \\ Blue &= \frac{B_{42} + B_{44}}{2} \end{aligned} \quad (7.3)$$

The interpolation on a green pixel in a red row (G_{34}):

$$\begin{aligned} Red &= \frac{R_{33} + R_{35}}{2} \\ Green &= G_{34} \\ Blue &= \frac{B_{24} + B_{44}}{2} \end{aligned} \quad (7.4)$$

Despite this interpolation is very simple, the results are unsatisfactory: it, as many other traditional color interpolation methods, usually results color edge artifacts in the image, due to the non-ideal sampling performed by the CFA. The term aliasing refers to the distortion that occurs when a continuous time signal is sampled at a frequency lower than twice its highest frequency. As stated in the Nyquist-Shannon sampling theorem, an analog signal that has been sampled can be perfectly reconstructed from the samples if the sampling rate exceeds $2B$ samples per second, where B is the highest frequency in the original signal. If the highest frequency in the original signal is known, this theorem gives the lower bound on sampling frequency assuring perfect reconstruction. On the other hand, if the sampling frequency is known, the Nyquist-Shannon theorem gives the upper bound to the highest frequency (called Nyquist frequency) of the signal to allow

the perfect reconstruction. In practice, neither of these two statements can be completely satisfied because they require band-limited original signals, which do not contain energy at frequencies higher than a certain bandwidth B . An example of band-limited signal is depicted in Fig.(7.2).

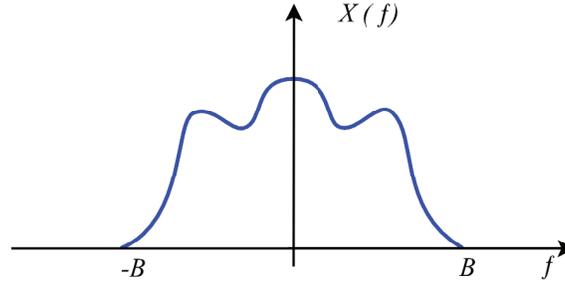


Figure 7.2 : Example of a bandlimited signal.

In real cases a "time-limited" or a "spatial-limited" signal can never be perfectly band-limited. For this reason, an anti-aliasing filter is often placed at the input of digital signal processing systems, to restrict the bandwidth of the signal to approximately satisfy the sampling theorem. In case of any imaging devices an optical low pass filter smoothes the signal in the spatial optical domain in order to reduce the resolution below the limit of the digital sensor, which is strictly related to the pixel pitch, X_s , which is the distance between two adjacent pixels. As explained in [2], the sampling frequency of the sensor is

$$f_s = \frac{1}{X_s} \quad (7.5)$$

To reproduce a spatial frequency there must be a pair of pixels for each cycle. One pixel is required to respond to the black half cycle and one pixel is required to respond to the white half cycle. In other words one pixel can only represent a half signal cycle, and hence the highest frequency the array can reproduce is half its sampling frequency and it is called Nyquist frequency:

$$f_N = \frac{1}{2X_s} \quad (7.6)$$

In a two dimensional array, having the same pixel pitch in both directions, Nyquist and sampling frequencies are equal in both X and Y axes. If a Bayer color filter array is applied on the sensor surface, the Nyquist and the sampling frequencies are different for the G channel and the R/B channels. Red and blue channels have the same pattern, so they have the same Nyquist frequency. In particular, let p be the monochrome pixel pitch; as noticeable from Fig.(7.3(a)) the red and blue horizontal and vertical pixel pitch X_s is

$$X_s = 2p \quad (7.7)$$

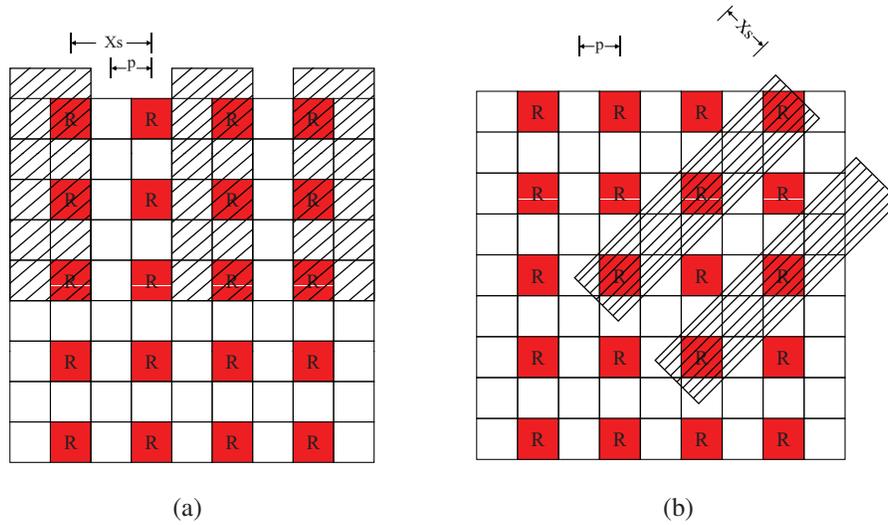


Figure 7.3 : Red array line pairs at the Nyquist frequency.

and hence the Nyquist frequency for red and blue channels in both horizontal and vertical directions is

$$f_{NR_{hv}} = \frac{1}{4p} \quad (7.8)$$

Looking at Fig.(7.3(b)), is possible to derive the diagonal spacing between two adjacent red/blue pixels:

$$X_s = \sqrt{2}p \quad (7.9)$$

the diagonal Nyquist frequency becomes:

$$f_{NR_{d}} = \frac{\sqrt{2}}{4p} \quad (7.10)$$

This means that the diagonal Nyquist frequency is larger than the horizontal/vertical one by a $\sqrt{2}$ factor.

As far as the green channel is concerned, the horizontal and vertical pixel pitch equals the monochrome pixel pitch (see Fig.(7.4)), and hence its Nyquist frequency equals that of the monochrome array:

$$f_{NG_{hv}} = \frac{1}{2p} \quad (7.11)$$

The diagonal Nyquist frequency, instead, equals that of the red/blue channels, which has been already shown in (7.10).

From this analysis it is easily derivable that red and blue channels are more affected by aliasing effects than the green channel. Despite the application of an optical anti-aliasing filter, aliasing artifacts often arise due to the way the signal is reconstructed in terms of color interpolation.

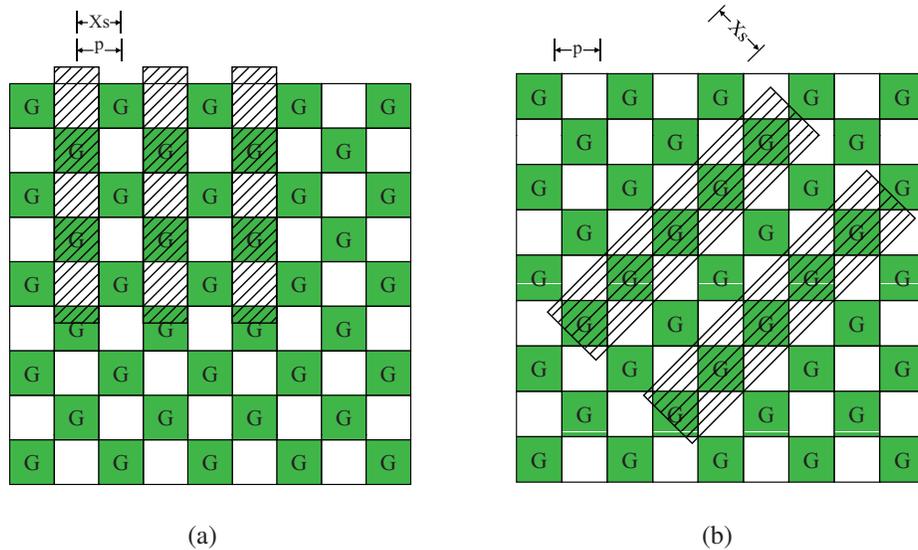


Figure 7.4 : Green array line pairs at the Nyquist frequency.

Color interpolation techniques should be implemented by considering the artifacts introduced by the sensor and the interactions with the other modules composing the image processing pipeline, as it has been well analyzed in [3]. This means that demosaicing approaches have to guarantee the rendering of high quality pictures avoiding typical artifacts, which could be emphasized by the sharpening module, thus drastically deteriorating the final image quality. In the meantime, demosaicing should avoid introducing false edge structures due to residual noise (not completely removed by the noise reduction block) or green imbalance effects. Green imbalance is a mismatch arising at G_R and G_B locations. This effect is mainly due to crosstalk [4].

In the last years a wide variety of works has been produced about color interpolation, exploiting a lot of different approaches [5]. In this Chapter we review some of the state of the art solutions devoted to demosaicing and antialiasing, paying particular attention to the patents [6].

7.2 Color Interpolation Techniques

Demosaicing solutions can be basically divided into two main categories: spatial-domain approaches and frequency-domain approaches.

7.2.1 Spatial-Domain Approaches

In this sub-section we review some recent solutions, devoted to demosaicing, which are typically fast and simple to be implemented inside a system with low capabilities (e.g., memory requirement, CPU, low-power consumption, etc.).

In the following we present techniques based on spatial and spectral correlations. With reference to Fig.(7.5), the methods, exploiting intrachannel correlation, separately process each channel of the color image (Fig.(7.5(a))). According to this principle, within a homogeneous image region, neighboring pixels share similar color values, thus a missing value can be retrieved by averaging the pixels close to it. On the other hand the majority of techniques which are based on local consideration and/or color spectral characteristics (interchannel) coming from all the color channels to interpolate each color channel, Fig.(7.5(b)).

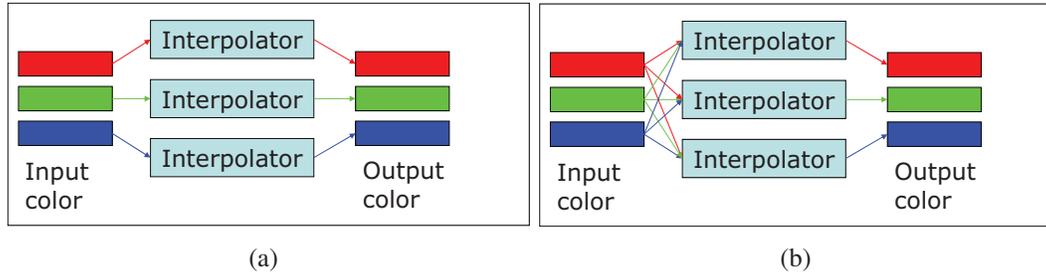


Figure 7.5 : Image processing paradigms: **(a)** Intrachannel; **(b)** Interchannel Correlation.

Spatial Correlation Based Approaches

One of the principles of color interpolation techniques is to exploit spatial correlation. According to this principle, within a homogeneous image region, neighboring pixels share similar color values, so a missing value can be retrieved by averaging the pixels close to it. In presence of edges, the spatial correlation principle can be exploited by interpolating along edges and not across them. Techniques which disregard directional information often produce images with zipper effect. Bilinear interpolation, explained in Section 7.1, belongs to this class of algorithms. On the contrary, techniques which interpolate along edges are less affected by this kind of artifact. Furthermore, averaging the pixels which are across an edge also leads to a decrease in the sharpness of the image at the edges.

Edge based color interpolation techniques are widely disclosed in literature, and they can be differentiated mainly according to the number of directions, the way adopted to choose the direction to use in the interpolation and the interpolation method.

The method in [7] discloses a technique which firstly interpolates the green color plane, then interpolates the remaining two planes. A missing G pixel can be interpolated horizontally, vertically or by using all the four samples around it. With reference to the neighborhood of Fig.(7.6) the interpolation direction is chosen through two values:

$$\Delta H = |-A3 + 2 \cdot A5 - A7| + |G4 - G6| \quad (7.12)$$

$$\Delta V = |-A1 + 2 \cdot A5 - A9| + |G2 - G8| \quad (7.13)$$

which are composed of Laplacian second-order terms for the chroma data and gradients for the green data, where the A_i can be either R or B .

Once the G color plane is interpolated, R and B at G locations are interpolated. In particular, a horizontal predictor is used if their nearest neighbors are in the same row, whereas a vertical predictor is used if their nearest neighbors are in the same column. Finally, R is interpolated at B locations and B is interpolated at R locations.

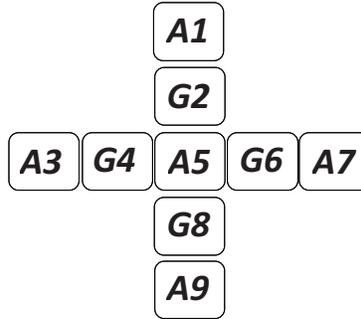


Figure 7.6 : Considered neighborhood.

Although the interpolation is not just an average of the neighboring pixels, wrong color can be introduced near edges. To improve the performances, in [8] a control factor of the Laplacian correction term is introduced. This control mechanism allows increasing the sharpness of the image, reducing at the same time wrong colors and ringing effects near edges. In particular, if the Laplacian correction term is greater than a predefined threshold, it is changed by calculating an attenuating gain, which depends on the minimum and maximum values of the G channel and of another color channel. A drawback of these methods is that G can be interpolated only in horizontal and vertical directions; R and B can be interpolated only in diagonal directions (in case of B and R central pixel) or in horizontal and vertical directions (in case of G central pixel).

The approach proposed in [9], similarly to the previous one, interpolates the missing G values in either horizontal or vertical direction, and chooses the direction depending on the intensity variations within the observation window. The variation filters, shown in Fig.(7.7), take into account both G and non- G intensity values. In this case, the interpolation of G values is achieved through a simple average of the neighboring pixels in the chosen direction, but the quality of the image is improved by applying a sharpening filter. One important peculiarity of this method is the $G_R - G_B$ mismatch compensator

$$[1 \quad -1 \quad -2 \quad 1 \quad 1] \quad \begin{bmatrix} 1 \\ -1 \\ -2 \\ 1 \\ 1 \end{bmatrix}$$

(a) Horizontal mask (b) Vertical mask

Figure 7.7 : Variation masks proposed in [9].

step, which tries to overcome the green imbalance issue. In some sensors the photo-sensitive elements that capture G intensity values at G_R locations can have a different response than the photosensitive elements that capture G intensity values at G_B locations. The $G_R - G_B$ mismatch module applies gradient filters and curvature filters to derive the maximum variation magnitude. If this value exceeds a predefined threshold value, the $G_R - G_B$ smoothed intensity value is selected, otherwise the original G intensity value is selected. To interpolate the missing R and B values, the color correlation is exploited. In fact, discontinuities of all the color components are assumed to be equal. Thus, color discontinuity equalization is achieved by equating the discontinuities of the remaining color components with the discontinuities of the green color component. Methods which use color correlation in addition to edge estimation usually provide higher quality images.

All the already disclosed methods propose an adaptive interpolation process in which some conditions are evaluated to decide between the horizontal and vertical interpolation. When neither a horizontal edge nor a vertical edge is identified, the interpolation is performed using an average value among surrounding pixels. This means that resolution in appearance deteriorates in the diagonal direction. Moreover, in regions near the vertical and horizontal Nyquist frequencies, the interpolation direction can abruptly change, thus resulting in unnaturalness in image quality. To overcome the above mentioned problems, the method in [10] prevents an interpolation result from being changed discontinuously with a change in the correlation direction. First of all, vertical (ΔV) and horizontal correlation values (ΔH) of a target pixel to be interpolated are calculated by using the equations in (7.12). Then, a coefficient term, depending on the direction in which the target pixel has higher correlation, is computed:

$$K = \begin{cases} 0 & \text{if } \Delta H = \Delta V \\ 1 - \frac{\Delta V}{\Delta H} & \text{if } \Delta H > \Delta V \\ \frac{\Delta H}{\Delta V} - 1 & \text{if } \Delta H < \Delta V \end{cases} \quad (7.14)$$

Thus K has values in the range $[-1,1]$.

The K coefficient is used to weight the interpolation data in the vertical or horizontal direction with the interpolation data in the diagonal direction. If K has a positive value ($\Delta V < \Delta H$), that is a vertical edge is found, a weighted average of the vertical interpolated value (V_{value}) and the two-dimensional interpolated value ($2D_{value}$) is calculated using the (7.15), where Ka is the absolute value of the coefficient K .

$$Out\ put = V_{value} \times Ka + 2D_{value} \times (1 - Ka) \quad (7.15)$$

Obviously, if K is a negative value a weighted average of the horizontal interpolated value and the two-dimensional interpolated value is computed. As a result, a proportion of either the vertical or horizontal direction interpolation data can be continuously changed without causing a discontinuous change in interpolation result when the correlation direction changes.

The approach proposed in [11] is composed by an interpolation step followed by a correction step. The authors consider the luminance channel as proxy for G color, and the chrominance channel as proxy for R and B . Since the luminance channel is more accurate, it is interpolated before the chrominance channels. The luminance is interpolated as accurate as possible in order to not produce wrong modifications in the chrominance channels. However, after the interpolation step, luminance and chrominances are orderly refined. The interpolation phase is based on the analysis of the gradients in four directions (east, west, north and south), defined as follows:

$$\begin{aligned}\Delta W &= \left| 2L_{(x-1,y)} - L_{(x-3,y)} - L_{(x+1,y)} \right| + \left| C_{(x,y)} - C_{(x-2,y)} \right| \\ \Delta E &= \left| 2L_{(x+1,y)} - L_{(x-1,y)} - L_{(x+3,y)} \right| + \left| C_{(x,y)} - C_{(x+2,y)} \right| \\ \Delta N &= \left| 2L_{(x,y-1)} - L_{(x,y-3)} - L_{(x,y+1)} \right| + \left| C_{(x,y)} - C_{(x,y-2)} \right| \\ \Delta S &= \left| 2L_{(x,y+1)} - L_{(x,y-1)} - L_{(x,y+3)} \right| + \left| C_{(x,y)} - C_{(x,y+2)} \right|\end{aligned}\quad (7.16)$$

Since the aim is to interpolate along edges and not across them, an inverted gradient function is formed:

$$f_{grad}(x) = \begin{cases} \frac{1}{x} & \text{if } x \neq 0 \\ 1 & \text{if } x = 0 \end{cases} \quad (7.17)$$

where x represents one of the gradient of (7.16). This function allows to weight more the smallest gradients and to follow the edge orientation. The interpolation of missing luminance values is performed using the normalized inverted gradient functions which weight both luminance and chrominance values in the neighborhood. The chrominance values are used in the interpolation of luminance to get a more accurate estimation. Similarly, chrominances are interpolated by using both luminance and chrominance data. The correction step comprises the luminance correction first, and then the chrominance correction.

The method in [12] aims to generate images with sharp edges. And also in this case a high frequency component, derived from the sensed color channel, is added to the low frequency component of the interpolated channels. This technique takes into account eight different directions, as it shown in Fig.(7.8), and uses 5×5 elliptical Gaussian filters to interpolate the low frequency component of each color channel (even the sensed one). For each available direction there is a different Gaussian filter, having the greater coefficients along the identified direction. These filters have the advantage of interpolating the missing information without generating annoying jaggy edges.

After having computed the low frequency component, for each color channel, an enhancement of the high frequencies content is obtained taking into account the color correlation (7.22). In particular, a correction term is calculated as the difference between the original sensed value and its low pass component, as it is retrieved through the directional Gaussian interpolation:

$$\Delta_{Peak} = G - G_{LPF} \quad (7.18)$$

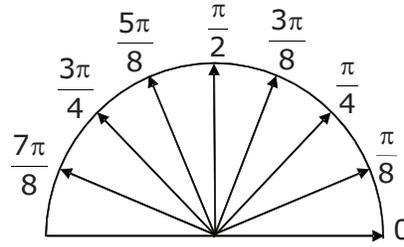


Figure 7.8 : Quantized directions for spatial gradients.

This correction term is then added to the low frequency component of the channels to be estimated:

$$H = H_{LPF} + \Delta_{Peak} \quad (7.19)$$

The low frequency component, in this method, is calculated according to the identified direction, so it is less affected by false colors than previous inventions. Moreover, this solution provides a simple and effective method for calculating direction and amplitude values of spatial gradients, without making use of a first rough interpolation of the G channel. More specifically, 3×3 Sobel operators are applied directly on the Bayer pattern to calculate horizontal and vertical gradients. The orientation of the spatial gradient at each pixel location is given by the following equation:

$$or(x,y) = \begin{cases} \arctan\left(\frac{P' * Sobel_y(x,y)}{P' * Sobel_x(x,y)}\right) & \text{if } P' * Sobel_x(x,y) \neq 0 \\ \frac{\pi}{2} & \text{otherwise} \end{cases} \quad (7.20)$$

where $P' * Sobel_y$ and $P' * Sobel_x$ are the vertical and horizontal Sobel filtered values, at the same pixel location. The orientation $or(x,y)$ is quantized in eight predefined directions. Since the image could be deteriorated by noise, and the calculation of direction could be sensitive to it, a more robust estimation of direction is needed. For this reason, Sobel filters are applied on each 3×3 mask within a 5×5 window, thus retrieving nine gradient data. In addition to the orientation, the amplitude of each spatial gradient is calculated, by using the following equation:

$$mag(x,y) = (P' * Sobel_x(x,y))^2 + (P' * Sobel_y(x,y))^2 \quad (7.21)$$

The direction of the central pixel is finally derived through the “weighted-mode” operator, which provides an estimation of the predominant amplitude of the spatial gradient around the central pixel. This operator substantially reduces the effect of noise in estimating the direction to use in the interpolation phase.

Spectral Correlation Based Approaches

In this class of algorithms final RGB values are derived taking into consideration the inter-channel color correlations in a limited region (Fig.(7.5(b))). Gunturk *et al.* [13] has demonstrated that high frequency components of the three color planes are highly correlated, but not equal. This suggests that any color component can help to reconstruct the high frequencies for the remaining color components. For instance, if the central pixel is red R , the green G component can be determined as:

$$G(i, j) = G_{LPF}(i, j) + R_{HPF}(i, j) \quad (7.22)$$

where $R_{HPF}(i, j) = R(i, j) - R_{LPF}(i, j)$ is the high frequency content of the R channel, and G_{LPF} and R_{LPF} are the low frequency components if the G and R channels, respectively.

This implies that the G channel can take advantage of the R and B information. Furthermore for real world images the color difference planes ($\Delta_{GR} = G - R$ and $\Delta_{GB} = G - B$) are rather flat over small regions, and this property is widely exploited in demosaicing and antialiasing techniques. This model using channel differences (that can be viewed as chromatic information), is nearer to the Human Color Vision system that is more sensitive to chromatic changes than luminance changes in low spatial frequency regions. Like the previous example, if the central pixel is R , the green component can be derived as:

$$G = R + \Delta_{GR} \quad (7.23)$$

The method proposed in [14] belongs to this class. The technique generates by first an estimation of all color channels (R , G and B) containing the Low Frequencies (LF) only. This is obtained by taking into consideration an edge strength metric to inhibit smoothing of detected edges. Then a difference between the estimated smoothed values and the original Bayer pattern values is performed to obtain the corresponding High Frequency (HF) values. Finally the low frequency channels and the corresponding estimated high frequency planes are combined into the final RGB image. In particular the high frequency values are obtained through the relations described in Table.7.1.

Table 7.1 : Color Correlations defined in [14].

	At a Red Pixel	At a Green Pixel	At a Blue Pixel
R	R	$R_{LPF} + G - G_{LPF}$	$R_{LPF} + B - B_{LPF}$
G	$G_{LPF} + R - R_{LPF}$	G	$G_{LPF} + B - B_{LPF}$
B	$B_{LPF} + R - R_{LPF}$	$B_{LPF} + G - G_{LPF}$	B

Each smoothed LF image is formed by a two-dimensional interpolation combined with a low-pass filtering excepted for pixels that maximize the edge strength metric. For example, if the central pixel is a G pixel the four adjacent G pixels, which will be taken into consideration to estimate the edge strength, are generated by interpolation (see Fig.(7.9)). Thus the measure of edge strength E_{ij} , that is proportional to the square of the actual edge difference, is then calculated according to:

$$E_{ij} = (G_{i,j} - G_{i,j-1})^2 + (G_{i,j} - G_{i,j+1})^2 + (G_{i,j} - G_{i-1,j})^2 + (G_{i,j} - G_{i+1,j})^2 \quad (7.24)$$

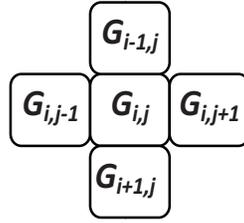


Figure 7.9 : Pattern of five pixels used to calculate an edge metric on a central G pixel of the LF (low frequency) G color channel.

By considering this edge metric the algorithm reduce the presence of color artifacts on edges boundaries.

The method [15] uses an adaptive interpolation technique for each type of Bayer Pattern pixel (R , B , green in the red row G_R and green in the blue row G_B). In particular five different interpolators are considered. Generally, to generate estimated values very close to actual pixel values it applies a nonlinear low pass filter (NLPF) that reflects the change rate of the data around the center pixel and the data of the central pixel, and by simultaneously applying a low pass filter (LPF), a band pass filter (BPF) and high pass filter (HPF) having linear characteristics, thus reducing aliasing and emphasizing high frequencies. As the process of the interpolation is strictly related to the local position on the Bayer pattern, the Table.7.2 is introduced as summary of the approach:

Table 7.2 : Color Correlation defined in [15].

	Center R	Center B	Center Gr	Center Gb
R'		(7.26)	(7.28)	(7.28)
G'	(7.25)	(7.25)		
B'	(7.26)		(7.27)	(7.27)

where the equations to take into consideration are:

$$C_{y,x} = \frac{(a_1 C_{y-1,x} + a_3 C_{y,x+1} + a_5 C_{y+1,x} + a_7 C_{y,x-1})}{(a_1 + a_3 + a_5 + a_7)} \quad (7.25)$$

$$C_{y,x} = \frac{(a_2 C_{y-1,x+1} + a_4 C_{y+1,x+1} + a_6 C_{y+1,x-1} + a_8 C_{y-1,x-1})}{(a_2 + a_4 + a_6 + a_8)} \quad (7.26)$$

$$C_{y,x} = \frac{(a_1 C_{y-1,x} + a_5 C_{y+1,x})}{(a_1 + a_5)} \quad (7.27)$$

$$C_{y,x} = \frac{(a_3 C_{y,x+1} + a_7 C_{y,x-1})}{(a_3 + a_7)} \quad (7.28)$$

where the coefficients a_i , with $i = 1, \dots, 8$, are weighting factor estimated through the distance among the central $C_{y,x}$ pixel and the surrounding values in a window of 5×5 pixels.

The usage of LPF, BPF and HPF in conjunction to the NLPF allows to reduce aliasing (at “edges”) and emphasizes the high frequencies components.

In [16] a method based on the smooth hue transition algorithms by using the color ratio rule is proposed. This rule is derived from the photometric image formation model, which assumes the color ratio is constant in an object. Each color channel is composed of the Albedo multiplied by the projection of the surface normal onto the light source direction. The Albedo is the fraction of incident light that is reflected by the surface, and is function of the wavelength (is different for each color channel) in a Lambertian surface (or even a more complicate Mondrian). The Albedo is constant in a surface, then the color channel ratio is hold true within the object region. This class of algorithms, instead of using inter-channel differences, calculates the green channel using a well-known interpolation algorithm (i.e., bilinear or bicubic), and then computes the other channels using the red to green and blue to green ratios, defined as:

$$H_b = \frac{B}{G} \text{ and } H_r = \frac{R}{G}. \tag{7.29}$$

An example of such method is described in [17]. In this work the Bayer data are properly processed by a LPF circuit and an adaptive interpolation module. The LPF module cuts off the higher frequency components of the respective color signals R , G and B and supplies R_{LPF} , G_{LPF} and B_{LPF} . On the other hand, the adaptive interpolation circuit calculates a local pixel correlation from the color signals R and G and executes interpolation with a pixel which maximizes the correlation to obtain a high resolution luminance signal.

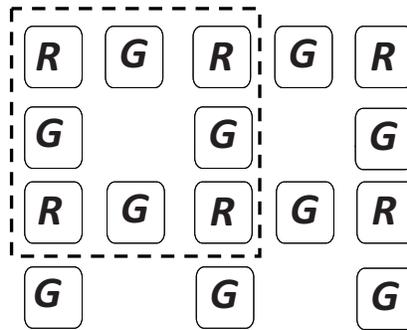


Figure 7.10 : RG pixel map for luminance interpolation.

The authors assume that, since the color signals R and G have been adjusted by the white balance module, they have almost identical signal levels and thus they can be considered as luminance signals. Taking into consideration the Bayer pattern selected in Fig.(7.10), they consider the luminance signals arranged as shown in Fig.(7.11), where the value Y_5 has to be calculated according to the surrounding values.

The correlation S for a set of pixels Y_n along a particular direction can be defined, similarly to the (7.29), as follows:

$$S = \frac{\min(Y_n)}{\max(Y_n)} \tag{7.30}$$

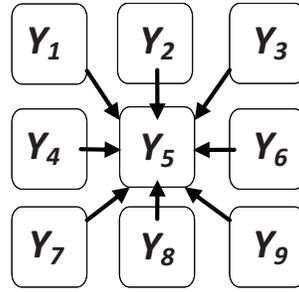


Figure 7.11 : Luminance map of analyzed signal.

where $S \leq 1$ and the maximum correlation is obtained when $S = 1$.

The correlation is calculated for the horizontal, vertical and diagonal directions, and interpolation is executed in a direction which maximizes the correlation. For instance, for the vertical direction:

$$\min(Y_n) = \min(Y_1, Y_4, Y_7) \cdot \min(Y_2, Y_8) \cdot \min(Y_3, Y_6, Y_9) \quad (7.31)$$

and

$$\max(Y_n) = \max(Y_1, Y_4, Y_7) \cdot \max(Y_2, Y_8) \cdot \max(Y_3, Y_6, Y_9) \quad (7.32)$$

The correlations in the horizontal and diagonal directions are computed in a similar way. If the direction which maximizes the correlation is the vertical one, the interpolation is executed as follows:

$$Y_5 = \frac{(Y_2 + Y_8)}{2} \quad (7.33)$$

Another way to decide the direction is to consider the similarities between the pixels. The dispersion degree σ_R of the color R is calculated as:

$$\sigma_R = \frac{\min(R_1, R_2, R_3, R_4)}{\max(R_1, R_2, R_3, R_4)} \quad (7.34)$$

If the dispersion degree is greater than a threshold, interpolation along a diagonal direction is executed. On the contrary, when the dispersion degree is small, correlation of the color R is almost identical in any directions, so it is possible to interpolate only G along the vertical or horizontal direction. This implies that the interpolation is executed only with G having the highest frequency, thus enabling to obtain an image of a higher resolution.

Once the luminance signal Y is interpolated, a high pass filter (HPF) is applied to Y and the color signal R and G . The HPF creates a luminance signal Y_{HPF} containing higher frequency components only. Finally, an adder combines the already computed color signals R_{LPF} , G_{LPF} and B_{LPF} with the higher frequency component luminance signal Y_{HPF} .

Non Adaptive Approaches

The pattern based interpolation techniques perform, generally, a statistical analysis, by collecting actual comparisons of image samples with the corresponding full-color images. Chen *et al.* [18] propose a method to improve the sharpness and reduce the color fringes with a limited hardware cost. The approach consists of two main steps:

1. Data training phase:
 - (a) Collecting samples and corresponding full-color images;
 - (b) Forming pattern indexes, by selecting the concentrative window for each color in the Bayer samples and quantizing all the values on the window;
 - (c) Calculating the errors between the reconstructed pixels and the actual color values;
 - (d) Estimating the optimal combination of pattern indexes to be sorted into a database.
2. Data practice phase:
 - (a) For each pixel a concentrative window is chosen, and within it, the pixels are quantized in two levels (Low, High) to form a pattern index, as shown in Fig.(7.12). This index is then used as key for the database matching.

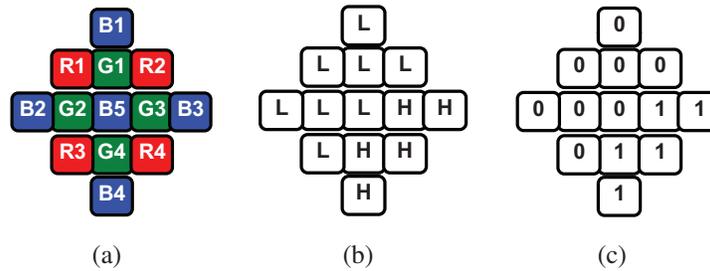


Figure 7.12 : Relationship between a color filter array and a concentrative window. (a) Bayer Pattern, (b) Quantization of acquired samples in two levels: Low (L) and High (H), (c) Resulting pattern index.

During the data training phase, the proposed method assumes that the reconstructed value (Rec_{value}) is function of the original value ($Orig_{value}$) and the feasible coefficient set ($feasible_coefficient_set$), which can be expressed as:

$$Rec_{value} = \frac{Orig_{value} * feasible_coefficient_set}{(sum_of_coefficients)} \quad (7.35)$$

Once the value has been calculated for each $feasible_coefficient_set$, the system chooses the set having the minimal error between the calculated values and the real value.

These results are then stored into the database. During the data-practice phase, the reconstruction is based on color differences rules applied to the pixel neighborhood.

A simpler technique [19] uses a plurality of stored interpolation patterns. To select the correct interpolation pattern an analysis of the input signal is performed using gradient and uniformity estimation. In practice, by first the G channel is interpolated using the 8 stored fixed patterns (Horizontal, Vertical, the two Diagonals and the four corners). To achieve this purpose the uniformity and the gradient are estimated in the surrounding of the selected G pixel. The minimum directional data estimation $G_v(i)$ ($i \in [1..8]$), obtained through the eight fixed patterns, defines the best match with the effective direction.

For example, Fig.(7.13.(a)) shows an interpolation pattern in which low luminance pixels are arranged along the diagonal.

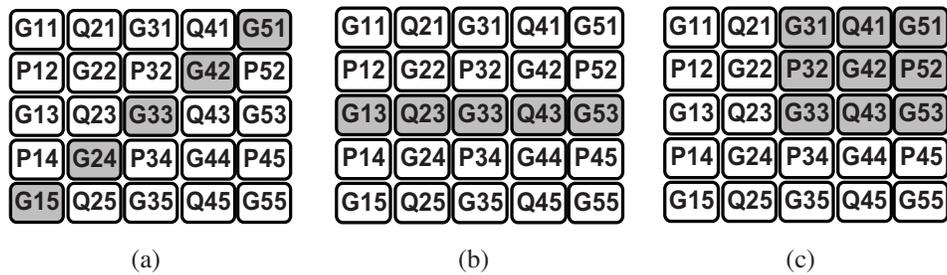


Figure 7.13 : Some samples of interpolation patterns.

The directional data $G_v(1)$, which represents a numerical value of the similarity between the surround of the pixel to be interpolated and the interpolation pattern, is obtained through the following expression:

$$G_v(1) = \frac{|G_{33} - G_{51}| + |G_{33} - G_{42}| + |G_{33} - G_{24}| + |G_{33} - G_{15}|}{4} \quad (7.36)$$

The remaining seven directional data are calculated in a similar manner, taking into account the fixed direction. The smallest directional data from $G_v(1)$ to $G_v(8)$ identifies the interpolation pattern which is the best fit to the image neighborhood of the pixel to be interpolated.

When one interpolation pattern only is present, providing the smallest directional value, it is chosen to perform the interpolation. On the contrary, when two or more interpolation patterns provide the smallest directional value, a correlation with the interpolation patterns of the surrounding pixels, whose optimum interpolation pattern has already been determined, is considered.

Specifically, if one of the interpolation patterns having the smallest value is the interpolation pattern of one surrounding G pixel, this pattern is chosen for performing the interpolation. Otherwise it is impossible to determine a specific pattern to use for the interpolation, and thus a simple low pass filter is applied.

If $G_v(1)$ is the smallest directional value:

$$G_0 = \frac{G_{15} + 2G_{24} + 2G_{33} + 2G_{42} + G_{51}}{8} \quad (7.37)$$

$$P_0 = \frac{P_{14} + P_{34} + P_{32} + P_{52}}{4} \quad (7.38)$$

$$Q_0 = \frac{Q_{23} + Q_{25} + Q_{43} + Q_{41}}{4} \quad (7.39)$$

where P and Q represent the R and B or B and R values.

If it is impossible to determine a specific pattern:

$$G_0 = \frac{(G_{22} + G_{24} + G_{42} + G_{44} + 4G_{33})}{8} \quad (7.40)$$

$$P_0 = \frac{(P_{34} + P_{32})}{2} \quad (7.41)$$

$$Q_0 = \frac{(Q_{23} + Q_{43})}{2} \quad (7.42)$$

Once the missing values for the G pixels have been processed, the algorithm calculates the missing values for the R and B pixels. If the interpolation patterns, estimated for the already processed G pixels, describe a fixed direction in the surrounding of the R/B pixel (that is several patterns indicate the same direction) then this pattern is used to perform the interpolation. Otherwise the numerical directional data are estimated. Like the G case, eight different interpolation patterns are stored in the interpolation storage memory and a directional data value is computed for each of these patterns. When there are two or more patterns having the smallest directional data value, correlations with the interpolation patterns of the already interpolated G pixels are evaluated. The reason why G pixels are taken into consideration instead of R and B pixels is that G pixels are more suitable for pattern detection than R and B pixels.

This class of techniques is very robust to noise, because it takes into consideration the interpolation patterns of the already processed pixels, but introduces jagged edges in abrupt diagonal transitions, due to the equations used in the interpolation step.

Iterative Approaches

In this category we collect all approaches that derive interpolation through an iterative process able to find after a limited number of cycles the final mosaiced image. In particular, in [20–22], starting from an initial rough estimate of the interpolation, the input data are properly filtered (usually using a combination of directional high-pass filters with some global smoothing) to converge versus stable conditions. These methods proceed in different ways with respect to the local image analysis but share the overall methodology.

In [20] a color vector image is formed containing the original Bayer values. After an initial estimate of the RGB original value for each pixel such quantity is updated by

taking into account two different functions: “roughness” and “preferred direction”. The final missing color are defined by finding the values that minimize a weighted sum of *Rough* and *CCF* (Color Compatibility Function) functions over the image by using the following formula:

$$Q = \sum_{(m,n)} \text{Rough}(m,n) + \lambda \sum_{(m,n)} \text{CCF}(m,n) \quad (7.43)$$

where λ is a positive constant while *Rough*(m,n) is defined in this case as the local summation of approximated local gradients and *CCF*(m,n) is a function that penalizes local abruptly changes. By using the classic Gauss-Siedel approach the method converges after 4-5 iterations.

In [21] and [22] the luminance channel is properly extracted from input Bayer data and analyzed in a multiscale framework by applying smoothing filtering along preferred directions. Chrominance components are smoothed by isotropically smoothing filters. The final interpolated image is obtained after a few iterations. Just before to start a new iteration the pixel values are reset to the original (measured) values.

7.2.2 Frequency-domain Approaches

Demosaicing is an ill posed problem and thus it cannot have a unique solution. This can be easily understood by considering that different real images can have the same mosaiced representation [23]. The mosaicing operation cannot be inverted and thus, it necessary to consider a priori assumptions to extrapolate the missing information. All the demosaicing algorithms use specific a priori assumptions to design the interpolator operator. One of the a priori assumption is the band limited of image signal and the limit is due to the sampling rate of the color channels.

In natural images, the energy spectrum is primarily present in a low frequency region and high frequencies along the horizontal and vertical axes [24], and the human visual system is more sensitive to these high frequencies than to the ones present at the corner of the spectrum. The demosaicing algorithms in the frequency domain exploit these band limit assumptions.

Fourier Transform Analysis and Processing

Several demosaicing algorithms in the Fourier domain have been proposed in literature [25–27] exploiting the spectrum properties of the CFA mosaiced images. The spectral representation of a CFA image can be directly derived from its representation in the spatial domain.

A color image I can be represented as:

$$I(x,y) = \{C_i(x,y)\}, i \in \{R, G, B\}, (x,y) \in \mathbb{N}^2 \quad (7.44)$$

where C_i are the color vectors in the lattice (x,y) . Thus an image is expressed as a vector of three dimensions for each pixel. The color triplets C_i form a linear vector space of

three dimensions. If we call I_{CFA} the spatial multiplexed version of the image I with a CFA pattern, we have:

$$I_{CFA}(x,y) = \sum_{i \in \{R,G,B\}} C_i(x,y) \cdot D_i(x,y) \quad (7.45)$$

where $D_i(x,y)$ are the sampling functions that have value 1 if the color channel is present at the location (x,y) , or 0 if not present. In case of the Bayer arrangement of CFA, the D_i represent the disjoint shifted lattices and can be expressed in terms of cosine modulation:

$$\begin{aligned} D_R(x,y) &= \frac{1}{4} (1 + \cos(\pi x)) (1 + \cos(\pi y)) \\ D_G(x,y) &= \frac{1}{2} (1 - \cos(\pi x) \cos(\pi y)) \\ D_B(x,y) &= \frac{1}{4} (1 - \cos(\pi x)) (1 - \cos(\pi y)) \end{aligned} \quad (7.46)$$

The mosaiced image I_{CFA} in the Fourier domain is the Fourier transform of the (7.45):

$$\hat{I}_{CFA}(u,v) = \sum_{i \in \{R,G,B\}} \hat{C}_i(u,v) * \hat{D}_i(u,v) = \hat{R}_{CFA}(u,v) + \hat{G}_{CFA}(u,v) + \hat{B}_{CFA}(u,v) \quad (7.47)$$

where $*$ denotes the convolution operator, the $\hat{\cdot}$ represents the Fourier Transform, \hat{R}_{CFA} , \hat{G}_{CFA} and \hat{B}_{CFA} are Fourier Transform of the sub-sampled color components. The modulation functions defined in (7.46) are based on cosine and have their Fourier Transform expressed in Dirac. These transforms can be compactly arranged in a matricial form:

$$\hat{D}_i(u,v) = \Delta(u)^T M_{3 \times 3} \Delta(v) \quad (7.48)$$

where

$$\Delta(u) = [\delta(u+0.5) \quad \delta(u) \quad \delta(u-0.5)]^T$$

and

$$\Delta(v) = [\delta(v+0.5) \quad \delta(v) \quad \delta(v-0.5)]^T$$

As expressed in (7.47), the Fourier Transform of the sub-sampled color channels can be derived by convolving the original \hat{C}_i channels with the corresponding modulation

functions $D_i(x,y)$; making the matrices of (7.48) explicit:

$$\begin{aligned}
\hat{R}_{CFA}(u,v) &= \hat{C}_R(u,v) * \left(\Delta(u)^T \begin{bmatrix} -\frac{1}{16} & \frac{1}{8} & -\frac{1}{16} \\ -\frac{1}{8} & \frac{1}{4} & -\frac{1}{8} \\ -\frac{1}{16} & \frac{1}{8} & -\frac{1}{16} \end{bmatrix} \Delta(v) \right) \\
\hat{G}_{CFA}(u,v) &= \hat{C}_G(u,v) * \left(\Delta(u)^T \begin{bmatrix} \frac{1}{8} & 0 & \frac{1}{8} \\ 0 & \frac{1}{2} & 0 \\ \frac{1}{8} & 0 & \frac{1}{8} \end{bmatrix} \Delta(v) \right) \\
\hat{B}_{CFA}(u,v) &= \hat{C}_B(u,v) * \left(\Delta(u)^T \begin{bmatrix} -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \end{bmatrix} \Delta(v) \right)
\end{aligned} \tag{7.49}$$

This matrix representation is useful because it clarifies how the samples are scaled replications of the Fourier transform of the full resolution channels. The \hat{G}_{CFA} formula in (7.49) points that the replications are placed on the diagonal directions only, while the \hat{R}_{CFA} and \hat{B}_{CFA} have replications also on the horizontal and vertical directions. The Fig.(7.14) shows the Fourier transform of the sub-sampled green channel and of the red and blue channels. In Fig.(7.14(b)) the spectrum of the whole I_{CFA} is shown. It is evident the overlapping among the base band and the shifted replication, that is the cause of color artifacts.

To overcome the aforementioned overlapping, Alleysson [28] started from the commonality between the human visual system (HVS) and the CFA based image sensors to sample one color only in each location (that is a pixels for imaging devices, a cone or a rod for the human eye) and thus spatial and chromatic information is mixed together. It is also known that the HVS encode the color information into luminance and opponent color signals. Similarly, for CFA sensors each color sample is composed by a spatial information due its position and chromatic information due to its spectral sensitivity. According to this representation, the (7.44) can be rewritten as:

$$I(x,y) = \{C_i(x,y)\} = \phi(x,y) + \{\psi_i(x,y)\} = \sum_{i \in \{R,G,B\}} p_i \cdot C_i(x,y) + \{\psi_i(x,y)\} \tag{7.50}$$

The spatial information, expressed by the scalar term ϕ , is composed by a weighted sum of each color channel, while $\{\psi_i\}$ is a vector of three opponent color components. Subtracting the luminance to the color image, the chrominance information is obtained. For CFA images the modulation functions can be rewritten as composed by a constant part p_i and by a fluctuation part with null mean \tilde{D}_i :

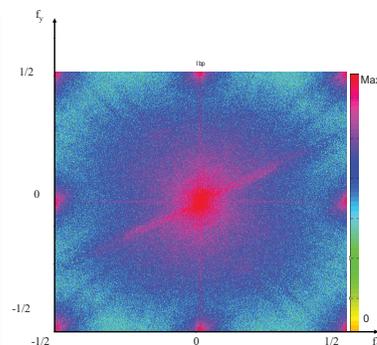
$$D_i(x,y) = p_i + \tilde{D}_i(x,y) \tag{7.51}$$

The p_i represent the probability of presence of each color channel in the CFA. Since in the Bayer pattern the green components are twice the red and blue pixels, then $p_R = \frac{1}{4}$, $p_G = \frac{1}{2}$ and $p_B = \frac{1}{4}$. According to the (7.45) the I_{CFA} is now:

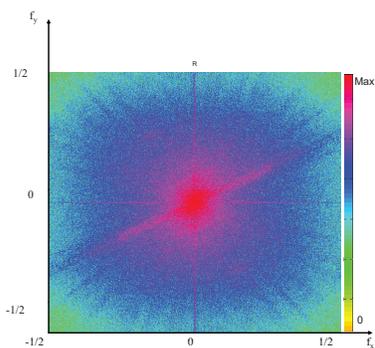
$$I_{CFA}(x,y) = \sum_{i \in \{R,G,B\}} p_i \cdot C_i(x,y) + \sum_{i \in \{R,G,B\}} C_i(x,y) \cdot \tilde{D}_i(x,y) \tag{7.52}$$



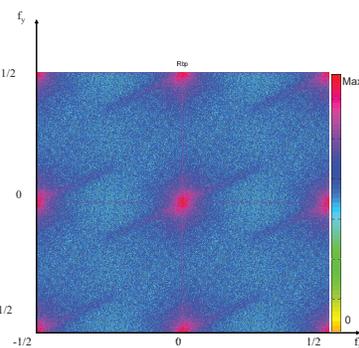
(a) Input image.



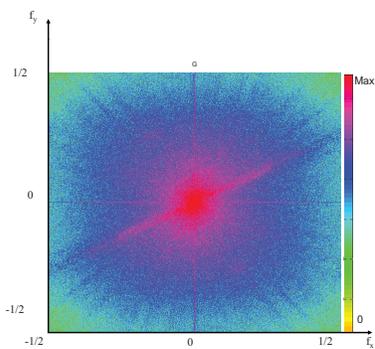
(b) Global CFA Spectrum.



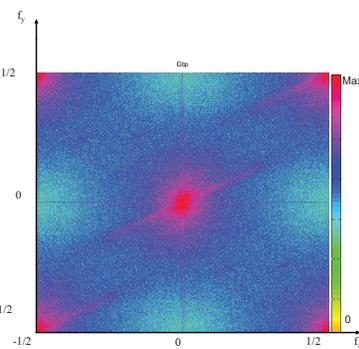
(c) RB original Spectrum.



(d) RB CFA Spectrum.



(e) G original Spectrum.



(f) G CFA Spectrum.

Figure 7.14 : Example of spectrum with relative channels.

The first term represents the luminance and the second vectorial term represents the chromatic components. this representation highlights how the luminance term in (7.52) is the same of (7.50), that is the luminance in CFA images is exactly present and not subjected to interpolation even if it is subjected to aliasing with chrominances. Thus a good estimation of luminance information is fundamental. The localization of luminance is performed on the Fourier domain. Exploding the (7.47) with the equations in (7.49), it easily to rewrite \hat{I}_{CFA} as:

$$\begin{aligned} \hat{I}_{CFA}(u, v) &= \sum_{i \in [R, G, B]} p_i \cdot \hat{C}_i(u, v) \\ &+ \frac{1}{8} \sum_{k \in \{-0.5, 0.5\}} \sum_{l \in \{-0.5, 0.5\}} [\hat{C}_R(u-k, v-l) - \hat{C}_B(u-k, v-l)] \\ &+ \frac{1}{16} \sum_{k \in \{-0.5, 0.5\}} \sum_{l \in \{-0.5, 0.5\}} [\hat{C}_R(u-k, v-l) - 2\hat{C}_G(u-k, v-l) + \hat{C}_B(u-k, v-l)] \end{aligned} \quad (7.53)$$

If we pose:

$$\begin{aligned} \hat{L}(u, v) &= \frac{1}{4}\hat{C}_R(u, v) + \frac{1}{2}\hat{C}_G(u, v) + \frac{1}{4}\hat{C}_B(u, v) \\ \hat{C}_1(u, v) &= \frac{1}{16} \sum_{k \in \{-0.5, 0.5\}} \sum_{l \in \{-0.5, 0.5\}} [\hat{C}_R(u-k, v-l) - 2\hat{C}_G(u-k, v-l) + \hat{C}_B(u-k, v-l)] \\ \hat{C}_2(u, v) &= \frac{1}{8} \sum_{k \in \{-0.5, 0.5\}} \sum_{l \in \{-0.5, 0.5\}} [\hat{C}_R(u-k, v-l) - \hat{C}_B(u-k, v-l)] \end{aligned} \quad (7.54)$$

where $\hat{L}(u, v)$ is the luminance, $\hat{C}_1(u, v)$ and $\hat{C}_2(u, v)$ are the chrominance, the (7.53) becomes:

$$\hat{I}_{CFA}(u, v) = \hat{L}(u, v) + \hat{C}_1(u, v) + \hat{C}_2(u, v) \quad (7.55)$$

The relations in (7.54) can be expressed in matricial form:

$$\begin{bmatrix} \hat{L}(u, v) \\ \hat{C}_1(u, v) \\ \hat{C}_2(u, v) \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \\ -\frac{1}{4} & 0 & \frac{1}{4} \end{bmatrix} \cdot \begin{bmatrix} \hat{C}_R(u, v) \\ \hat{C}_G(u, v) \\ \hat{C}_B(u, v) \end{bmatrix} \quad (7.56)$$

The inverse of this matrix represents the relation between the RGB values in the CFA image and the luminance/chrominance signals in the Fourier domain.

The spectrum of $\hat{L}(u, v)$ is not shifted, $\hat{C}_1(u, v)$ is located at the corner of the spectrum, while $\hat{C}_2(u, v)$ is located at the sides of the spectrum, as shown in Fig.(7.15) and Fig.(7.16). The smoothness of the color difference channels implies a more limited band for $\hat{C}_1(u, v)$ and $\hat{C}_2(u, v)$ and, consequently, the replication are more compact and less overlapping than the R and B subsampled channels. This allows to design better performing filters to discriminate luminance from the shifted bands than in the R , G and B representation.

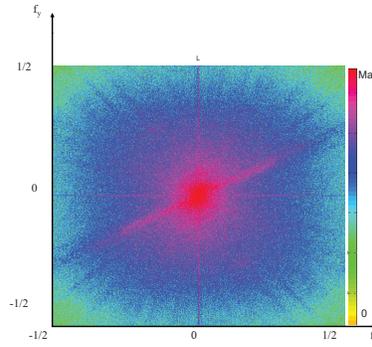


Figure 7.15 : Luminance Spectrum.

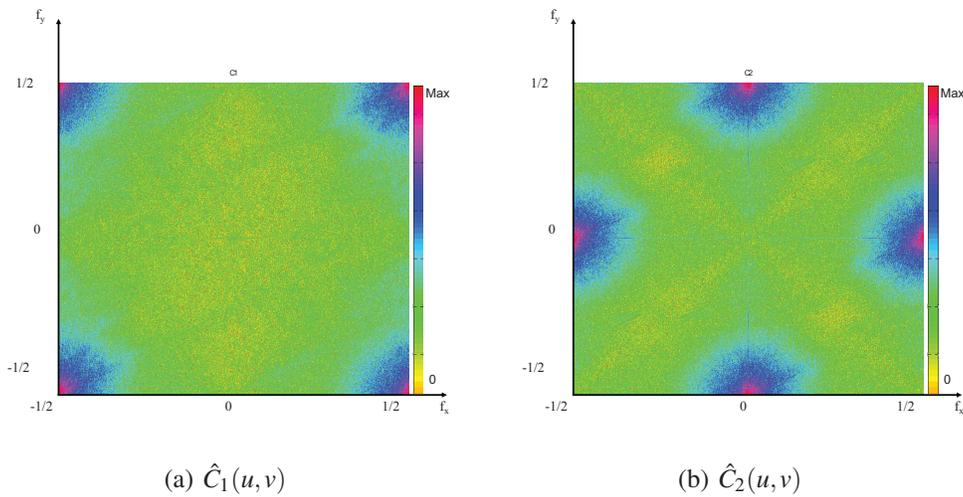


Figure 7.16 : Chrominances spectrum.

Alleyson [28] proposed the filter in Fig.(7.17) to select the luminance. This filter is able to cut the frequencies where \hat{C}_1 and \hat{C}_2 are located, leaving the luminance unchanged, as can be seen in Fig.(7.17(b)). Other studies [29,30] proposed different filters, depending on the adopted filter design methodology. The demosaicing algorithms based on this frequency analysis start estimating the luminance by filtering the \hat{I}_{CFA} image.

The luminance is then subtracted to the CFA image, obtaining the chrominance. The estimated chrominance is still subsampled and multiplexed. A further demultiplexing step separates the chrominances in three channels, containing each component color where it is defined and zeros otherwise. The final step is the interpolation to recover the missing chrominance information. The interpolation can be simple because it is applied to smooth channels. The results of this interpolation are the difference channels $R-L$, $G-L$ and $B-L$. The estimated Luminance is added to these channels to recover the R , G and B signals.

Dubois [31] proposed an alternative demosaicing approach. It is based on the initial estimation of the chromatic components C_1 and C_2 . These bands are located at the corners

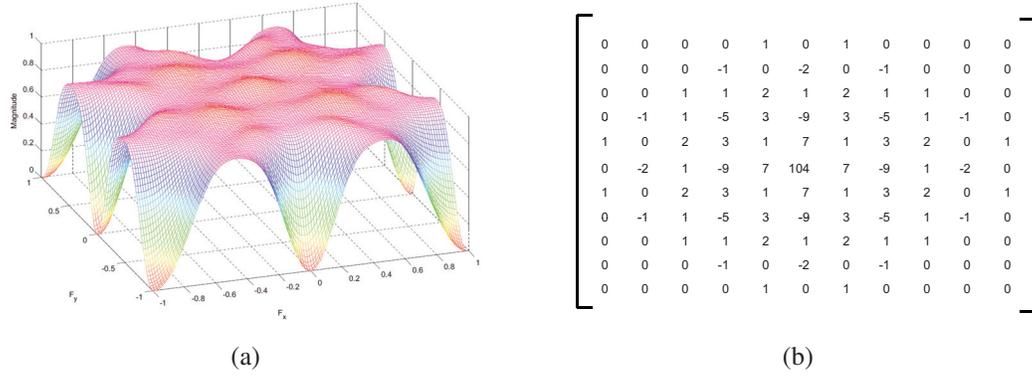


Figure 7.17 : Selection filter proposed by Alleysson [28].

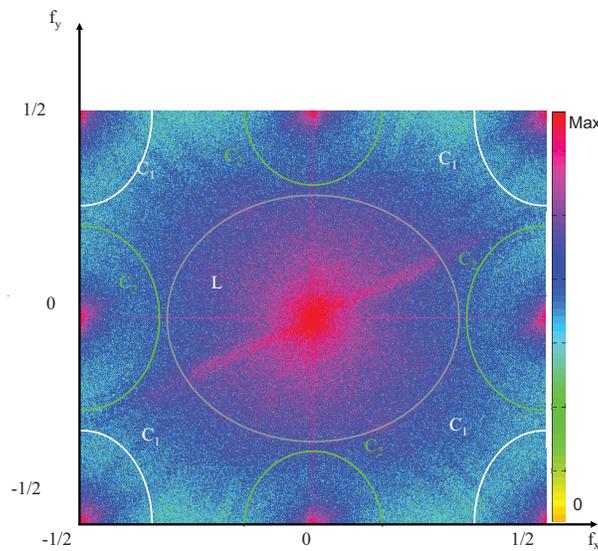


Figure 7.18 : Spectrum areas analysis.

and sides of the Fourier spectrum, and can be isolated using bandpass filters. the first one, H_1 used to estimate C_1 , is centered at frequency $(0.5, 0.5)$:

$$\hat{C}_1(u, v) = \hat{I}_{CFA}(u, v) \cdot H_1(u, v) \tag{7.57}$$

The result of this filtering is shifted in base band. Analyzing in Fig.(7.18) the I_{CFA} spectrum, is noticeable that the crosstalk in CFA images is mainly in between luminance and C_2 components. Given multiple and shifted copies of the signal C_2 , they can be exploited to better recover the original signal. I_{CFA} is filtered by two other bandpass filters H_{2A} and H_{2B} . The results are demodulated in baseband to estimate the C_{2A} and C_{2B} , the two sub-bands so that $C_2=C_{2A} + C_{2B}$, and placed on the vertical and horizontal axes of the spectrum. The more annoying artifact in demosaiced images is due to the crosstalk caused by the luminance energy near the frequencies $(\frac{1}{2}, 0)$ and $(0, \frac{1}{2})$ where the

modulated C_2 is present. In this case nothing can be done to perfectly separate the signals. However, the overlapping is often present in only one of the two bands. Leveraging on this behavior, the C_2 can be recovered as an adaptive sum of C_{2A} and C_{2B} , where the component suffering less from crosstalk is weighted more. The weighting function is modulated by a local estimation of the crosstalk, obtained analyzing the energy in two bands along the horizontal and vertical axes, the results are the local average energies e_x and e_y . The estimate of C_2 is obtained through a weighted sum:

$$C_2(x,y) = w(x,y)C_{2A}(x,y) + (1 - w(x,y))C_{2B}(x,y) \quad (7.58)$$

where

$$w = \frac{e_y}{e_x + e_y} \quad (7.59)$$

Once C_1 and C_2 have been estimated (in baseband), the luminance L is recovered by subtracting them to the I_{CFA} image. The inverse of the (7.56) matrix is at last applied to yield the R, G, B values.

Wavelets Based Algorithms

Another exploited research direction for demosaicing is in the wavelet domain [32]. In [33], to overcome the problem of quincoux G pattern, the G_r (green samples on red rows) and G_b (green samples on blue rows) are separately processed, thus the color channels to be considered are four: R , B , G_r and G_b . Each plane has a dimension half respect to the original one. The interpolation processes these four planes and thus acts as a zooming [34]. The wavelets, as shown in Fig.(7.19) represent an image into sub-bands. The LL band contains the most of energy for the image signal. Ignoring the remaining bands, it is possible to reconstruct the image, where the zooming is mainly the antitransform of wavelets. In [33] an approach to interpolate the wavelet coefficients of the other sub-bands is proposed using a local spatial analysis to estimate the missing coefficients. This approach is based on the fact that, using the DWT5/3 (the same wavelets transform used in the JPEG2000 standard), if a coefficient has a value close to zero, the corresponding image region is smooth/ homogeneous. On the opposite, if the value is high, in the corresponding image region there is great variability. Thus a simple spatial correlation estimator of the input image values p_i is considered (in both horizontal and vertical direction):

$$\begin{aligned} \Delta H &= |p_{i,j} - p_{i,j+1}| \\ \Delta V &= |p_{i,j} - p_{i+1,j}| \end{aligned} \quad (7.60)$$

Let consider the case of estimating the coefficients wav_coef_i of the HL band. If the value of ΔH is low, then the correspondent wavelets coefficient in HL is 0 because there is a high correlation between adjacent pixels. If this value is too high, there is a low correlation and the interpolation takes into consideration other pixels:

$$wav_coef_i = f(p_{i,j}, p_{i,j-1}) \quad (7.61)$$

If the value is in a predetermined range the coefficients are found using correlated pixels:

$$wav_coef_i = f(p_{i,j}, p_{i,j+1}) \quad (7.62)$$

This approach is the same for *LH* band, considering the ΔV threshold measures. The *HH* band is not interpolated due to the lower correlation of the input pixel values and the wavelets coefficients in this band. The demosaicing approach based on the coefficient interpolation is shown in Fig.(7.19).

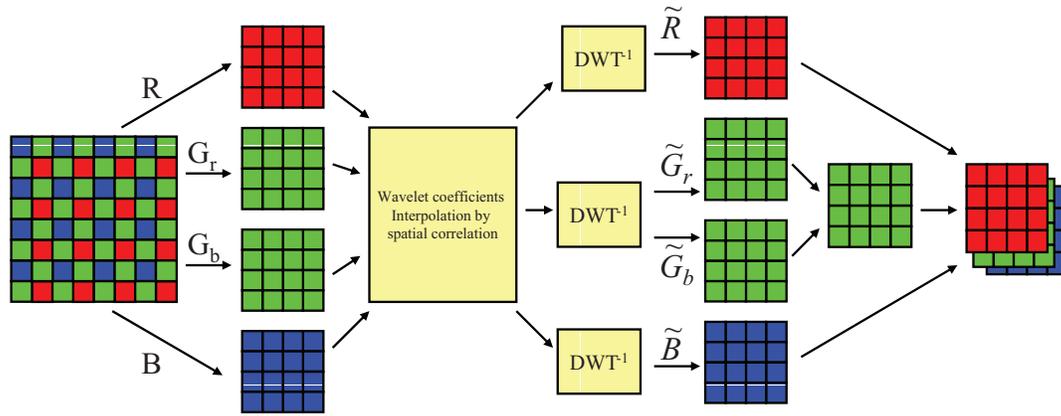


Figure 7.19 : Wavelets-based color interpolation.

This approach does not take into consideration any channel correlation, while another important demosaicing algorithm that uses the interchannel correlation has been proposed by Gunturk *et al.* [13]. Starting from the observation that on natural scenes all the three channels are very likely to have the same edge content, the authors show that the high frequencies subbands *LH*, *HL* and *LL* of each color are highly correlated. This can be expressed in the form

$$\begin{aligned} |LH(K_{i,j}) - LH(G_{i,j})| &< threshold \\ |HL(K_{i,j}) - HL(G_{i,j})| &< threshold \\ |HH(K_{i,j}) - HH(G_{i,j})| &< threshold \end{aligned} \quad (7.63)$$

where *K* is the *R* or *B* channel in the CFA image. The aliasing is removed using a Projection Onto Convex Sets (POCS) approach [35, 36]. The constraints set are based on the interchannel differences and on the observed data (original CFA pixels). This POCS method projects the initial estimate onto this constraint set to reconstruct the red and blue channels. The observation constraint set ensures that the interpolated color channels are consistent with the observed data; that is, the color samples captured by the sensor can not change during the reconstruction process. The projection onto the "observation" constraint set is performed by inserting the observed data into their corresponding locations in the color channels at each iteration. The second constraint set imposes the similarity

of high frequency of the color channels. The projection onto the "detail" constraint set is performed by first decomposing the color channels into LL , LH , HL , HH subbands and then updating the high frequency subbands of the R and B channels only if the detail subbands of the difference planes $R-G$ and $B-G$ exhibit high values and, at last, restoring them with a bank of synthesis filters. This constraint is able to drastically reduce the aliasing.

A more detailed description of the whole demosaicing algorithm is the following:

1. An initial guess of the full color image is obtained using a simple linear interpolation;
2. The green channel is updated using the high frequencies of the red/blue channels:
 - (a) The red and blue channels in the CFA image are a downsampled version of the full color image channels;
 - (b) Consider a downsampled version of the green channel corresponding to all the interpolated data in the red and blue location separately;
 - (c) Decompose the blue and corresponding green channel, then the red and corresponding green channel into subbands;
 - (d) The LH , HL and HH subband of the two green channels are replaced with the related subbands of the red and blue channels, that corresponds to set to 0 the *threshold* in the (7.63);
 - (e) Reconstruct the green channel through the inverse transform and place these new pixels in the initial guess of the green channel;
3. Iterate until a stop criterion is reached:
 - (a) Decompose the channels and update the red and blue high frequency coefficients that do not verify the (7.63);
 - (b) Reconstruct the red and blue channels, and replace the obtained values with the original ones (in the CFA image) at the red/blue location.

7.3 Post-processing Techniques for Aliasing Correction

The two main types of demosaicing artifacts are false colors and zipper effect. False colors are those artifacts corresponding to noticeable color errors as compared to the original image. One example is shown in Fig.(7.20(a)), where the left hand is the full-color original image and the right hand is the demosaiced image with false colors. The zipper effect refers to abrupt or unnatural changes of color differences between neighboring pixels, manifesting as an "on-off" pattern. One example is shown in Fig.(7.20(b)), where the left hand is the full-color image and the right hand is the demosaiced image with the zipper effect around the fence region. An explanation of how the false colors arise in color interpolation is shown in Fig.(7.21). In Fig.(7.21(a)), a graphical representation of the light intensity distribution incident to an image sensing array comprising, for example 16

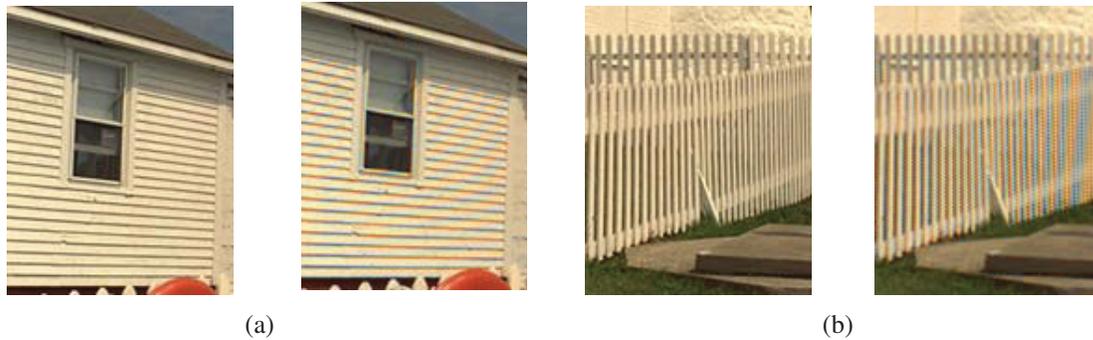


Figure 7.20 : Example of aliasing artifacts.

pixels, is depicted. For simplicity it will be assumed that the illumination comprises two colors A (filled circle) and B (square) wherein each color is defined by a selected range of wavelengths different from the selected range of wavelengths which defines the other color. As depicted in Fig.(7.21(a)), the incident illumination defines a sharp grey edge between pixels 6 and 7 and a sharp grey to color transition between pixels 12 and 13. Fig.(7.21(b)) shows the case in which the illumination incident to an image sensing array having a filter arrangement in which alternate pixels are overlapped by filter transmitting either color A or color B. Thus, each pixel receives a single color of illumination, and linear interpolation between the pixels which sample each color provides the color distribution as shown graphically in Fig.(7.21(c)). From this figure, it is evident that the pixels 6 and 7 on each side of the grey edge no longer provide equal intensities for the colors A and B and thus will provide a highly visible color artifact or fringe in the reconstructed image. From this visual example, we can derive that false colors arise if spectral correlation is not well exploited. This concept is also well disclosed in [37], where it is also explained that zipper effects manifest if spatial correlation is disregarded.

Although most of demosaicing solutions aim to eliminate false colors, some artifacts still remain. Thus imaging pipelines often include a post-processing module, with the aim of removing residual artifacts [38]. Post-processing techniques are usually more powerful in achieving false colors removal and sharpness enhancement, because their inputs are fully restored color images. Moreover, to fit some quality criteria, they can be applied more than once. For obtaining better performances, the antialiasing step often follows the color interpolation process, as a postprocessing step. The following subsections disclose a variety of state of the art techniques for false colors and zipper effects reduction.

7.3.1 False Colors Cancellation

Many techniques have been proposed in literature for reducing false colors. The conventional approach to solve this problem is to eliminate the color fringes at the expense of image sharpness by blurring the picture, so that the edges are not sharp enough to create a color fringe. Blurring the image in this manner, however, has its obvious disadvantages resulting in a reduction in resolution. Therefore it is necessary to provide a demosaicing

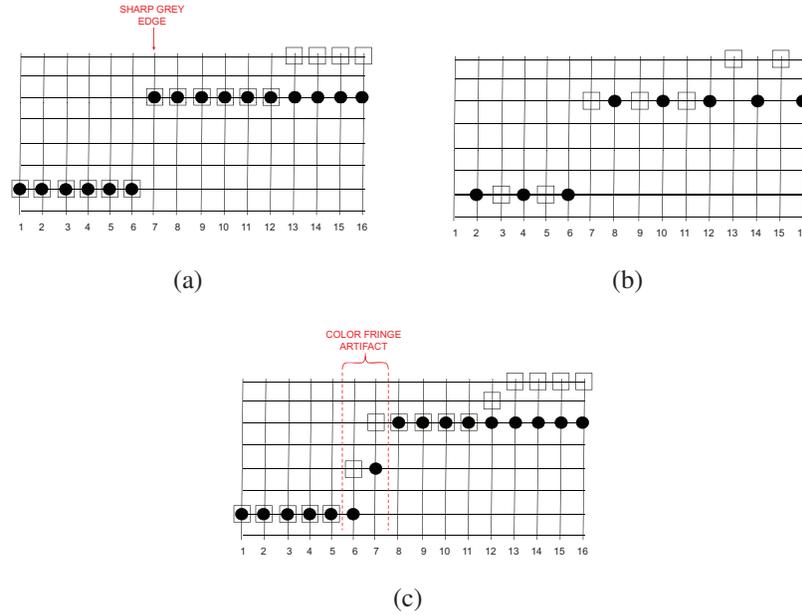


Figure 7.21 : Examples of an aliasing artifact caused by color interpolation.

artifact removal technique which reduces color fringing without the amount of blurring otherwise required. An interesting technique to solve color fringes without blurring the images was proposed by Freeman [39]. This approach starts from the consideration that in natural images there is a high correlation between the red, green and blue channel, especially for the high frequencies, so they are likely to have the same texture and edge locations. Because of this inter-channel (or spectral) correlation, the difference between two colors in a neighborhood is nearly constant, while it rapidly increases and decreases in the area of sharp grey edges, where color interpolation has introduced false colors. With reference to the example already shown in Fig.(7.21), Fig.(7.22) represents the difference between colors A and B for each pixel of Fig.(7.21(c)).

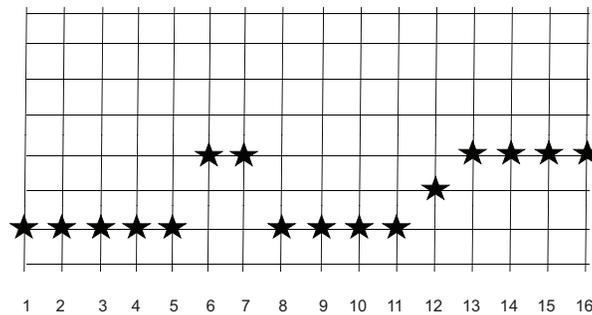


Figure 7.22 : Difference between colors A and B for each pixel.

The rapid increase and decrease in the difference between the two colors in pixels 6 and 7 is a characteristic of the objectionable color fringing and not simply a sudden rise

in the difference between colors A and B as occurs after pixel 11 and which is indicative of a change from one color to a different color. Thus, it is not desirable to create such color spikes as a result of the method of interpolation chosen. A better estimate of the actual difference between the values for the colors A and B is provided by the graphical representation of Fig.(7.23), where the sharp peaks and valleys are removed and the other sharp transitions retained. Toward this goal a median filter, with a width of N pixels, can be used to replace each value in the graph of Fig.(7.22) with the median value of the nearest N pixels. For example, if the width of the median filter is selected to be five pixels, then the value at pixel 6 will become the median value of the pixels 4, 5, 6, 7 and 8.

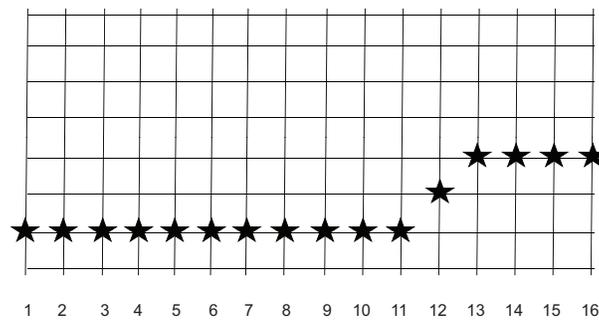


Figure 7.23 : Median filtered difference between colors A and B for each pixel.

Since the median values for each pixel are derived from the values of color A minus the values of color B for each pixel, subtracting the median values from the values of the color A provides the value of the color B for those pixels that receive only A colored light. Similarly, adding the median values to the values of the color B provides the value of the color A for those pixels that receive only B colored light. As depicted in Fig.(7.24), the Freeman's approach operates to actively reconstruct the sharp grey edge between pixels 6 and 7 while maintaining the color divergence starting at the pixel 11.

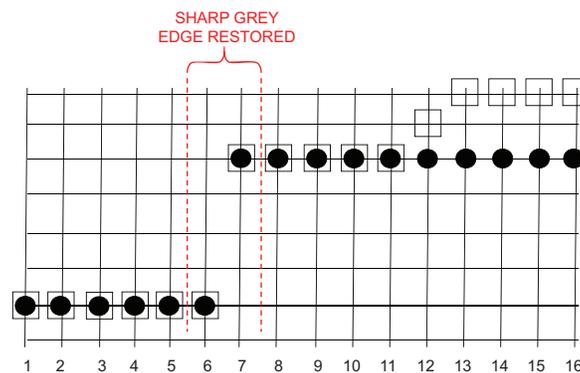


Figure 7.24 : New reconstruction from sampled data.

This method, in a three colors system, operates according to the following rules to obtain the values for the two missing colors of each pixel.

1. Pixels which receive only Red light:

$$\begin{aligned}\hat{R}(i, j) &= R(i, j) \\ \hat{G}(i, j) &= R(i, j) + v_{GR}(i, j) \\ \hat{B}(i, j) &= R(i, j) - v_{RB}(i, j)\end{aligned}\quad (7.64)$$

2. Pixels which receive only Green light:

$$\begin{aligned}\hat{R}(i, j) &= G(i, j) - v_{GR}(i, j) \\ \hat{G}(i, j) &= G(i, j) \\ \hat{B}(i, j) &= G(i, j) - v_{GB}(i, j)\end{aligned}\quad (7.65)$$

3. Pixels which receive only Blue light:

$$\begin{aligned}\hat{R}(i, j) &= B(i, j) + v_{RB}(i, j) \\ \hat{G}(i, j) &= B(i, j) + v_{GB}(i, j) \\ \hat{B}(i, j) &= B(i, j)\end{aligned}\quad (7.66)$$

where

$$v_{CD}(i, j) = \text{median} \{C(k, l) - D(k, l) | (k, l) \in H\} \quad (7.67)$$

H denotes the support of the $N \times N$ local window centered in (i, j) , C and D denote two of the color channels. Analyzing the previous rules, is evident that the original CFA-sampled color value at each pixel is not altered, and it is combined with median-filtered inter-channel differences to obtain the other two missing color values. In general, Freeman's method is rather effective in suppressing demosaicing artifacts, while preserving sharp edges. However, some demosaicing artifacts, especially zipper effects, still remain around sharp edges and fine details. This is partly due to the fact that each pixel has independent inter-channel differences, and filtering the differences separately does not take into account the spectral correlation between color planes. To incorporate median filtering with the spectral correlation for more effective suppression of demosaicing artifacts, Lu and Tan's approach [38] lifts the constraint of keeping the original CFA-sampled color values intact. Furthermore, it makes use of the latest processed color values to filter the subsequent pixels so that estimation errors can be effectively diffused into local neighborhoods. Specifically, it adjusts the three color values at the central pixel of a local window (the window size is equal to the support of the median filter) as follows:

$$\begin{aligned}\hat{G}(i, j) &= \frac{(R(i, j) - v_{RG}(i, j)) + (B(i, j) - v_{BG}(i, j))}{2} \\ \hat{R}(i, j) &= \hat{G}(i, j) + v_{RG}(i, j) \\ \hat{B}(i, j) &= \hat{G}(i, j) + v_{BG}(i, j)\end{aligned}\quad (7.68)$$

This approach removes more false colors and artifacts than Freeman's method, but it considerably blurs images, because it adjusts the green channel of each pixel through an average of both the red and blue values of the same pixel. Another interesting technique, which is proposed in [40], updates the R, G, B values adaptively, modifying also the

original pixel value which could be corrupted, due to the effect of noise. Two updated values for the green channel are calculated using each color difference domain:

$$\begin{aligned} G^R(i, j) &= R(i, j) + v_{GR}(i, j) \\ G^B(i, j) &= B(i, j) + v_{GB}(i, j) \end{aligned} \quad (7.69)$$

where

$$\begin{aligned} v_{GR}(i, j) &= \text{median} \{G(k, l) - R(k, l) \mid (k, l) \in A\} \\ v_{GB}(i, j) &= \text{median} \{G(k, l) - B(k, l) \mid (k, l) \in A\} \end{aligned} \quad (7.70)$$

and A denotes the support of the 5×5 local window centered in (i, j) .

The updated G value is determined by the weighted sum of two updated G^R and G^B values of each color difference domain and original G value. Subsequently, R and B values are updated using the updated G value. This process is expressed as:

$$\begin{aligned} \hat{G}(i, j) &= \frac{1}{2}G(i, j) + \frac{1}{2} \{ (1 - a(i, j))G^R(i, j) + a(i, j)G^B(i, j) \} \\ \hat{R}(i, j) &= \frac{1}{2}R(i, j) + \frac{1}{2} \{ \hat{G}(i, j) - v_{GR}(i, j) \} \\ \hat{B}(i, j) &= \frac{1}{2}B(i, j) + \frac{1}{2} \{ \hat{G}(i, j) - v_{GB}(i, j) \} \end{aligned} \quad (7.71)$$

where $a(i, j)$ is a weight, expressed as:

$$a(i, j) = \frac{\sigma_{(G-R)}^2(i, j)}{\sigma_{(G-R)}^2(i, j) + \sigma_{(G-B)}^2(i, j)}, 0 < a(i, j) < 1 \quad (7.72)$$

$\sigma_{(G-R)}^2$ and $\sigma_{(G-B)}^2$ represent the variances of interchannel differences.

As it is apparent from the (7.71), the color correction algorithm proposed in [40], thanks to the variance information, weights more the flatter color difference domain than the other. Moreover, the initially interpolated value is not totally exchanged by the updated value, but it is equally weighted for correction. Subsequently, the color values of the central pixel are replaced by \hat{R} , \hat{G} and \hat{B} so that they will be involved in filtering the updating pixels.

The local statistics are effectively estimated from a running square window as follows:

$$\begin{aligned} E[A(i, j)] &= \frac{\sum_{k,l \in A} e(k, l) \cdot A(k, l)}{\sum_{k,l \in A} e(k, l)} \\ \sigma_A^2(i, j) &= \frac{\sum_{k,l \in A} e(k, l) \cdot (A(k, l) - E[A(i, j)])^2}{\sum_{k,l \in A} e(k, l)} \\ e(k, l) &= 1 - (A(i, j) - A(k, l)) \end{aligned} \quad (7.73)$$

Such technique has the disadvantage of weighting the unfiltered values together with the filtered ones, so false colors are reduced, without being completely removed.

In [41], the authors propose to exploit the original uncorrupted Bayer CFA data, present in the demosaiced image, to correct erroneous color components produced by CFA interpolation with a localized color ratio model. This technique is based on the assumption that pixels with similar hues but different intensities should exhibit similar (if

not identical) R/G and B/G color ratios. The post-processing procedure starts updating the green channel at the original R and B spatial locations as follows:

$$\hat{G}(i, j) = H(i, j) \text{mean}_{(k,l) \in \zeta} \left(\frac{G(k, l)}{H(k, l)} \right) \quad (7.74)$$

where H is the R or B central channel and $\zeta = \{(i-1, j), (i, j-1), (i, j+1), (i+1, j)\}$. In (7.74) the sampled G values are used together with the interpolated H values to obtain a local color ratio description G/R or G/B . The proposed local color ratio creates a model of the hue for the region under consideration and uses it to estimate the G component based on the original component H . It should be noted that this solution avoids extreme transitions in hue in the postprocessed images, thus reducing false colors. The second step consists in updating the R channel at B locations and the B channel at R locations, according to the 7.75

$$\hat{H}(i, j) = \hat{G}(i, j) \text{mean}_{(k,l) \in \zeta} \left(\frac{H(k, l)}{\hat{G}(k, l)} \right) \quad (7.75)$$

where $\zeta = \{(i-1, j-1), (i-1, j+1), (i+1, j-1), (i+1, j+1)\}$. The (7.75) differs from the (7.74) mainly because ζ is modified to take into account locations of the original R or B CFA data. In the last step of the algorithm R and B color values at G original locations are updated. The (7.75) is applied now using the original G values $G(i, j)$ and $\zeta = \{(i-1, j), (i, j-1), (i, j+1), (i+1, j)\}$. Moreover, two of the values $H(k, l)$ of (7.75) are original components and the other two are corrected components previously obtained using (7.75).

Some newer approaches address the problem of removing color artifacts in the YCrCb domain, instead of the classic RGB color space. In fact, if there is a strong edge in the R channel, there is usually a strong edge at the same location in the G and B channels; on the contrary, the YCrCb domain is less correlated, as demonstrated in [42]. Although edges still tend to be strong in the Y (luminance) plane, the chrominance planes (Cr and Cb) are smoother than the RGB plane, and hence they are more suitable for interpolation. The simplest way to remove color artifacts consists in correcting both the chrominance planes by simply blurring them. One liability with this approach is that there is no discrimination between false colors and genuine chrominance details. Consequently, sharp colored edges in the image begin to bleed color as the blurring becomes more aggressive. Adams *et al.* in [43] address the problem of eliminating low-frequency colored patterns, such as color Moiré, by filtering chrominances according to an activity value depending on the nearby luminance and chrominances. To remove spikes or valleys from these signals, which usually change smoothly, a median filter can be applied, instead of blurring the chrominance planes through an average filter. A median filter can remove false colors pretty well from the image edges, but it could introduce color bleeding artifacts in sharp colored edges. For this reason the technique proposed in [44] modifies chrominance values with respect to luminance and local chromatic dynamic ranges, in order to not reduce chromaticity too much in regions with uniform colors (see also [45]). The dynamic chromatic ranges (DCr and DCb) and the dynamic luminance range (DY) are evaluated in a 5x5 neighborhood of

the pixel to be corrected. For each pixel of interest, dynamic luminance and chrominance ranges may be computed as the difference between the maximum and the minimum value in the local neighborhood, as follows:

$$\begin{aligned} DY &= \max_I(Y) - \min_I(Y) \\ DCr &= \max_I(Cr) - \min_I(Cr) \\ DCb &= \max_I(Cb) - \min_I(Cb) \end{aligned} \quad (7.76)$$

where I is the local neighborhood of the central pixel. Both the dynamic chromatic and luminance ranges are used to calculate a parameter, named *CorrectionFactor*, which determines the strength of the filter on chrominances. To remove possible false colors around sharp edges, the filtering action should be strong. On the contrary, if the luminance edge is weaker than both the chrominance edges, color bleeding has to be avoided by reducing the strength of the filter. Thus, the following equation is used to calculate this parameter:

$$CorrectionFactor = \begin{cases} DY & \text{if } DY = \min_I(DY, DCr, DCb) \\ \max_I(DY, DCr, DCb) & \text{otherwise} \end{cases} \quad (7.77)$$

The *CorrectionFactor* determines the power of the false colors correction, according to the following rule:

$$\begin{aligned} Cr &= medianCr_I + f(CorrectionFactor) \cdot (originalCr_I - medianCr_I) \\ Cb &= medianCb_I + f(CorrectionFactor) \cdot (originalCb_I - medianCb_I) \end{aligned} \quad (7.78)$$

where $f(x)$ is defined as:

$$f(x) = e^{-\frac{1}{2} \left(\frac{x}{sigma} \right)^2} \quad (7.79)$$

where *sigma* is a fixed parameter. The (7.78) updates each chrominance value with a weighted average of the original chrominance (*originalCr* and *originalCb*) and the median value of the chrominance in the neighborhood (*medianCr* and *medianCb*). The weights depend on the *CorrectionFactor* parameter, through the function $f(x)$. The function $f(x)$ is the right part of a Gaussian function with expected value equal to zero and standard deviation equal to *sigma*. Fig.(7.25) illustrates the trend of the $f(x)$, for a given *sigma* value ($sigma = 10$).

The function $f(x)$ rapidly decreases as the x increases, according to the (7.79). The value of the standard deviation *sigma* determines how fast $f(x)$ approaches zero. With reference to the (7.78), low values of the *CorrectionFactor* imply a greater contribution of the original chrominance value; on the contrary, as the *CorrectionFactor* increases a higher weight is assigned to the median value. The function $f(x)$ avoids discontinuous corrections when dynamic ranges change; in fact proportions of both the original value and the median filtered value are continuously varied to form the final value. This soft-threshold methodology avoids abrupt transitions between corrected and non-corrected pixels.

Artifacts suppression can also be implemented in the frequency domain because various artifacts often occur in high-frequency components. More specifically, in [46] the

authors propose to correct color artifacts by using the high bands inter-channel correlation of the three primary colors. Each pixel is separated in its low and high frequency components; then the high frequencies of the unknown components are replaced with the high frequencies of the Bayer-known component. The low-frequency component is preserved unchanged since the low-frequency components of the color channels are less correlated. For example, for a green pixel in the location (i, j) , the green value can be decomposed as:

$$G(i, j) = G^l(i, j) + G^h(i, j) \quad (7.80)$$

where G^l and G^h denote the low and high frequency components of the G channel, respectively. R and B components at G locations are corrected according to the 7.81

$$\begin{aligned} \hat{R}(i, j) &= R^l(i, j) + G^h(i, j) \\ \hat{B}(i, j) &= B^l(i, j) + G^h(i, j) \end{aligned} \quad (7.81)$$

The correction at R and B original locations is performed in a similar way. The selection of the low-frequency components is performed using a low-pass filter while the high frequencies are calculated subtracting the low-frequency values. An effective approach is to select the low and high frequencies using a 1-D filter, so the interpolation is carried out only along the edges of the image.

7.3.2 Zipper Cancellation

Zipper effect is an artifact caused by a not correct spatial correlation exploitation. An artifact introduced by a wrong edge-estimation is usually difficult to remove in a post-processing phase, so it should be avoided during the interpolation step. Nevertheless, some techniques exist which reduce this effect. The simplest approach is the application

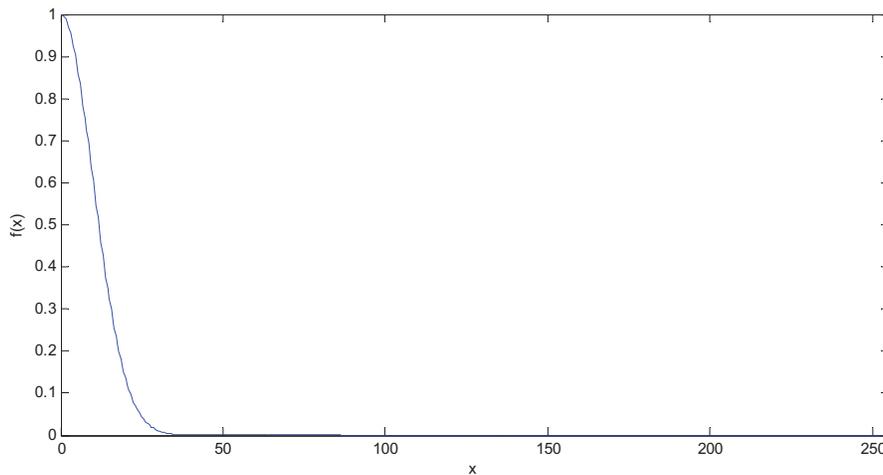


Figure 7.25 : Plot of $f(x)$ vs. x , with $\sigma = 10$.

of a heavy low pass filter to the demosaiced image. An example of antizipper filter is:

$$AZ = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad (7.82)$$

This filter removes zipper artifacts, but at the same time consistently reduces resolution thus removing genuine spatial details, which may not be recovered by further image processing algorithms.

In [47] the authors propose a method to eliminate the false color and zipper effect based on an adaptive scheme allowing to determine the specific artifact affecting the pixels. The authors use the spectral correlation between color planes to detect and reduce the artifacts. The block diagram representing the demosaicing artifact removal algorithm is shown in Fig.(7.26). Before processing each pixel, the zipper detector block produces a control signal which enables either the false colors removal algorithm or the zipper effect removal algorithm. More specifically, zipper effect arises when the following three

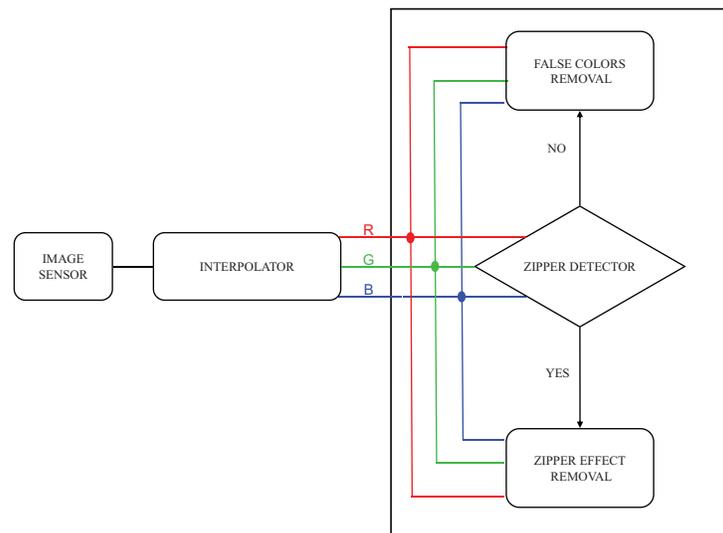


Figure 7.26 : Block diagram of the antialiasing algorithm proposed in [47].

conditions are satisfied:

1. Along the horizontal (vertical) direction passing through the central pixel, the inter-channel difference between the green channel and the other one (for which there are the original sampled values) is almost constant;
2. The trend of the central pixel channel in the vertical (horizontal) direction, the inter-channel difference between the green channel and the other one (for which there are the original sampled values), is not constant;
3. The trend of the central pixel channel in the vertical (horizontal) direction is increasing or decreasing, or there is a minimum or a maximum in the central pixel.

Zipper effects are then removed using the following two equations:

1. G central pixel

$$\begin{aligned}\hat{G}(i, j) &= G(i, j) \\ \hat{H}(i, j) &= f(H_{SURROUND}) \\ \hat{J}(i, j) &= G(i, j) + v_{JG}\end{aligned}\quad (7.83)$$

2. R/B central pixel (e.g., H)

$$\begin{aligned}\hat{G}(i, j) &= f(G_{SURROUND}) \\ \hat{H}(i, j) &= H(i, j) \\ \hat{J}(i, j) &= \hat{G}(i, j) + v_{JG}\end{aligned}\quad (7.84)$$

where $f(\cdot)$ is an average operator, whose inputs are the surrounding pixels ($H_{SURROUND}$ or $G_{SURROUND}$) along the direction having almost constant inter-channel differences and v_{JG} is calculated using the (7.67).

The post-processing approach described in [48] is not only based on the color difference model, but also uses fully adaptive edge-sensing mechanism based on the aggregated absolute differences between the CFA inputs. The spectral correlation between the G and R (or B) components of the full-color image is utilized in the proposed post-processing process to further improve color appearance of the image. Based on the color difference model, the proposed post-processor reevaluates the G components produced by the demosaicking process as follows:

$$\hat{G}(i, j) = H(i, j) + \frac{\sum_{(k,l) \in \zeta} w(k, l) (G(k, l) - H(k, l))}{\sum_{(k,l) \in \zeta} w(k, l)} \quad (7.85)$$

where $\zeta = \{(i-1, j), (i, j-1), (i+1, j), (i, j+1)\}$ denotes the locations of the original G components surrounding the interpolated location (i, j) ; $H(i, j)$ denotes the original R (or B) component at the position under consideration and $w(k, l)$ are the edge-sensing weights, which have to satisfy two conditions:

1. each weight is a positive number, $w(k, l) \geq 0$;
2. the summation of all the weights, $\sum_{(k,l) \in \zeta} w(k, l)$, is equal to unity.

More specifically, these weights are calculated as follows:

$$w(k, l) = \frac{1}{1 + d(k, l)} \quad (7.86)$$

where $d(k, l)$ is the aggregated absolute difference between the G sampled values:

$$d(k, l) = \sum_{(g,h) \in \zeta} |G(k, l) - G(g, h)| \quad (7.87)$$

These weights are used to regulate the contribution of the neighboring input components $G(k, l)$ in the (7.85). In fact, when no edge is positioned across the directions in which

the image is post-processed, the corresponding aggregated absolute difference $d(k, l)$ is small and the CFA component $G(k, l)$, via its corresponding weight $w(k, l)$, contributes greatly in (7.85). The opposite is true in case of an edge. After the post-processing of the G channel is complete, the R (or B) component at B (or R) locations is post-processed as follows:

$$\hat{H}(i, j) = \hat{G}(i, j) + \frac{\sum_{(k,l) \in \zeta} w(k, l) (H(k, l) - G(k, l))}{\sum_{(k,l) \in \zeta} w(k, l)} \quad (7.88)$$

The weights are computed as in (7.86), with $d(k, l) = \sum_{(g,h) \in \zeta} |H(k, l) - H(g, h)|$ where $\zeta = \{(i-1, j-1), (i-1, j+1), (i+1, j-1), (i+1, j+1)\}$.

Finally, the R and B components at G locations are processed. In this case the (7.88) is applied again with $\zeta = \{(i-1, j), (i, j-1), (i+1, j), (i, j+1)\}$ which are the locations of the R (or B) pixels surrounding the central G pixel.

Bibliography

- [1] B. Bayer, "Color imaging array," *U.S. Patent 3,971,065*, 1976.
- [2] R. Palum, "Image sampling with the Bayer color filter array," in *Image Processing, Image Quality, Image Capture Systems Conference (PICS-01)*, (Montral, Quebec, Canada), pp. 239–245, IS&T, 2001. ISBN 0-89208-232-1.
- [3] J. E. Adams, "Interactions between color plane interpolation and other image processing functions in electronic photography," in *SPIE-IS&T Electronic Imaging*, pp. 144–151, 1995.
- [4] K. Hirakawa, "Cross-talk explained," in *IEEE International Conference on Image Processing (ICIP 2008)*, pp. 677–680, 2008.
- [5] M. Guarnera, G. Messina, and V. Tomaselli, "Adaptive color demosaicing and false color removal," *Journal of Electronic Imaging, Special Issue on Digital Photography, SPIE and IS&T*, vol. 19, no. 2, pp. 1–16, 2010.
- [6] S. Battiato, M. Guarnera, G. Messina, and V. Tomaselli, "Recent patents on color demosaicing," *Recent Patents on Computer Science*, Bentham Science Publishers Ltd, vol. 1, no. 2, 2008.
- [7] J. F. Hamilton and J. E. Adams, "Adaptive color plane interpolation in single sensor color electronic camera," *U.S. Patent*, no. US05629734A, 1997.
- [8] O. Kalevo and H. Rantanen, "Color filter array interpolation," *U.S. Patent Application*, no. US20040218073A1, 2004.
- [9] T. Nguyen, "System and method for asymmetrically demosaicing raw data images using color discontinuity equalization," *U.S. Patent Application*, no. US20020167602A1, 2002.
- [10] K. Tajima, "Image processing apparatus," *U.S. Patent Application*, no. US20050058361A1, 2005.
- [11] A. Nilsson and P. Nordblom, "Weighted gradient based and color corrected interpolation," *U.S. Patent Application*, no. US20070292022A1, 2007.
- [12] G. Messina, M. Guarnera, V. Tomaselli, A. Bruna, G. Spampinato, and A. Castorina, "Color interpolation method of an image acquired by a digital sensor by directional filtering," *U.S. Patent*, no. US7305123B2, 2007.
- [13] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 997–1013, 2002.
- [14] A. Hunter and S. Pollard, "Electronic image colour plane reconstruction," *U.S. Patent Application*, no. US20030086606A1, 2003.

- [15] H. Lee, "Interpolator, method, and digital image signal processor for adaptive filtering of Bayer pattern color signal," *U.S. Patent Application*, no. US20040196395A1, 2004.
- [16] T. Chen, "A study of spatial color interpolation algorithms for single-detector digital cameras." Internet Link. <http://scien.stanford.edu/class/psych221/projects/99/tingchen/main.htm>.
- [17] S. Nohda, "Image pickup apparatus having an interpolation function," *U.S. Patent*, no. US6295087B1, 2001.
- [18] C. Chen, C. Lin, C. Wu, Z. Yin, and C. Wang, "Demosaicking method and apparatus for color filter array interpolation in digital image acquisition systems," *U.S. Patent*, no. US7292725B2, 2007.
- [19] K. Kimura and T. Yoshida, "Data processing apparatus, image processing apparatus, camera, and data processing method," *U.S. Patent Application*, no. US20050007470A1, 2005.
- [20] D. Keren, "Image demosaicing method," *U.S. Patent*, no. US6625305B1, 2003.
- [21] Y. Hel-or and D. Keren, "Image demosaicing method utilizing directional smoothing," *U.S. Patent*, no. US6404918B1, 2002.
- [22] Y. Hel-or and D. Keren, "Image demosaicing method utilizing directional smoothing," *U.S. Patent*, no. US6618503B2, 2003.
- [23] D. Alleysson and B. C. de Lavarène, "Frequency selection demosaicking: a review and a look ahead," *Visual Communications and Image Processing (VCIP-2008)*, vol. 6822, no. 1, p. 13, 2008.
- [24] G. M. Farinella, S. Battiato, G. Gallo, and R. Cipolla, "Natural versus artificial scene classification by ordering discrete fourier power spectra," in *SSPR & SPR '08: Proceedings of the 2008 Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, (Berlin, Heidelberg), pp. 137–146, Springer-Verlag, 2008.
- [25] J. W. Glotzbach, R. W. Schafer, and K. Illgner, "A method of color filter array interpolation with alias cancellation properties," in *IEEE International Conference on Image Processing (ICIP-2001)*, pp. 141–144, 2001.
- [26] K. Hirakawa and P. J. Wolfe, "Spatio-spectral color filter array design for enhanced image fidelity," in *IEEE International Conference on Image Processing (ICIP-2007)*, pp. 81–84, 2007.
- [27] K. Hirakawa and P. Wolfe, "Spatio-spectral color filter array design for optimal image recovery," *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1876–1890, 2008.

- [28] D. Alleysson, S. Susstrunk, and J. Herault, "Linear demosaicing inspired by the human visual system," *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 439–449, 2005.
- [29] N. Lian, L. Chang, Y.-P. Tan, and V. Zagorodnov, "Adaptive filtering for color filter array demosaicking," *IEEE Transactions on Image Processing*, vol. 16, no. 10, pp. 2515–2525, 2007.
- [30] D. Alleysson, S. Süssstrunk, and J. Héroult, "Color demosaicing by estimating luminance and opponent chromatic signals in the fourier domain," in *Color Imaging Conference*, pp. 331–336, 2002.
- [31] E. Dubois, "Filter design for adaptive frequency-domain Bayer demosaicking," in *IEEE International Conference on Image Processing (ICIP 2006)*, pp. 2705–2708, 2006.
- [32] D. Menon and G. Calvagno, "Demosaicing based on wavelet analysis of the luminance component," in *IEEE International Conference on Image Processing (ICIP 2007)*, vol. 2, pp. 181–184, 2007.
- [33] G. Spampinato, A. Bruna, G. Sanguedolce, E. Ardizzone, and M. L. Cascia, "Improved color interpolation using discrete wavelet transform," *Image and Video Communications and Processing 2005*, vol. 5685, no. 1, pp. 753–760, 2005.
- [34] K. Revathy, G. Raju, and R. P. Nayar, "Image zooming by wavelets," *Fractals - an Interdisciplinary Journal on the Complex Geometry*, vol. 8, no. 3, pp. 247–254, 2000.
- [35] H. Stark and P. Oskoui, "High-resolution image recovery from image-plane arrays, using convex projections," *Journal of the Optical Society of America A*, vol. 6, pp. 1715–1726, 1989.
- [36] M. K. Özkan, A. M. Tekalp, and M. I. Sezan, "Pocs-based restoration of space-varying blurred images," *IEEE Transactions on Image Processing*, vol. 3, no. 4, pp. 450–454, 1994.
- [37] L. Chang and Y.-P. Tan, "Effective use of spatial and spectral correlations for color filter array demosaicking," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 355–365, 2004.
- [38] W. Lu and Y.-P. Tan, "Color filter array demosaicking: new method and performance measures," *IEEE Transactions on Image Processing*, vol. 12, no. 10, pp. 1194–1210, 2003.
- [39] T. W. Freeman, "Median filter for reconstructing missing color samples," *US Patent*, no. US4724395, 1988.

- [40] C. W. Kim and M. G. Kang, "Noise insensitive high resolution demosaicing algorithm considering cross-channel correlation," in *IEEE International Conference on Image Processing (ICIP 2005)*, pp. 1100–1103, 2005.
- [41] R. Lukac, K. Martin, and K. N. Plataniotis, "Demosaicked image post-processing using local color ratios," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 6, pp. 914–920, 2004. <http://www.dsp.utoronto.ca/~lukacr/download>.
- [42] W. C. Chan, O. C. Au, and M. F. Fu, "A novel color interpolation framework in modified ycbcr domain for digital cameras," in *IEEE International Conference on Image Processing (ICIP 2003)*, pp. 925–928, 2003.
- [43] J. E. Adams JR., J. F. Hamilton JR., and J. A. Hamilton, "Removing color aliasing artifacts from color digital images," *US Patent*, no. US6804392, 2004.
- [44] V. Tomaselli, M. I. Guarnera, and G. Messina, "False-color removal on the YCC color space," in *SPIE-IS&T Electronic Imaging*, vol. 7250, 2009.
- [45] R. P. Maurer, "Reduction of chromatic bleeding artifacts in image," 2003.
- [46] D. Menon, S. Andriani, and G. Calvagno, "Demosaicing with directional filtering and a posteriori decision," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 132–141, 2007. <http://www.danielemenon.it/pub/dfapd/dfapd.html>.
- [47] M. I. Guarnera, G. Messina, and V. Tomaselli, "Method and system for demosaicing artifact removal, and computer program product therefor," *US Patent Application*, no. US20060087567A1, 2006.
- [48] R. Lukac and K. N. Plataniotis, "A robust, cost-effective postprocessor for enhancing demosaicked camera images," *Real-Time Imaging, Special Issue on Spectral Imaging II*, vol. 11, no. 2, pp. 139–150, 2005. <http://www.dsp.utoronto.ca/~lukacr/download>.

Red Eyes Removal

G. Messina

Advanced System Technology - Catania Lab, STMicroelectronics, Italy.

T. Meccio

Image Processing Lab, University of Catania, Italy.

Abstract: Since the large diffusion of mobile devices with embedded camera and flashgun, the red eye artifacts have de-facto become a critical problem. Red eyes are caused by the flash light reflected off the blood vessels of the human retina. This effect is more pronounced when the flash light is closer to the camera lens, which often occurs in compact imaging devices. To reduce these artifacts, most cameras have a red-eye flash mode which fires a series of pre-flashes prior picture acquisition. The biggest disadvantage of the pre-flash approach is power consumption (flash is the most power-consuming part of imaging devices), and thus it is not suitable for power-constrained systems (e.g., mobile devices). Moreover, this approach does not guarantee total prevention of red eye artifacts. Red eye removal must then be performed in post-processing, through the use of automatic correction algorithms. The aim of this Chapter is to depict the state of the art of automatic detection and correction of red eyes, taking into account strong points and drawbacks of the most well-known techniques, with particular emphasis on the image degradation risk associated to false positives in red eye detection and to wrong correction of red eyes. Furthermore the problem of estimating the quality of the final result, without reference image, is examined.

8.1 Introduction

Red eye artifacts are a well-known problem in digital photography . They are caused by direct reflection of light from the blood vessels of the retina through the pupil to the camera objective. When taking flash-lighted pictures of people, light reflected from the retina forms a cone, whose angle α depends on the opening of the pupil. Be β the angle between the flash-gun and the camera lens (centered on the retina), the red eye artifact is formed if the red light cone hits the lens, that is, if α is greater than or equal to β (see Fig.(8.1)). Small compact devices and point-and-click usage, typical of non-professional photography, greatly increase the likelihood for red eyes to appear in acquired images.

High-end cameras often feature a separate flash with an extensible and steerable bracket, which allows for more distance between the flash and the lens, thus reducing the probability for red eyes to appear. One preventive measure suitable to both high-end and low-end devices is to make additional flashes before actually taking the photograph (pre-flash). This method, first proposed by Kodak [1], gives time to pupils to shrink in order to reduce the reflectance surface, thus making red eyes less likely. It is important that enough time elapses between flashes to account for the response time of the pupils (see Fig.(8.2)). This approach is effective, but it has the disadvantage of greatly increasing power consumption, which may be problematic for power-constrained mobile devices. Also, the additional flashing may sometimes be uncomfortable to people.

Red eye prevention methods reduce the probability of the phenomenon but don't remove it entirely. Most of the times, then, the picture must be corrected during post-

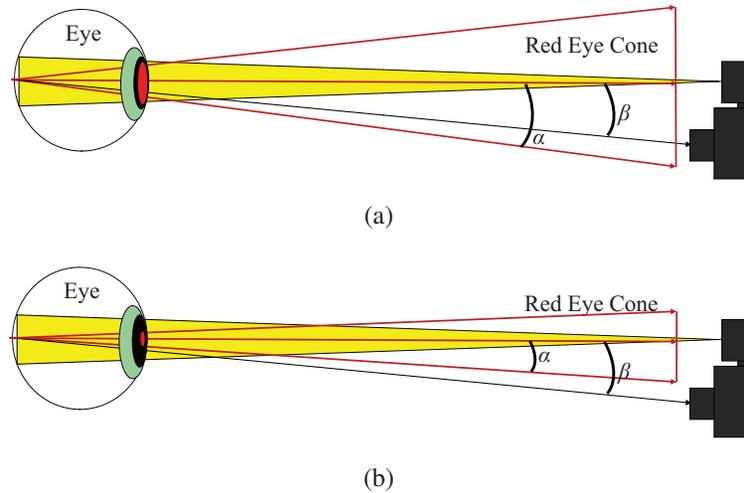


Figure 8.1 : The red eye is caused by the reflection of the flash off the blood vessels of the retina. The camera will record this red hue if the angle β is not greater than α (a), otherwise the red eye is not recorded (b).

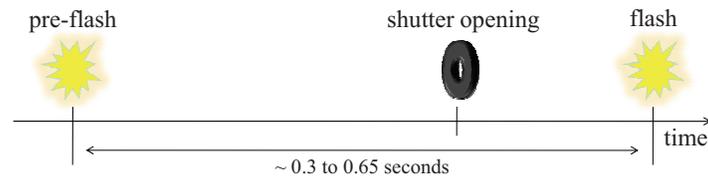


Figure 8.2 : Timeline explaining the pre-flash approach. Before the actual acquisition, a flash is fired. After a short time, the shutter opens and light enters the sensor. At the end of the exposure time the “true” flash is fired. Time between flashes is such that the pupils have time to react and shrink.

processing. Red eye removal is a very challenging task: red eyes may vary in shape and color, and may also differ in position and size relative to the whole eye. Sometimes light is reflected on a part of the retina not covered with blood vessels, yielding a yellow or white reflection (golden eyes). Some examples of the phenomenon are showed in Fig.(8.3). Designing a system which can effectively address all the possible cases is very difficult.

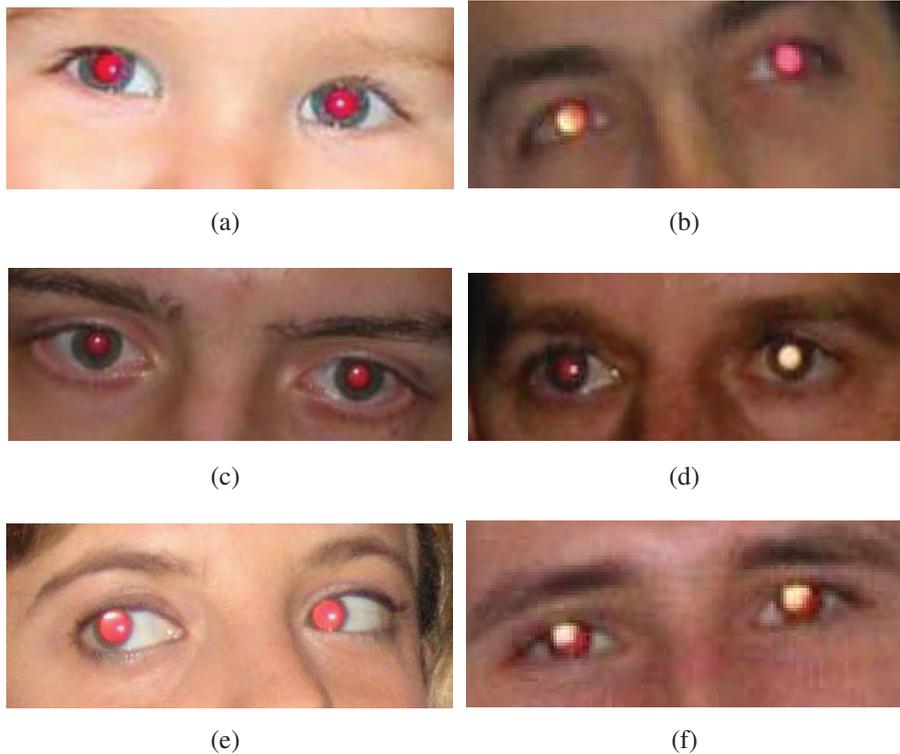


Figure 8.3 : Examples of the variability of the red eye phenomenon. Golden eyes are also visible.

For red eyes to be removed, they must be first reliably detected then properly corrected. Detection methods are divided into semi-automatic methods, which ask the user to manually localize and point the red eyes, and automatic methods, which detect the red eyes themselves. In the first case the eyes are manually selected using a visual interface (e.g., Adobe Photoshop [2], Corel Paint Shop Pro [3], ACDSee [4], etc.). This is feasible because eyes are easy to localize for men, but requiring manual intervention for every picture taken is unsuitable for non-professional usage; moreover, it may be difficult to have such an interface on a mobile device.

Automatic methods attempt to find red eyes on their own. Since they don't require user intervention, they are easier to use and more appealing, thus suitable for embedded devices. However, automatic detection of red eyes is a very challenging task, due to the variability of the phenomenon and the general difficulty in reliably discerning the shape of eyes from other details.

Red eye correction techniques, on the other hand, may be more or less invasive. Gen-

erally speaking, “easier” cases may be addressed with a softer correction, while sometimes a stronger intervention is needed. Since the aim is to provide a corrected image which looks as natural as possible, a less invasive correction is preferred when the natural aspect of the eye is reconstructible from the acquired image.

This Chapter aims to provide an overview of well-known automatic red eye detection and correction techniques, pointing out working principles, strengths and weaknesses of the various solutions. For further information about red eye removal, see recent surveys by Gasparini *et al.* on academic papers [5], on patents [6] and some of our personal works [7–9].

This Chapter is organized as follows. Section 8.2 explores red eye detection. Section 8.3 describes methods for red eye correction, while section 8.4 gives an insight into the problem of unwanted and improper corrections, showing their side effects. Lastly, Section 8.5 provides criteria to evaluate the quality of the results.

8.2 Eye Detection

The main difficulty in detection of red eyes is their great degree of variability. In the easier cases, the pupil has a normal shape and size and differs from a regular one only by its color. However, it is not uncommon for the red reflection to spread over the iris generating an unnatural luminance distribution. Usually a small white glint is also present, representing the direct reflection of the flash on the surface of the eye and giving the eye much more naturalness.

Typical red eye detection approaches involve extraction of red zones combined with skin extraction, shape template matching, and/or face detection. Some approaches also make use of classifiers to further refine their results.

8.2.1 Color Based

Color based approaches are the simplest ones. They are based on detecting red zones which may correspond to red eye artifacts. As a typical constraint for the position of the red eyes, they also detect the human skin, then consider some criteria about the relative position of the red eyes and the skin (usually, the eyes must be almost completely surrounded by nearby skin). Some color based approaches also detect the sclera (the white part of the eye), distinguishing it from the skin. Possible constraints may be imposed about the geometry of the red zones, such as discarding candidates too much elongated to represent a red pupil. This kind of approaches is quite simple, but does not take into account more complex features like, e.g., the presence of the various parts of the eye or the detection of the face.

One of the biggest problems of color-based techniques is characterizing exactly the colors to look for. Usually, interesting portions of the color space (corresponding to red, skin color, etc.) are delimited by hard thresholds, but they may also be delimited by soft margins, yielding a fuzzy probability for the color to belong to the region. However, finding proper boundaries for the regions is a challenging task. The color of red eyes is

heavily influenced by the type of flash used, the sensor and the processing pipeline. While this is not a big issue, since the thresholds may be fine-tuned to adapt to the acquisition system, there are external factors which may influence the color of the eyes, including (but not limited to) the age of the person, the opening of the pupils, the distance from the camera, and the angle between the eyes and the flash. The variability is so high that even the same subject in one picture may have two different colored red eye artifacts, or a red eye and a regular one (see Fig.(8.4)). Moreover, if the flash is not very strong (as is often the case with mobile devices), the external illuminant may produce a noticeable color cast on the picture, which adds another degree of variability to the colors. Similar considerations apply to the color of the skin and of the sclera.



Figure 8.4 : Picture (a) shows two very different red eyes; picture (b) shows one red eye along with a regular one.

The red color region may be defined in different color spaces. In the RGB space, a possible definition is [10]:

$$\begin{cases} R > 50 \\ R/(R+G+B) > 0.40 \\ G/(R+G+B) < 0.31 \\ B/(R+G+B) < 0.36 \end{cases} \quad (8.1)$$

Often, instead of hard thresholds, a *Redness* function is provided. This function is an estimate of how well the color of each pixel resembles a red eye artifact, and is used as a way to define soft margins for the red color region. Some possible redness functions [11–14] are:

$$Redness = (R - \min \{G, B\}) \quad (8.2)$$

$$Redness = \frac{R^2}{(G^2 + B^2 + 14)} \quad (8.3)$$

$$Redness = \frac{\max \{0, (R - \max \{G, B\})\}^2}{R} \quad (8.4)$$

$$Redness = \max \left\{ 0, \frac{2R - (G + B)}{R} \right\}^2 \quad (8.5)$$

As an alternative to select an interesting portion of the color space, it is possible to compare a redness function with a luminance function, discarding pixels whose luminance is more noticeable than the redness [15]:

$$\text{Redness} = R - (G + B) / 2 \quad (8.6)$$

$$\text{Luminance} = 0.25R + 0.6G + 0.15B \quad (8.7)$$

$$\text{RedLum} = \max \{0, 2 \cdot \text{Redness} - \text{Luminance}\} \quad (8.8)$$

Search for red regions may also be performed in color spaces different from RGB, such as YCC [16] or HSL [17]. See Section 8.2.5 for a complete example of red eye search and patch extraction performed in the HSL space.

Given a particular choice for the red color region, it is possible to convert each image to a representation which shows whether each pixel belongs to the region. Such representations are called redness maps. According to the employed definition for the red color region (hard-thresholded or soft-delimited), the redness map is a black-and-white or full-grayscale image (in the latter case, the redness function is adjusted to the possible maximums and minimums of the redness function, or to the maximums and minimums over each particular image). Fig.(8.5) and Fig.(8.6) show redness maps computed using the above formulas.

Skin extraction may be performed in a similar way as red color extraction. For further information about skin extraction, see Chapter 3 and Chapter 5.

Other color based information useful to detect red eyes may be gained searching for the sclera [18] and selecting the zones where the flash has noticeably affected the image (discarding, e.g., a distant background) [19]. Using thresholding and morphological operators to combine different masks, it is possible to effectively extract red pupils, as shown in Section 8.2.5.

8.2.2 Shape Based

Shape based approaches attempt to find eyes exploiting simple information about their shape. They typically use templates which are matched at different positions and resolutions, in order to search the image for shapes which may correspond to eye features. The region of interest is then restricted to zones where the response of the templates is stronger. Using simple circular or square templates it is possible to recognize, e.g., the difference in intensity between the inner pupil and the outer skin and sclera. Slightly more complex templates may be useful in locating the other parts of the eye, which helps to effectively assess the presence or the absence of a red eye [20].

Edge detection filters may also be useful to extract information about shape. It is possible to use them in conjunction with color tables to make advantage of both spatial and chromatic information [21].



Figure 8.5 : Examples of redness maps. **(a)** Original image; **(b-f)** redness maps obtained from (8.1), (8.2), (8.3), (8.4), (8.5), respectively.



Figure 8.6 : (a) Redness map obtained from (8.6); (b) redness vs. luminance map computed according to (8.8).

8.2.3 Pairing Verification

One of the most obvious constraints which can be used to filter out false detections is eye pairing verification [22]. It is based on the assumption that every eye found must be paired with the other one on the same subject's face. The two eyes must have the same size, and they must be in a certain range of distances (possibly proportional to the size, in order to account for the distance of the subject from the camera) from each other, in a horizontal or almost horizontal direction. If an eye can't be paired because it has no suitable match, it is discarded, since it is most probably a false detection.

This approach is effective, since it is very unlikely for two false positives to satisfy the pairing criteria, but it presents a major drawback: if a face is partially occluded, so that only one eye is visible in the picture, and that eye is red, it will not be corrected, since it can't be matched to the other one. The same problem will occur when both eyes are visible but only one is red, or when both are red but only one is detected, possibly due to a difference in color (see Fig.(8.7)).

8.2.4 Face Detection

The most sophisticated kind of approach to red eye detection is based on face detection [12]. Restricting the search region to the zones where faces are detected, it is possible to discard a great number of false positives.

In details, a face detection system determines the locations and sizes of human faces in arbitrary digital images by making use in some cases of ad-hoc facial features. The localization is done by considering a bounding box that encloses the region of interest. The detection problem is often achieved as a binary pattern classification task; the content of a given part of an image is transformed into features used to train a classifier on example faces able to decide whether that particular region of the image is a face, or not. For practical situations is very common to employ a sliding-window technique just using the



(a)



(b)



(c)

Figure 8.7 : In picture (a), only one of the eyes is visible; the red eyes in picture (b) are very different, and in most cases only one of them will be properly detected; in picture (c) only one of the eyes is affected by the red eye phenomenon. In all these cases, the pairing verification will fail.

classifier on small portions of an image (usually squared or rectangular), at all locations and scales, as either faces or non-faces. In the more general case the face localization is achieved regardless of position, scale, in-plane rotation and orientation, pose (out-of-plane rotation) and illumination. Further source of problems are the presence or absence of structural components (e.g., beards, mustaches and glasses), the facial expression that has a great impact over the face appearance and the occlusions that occur when faces may be partially occluded by other objects.

To implement a robust face detector it is fundamental to fix some points relative to the specific application. In particular it is important to decide the facial representation, the involved pre-processing, the particular "cues" (e.g., colors, shape, etc.) and the classifier design.

In literature a lot of approaches have been published with different capabilities, advantages and limitations. Of course, the implementation of a face detector system inside an embedded device system requires ad-hoc peculiarities due the limited available resources. The constrained domain imposes to consider methods able to guarantee a reasonable trade-off between robustness and computational issues. For this reason it is out of the scope of this Section to provide a detailed review of all related technologies. See [23,24] for more specific details.

One of the most popular algorithms in the field is due to Viola and Jones [25]. For this reason we have decided to describe it in a more details just to give also some useful suggestions for a practical implementation. The authors introduced for the first time the



Figure 8.8 : Face Detection Example.

concept of "integral image", a way to compute efficiently local features in an incremental way just to find in a suitable way the underlying scale of the face to be localized. The integral image $Int(x,y)$ of a given image I at location (x,y) is defined as follows:

$$Int(x,y) = \sum_{(i \leq x, j \leq y)} I(i,j) \quad (8.9)$$

It is computed using just a single pass over the image I as described in [25]. By proper managing such image it is possible to compute any processing over rectangular patches in a very efficient manner. The corresponding rectangular features can then be computed as simple differences between adjacent rectangular sums. Although the similarity with the Haar features [26] are pretty evident, the proposed strategy is able to obtain effective results in a more efficient way.

The remaining important contribution of the seminal work of Viola and Jones was the introduction of an ad-hoc classifier making use of a learning approach based on Adaboost [27]. This classifier is able to discriminate, among a large set of potential features, a smaller number of elements without lacking too much of accuracy. The system is able to select, among others, a small number of features just considering a boosting approach [28] that, given an exhaustive set of positive and negative examples, with a greedy algorithm, decides the best set of features to be considered both in terms of robustness and fast detection rate (see Fig.(8.9) an example of rectangular features.).

Finally, they described a way to combine efficiently, in a cascade approach, the output of different classifiers just to speed-up the overall process. With respect to the former approaches this method was the first able to work with sufficient accuracy in real time application.

Of course, the underlying ingredients of a face detector can be improved in several ways just providing to the overall flow further information to be processed. In [29] is proposed a learning based face detector able to find human faces in a very fast way. To

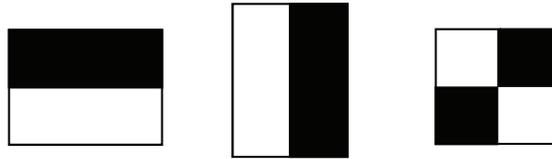


Figure 8.9 : Some examples of rectangular features managed by the algorithm of Viola and Jones. Each feature can be computed in a very efficient way just using the concept of integral image.

further speed up the process, a face rejection cascade is constructed to remove most of negative samples while retaining all the face samples. To do that a series of skin color features [30] are used as useful prior.

The face localization can be thought as the pre-processing step needed to recognize the person which the face belong to. Person identification is, of course, a more difficult task especially if implemented in a so constrained domain such as imaging devices. Also in this field there is a huge number of papers published over the past decades, specifically devoted to biometric world. By the way, personal photos have an associated context, often already available to the user of the photo management system. In newer systems the combination of user feedback with EXIF meta information (and, if available, with GPS location) can give an effective improvement to the unsupervised recognition [31].

First commercial products are available also to be used in novel application context such as social network [32,33], etc.

The quality of the detection greatly depends on the quality of the face detector. Sometimes it is limited to frontal upright faces, while red eye artifacts may be located in profile or three-quarter views of subjects (especially when taking snapshots). Therefore, face detectors with such limitations are not suitable for red eye detection. Another important degree of variability is the age of the subject: children are difficult to detect, since their faces have a different shape and different features than those of adults. Nonetheless, they have a higher chance to present red eye artifacts, since their pupils are usually more open. Thus, it is important for face detection to be robust both to the angle of view and to the age of the subject.

Another important issue related to face detection is that it doesn't help discard false detections on the face, which are usually critical. An additional constraint which may be imposed is to only accept eyes located in the upper half of the detected face. This helps filter out some false detections (e.g., lips or tongue) but it keeps the ones near the eyes (e.g., details of glasses or pimples on the forehead).

8.2.5 Example of Red Patch Extraction

A complete procedure to extract red eye patches is depicted here. The RGB image is transformed into HSL color space, in order to easily distinguish the red zones using the

H, S and L channels. Let $r, g, b \in [0, 1]$ be the red, green and blue coordinates, respectively, of a color in the RGB space. Let max be the greatest of r, g , and b , and min the least. The hue angle $h \in [0, 360]$ for HSL space, is derived as follows:

$$h = \begin{cases} 0 & \text{if } max = min \\ (60^\circ \times \frac{g-b}{max-min} + 360^\circ) \bmod 360^\circ, & \text{if } max = r \\ (60^\circ \times \frac{b-r}{max-min} + 120^\circ), & \text{if } max = g \\ (60^\circ \times \frac{r-g}{max-min} + 240^\circ), & \text{if } max = b \end{cases} \quad (8.10)$$

Saturation and lightness $s, l \in [0, 1]$ are computed as follows:

$$l = \frac{1}{2}(max + min) \quad (8.11)$$

$$s = \begin{cases} 0 & \text{if } max = min \\ \frac{max-min}{2l}, & \text{if } l \leq \frac{1}{2} \\ \frac{max-min}{2(1-l)}, & \text{if } l > \frac{1}{2} \end{cases} \quad (8.12)$$

The value of h is generally normalized to lie between 0 and 360° , and $h = 0$ is used when $max = min$ (that is, for grays) though the hue has no geometric meaning there, where the saturation s is zero. Red values are taken just considering $-60^\circ \leq h \leq 20^\circ$ and saturation $s \geq 0.6$.

Employing such filtering, a binary mask with the red zones is properly derived. The mask is not always perfect because the original image can contain isolated red pixels. A closing operation (i.e., a combination of dilation and erosion morphological operations) is needed. In our approach we have used the following 3×3 structuring element:

$$m = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (8.13)$$

Once the closing operation has been accomplished, a search of the connected components is achieved using a simple scanline approach. Each group of connected pixels is labeled and analyzed through simple geometric constraints. The single region of connected pixels is classified as possible red-eye candidate if the following constraints are satisfied:

- the size S_i of the connected region i is within the range $[Min_{size}, Max_{size}]$, where Max_{size} and Min_{size} have been heuristically derived.
- the binary roundness constraint R_i , of the connected region i is verified:

$$R_i = \begin{cases} true & \rho_i \in [Min_\rho, Max_\rho]; \eta_i \leq Max_\eta; \xi_i \gg 0 \\ false & otherwise \end{cases} \quad (8.14)$$

where

$\rho_i = \frac{4\pi \times A_i}{P_i^2}$ is the ratio between the estimated area A_i and the perimeter P_i of the connected region; the more this value is near 1 the more the shape will be similar to a circle; the thresholds have been fixed in our experiments in $Max_\rho = 1.40$ and $Min_\rho = 0.50$.

$\eta_i = \max\left(\frac{\Delta_{x_i}}{\Delta_{y_i}}, \frac{\Delta_{y_i}}{\Delta_{x_i}}\right)$ is the distortion of the connected region along the axes. In our experiments the $Max_\eta = 2$ was used as threshold.

$\xi_i = \frac{A_i}{\Delta_{x_i}\Delta_{y_i}}$ is the filling factor, the more this parameter is near 1 the more the area is filled.

In Fig.(8.10) all the involved steps in filtering pipeline are shown. The regions which satisfy all the constraints are used to extract the red eye patches candidates from the original input image (Fig.(8.11)). The derived patches are resampled to a fixed size (i.e., 30×30 pixels) for further classification purposes (see Section 10.4.1).

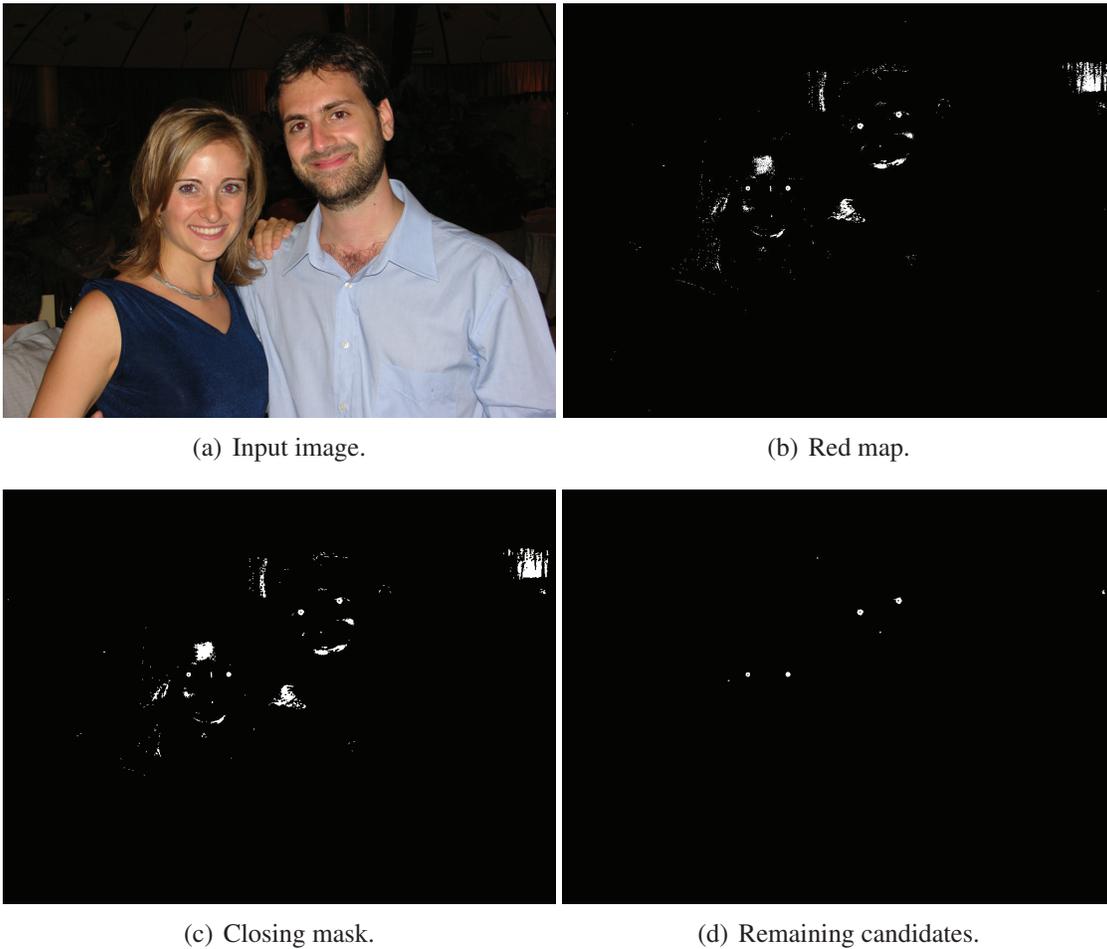


Figure 8.10 : Red patches extraction from a CCD input image.

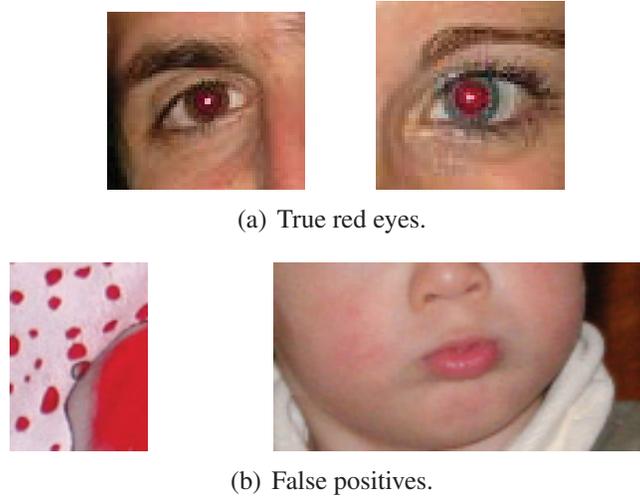


Figure 8.11 : Examples of possible candidates after red patches detection.

8.3 Red Eye Correction

The goal of red eye correction is to modify the image in such a way that it looks as natural as possible, given the assumption that there are red eye artifacts in the detected zones (according to the eye detector, the assumption may be given for sure or with a certain degree of probability).

According to the extent to which the artifact has corrupted the image, the correction algorithm may need to adjust the hue, brightness, luminance distribution, and/or even the shape and size of the pupil. Since naturalness of the image is the goal, it is best to use a minimally invasive technique to correct each case. This also means that a way to distinguish the gravity of each artifact (either in the detection phase or at the very beginning of the correction phase) is to be preferred, in order to be able adapt the correction method on a case-by-case basis [34].

8.3.1 De-saturation

In the simplest cases, the eye has its regular shape, and the artifact only consists in the wrong color of the pupil. In these cases, the optimal solution is equally simple: the red eye is desaturated, that is, its chrominance is (totally or partially) suppressed, while its luminance is left intact or only slightly lowered (see Fig.(8.12)).

One simple way of desaturating red pupils is to replace each pixel with a gray shade at 80% of original pixel luminance [16]. An adaptive desaturation may be performed in the CIELAB color space by stretching the lightness values of the pupil so that its darkest point becomes black [35]:

$$\begin{aligned}
 L_{corrected}^* &= \frac{\max L^*}{(\max L^* - \min L^*)} (L^* - \min L^*) \\
 a_{corrected}^* &= 0 \\
 b_{corrected}^* &= 0
 \end{aligned} \tag{8.15}$$

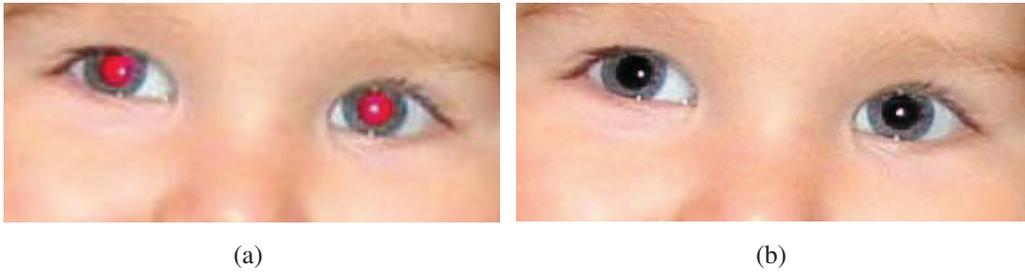


Figure 8.12 : In the simplest cases, pupil desaturation produces good results.

Desaturation may suffer from a boundary effect: the transition between the corrected and uncorrected area may be noticeable and unpleasant. Moreover, some pixels outside the pupil may be incorrectly considered to be part of the red eye artifact and desaturated. To overcome these problems, a smoothing (usually Gaussian) mask may be used to modulate the strength of the correction. For each pixel (i, j) in the red eye artifact area, be $c_{original}(i, j)$ its color in the uncorrected image, $c_{target}(i, j)$ the target color of the correction and $m(i, j)$ the value of the smoothing mask; the final corrected color $c_{corrected}(i, j)$ is then:

$$c_{corrected}(i, j) = c_{target}(i, j) \cdot m(i, j) + c_{original}(i, j) \cdot (1 - m(i, j)) \quad (8.16)$$

8.3.2 Inpainting

In the hardest cases, a more invasive correction is needed. Often, the distribution of reflected light is influenced by the direction of the flash with respect to the face. Sometimes eyes present the “washed out” effect, where the reflected light spreads off the pupil onto the iris. In these cases a simple desaturation may yield incorrect and unnatural results (see Fig.(8.13)).

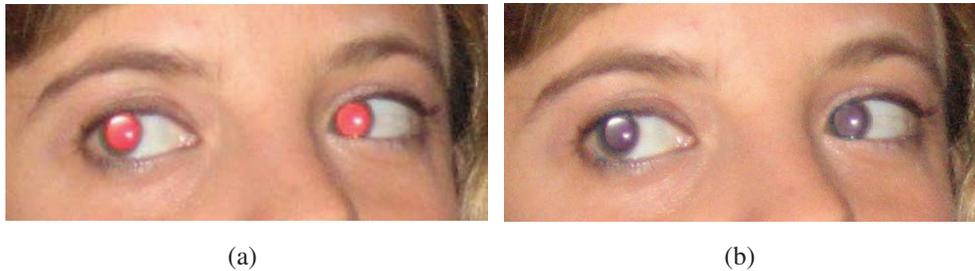


Figure 8.13 : When reflected light spreads over the iris, simple desaturation gives unnatural results.

It is then necessary to use a more complex method to reconstruct a realistic image of the eye. Inpainting may vary from an adaptive recoloring of red pixels to a complete

redrawing of iris and pupil [36]. The results, however, tend to be unrealistic, up to the point that they sometimes resemble glass eyes (see Fig.(8.14)¹).

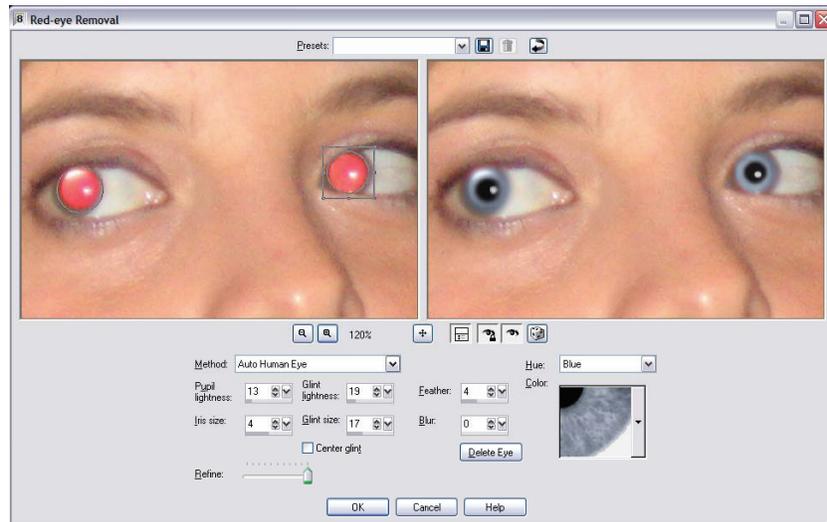


Figure 8.14 : Correction of washed-out red eyes with an inpainting technique.

8.3.3 Flash/no-Flash

Another way of obtaining simultaneous detection and correction of red eye artifacts is the “flash/no-Flash” technique [37], which aims to combine the advantages of taking a non-flashed picture and a flashed one. The main idea is to take a high-quality flashed picture and a low-quality non-flashed one, which is used to detect the red eyes and recover the natural colors of the affected zones (see Fig.(8.15)²).

The method works as follows: two pictures are taken in quick succession. The first one is shot without flash with high sensitivity, large lens aperture and with a short (for a non-flashed picture in low light conditions) exposure time. This yields a dark and noisy picture with small depth of focus, but still suitable to help recover the unaltered colors of the eyes. The second one is a regular flashed picture, which represents the “real” picture to correct. It is important that the two pictures are taken with the same focal length and that very little time elapses in between, in order to prevent misalignment. Search for red eye artifacts is performed in a luminance-chrominance color space, usually CIELAB. The a^* channel is used as a measure of redness. Pixels whose a^* component exceeds a certain threshold are considered red. Among such pixels, those whose difference between the a^* channel in the flashed image and the same channel in the non-flashed image is larger than another threshold are marked as possible red eye pixels. Morphological operators are used to cluster them into blobs, discarding isolated pixels or very narrow regions as noisy results.

¹Corel Paint Shop Pro Red-eye Removal tool.

²Picture taken from Petschnigg *et al.* [38].

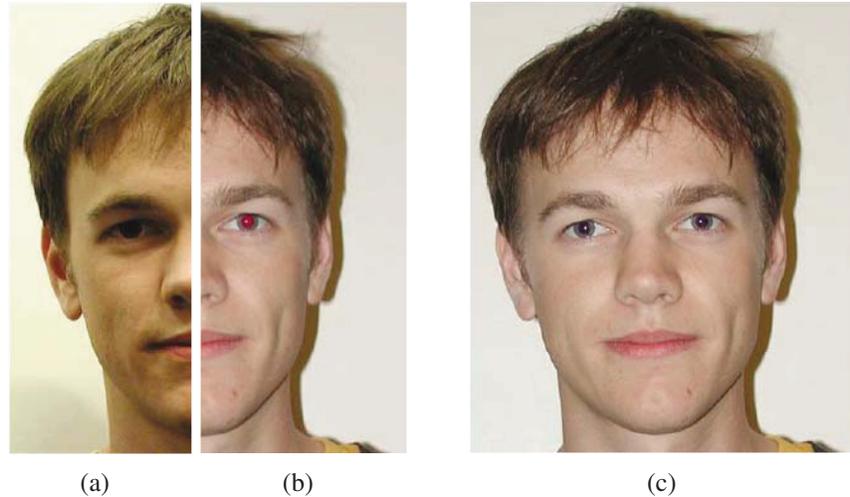


Figure 8.15 : Flash/no-Flash technique. **(a)** Dark non-flashed picture used to recover the correct color of the eyes; **(b)** high-quality picture affected by red eye artifacts; **(c)** corrected picture.

To correct red eyes using information from the non-flashed picture, it is important to first compensate differences in color cast between the two images. To this end, for each of the chroma channels a^* and b^* , the difference between the two images is averaged over all non-red eye pixels, thus obtaining a color compensation term. Correction of red eye artifacts is then performed by substituting the chrominance of affected pixels in the flashed image with the chrominance of the corresponding pixels in the non-flashed image, then adding the color compensation term.

The approach is quite simple and theoretically effective, but it presents a number of drawbacks. First of all, the memory and processing requirements double, since there are two pictures being taken in place of one. Moreover, the images may suffer from registration problems, or they may simply be misaligned due to movements of the hand or of the subjects. This makes this method especially unsuitable for snapshots, where people may be caught while moving. Another important issue of this approach is uneven illumination, which is recorded by the non-flashed image but not by the flashed one: a dark shadow on a red detail (such as the shadow of the nose projected on the lips) may trigger a false detection, which in turn causes image degradation (especially if the chrominance of the shaded part is not correctly perceived due to insufficient illumination).

8.4 Correction Side Effects

8.4.1 False Positive

One of the biggest issues in red eye removal is false positives in the detection phase. Correcting a red detail falsely detected as a red eye artifact may have a much more displeasing effect than leaving an artifact uncorrected. For this reason, getting as few false positives

as possible is more important than catching as many red eyes as possible. Examples of image degradation resulting as correction of false positives are shown in Fig.(8.16).

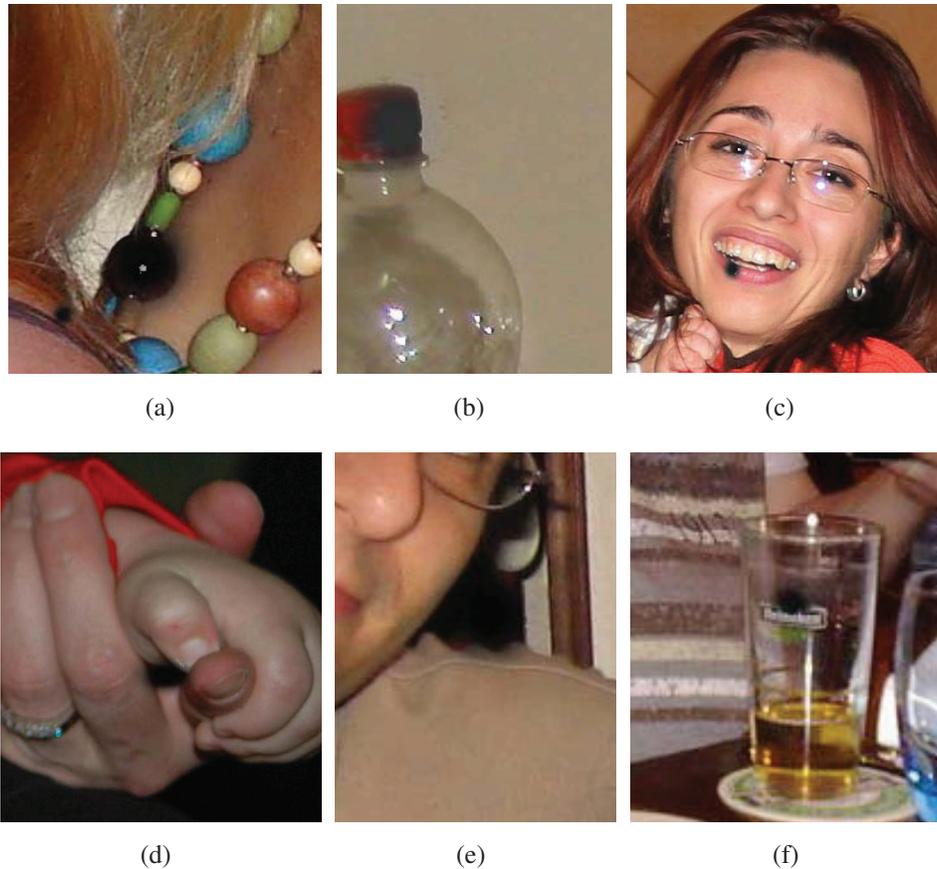


Figure 8.16 : Examples of corrections of false positives. Some are barely noticeable, while others are totally unacceptable.

False positives can be classified according to the severity of the associated degradation risk, as discussed in Section 8.5.

8.4.2 Partial Detection/Correction

Sometimes eyes are properly detected, but wrongly corrected. In such cases unnatural corrections appear in the final image. Unnatural corrections, like false positives, are very undesirable, since they are often more evident and displeasing than untouched red eyes. One type of unnatural correction is partial correction, caused by an incorrect segmentation of the red eye zone (possibly due to a difference in hue or luminance between the detected and the undetected parts).

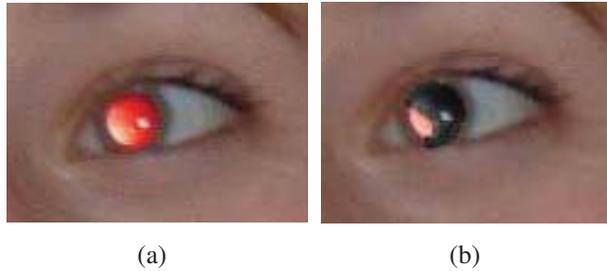


Figure 8.17 : Partial red eye correction, where the brighter area was not considered to belong to the red pupil.

Noisy Correction

Noisy correction is another kind of unnatural correction. Noisy corrections appear when, in presence of heavy image noise, red pixels are present around the pupil. In this case, the detector may assume that such pixels belong to the red eye, and correction may spread over the iris, giving a strange and unnatural look to the corrected eye.

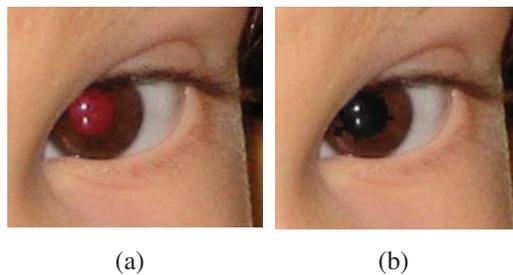


Figure 8.18 : Correction of the red pupil extends over the iris, due to red pixels caused by image noise.

It is worthwhile to note that a strong lossy image compression (e.g., low-bitrate JPEG) may cause the same phenomenon: however, in the context of automatic algorithms which act just after the picture is taken, it is reasonable to assume that red eye removal is performed before image compression (to improve red eye detection and to avoid compressing twice).

Dead Eye

Sometimes red eyes are properly detected, but the corrected image just doesn't look natural. This may happen when a wrong luminance distribution, caused by reflected light, is kept through the correction and is evident in the resulting image. This may also happen when the color of the corrected pupils isn't quite natural, possibly because the correction isn't strong enough. Finally, the absence of the glint, which may be due to inpainting or excessive correction, may cause the eye to look "dead".

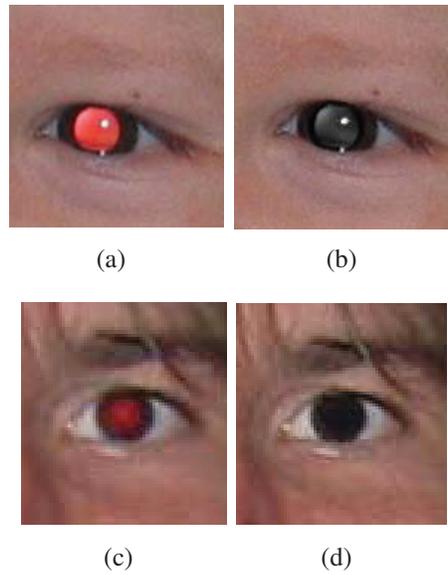


Figure 8.19 : In some cases, an unnatural luminance distribution is visible in the corrected image (b). Sometimes, instead, the eye has a “dead” look, due to the absence of the glint (d).

8.5 Quality Criteria

The formulation of a quality metric allows to choose the best solution and to adjust parameters of the algorithm in the best way. To achieve a quality control on red eye removal algorithm is a challenging issue. Usually the quality of the algorithm is estimated considering the ratio between corrected eyes and false positives. Obviously this is strictly related to the nature of the database and the quantity of images. Safonov [21] introduced an interesting quality metric that permit to remove correlation between quantity and quality.

First of all the author enumerated all possible cases, further he prioritized them using Analytic Hierarchy Process (AHP) [39]. Obviously a representative set of photos affected by red eye defect should be used for calculation of these unwanted cases. Furthermore good solutions must have low False Negatives (FN) and False Positives (FP), ideally FN and FP are equal to zero. However the severity of the False Positives differs significantly. Almost indistinguishable small FP on foreground is undesirable but sometimes allowable. Visible FP on foreground especially on human faces and bodies is absolutely not allowable; such FP artifacts damage photo more than red eyes. Therefore he divided False Positives in two classes: FP_c is the number of critical FP and FP_n is the number non-critical FP.

A similar situation is described for the False Negatives. Several red eye regions are relatively large and well distinguishable; other regions are small and have low local contrast. Detection of the first red eyes is defined as mandatory by Safonov, whereas detection of the second regions is desirable. Accordingly to such hypotheses he divided all FN in

two groups: FN_m is defined as the number of regions which are mandatory for detection; FN_d is the number of regions which are desirable for detection.

One more unwanted situation is the correction of only one eye from pair. For semi-automatic approaches it is not so crucial because users have possibility to correct the second eye manually, but for embedded implementations it is quite unpleasant. N_p is then defined as the number of faces with one corrected eye from pair of red eyes.

The retouching quality is important too. Regarding correction Safonov distinguished two cases: if the corrected eye looks worse of the original red eye, for example only part of the red region is corrected, it is an irritating case; it is noticeable that eye has been corrected but it does not irritate strongly. Accordingly C_I is the number of irritating cases and C_n is the number of situations when retouching is noticeable.

Table 8.1 : Analytic Hierarchy Process Table. Where the coefficients a_i , that refer to the assigned importance values in each row, are used to estimate the Geometric Mean. The weight is estimated in percentage from the sum of the Geometric Means (=9.60).

Req. Quality	FN_m	FN_d	FP_c	FP_n	N_p	C_i	C_n	$\sqrt[7]{\prod_{i=1}^7 a_i}$	Weight %
FN_m	1.00	5.00	0.20	5.00	1.00	0.20	5.00	1.26	13.13%
FN_d	0.20	1.00	0.33	5.00	0.20	0.20	5.00	0.68	7.08%
FP_c	5.00	3.00	1.00	3.00	5.00	5.00	5.00	3.43	35.73%
FP_n	0.20	0.20	0.33	1.00	0.20	0.20	1.00	0.34	3.54%
N_p	1.00	5.00	0.20	5.00	1.00	1.00	5.00	1.58	16.46%
C_i	5.00	5.00	0.20	5.00	1.00	1.00	5.00	1.99	20.72%
C_n	0.20	0.20	0.20	1.00	0.20	0.20	1.00	0.32	3.33%

As described above Safonov uses prioritization of the factors through AHP table (see 8.1) according to observers opinions. The simplest way for filling the table is: if left item is more important than top then cell is assigned to 5; if severity of the two items are the same then cell is set to 1; if top item is more important than left then cell is set to 1/5. Taking into account weights from AHP table, and taking into consideration a global weight of 10, for all the features, Safonov proposes the following quality criterion:

$$Q_c = \frac{N_t - 1.3 \cdot FN_m - 0.7 \cdot FN_d - 3.6 \cdot FP_c - 0.4 \cdot FP_n - 1.6 \cdot N_p - 2.1 \cdot C_i - 0.3 \cdot C_n}{N_t} \quad (8.17)$$

where N_t is total number of red eyes.

8.6 Future Trends

With research on red eye removal algorithms continuously advancing, more and more detection techniques adapt a two-step method, consisting in a first candidate extraction phase and a subsequent classification phase [10, 20, 21, 40, 41]. The candidate extraction phase detects the possible red eyes in the picture, and is much more permissive than a

stand-alone detection technique. This way, it detects a lot of false positives, but misses very few (if any) red eyes. The classification phase, then, is used to validate or reject each candidate, according to various features computed over the candidate patch. The problem of classifying image content is discussed in Chapter 10.

A new field of research which might prove of interest for red eye removal is the dark flash technology [42], which is a variant of the flash/no-flash technique employing a flash outside the visible spectrum of the light. This technology, mainly invented to avoid producing a disturbing burst of light each time a photograph is taken, might become crucial for prevention of the red eye artifacts. However, as it was invented very recently, it probably needs a good amount of development before it reaches acceptable performances over a wide range of real use cases.

Bibliography

- [1] J. M. Mir, "Apparatus & method for minimizing red-eye in flash photography," *U.S. Patent*, no. US4285588, 1981.
- [2] Adobe Photoshop. www.adobe.com/products/photoshop.
- [3] Corel Paint Shop Pro. www.jasc.com.
- [4] ACDSee. www.acdsee.com.
- [5] F. Gasparini and R. Schettini, *Single-Sensor Imaging: Methods and Applications for Digital Cameras*, ch. Automatic Red-Eye Removal for digital photography, pp. 429–457. CRC Press, 2008.
- [6] F. Gasparini and R. Schettini, "A review of redevye detection and removal in digital images through patents," *Recent Patents on Electrical Engineering*, vol. 2, no. 1, pp. 45–53, 2009.
- [7] S. Battiato, G. Farinella, M. Guarnera, G. Messina, and D. Ravì, "Red-eyes removal through cluster based linear discriminant analysis," in *International Conference on Image Processing (ICIP 2010)*, (Hong Kong), 2010.
- [8] S. Battiato, G. Farinella, M. Guarnera, G. Messina, and D. Ravì, "Boosting gray codes for red eyes removal," in *International Conference on Pattern Recognition (ICPR 2010)*, (Istanbul (TK)), 2010.
- [9] S. Battiato, G. Farinella, M. Guarnera, G. Messina, and D. Ravì, "Red-eyes removal through cluster based boosting on gray codes," *EURASIP Journal on Image and Video Processing, Special Issue on Emerging Methods for Color Image and Video Quality Enhancement*, 2010. To appear.
- [10] L. Zhang, Y. Sun, M. Li, and H. Zhang, "Automated red-eye detection and correction in digital photographs," in *IEEE International Conference on Image Processing (ICIP-04)*, pp. 2363–2366, 2004.
- [11] A. Held, "Model-based correction of red-eye defects," in *IS&T Color Imaging Conference (CIC-02)*, pp. 223–228, 2002.
- [12] M. Gaubatz and R. Ulichney, "Automatic red-eye detection and correction," in *IEEE International Conference on Image Processing (ICIP-02)*, vol. 1, (Rochester, New York, USA), pp. 804–807, 2002.
- [13] B. Smolka, K. Czubin, J. Y. Hardeberg, K. N. Plataniotis, M. Szczepanski, and K. W. Wojciechowski, "Towards automatic redevye effect removal," *Pattern Recognition Letters*, vol. 24, no. 11, pp. 1767–1785, 2003.

- [14] F. Gasparini and R. Schettini, "Automatic redevye removal for smart enhancement of photos of unknown origin," in *Visual Information and Information Systems (VISUAL-2005)*, vol. 3736 of *Lecture Notes in Computer Science*, pp. 226–233, 2005.
- [15] J. Willamowski and G. Csurka, "Probabilistic automatic red eye detection and correction," in *IEEE International Conference on Pattern Recognition (ICPR-06)*, pp. 762–765, 2006.
- [16] A. J. Patti, K. Konstantinides, D. Tretter, and Q. Lin, "Automatic digital redevye reduction," in *IEEE International Conference on Image Processing (ICIP-98)*, vol. 1, (Chicago, Illinois), pp. 55–59, 1998.
- [17] P. Benati, R. Gray, and P. Cosgrove, "Automated detection and correction of eye color defects due to flash illumination," *U.S. Patent*, no. US5748764, 1998.
- [18] F. Volken, J. Terrier, and P. Vandewalle, "Automatic red-eye removal based on sclera and skin tone detection," in *Third European Conference on Color in Graphics, Imaging and Vision (CGIV-06)*, pp. 359–364, Society for Imaging Science and Technology, 2006.
- [19] A. M. Ferman, "Automatic detection of red-eye artifacts in digital color photos," in *IEEE International Conference on Image Processing (ICIP-08)*, vol. 1, (San Diego, California), pp. 617–620, 2008.
- [20] H. Luo, J. Yen, and D. Tretter, "An efficient automatic redevye detection and correction algorithm," in *IEEE International Conference on Pattern Recognition (ICPR-04)*, pp. 883–886, 2004.
- [21] I. V. Safonov, "Automatic red-eye detection," in *GraphiCon, International Conference on Computer Graphics & Vision (GraphiCon-07)*, (Moscow, Russia), 2007.
- [22] J. S. Schildkraut and R. T. Gray, "A fully automatic redevye detection and correction algorithm," in *IEEE International Conference on Image Processing (ICIP-02)*, vol. 1, (Rochester, New York, USA), pp. 801–803, 2002.
- [23] M.-H. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 34–58, Jan 2002.
- [24] R. Hsu, M. Abdel-Mottaleb, and A. Jain, "Face detection in color images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 696–706, 2002.
- [25] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

- [26] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *International Conference on Image Processing (ICIP 2002)*, vol. 1, pp. I-900–I-903 vol.1, 2002.
- [27] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *European Conference on Computational Learning Theory (EuroCOLT 1995)*, pp. 23–37, Springer-Verlag, 1995.
- [28] R. Schapire, "The boosting approach to machine learning: An overview," in *MSRI Workshop on Nonlinear Estimation and Classification*, 2001.
- [29] L. Hongliang, K. Ngan, and L. Qiang, "Faceseg: Automatic face segmentation for real-time video," *IEEE Transactions on Multimedia*, vol. 11, pp. 77–88, Jan. 2009.
- [30] S. Phung, A. Bouzerdoun, and D. Chai, "Skin segmentation using color pixel classification: analysis and comparison," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 148–154, Jan. 2005.
- [31] N. O'Hare and A. Smeaton, "Context-aware person identification in personal photo collections," *IEEE Transactions on Multimedia*, vol. 11, pp. 220–228, Feb. 2009.
- [32] Google, "Picasa - name tags," 2009. website: <http://picasa.google.com>, video demo: <http://www.youtube.com/watch?v=teeGF-w5Cpw>.
- [33] Face.com, "Photofinder - face recognition application for Facebook.," 2009. website: <http://www.face.com>, blog: <http://blog.face.com>.
- [34] L. Marchesotti, M. Bressan, and G. Csurka, "Safe red-eye correction plug-in using adaptive methods," in *International Conference on Image Analysis and Processing - Workshops (ICIAPW-07)*, pp. 192–165, 2007.
- [35] J. Y. Hardeberg, "Red eye removal using digital color image processing," in *Image Processing, Image Quality, Image Capture Systems Conference (PICS-01)*, (Montréal, Canada), pp. 283–287, 2001.
- [36] S. Yoo and R.-H. Park, "Red-eye detection and correction using inpainting in digital photographs," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 3, pp. 1006–1014, 2009.
- [37] X.-P. Miao and T. Sim, "Automatic red-eye detection and removal," in *IEEE International Conference on Multimedia and Expo (ICME-04)*, (Taipei, Taiwan), pp. 1195–1198, 2004.
- [38] G. Petschnigg, R. Szeliski, M. Agrawala, M. F. Cohen, H. Hoppe, and K. Toyama, "Digital photography with flash and no-flash image pairs," *ACM Transactions on Graphics, SIGGRAPH 2004 Conference Proceedings*, vol. 23, no. 3, pp. 664–672, 2004.

- [39] T. L. Saaty, *Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World*, vol. 2. Analytic Hierarchy Process Series, 2001. New Edition.
- [40] S. Ioffe, “Red eye detection with machine learning,” in *IEEE International Conference on Image Processing (ICIP-03)*, vol. 2, (Barcelona, Catalonia, Spain), pp. 871–874, 2003.
- [41] S. Battiato, M. Guarnera, T. Meccio, and G. Messina, “Red eye detection through bag-of-keypoints classification,” in *International Conference on Image Analysis and Processing (ICIAP-09)*, pp. 528–537, 2009.
- [42] D. Krishnan and R. Fergus, “Dark flash photography,” *ACM Transactions on Graphics, SIGGRAPH 2009 Conference Proceedings*, vol. 28, no. 3, pp. 1–11, 2009.

Video Stabilization

T. Meccio, G. Puglisi

Image Processing Lab, University of Catania, Italy.

G. Spampinato

Advanced System Technology - Catania Lab, STMicroelectronics, Italy.

Abstract: To make a high quality video with a hand-held camera is a very difficult task. The unwanted movements of our hands typically blur and introduce disturbing jerkiness in the recorded video. Moreover this problem is amplified when a zoom lens or a digital zoom is used. To solve this problem many video stabilization techniques have been developed. Optical based approaches measure camera shake and control the jitter acting on lens or on the CCD/CMOS sensor. On the other hand digital video stabilization techniques make use only of information drawn from images and do not need any additional hardware tools. This Chapter describes the algorithms typically involved in the video stabilization pipeline (motion estimation, unwanted movement detection, frame warping) highlighting their issues and weak points.

9.1 Introduction

In the last decade multimedia devices (PDAs, mobile phones, etc.) have been dramatically diffused. Moreover the increasing of their computational performances combined with a higher storage capability permits them to elaborate large amount of data. These devices, typically small and thin, usually have video acquisition capability. However making a stable video with these devices is a very challenging task especially when a zoom lens or a digital zoom is used. Due to user's hands shaking, the recorded videos suffer from annoying perturbations. The same problem arises in presence of cameras placed on mobile supports (car, airplane) or fixed cameras operating outdoors. The atmospheric conditions (e.g., the wind) and the vibrations created by passing vehicles make the recorded video unstable.

Video stabilization allows to acquire video sequences without disturbing jerkiness, removing unwanted camera movements. Video quality is then improved and the higher level algorithms present in the device (segmentation, tracking, recognition) can also work properly [1, 2]. Moreover higher bit rate compression can be obtained from the stabilized video with respect to the unstable one.

Many stabilization approaches have been developed [3, 4]. Some techniques, by using mechanical or electronic tools, measure camera shake and then control the jitter acting on lens (OIS - Optical Image Stabilization) or on the CCD/CMOS sensor. In these approaches both steps are applied before the acquisition avoiding any post-processing computation and image deformation with the cost of some extra mechanical or optical devices. OIS has no computational complexity (all the steps are applied in the lens before the acquisition) but requires high expensive optical systems and enough space around the camera lens, making the integration in very small and thin systems, like imaging phones, difficult. On the other hand, digital video stabilization techniques make use only of infor-

mation retrieved from the analysis of the video to estimate physical camera motion without any additional mechanical apparatus but with some extra computational cost and the risk of generating image deformation. However, these approaches may be implemented easily both in real-time and post-processing systems.

9.2 Digital Approaches

Digital video stabilization algorithms, in general, are made up of three stages (Fig.(9.1)): motion estimation, unwanted movement detection and image warping (see [5] for an alternative scheme). The first stage is devoted to find the parameters relative to the transformation occurred between adjacent frames. Translational, similarity and affine are the most common adopted motion models. The second step discriminates intentional motion (e.g., panning) from the unwanted motion (jitter). Typically motion smoothness considerations are taken into account in this process (jitter is a high frequency signal). The final step consists of the reconstruction of the stabilized image through a proper warping. In order to do so, captured sequences must be larger than the final produced video.



Figure 9.1 : Video stabilization algorithms are made up of three steps: motion estimation, unwanted movement detection, image warping.

9.2.1 Motion Estimation

The problem of motion estimation/image registration has been widely investigated and many solutions have been proposed. The existing approaches can be classified in two categories: direct methods [6] and feature based methods [7]. The former techniques aim to recover the unknown parameters through global minimization criteria based on direct image information. Some assumptions (e.g., brightness constancy) are typically used as starting point. On the contrary feature based approaches first locate reliable features in the image and then estimate the motion parameters considering their correspondences.

A lot of interframe transformation models have been used in the video stabilization field. Due to the real time constraints of the embedded devices many approaches consider

only a 2D translational model. This model associates a point (x_i, y_i) in frame I_n with a point (x_f, y_f) in frame I_{n+1} with the following transformation:

$$\begin{aligned}x_f &= x_i + T_x \\y_f &= y_i + T_y\end{aligned}\quad (9.1)$$

where T_x and T_y are x-axis and y-axis shifts.

Even if the accuracy of these approaches is limited by the simplicity of the model, they are typically cheap and can be made robust without too much effort. Moreover the compensation step can be trivially implemented (it is a simple cropping) without performing any kind of interpolation.

Recently, thanks to the increasing computational resources of embedded devices, more advanced models have been considered. A 2D similarity model has been adopted as a good trade-off between complexity and computational costs:

$$\begin{aligned}x_f &= x_i \lambda \cos \theta - y_i \lambda \sin \theta + T_x \\y_f &= x_i \lambda \sin \theta + y_i \lambda \cos \theta + T_y\end{aligned}\quad (9.2)$$

where λ is the zoom parameter, θ the rotation angle, T_x and T_y respectively x-axis and y-axis shifts.

This model permits to obtain better performances in terms of accuracy: rotational jitter is taken into account. Moreover it is also reliable in presence of zooming or forward walking of the user (scale parameter is not present in a simple 2D translational model). However, the design of fast and robust approaches is not a trivial task. Finally, motion compensation has to be performed by means of interpolation.

Whatever the model used, the motion estimation algorithms have to be able to deal with critical conditions. Very often, in real videos, there are many conditions that degrade, if not properly managed, the performances of video stabilization algorithms. In presence of homogeneous regions (Fig.(9.2(a))), periodic patterns (Fig.(9.2(b))) [8] and fast illumination changes, the local motion estimators sometimes produce wrong vectors. Moreover, the movement of the objects in the scene (Fig.(9.3)) can mislead the global motion vector estimation. Although their vectors are correctly computed they do not describe camera movements. Finally, zooming and forward walking of the user can create some problem if they are not taken into account in the motion model (e.g., a translational motion model).

Block Based Approaches

Many approaches compute frame alignment considering image intensity values. Some techniques (*block based*) first divide the image in blocks, typically square, and then search the corresponding ones in the next frame. Matching is performed within a search window minimizing a proper error function (Fig.(9.4)).

A lot of video stabilization approaches based on block matching motion estimation have been developed [9–15]. These approaches typically, taking into account the limited block size and the high frame rate, assume only a translational motion for each block. This information, coming from different spatial locations in the frame, is then used to

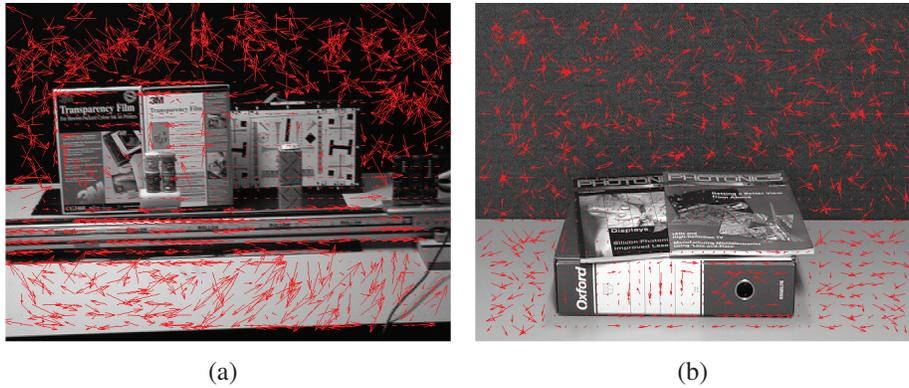


Figure 9.2 : Local motion estimator sometimes fails in presence of homogeneous regions **(a)** and periodic pattern **(b)**.

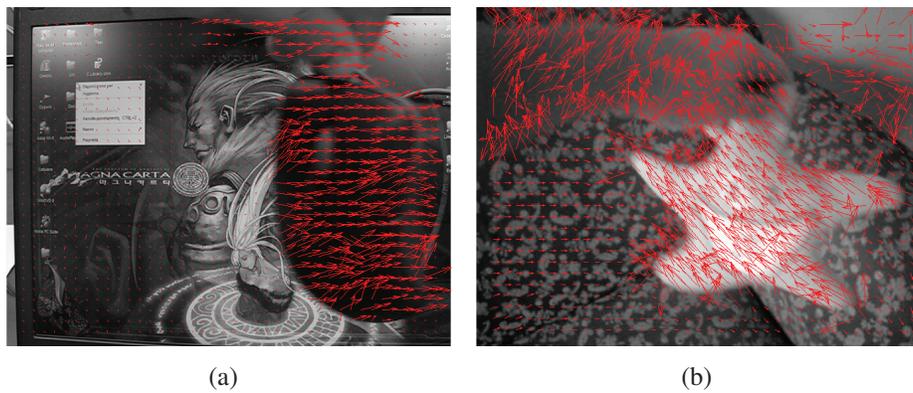


Figure 9.3 : Although correctly computed, motion vectors corresponding to moving objects, mouse **(a)** and soft star **(b)**, do not describe overall frame motion and can deceive video stabilization algorithms.

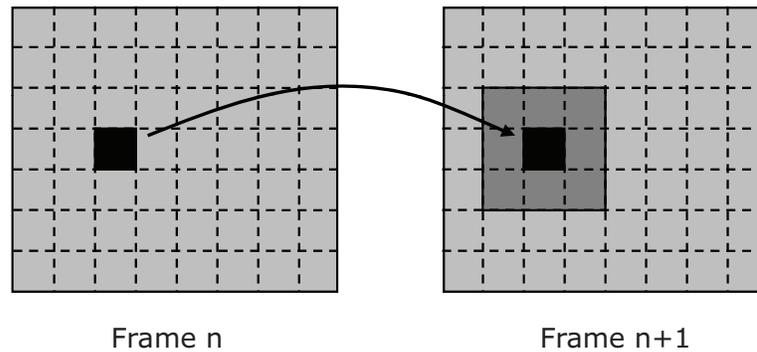


Figure 9.4 : Block matching approaches split the image in square blocks (in black) and search the corresponding ones in the next frame. The matching procedure is performed within a search window (dark gray).

compute (through Least Squares) the global motion vector (e.g., two translations, one zoom factor, one rotation in the similarity model). This methodology just filtering out outliers works pretty well.

In the following Section one of these approaches will be described in detail.

Block Based Approaches: a Case Study

In this Section a block based motion estimation algorithm will be described [14–16]. The main ideas involved in the described approach (e.g., vectors filtering criteria) have been used, although in a different way, in many other approaches [9, 11, 17, 18].

The algorithm proposed in [15] consists of several steps. First of all a *BMA* (*Block Matching Algorithm*) module, starting from a pair of consecutive frames, computes the local motion vectors (through block matching). These vectors are then filtered by *Pre-filtering* module through some simple rejection rules based on the goodness of matching, block homogeneity and vectors similarity with the surrounding blocks. The information retrieved in the previous iteration is used by the *Memory Filter* step to perform further filtering. Afterward *Robust Estimator* estimates the inter-frame transformation parameters through some Least Squares iterations. Finally some error measures, to be used by *Memory Filter*, are computed by *Error Matrix Computing* module. In Fig.(9.5) an example of vector filtering is reported. Only reliable motion vectors are retained.

The Pre-filtering module filters the unreliable local motion vectors generated by the *BMA* module (which typically computes many wrong motion vectors). These errors usually depend on the scene content (i.e., homogeneous regions, periodic patterns) and the number of wrong motion vectors usually increases using an approximate BMA [19, 20]. These algorithms, in order to speed up the matching process, consider only a part of the search space; hence they are likely to be misled by local minima. To filter out these vectors, not useful for global motion estimation, the authors apply the following consid-

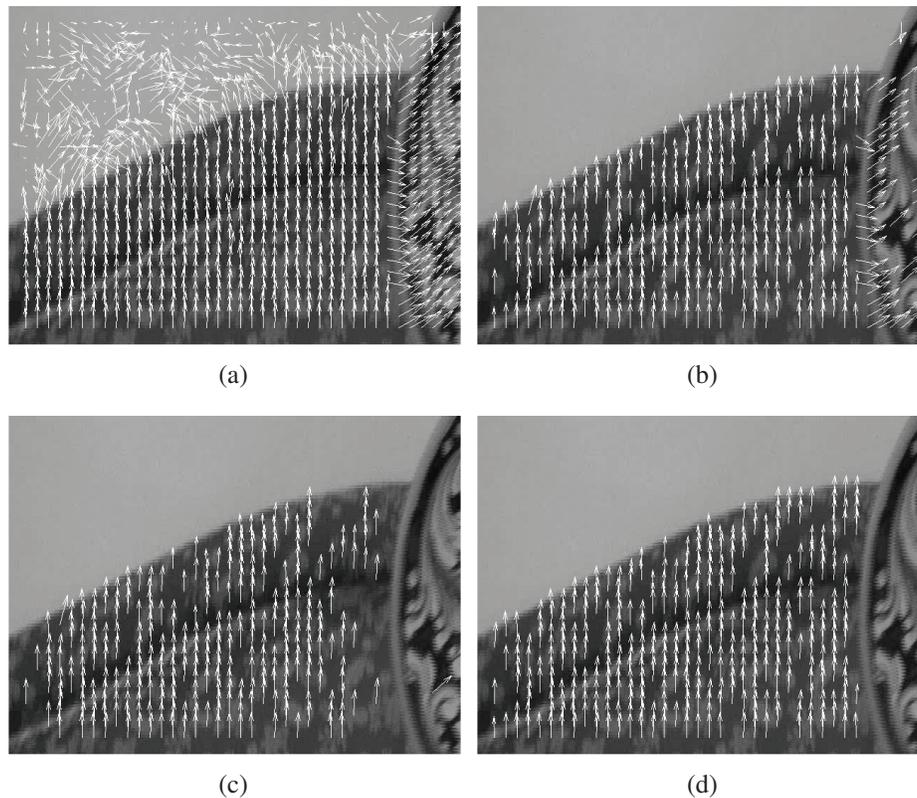


Figure 9.5 : An example of the filtering process performed by [15]. First, the vectors produced by BMA **(a)** are filtered out considering local blocks similarity, local blocks "activity" and matching effectiveness **(b)**. Even if many wrong vectors have been filtered out, some still remain (the big object on the right). A second filtering step based on the blocks history is typically able to solve this kind of problem **(c)**. Finally a robust estimation process obtains a set of reliable motion vectors **(d)** used for the inter-frame parameter estimation.

erations: the *SAD* (sum of absolute differences) values have to be low (effective match); local motion vectors have similar values in their neighbourhood (*NS*); local motion vectors referred to homogeneous blocks are not reliable (*Unhom*).

The corresponding indexes are:

$$SAD = \sum_{h=0}^{H-1} \sum_{k=0}^{W-1} |B_1(h,k) - B_2(h,k)| \quad (9.3)$$

where B_1 and B_2 are two corresponding blocks of $W \times H$ size.

$$NS(i,j) = \frac{1}{8} \sum_{k=-1}^1 \sum_{h=-1}^1 |Mv_x(i,j) - Mv_x(i+k,j+h)| \\ + |Mv_y(i,j) - Mv_y(i+k,j+h)| \quad (9.4)$$

where $Mv_x(i,j)$ and $Mv_y(i,j)$ are the components (along x and y axes respectively) of local motion vector centered in (i,j) .

$$Unhom_x = \sum_{h=0}^{H-1} \sum_{k=1}^{W-1} |B(h,k) - B(h,k-1)| \quad (9.5)$$

$$Unhom_y = \sum_{h=1}^{H-1} \sum_{k=0}^{W-1} |B(h,k) - B(h-1,k)| \quad (9.6)$$

where B is a block of $W \times H$ size.

According to the (9.3), (9.4), (9.5) and (9.6) all the vectors are properly filtered. These filterings remove all the wrong vectors computed by a generic block matching algorithm. However, due to moving objects in the scene, there are vectors correctly computed by *BMA* (hence not filtered by *Pre-filtering* module) that must be deleted in order to have a good inter-frame parameters estimation. If there are big moving objects in the scene, single image information is not enough to filter out these vectors and the output sequence will follow the moving object.

In order to make the algorithm robust to such situations, *Memory Filter* module uses previous frame information by propagating the computed error through consecutive frames (wrong regions at frame t are wrong at frame $t+1$ with a high probability).

All the steps described above have been designed to remove outliers. The remaining vectors are then used to compute interframe transformation parameters. The authors of [15] use a similarity model (9.2). Considering n motion vectors the following over-constrained linear system can be obtained:

$$A \cdot p = b \quad (9.7)$$

$$A = \begin{pmatrix} x_{i1} & -y_{i1} & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ x_{in} & -y_{in} & 1 & 0 \\ y_{i1} & x_{i1} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ y_{in} & x_{in} & 0 & 1 \end{pmatrix}$$

$$p = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} \quad b = \begin{pmatrix} x_{f1} \\ \vdots \\ x_{fn} \\ y_{f1} \\ \vdots \\ y_{fn} \end{pmatrix}$$

where $p_1 = \lambda \cos \theta$, $p_2 = \lambda \sin \theta$, $p_3 = T_x$, $p_4 = T_y$.

Vector computation may be affected by noise so it is useful to apply a linear Least Squares Method on the set of redundant equations to obtain the parameters vector.

$$p = (A^t \cdot A)^{-1} \cdot A^t \cdot b \quad (9.8)$$

All the similarity model parameters λ , θ , T_x , T_y can be easily derived from p vector components in the following way:

$$\lambda = \sqrt{p_1^2 + p_2^2} \quad (9.9)$$

$$\theta = \arctan\left(\frac{p_2}{p_1}\right) \quad (9.10)$$

$$T_x = p_3 \quad (9.11)$$

$$T_y = p_4 \quad (9.12)$$

The whole set of local motion vectors probably includes wrong matches or correct matches belonging to moving objects in the scene. Obviously, there are some correct pairs that represent real camera shakes but several points are not related to such information. Least Squares Method does not perform well when there is a large portion of outliers with respect to the total number of vectors. However, outliers can be identified and filtered out of the estimation process obtaining a better accuracy. In order to cope with outliers the authors of [15] perform two least squares iteration. The parameters $(\lambda, \theta, T_x, T_y)$ obtained by the first iteration are used to evaluate the goodness of the vectors. Two error measures have been taken into account: the square Euclidean distance between estimated and measured motion vectors and the square of the cosine of the angle between them (Fig.(9.6)). The combination allows obtaining a reliable filtering of the outliers.

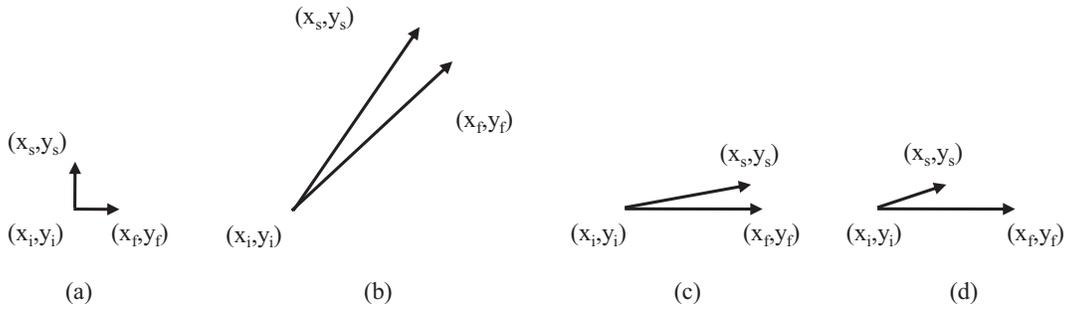


Figure 9.6 : Euclidean distance is the same for (a) and (b) but different for (c) and (d). Conversely the cosine distance is different for (a) and (b) but the same for (c) and (d). Both distances sometimes fail. However, combining them together the overall system works pretty well.

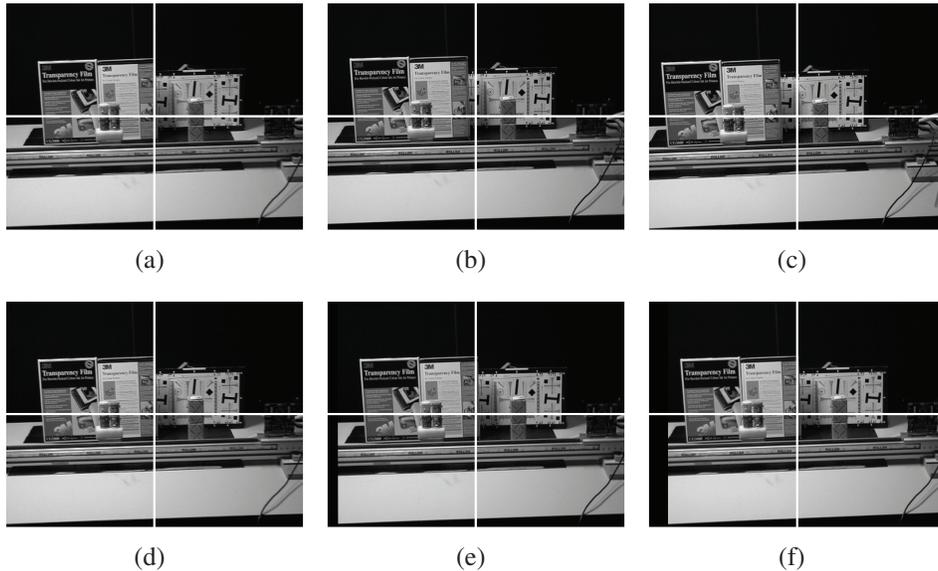


Figure 9.7 : Original (a, b, c) and stabilized frames (d, e, f) taken from a video with illumination changes and zoom. The grid is overlaid for better visualization.

Finally *Error Matrix Computing* computes an error measure (Euclidean distance) to be used by *Memory Filter*. Some examples of stabilized videos by [15] have been reported in Fig.(9.7), Fig.(9.8), Fig.(9.9).

Feature Based Approaches

Feature based techniques allow to directly retrieve the global motion vector through the analysis of a particular feature.

Censi *et al.* [21] extract corners like points [22] from the first frame and track them

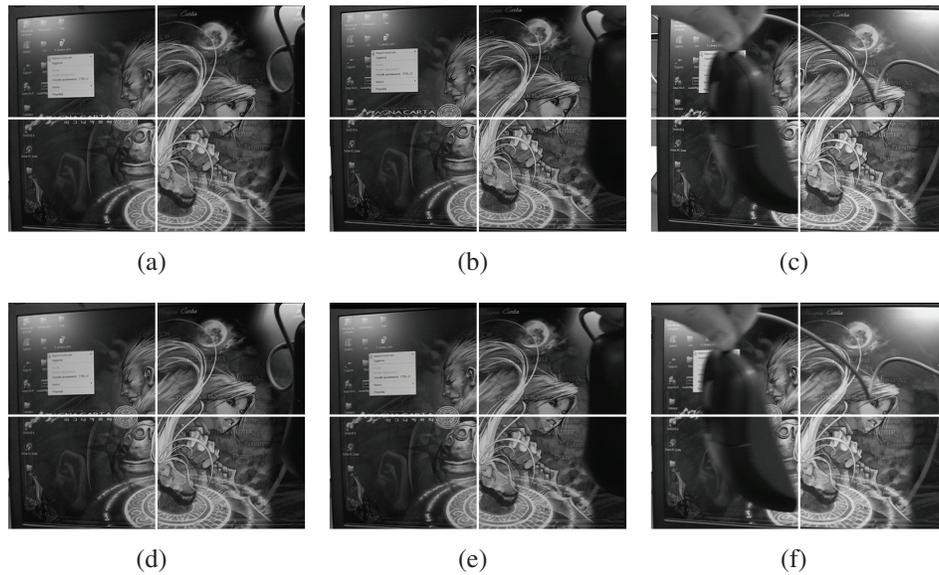


Figure 9.8 : Original (a, b, c) and stabilized frames (d, e, f) taken from a video with a big moving object. The grid is overlaid for better visualization.

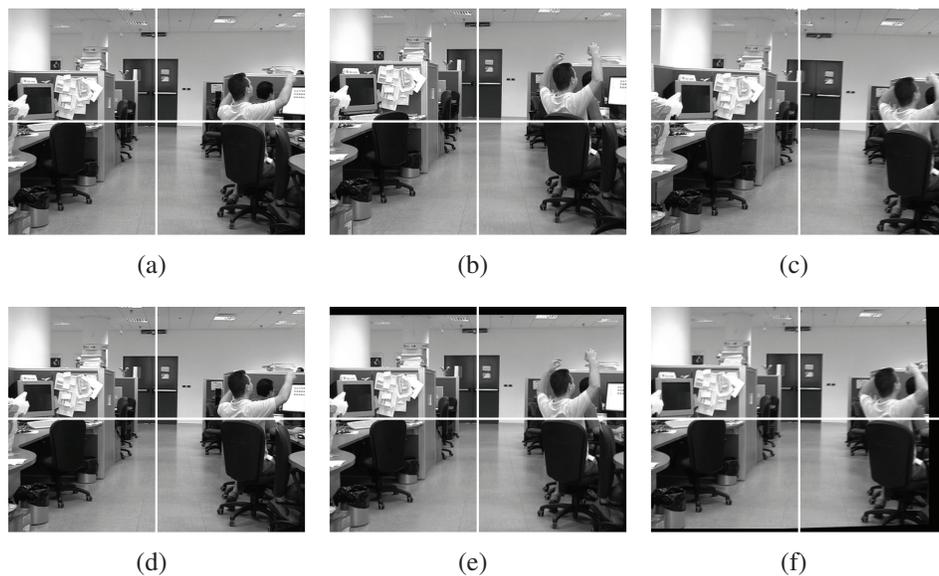


Figure 9.9 : Original (a, b, c) and stabilized frames (d, e, f) taken from a video with forward walking of the user. The grid is overlaid for better visualization.

in the following frames. In order to predict local features trajectory, the authors use a constant velocity Kalman filter. Moreover, to cope with tracker failures or moving objects, they use a robust rejection rule (X84 [23]) to discard outliers. Finally, the matching is performed considering a linear projective transformation (homography) between the corresponding filtered points.

The method proposed in [24] uses SIFT features to estimate interframe motion. First of all, it discards very long local motion vectors produced by SIFT matching. Subsequently it computes the interframe transformation parameters (scale, rotation, x and y shifts) with Least Squares estimation. In order to deal with outliers and obtain better performance, they refine the estimated values through a particle filter. The weighting parameters of the filter are computed through both intensity and edge based similarity between estimated and reference images.

Jimenez & Salas [25] developed a stabilization algorithm for cameras placed on poles in outdoor environments. They extract corner-like features from the image and compute a first rough transformation (homography) via Least Squares. An iterative refining step based on the Euclidean distance between estimated and measured point position is then performed. Finally, the quality of the computed homography is evaluated with some criteria based on spatial distribution of features.

Del Bianco *et al.* [26] compute reliable points through fast feature-based motion estimation techniques (FFME) [27]. These features are robust with respect to noise, aperture problem, illumination changes and small variations of 3D viewpoint. Abrupt rotations and scale variations are not taken into account: due to the high temporal correlation they are not likely to be in the videos. The authors compute interframe transformation parameters through a combination of RANSAC (RANdom SAMple Consensus) [28] and LMedS (Least Median of Squares) [29].

Adams *et al.* [30] propose a real time algorithm for viewfinder alignment. First of all, they compute four 1-D projections of the edge image. Afterward, aligning the projections coming from consecutive frames, they obtain a first estimation of the translational shift parameters. Corner points are then extracted and, through translational information, a simplified matching is performed. Finally Least Squares compute the four interframe transformation parameters (rotation, zoom, translations).

9.2.2 Unwanted Movement Detection

One of the main challenges in video stabilization is distinguishing between voluntary and involuntary movements, and only correcting the latter. Voluntary movements are slower and more regular, thus they can be filtered out using a High-Pass filter. To this end, commonly used filters are Infinite Impulse Response filters and Kalman filters.

Infinite Impulse Response Filters

Infinite Impulse Response (IIR) filters are filters whose response to an impulse is infinite over time [31]. An IIR filter can be expressed in the general form:

$$H(z) = \frac{\sum_{l=0}^{L-1} b_l z^{-l}}{1 + \sum_{m=1}^M a_m z^{-m}} \quad (9.13)$$

In this representation, a filter is identified by the coefficients b_l and a_m , and designing one means finding proper coefficients so that the transfer function $H(z)$ satisfies the needed specifications.

Given a transfer function, there are several configurations in which it can be implemented. One is the direct form-I, defined by the Input/Output (I/O) equation:

$$y(n) = \sum_{l=0}^{L-1} b_l x(n-l) - \sum_{m=1}^M a_m y(n-m) \quad (9.14)$$

For example, the direct form-I equation associated to the following filter:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (9.15)$$

is:

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) - a_1 y(n-1) - a_2 y(n-2) \quad (9.16)$$

An IIR filter can also be represented as two cascaded filters $H_1(z)$ and $H_2(z)$:

$$H(z) = H_1(z)H_2(z) \quad (9.17)$$

or equivalently:

$$H(z) = H_2(z)H_1(z) \quad (9.18)$$

where:

$$H_1(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} \quad (9.19)$$

and

$$H_2(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (9.20)$$

IIR filters are sensitive to quantization effects, mainly due to their feedback nature. When applied to unwanted movement filtering, they can show some oscillations immediately after the end of a voluntary movement. To resolve this problem, a constraint may be imposed on the filter, forcing it to stop when the input taps are zeroes. Usually a low-order filter (e.g., second-order) is chosen for the lower group delay and taps requirements (increasing performance).

Kalman Filter

The Kalman filter is a method to recursively solve the problem of discrete data linear filtering [32, 33]. More specifically, it estimates the state $x \in R^n$ of a process governed by the linear stochastic difference equation:

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \quad (9.21)$$

through a measurement $z \in R^m$ defined as:

$$z_k = Hx_k + v_k \quad (9.22)$$

The matrix A represents the way each state affects the subsequent one. The column vector B represents the relationship between the optional control input $u \in R$ and the process state. The matrix H represents the model which maps the state space into the observation space. The random variables w_k and v_k represent, respectively, the process noise and measurement noise, which are assumed to be normally distributed and independent from each other:

$$p(w) \cong N(0, Q) \quad (9.23)$$

$$p(v) \cong N(0, R) \quad (9.24)$$

Given the above assumptions, two estimates of the process state at each step k may be defined: an a priori state estimate \bar{x}_k , based on the known information about the process prior to step k , and an a posteriori state estimate \tilde{x}_k , based on the measurement z_k . According to these definitions, the a priori estimate error is $\bar{e}_k \equiv x_k - \bar{x}_k$ and the a posteriori estimate error is $e_k \equiv x_k - \tilde{x}_k$.

The *a priori* estimate error covariance is then:

$$\bar{P}_k = E[\bar{e}_k \bar{e}_k^T] \quad (9.25)$$

and the *a posteriori* estimate error covariance is:

$$P_k = E[e_k e_k^T] \quad (9.26)$$

To obtain the equations for the Kalman filter, the first step is to define an a posteriori state estimate as a linear combination of the a priori estimate and a weighted difference between the actual measurement and a measurement prediction:

$$\tilde{x}_k = \bar{x}_k + K(z_k - H\bar{x}_k) \quad (9.27)$$

The difference $(z_k - H\bar{x}_k)$ is called measurement innovation or residual. It represents the disparity between the prediction $H\bar{x}_k$ and the actual measurement z_k . A zero residual means perfect agreement between the two. The matrix K represents the gain or blending factor and is chosen to minimize the a posteriori error covariance (9.26).

To perform this minimization, it is possible to substitute (9.27) into the definition of e_k and then substitute the resulting formula into (9.26). Differentiating the trace of the resulting covariance with respect to K , imposing that the resulting derivative be zero, and then solving for K , the resulting formula is:

$$K_k = \bar{P}_k H^T (H \bar{P}_k H^T + R)^{-1} \quad (9.28)$$

The matrix R represents the measurement error covariance. As it becomes smaller, the gain K increases, thus the residual is weighed more. Specifically, $\lim_{R_k \rightarrow \infty} K_k = H_k^{-1}$. The residual is weighed less, instead, as the a priori estimate error covariance \bar{P}_k becomes closer to zero: $\lim_{\bar{P}_k \rightarrow \infty} K_k = 0$. In this way, the factor K is used to balance the “trust” between the actual measurement z_k and the predicted measurement $H\bar{x}_k$.

Discrete Kalman Filter Algorithm

The Kalman filter estimates the evolution of process states over time, obtaining feedback in the form of (possibly noisy) measurements. The equations governing the filter are of two kinds: time update equations and measurement update equations. The former compute the a priori estimate based on the current state, while the latter provide feedback by including the new measurement to obtain an improved a priori estimate. Together, they form a set of equations similar to those of a predictor-corrector algorithm for numerical problems.

The discrete Kalman filter time update equations are:

$$\bar{x}_k = A\tilde{x}_{k-1} + Bu_k \quad (9.29)$$

$$\bar{P}_k = AP_{k-1}A^T + Q \quad (9.30)$$

where matrices A and B are from (9.21) and Q is from (9.23).

The discrete Kalman filter measurements update equations are instead:

$$K_k = \bar{P}_k H^T (H \bar{P}_k H^T + R)^{-1} \quad (9.31)$$

$$\tilde{x}_k = \bar{x}_k + K(z_k - H\bar{x}_k) \quad (9.32)$$

$$P_k = (I - K_k H) \bar{P}_k \quad (9.33)$$

(9.31) and (9.32), which were discussed above ((9.28) and (9.27)), respectively compute the Kalman gain and, given the new measurement z_k , generate the new a posteriori estimate. (9.33) calculates the a posteriori estimate error covariance.

After each update cycle, the process is repeated with the previous a posteriori estimate used to predict the new a priori estimate.

Motion Filtering with Kalman

As stated above, the Kalman filter has been used for *Motion Filtering* [34,35]. The global displacement of the camera along each of the axes can be expressed as (constant velocity model):

$$x_n = Ax_{n-1} + w_n \quad (9.34)$$

$$z_n = c^T x_n + u_n \quad (9.35)$$

where

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad c = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

and z_n is the global movement (estimated through a global motion estimator), $c^T x_n$ is the voluntary motion, w_n and u_n process and measurement noise terms.

Ideally, at each frame, a corrective displacement of $-u$ pixels is needed in order to stabilize it. Kalman update equations ((9.29) to (9.33)) are used to filter the motion. The involuntary motion is estimated as:

$$u = z - c^T x_n \quad (9.36)$$

For reasons related to the size of the acquired frame, a constraint is imposed so that the magnitude of the correction does not exceed a certain value d . When this is not the case, the state vector x is updated as follows:

$$x = x + \text{sign}(u)(|u| - d)Pc(c^T Pc)^{-1} \quad (9.37)$$

So that the correction respects the constraint.

9.2.3 Frame Warping

After the estimation of the roto-translational parameters, each frame must be corrected in order to obtain the stabilized sequence. Correction is achieved through frame warping, which is mathematically defined by a mapping from the space (u, v) of source image coordinates to the space (x, y) of destination image coordinates.

According to the choice of dependent and independent variables, mappings are divided into *forward mappings* and *inverse mappings* [36].

Forward Mapping

Forward mapping expresses destination coordinates as functions of source coordinates:

$$x = x(u, v); \quad y = y(u, v) \quad (9.38)$$

In this case, pixels in the source image are directly mapped into the destination image (Fig.(9.10)).

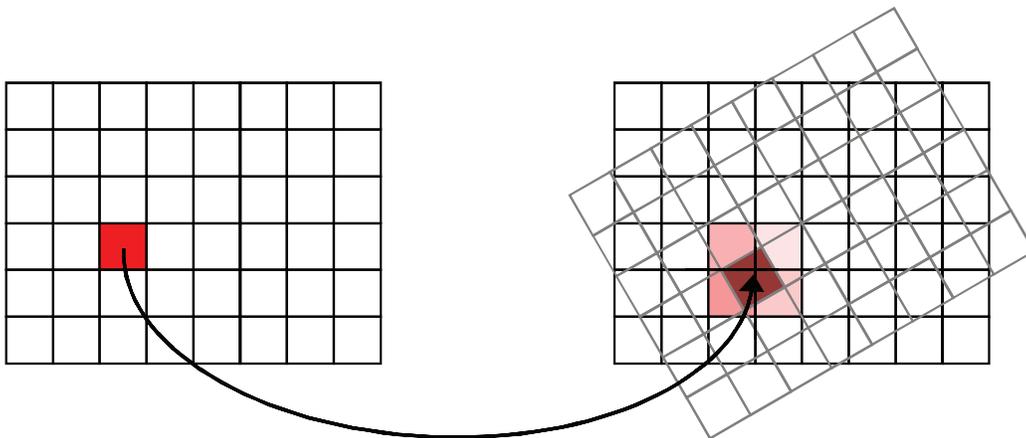


Figure 9.10 : Forward mapping with bilinear interpolation. In forward mapping the value of each pixel in the source image is spread over the nearest pixels in the destination image.

Forward mapping has several drawbacks. Source pixels very seldom map perfectly to destination pixels, since coordinates may not be integer values, and the shape of the pixels may not fit due to rotation. This means that the value of each source pixel may need to be distributed among more destination pixels, which isn't trivial to do since each destination pixel, in turn, would receive contributions from more source pixels. Improper distribution of color contributions in forward mapping may produce undesirable artifacts, like non-uniform intensity and/or holes, in the destination image. To overcome these problems, inverse mapping is usually preferred.

Inverse Mapping

Inverse mapping expresses source coordinates as functions of destination coordinates:

$$u = u(x, y); \quad v = v(x, y) \quad (9.39)$$

In this case, for each destination pixel, the corresponding coordinates in the source image are obtained from the mapping (Fig.(9.11)).

As already seen for forward mapping, the obtained coordinates may not map exactly to pixels in the source image. However, this can be addressed by means of pixel interpolation, since each destination pixel can be computed independently. *Nearest Neighbour* interpolation is usually not preferred, as it introduces evident aliasing effects, like jagged edges and blocky zones. *Bilinear Interpolation* is preferred, even if it tends to slightly blur the image.

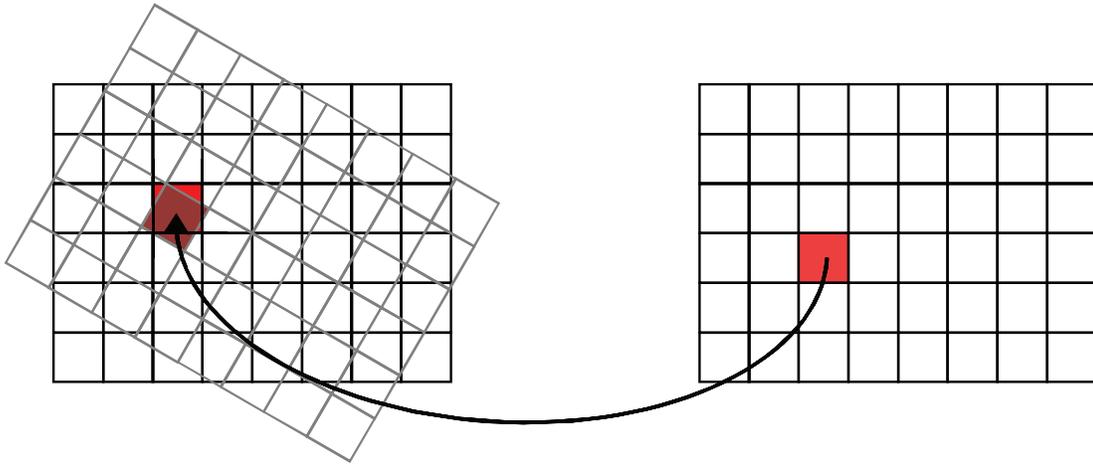


Figure 9.11 : Inverse mapping with bilinear interpolation. In inverse mapping the value of each pixel in the destination image is interpolated from the nearest pixels in the original image.

9.3 Conclusions

Digital video stabilization allows to acquire video sequences without disturbing jerkiness by removing from the image sequence the effects caused by unwanted camera movements. In this Chapter the main steps involved in video stabilization systems have been described in details: motion estimation (block based and feature based approaches), unwanted movement detection (IIR and Kalman filter), frame warping (forward and inverse mapping).

Bibliography

- [1] L. Mercenaro, G. Vernazza, and C. Regazzoni, "Image stabilization algorithms for video-surveillance application," in *Proceedings of International Conference on Image Processing (ICIP-01)*, pp. 349–352, Thessaloniki (Greece), 2001.
- [2] P.-M. Jodoin, J. Konrad, V. Saligrama, and V. Veilleux-Gaboury, "Motion detection with an unstable camera," in *Proceedings of International Conference on Image Processing (ICIP-08)*, pp. 229–232, San Diego (USA), October 2008.
- [3] R. Lukac, *Single-Sensor Imaging: Methods and Applications for Digital Cameras*. CRC Press, 2008.
- [4] S. Battiato and R. Lukac, *Encyclopedia of Multimedia*, ch. Video Stabilization Techniques, pp. 941–945. Springer-Verlag, October 2008.
- [5] K.-Y. Lee, Y.-Y. Chuang, B.-Y. Chen, and M. Ouhyoung, "Video stabilization using robust feature trajectories," in *Proceedings of IEEE International Conference on Computer Vision (ICCV 2009)*, pp. 1397–1404, 2009.
- [6] M. Irani and P. Anandan, "About direct methods," in *Proceedings of International Workshop on Vision Algorithms, held during ICCV*, pp. 267–277, Corfu (Greece), 1999.
- [7] P. H. S. Torr and A. Zisserman, "Feature based methods for structure and motion estimation," in *Proceedings of International Workshop on Vision Algorithms, held during ICCV*, pp. 278–294, Corfu (Greece), 1999.
- [8] S. Battiato, G. Puglisi, and A. R. Bruna, "Regular texture removal for video stabilization," in *Proceedings of International Conference on Pattern Recognition (ICPR-08)*, Florida (USA), 2008.
- [9] F. Vella, A. Castorina, M. Mancuso, and G. Messina, "Digital image stabilization by adaptive block motion vectors filtering," *IEEE Transactions on Consumer Electronics*, vol. 48, pp. 796–801, 2002.
- [10] S. W. Jang, M. Pomplun, G. Kim, and H. I. Choi, "Adaptive robust estimation of affine parameters from block motion vectors," *Image and Vision Computing*, pp. 1250–1263, August 2005.
- [11] S. Soldatov, K. Strelnikov, and D. Vatolin, "Low complexity global motion estimation from block motion vectors," in *Proceedings of GraphiCom*, Novosibirsk Akademgorodok (Russia), July 2006.
- [12] S. Auberger and C. Miro, "Digital video stabilization architecture for low cost devices," in *Proceedings of International Symposium on Image and Signal Processing and Analysis*, pp. 474–479, Zagreb (Croatia), September 2005.

- [13] J. Lee, Y. Park, S. Lee, and J. Paik, "Statistical region selection for robust image stabilization using feature-histogram," in *Proceedings of International Conference on Image Processing (ICIP-09)*, Cairo (Egypt), 2009.
- [14] S. Battiato, A. R. Bruna, and G. Puglisi, "A robust video stabilization system by adaptive motion vectors filtering," in *Proceedings of International Conference on Multimedia and Expo (ICME-08)*, pp. 373–376, Hannover (Germany), June 2008.
- [15] S. Battiato, A. R. Bruna, and G. Puglisi, "A robust block based image/video registration approach for mobile imaging devices," *IEEE Transactions on Multimedia*, 2010. To appear.
- [16] A. R. Bruna, S. Battiato, and G. Puglisi, "Method of filtering a video sequence image from spurious motion effects." US Patent 12272591, filed on November 2008.
- [17] W. ho Cho and K.-S. Hong, "Affine motion based CMOS distortion analysis and CMOS digital image stabilization," *IEEE Transactions on Consumer Electronics*, vol. 53, pp. 833–841, August 2007.
- [18] A. U. Batur and B. E. Flinchbaugh, "Video stabilization with optimized motion estimation resolution," in *Proceedings of International Conference on Image Processing (ICIP-06)*, pp. 465–468, Atlanta (USA), 2006.
- [19] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in *Proceedings of National Telecommunications Conference*, vol. 4, pp. G5.3.1–G5.3.5, 1981.
- [20] F. Rovati, D. Pau, E. Piccinelli, L. Pezzoni, and J.-M. Bard, "An innovative, high quality and search window independent motion estimation algorithm and architecture for mpeg-2 encoding," *IEEE Transactions on Consumer Electronics*, vol. 46, pp. 697–705, August 2000.
- [21] A. Censi, A. Fusiello, and V. Roberto, "Image stabilization by features tracking," in *Proceeding of International Conference on Image Analysis and Processing (ICIAP-99)*, pp. 665–667, Venice (Italy), September 1999.
- [22] C. G. Harris and M. Stephens, "A combined corner and edge detector," in *Proceeding of Alvey Vision Conference*, pp. 147–151, Manchester (UK), 1988.
- [23] F. Hampel, P. Rousseeuw, E. Ronchetti, and W. Stahel, *Robust Statistics: the Approach Based on Influence Functions*. Wiley Series in probability and mathematical statistics, John Wiley & Sons, 1986.
- [24] J. Yang, D. Schonfeld, C. Chen, and M. Mohamed, "Online video stabilization based on particle filters," in *Proceedings of International Conference on Image Processing (ICIP-06)*, pp. 1545–1548, Atlanta (USA), October 2006.

- [25] H. Jiménez and J. Salas, “Dynamic selection of characteristics for feature based image sequence stabilization,” in *Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS-08)*, vol. 5259 of *Lecture Notes in Computer Science*, pp. 290–297, Juan-les-Pins (France), Springer, 2008.
- [26] C. R. del Blanco, F. Jaureguizar, L. Salgado, and N. García, “Automatic feature-based stabilization of video with intentional motion through a particle filter,” in *Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS-08)*, vol. 5259 of *Lecture Notes in Computer Science*, pp. 356–367, Juan-les-Pins (France), Springer, 2008.
- [27] C. R. del-Blanco, F. Jaureguizar, L. Salgado, and N. García, “Motion estimation through efficient matching of a reduced number of reliable singular points,” in *Society of Photo-Optical Instrumentation Engineers (SPIE-08) Conference Series*, vol. 6811, 2008.
- [28] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [29] P. Meer, C. V. Stewart, and D. E. Tyler, “Robust computer vision: an interdisciplinary challenge,” *Computer Vision and Image Understanding*, vol. 78, pp. 1–7, 2000.
- [30] A. Adams, N. Gelfand, and K. Pulli, “Viewfinder alignment,” *Computer Graphics Forum*, vol. 2, no. 27, pp. 597–606, 2008.
- [31] L. Litwin, “FIR and IIR digital filters,” *IEEE Potentials*, vol. 19, pp. 28–31, October–November 2000.
- [32] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Transaction of the ASME Journal of Basic Engineering*, vol. 82, pp. 33–45, 1960.
- [33] G. Welch and G. Bishop, “An introduction to the kalman filter.” tutorial on the Kalman filter presented at ACM SIGGRAPH, 2001.
- [34] S. Erturk, “Image sequence stabilization based on kalman filtering of frame positions,” *Electronics Letters*, vol. 37, no. 20, pp. 1217–1219, 2001.
- [35] M. Tiko and M. Vehvilainen, “Constraint motion filtering for video stabilization,” in *IEEE International Conference on Image Processing (ICIP-05)*, pp. 569–572, Genoa (Italy), September 2005.
- [36] G. Wolberg, *Digital Image Warping*. Wiley-IEEE Computer Society Press, August 1990.

Image Categorization

G. M. Farinella

Image Processing Lab, University of Catania, Italy.

D. Ravi

Advanced System Technology - Catania Lab, STMicroelectronics, Italy.

Abstract: Vision is perhaps the most important sense for humans. Among the different complex tasks accomplished by the Human Visual System, the categorization is a fundamental process that allows humans to effectively interpret their surroundings efficiently and rapidly. Computer Vision researchers are increasingly using algorithms from Machine Learning to build robust and reusable machine vision systems that act taking into account the visual content of images. Since learning is a key component of biological vision systems, the design of artificial vision systems that learn and adapt represent one of the most important trend in modern Computer Vision research. Despite the advances in the context of single sensor imaging devices, this technology is still quite far from the ability of automatically categorize and exploit the visual content of the scene during (or after) acquisition time. Different constraints should be considered in order to transfer the ability of inferring the category of a scene in imaging devices domain. Indeed, these devices have limited resources in terms of memory and computational power, and the image data format change over time through the imaging pipeline (i.e., from Bayer Pattern at acquisition time to JPEG format after acquisition time). This Chapter presents Computer Vision and Machine Learning techniques within the application contexts of scene recognition and red-eye detection. The techniques introduced here could be used in building complex imaging pipeline in which image categorization (e.g., scene recognition, red-eye detection) is exploited to drive other tasks (e.g., white balance, red eye removal).

10.1 Introduction

Vision consists of processing images of scenes so as to make explicit what needs to be known about them [1]. Visual categorization is a fundamental cognitive process that refers the ability to group visual stimuli into meaningful categories. This aptitude allows humans to efficiently and rapidly analyze their surroundings. Humans Vision System (HVS) is able to categorize complex visual scenes at a single glance, despite the number of objects with different poses, colors, shadows and textures that may be contained in the scenes. The understanding of the robustness and rapidness of this human ability has been a focus of investigation for the cognitive sciences community over many years [2]. Seminal studies in computational vision have portrayed scene recognition as a progressive reconstruction of the input from local measurements (e.g., edges, surfaces) [1]. In contrast, some experimental studies have suggested that categorization of real world scenes may be initiated from the encoding of the global configuration, bypassing most of the information about local concepts and objects [3]. The ability to exploit the global configuration of a seen scene is achieved by humans through the exploration of holistic cues (e.g., texture) [4]. Those cues can be processed as single entity over the entire human

visual field without requiring attention to local features [4, 5]. Recent studies suggested that humans rely on local information as much as on global information to perform scene recognition [6, 7]. Clearly, learning is a key component of HVS because it allows the adaptation of the cortex to learn new tasks [8].

The Human Visual System and related studies of Cognitive Sciences community have stimulated researches in Computer Vision in building artificial image categorization systems. Motivations beyond that of pure scientific curiosity are provided by several important applications: content-based image retrieval (CBIR) [9], object detection and recognition [10], semantic organization of image databases [11], place recognition for robot navigation systems [12], direct marketing on multimedia messaging services domain (MMS) [13, 14]. In the context of the current book, a software engine able to automatically infer the category of a scene could be helpful to drive different tasks performed by single sensor imaging devices during acquisition time (e.g., autofocus, autoexposure, white balance) [15, 16] or during post acquisition time (e.g., image enhancement, image coding) [17, 18].

The term categorization (or detection) is used in Computer Vision to designate the task of between-class classification (e.g., *red-eye* vs. *non-eye*) [8]. In other words, categorizing an image consists of determining those categories (e.g., *forest*, *office*, *inside city*, etc.) to which an image belongs. Note that an image can belong to multiple categories, and that image categories can be very broad (e.g., *bedroom*, *highway*) or narrow (e.g., *inside city*, *street*) in terms of visual content.

This Chapter introduces basic techniques of Computer Vision and Machine Learning applied in two contexts of interest for single sensor imaging domain: scene recognition and red-eyes detection. Specifically, next Section introduces fundamental concepts of Machine Learning used in Computer Vision for image categorization. Section 10.3 presents a scene categorization engine useful to discriminate between *Natural* and *Artificial* scenes¹ as well as between multiple classes of scenes usually acquired by a consumer imaging device (e.g., *Portrait*, *Landscape* and *Document*) [19]. To this aim a holistic representation of the scene in the Discrete Cosine Transform domain (fully compatible with the JPEG format) is exploited. The advised representation is coupled with logistic classifiers to perform categorization. In Section 10.4 the problem of discriminating between *red-eyes* vs. *non-eyes* patches is taken into account. The detection of red-eyes is a fundamental task for automatic red-eyes removal frameworks (see Chapter 8 for more details). The technique described in Section 10.4 makes use of a two step approach to identify the red-eyes. First, red-eyes candidates are extracted from the input image by using an image filtering pipeline. A set of linear discriminant classifiers is then learned on the clustered patches space, and hence employed to distinguish between *red-eyes* and *non-eyes* patches.

¹In this Chapter the term *Artificial* refers to images in which are depicted man-made environments (*cities*, *buildings*, *streets*, etc.) whereas *Natural* refers to images in which natural landscapes are represented (*open country*, *mountain*, *forest*, *coast*, etc.).

10.2 Fundamentals and Background

In modern Computer Vision, the categorization problem is typically considered as a supervised learning problem [8]. Given an image as input, the desired output of the machine vision system is a discrete label indicating the class of the image (e.g., is it a red-eye?). The systems are trained with a set of image examples previously labeled (called training set). To this aim, each image I in the training database is represented as a feature vector $\mathbf{f} = F(I)$ through a feature extraction algorithm F (e.g., images are represented as histograms of oriented gradients). Once the learning phase is completed, each new test image to be classified is first represented as feature vector and then categorized by using the previous learned classifiers². Dimensional reduction procedures (e.g., Principal Component Analysis) are typically adopted to obtain the final feature space in which learning and categorization phases are performed.

Image categorization engines must be able to generalize on different viewing condition (e.g., red-eye of different face poses, different light conditions, etc.) as well as on different examples of the class (e.g., Asian/European shape of eyes). The difficulty to achieve good generalization depends on the size and composition of the training set as well as on the variability covered by training samples. For instance, a system trained on frontal red-eyes would be unable to detect a red-eye from any viewpoint. On the other hand, if the same machine is trained with a large set of examples covering the relevant variability, it may be capable of accomplish the task.

Most of the systems for image categorization use set of binary classifiers, one for each image category C_i . In each of the two-class categorization problem, given an image I , the problem is to understand if it contains a particular visual class C_i or not, taking into account the representation of I into the feature space ($\mathbf{f} = F(I)$). From a statistical point of view, the task of categorization becomes a comparison between two probability scores:

- $P(C_i|\mathbf{f})$: the probability of having C_i given the feature vector $\mathbf{f} = F(I)$;
- $P(Other|\mathbf{f})$: the probability of not having C_i given the feature vector $\mathbf{f} = F(I)$.

By using the Bayes' theorem [20], the ratio between the above probabilities can be written as follows:

$$\frac{P(C_i|\mathbf{f})}{P(Other|\mathbf{f})} = \frac{P(\mathbf{f}|C_i)}{P(\mathbf{f}|Other)} \frac{P(C_i)}{P(Other)} \quad (10.1)$$

The term on the left-hand side of the above equation is called *Posterior Ratio*, the first term of right-hand side is called *Likelihood Ratio*, whereas the second term of right-hand side is called *Prior Ratio*. The *Likelihood* finalizes the probability of observing the particular instance $\mathbf{f} = F(I)$ given some models of the image class under consideration. There are two different class of techniques used to deal with (10.1) [21]:

²The learning procedure is typically made out of the devices for those devices with limited resources in terms of memory and computational power.

- *Discriminative Methods*: they estimate the posterior probability ratio directly by finding boundaries of the classes in the feature space. Methods are called “discriminative” when the *Posterior Ratio* $\frac{P(C_i|\mathbf{f})}{P(\text{Other}|\mathbf{f})}$ can be viewed as a function discriminating directly the target classes for any given instance of images.
- *Generative Methods*: such techniques learn a model of the joint probability distribution $P(\mathbf{f}, C_i) = P(\mathbf{f}|C_i)P(C_i)$ and make prediction by using Bayes rule to calculate the posterior probability $P(C_i|\mathbf{f})$. The training data are used to learn a model for the likelihood $P(\mathbf{f}|C_i)$ and the prior distribution $P(C_i)$. These methods are called “generative” because the probability distribution $P(\mathbf{f}|C_i)$ can be viewed as describing how to generate random instances of an image conditioned on the target class.

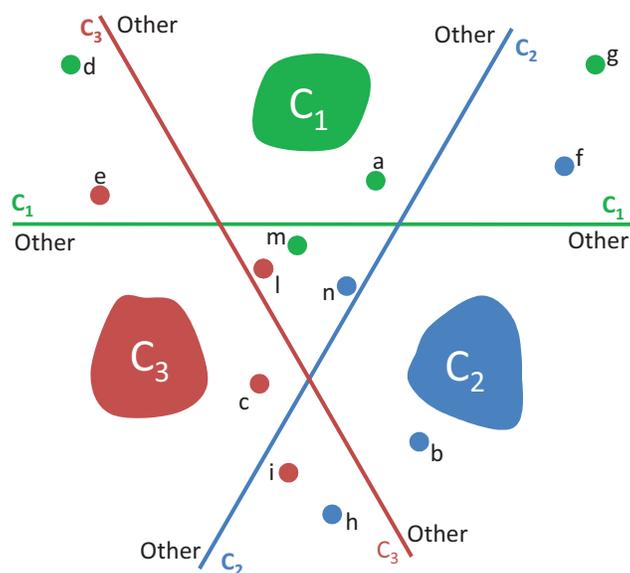
Different researchers faced the difficult task to compare generative and discriminative learning [22–24]. Discriminative classifiers (i.e., Logistic Classification and Linear Discriminant Analysis [25]) are used in further Sections of this Chapter just for simplicity of explanation. Similar considerations and results to those presented here can be argued by using generative classifiers rather than discriminative ones.

To achieve multiclass classification with a set of binary discriminative classifiers, the adoption of some combination strategies is required [25]:

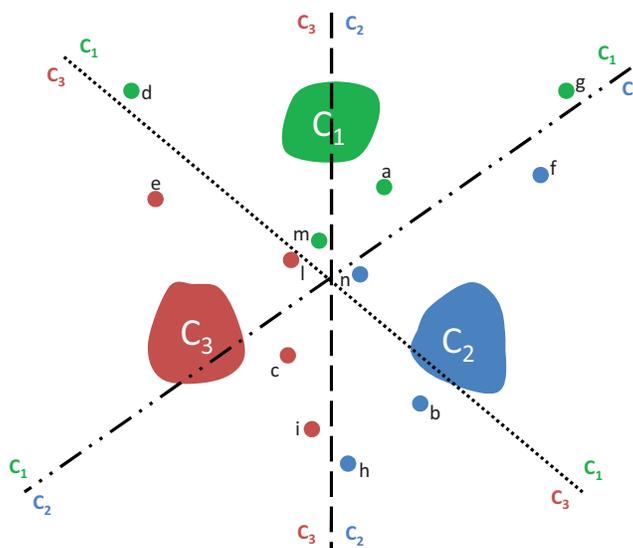
- The one-against-all approach (Fig.(10.1(a))), builds N binary classifiers (one for each class). The i^{th} classifier is trained to discriminate samples in class C_i (the positive class) from those in the remaining classes (the negative class). After the training phase of all N binary classifiers the corresponding N binary discrimination functions are evaluated to establish the class of a new sample. In the ideal case, for a given sample, only one classifier will be positive giving a clear indication of the class (e.g., points a, b, c in Fig.(10.1(a))). If there is more than one classifier which is positive, the new sample is assigned to the class for which the distance to the hyperplane defined by the discriminative function is the largest (e.g., points d, e, f, g, h, i in Fig.(10.1(a))). If all classifiers are negative, then the new sample is assigned to the class with smallest distance to the hyperplane defined by the discriminative function (e.g., points l, m, n in Fig.(10.1(a))).
- The one-against-one approach (Fig.(10.1(b))) builds $N(N - 1)/2$ classifiers. Each classifier discriminates between two classes. A sample is assigned using a voting procedure (e.g., in Fig.(10.1(b)) the points a and g have two votes for class C_1 and one vote for class C_2 , so these points are assigned to the class C_1). Alternatively, the pairwise classifiers can be arranged in trees, where each node represents a classifier and the leafs are labeled with the names of the involved classes [26].

10.3 Scene Recognition

Recent studies demonstrate that Image Generation Pipeline (IGP) (See Chapter 1) of single sensor imaging devices can be effectively improved by considering scene recognition



(a) One-against-all



(b) One-against-one

Figure 10.1 : Example of one-against-all and one-against-one approaches. Colored regions indicate training samples of three different classes. Points indicate new samples classified taking into account the learned decision functions.

engines. In [15] authors describe a powerful imaging pipeline for color constancy in which a scene recognition engine (i.e., *Natural* vs. *Artificial*) is used to drive the en-

hancement task. The results achieved in [15] make stronger the idea that the content of a scene (e.g., its category) is useful for IGPs' tasks performed during acquisition time (e.g., autoexposure - Chapter 3, autofocus - Chapter 4, white balance - Chapter 5) or just after acquisition (e.g., image coding - Chapter 11).

Many Computer Vision researchers have proved that holistic approaches can be efficiently used to solve the problem of rapid and automatic scene classification [5, 9, 27–33]. In particular holistic approaches are able to recognize a scene bypassing the recognition of the objects inside the scene. Most of the holistic approaches in literature share the same basic structure that can be schematically summarized as follows:

1. A suitable features space is built (e.g., visual vocabulary [30]). This space must emphasize specific image cues such as, for example, corners, oriented edges, textures, etc.
2. Each image under consideration is projected into this space. A descriptor is built considering the image as a whole entity (e.g., visual words histogram [30]).
3. Scene classification is obtained by using Pattern Recognition and Machine Learning algorithms on the new holistic representation of the images (e.g., by using K-nearest neighbors [30]).

A wide class of techniques based on the above scheme work extracting features on perceptually uniform color spaces (e.g., CIE Lab). Typically, filter banks [5, 30, 32] or local invariant descriptors [28, 29] are employed to capture image cues and to build the visual vocabulary to be used in a bag of visual words model [34]. An image is considered as a distribution of visual words and this holistic representation is used for classification purpose. Eventually local spatial constraints are added in order to capture the spatial layout of the visual words within images [29, 30, 32].

Alternatively the frequency domain can be a useful and effective source of information to encode holistically an image for scene recognition. The statistics of natural images on frequency domain reveal that there are different spectral signatures for different image categories [10]. Although these studies have demonstrated that frequency domain can be useful to discriminate between different classes scenes, just a few researchers have exploited such domain for scene recognition tasks [35–37]. A recent review in the field can be found in [33].

This Section describes a scene recognition framework in which the holistic representation of the scene is built exploiting features extracted on Discrete Cosine Transform domain. An early version of this work has been published in [19]. Logistic classifiers [25] are trained and then used to infer the category of a new observed scene. A one-against-all method is used to perform multiclass classification (see Section 10.2). The proposed approach is fully compatible with JPEG format and may be easily employed in single sensor imaging devices domain.

Images are represented as histograms of oriented blocks [11] coupled with statistical weights to evaluate how important is an orientation in discriminating the classes under

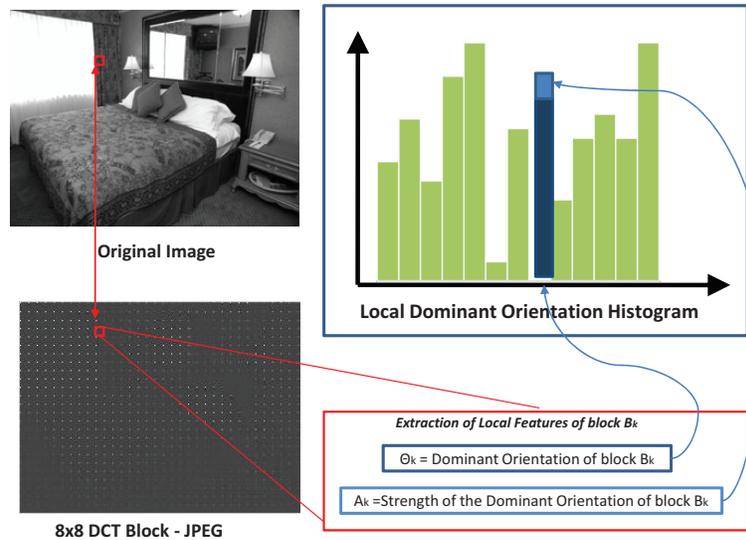


Figure 10.2 : Images are represented as histograms of local dominant orientations weighted by their strengths.

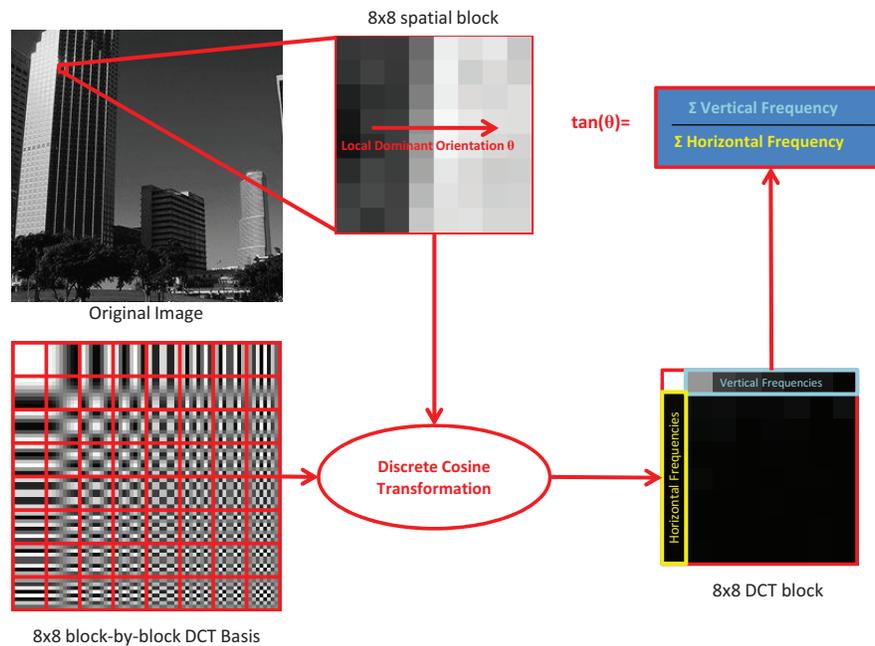


Figure 10.3 : Local dominant orientation extraction process.

consideration [38]. Two local features are extracted and used to represent each 8×8 spatial block belonging to a given image (Fig.(10.2)): the dominant orientation of the block and its strength. These two local information are extracted, for each block, directly on compressed domain considering the corresponding 8×8 DCT blocks [39]. Specifically, the ratio between the sum of the DCT coefficients corresponding to the horizontal and

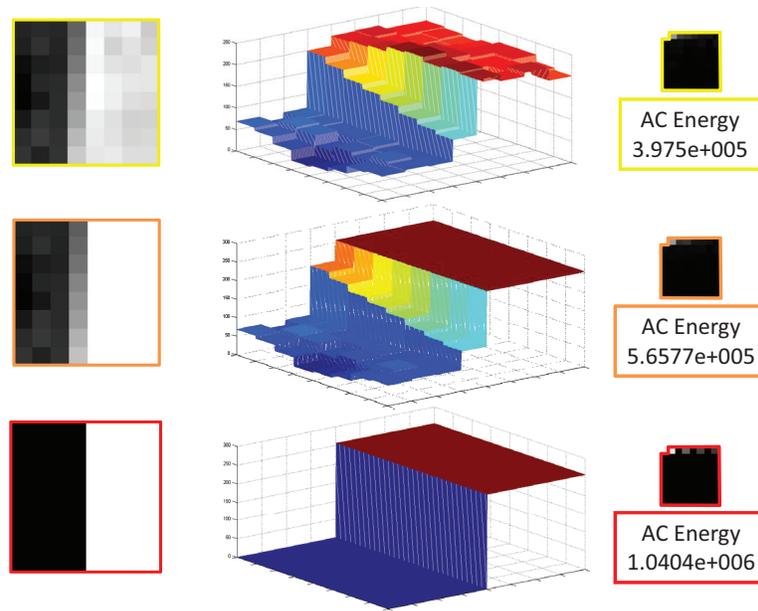


Figure 10.4 : The overall AC energy contained on a 8×8 DCT block is related to the strength of the local dominant orientation of the corresponding spatial block. Three 8×8 spatial blocks with same local dominant orientation and different strength are represented in the left column. The strengths of the blocks (in increasing order from top to bottom) are represented in the middle. The AC energy of each block extracted from DCT domain is reported in the right column. The AC energy increases from top to bottom in accordance with the strengths of the corresponding blocks.

vertical frequencies of the 8×8 DCT basis functions, are used to establish the tangent of the local dominant orientation angle of a spatial block (Fig.(10.3)). The overall AC energy contained in the 8×8 DCT block is related to the strength of the local dominant orientation (LDO) of the spatial block (Fig.(10.4)). The extracted local features are then used to build a holistic representation of the image as a distribution of local dominant orientations. This representation is coupled with *TF-IDF* weighting scheme [38] to statistically capture the most discriminative orientations between classes of scenes.

In the context of single sensor imaging devices, the primary contributions of the aforementioned approach can be summarized as follows:

- It is directly implemented in the DCT domain and compliant with the JPEG format. The full image decoding is not required to extract the features used to represent the scene. Local features are picked-out by using simple operations in compressed DCT domain. Bank of filters or other feature extraction process are not required.
- The global representation of the scene is obtained grouping together the extracted local information. A very compact low dimensional vector is used to describe the content of a scene.

- No extra information (e.g., visual vocabulary) has to be stored in memory to build and manage the holistic representation of a scene.
- A simple discriminative model is used to perform classification. A very compact low dimensional vector of parameters is maintained to perform classification. The classifier can be trained out of the device (e.g., just considering a dataset properly collected). The learned parameters of the classification model may be stored into the device and used to classify a new observed scene through a simple decision function.
- Despite the proposed approach works on compressed and constrained domain the classification rate closely match current state-of-the-art methods.

10.3.1 Histograms of Oriented DCT Blocks

This Section introduces the holistic representation used in the aforementioned scene recognition framework. For sake of simplicity of explanation, the task of *Natural* vs. *Artificial* classification is taken into account to introduce the holistic representation of the scene. The same representation is used to discriminate between the following classes of scenes:

- *Natural, Artificial*;
- *Open, Closed*;
- *Indoor, Outdoor*;
- *Document, Landscape, Portraits*.

As well known, the power spectrum of an image contains enough relevant information about its global structure [11, 37, 40]. Different studies pointed out that information extracted in frequency domain can be holistically encoded to represent the scene for *naturalness* classification. In this specific case, the discrimination between *Natural* vs *Artificial* scenes is based on the fact that straight horizontal and vertical lines dominate man-made structures whereas most natural landscapes have textured zones and undulating contours. Therefore, scenes having a distribution of edges (in spatial domain) commonly found in natural landscapes would have a high degree of *naturalness* whereas scenes with distribution of edges (in spatial domain) biased toward vertical and horizontal orientations would have a low degree of *naturalness* [41]. These considerations lead to claim that the distribution of edges' orientations within an image can help to understand the *naturalness* of the scene and hence may be used for *Natural* vs. *Artificial* classification.

To discriminate between *Natural* and *Artificial* scenes a global representation of the image is built after estimating the local dominant orientation and strength of each 8×8 block belonging to the grayscale image. Specifically, this local information is obtained considering each block within an image encoded in the DCT domain.

Shen and Sethi [39] proposed the first image processing approach to extract edge directly on DCT domain. The method can be used to extract edge information during

the JPEG encoding of the image (or after) directly in the encoded compressed image domain. Specifically, during JPEG encoding, the image is partitioned in 8×8 pixel-by-pixel non overlapped blocks transformed by using DCT basis. Each block is then transformed taking into account the 8×8 DCT basis to produce the 8×8 coefficient array where the (0,0) element (top-left) is the DC (zero-frequency) component and entries with increasing vertical and horizontal index values represent higher vertical and horizontal spatial frequencies. For each block $B_k(x,y)$ in the original image, the corresponding DCT coefficients $D_k(u,v)$ are generated by using the following equation:

$$D_k(u,v) = \alpha(u)\alpha(v) \sum_{x=0}^7 \sum_{y=0}^7 B_k(x,y) \cos\left(\frac{\pi(2x+1)u}{16}\right) \cos\left(\frac{\pi(2y+1)v}{16}\right) \quad (10.2)$$

where

$$\alpha(f) = \begin{cases} \frac{1}{\sqrt{8}} & f = 0, \\ \sqrt{\frac{1}{4}} & 1 \leq f \leq 7. \end{cases} \quad (10.3)$$

From this representation the following equation can be used to obtain the edge orientation of $B_k(x,y)$ [39]:

$$\tan(\theta_k) = \frac{\sum_{u=1}^7 D_k(u,0)}{\sum_{v=1}^7 D_k(0,v)} \quad (10.4)$$

Further investigation of Ladret *et al.* [11] demonstrated that a better estimate of the orientation in terms of stability (for both positive and negative orientations), lower offset effects and accuracy, can be obtained by using the following equation:

$$\tan(\theta_k) = \frac{\sum_{u=1;u=u+2}^7 D_k(u,0)}{\sum_{v=1;v=v+2}^7 D_k(0,v)} \quad (10.5)$$

It is important to note that in (10.4) and (10.5) the local dominant orientation of each 8×8 image block is computed directly on compressed domain.

The local variance of each 8×8 DCT block (the AC energy) is a good indicator of the strength of the edge whose orientation has been evaluated by using (10.5) (Fig.(10.4)). The strength can properly weight each edge according to its importance. To evaluate the strength of each block the overall AC energy is computed by using the following formula [39]:

$$A_k = H_k + V_k + \sum_{u=1}^7 \sum_{v=1}^7 D_k^2(u,v) \quad (10.6)$$

$$H_k = \sum_{u=1}^7 D_k^2(u,0) \quad (10.7)$$

$$V_k = \sum_{v=1}^7 D_k^2(0,v) \quad (10.8)$$

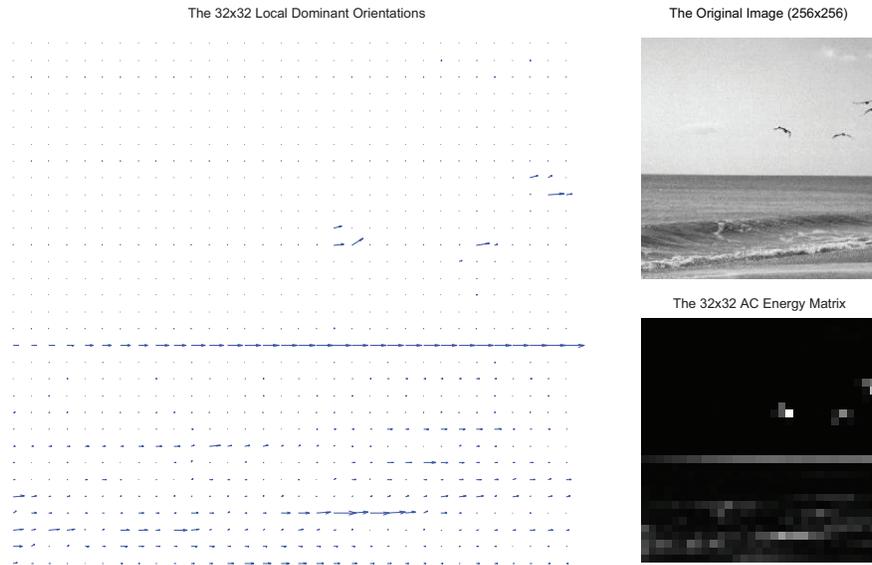
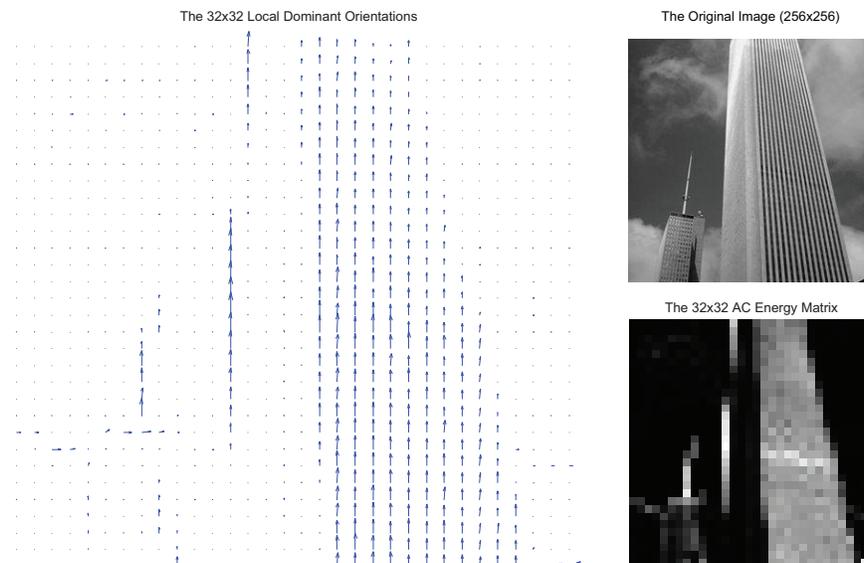
(a) A *Natural* scene.(b) An *Artificial* scene.

Figure 10.5 : Local dominant orientations and the corresponding AC energies. The local dominant orientation of each 8×8 block is shown on the left of the *Natural/Artificial* image, whereas the corresponding AC energies are shown bottom-right.

Fig.(10.5(a)) illustrates a *Natural* scene, the local dominant orientation of each block estimated by using (10.5) and the relative strength (AC Energy) estimated with (10.6). Fig.(10.5(b)) illustrates an *Artificial* scene, the local dominant orientation of each block estimated by using (10.5) and the relative strength estimated with (10.6). The arrows

lengths on the local dominant orientations plot are proportional to the norm of the vector with components respectively equal to the total horizontal energy (10.7) and the total vertical energy (10.8) of each 8×8 DCT block.

As shown in Fig.(10.5(a)) and Fig.(10.5(b)), the *Natural* scenes present many horizontal edges (e.g., due the horizon in the scene) whereas in the *Artificial* scenes the vertical edges are prominent (e.g., due the buildings in the scene). The strength (AC Energy) of each block indicates how much the corresponding local dominant orientation should be taken into account. The strength (AC Energy) corresponding to homogeneous or textured image blocks (e.g., sky blocks, sea blocks, clouds blocks, etc.) is lower than the corresponding strength (AC Energy) of edge image blocks (e.g., horizon blocks).

A holistic representation of the scene can be built just analyzing the distribution of the local dominant orientations weighted taking into account their corresponding strengths [11]. Specifically, for each grayscale image I coded with K blocks in $DCT_{8 \times 8}$ domain, let $\{\theta_1, \dots, \theta_K\}$ the K local dominant orientations extracted by using (10.5) and let $\{A_1, \dots, A_K\}$ the K AC energies extracted by using (10.6).

The d -dimensional features vector $LDO(DCT_{8 \times 8}(I)) = [f_{\hat{\theta}_1}, f_{\hat{\theta}_2}, \dots, f_{\hat{\theta}_d}]^T$ used to represent the whole image I is obtained as follows:

$$f_{\hat{\theta}_i} = \frac{N(\hat{\theta}_i)}{SN}, \quad \forall i \in \{1, \dots, d\} \quad (10.9)$$

where:

$$N(\hat{\theta}_i) = \sum_{A_k \in \Theta_i} \log(A_k);$$

$$\hat{\theta}_i \in [-90, 90], \quad \hat{\theta}_1 = -90, \quad \hat{\theta}_{i+1} = \hat{\theta}_i + \frac{180}{d}, \quad \hat{\theta}_{d+1} = 90;$$

$$\Theta_i = \{A_k \mid \hat{\theta}_i < \theta_k \leq \hat{\theta}_{i+1}, A_k > \zeta, k = 1 \dots K\};$$

$$SN = \sum_{n=1}^d N(\hat{\theta}_n) \text{ is the normalization constant;}$$

d is the number of orientation bins;

ζ is a threshold useful to discard the marginal orientations.

The *LDO* representation has been coupled with the *TF-IDF* weighting scheme [38] to enhance the most discriminative orientations between different classes. The final *TF-IDF-LDO* descriptor has been used as feature vector for image categorization.

To effectively build the *TF-IDF-LDO* of an image I , the number of bins to be considered (the parameter d) and the threshold used to consider only the significant orientations (the parameter ζ) must be fixed. To achieve this aim the benchmark algorithm K-nearest neighbors (KNN) and the leave-one-out cross validation procedure [25] are employed. When the best representation parameters have been fixed, these are used in training the logistic classification model [25] employed for the final classification. The experiments pointed out that the best performances are obtained with $d = 32$ and ζ equal to 10% of

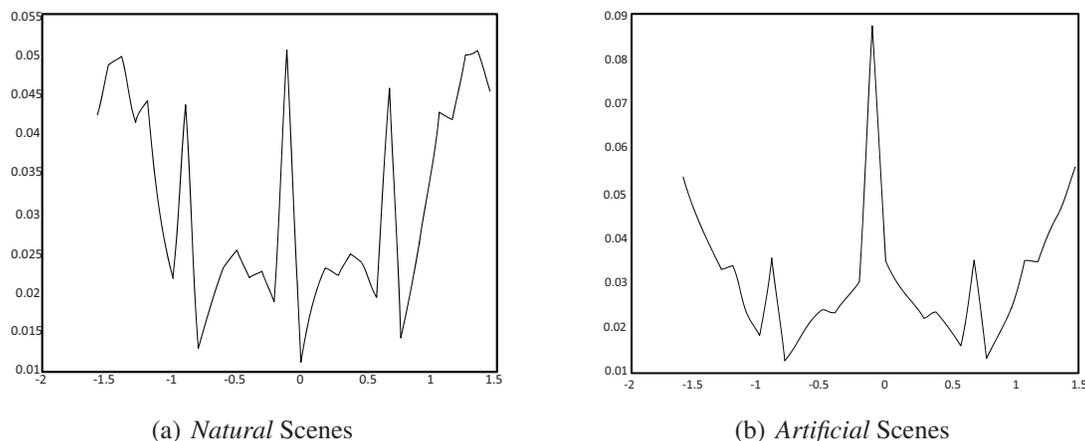


Figure 10.6 : The mean local dominant orientations distributions of *Natural* and *Artificial* Scenes.

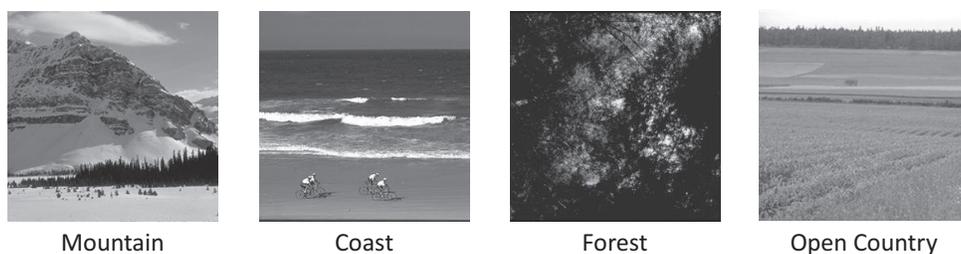


Figure 10.7 : Some examples of *Natural* scenes used to compute the average local dominant orientations distributions reported in Fig.(10.6(a)).

the maximal A_k extracted from the image I under consideration during the representation phase.

Fig.(10.6) reports the mean local dominant orientations distributions of *Natural* and *Artificial* scenes. The distributions in Fig.(10.6(a)) have been computed averaging the *TF-IDF-LDO* representations of about 1400 *Natural* scenes (Fig.(10.7)): 410 *Open Country*, 328 *Forest*, 360 *Coast*, 374 *Mountain*. The distributions in Fig.(10.6(b)) have been computed averaging the *TF-IDF-LDO* representations of about 3200 *Artificial* scenes (Fig.(10.8)): 216 *Bedroom*, 241 *Suburban*, 311 *Industrial*, 210 *Kitchen*, 289 *Living Room*, 260 *Highway*, 308 *Inside City*, 292 *Street*, 356 *Tall Building*, 215 *Office*, 315 *Store*. All images are in gray scale and encoded in JPEG format with different resolution and compression ratio. This dataset has been used to perform the *Natural* vs. *Artificial* classification experiments.

The *Natural* vs. *Artificial* classification engine is based on the differences concerning the “shape” of the *TF-IDF-LDO* distributions (Fig.(10.6)). This representation has been demonstrated also useful to discriminate between other classes of scenes.

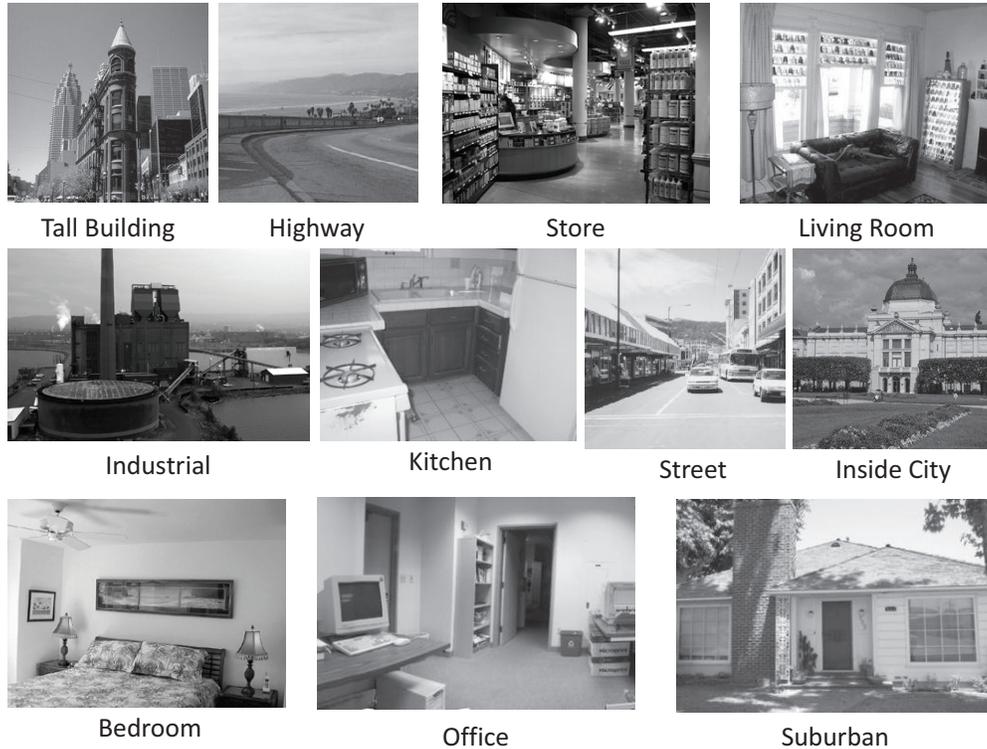


Figure 10.8 : Some examples of *Artificial* scenes used to compute the average local dominant orientations distributions reported in Fig.(10.6(b)).

10.3.2 Experimental Settings and Results

To effectively build the *TF-IDF-LDO* of an image I , the number of bins to be considered (the parameter d) and the threshold used to take into account only the significant orientations (the parameter ζ) must be fixed. In our experiments, we used a K-nearest neighbors [25] to fix the best representation parameters, whereas a logistic classification model [25] is employed as final framework for classification.

All experiments that involve K-nearest neighbors for class discrimination have been performed by using of the leave-one-out cross validation procedure (LOOCV) [25]. Each run of LOOCV involved a single image sample as test to be classified, and the remaining image samples as the training data. The final classification results are obtained averaging on the results of all the LOOCV runs. The following parameters have been involved into the experiments: the number d of orientation bins, the strength threshold ζ , the similarity measure S used by KNN, and the number of neighbors K used in the nearest neighbor rule. Each experiment was related to the evaluation of the classification performance by using a point into the parameter space $d \times \zeta \times S \times K$ (properly sampled in a grid of 960 points). The similarity measure S used in the K-nearest neighbors classifier has a clear impact in the classification results. In the experiments performed to fix representation parameters, different similarity measures have been tested. Let $L^{(1)} = TF-IDF-LDO(DCT_{8 \times 8}(I^{(1)})) =$

$[f_{\hat{\theta}_1}^{(1)}, f_{\hat{\theta}_2}^{(1)}, \dots, f_{\hat{\theta}_d}^{(1)}]^T$ and $L^{(2)} = TF-IDF-LDO(DCT_{8 \times 8}(I^{(2)})) = [f_{\hat{\theta}_1}^{(2)}, f_{\hat{\theta}_2}^{(2)}, \dots, f_{\hat{\theta}_d}^{(2)}]^T$ the local dominant orientations distributions of two different images $I^{(1)}$ and $I^{(2)}$. The following similarity measures have been considered: Absolute Difference (10.10), metric based on Bhattacharyya Coefficient (10.11), χ^2 Distance (10.12), Jeffrey Divergence (10.13), Pearson Correlation Coefficient (10.14), Square Difference Distance (10.15), Weighted Euclidean Distance (10.16). Among the similarity measures used in these experiments, the Euclidean distance on the real Fourier coefficient was considered as previously suggested in [11]. In this last case, the local dominant distribution is treated as a discrete signal and the Fourier transformation is employed. The Euclidean distance between the first two Fourier components is used to establish the similarity between scenes. More formally in the following list all similarity measures are pointed out:

$$S(L^{(1)}, L^{(2)}) = \sum_{n=1}^d |f_{\hat{\theta}_n}^{(1)} - f_{\hat{\theta}_n}^{(2)}| \quad (10.10)$$

$$S(L^{(1)}, L^{(2)}) = \sqrt{1 - \sum_{n=1}^d \sqrt{f_{\hat{\theta}_n}^{(1)} * f_{\hat{\theta}_n}^{(2)}}} \quad (10.11)$$

$$S(L^{(1)}, L^{(2)}) = \sum_{n=1}^d \frac{(f_{\hat{\theta}_n}^{(1)} - m_i)^2}{m_i} \quad (10.12)$$

where $m_i = \frac{f_{\hat{\theta}_n}^{(1)} + f_{\hat{\theta}_n}^{(2)}}{2}$;

$$S(L^{(1)}, L^{(2)}) = \sum_{n=1}^d \left\{ \left[f_{\hat{\theta}_n}^{(1)} * \log \left(\frac{f_{\hat{\theta}_n}^{(1)}}{m_i} \right) \right] + \left[f_{\hat{\theta}_n}^{(2)} * \log \left(\frac{f_{\hat{\theta}_n}^{(2)}}{m_i} \right) \right] \right\} \quad (10.13)$$

where $m_i = \frac{f_{\hat{\theta}_n}^{(1)} + f_{\hat{\theta}_n}^{(2)}}{2}$;

$$S(L^{(1)}, L^{(2)}) = 1 - \frac{\sum_{n=1}^d \left(\frac{f_{\hat{\theta}_n}^{(1)} - \mu^{(1)}}{\sigma^{(1)}} * \frac{f_{\hat{\theta}_n}^{(2)} - \mu^{(2)}}{\sigma^{(2)}} \right)}{d} \quad (10.14)$$

where $\mu^{(1)}$ and $\sigma^{(1)}$ are respectively mean and standard deviation of the vector $L^{(1)}$, whereas $\mu^{(2)}$ and $\sigma^{(2)}$ are respectively mean and standard deviation of the vector $L^{(2)}$;

$$S(L^{(1)}, L^{(2)}) = \sum_{n=1}^d (f_{\hat{\theta}_n}^{(1)} - f_{\hat{\theta}_n}^{(2)})^2 \quad (10.15)$$

$$S(L^{(1)}, L^{(2)}) = \sum_{n=1}^d w_n * (f_{\hat{\theta}_n}^{(1)} - f_{\hat{\theta}_n}^{(2)})^2 \quad (10.16)$$

where $w_n = f_{\hat{\theta}_n}^{(1)}$ if $f_{\hat{\theta}_n}^{(1)} \neq 0$, $w_n = 1$ otherwise.

The experiments pointed out that the best parameters to be used when the proposed holistic representation is coupled with KNN are the following: $d=32$, ζ equal to 10% of the maximal A_k extracted from the image I under consideration, the similarity measure based on Bhattacharyya Coefficient, and $K=3$. Results for *Natural* vs *Artificial* classification are reported in Table 10.1 (the x-axis represents the inferred classes while the y-axis represents the ground-truth category).

Table 10.1 : *Natural* vs *Artificial* classification result obtained employing KNN. The average classification rates for the *Natural* and *Artificial* classes are listed along the diagonal. The average accuracy is 94.10%.

	Natural	Artificial
Natural	94.15	5.85
Artificial	5.94	94.06

Good results are obtained when K-nearest neighbors (KNN) is used as classifier, but its implementation in domain with limited time, space and computational power resources (e.g., digital camera, mobile phone, etc.) isn't straightforward. Indeed, K-nearest neighbor is a memory based classifier³ whose computational costs and space resources exceed the constrained domain of single sensor imaging devices. To overcome these difficulties we propose to use a logistic model for classification purpose [25]. The basic assumption is that the difference between the logarithms of the class-conditional density functions is linear in the vectors \mathbf{f} representing the images through *TF-IDF-LDO*:

$$\log(P(\mathbf{f}|\textit{Artificial})) - \log(P(\mathbf{f}|\textit{Natural})) = w_0 + w_1 f_{\hat{\theta}_1} + \dots + w_d f_{\hat{\theta}_d} \quad (10.17)$$

Such basic assumption is equivalent to [25]:

$$P(\textit{Natural}|\mathbf{f}) = \frac{1}{1 + e^{(w'_0 + w_1 f_{\hat{\theta}_1} + \dots + w_d f_{\hat{\theta}_d})}} \quad (10.18)$$

$$P(\textit{Artificial}|\mathbf{f}) = \frac{e^{(w'_0 + w_1 f_{\hat{\theta}_1} + \dots + w_d f_{\hat{\theta}_d})}}{1 + e^{(w'_0 + w_1 f_{\hat{\theta}_1} + \dots + w_d f_{\hat{\theta}_d})}} \quad (10.19)$$

where $w'_0 = w_0 + \log\left(\frac{P(\textit{Artificial})}{P(\textit{Natural})}\right)$. In the experiments equiprobability is assumed for the priors $P(\textit{Natural})$ and $P(\textit{Artificial})$, hence $w'_0 = w_0$.

It is interesting to note that the decision about discrimination between *Natural* vs. *Artificial* scenes is determined solely by the following linear function:

$$g(\mathbf{f}) = w'_0 + w_1 f_{\hat{\theta}_1} + \dots + w_d f_{\hat{\theta}_d} \quad (10.20)$$

³e.g., the training set must be taken in memory for classification purpose.

Specifically, a new observation is assigned to the class *Natural* if the value of the function (10.20) is negative, otherwise is assigned to the class *Artificial*. Hence, after learning the parameters of the logistic classification model (out of the devices), a simple evaluation of a linear function can be used for classification purpose; this leads to overcome the difficulties due to the constrained domain of consumer imaging devices.

The experiments performed troughs KNN pointed out that the best representation of the scene is encoded in a 32-dimensional vector. For the logistic classifiers, a 33-dimensional vector relative to the parameters has been learned. The parameters may be estimated (e.g., out of the devices) using a classic Maximum Likelihood Estimation (MLE) approach [25]. More specifically, to learn the parameters of the logistic model the likelihood function is maximized using all the training observation. The function to be maximized is the following:

$$\log(L) = \sum_{\mathbf{f} \in \text{Artificial}} g(\mathbf{f}) - \sum_{\text{all } \mathbf{f}} \log[1 + \exp(g(\mathbf{f}))] \quad (10.21)$$

The gradient of (10.21) with respect to the involved parameters is

$$\frac{\partial \log(L)}{\partial w'_0} = |\text{Artificial}| - \sum_{\text{all } \mathbf{f}} P(\text{Artificial}|\mathbf{f}) \quad (10.22)$$

$$\frac{\partial \log(L)}{\partial w_j} = \sum_{\mathbf{f} \in \text{Artificial}} f_{\hat{\theta}_j} - \sum_{\text{all } \mathbf{f}} f_{\hat{\theta}_j} P(\text{Artificial}|\mathbf{f}), \quad j = 1, \dots, d \quad (10.23)$$

Taking into account the likelihood (10.21) and its derivative (10.22 and 10.23), an iterative optimization procedure can be used to obtain a set of parameter values $\hat{w}'_0, \dots, \hat{w}_d$ for which the function $\log(L)$ reaches a local maximum. The learning rules are the following (steepest gradient ascent in the likelihood):

$$w'_0{}^{[t+1]} = w'_0{}^{[t]} + \eta \frac{\partial \log(L)}{\partial w'_0{}^{[t]}} \quad (10.24)$$

$$w_j^{[t+1]} = w_j^{[t]} + \eta \frac{\partial \log(L)}{\partial w_j^{[t]}}, \quad j = 1, \dots, d \quad (10.25)$$

where η is a learning rate. These are batch learning rules, since all samples in the training set are considered at once⁴. In the experiments we use $\eta = 0.1$ and the initial starting values of logistic classification parameters are chosen randomly in $[-0.01, 0.01]$. The parameters are iteratively upgraded through the learning rules above until convergence in order to obtain the final parameters $\hat{w}'_0, \dots, \hat{w}_d$ to be used in the classification phase.

⁴For real-time learning and adaptation applications, an online learning rules considering one data sample at a time may be used.

Once the learning phase is completed, a new image can be assigned to a class $\hat{c} \in \{\textit{Natural}, \textit{Artificial}\}$ according to a maximum a posteriori (MAP) rule:

$$\hat{c} = \arg \max_c P(c | \mathbf{f}, \hat{w}'_0, \dots, \hat{w}'_d) \quad (10.26)$$

All experiments in which a logistic classifier is involved to discriminate between classes of scenes are repeated ten times with different randomly selected training (50%) and test images (50%). Image representation parameters are fixed by taking into account the experiments performed with KNN approach. The confusion matrices [42] were recorded at each run for evaluation purpose. The classification results are obtained averaging on the results of all ten runs.

In Table 10.2 the classification results obtained using the logistic classification model are reported. The average accuracy is 93.01%. Although the recognition results are about 1.09% less than the results obtained by using KNN (see Table 10.1 to compare KNN vs. Logistic results), one should not overlook that the proposed method outperforms KNN in constrained domain (e.g., limited resources in term of space, time and computational power).

Table 10.2 : Confusion matrix obtained through logistic classification. The average classification rates for the *Natural* and *Artificial* classes are listed along the diagonal. The average accuracy is 93.01%.

	Natural	Artificial
Natural	93.03	6.97
Artificial	6.70	93.30

Classification Performances on Other Classes of Scenes

The proposed classification framework can be applied to other classes of scene. This Section shows preliminary results obtained by applying the proposed approach on other classes of scene: *Open* vs. *Closed*, *Indoor* vs. *Outdoor*. These classes may be useful to properly address some parameters of IGP pipeline employed within imaging devices [15, 43]. Moreover, this Section presents a simple extension of the proposed method to work with multiple classes. Three classes of scenes that are usually acquired by a digital camera or a mobile phone are considered: *Landscape*, *Document* and *Portraits*. All the experiments of this Section have been done employing the logistic classification model with the same parameters pointed out by the experiments described in previous Sections.

First, let us examine the performance of the proposed approach to discriminate *Open* vs. *Closed* scenes. A *Closed* scene is a scene with small perceived depth, whereas an *Open* scene is a scene with big perceived depth. The database used for *Open* vs. *Closed* classification is composed of eight basic scene categories collected by the authors of [40]: *Coast* (360 images), *Open Country* (410 images), *Street* (292 images), *Highway* (260 images), *Forest* (328 images), *Mountain* (374 images), *Tall Building* (356 images), *City* (308 images). The first four basic classes have been considered as *Open* scenes (1322

images), whereas the other classes have been considered *Closed* scenes (1366 images). Strongly ambiguous scenes in term of *openness* were discarded (e.g., street scenes with no perceived depth). Images were considered in grayscale, even when color images are available. Some examples of *Open* and *Closed* scenes used in the experiments are given in Fig.(10.9). Tests on this dataset are repeated twenty times with different randomly selected training (75%) and test (25%) images. The parameters involved in the logistic classification model have been learned at each run from the training set through classic MLE procedure (see previous Section) and the confusion matrices were recorded at each run. The final *Open* vs. *Closed* classification results are obtained averaging on the results of all twenty runs.

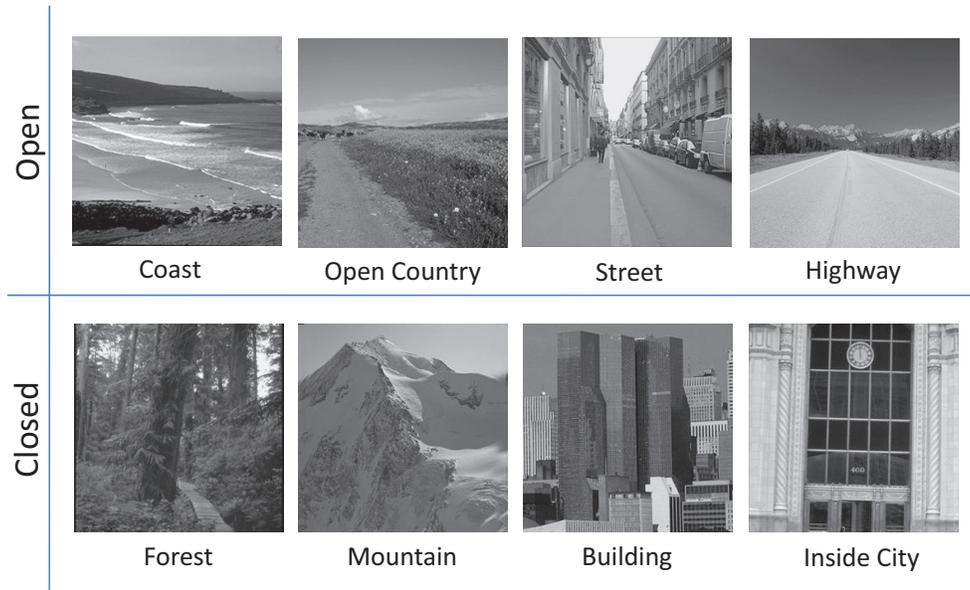
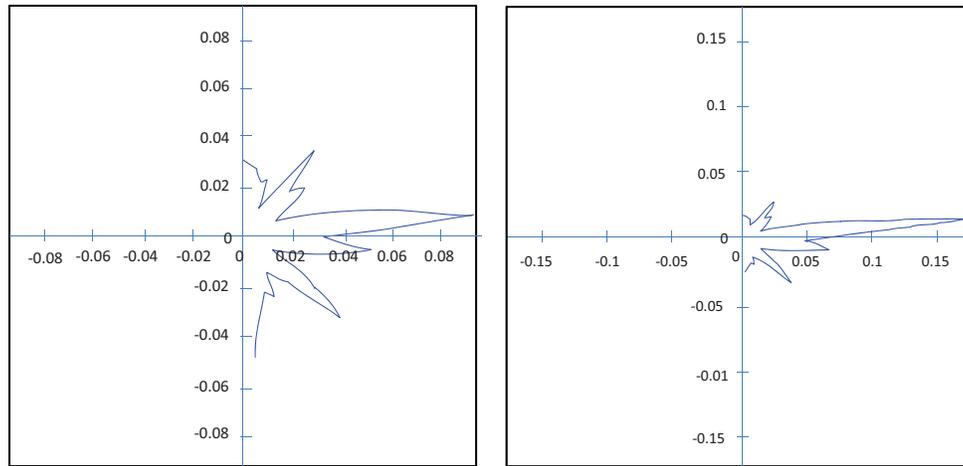


Figure 10.9 : Some examples of *Open* and *Closed* scenes used to compute the average local dominant orientations distributions reported in Fig.(10.10).

Fig.(10.10) reports the polar version of the mean local dominant orientations distributions of *Open* and *Closed* scenes. The distributions in Fig.(10.10(a)) have been computed averaging the *TF-IDF-LDO* representations of the *Open* scenes of the dataset described above. The distributions in Fig.(10.10(b)) have been computed averaging the *TF-IDF-LDO* representations of the *Closed* scenes of the dataset described above.

In Table 10.3 the classification results obtained on *Open* vs. *Closed* classification are reported (90.33% Accuracy).

Next, let us examine the performances of *In* vs. *Out* classification. The database used for these experiments is composed of nine basic scene categories collected in [27, 40]: *Coast* (360 images), *Highway* (260 images), *Mountain* (374 images), *Open Country* (410 images), *Street* (292 images), *Bedroom* (216 images), *Kitchen* (210 images), *Living Room* (289 images), *Office* (215 images). The first five basic classes have been considered as *Outdoor* scenes (1669 images), whereas the other classes have been considered *Indoor*



(a) Mean Polar *TF-IDF-LDO* of *Open* Scenes (b) Mean Polar *TF-IDF-LDO* of *Closed* Scenes

Figure 10.10 : The polar version of the mean local dominant orientations distributions of *Open* and *Closed* scenes.

Table 10.3 : *Open* vs. *Closed* classification results. The average classification rates for the *Open* and *Closed* classes are listed along the diagonal. The average accuracy is **90.33%**.

	Open	Closed
Open	89.56	10.44
Closed	8.65	91.35

scenes (930 images). Some examples of *Indoor* and *Outdoor* scenes, used in the experiments, are given in Fig.(10.11). Also in this case the images were considered in grayscale, even when color images are available. The experiments on this dataset are repeated twenty times with different randomly selected training (75%) and test (25%) images. The parameters involved in the logistic classification model have been learned at each run from the training set through classic MLE procedure (see previous Section) and the confusion matrices were recorded at each run. The final *Indoor* vs. *Outdoor* classification results are obtained averaging on the results of all twenty runs.

Fig.(10.12) reports the polar version of the mean local dominant orientations distributions of *Outdoor* and *Indoor* scenes. The distributions in Fig.(10.12(a)) have been computed averaging the *TF-IDF-LDO* representations of the *Outdoor* scenes of the dataset described above. The distributions in Fig.(10.12(b)) have been computed averaging the *TF-IDF-LDO* representations of the *Indoor* scenes of the dataset described above. Also in this case the “shapes” of the mean *TF-IDF-LDOs* of the two involved classes are quite different.

In Table 10.4 the classification results obtained on *Outdoor* vs. *Indoor* classification are reported (90.89% Accuracy).

Finally, let us examine a simple extension of the proposed approach to work with

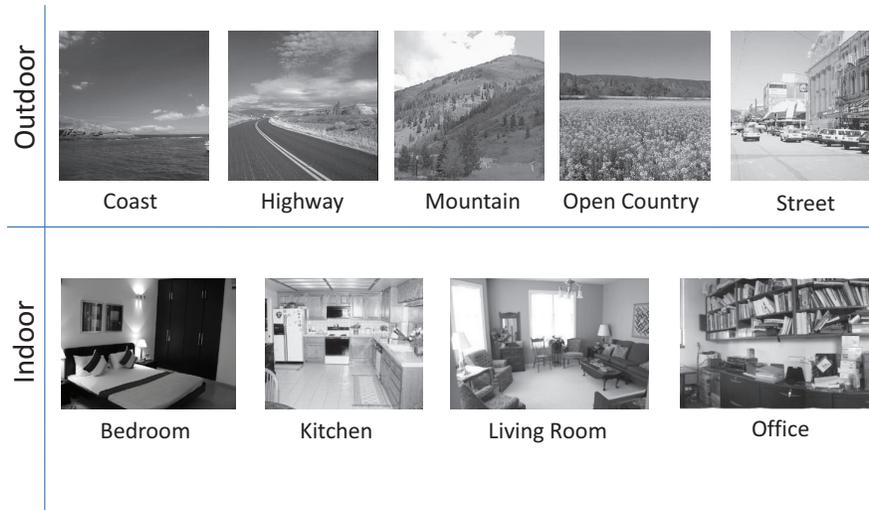


Figure 10.11 : Some examples of *Outdoor* and *Indoor* scenes used to compute the average local dominant orientations distributions reported in Fig.(10.12).

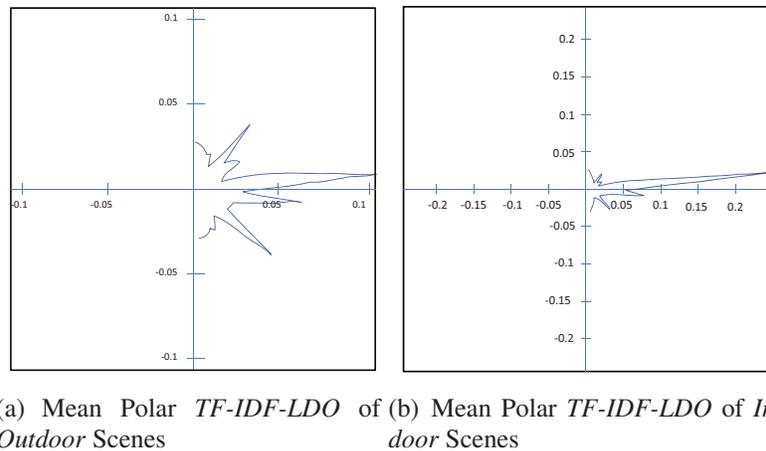


Figure 10.12 : The polar version of the mean local dominant orientations distributions of *Outdoor* and *Indoor* scenes.

Table 10.4 : *Outdoor* vs. *Indoor* classification results. The average classification rates for the *Outdoor* and *Indoor* classes are listed along the diagonal. The average accuracy is 90.89%.

	Outdoor	Indoor
Outdoor	90.39	9.61
Indoor	8.60	91.40

multiple classes. Specifically, we are interested in discriminating between three classes: *Document*, *Landscape*, *Portrait*. Scenes belonging to these three classes are usually acquired by a consumer imaging device. The database used in the experiments is composed

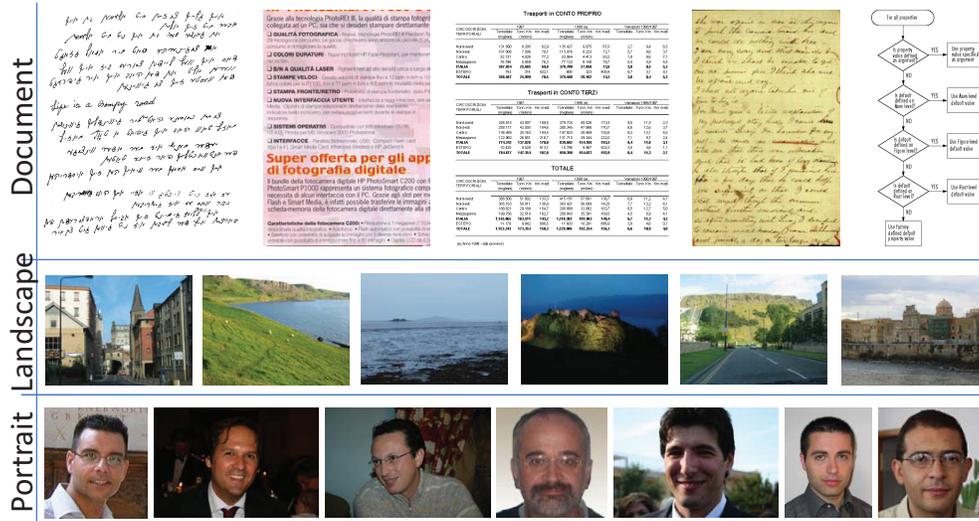


Figure 10.13 : Some examples of Document, Landscape and Portrait scenes used to compute the average local dominant orientations distributions reported in Fig.(10.15).

by 1532 colour images. In particular, 382 images are Landscape scenes, 874 images are Document scenes and 276 images are Portraits scenes. Some examples of images that have been used in our experiments are depicted in Fig.(10.13). All the images were pre-processed to be considered in 256×256 grayscale thumbnail version (Fig.(10.14)). The 32×32 DCT blocks of each thumbnail have been used to build the TF-IDF-LDO representation as described in Section 10.3.1. Also in this case we used $d=32$ orientation for the TF-IDF-LDO representation, and a strength threshold ζ equal to 10% of the maximal A_k extracted from the image I under consideration.

Fig.(10.15) reports the polar version of the mean local dominant orientations distributions of the three involved classes of scenes. The distributions in Fig.(10.15) have been computed averaging the TF-IDF-LDO representations of the Document, Landscape and Portrait scenes of the dataset. Also in this case the “shapes” of the mean TF-IDF-LDOs are quite different.

To perform our experiments on the $N = 3$ classes of scenes mentioned above, we employed the one-against-all method [25]. This method constructs N binary classifiers (e.g., N Binary Logistic Classifiers). The i^{th} classifier is trained to discriminate samples in class C_i (the positive class) from those in the remaining classes (the negative class). Thus, using a logistic model as binary classifier, after the training phase of all N binary classifiers through classic MLE procedure, the corresponding N binary discrimination functions (10.27) are evaluated to establish the class of a new sample \mathbf{f} .

$$g_i(\mathbf{f}) = w'_{i,0} + w_{i,1}f_{\hat{\theta}_1} + \dots + w_{i,d}f_{\hat{\theta}_d}, \quad i=1, \dots, N \tag{10.27}$$

Ideally, for a given sample \mathbf{f} , the quantity $g_i(\mathbf{f})$ will be positive for one value of n and

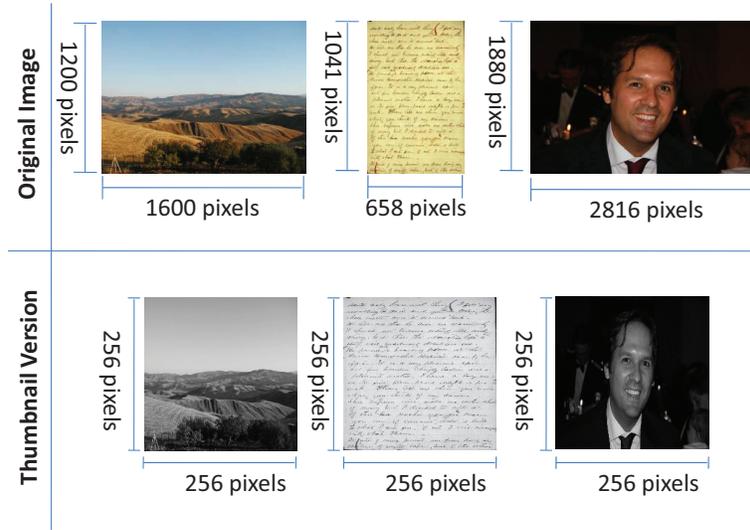


Figure 10.14 : *Landscape, Document and Portraits* images were preprocessed and resized to work with 256×256 grayscale thumbnail version.

negative for the remainder, giving a clear indication of the class. If there is more than one class for which the quantity $g_i(\mathbf{f})$ is positive, the \mathbf{f} sample may be assigned to the class $\hat{c} \in \{C_1, C_2, \dots, C_N\}$ for which the distance to the hyperplane is the largest (10.28).

$$\hat{c} = \arg \max_i \left(\frac{g_i(\mathbf{f})}{\|\mathbf{w}_i\|} \right) \quad (10.28)$$

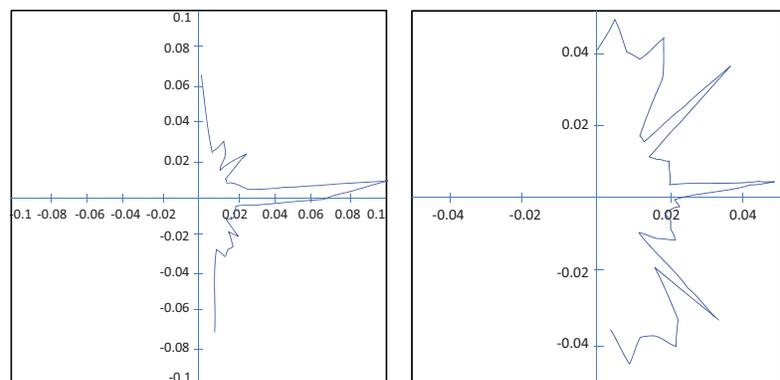
If all values $g_i(\mathbf{f})$ are negative, then the \mathbf{f} sample is assigned to the class with smallest distance to the hyperplane (10.29).

$$\hat{c} = \arg \min_i \left(\frac{g_i(\mathbf{f})}{\|\mathbf{w}_i\|} \right) \quad (10.29)$$

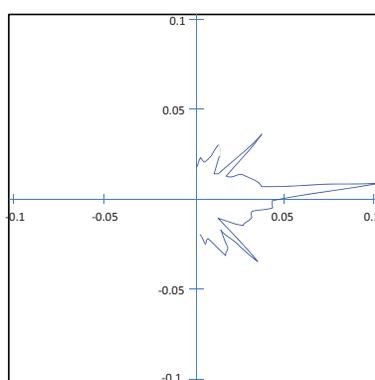
The experiments on the dataset containing *Landscape, Document* and *Portraits* scenes have been repeated twenty times with different randomly selected training (50%) and test (50%) images. The parameters involved in the multi-class logistic model have been learned at each run from the training set and the confusion matrices were recorded at each run. The final classification results are obtained averaging on the results of all twenty runs. Table 10.5 reports the confusion matrix obtained averaging on the results of all twenty runs. The average accuracy is 87.62%.

10.4 Red Eyes Detection

Since the large diffusion of mobile devices with embedded camera and flashgun, the red-eyes artifacts have de-facto become a critical problem. Different methods have been proposed in literature [44–46] to detect and remove red-eyes artifacts (see Chapter 8 for



(a) Mean Polar *TF-IDF-LDO* of Document Scenes of (b) Mean Polar *TF-IDF-LDO* of Landscape Scenes



(c) Mean Polar *TF-IDF-LDO* of Portrait Scenes

Figure 10.15 : The polar version of the mean local dominant orientations distributions of *Document*, *Landscape* and *Portrait* scenes.

Table 10.5 : Classification results obtained applying the proposed approach on the three classes usually acquired by an imaging device: *Portrait*, *Landscape* and *Document*. The average classification rates for the three classes are listed along the diagonal. The average accuracy is **87.62%**.

	Portrait	Landscape	Document
Portrait	94.15	2.05	3.8
Landscape	4.34	87.00	8.66
Document	10.18	8.09	81.73

more details). In this Section a two stage algorithm for red-eye detection is presented. The first stage is mainly based on the technique proposed in [47] taking into account a different color space and only a subset of the constraints about the roundness of the red-eyes regions. Candidate red-eye patches are extracted from the input image through an image filtering pipeline as detailed in Chapter 8. In the second stage a multi-modally

classifier, obtained by using clustering and Linear Discriminant Analysis (LDA), is used to distinguish between true red-eye patches vs. other patches [44]. The proposed cluster-based LDA is used to deal with the multi-modally nature of the input space.

10.4.1 Red Patch Categorization

The main aim of the categorization stage is the elimination of false positive red-eyes in the set of patches obtained performing the filtering pipeline described in Chapter 8.

At this stage the problem become a binary classification problem. In particular, the aim is to discriminate between *eyes* vs. *other* patches. Among the discriminative methods successfully used for binary classification, Linear Discriminant Analysis (LDA) [25] is probably the most widely used. LDA assumes a prior knowledge about the classes of the training data and transforms the input data space into a new one useful for classification. In the transformed space the classification is performed through a decision function (e.g., by using nearest neighbors rule). Specifically, LDA find the projection directions \mathbf{D} on

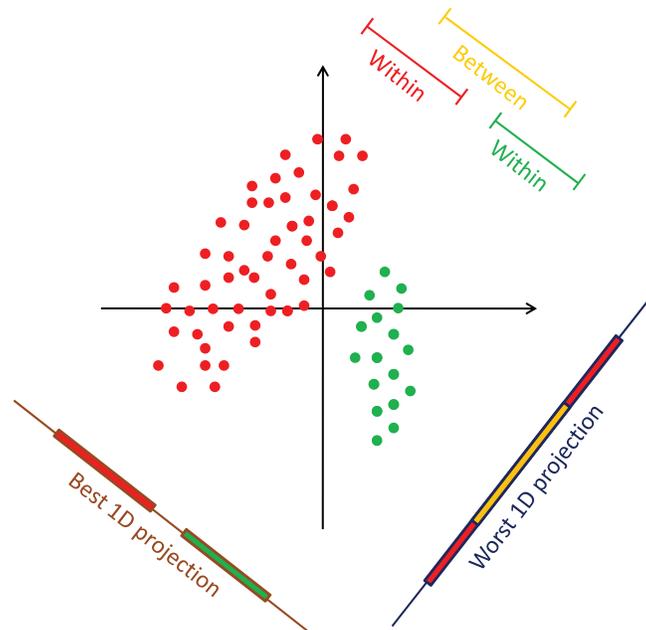


Figure 10.16 : Linear Discriminant Analysis finds a linear transformation function so that when the samples of different categories are projected on the new direction, between-class variance is maximized, whereas within-class variance is minimized. In this example, the best projection and the worst projection correspond respectively to the directions colored in brown and blue. Samples projected on to the worst direction overlap (i.e., the yellow region on the blue line). The two classes are separate considering the best projection (brown line).

which the within-class scatter is minimized while the between-class scatter is maximized

(Fig.(10.16)). This is done maximizing the ratio of the determinant of the between-class scatter matrix \mathbf{B} to the determinant of the within-class scatter matrix \mathbf{W} of the training data:

$$\mathbf{D} = \underset{\mathbf{U}}{\operatorname{argmax}} \frac{|\mathbf{U}^T \mathbf{B} \mathbf{U}|}{|\mathbf{U}^T \mathbf{W} \mathbf{U}|} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_N]. \quad (10.30)$$

The solution $\{\mathbf{d}_i | i = 1, 2, \dots, N\}$ is a set of eigenvectors of \mathbf{B} and \mathbf{W} (i.e., $\mathbf{B}\mathbf{d}_i = \lambda_i \mathbf{W}\mathbf{d}_i$) [25]. In the classification stage, the new data samples are simply projected onto the LDA directions to form feature vectors to be classified by using a decision function. Usually, Principal Component Analysis (PCA) is performed first to overcome the complication of a singular within-class scatter matrix [48].

LDA has been successfully employed for different classification task (e.g., face recognition). However, the method fails to solve non-linear problems [49], because LDA only considers a single linear transformation in the global coordinate system. Due to the multimodally nature of the red-eye patches (i.e., colours, orientation, shape, etc.), a simple LDA could fail during classification task. To overcome this problem the technique proposed in this Section performs a clustering of the input space first, and then apply LDA on each cluster. More specifically, during the learning phase, the patches are clustered by using K-means [25] in their original color space producing the subsets of the input patches with the relative prototypes; hence an LDA transformation is learned for each cluster. PCA transformation is performed for each cluster and the dimension of the input space is reduced before LDA computation. During the classification stage, a new patch is assigned to a cluster according to the closest prototype, then the patch is transformed by using the PCA and LDA basis corresponding to the associated cluster. The features obtained projecting patches on the LDA space are used in combination with a simple decision function to achieve the final classification. Specifically, for each cluster we have considered the projection of patches on the LDA direction corresponding to the maximum eigenvalue. The classification of a patch is hence obtained taking into account a threshold properly learned for the cluster under consideration.

10.4.2 Experimental Settings and Results

The described red-eye detector system has been tested on a dataset of 450 images in which 1056 red-eyes have been manually labeled. The dataset has been collected from various sources, including DSLR images, compact cameras, personal collections and Internet photos. Single red-eyes, as well as high variability of red-eyes colors, poses and shapes have been considered in building the dataset.

In the first stage of the aforementioned method, the filtering pipeline (see Chapter 8.2.5) has pointed out 4081 patches as possible red-eyes candidates. In particular, 957 patches were true positives whereas 3124 patches were false positives. These patches have been further classified by using the method described in Section 10.4.1. In order to evaluate the classification performance of the proposed cluster-based LDA, the leave-one-out cross validation procedure (LOOCV) has been employed. Each run of LOOCV has involved a single patch obtained from the red-eyes extraction phase (see Chapter 8.2.5)

as test to be classified, and the remaining patches as the training data. This is repeated to guarantee that each input patch is used once as patch to be classified.

At each run of LOOCV the training patches have been clustered and then LDA learning phase has been performed on each cluster after dimensional reduction in the PCA space (1% of reconstruction error). For each cluster, the direction related to the maximum eigenvalue of the corresponding LDA transformation has been considered as discriminative feature for classification purpose. The threshold used as discriminative function for each cluster-based LDA has been fixed in order to obtain the best trade-off between false positive (to be minimized) and hit rate (to be maximized) for the training set under consideration.

In the classification stage, the test patch has been first assigned to the closest cluster taking into account the learned prototypes, then projected in the reduced PCA space, and finally projected on the corresponding LDA direction. The derived feature is used for classification taking into account the threshold associated to the selected cluster. The final *red-eye* vs. *other* classification results have been obtained averaging on the results of the overall LOOCV runs.

The average accuracy obtained with the proposed method was 99.68%. These results have been obtained by using three clusters (Fig.(10.17)). Table 10.6 reports the average confusion matrix obtained considering all the leave-one-out runs (the x-axis represents the inferred classes while the y-axis represents the ground-truth category). The proposed method obtained high hit rate and very low false positive rate on the patches detected through the image filtering pipeline.

In order to point out the usefulness of the cluster-based LDA, the classification tests have been repeated by employing classic LDA. In Table 10.7 the average confusion matrix obtained employing classic LDA and LOOCV procedure is reported. The results confirm that a cluster-based LDA is useful to reduce the false positive rate since it can better deal with multi-modally nature of the data.

In Fig.(10.18) two examples of misclassified patches are reported. In Fig.(10.18(a)) a “golden” eye is depicted (another possible artefact due to similar acquisition problem). The underlying structure of the patch in Fig.(10.18(b)) is probably the main reason of misclassification.

Taking into account both the filtering and the classification stages, the overall accuracy of the proposed red-eyed detector is 90.06%. This means that 951 red-eyes have been correctly detected with respect to the 1056 red-eyes of the 450 input images, whereas only 7 false positives have been introduced. These results compares favorably with accuracy performances of other state-of-the-art solutions [50, 51].

Table 10.6 : Confusion matrix obtained employing cluster based LDA.

	Eyes	Other
Eyes	99.37	0.63
Other	0.22	99.78

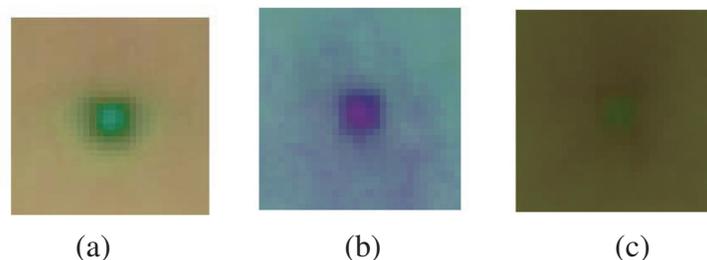
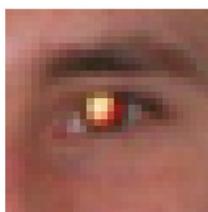


Figure 10.17 : Example of clusters prototypes obtained in a LOOCV run.

Table 10.7 : Confusion matrix obtained employing classic LDA.

	Eyes	Other
Eyes	99.58	0.42
Other	8.52	91.48



(a) False Negative.



(b) False Positive.

Figure 10.18 : Examples of misclassified patches.

10.5 Summary and Conclusion

The problem of categorization is currently of great interest for Computer Vision research community. Basic concepts useful to build a categorization engine by learning from image examples have been briefly introduced in Section 10.2. Basic Computer Vision and Machine Learning techniques useful for scene recognition and red-eye detection in the context of single sensor imaging devices have been presented respectively in Section 10.3 and 10.4. Since the new concept of camera 2.0 is becoming more and more realistic [52], and considering the recent advancements in exploiting the content of images as intermediate steps for many other related tasks (e.g., object detection, color constancy, etc), lead us to believe that the development of imaging pipeline involving complex task represents an important challenge in the research area of Image Processing for Embedded Devices.

Bibliography

- [1] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman, 1982.
- [2] S. Thorpe, D. Fize, and C. Marlot, “Speed of processing in the human visual system,” *Nature*, vol. 338, pp. 520–522, June 1996.
- [3] I. Biederman, “Aspects and extension of a theory of human image understanding,” in *Computational Processes in Human Vision: An Interdisciplinary Perspective*, 1988.
- [4] A. Oliva and A. Torralba, “Building the gist of a scene: The role of global image features in recognition,” *Visual Perception, Progress in Brain Research*, pp. 251–256, 2006.
- [5] L. W. Renninger and J. Malik, “When is scene recognition just texture recognition?,” *Vision Research*, vol. 44, pp. 2301–2311, 2004.
- [6] J. Vogel, A. Schwaninger, C. Wallraven, and H. H. Bühlhoff, “Categorization of natural scenes: Local versus global information and the role of color,” *ACM Transactions on Applied Perception*, vol. 4, no. 3, p. 19, 2007.
- [7] A. Quattoni and A. Torralba, “Recognizing indoor scenes,” in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR-09)*, pp. 413–420, 2009.
- [8] B. Heisele, A. Verri, and T. Poggio, “Learning and vision machines,” *Proceedings of the IEEE*, vol. 90, pp. 1164–1177, July 2002.
- [9] J. Vogel and B. Schiele, “Semantic modeling of natural scenes for content-based image retrieval,” *International Journal of Computer Vision*, vol. 72, pp. 133–157, April 2007.
- [10] A. Torralba, “Contextual priming for object detection,” *International Journal of Computer Vision*, vol. 53, no. 2, pp. 169–191, 2003.
- [11] P. Ladret and A. Guérin-Dugué, “Categorisation and retrieval of scene photographs from JPEG compressed database,” *Pattern Analysis & Application*, vol. 4, pp. 185–199, June 2001.
- [12] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin, “Context-based vision system for place and object recognition,” in *IEEE International Conference on Computer Vision (ICCV-03)*, pp. 273–280, 2003.
- [13] S. Battiato, G. M. Farinella, G. Giuffrida, C. Sismeiro, and G. Tribulato, “Using visual and text features for direct marketing on multimedia messaging services domain,” *Multimedia Tools Applications*, vol. 42, no. 1, pp. 5–30, 2009.

- [14] S. Battiato, G. Farinella, G. Giuffrida, C. Sismeiro, and G. Tribulato, "Exploiting visual and text features for direct marketing learning in time and space constrained domains," *Pattern Analysis and Applications*, 2009. <http://dx.doi.org/10.1007/s10044-009-0145-2>.
- [15] S. Bianco, G. Ciocca, C. Cusano, and R. Schettini, "Improving color constancy using indoor-outdoor image classification," *IEEE Transactions on Image Processing*, vol. 17, pp. 2381–2392, December 2008.
- [16] F. Naccari, S. Battiato, A. Bruna, A. Capra, and A. Castorina, "Natural scene classification for color enhancement," *IEEE Transactions on Consumer Electronics*, vol. 51, pp. 234–239, February 2005.
- [17] S. Battiato, M. Mancuso, A. Bosco, and M. Guarnera, "Psychovisual and statistical optimization of quantization tables for DCT compression engines," in *IEEE International Conference on Image Analysis and Processing (ICIAP-01)*, pp. 602–606, 2001.
- [18] S. Battiato, A. Bosco, A. Castorina, and G. Messina, "Automatic image enhancement by content dependent exposure correction," *EURASIP Journal on Applied Signal Processing*, vol. 12, no. 3, pp. 1849–1860, 2004.
- [19] S. Battiato, G. M. Farinella, G. Gallo, and E. Messina, "Naturalness classification of images into DCT domain," in *SPIE-IS&T 21th Annual Symposium Electronic Imaging Science and Technology 2009 - Digital Photography V (SPIE-09)*, 2009.
- [20] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, August 2006.
- [21] T. Jebara, *Machine Learning: Discriminative and Generative (Kluwer International Series in Engineering and Computer Science)*. Norwell, MA, USA: Kluwer Academic Publishers, 2003.
- [22] J.-H. Xue and D. M. Titterington, "Comment on discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," *Neural Processing Letters*, vol. 28, no. 3, pp. 169–187, 2008.
- [23] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Advances in Neural Information Processing Systems (NIPS-02)* (T. G. Dietterich, S. Becker, and Z. Ghahramani, eds.), (Cambridge, MA), pp. 841–848, MIT Press, 2002.
- [24] B. Efron, "The efficiency of logistic regression compared to normal discriminant analysis," *Journal of the American Statistical Association*, vol. 70, no. 352, pp. 892–898, 1975.
- [25] A. R. Webb, *Statistical Pattern Recognition (2nd Edition)*. John Wiley & Sons, Ltd., November 2002.

- [26] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification," in *Advances in Neural Information Processing Systems (NIPS-99)*, pp. 547–553, 1999.
- [27] F.-F. Li and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR-05)*, pp. 524–531, IEEE Computer Society, 2005.
- [28] A. Bosch, A. Zisserman, and X. Muñoz, "Scene classification via pLSA," in *European Conference on Computer Vision (ECCV-06)*, pp. 517–530, 2006.
- [29] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR-06)*, pp. 2169–2178, 2006.
- [30] S. Battiato, G. M. Farinella, G. Gallo, and D. Ravì, "Scene categorization using bag of textons on spatial hierarchy," in *IEEE International Conference on Image Processing (ICIP-08)*, pp. 2536–2539, 2008.
- [31] A. Bosch, A. Zisserman, and X. Muñoz, "Scene classification using a hybrid generative/discriminative approach," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 712–727, 2008.
- [32] S. Battiato, G. M. Farinella, G. Gallo, and D. Ravì, "Spatial hierarchy of textons distributions for scene classification," in *15th International Multimedia Modeling Conference on Advances in Multimedia Modeling (MMM-09)*, vol. 5371 of Lecture Notes in Computer Science, pp. 333–343, Springer-Verlag, 2008.
- [33] G. Farinella and S. Battiato, *Representation Models and Machine Learning Techniques for Scene Classification*, ch. Pattern Recognition and Machine Vision. P. S. P. Wang, River Publishers, 2010.
- [34] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *ECCV International Workshop on Statistical Learning in Computer Vision*, pp. 1–22, 2004.
- [35] A. Torralba and A. Oliva, "Semantic organization of scenes using discriminant structural templates," in *IEEE International Conference on Computer Vision (ICCV-99)*, pp. 1253–1258, 1999.
- [36] J. Luo and M. R. Boutell, "Natural scene classification using overcomplete ICA," *Pattern Recognition*, vol. 38, no. 10, pp. 1507–1519, 2005.
- [37] G. M. Farinella, S. Battiato, G. Gallo, and R. Cipolla, "Natural versus artificial scene classification by ordering discrete Fourier power spectra," in *International Workshop on Structural, Syntactic, and Statistical Pattern Recognition (SSPR & SPR - 08)*,

- vol. 5342 of Lecture Notes in Computer Science, pp. 137–146, Springer-Verlag, 2008.
- [38] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” in *Information Processing & Management*, vol. 24, p. 513523, 1988.
- [39] B. Shen and I. K. Sethi, “Direct feature extraction from compressed images,” in *Storage and Retrieval for Image and Video Databases (SPIE-96)*, pp. 404–414, 1996.
- [40] A. Oliva and A. Torralba, “Modeling the shape of the scene: a holistic representation of the spatial envelope,” *International Journal of Computer Vision*, vol. 42, pp. 145–175, 2001.
- [41] A. Guérin-Dugué and A. Oliva, “Classification of scene photographs from local orientations features,” *Pattern Recognition Letters*, vol. 21, no. 13-14, pp. 1135–1140, 2000.
- [42] R. Kohavi and F. Provost, “Glossary,” *Machine Learning Journal*, vol. 30, p. 271274, 1998.
- [43] R. Lukac, *Single-Sensor Imaging: Methods and Applications for Digital Cameras*. CRC Press, 2008.
- [44] S. Battiato, G. Farinella, M. Guarnera, G. Messina, and D. Ravì, “Red-eyes removal through cluster based linear discriminant analysis,” in *International Conference on Image Processing (ICIP 2010)*, (Hong Kong), 2010.
- [45] S. Battiato, G. Farinella, M. Guarnera, G. Messina, and D. Ravì, “Boosting gray codes for red eyes removal,” in *International Conference on Pattern Recognition (ICPR 2010)*, (Istanbul (TK)), 2010.
- [46] S. Battiato, G. Farinella, M. Guarnera, G. Messina, and D. Ravì, “Red-eyes removal through cluster based boosting on gray codes,” *EURASIP Journal on Image and Video Processing, Special Issue on Emerging Methods for Color Image and Video Quality Enhancement*, 2010. To appear.
- [47] F. Volken, J. Terrier, and P. Vandewalle, “Automatic red-eye removal based on sclera and skin tone detection,” in *European Conference on Color in Graphics, Imaging and Vision (CGIV-06)*, pp. 359–364, 2006.
- [48] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.
- [49] T.-K. Kim and J. Kittler, “Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 318–327, 2005.

- [50] I. V. Safonov, “Automatic red-eye detection,” in *International conference on the Computer Graphics and Vision (GraphiCon-07)*, Moscow, Russia, 2007.
- [51] A. M. Ferman, “Automatic detection of red-eye artifacts in digital color photos,” in *IEEE International Conference on Image Processing (ICIP-08)*, San Diego, California, vol. 1, pp. 617–620, 2008.
- [52] Stanford Computer Graphics Laboratory, “Camera 2.0: New computing platforms for computational photography.” <http://graphics.stanford.edu/projects/camera-2.0/>.

Image and Video Coding and Formatting

A. Bruna, A. Buemi, G. Spampinato

Advanced System Technology - Catania Lab, STMicroelectronics, Italy.

Abstract: The data acquired by the sensor have to be processed by the coprocessor or the host microprocessor, so both the systems have to share the same communication protocol and data format. Moreover, at the end of the image generation pipeline the image must be coded in a standard format in order to be read by every external device. Usually the sensor provides the image in the Bayer format. In the past the Bayer data were stored and transmitted using proprietary format and protocol; such solution has the drawback that every designer had to use the same proprietary interface to manage the sensor data. In the latest years the majority of companies making, buying or specifying image devices proposed a new standard called Standard Mobile Imaging Architecture (SMIA). It allows interconnecting sensors and hosts of different vendors. Concerning the output of the coprocessor, several standard formats are available. For still images the most frequently used are the Joint Picture Expert Group (JPEG) with a lossy compression, the Targa Interchange Format (TIF) with a lossless compression. In the top level imaging devices the output of the sensors can also be stored directly by making use of a proprietary file format, such as the Nikon Electronic Image Format (NEF), the Canon RAW File Format (CRW), etc. For videos the most used are MJPEG, MPEG-4, H.263 and H264 standards. This chapter besides presenting the main data formats gives also a short description to the next JPEG XR Image Coding Standard. Moreover some techniques concerning the compression factor control and the error detection and concealment are introduced.

11.1 Introduction and Motivation

Images and videos contents require a huge amount of memory to be stored. As example, a 12 Mpixels RGB image requires 34 MBytes using 8 bits per color channel. Although it could not be a problem with the actual memory size, it is convenient to compress images for different reasons: data are redundant and there is no reason to store them uncompressed; less size means less time to be transmitted, less bandwidth, less power consumptions, etc. Concerning the videos the problem is more evident: for an HDTV video 720p format (1280x720 pixels), at 50 HZ (frames per seconds), the required memory should be 132 MBytes/s if uncompressed! The compression is performed at the end of the image pipeline, just before the data transmission or storing. We mention here also algorithms to compress the Bayer data, a desirable feature employed in many situations. Any device containing an image sensor improves its performances if the amount of data to be processed is reduced. Thus, in an imaging device the transmissions of coded data from the sensor to the co-processor allows saving memory and bandwidth, whilst in imaging devices for video conferencing the input compression provides both better frame rates and bigger image size, that means reduced space requirements and increased image quality [1]. One more application would be wireless phones that send and receive still or video images [2]. Moreover the software pipeline requires a frame buffer whose size could decrease since the compression makes possible transmission of reduced data. The conventional scheme of an Image Generation Pipeline (IGP) consists of three basic steps:

the acquisition of the image in Color Filter Array (CFA) format, the color interpolation of the captured data yielding a full color image and the compression for image storing. Such scheme implies that the redundancy is firstly introduced by the demosaicing process and then reduced in the compression step. Moving the coding operation before the pixel interpolation retains more pertinent information allowing lower compression ratio and higher image quality [3]. But is important to point out that any error introduced in the co-decoding process of the data captured in CFA pattern is propagated to the following stages of the pipeline, with the risk of introducing artifacts to the final image. For this reason the basic requirement of a Bayer pattern coding algorithm is to preserve the information providing lossless or visually lossless compression, at the cost of a limited compression ratio. Other constraints are efficiency, low computational and memory costs and a Fixed Length Coding (FLC), especially for application that requires real time transmission.

11.2 Still Image Compression Formats

Image compression is basically the application of data compressions techniques on digital images. The compression of still images, in particular, is based on reducing the spatial redundancies. A lot of algorithms are in literature and they can be grouped in two branches: lossless (without loss of information) and lossy (some information is discarded). In imaging devices both methods are used, depending on the application. The most used still image compression format is the JPEG. Concerning the lossless data formats, there is not a standard: every company has its own file format (RAW data format). The JPEG 2000 is also described; despite its performances, it is not yet used due to its high complexity.

11.2.1 JPEG

The JPEG compression standard [4] was released in 1986. Its name stands for "Joint Photographic Experts Group" [5] (i.e., the committee that defined it). The JPEG became - and still is - the most used image compression methods since it allows a good trade-off between compression factor and image quality. Best results are obtained for natural images, while the behavior to synthetic images is not good. Moreover, increasing the resolution, the compression ratio usually increases. Despite other formats (sometimes better than the JPEG) have been developed in the last years, the JPEG is still the most used. The standard is composed by 4 parts:

- Part 1 [4] describes the requirements and guidelines;
- Part 2 [6] describes the compliance testing (i.e., the tests and the results to be conform to the Part 1);
- Part 3 [7] is an extension and it contains additional guidelines to increase the standard performances;
- Part 4 [8] describes methods used in the extended JPEG.

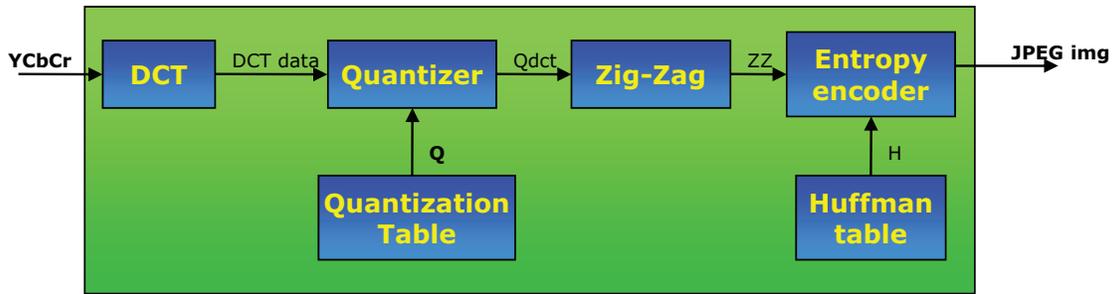


Figure 11.1 : JPEG encoder scheme.

It must be noted that the JPEG Part 1 describes a method of compressing digital image, regardless the image formatting (e.g., YUV, RGB, etc.) but the common de facto image format is YCrCb, as described in the SPIFF document (i.e., Part 4 of the standard).

Along the years, a lot of work was focused to optimize each step of the algorithm [9–11] obtaining both hardware or software implementations, some of them available for free. Other works are focused on error detection and concealment [12], on better visual quality [13–16] and to control the compression factor [17–19]. We will discuss about all these aspects in the next sections.

JPEG Encoder and Decoder Basic Schemes

Fig.(11.1) shows the basic schema of a typical JPEG encoder. The input image is usually represented in the RGB data format. Red, green and blue data are highly correlated [20]. In order to increase the compression factor, it is better to decorrelate data to further discard less important information. The better decorrelation space for this purpose is the YCbCr. The converting formulas are:

$$\begin{cases} Y &= 0.299R + 0.587G + 0.114B \\ Cb &= (B - Y)/2 + 0.5 \\ Cr &= (R - Y)/2 + 0.5 \end{cases} \quad (11.1)$$

Moreover, since the human visual system is more sensitive to luminance than to chrominance data, the chromatic channels are usually sub-sampled. The most used sub samplings are the so-called 4:4:4 (i.e., no sub-sampling), the 4:2:2 (chromatic data are sub-sampled only in the horizontal direction) and the 4:2:0 (chromatic data are sub-sampled both in the horizontal and in the vertical directions). The image is then partitioned into non-overlapping 8 x 8 blocks. The forward Discrete Cosine Transform (DCT) is applied to each block. The main advantages of DCT are:

- The energy compaction performance is nearly optimal, closest to the KLT (Karhunen-Loeve Transform) [20];
- The DCT coefficients are real numbers;

- DCT is a reversible linear transform and provides a set of orthonormal discrete basis functions;
- Many fast algorithms for forward and inverse DCT are known [21, 22].

The conversion formula is the following:

$$F(u, v) = \frac{1}{4}C(u)C(v) \left[\sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right] \quad (11.2)$$

While the inverse formula is:

$$f(x, y) = \frac{1}{4} \left[\sum_{x=0}^7 \sum_{y=0}^7 C(u)C(v)F(u, v) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right] \quad (11.3)$$

where:

$$\begin{cases} C(u), C(v) = \frac{1}{\sqrt{2}} & \text{if } u, v = 0 \\ C(u), C(v) = 1 & \text{otherwise} \end{cases} \quad (11.4)$$

$f(x, y)$ are the image data in the spatial domain, $F(u, v)$ are the image data in the transformed domain.

The 64 data are quantized according to a quantization table. The quantization factors are not defined in the standard and can differ between luminance and chrominances. Lower are the quantization factors, higher is the image quality. The standard suggests two quantization tables (Q_y for the luminance and Q_c for both Cr and Cb components), as shown below:

$$Q_y = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \quad (11.5)$$

$$Q_c = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix} \quad (11.6)$$

A rounding is also performed to obtain integer values. The first coefficient $F(0, 0)$ of each block is called *DC* while the others are called *AC* coefficients. The *DC*, in particular,

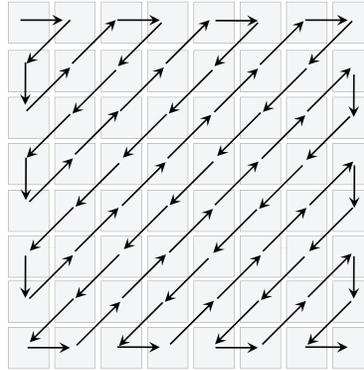


Figure 11.2 : Zig zag scan.

is proportional to the mean value of the block in the spatial domain. Usually it is similar to the adjacent blocks, so the difference with the *DC* coefficient of the previous block is used for encoding. Conversely, the *AC* coefficients are difficult to predict, so they are encoded directly. A zig zag reordering according to the Fig.(11.2), is then employed to obtain a single vector. In this way a long sequence of zero coefficients is placed in the latest elements of the vector. The last step is the entropy encoder. It is composed by a run-length and a variable length encoder. The *DC* value is encoded as difference with the *DC* of the previous block and then using a variable length encoder. The *AC* data are firstly grouped in pairs containing the number of zeros and the non zero value. If all the remaining values are zero a special word, called End Of Block (EOB), is used. A further variable length encoder generates a word for every pair. The standard does not impose a particular variable codes table, but suggests one table for the *DC* and one for the *AC* coefficients. It is up to the implementation to use the standard or its own tables. The decoding process is composed by the same processes but in the reverse order, i.e. variable length decoding, inverse zig zag reordering, *DC* computation (by summing the value of the previous block), inverse quantization and inverse *DCT*.

11.2.2 RAW Data Format

The JPEG standard is basically a lossy algorithm used in the imaging devices for encoding after the image reconstruction. In the last years the customers asked for the raw data (i.e., the data coming directly from the sensor without any processing or loss of information). So the manufacturers, especially for high end still cameras, inserted the possibility to save directly the raw data coming from the sensor. Even if some attempts were done to standardize the raw file format (e.g., Adobe's DNG, OpenEXR, etc.), each company has its own (proprietary) file format: Nikon (NEF), Canon (CR2), Sony (SR2), etc. Usually these file formats are based on the TIFF specifications and they contains not only the raw data but also meta-data (e.g., exposure settings, flash, focal length, etc.) useful for the post-processing reconstruction. The main advantages are that all the process to obtain the RGB image can be performed off-line and the user can play with the data to obtain the desired picture. There are mainly two drawbacks: the first is that each company must

provide an appropriate reader to view and process the raw image. The second drawback is that the file size is usually huge.

11.2.3 JPEG2000

The compression standard JPEG2000 [23, 24] was born at the beginning of the third millennium with the aspiration of overcoming existing standards in terms of image quality. Moreover it has been created to provide features and functionalities that the current standards cannot address [25]. In particular, the JPEG2000 target was to match the requirements of a diversity of applications: Internet, printing, scanning, digital photography, mobile applications, E-commerce and so on. Unification of lossless and lossy compression mode, progressive transmission by pixel accuracy and by resolution, robustness to bit-errors, Region Of Interest (ROI) coding, are only few representative features provided by JPEG2000. Despite the large interests of such characteristics, the increase of the band capability in years following the standard publication and the high complexity of the JPEG2000 architecture reduced its possibility of application. For example, transmission at very low bit rate is no more a basic need. However, several features are object of interest in specific applications (e.g., digital cinema high resolution video [26]). This Section provides a brief overview of the JPEG2000 image compression standard.

JPEG2000 Main Features

One of the main goals of the new standard is achieving the best performances at low bit rates: JPEG2000 has been designed to offer performances superior to the current standards at bit-rates below 0.25 bpp. This feature was originally desirable for application that needs network image transmission. Also the blocking effect, one of the main defect of JPEG at high compression ratio, is reduced thanks to the so-called "tiling", that is the partitioning of the image in rectangular blocks of arbitrary size (up to the whole image size). The possibility of overlap adjacent tiles is also provided in order to reduce the blocking artifacts even if small tile size is defined.

Fig.(11.3) shows a comparison between JPEG and JPEG2000 coding at a bit-rate of 0.125 bpp. Note that information lost in the JPEG-coded image (Fig.(11.3(a))) are partially preserved in the JPEG2000 image (Fig.(11.3(b))), even if the global quality is low due to the high compression ratio. It's important to point out that the gap of the performances decreases as the bit-rate increase. One more basic feature of the JPEG2000 is the possibility of performing lossless coding, a useful opportunity for several application (e.g., medical images processing). Moreover, the Region of Interest (ROI) coding allows to code specific image zones at higher bit-rate. Such possibility makes possible, for example, preserving the central zone of the image when the "portrait" option is enabled in imaging devices. Strictly related with the ROI option is the random codestream accessing that allows to select and restore the ROI with higher fidelity respect to the rest of the image. Progressive transmission by pixel accuracy and resolution is another interesting

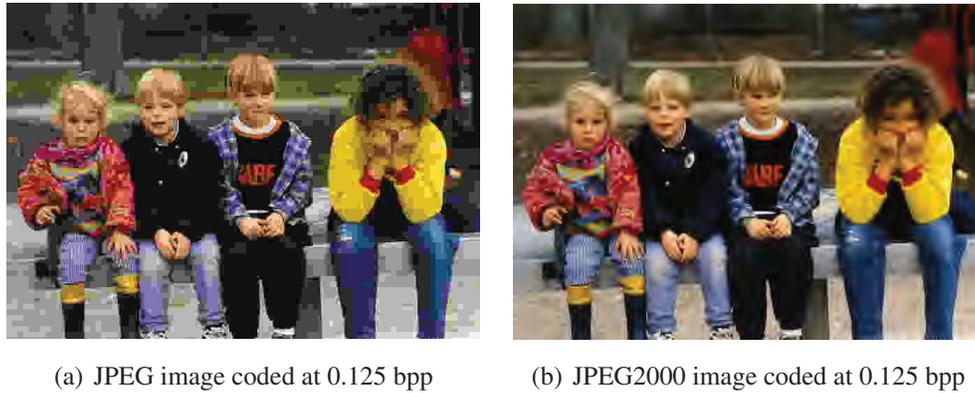


Figure 11.3 : Comparison between JPEG and JPEG2000 compression at very low bit-rate.

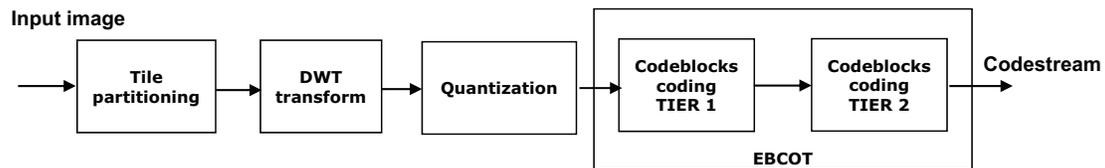


Figure 11.4 : JPEG2000 encoder scheme.

feature because allows restoring of images with different resolutions and pixel accuracy, as needed or desired, for different target devices.

JPEG2000 Coder and Decoder Basic Schemes

Fig.(11.4) shows the core encoding scheme of JPEG2000. The input image is firstly divided into tiles processed independently. For each tile the Discrete Wavelet Transform (DWT) is then performed and each subband is quantized and divided into frames and each frame is split in codeblocks. The compression is then performed by applying on each codeblock the entropy coding that consists in two steps Tier1 and Tier2 and it is performed by the EBCOT (Embedded Block Coding with Optimal Truncation) algorithm [27]. The EBCOT allows to obtain the bitstream resolution and SNR scalability, random access capability, because each codeblock is done independently. The decoding scheme is illustrated in Fig.(11.5) and it is basically the reverse process of the encoder.

11.3 Video Compression Formats

The video compression formats are based on reducing the spatial redundancies (as for the still images) and also the temporal redundancies. Temporal adjacent frames, in fact,

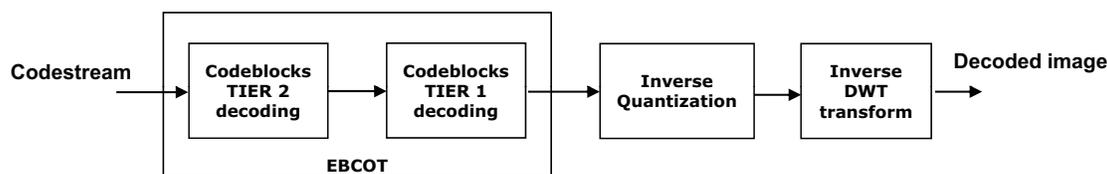


Figure 11.5 : JPEG2000 decoder scheme.

are usual almost similar, hence only the differences between them can be stored. As for the still images, a lot of standards have been published; in this Chapter will be shortly described the Motion JPEG (MJPEG), the H.263, the MPEG-4 and the H.264.

11.3.1 MJPEG

Motion JPEG is a video format where each frame of the sequence is compressed using the JPEG compression standard. The stream is usually encapsulated in Microsoft's AVI format or Apple's QuickTime MOV container. It is not an international standard (i.e., there is no official document containing its specification). This raises compatibility concerns among different implementations.

It was usually used for the video editing, but now it is also used by many portable devices. It is similar to the MPEG compression format using only I frames, avoiding P and B frames. Moreover the editing of the sequence does not affect the video quality (because there is no need to decode and re-encode the frames) and the encoder is very simple (because there is no need of motion estimation techniques, etc.), but the achieved compression ratio is low.

11.3.2 H.263

H.263 video standard [28] has been designed for low bit-rate communications, to replace H.261 [29] for video conferencing in most applications, even if the coding algorithms of the two standards are similar.

The H.263 standard, like H.261, is arranged in a hierarchical structure with four layers. From top to bottom the layers are:

1. **Picture**, which consists of a picture header followed by data for Group of Blocks, eventually followed by an end-of-sequence code and stuffing bits.
2. **Group of Blocks (GOB)**, which consists of a GOB header followed by data for macroblocks. Each GOB contains one or more rows of macroblocks. For the first GOB in each picture (with number 0), no GOB header shall be transmitted.
3. **Macroblock**, which consists of a macroblock header followed by data for blocks.

4. **Block**, which comprises four luminance blocks and one of each of the two color difference blocks (if not in PB-frames mode).

For further details please see the standard documentation [28].

Main Innovation of H.263

The differences between the H.261 and H.263 coding algorithms are:

1. **Half pixel precision**, used for motion compensation, whereas H.261 used full pixel precision.
2. **Optional parts of the hierarchical structure**, so the codec can be configured for a lower data-rate or better error recovery.
3. **New resolution support**. In addition to QCIF and CIF that were supported by H.261 there is SQCIF, 4CIF, and 16CIF.

Moreover, now four optional negotiable options have been included in H.263 to improve performance. They are:

1. **Unrestricted Motion Vector mode (Annex D)**. Motion vectors are allowed to point outside the picture. The edge pixels are used as prediction for the "not existing" pixels. With this mode a significant gain is achieved if there is movement across the edges of the picture, especially for the smaller picture formats.
2. **Syntax-based Arithmetic Coding mode (Annex E)**. Arithmetic coding is used instead of variable length coding. The SNR and reconstructed pictures will be the same, but significantly fewer bits will be produced.
3. **Advanced Prediction mode (Annex F)**. Overlapped block motion compensation (OBMC) is used for the luminance part of P-pictures. Four 8x8 vectors instead of one 16x16 vector are used for some of the macroblocks in the picture. Four vectors use more bits, but give better prediction. Especially a subjective gain is achieved because OBMC results in less blocking artifacts.
4. **Forward and Backward frame prediction or PB-frames mode (Annex G)**. A PB-frame consists of two pictures being coded as one unit. A PB-frame consists of one P-picture predicted from the previous decoded P-picture and one B-picture which is predicted from both the previous decoded P-picture and the P-picture currently being decoded. The name B-picture was chosen because parts of B-pictures may be bidirectional predicted from the past and future pictures. With this coding option, the picture rate can be increased considerably without increasing the bit-rate much.

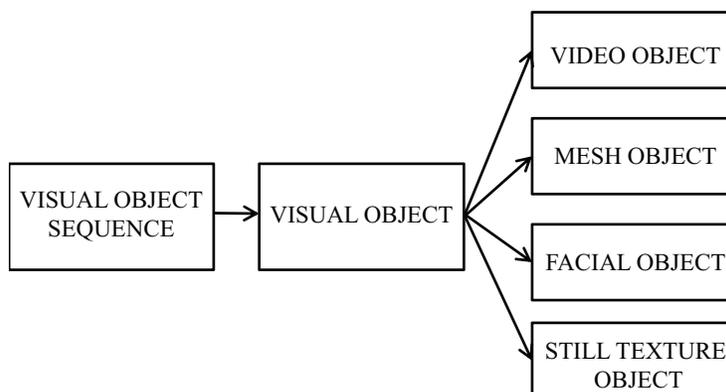


Figure 11.6 : Hierarchical structure of MPEG-4 standard.

11.3.3 MPEG-4

In MPEG-4 video standard [30], like H.263 [28], synthetic objects and their attribution are structured in a hierarchical manner, as indicated in Fig.(11.6). Visual Object Sequence (VOS) is the highest syntactic structure of the coded visual bitstream. A visual object sequence commences with a `VisualObjectSequenceStartCode` which is followed by one or more visual objects coded concurrently. The visual object sequence is terminated by a `VisualObjectSequenceEndCode`. A Visual Object (VO), the following structure in hierarchical order, commences with a `VisualObjectStartCode`, is followed by profile and level identification, a `VisualObjectID`, and finish with a video object, a mesh object, a face object or a still texture object. For video application, the video object is the most important, so we will focus on it. Its structure is reported in Fig.(11.7).

A video object starts with a `VideoObjectStartCode` and it is followed by one or more video object layers, which consists of one or more group of Video Object Plane (VOP). A VOP contains a single frame of an object. There are four types of VOPs that use different coding methods:

1. **Intra-coded (I)**, which is coded using information only from itself, so it can be reconstructed independently from other VOPs.
2. **Predictive-coded (P)**, which is a VOP coded using motion compensated prediction from a past reference VOP.
3. **Bidirectional predictive-coded (B)**, which is a VOP which is coded using motion compensated prediction from a past and/or future reference VOP(s).
4. **Sprite (S)**, which is a VOP for a sprite object, used for codification of not moving objects in the scene.

Data contained in a VOP are divided into macroblocks. A macroblock contains a section of the luminance component and the spatially corresponding chrominance com-

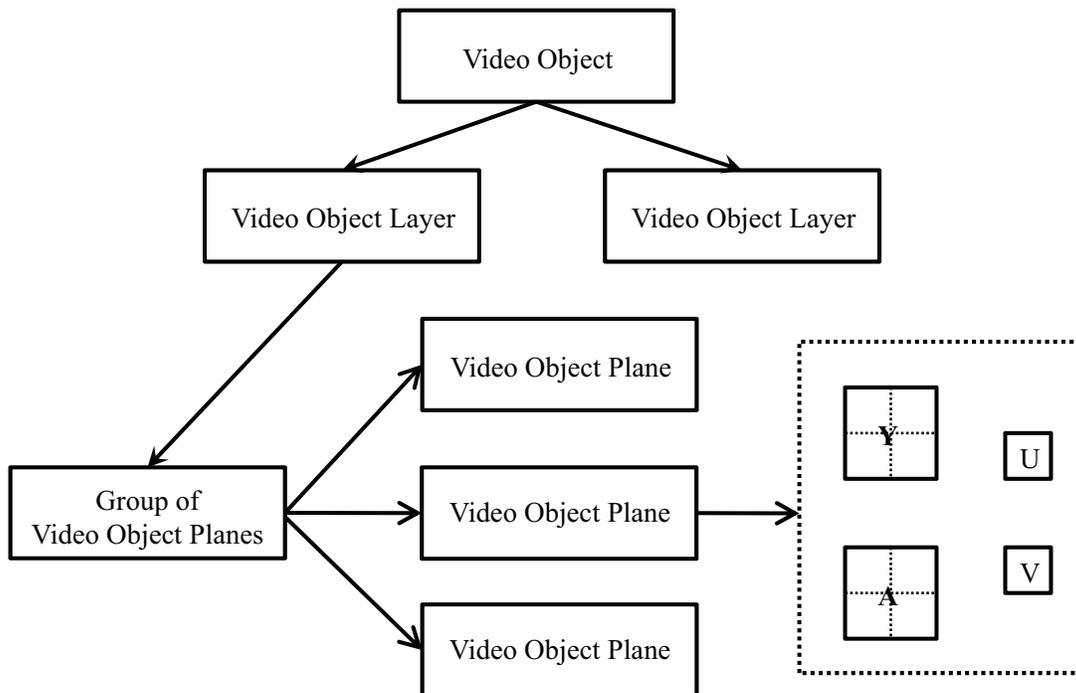


Figure 11.7 : Video Object hierarchical structure.

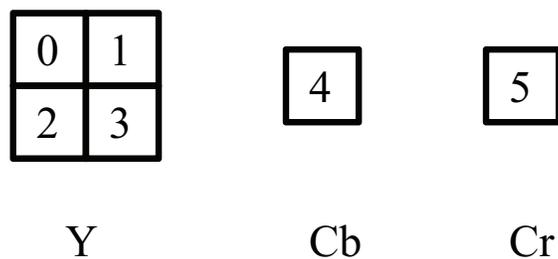


Figure 11.8 : 4:2:0 macroblock structure.

ponents. A skipped macroblock is one for which no information is transmitted. Presently there is only one chrominance format for a macroblock, namely, 4:2:0 format. A 4:2:0 macroblock consists of 6 blocks, having dimensions of 8x8 each. This structure holds 4 Y, 1 Cb and 1 Cr blocks and the block order is depicted in Fig.(11.8).

To provide forward compatibility with video codecs designed using the earlier video coding specification standard H.263 [28], the *ShortHeader* format is included in MPEG-4 video standard. More precisely, a *ShortHeader* bitstream is an H.263 baseline bitstream without any further annexes.

Main Innovation of MPEG4

In comparison with H263, MPEG4 adds a layer of error resilience. This is achieved through the following MPEG-4 codec capabilities:

1. **Resynchrhonization.** A resynchronization marker can reduce the error propagation caused by the nature of variable length code (VLC) into single frame. In MPEG-4, the resynchronization marker is inserted at the top of a new group of blocks GOB with the header information (multiplexed block number [MBN], quantization parameters and optional HEC), so that decoding can be done independently.
2. **Data partitioning.** A new synchronization code named motion marker separates the Motion Vector (MV) and Discrete Cosine Transform (DCT) field to prevent from inter-field error propagation, thus allowing effective error concealment to be performed. When errors are detected solely in the DCT field, that multiplexed block (MB) will be reconstructed using correct MV. This results in natural motion better than simple MB replacement of the previous frame.
3. **Reversible Variable Length Code (RVLC).** The RVLC enables forward and backward decoding without significant impact on coding efficiency. This feature localizes error propagation ideally into single MB.
4. **Adaptive Intra Refresh (AIR).** Different from the conventional cyclic intra refresh, AIR employs motion-weighted intra refresh, which results in better perceptual quality with quick recovery in corrupted objects.
5. **Error detection and concealment.** Errors can be detected through exception or violation in the decoding process, and then concealment will be applied. The functionality is included for mobile application.

11.3.4 H.264

H.264 standard, otherwise known as MPEG-4 Part 10 or AVC (Advanced Video Coding) [31], takes the MPEG-4 format to another level. It focuses on lossless methods and aims for flexibility on a variety of network types and application domains. H.264/AVC video standard has achieved a significant improvement in rate-distortion efficiency, providing approximately a 50% bit rate savings for equivalent perceptual quality relative to the performance of prior standards [32].

Main Innovation of H.264

H.264/AVC/MPEG-4 Part 10 contains a number of new features that allows it to improve coding efficiency. In particular, some of such key features include:

1. **Context-adaptive entropy coding.** Two entropy coding methods applied in H.264 to improve performance relative to prior standard designs:

- (a) **Context-adaptive binary arithmetic coding (CABAC)** is an algorithm to lossless compress syntax elements in the video stream knowing the probabilities of syntax elements in a given context. CABAC compresses data more efficiently than CAVLC but requires considerably more processing to decode.
 - (b) **Context-adaptive variable-length coding (CAVLC)** is a lower complexity alternative to CABAC for the coding of quantized transform coefficient values. Although lower complexity than CABAC, CAVLC is more elaborate and more efficient than the methods typically used to code coefficients in other prior designs.
2. **Short word-length transform.** All prior standard designs have effectively required encoders and decoders to use more complex processing for transform computation. While previous designs have generally required 32-bit processing, the H.264/AVC design requires only 16-bit arithmetic.
 3. **Exact-match inverse transform.** In previous video coding standards, the transform used for representing the video was generally specified only within an error tolerance bound, due to the impracticality of obtaining an exact match to the ideal specified inverse transform. As a result, each decoder design would produce slightly different decoded video, causing a "drift" between encoder and decoder representation of the video and reducing effective video quality. H.264/AVC is the first standard to achieve exact equality of decoded video content from all decoders.

In addition, some features have been designed to enhance coding efficiency:

1. **Multiple reference picture motion compensation.** Predictively coded pictures (called "P" pictures) in previous standards used only one previous picture to predict the values in an incoming picture. The new design extension allows an encoder to select, for motion compensation purposes, among a larger number of pictures that have been decoded and stored in the decoder (up to 16 reference frames).
2. **Variable block-size motion compensation with small block sizes.** This standard supports more flexibility in the selection of motion compensation block sizes and shapes than any previous standard. Supported luma prediction block sizes are: 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4. Chroma prediction block sizes are correspondingly smaller according to the chroma sub-sampling in use. The new H.264/AVC design is based primarily on a 4x4 transform. This allows the encoder to represent signals in a more locally-adaptive fashion, which reduces artifacts known as ringing.
3. **Quarter-sample-accurate motion compensation.** Most prior standards enable half-sample motion vector accuracy at most. The new design improves up on this by adding quarter-sample motion vector accuracy.
4. **Weighted prediction.** A new innovation in H.264/AVC allows the motion compensated prediction signal to be weighted and offset by amounts specified by the

encoder. This can dramatically improve coding efficiency for special cases, like fading in and out of a video.

5. **In-the-loop deblocking filtering.** Block-based video coding produces artifacts known as blocking artifacts. These can originate from both the prediction and residual difference coding stages of the decoding process. Application of an adaptive deblocking filter is a well-known method of improving the resulting video quality. The deblocking filter in the H.264/AVC design is brought within the motion-compensated prediction loop, so that this improvement in quality can be used in inter-picture prediction to improve the ability to predict other pictures as well.

11.4 Compression Factor Control

In this Section some techniques to control the compression factor control for still images and video formats will be described. These algorithms allow obtaining, with certain limits, the required bit rate. It is useful when a fixed file size is required or a maximum bandwidth is available for transmission. In particular, examples applied on Bayer data, on JPEG, on JPEG2000 and on MPEG-4 data streams will be presented.

11.4.1 Bayer Domain

Big efforts have been spent recently on developing algorithm devoted to solve the problem of compressing Bayer data at an appreciable bit rate with no visually loss in the output image. The most trivial, inexpensive solution (both in terms of computational complexity and hardware resources) is to split the Bayer image color channels and compress them independently using an efficient compression algorithm (i.e., Differential Pulse Code Modulation (DPCM) [20] or JPEG [33, 34] or JPEG-LS [35]). Although traditional coding techniques offer good performances on full color images, most of them do not offer the same performances with images captured by CFA digital sensors [36]. More sophisticated compression methods for images in Bayer pattern format are based on Wavelet transform [1, 37, 38] and properly modified JPEG [39]. Vector Quantization (VQ) [40] is a useful tool for achieve a good trade-off between image quality and compression rate on Bayer data. A VQ-based approach is the Predictive Vector Quantization (PVQ) [41] and it consists of two independent stages. Firstly, the input image in CFA format is compressed using a non-uniform vector quantizer, then a code describing each pair of adjacent pixels of the same color channel is built. The code generation step exploits the chromatic spatial correlation and several Human Visual System (HVS) heuristics to improve the compression efficiency. The resulting codestream is then further compressed applying a lossless DPCM algorithm. A similar approach is the Predictive Vector Coding (PVC) [42], that is based on the idea of applying the vector quantization on the prediction errors rather on the pixel pairs.

The following subsections describe some algorithms for Bayer data compression in more details.

Predictive Vector Quantization

Given a vector $[X_1, \dots, X_N]$ of size N , basic concept of Vector Quantization (VQ) [40] can be described geometrically. The associated binary representation can be seen as a set of N coordinates locating a unique point in the N -dimensional space. The quantization is performed partitioning the space with N -dimensional cells (e.g., hyperspheres or hypercubes) with no gaps and no overlaps. As the point defined by the input vector falls in one of these cells, the quantization process returns a single vector associated with the selected cell. Finally, such vector is mapped to a unique binary representation, which is the actual output of the vector quantizer. This binary representation (code) can have fixed or variable length. A vector quantizer is said to be "uniform" if the same quantization step is applied to each vector element, so that the N -dimensional space is divided into regular cells. If the space is partitioned into regions of different size, corresponding to different quantization steps, the quantizer is called "not-uniform". The "target" vector is called "codevector" and the set of all codevectors is the "codebook".

A uniform vector quantizer processes the input vector $[X_1, \dots, X_N]$ applying the same quantization step Q to each sample X_i (with $1 < i < N$) according to the following formula:

$$VQ(X_i) = \lfloor \frac{X_i}{Q} \rfloor \cdot Q, \quad i = 1, 2, \dots, N. \quad (11.7)$$

Resulting quantizer vector $[Y_1, \dots, Y_N]$ contains the reconstruction points for all samples of the N -dimensional space that fall in the range

$$Y_i < X_i < Y_i + Q, \quad i = 1, 2, \dots, N. \quad (11.8)$$

Given a 2-D image, it is described by a 2-dimensional vector of luminance values falling into the range $[0, \dots, 2^{bitDepth} - 1]$. Each pair of adjacent pixel is then mapped into a 2-D histogram partitioned in regular cells, which corresponds to the fixed quantization steps. The main drawback of uniform vector quantization approach is that all the regions of the image are quantized at the same factor independently from their information content. Since points along the diagonal of the histogram correspond to pixels of similar luminance values (homogeneous regions), while points that fall far from the diagonal were generated by pair of pixels of different luminance values (edge regions), a more sophisticated vector quantizer could be defined.

Exploiting simple properties of the HVS, non-uniform quantizer can be used to reduce the perceptual irrelevancy by quantizing more roughly where the quantization error is less visible. This is the basic idea of the Predictive Vector Quantization (PVQ) [41]. The visibility of the error depends on two masking effects. Since quantization errors are less visible along edges, PVQ performs a finer quantization near the diagonal of the histogram using smaller cells. Furthermore, high intensity luminance could be quantized more roughly, so PVQ performs finer quantization near the bottom left corner of the histogram, as showed in Fig.(11.9). A Bayer pattern image of size $N \times N$ contains $N^2/2$ green pixels, $N^2/4$ red pixels and $N^2/4$ blue pixels, so a scheme to build pairs of pixels of the same component should be chosen (Fig.(11.10)).

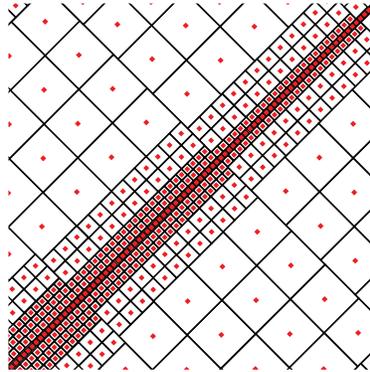


Figure 11.9 : Non-uniform partitioning of the 2-D histogram defining an adaptive quantizer.

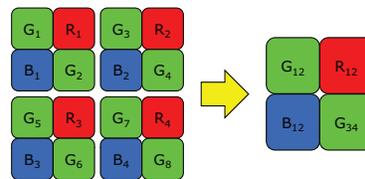
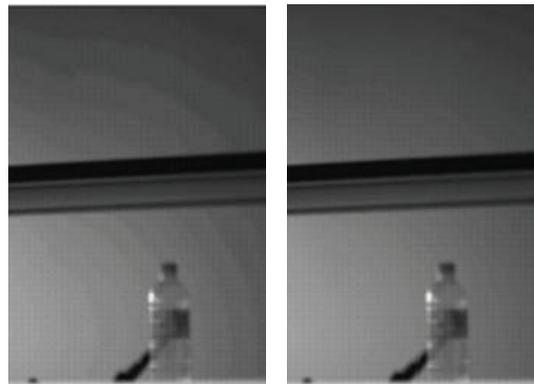


Figure 11.10 : Pixel pairs construction scheme.



(a) Uniform VQ (b) not uniform VQ

Figure 11.11 : Comparison between uniform and not uniform Vector Quantizers.

Moreover, a code is assigned to each reconstruction point corresponding to a pair $\langle C_i, C_j \rangle$ of pixels in a color component, applying an appropriate function f :

$$C_{ij} = f(\langle C_i, C_j \rangle) \tag{11.9}$$

Assuming that the Bayer image to be coded has a bit depth of eight bit, the not uniform vector quantizer must be defined by 512 reconstruction points, so a 9-bits code is assigned to each pair of pixels in the original image and a bitrate of 4,5 bit per pixel (bpp)



(a) RGB from uncompressed CFA (b) RGB from co-decoded CFA

Figure 11.12 : Comparison between full color images obtained from the same Bayer with and without compression.

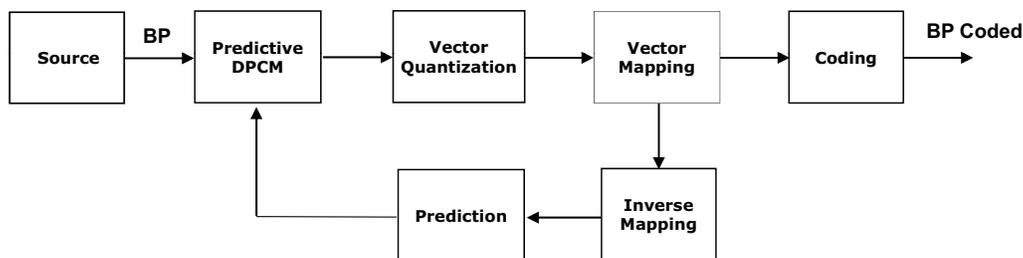


Figure 11.13 : PVC compression basic scheme.

is achieved. Fig.(11.11) shows a comparison between the compressed images obtained using uniform VQ (Fig.(11.11(a))) and not-uniform vector-quantizer (Fig.(11.11(b))). In the first case the quantization errors are evident in the homogeneous regions of the image. Fig.(11.12) shows an example of PVQ compression performances. The image on the left has been obtained performing the color interpolation starting from the original Bayer pattern, the image on the right has been obtained processing the same CFA data after the PVQ compression/decompression procedure. Note that the full color images are comparable in terms of visual quality.

Predictive Vector Coding

One more compression algorithm based on VQ and suitable for Bayer pattern images is the Predictive Vector Coding (PVC) [42], whose basic scheme is showed in Fig.(11.13). It consists of four basic steps:

1. **Prediction:** application of the DPCM algorithm to compute the difference between

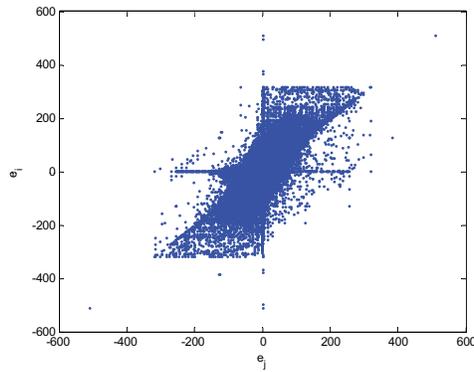


Figure 11.14 : Typical DPCM error distribution.

the actual and the predicted pixel values gathered in vectors of two consecutive elements of the same color channel.

2. **Vector Mapping:** reduction of the symmetry 2-D maps of the prediction errors distribution.
3. **Vector Quantization:** VQ lossy compression of the data.
4. **Code Generation:** generation of an n bit code for each couple of pixels. In the following discussion the input is assumed to be a 10 bpp Bayer data and the generated code is 12 bits length for a final bit rate of 6 bpp.

The DPCM step is based on the idea of reducing the entropy of the source by coding the difference between the current value and a prediction of the value itself. The PVC assumes that adjacent samples of the same color components have similar brightness values. The prediction function is performed with 2-dimensional vectors obtained applying VQ. In particular, let (V_i, V_j) be the vector to be coded. The second value of the previous vector (V_{i-1}, V_{j-1}) is used to build a vector of two identical components (V_{j-1}, V_{j-1}) which is the predictor for (V_i, V_j) . This strategy has been chosen because V_{j-1} is spatially closer to (V_i, V_j) than V_{i-1} and usually closer samples are statistically more correlated. The error vectors (e_i, e_j) are computed as the difference between the vector to be coded and the prediction vector:

$$(e_i, e_j) = (V_i, V_j) - (V_{j-1}, V_{j-1}) = (V_i - V_{j-1}, V_j - V_{j-1}). \quad (11.10)$$

A typical error distribution for this kind of prediction scheme is showed in Fig.(11.14).

Note that such distribution has been obtained collecting statistics on a large dataset of Bayer images. The VQ should be designed to optimize such distribution preserving the samples that fall near the origin.

Moreover, observe that in the prediction error distribution is evident an odd symmetry that can be exploited to reduce the size of the table used for the VQ because the vectors



Figure 11.15 : Outer Quantization regions in PVC.

falling in the third and in the fourth quadrant can be mapped in the first and in the second one, so just the two upper quadrants have to be quantized. This task is performed by the "Vector Mapping" stage of the algorithm: it checks whether the input vector falls in the upper part of the diagram or not. In the first case no changes occur, while, in the other case, the sign of the values is changed. One bit is used in the final code in order to take into account such mapping.

A grayscale 10 bit image is described by 2-dimensional vectors of brightness values falling into the range $[0, \dots, 1023]$. In the PVC model, each codevector represents a 2-dimensional input vector. The algorithm uses a not uniform VQ exploiting two HSV properties: the quantization errors are less visible along the edges and the eyes discriminates better the details at low luminance levels. Thus, in the areas near the origin (where the prediction error is low) a fine quantization is performed and most information is preserved. On the contrary, in the area far from the origin, a coarse quantization is applied due to the presence of boundaries and more information is loss. Furthermore, since DPCM drifts the values towards zero, a very high percentile of input samples will fall in the area around zero.

The PVC vector quantizer divides the two upper quadrants of the 2-dimensional quantization space into 32 "macro-regions" shaped and distributed to minimize the quantization error. Each region has different size and position in the quantization board and it has been divided into 64 "sub-regions". Such regions have been obtained dividing the horizontal and vertical dimension by a constant number. In this way, bigger regions cover bigger areas and the quantization is stronger (more loss of information), while in smaller areas a lighter quantization is applied and most information is preserved (Fig.(11.15)).

Each vector (e_1, e_2) to be quantized is approximated with the nearest couple in the corresponding sub-region.

The final step is the code generation that defines the final Bayer compression format. The code representing the vector quantized samples is a fixed length code. It summarizes information about the vector mapping, the region where the point falls and the quantization steps applied in each region. The first bit is the "Vector Mapping" bit, indicating if the swap between upper and bottom quadrants has happened or not. Next bits following indicate the index of the region in the quantization table. The length of this part of the code depends on the number of regions in the quantization table. The remaining bits give

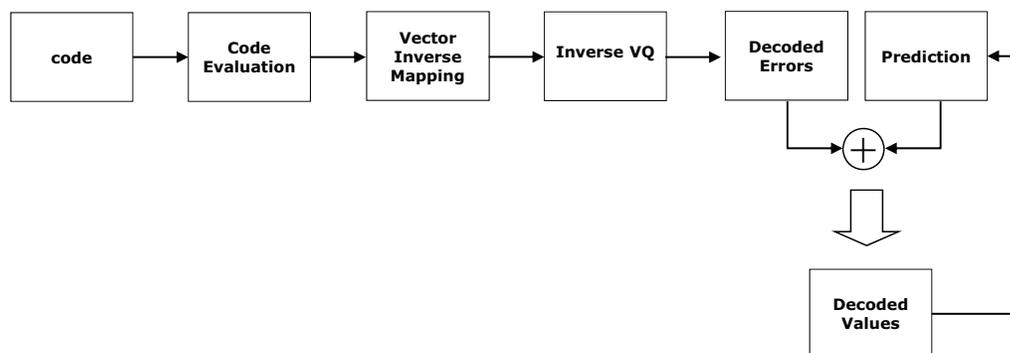


Figure 11.16 : Decoding scheme.

information on the number of steps, in both vertical and horizontal direction inside the region. In this discussion we assumed that a 12-bits code should be generated, in order to represent samples falling in a space partitioned into 32 regions and 64 "sub-regions". Thus, the code reserves five bits to index 32 regions and 6 bits to index one of the 64 sub-regions. Different code structure could be defined if the space partitioning or the target bit-rate changes.

The PVC decoding procedure consists of three main steps. The first one is the code evaluation allowing the extraction of the compressed values. The second one is the "Inverse Vector Mapping". It assigns the right sign to values depending on the inversion flag. Then the retrieving of the original values is obtained adding the decoded prediction error to the previously restored vector. Since the predictor is equal both in the encoder and the decoder, the predicted values are equal in the two parts of the processes, so there is no propagation of error during the decoding process. The decoder block diagram is shown in Fig.(11.16).

11.4.2 JPEG

The JPEG uses variable size codes. It means that, given a set of coding parameters, the file size depends on the image content. It could be a problem since it does not allow to know a priori the number of photo that can fit in a fixed size memory storage or the required bandwidth to transmit it. So a lot of researches focused on the compression factor control. Taking a look to the Fig.(11.1), the Huffman tables and the Quantization tables can be used to modify the compression ratio. Optimizing the Huffman tables [19], the file size decreases. Unfortunately the optimization is expensive because the image must be scanned twice collecting all the pairs $(run, value)$. Moreover the compression ratio is fixed and the improvement is not so high since it is basically a lossless compression. The most used technique to control the compression factor is hence modifying the quantization tables. Increasing each quantization value correspond to increase the number of zeros, hence reducing the number of pairs $(run, value)$. Each quantization value can be modified

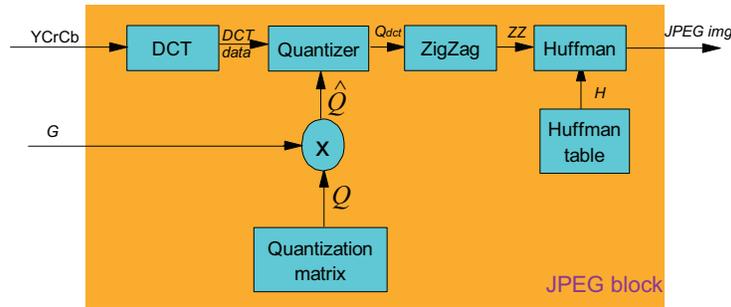


Figure 11.17 : JPEG - Block based schema to control the compression factor control.

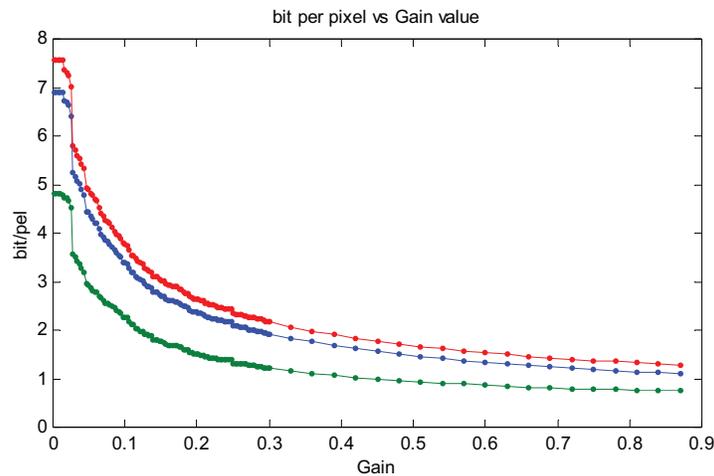


Figure 11.18 : Bit rate vs Gain for three different images.

independently, but usually the strategy is to modify a fixed quantization table using a single multiplier:

$$\hat{Q} = Gain \cdot Q \quad (11.11)$$

where \hat{Q} is the modified quantization table, $Gain$ is a scalar value and Q is the original quantization table. The compression factor control algorithm aims to find the gain factor $Gain$ to obtain a file size as close as possible to a target value. The modified block based schema is shown in the Fig.(11.17).

In Fig.(11.18) is depicted the bit-rate obtained varying in increasing way the gain for three different images. It shows that the bit rate depends on the image content (since different content generates a different number of pairs $(run, value)$); the bit rate has discontinuities (especially when the gain causes a lot of quantization coefficients to vary); the curve is monotonically non-increasing (by increasing the gain, the number of pairs cannot increase).

The compression factor control algorithms can be evaluated through four different

aspects:

- precision;
- image quality;
- speed;
- resources (computational cost, power consumption, etc.).

The ideal algorithm cycles indefinitely the gain factor choosing the value providing the lowest error from the target. Of course it is very expensive algorithm (in terms of resources), the computational time is unpredictable and it is very slow. Less expensive algorithms can be divided in two types:

1. Constant precision;
2. Constant cycles number.

In the Constant precision algorithms the cycle is repeated until the bit/rate fits the prefixed precision range. These algorithms have high precision, but the computational time and the power consumption are not predictable. In the constant cycles number algorithms the cycle is computed n -times ($n > 0$). The precision is not predictable but the computational time and the power consumption are predictable and they are used for real time implementation. The goodness of an algorithm is to obtain the best precision in few cycles. The pseudo-code of a constant precision algorithm is the following:

```

Init (threshold, G, target);
repeat
    bitrate = JPEGcompress(img, G);
    G = modifyG(bitrate, target, G);
until ( $abs(bitrate-target) \leq threshold$ ) ;

```

Where:

img is the image;

G is the current gain value;

target is the target bit rate;

threshold is the required precision;

Init() set the initial values;

JPEGcompress() is a function that compress the image and return the bit count;

modifyG() is the core of the rate control and modify the Gain value according to the current results.

The pseudo-code of a constant cycles number algorithm is the following:

Some algorithms are available in [17, 18, 43].

```

Init (ncycles, G, target);
for  $i=1$  to ncycles do
    bitrate = JPEGcompress(img, G);
    G = modifyG(bitrate, target, G);
end

```

11.4.3 JPEG2000

One of the most impressive features of the standard JPEG2000 [23] is the possibility of provide a bit-stream that is both resolution scalable (if DWT is adopted) and quality scalable (if multiple layers are provided). Such feature has been developed to make the standard suitable for any application where bandwidth constraints require maximum limited data transmission and/or a limited display resolution is available [44]. The scalable bit-stream is built thanks to the JPEG2000 core-coding engine based on the 2-tiers paradigm of the EBCOT algorithm [27, 45].

The first tier provides a low level coding of the block samples belonging to the various image sub-bands by means of a context adaptive BAC (Binary Arithmetic Coding) referred as MQ coder [34]. Then all the blocks are processed again to produce a full packed bit-stream. They are subparts of code-blocks bit-stream that are spread across different quality layers with each quality layer increasingly improving the final image. The optimal block truncation is decided within Rate-Distortion (RD) optimization algorithms [46] to produce layers of fixed bit rate or distortion. RD problems can be easily solved relying on the well-known Lagrangian multiplier [47] method. For example, if a target rate R_{max} is desired for a given layer the following problem must be solved:

$$\text{minimize} \quad \sum_{i=1}^{N-1} D_i^{n_i} \quad \text{such that} \quad R = \sum_{i=1}^{N-1} R_i^{n_i} \leq R_{max} \quad (11.12)$$

where N is the number of blocks, n_i is the chosen truncation point for the i -th block, $D_i^{n_i}$ and $R_i^{n_i}$ are, respectively, distortion and rate for a block at a given truncation point. The candidates truncations points are $3 \cdot K - 1$ (where K is the bit depth) and are produced by the MQ-coder in three different coding steps. Solving the (11.12) corresponds to solving the following:

$$\text{minimize} \quad D(\lambda) + \lambda \cdot R(\lambda) = \sum_{i=1}^{N-1} D_i^{n_i} + \lambda \cdot R_i^{n_i} \quad (11.13)$$

for a $\lambda > 0$ found through bisection to achieve the target rate. This rate-control mechanism, described in the standard in full details [23], is known in literature as PCRD (Post Compression Rate Distortion Optimization) and its role is to find the optimal λ_{opt} which can make $\sum_{i=1}^{N-1} D_i^{n_i}$ minimize subject (ideally) to the constraint $\sum_{i=1}^{N-1} R_i^{n_i} = R_{max}$. Such approach, as most of the algorithms used for compression factor control, has the main purpose of assuring a fixed file size. This constraint affects heavily the choice of the trade-off between compression ratio and image quality and poses some practical restrictions both in terms of memory allocation and overall computation. Several techniques

able to improve the visual quality of the image for a fixed bit rate have been proposed to allow better performances of the compression standard JPEG [48–52]. In the same way, some basic property of the Human Visual System could be exploited to code fewer bits to represent perceptual less important areas of the image coded using the JPEG2000 [44].

11.4.4 MPEG4

In the MPEG-4 standard the compression factor control is obtained modifying a quantization parameter. It can be different for each macro-block, hence the precision can be very high. Video sequences are subjected to constraints in terms of bandwidth (i.e., bit/s) when they are transmitted for real time applications (e.g., video conferences, etc.) or to fit globally in the storage memory (e.g., video recorder, video messaging, etc.). In the first case the constraint must be satisfied at frame level regardless the picture quality. In the second case the constraint is at sequence level, hence the bits usage can differ for each frame depending on the image complexity. Algorithms developed for the first case are called Constant Bit Rate (CBR), while algorithms belonging to the second case are called Variable Bit Rate (VBR). The main features of the CBR algorithms are:

- the Quantization Parameter (QP) is modified at macroblock level;
- a very hard rate control is performed;
- a variable quality picture is obtained.

The main features of the VBR algorithms are:

- the QP is modified at frame level;
- a constant quality picture is obtained;
- the quality between adjacent frames is similar;
- the rate is not ensured at frame level.

In the following we report the Scalable Rate Control (SRC) algorithm as proposed by the standard (described in the Annex L of the standard [30]). The main features can be resumed as follows:

- it can be used for both VBR and CBR;
- it is scalable for various bit rates (e.g., 10kbps to 1Mbps);
- it is scalable for various spatial resolutions (e.g., QCIF to CIF);
- it is scalable for various temporal resolutions (e.g., 7.5 fps to 30 fps);
- it is scalable for various coders (e.g., DCT and wavelet);
- it can handle I, P, and B pictures.

The SRC scheme assumes that the encoder rate distortion function can be modeled as:

$$R = \frac{X1 \cdot S}{Q} + \frac{X2 \cdot S}{Q^2} \quad (11.14)$$

where R is the target of bits for the frame; S is a complexity measure of the frame; Q is the quantization parameter of the frame; $X1$ and $X2$ are the modeling parameters. The algorithm works in two steps:

1. Before the frame encoding:
 - the target bit rate (R) is assigned to the frame based on the bits needed for the previous frame and on the content of the current frame;
 - the Q value is retrieved using the rate distortion model curve.
2. After the frame encoding:
 - the modeling parameters ($X1$, $X2$) are updated;
 - skip the next frames if needed.

Another concept to be kept in consideration is the so called Video Buffer Verifier (VBV). For real time applications, where the device has a fixed size buffer, it is important to avoid buffer overflow, otherwise data will be lost. The VBV is a theoretical video buffer model introduced in the standard to avoid overflow and underflow. A standard compliant video stream will never overflow nor underflow. The encoder must control that the VBV requirements are fitted. The analysis of the VBV is out of the scope of this book. More information can be read in the Annex D of the standard [30].

An example in pseudo-code of the SRC used as VBR is the following:

```

Init (ncycles, G, target);
for  $i=1$  to  $ncycles$  do
  S(t) = Complexity (CurrentFrame);
  R(t) = ComputeR (R(t-1), S(t));
  Q(t) = ComputeQ (R(t), S(t), X1, X2);
  T(t) = EncodeFrame (Q(t));
  UpdateX (&X1, &X2, T(t));
  SkipFrame (T(t), VBvinfos);
end

```

where $T(t)$ is the number of bits spent to encode the frame and $VBvinfos$ controls the fullness of the Video Buffer.

11.5 Error Detection and Concealment

The error detection and error concealment techniques are becoming more and more important thanks to the growing interest in still picture and video delivery over wireless

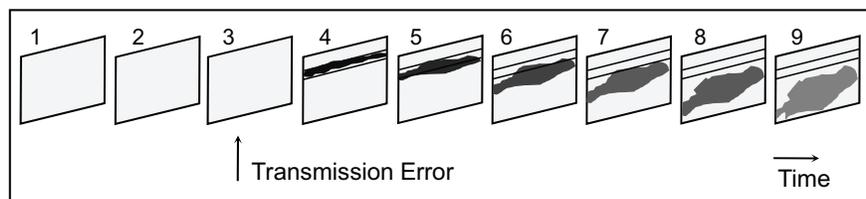


Figure 11.19 : Propagation errors effect over time.

channels, in particular for mobile applications. The main problem for such type of applications and in general for any standard video compression, is that the transmitted bit-stream is very sensitive to the channel noise due to the presence of the Variable Length Codes (briefly VLCs). In fact, when the decoder detects an error in a VLC, it doesn't know where the next VLC starts and then it can only look, if possible, at the next resynchronization point or, in the worst case, at the end of the frame, losing a lot of information. Moreover, in video decoding, due to the presence of the motion compensation algorithms, a single bit error during the transmission produces a noticeable degradation affecting not only the current frame, but also the subsequent frames. These particular kinds of problems are known as "propagation errors" as showed in Fig.(11.19).

Error Detection

The error detection techniques consist in a set of testing procedure needed to avoid the decoding of corrupted data. These checks are a basic feature of the decoder because they make it able to prevent decoding failure improving at the same time the overall video quality. Generally, in the video or still picture standard the following tests are executed and verified:

1. An illegal VLC is received.
2. A semantic error is detected:
 - (a) More than 64 DCT coefficients are decoded in a block (if IDCT is present).
 - (b) Inconsistent resynchronization header information (if resynchronization markers are present).

Error Concealment

The error concealment techniques are applied when an error is detected in order to replace the corrupted information with some other information that can be represented in a more graceful way ("graceful degradation"). The decoder must decide which part of the frame is assumed to be error free and which part will be concealed. The error concealment methods described here can be divided into three main groups [53]:



Figure 11.20 : Video Packet composition in MPEG-4.

1. **Spatial concealment.** It consists in estimating missing pixels by smoothly extrapolating surrounding pixels (for example using bilinear interpolation from the neighborhood pixels). In general, spatial concealment methods are the most complex, since a computation must be done for each pixel and correctly recovering missing pixels is extremely difficult.
2. **Frequency concealment.** In this case, some low-order DCT coefficients of the missing blocks are estimated using either the corresponding DCT coefficient of neighboring blocks, or the neighbor's DC values.
3. **Temporal concealment.** Usually two methods are used:
 - (a) **Temporal interpolation.** It consists in copying the pixels at the same spatial location in the previous frame (freeze frame). It is effective when there is no motion, but there are potential problems in the presence of intensive motion.
 - (b) **Motion compensated temporal interpolation.** It usually uses motion vector to estimate missing block as motion-compensated block from prior frame. It can also use an averaging of the surrounding motion vectors (if present) or it can be computed with some other heuristics.

11.5.1 MPEG4

Some error detection and concealment improvements for the video standard MPEG-4 have been proposed in [54]. In particular, we put into evidence the techniques called *macroblock number checking* and *improved video packet concealment*.

Error Detection

The *macroblock number checking* method performs a smart check to verify the consistency between the current Video Packet (briefly VP) data and the following one. The VP approach is aimed to provide periodic resynchronization markers to the bitstream. This allows the decoder resynchronization, when an error has been detected. If the decoder finds a correct VLC longer than the real one inside the VP, besides quality degradation there is the risk that the decoder can read subsequent data and then mix information of two different VPs. This could happen, of course, without detecting any errors. In particular, in MPEG-4 the composition of a single VP is indicated in Fig.(11.20).

The Resync Marker is followed by this information, in which the first three elements (from *macroblock_number* to HEC) represent the VP Header:



Figure 11.21 : Frame obtained with MPEG-4 standard checks, when the decoder read a VLC longer than the real one.

1. **macroblock_number**: the number of the first Macroblock (briefly MB) contained in this VP.
2. **quant_scale**: the quantization parameter.
3. **Header Extension Code (HEC)**: additional information.
4. **Macroblock Data**: video information.

When a VLC longer than the real one is found inside the Macroblock Data, it is easy to understand that probably the data mixing of two subsequent VPs has happened and this will degrade the frame quality (see Fig.(11.21)).

To avoid this kind of artifacts, it is important to know in advance, before entirely decoding the current VP, the position of the next resync marker. In particular, for MPEG-4, examining the composition of the VP (see Fig.(11.20)), it is noticeable that, to know how many MBs are contained in the VP, the decoder needs to decode all the VP. To know this information in advance, the decoder should check that the number of MBs, read in the Macroblock Data in the current VP, is consistent with the MB number read in the next VP header. In this case the decoder can detect this kind of error and it can conceal the whole VP.

More exactly the sequence of the actions is:

1. Read the VP Header in the current VP, obtaining the macroblock_number information (*current_mb_number*).
2. Jump to the next VP, obtaining the macroblock_number information (*next_mb_number*).
3. Continue decoding current VP, obtaining the number of macroblocks contained in the current VP (*mb_in_VP*).



Figure 11.22 : Frame obtained using MB numbers checking, with motion compensated MBs from last frame.

4. If *current_mb_number* plus *mb_in_VP* differs from *next_mb_number*, conceal the current VP.

In this case the correspondent frame is shown in Fig.(11.22).

Error Concealment

The Annex E algorithm [30] describes the behavior of the decoder error concealment treatment, making use of the following notations:

1. *L*: total number of bits for DCT coefficients part in a VP.
2. *N*: total number of macroblocks (MBs) in a VP.
3. *L1*: number of bits which can be decoded in a forward decoding.
4. *L2*: number of bits which can be decoded in a backward decoding.
5. *N1*: number of MBs which can be completely decoded in a forward decoding.
6. *N2*: number of MBs which can be completely decoded in a backward decoding.
7. *f_mb(S)*: number of decoded MBs when *S* bits can be decoded in a forward direction (equal to or more than one bit can be decoded in a MB, *f_mb(S)* counter is up).
8. *b_mb(S)*: number of decoded MBs when *S* bits can be decoded in a backward direction.
9. *T*: threshold (90 is used now).

where:

$$0 \leq N1 \leq (N - 1) \quad (11.15)$$

$$0 \leq N2 \leq (N - 1) \quad (11.16)$$

Depending on the value of $(L1+L2)$ and $(N1+N2)$, we can distinguish the following four different strategies. More exactly, *Strategy 1* is activated when:

$$((L1 + L2) < L) \quad \text{and} \quad ((N1 + N2) < N) \quad (11.17)$$

Strategy 2 when:

$$((L1 + L2) < L) \quad \text{and} \quad ((N1 + N2) \geq N) \quad (11.18)$$

Strategy 3 when:

$$((L1 + L2) \geq L) \quad \text{and} \quad ((N1 + N2) < N) \quad (11.19)$$

Strategy 4 when:

$$((L1 + L2) \geq L) \quad \text{and} \quad ((N1 + N2) \geq N) \quad (11.20)$$

Depending on the activated Strategy, the following actions will be taken:

1. *Strategy 1*: The first $f_mb(L1-T)$ MBs are decoded in forward direction, MBs of the dark part are concealed and the last $b_mb(L2-T)$ MBs are decoded in backward direction, as indicated in Fig.(11.23).
2. *Strategy 2*: The first $(N-N2-1)$ MBs are decoded in forward direction, MBs of the dark part are concealed and the last $(N-N1-1)$ MBs are decoded in backward direction, as indicated in Fig.(11.24).
3. *Strategy 3*: The first $(N-b_mb(L2))$ MBs are decoded in forward direction, MBs of the dark part are concealed and the last $(N-f_mb(L1))$ MBs are decoded in backward direction, as indicated in Fig.(11.25).
4. *Strategy 4*: The first $\min(N-b_mb(L2), N-N2-1)$ MBs are decoded in forward direction, MBs of the dark part are concealed and the last $\min(N-f_mb(L1), N-N1-1)$ MBs are decoded in backward direction, as indicated in Fig.(11.26).

An improvement technique called *improved video packet concealment* [54], can be used to improve the above schemes. The main idea of this technique is indicated in Fig.(11.27). For each Video Packet containing errors, it is possible to determinate a priori which part of the VP is assumed to be without errors, so just decoded in forward or backward direction (VP concealment limitation, identified by the gray zone) and which part will be concealed (VP concealment extension, identified by the dark zone). Afterwards, the application of Annex E shall determinate how to handle the uncertain part (white zone). Numerical examples of this technique are reported in [54]. Please refer to it for further details.

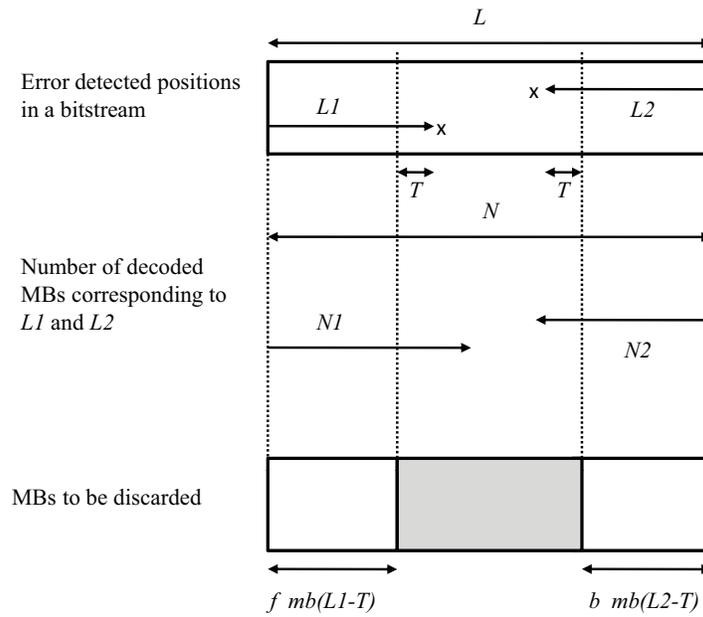


Figure 11.23 : MPEG-4 standard Annex E Strategy 1.

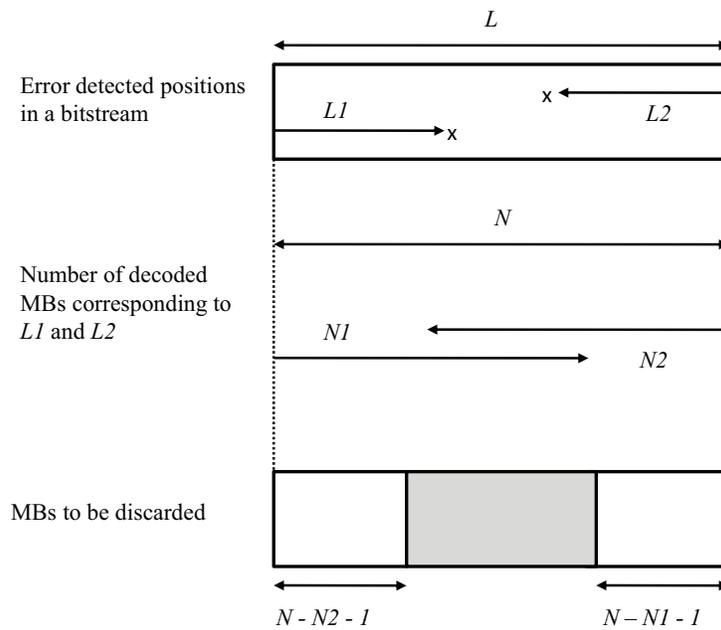


Figure 11.24 : MPEG-4 standard Annex E Strategy 2.

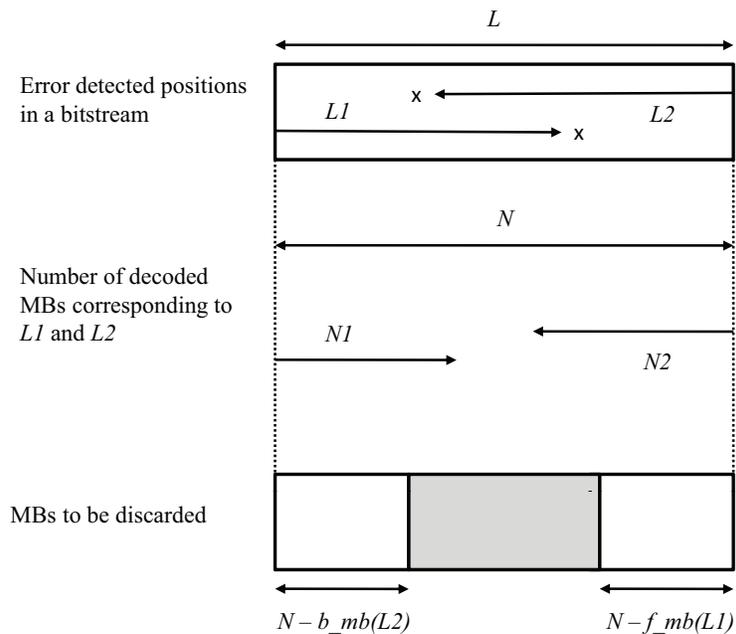


Figure 11.25 : MPEG-4 standard Annex E Strategy 3.

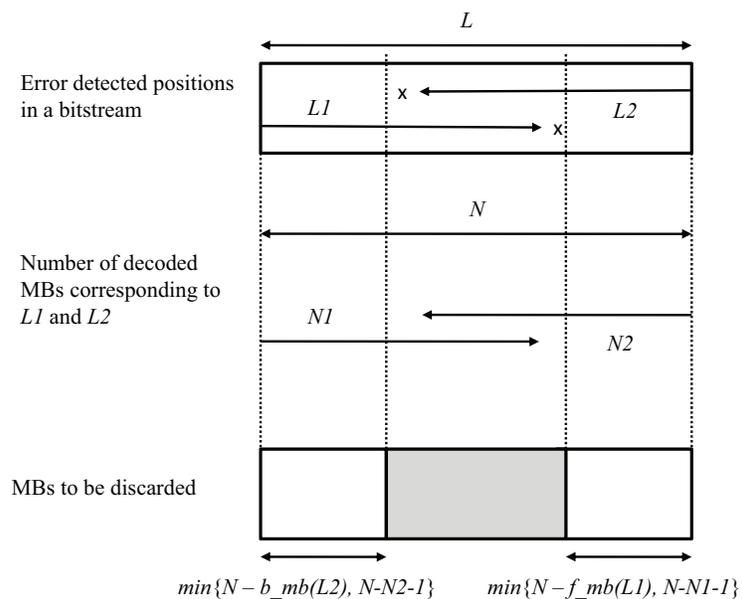


Figure 11.26 : MPEG-4 standard Annex E Strategy 4.



Figure 11.27 : Video Packet concealment limitation and extension (X = points in which errors were detected).



Figure 11.28 : Left: Image decoded with JPEG (compression 50%). Right: Image with error decoded with JPEG standard.

11.5.2 JPEG

While in MPEG-4 standard a lot of error resilience techniques can be used, as indicated in previous sections, JPEG standard does not have many possibilities to manage errors. When an error is encountered in JPEG, we have to skip picture information up to the end of the picture itself, with a tremendous loss of information (Fig.(11.28)).

The only error resilience technique allowed in JPEG is the resynchronization system. Similar consideration made for MPEG-4 (about resync marker) can be extended to this standard.

In JPEG standard, the restart markers (RST) are added to the compressed data between each restart interval. They have a two bytes value which cannot be generated by the coding procedures, to be used to resynchronize the decoder. There are 8 unique restart markers ($m = 0 - 7$) which repeat in sequence from 0 to 7, starting with zero for each scan, to provide a modulo 8 restart interval count.

By comparing the expected restart interval number to the value in the next RST m marker in the compressed image data, the decoder can usually recover synchronization. It then fills in missing lines in the output data by replication or some other suitable procedure, and continues decoding. Of course, the reconstructed image will usually be highly corrupted for at least a part of the restart interval where the error occurred.

Since JPEG is a Variable Length Code (VLC) based standard, when a VLC is longer than the real one is found inside the compressed data between each restart interval, it is easy to understand that probably the data mixing of two subsequent restart interval has happened and this degrade the frame quality. The technique called *Macroblock Number Checking*, used for MPEG-4, can be extended also for JPEG standard, as follows [55]:

1. Read the first RST m .

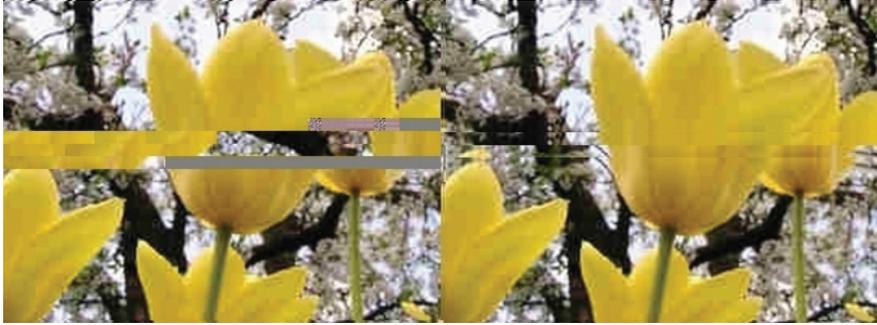


Figure 11.29 : Left: Image with error decoded with JPEG standard with resync markers. Right: Image with error decoded with using Macroblock Number Checking algorithm.

2. Jump to the next RST_n .
3. If n differs from $m+1$, conceal the compressed data between RST_m and RST_n .

An example of the application of such algorithm is shown in Fig.(11.29). In this example a resync marker is used every two lines of decoding macroblocks. While in the JPEG standard the mixing of two subsequent restart intervals is evident, with the application of *Macroblock Number Checking*, two lines of uncorrected decoded macroblocks (the ones included in the two subsequent restart interval) are identified and concealed re-copying the previous and sub-sequent correctly decoded row of macroblocks.

11.6 The JPEG XR Image Coding Standard

A new image coding standard, JPEG XR ((ITU-T T.832 — ISO/IEC 29199-2) has been recently released from JPEG committee; differently than JPEG 2000 it seems to be an effective alternative to the classic JPEG (see Chapter 11). It is able to manage high dynamic range (HDR) imagery applications because it supports natively such format. Some of the advanced peculiarities, already present in JPEG 2000 but never used, such as scalability (quality, bit-rate), ROI coding are available; also a unified system for both lossless and lossy coding is present.

The performances with respect to its predecessors JPEG and JPEG 2000 are fully comparable; for very low bit-rate JPEG-XR is very close in terms of quality to the JPEG2000 but on the other hand its overall complexity is very low. The JPEG committee began the standardization of JPEG XR technology in 2007. The initial design proposal was submitted by Microsoft, based on its HD photo technology, while the subsequent development and future evolution of JPEG XR as a standard has been the responsibility of the JPEG committee.

The core of the encoding engine is the spatial transform that converts the image data to a frequency domain representation. A lifting-based reversible hierarchical lapped biorthogonal transform (LBT) is used. The transform requires only a small number of

integer processing operations for both encoding and decoding. It is exactly invertible in integer arithmetic and hence supports lossless image representation. The transform is based on two basic operators: the core transform and the optional overlap filtering. The core transform is similar to the DCT and can exploit spatial correlation within a block-shaped region. See [56] for more details.

Of course the success of any standard is driven by a wide range of factors (not only related to the technological aspects) but the new standard could have a chance especially for applications requiring more than usual 24 RGB representation.

Bibliography

- [1] T. Toi and M. Ohta, "A subband coding technique for image compression in single CCD cameras with Bayer color filter arrays," *IEEE Transactions on Consumer Electronics*, vol. 45, pp. 176–180, February 1999.
- [2] C. C. Koh, J. Mukherjee, and S. K. Mitra, "New efficient methods of image compression in digital cameras with color filter array," *IEEE Transactions on Consumer Electronics*, vol. 49, pp. 1448–1456, November 2003.
- [3] X. Xie, G. Liand, Z. Wang, C. Zhang, D. Li, and X. Li, "A novel method of lossy image compression for digital image sensors with Bayer color filter arrays," *IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 5, pp. 4995–4998, May 2005.
- [4] Joint Photographic Experts Group, "Information technology - digital compression and coding of continuous-tone still images - requirements and guidelines," *ISO/IEC IS 10918-1 — ITU-T Recommendation T.81*, September 1992.
- [5] Joint Photographic Experts Group. www.jpeg.org.
- [6] Joint Photographic Experts Group, "Information technology - digital compression and coding of continuous-tone still images - compliance testing," *ISO/IEC IS 10918-1 — ITU-T Recommendation T.83*, November 1994.
- [7] Joint Photographic Experts Group, "Information technology - digital compression and coding of continuous-tone still images - extensions," *ISO/IEC IS 10918-1 — ITU-T Recommendation T.84*, July 1996.
- [8] Joint Photographic Experts Group, "Information technology - digital compression and coding of continuous-tone still images - registration of jpeg profiles, spiff profiles, spiff tags, spiff colour spaces, appn markers, spiff compression types and registration authorities (regaut)," *ISO/IEC IS 10918-1 — ITU-T Recommendation T.86*, June 1998.
- [9] L. Agostini and S. Bampi, "Integrated digital architecture for JPEG image compression," *Proceedings of European Conference on Circuit Theory and Design (ECTD01)*, vol. III, pp. 181–184, 2001.
- [10] L. Agostini, I. Silva, and S. Bampi, "Pipelined fast 2-D DCT architecture for jpeg image compression," *Proceedings of the 14th Symposium on Integrated Circuit and System Design (SBCCI01)*, September 2001.
- [11] J. Yusof, Z. Aspar, and I. Suleiman, "Field programmable gate array (FPGA) based baseline JPEG decoder," *Proceedings of Conference on Convergent Technologies For The Asia-Pacific (TENCON00)*, vol. 3, pp. 218–220, 2000.

- [12] Y.-H. Han and J.-J. Leou, "Detection and correction of transmission errors in JPEG images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 221–231, April 1998.
- [13] G. Spampinato, A. Castorina, A. Bruna, and A. Capra, "Chromatic enhancement technique for JPEG images," *14th International Workshop on Systems, Signals and Image Processing (IWSSIP07), 2007 and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services (EC-SIP-M07)*, pp. 161–164, June 2007.
- [14] S. Battiato, M. Mancuso, A. Bosco, and M. Guarnera, "Psychovisual and statistical optimization of quantization tables for DCT compression engines," *Proceedings of International Conference on Image Analysis and Processing (ICIAP01)*, September 2001.
- [15] G. Sreelekha and P. Sathidevi, "An improved JPEG compression scheme using human visual system model," *14th International Workshop on Systems, Signals and Image Processing (IWSSIP07), 2007 and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services (EC-SIP-M07)*, pp. 98–101, June 2007.
- [16] M. Ramos and R. De Queiroz, "Adaptive rate-distortion-based thresholding: application in JPEG compression of mixed images for printing," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP99)*, vol. 5, pp. 2431–2434, March 1999.
- [17] A. Bruna, A. Capra, S. Battiato, and S. La Rosa, "Advanced DCT rate control by single step analysis," *Proceedings of International Conference on Consumer Electronics (ICCE05)*, pp. 453–454, January 2005.
- [18] M. Enari, N. Hoshi, and H. Takizawa, "Image processing apparatus and method using plural amounts corresponding to encoded data to control encoding," *US Patent 5838826*, November 1998.
- [19] S. Battiato, A. Bruna, G. Di Blasi, G. Gallo, and C. Bosco, "Statistical modeling of huffman tables coding," *Proceedings of International Conference on Image Analysis and Processing (ICIAP05)*, 2005.
- [20] R. Gonzales and R. Woods, *Digital Image Processing*. Addison Wesley, 1993.
- [21] P. Yip and K. R. Rao, *Discrete Cosine Transform: Algorithms, Advantages, and Applications*. Academic Press, 1990.
- [22] T. C. Tan, G. Bi, and H. N. Tan, "Fast recursive algorithms for 2-d discrete cosine transform," *Signal Processing*, vol. 80, pp. 1917–1935, September 2000.
- [23] "Jpeg 2000 image coding system part I," 2000.

- [24] S. Battiato, A. Buemi, G. Impoco, and M. Mancuso, "JPEG 2000 coded images optimization using a content-dependent approach," *IEEE Transactions on Consumer Electronics*, vol. 48, no. 3, pp. 400–408, 2002.
- [25] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG 2000 still image coding system: an overview," *IEEE Transactions on Consumer Electronics*, vol. 46, pp. 1103–1127, November 2000.
- [26] B. Shi, L. Liu, and C. Xu, "Comparison between JPEG 2000 and H.264 for digital cinema," *Proceedings of IEEE International Conference on Multimedia and Expo (ICME08)*, pp. 725–728, June 2008.
- [27] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, pp. 1158–1170, July 2000.
- [28] "Video coding for low bit rate communication," December 1995.
- [29] "Video codec for audiovisual services at px64 kbit/s," December 1990.
- [30] "Coding of audio-visual objects-part 2: Visual," April 1999.
- [31] "Draft ITU-T recommendation and final draft international standard of joint video specification," November 2002.
- [32] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," July 2003.
- [33] G. K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM*, pp. 30–44, December 1991.
- [34] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Compression Standard*. New York: Van Nostrand Reinhold, November 1993.
- [35] "Information technology - lossless and near lossless compression of continuous-tone still images: Baseline," December 1999.
- [36] S. Battiato, A. Bruna, A. Buemi, and F. Naccari, "Coding techniques for CFA data images," *Proceeding of IEEE International Conference on Image Analysis and Processing (ICIAP03)*, pp. 418–423, 2003.
- [37] T. Acharya, L. Karam, and F. Marino, "Compression of color images based on a 2-dimensional discrete wavelet transform yielding a perceptually lossless image," *U.S. Patent 6,154,493*, 2000.
- [38] D. L. Gall and A. Tabatabai, "Subband coding of digital images using symmetric shor kernel filters and arithmetic coding techniques," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP88)*, vol. 2, pp. 761–764, 1988.

- [39] S.-Y. Lee and A. Ortega, "A novel approach of image compression in digital cameras with a bayer color filter array," *Proceedings of International Conference on Image Processing (ICIP01)*, vol. 3, pp. 482–485, 2001.
- [40] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Transactions On Information Theory*, vol. 44, October 1998.
- [41] S. Battiato, A. Buemi, L. Della Torre, and A. Vitali, "A fast vector quantization engine for CFA data compression," *Proceedings of IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP03)*, pp. 123–127, June 2003.
- [42] A. Buemi, A. Bruna, F. Vella, and A. Capra, "Bayer pattern compression by prediction errors vector quantization," *Proceedings of International Conference on E-business and Telecommunication Networks (ICETE04)*, pp. 292–297, August 2004.
- [43] A. Bruna, S. Smith, F. Vella, and F. Naccari, "JPEG rate control algorithm for multimedia," *IEEE International Symposium on Consumer Electronics (ISCE04)*, pp. 114–117, September 2004.
- [44] S. Battiato, A. Bruna, A. Buemi, and A. Castorina, "Analysis and characterization of JPEG 2000 standard for imaging devices," *IEEE Transactions on Consumer Electronics*, vol. 49, pp. 773–779, November 2003.
- [45] D. Taubman, E. Ordentlich, M. Weinberger, and G. Seroussi, "High performance scalable image compression with EBCOT," *Proceeding of IEEE International Conference on Image Processing (ICIP01)*, vol. 2, pp. 33–36, 2001.
- [46] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and videocompression," *IEEE Signal Processing Magazine*, vol. 15, pp. 23–50, November 1998.
- [47] H. Everett, "Generalized lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, vol. 11, pp. 399–417, 1963.
- [48] A. Bruna and M. Mancuso, "JPEG compression factor control: A new algorithm," *Proceeding of IEEE International Conference on Consumer Electronics (ICCE01)*, pp. 206–207, June 2001.
- [49] C. Chou and Y. Li, "A perceptually tuned subband image coder based on the measure of just-noticeable-distortion profile," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 5, pp. 467–476, December 1995.
- [50] S. Daly, W. Zeng, J. Li, and S. Lei, "Visual masking in wavelet compression for JPEG 2000," *SPIE-IS&T Electronic Imaging: Image and Video Communication and Processing*, January 2000.
- [51] M. G. Perkins and T. Lookabaugh, "A psychophysically justified bit allocation algorithm for subband image coding systems," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP89)*, pp. 1815–1818, May 1989.

- [52] K. A. Wen, C. Lu, and M. C. Tsai, "The transform image codec based on fuzzy control and human visual system," *IEEE Transactions on Fuzzy Systems*, vol. 3, pp. 253–259, August 1995.
- [53] S. Cei and P. Cosman, "Comparison of error concealment strategies for MPEG video," *Proceedings of Wireless Communications and Networking Conference (WCNC99)*, vol. 1, pp. 329–333, September 1999.
- [54] G. Spampinato, M. Guarnera, F. Vella, and A. Buemi, "Improved error detection and concealment techniques for MPEG-4," *Proceedings of IEEE International Conference on Consumer Electronics (ICCE03)*, pp. 394–395, June 2003.
- [55] G. Spampinato, M. Guarnera, F. Vella, and A. Buemi, "Efficient error handling for video and still picture standards," *Journal of the IEEE Transactions on Consumer Electronics*, pp. 1391–1394, November 2003.
- [56] F. Dufaux, G. Sullivan, and T. Ebrahimi, "The JPEG XR Image Coding Standard," *IEEE Signal Processing Magazine*, vol. vol. 26, no. no.6, pp. pp. 195–199, 204–204, 2009.

Quality Metrics

Ivana Guarneri

Advanced System Technology - Catania Lab, STMicroelectronics, Italy.

Abstract: The image quality depends on compromises made in the design of the algorithms and devices for image capture, transfer, storage, and display. Quality assessment plays, consequently, a decisive role to successfully promote image processing algorithms and system performances. The evaluation of image quality is basically a human issue, but subjective metrics are computationally expensive and not practical for real-time applications. So, in the last decades, a great deal of efforts has been put into the development of objective quality metrics able to automatically predict perceived quality. To design an objective measure capable to be in close agreement with subjective test is a difficult task because the human visual system has a multifaceted structure and is not totally known. The tracked problem is to emulate human vision which is a cognitive activity and not a pure image sensor process. In this Chapter a classification of quality indexes, from classical to recent approaches is reported. For a better understanding of quality assessment topic, some of the principal human phenomena involved in the development of objective perceptual metrics are also explained.

12.1 Introduction

To build an automatic quality measure able to assess the overall quality of a given image is a very complex task. The difficulties are mainly due to the fact that the image quality is intrinsically a subjective issue. The more reliable and effective quality assessment method is the subjective one, but, on the other side, the subjectiveness implies that a given image could be judged by viewers in different ways. This happens because the viewer's ability to express a quality score comes from a complex analysis of the image made by the Human Visual System (HVS). Many factors are involved in the process of image quality evaluation, from the physical structure of eye-apparatus to the cognitive and psychological personal reaction. The majority of these aspects is not totally known, so many researches are still interested to the image quality topic. The ability of HVS to assess the image quality without a reference is the most complex visual system behavior to emulate. The human is able to formulate a judge only by considering how much pleasing is the perceived sensation derived by looking at the image without the needs of having a reference, and the satisfaction degree is the final expression of human subjectiveness. In the last decade many works about automatic quality measures have been proposed and in the following sections is reported a brief description of the main developments dealing with the image quality metrics.

12.2 Full, Reduced and No-Reference Approaches

When comparing two images, many are the features used to gauge the overall image quality as the contrast, the focus, the dynamic range, the noise, the compression artifacts and so on. The most common criterion to classify quality metrics is based on the avail-

ability of the original image/video signal, which is considered to be of high quality and can be used as a reference in evaluating the quality of a distorted image. According to the amount of information that is available, objective metrics can be classified in: Full Reference (FR), Reduced Reference (RR) and No-Reference (NR).

Full Reference

Full Reference metric (FR) assesses the quality of a distorted image relatively to a reference image which is assumed to be an original perfect version of the test image. In this case both the target image and the distorted one are required. To implement a FR index is not a complicated task, it regards the measurement of the similarity between two images [1–3]. The difficulty is to find a measure able to assess a wide range of impairments that well correlates with the judgment given by a human observer. On the other hand, the FR metric is not applicable in real-time applications where the reference image is often not available.

Reduced Reference

Reduced Reference metrics (RR) require the distorted image and partial information about the reference one. These kinds of metrics are mostly utilized to assess the quality at the end of a video or image transmission system [4–6]. The encoder, together with the digital transmission method and the decoder, can produce some lost of quality in the final reconstructed image or video. Usually, together with the input image, are transmitted also some parameters relative to the input (compression rate, etc.). These features are used by the RR metric to assess the quality of transferred images or videos.

No Reference

No Reference (NR) or 'blind' metrics only require the distorted image. The quality assessment in absence of a reference is a very difficult task and it becomes quite impossible to evaluate the quality of an image affected by a wide variety of distortion types. The NR metric works well when it is necessary to assess a specific impairment. Knowing previously the model of the distortion it is possible to build a metric able to measure the amount of specific impairment. NR works have been proposed to evaluate the quality in different fields of image processing as the High-Definition Video [7], or the JPEG compression [8–11], or the image fusion [12–14].

12.3 Subjective Metrics

The FR, RR and NR, are objective measures of the image quality, i.e., their design is primarily based on extracting and analyzing some specific features or artifacts in the image and, consequentially, on the automatic formulation of a quality score. On the other hand for subjective quality evaluation is intended the quality assessment provided by a human viewer. Due to the complex analysis made by the human brain the subjective evaluation

is the most accurate way to determine the image quality. Considering that each viewer has its personal and unique perception of quality, a method to formulate a subjective quality score consists in the averaging many viewer's quality judges. The judges should be collected by using the same evaluation method, the same scale rating and the same environment condition (room-lighting, type of monitor, distance from monitor, etc.). To set up a subjective test is expensive both in terms of time and resources and, cannot be easily performed in real-time. In the recent years the International Telecommunication Union (ITU) has developed recommendations on subjective quality assessment methodologies, which describe the main directions to follow for embedding a subjective quality evaluation test. One of the most important is the Recommendation BT.500 [15] which provides the fundamental description of the subjective image and video quality assessment methods, recently replaced by BT.6/150 (2009) [16]. Another important document for subjective quality assessment is the final report of 2003 [17] provided by the Visual Quality Expert Group (VQEG). These two important documents are described in the Sections 12.3.1 and in 12.3.2 respectively.

12.3.1 ITU-R Recommendation BT.500

The VQEG [18], formed in 1997, has given an important contribution to the video and image quality assessment. The ITU-R BT.500 [15], released by VQEG, explains standard viewing condition, criteria for observers, test scenes selection and specific assessment procedures. The method for subjective video and still images quality analysis involves a group of prior selected human viewers which analyze the video sequence and express their personal image quality score. The mean of all the collected judges constitutes the Mean Opinion Score (MOS). The analysis procedure can be described in two main phases. The former regards the selection of the test video (which are H.263, H.264, and MPEG-2 compressed video sequences, see Chapter 11), the selection of the 20 human viewers, the set up of the testing environment, and, the selection of the specific method to apply. The latter phase regards the gathering of all quality scores and the calculation of final MOS. The viewers formulate their personal judgments about quality in semantic terms, the score expresses how much unpleasant is the impairment sensation coming from the test image. The scale proposed in this recommendation is reported in Fig.(12.1). The rank varies in the range [1-5] where the lower is the score the weaker is the image quality. To validate a new proposed quality metric, it must be performed on VQEG's video sequences and, then, compared with the MOS results. But the VQEG image/video databases are not royalty free, so the new index is usually provided to the VQEG to be tested on their mandatory databases, in this way is possible to compare the index results with the subjective ones. Four are the test methodologies proposed by VQEG:

- Double Stimulus Continuous Quality Score (DSCQS);
- Double Stimulus Impairment Scale (DSIS);
- Single Stimulus Continuous Quality Evaluation (SSCQE);
- Simultaneous Double Stimulus for Continuous Evaluation (SDSCE).

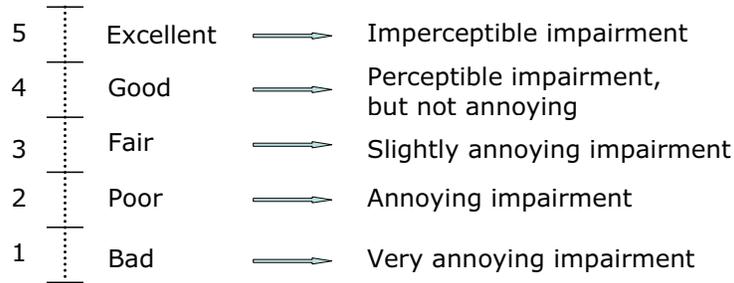


Figure 12.1 : Scale of MOS.

Double Stimulus Continuous Quality Scale

The Double Stimulus Continuous Quality Scale (DSCQS) method shows to viewers multiple short sequence pairs, of about 10 seconds, consisting of "reference" and "test" sequences. The short sequences are presented twice in alternating fashion and in a randomly order. Moreover, subjects are not informed which the reference sequence is or the test sequence is. The viewers should express a judge for each sequence on a continuous quality scale ranging from "bad" to "excellent". The analysis is based on the difference in rating for each pair, which is often calculated from an equivalent numerical scale from 0 to 100.

Double Stimulus Impairment Scale

The Double Stimulus Impairment Scale (DSIS) method, differently than before, always shows the reference before the test sequence, and the two sequences are proposed just once. In this case viewers know which is the reference and their score should express, in a continuous way, how much annoying is perceived the impairment on the test. The evaluation is expressed as one of the five levels of a scale ranging from "very annoying" to "imperceptible" (Fig.(12.1)).

Single Stimulus Continuous Quality Evaluations

The idea of a continuous evaluation is affected by some lacks. In particular, the previous described methodologies fix, as viewing time duration for the score formulation, a period of 10s; this is an insufficient interval of time for the viewer to formulate an appropriate judge relative to the context-related artifacts observed in the scene. To overcome this lack the Single Stimulus Continuous Quality Evaluations (SSCQE) method has been proposed. Performing the SSCQE the viewers watch a program of typically 20-30 minutes instead of seeing separate short sequence pairs. Using a slider, the subjects continuously rate the instantaneously perceived quality on a scale ranging from "bad" to "excellent".

Simultaneous Double Stimulus for Continuous Evaluation

The SSCQE method tries to reproduce the viewing condition of a real situation by not using a reference image sequence. However, there are cases where it is requested to have a fidelity measure, hence reference conditions must be introduced. The Simultaneous Double Stimulus for Continuous Evaluation (SDSCE) method has been developed starting from the SSCQE, by making slight variation concerning the way of presenting the images to the subjects and the used rating scale. The method can be suitably applied to all those cases where fidelity of visual information affected by time-varying degradation has to be evaluated.

The four described methods generally have different application. The DSCQS, being sensitive to small differences, is the preferred method when the quality of test and reference sequences is similar. The DSIS is suitable for evaluating clearly visible impairments such as artifacts caused by transmission errors. The SSCQE and the SDSCE have been designed in order to relate well to the time varying quality of today's compressed video system. The methods reported in [15] are all video testing procedures which can be adapted in the still domain, i.e., the assessment made on a video can be reported to each single frame.

12.3.2 The 2003 VQEG's Report

Another important document relative to the image quality assessment, written by the VQEG, is the final report of 2003 [17]. It introduces some fundamental concepts today widely used to build a new objective metrics, in particular, the report is articulated in two parallel sections. The first one explains a new subjective test, and, the second one suggests methodologies to evaluate the metric accuracy and consistency. The subjective test is implemented by using the DSCQS according to which the viewers should express their opinion score both for the source and the test sequences. To build a model sufficiently independent by specific impairment, two video sequences affected by a wide range of distortion types are processed. The final judgment is expressed as a difference score between the rating for the reference sequence minus the rating for the test one (Differential Mean Opinion Score - DMOS). The second part of the test introduces new methodologies for correctly design a metric. An Objective Metric (OM) should be able to predict the subjective score with a low error, furthermore it should predict the degree to which the models prognostication agrees with the relative magnitudes of subjective quality ratings, and, finally, it should be robust in presence of different impairments. A well designed metric should satisfy the following statistical properties:

- **PREDICTION ACCURACY:** the ability to predict the subjective quality score with low error;
- **PREDICTION MONOTONICITY:** the ability to predict the degree of model results with the subjective score;
- **PREDICTION CONSISTENCY:** the degree of robustness of the model with respect to a wide range of impairments.

These attributes are verified not directly on the OM but on a function of OM, the so called "predicted DMOS". This is dictated by the observation that the outputs of the OM should be correlated with the viewer Difference Mean Opinion Scores in a *predictable and repeatable* manner. So, a nonlinear regression function was used to transform the OM values to a set of predicted-DMOS (DMOS_p):

$$DMOS_p = \frac{b_1}{1 + e^{(-b_2 \cdot (OM - b_3))}} \quad (12.1)$$

where b_1 , b_2 and b_3 are determined by the logistic regression of the *OM* with the *DMOS*. This kind of function has been introduced to better handle the problem of non-linearity due to the subjective rating. The OM's prediction performance in terms of accuracy, monotonicity and consistency, is evaluated on the sets of subjectively measured DMOS and predicted DMOS_p, by using metrics specified in [17].

12.4 Objective Metrics

Objective measures are used to estimate the image quality in an automatic manner, usually by measuring the difference between a reference image and a distorted one. Indexes belonging to the FR class are the Signal to Noise Ratio (6.2), the Mean Squared Error (MSE) and the Peak Signal-to-Noise Ratio (PSNR). The MSE between a reference image I and its distorted version \bar{I} , is defined as:

$$MSE = \frac{1}{N \cdot M} \sum_{n=1}^N \sum_{m=1}^M [I(n, m) - \bar{I}(n, m)]^2 \quad (12.2)$$

where $(M \times N)$ is the image size. It should be observed that the MSE strongly depends on the image intensity scaling. A mean-squared error of 100.0 for an 8-bit image (with pixel values in the range 0-255) looks dreadful; but a MSE of 100.0 for a 10-bit image (pixel values in [0,1023]) is barely noticeable. The PSNR, measured in decibel (dB), avoids this problem by scaling the MSE according to the image range:

$$PSNR = 10 \log_{10} \frac{S^2}{MSE} \quad (12.3)$$

where S is the maximum pixel value.

Both the MSE and the PSNR are indexes widely used because easy to compute, and, capable to give an effective difference value between a signal and its distorted version. On the other hand, the MSE and PSNR are not considered "ideal" quality measures, it can happen that one image with a PSNR of 20dB may look much better than another image with a PSNR of 30dB. This happens because the PSNR is not a reliable predictor of the perceived visual quality. Two examples of this deficiency are showed in Fig.(12.2) [19]. The first row shows a reference image (Fig.(12.2(a))) and two tests obtained by using two different quantization techniques, the first one, indicated as Cut Off (CO), uses a coarse truncation of the intensity values from 8 to 4 bits, cutting off the 4 less significant

bits; the other one is the Improved Gray Scale (IGS) technique [20], which avoids the coarse approximation by adding a pseudo-random noise before truncation. The second example shows a reference image (Fig.(12.2(d))) which has been distorted by adding impulsive noise and successively cleared by using two different de-noising algorithms. The two examples give an idea of the limits of PSNR in terms of correlation with subjective evaluation. This misleading evaluation happens because the PSNR and the MSE can be



Figure 12.2 : Two example where the PSNR fails. In both cases, the PSNR does not well correlate with subjective quality preference.

considered fidelity metrics which are based solely on a quantitative mean. As explained in [21], there are some implicit assumptions to make when the PSNR or the MSE are used as fidelity criterion:

- "Signal fidelity is independent of temporal or spatial relationships between the samples of the original signal". It means that if the reference image and the distorted one are randomly re-ordered in the same way then their MSE or their PSNR remain the same.

- "Signal fidelity is independent of any relationship between the original signal and the error signal". It means that, adding a given error signal to an original one, the PSNR or the MSE remain the same.
- "Signal fidelity is independent of the signs of the error signal samples".
- "All signal samples are equally important to signal fidelity".

These assumptions fix some important limitations relative to the interaction of signal with the error. But, the point is that not one of all above assumptions is relevant in the context of measuring the visual perception of image fidelity.

12.5 Perceptual Objective Metrics

The purpose of objective metrics is to automatically assess the quality of images in agreement with human judgments. Many different objective criteria have been proposed in literature, some approaches are simply based on statistical properties of the image, others are based on complex models of the HVS. With the terms "perceptual objective metric" here is intended metrics which model some HVS behaviors. The design of a perceptual objective metric is a very difficult task because the HVS is not totally known, and there are also some of them which are quite impossible to emulate (i.e., the capability to judge an image without having a reference). Anyway, for planning a perceptual objective metric it is necessary to be familiar with the principal phenomena concerning the human vision. A light stimulus which hits a human eye activates a mechanism of signal interpretation which involves many aspects, as the spatial frequency sensitivity, the contrast masking or the semi-local masking, the pooling, etc. The brain elaborates the incoming information and formulates a final quality score according to its personal subjectiveness. To well understand which mechanisms are involved on the subjective quality assessment a brief overview of some of them is reported below.

12.5.1 Lightness Perception

The human visual system is a dynamic system able to adapt to a large range of light intensities. This phenomenon is due, essentially, to three mechanisms:

- the mechanical variation of the papillary aperture;
- the chemical processes in the photoreceptors;
- the adaptation at the neural level.

These mechanisms, together with many others, determine a very strong non linear behavior of the visual system, as the not linearly perception of the real luminance of the observed object. Many subjective experiments have been made in order to understand this behavior, here is described one of the most simple. Observers were supplied with

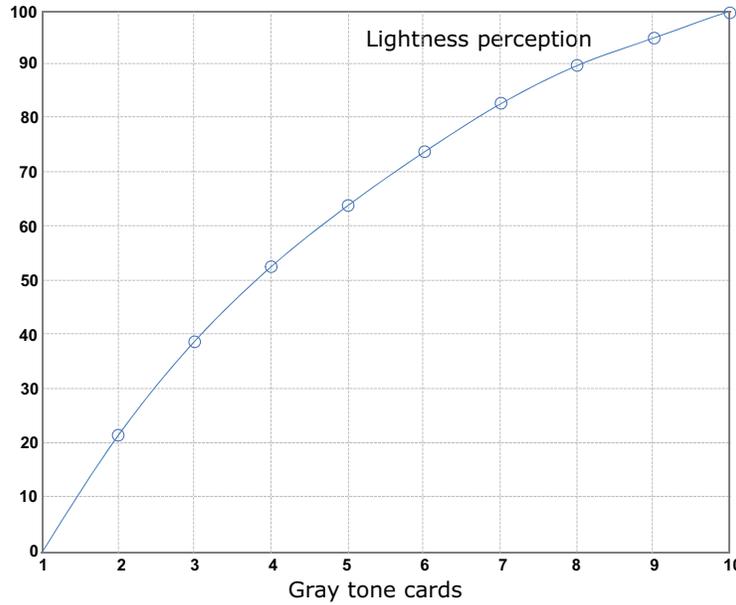


Figure 12.3 : Non linear relationship between equal-step gray colors and perceived luminance.

about 100 tone-cards of a different gray color. They were asked to select 10 of them, so that they span the range between black and white in an equidistant manner. The sensation associated with a tone is taken to be the number of steps from black, so the curve of steps-number versus luminance is showed in Fig.(12.3).

The relationship is clearly non-linear. Many authors had studied this aspect of visual system using psycho-visual experiment in order to tune their proposed formulas. Two functions widely used in literature are the ones proposed by Daly [22] and by Mannos-Sakrison [23]. The equation (12.4) is a simplified version of the Daly's formula, obtained by assuming that the adaptation level for an image pixel is solely determined from that pixel (L' is the lightness perceived and L represents the lightness of the pixel expressed in cd/m^2 , see Fig.(12.4)). In the Mannos-Sackrison's formula (12.5) the lightness perceived L' is a function of the mean luminance level L_m . The Fig.(12.5) shows the Mannos-Sakrison formula in the pixel range [0,255].

$$L' = L / (L + 12.6L^{0.63}) \quad (12.4)$$

$$L' = (L/L_m)^{0.333} \quad (12.5)$$

12.5.2 Color Encoding

The perception of the color is a complex physical mechanism of HVS. The main actors are the photoreceptors which are specialized neurons that make use of light-sensitive photochemicals to convert the incident light energy into signals that can be interpreted by the

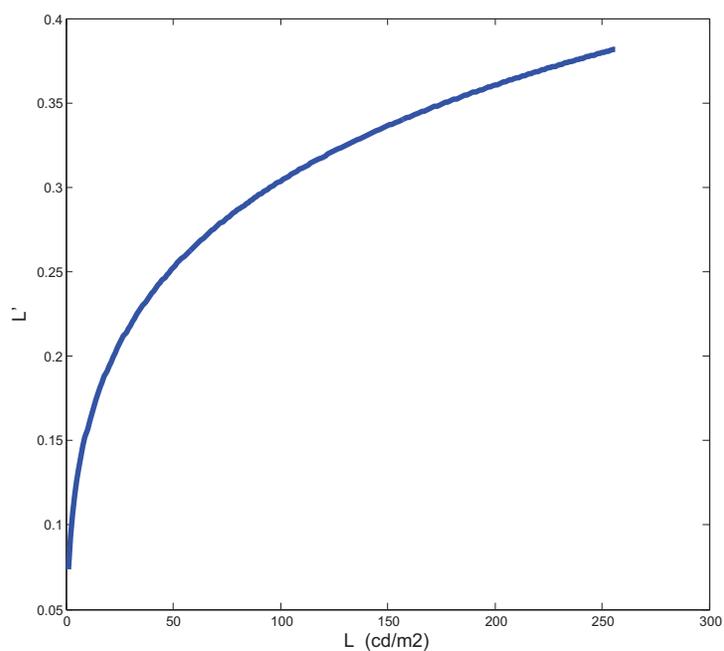


Figure 12.4 : Lightness perception formula proposed by Daly.

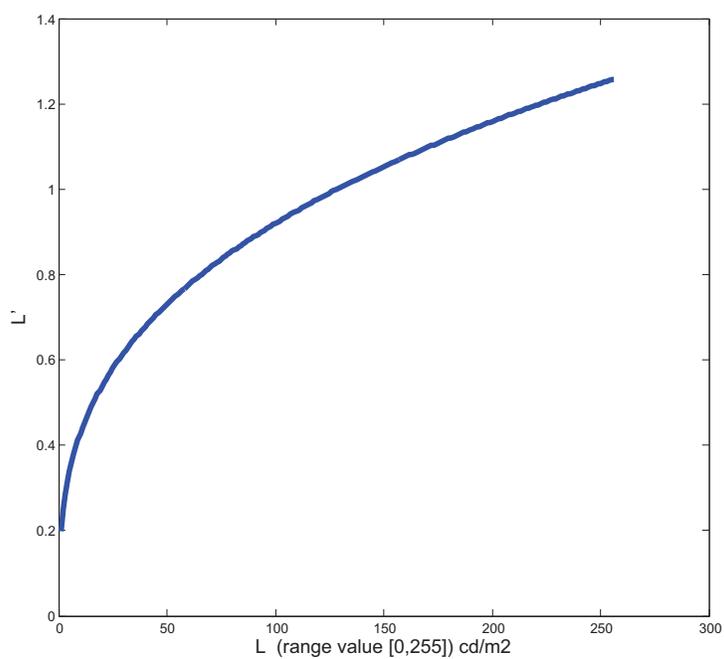


Figure 12.5 : Lightness perception formula proposed by Mannos-Sakrison.

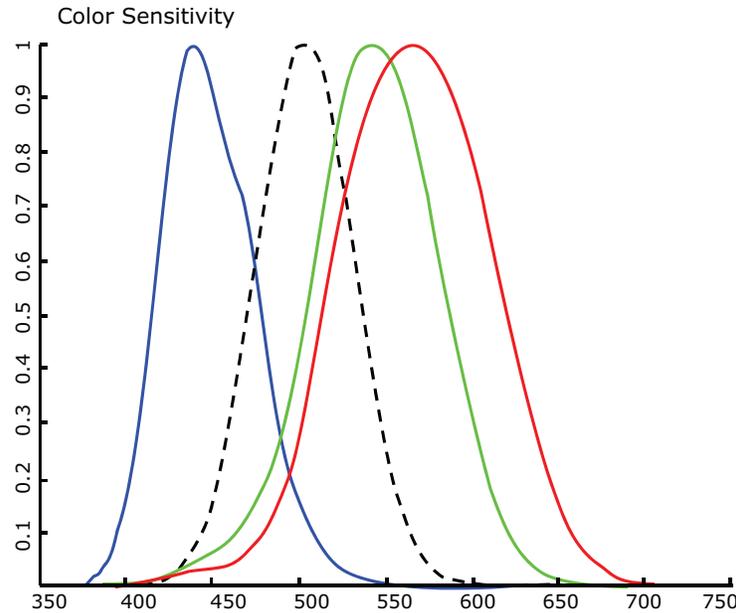


Figure 12.6 : Normalized cone sensitivity functions of photoreceptors with respect to the visible electromagnetic spectrum; the colored lines are relative to the L, M, and S cones, while the dashed line is relative to the rods.

brain. There are two different types of photoreceptors, namely *rods* and *cones*. Rods are responsible for scotopic vision¹ at low light levels, while cones are responsible for photopic vision² at high light levels. Even if the rods are very sensitive light detectors and sample the retina very finely, the visual acuity under scotopic conditions is poor. This happens because all the rods signals are conveyed onto a single neuron, this corresponds to a high sensitivity but, also, to a low resolution. The cones can be classified in three types according to their photo-chemicals spectral sensitivity: L-cones, M-cones, and S-cones. The L-cones are sensitive to the long wavelength, the M-cone to the medium and the S-cone to the short ones. They form the basis of color perception. The absorption spectra of the three cone types [24] are shown in Fig.(12.6).

Physiological experiments show that cones have an excitatory or an inhibitory effect on ganglion cells in the retina and on cells in the lateral geniculate nucleus. Depending on the cone types, certain excitation/inhibition pairings occur much more often than others: neurons excited by "red" L-cones are usually inhibited by "green" M-cones, and, neurons excited by "blue" S-cones are often inhibited by a combination of L- and M-cones. Hence, the receptive fields of these neurons suggest a connection between neural signals and perceptual opponent colors. Although many studies on the color perception of HVS have

¹The scotopic vision is the eye vision under low levels of illumination (10^{-2} to 10^{-6} cd/m^2), it is produced exclusively through rod cells and it is completely lacking in color.

²The photopic vision is the eye vision under well-lit conditions, it allows color perception provided by the cone cells.

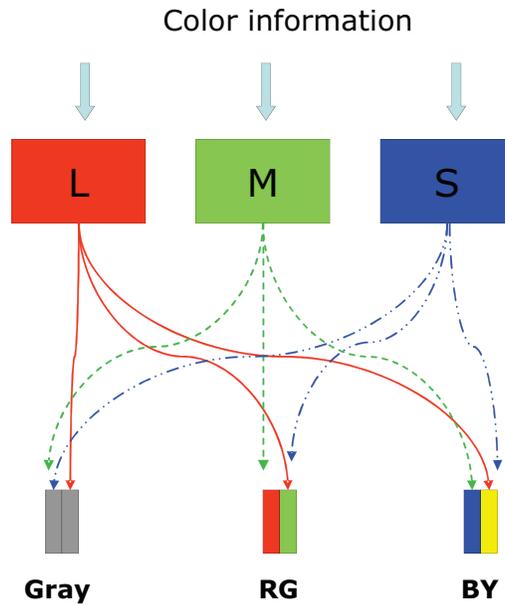


Figure 12.7 : Photo-receptors combine color information in one achromatic and two chromatic channels.

been done, the right opponent-color encoding is still a subject to debate (Fig.(12.7)).

The Fig.(12.8) shows a simple experiment able to demonstrate, in a practical way, the mechanism of opponent color in our visual system, the example helps to understand how the HVS build the named "opponent after image". The "after image" can be visualized following the next steps:

1. fixate upon the black spot in the center of the yellow circle surrounded by the blue square for at least 30 seconds;
2. move your gaze to fixate the black spot in the white area (on the right);
3. you will see the "after image" in the opposite colors (a blue circle into a yellow square).

12.5.3 Opponent Color Space

The RGB color spaces are device dependent and not perceptually linear, so, even if widely used for coding digital images, they cannot be used for HVS based models. The RGB are usually converted into a perceptually linear and device independent color space (e.g., the sRGB). According to the opponent theory, the sensation of red and green, as well as blue and yellow, are processed in separate visual pathways [25, 26], so, in the opponent color space three channels are defined as: black-white (BW), red-green (RG) and blue-yellow

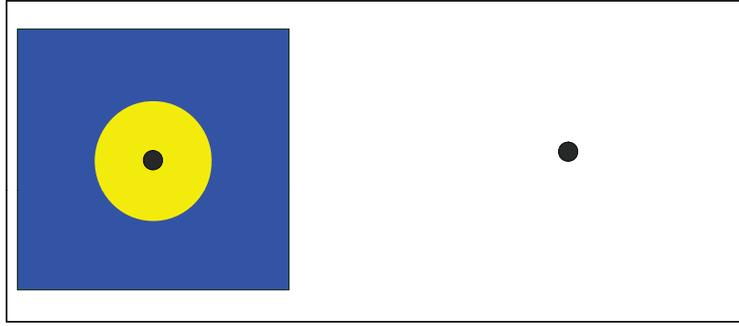


Figure 12.8 : Example of hue-cancellation experiment.

(BY). The sRGB is converted in opponent color space through a series of transformations.

$$\begin{bmatrix} sR \\ sG \\ sB \end{bmatrix} \rightarrow \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \begin{bmatrix} L \\ M \\ N \end{bmatrix} \rightarrow \begin{bmatrix} BW \\ RG \\ BY \end{bmatrix} \quad (12.6)$$

Firstly, the input image is converted from the RGB color space to CIE XYZ tristimulus which is device independent.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4306 & 0.3415 & 0.1784 \\ 0.2220 & 0.7367 & 0.0713 \\ 0.0202 & 0.1295 & 0.9393 \end{bmatrix} \times \begin{bmatrix} sR \\ sG \\ sB \end{bmatrix} \quad (12.7)$$

Secondly, the tristimulus values are converted in the L, M and S cones responses by using the following formula:

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.240 & 0.854 & -0.044 \\ -0.389 & 1.160 & 0.085 \\ -0.001 & -0.354 & 0.573 \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (12.8)$$

Finally, the LMS values are converted to the opponent color space WB, RG and BY, by using the following transformation:

$$\begin{bmatrix} BW \\ RG \\ BY \end{bmatrix} = \begin{bmatrix} 0.990 & -0.106 & -0.094 \\ -0.669 & 0.742 & -0.027 \\ -0.212 & -0.354 & 0.911 \end{bmatrix} \times \begin{bmatrix} L \\ M \\ S \end{bmatrix} \quad (12.9)$$

Differently to some widely used color spaces for quality metric design, as the CIE L*a*b and the CIR L*u*v [27], the opponent color space has the advantage that it well separates the color perception from the pattern sensitivity, this aspect contributes to a suitable modularity of the quality metric model.

12.5.4 Luminance Contrast

The visual system detects changes in luminance rather than absolute values. Usually the luminance contrast is expressed as a ratio of the type:

$$\text{Luminance Contrast} = \frac{\text{Luminance change}}{\text{Luminance adaptation}} \quad (12.10)$$

The rationale behind this form is that a small difference is negligible if the average luminance is high, while the same small difference matters if the average luminance is low.

Different formulas describe the luminance contrast and, most of them, reflect the fact that the adaptation state of the eye is affected differently by various stimulus patterns.

One of the most known luminance contrast definition is the Weber one (12.11), where the visual stimulus is a small feature present on a large uniform background, i.e., the average luminance is approximately equal to the background luminance:

$$C_W = \frac{L - L_{background}}{L_{background}} \quad (12.11)$$

where L and $L_{background}$ represent, respectively, the luminance of the feature and the luminance of the background.

When the visual stimulus is a sine wave grating than the Michelson's formula is used. This formula expresses the contrast as the ratio between the difference and the sum of light and dark bar luminances.

$$C_M = \frac{L_{max} - L_{min}}{L_{max} + L_{min}} \quad (12.12)$$

In the Michelson's equation (12.12) L_{max} and L_{min} are respectively the maximum and minimum luminances of the bars of the grating. This formula is commonly used for patterns where both bright and dark features are equivalent and take up similar fractions of the area. The denominator represents twice the average of the luminance.

Quantitative comparisons among the various contrast measures are not properly correct because they are designed to describe different spatial patterns and have different mathematical behaviors. Nevertheless, under simplified and limited conditions it can be acceptable to perform few comparisons. One case is when the feature and the background appear to have lightness ranging from black to white. The Fig.(12.9) shows a comparison between Weber and Michelson laws, in particular the red curve represents the Weber contrast obtained by fixing a feature of 50 luminance and varying the background in the range [55,255]. Similarly, the green curve represents Michelson values obtained superimposing a dark bar of 50 luminance on a background varying also in the range [55,255]. The difference between these two measures is small when contrasts are low, but as contrasts increase, the difference can be substantial. Considering a Weber contrast of 50%, obtained with a 100 luminance background, than we can have the same Michelson contrast by increasing the light bar (or the background for this example) till 150 luminance.

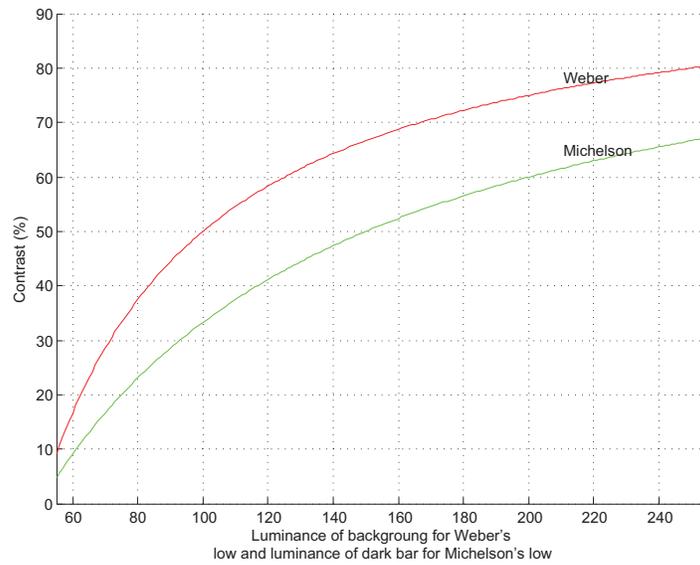


Figure 12.9 : Weber vs Michelson example. In the Weber experiment the feature has been fixed at 50 luminance and the background varies in the range [55,255]. Analogously, in the Michelson experiment the dark bar has been fixed at 50 luminance and the white bar (or background) varies also in the range [55,255].

The human visual system is sensible, in a different way, to stimulus frequencies. The Fig.(12.10) shows the "Champell-Robson" chart [28] used to demonstrate, in a very simple way, how the HVS perceives the contrast. The chart shows the pixel values which have a sinusoidal trend along the horizontal direction, and, the frequency increases exponentially from left to right, while the contrast decreases exponentially from bottom to top. The minimum and the maximum luminance remain constant along a given horizontal line through the image. The alternated bright and dark bars should appear to have equal height everywhere in the image, but our visual system perceives bars in the middle taller than bars at the sides. This happens because human eyes are more sensitive to average frequencies than to low or high ones.

The Fig.(12.11) shows the Contrasts Sensitivity Function (CSF) (12.13) proposed by Mannos-Sakrison.

$$CSF(f) = 2.6 \cdot (0.0192 + 0.114f) \cdot e^{(-0.114f)^{1.1}} \quad (12.13)$$

The frequency is usually expressed in cycle for degree (cpd). The cycle number indicates how many periods of the sinusoidal signal are shown (Fig.(12.12)). It can be observed that the contrast sensitivity is maximum for frequencies of about 4-6 cpd (Fig.(12.11)).

Another aspect related to the contrast perception is the HVS capability to perceive small differences in luminance; the lower variation (i.e., the minimal difference that can be perceived) depends on the background luminance. This minimum difference is called

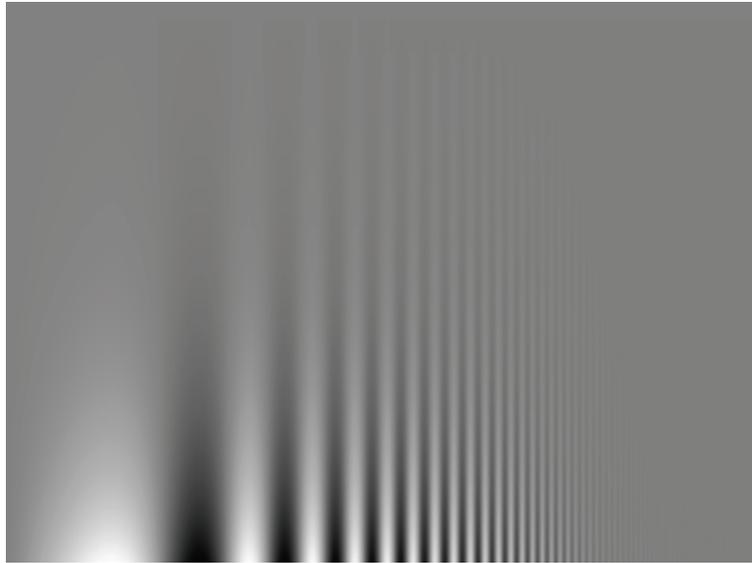


Figure 12.10 : "Champell-Robson" chart.

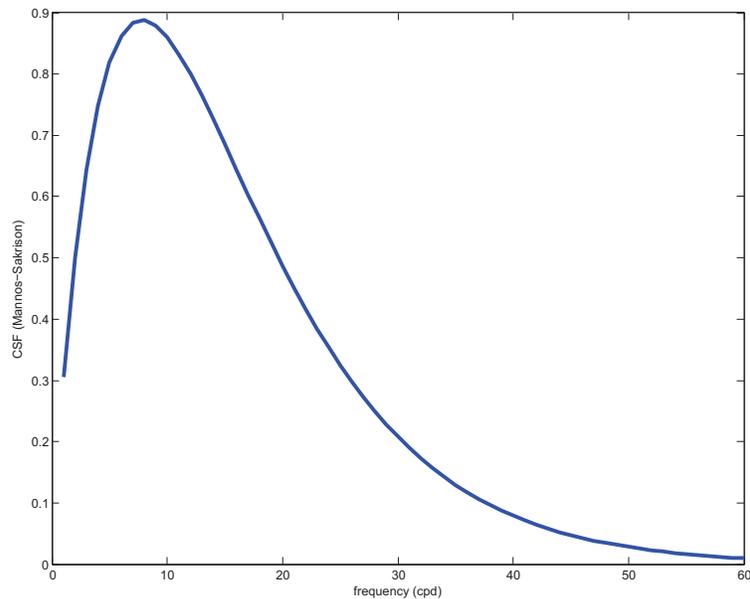
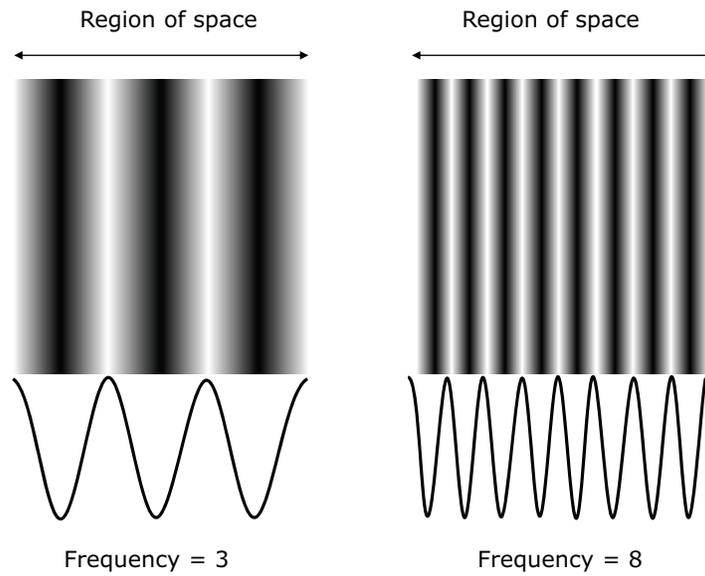


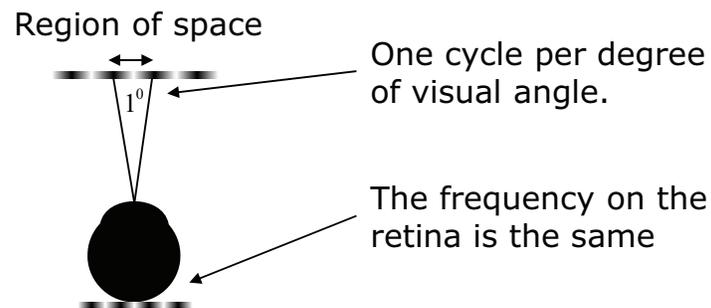
Figure 12.11 : Mannos-Sakrison formula of Contrast Sensitivity Function.

Just Noticeable Difference (*JND*) and it can be measured with a simple psycho-visual experiment.

For instance, the visual experiments for the case of Weber contrast is carried on as follows. A luminous patch is projected on a uniform background having L_a lightness. The patch intensity is slowly increased until it will be discernable from the background.



(a) $SF = n\text{-cycles/region of space}$



(b) Cycles/degree visual angle

Figure 12.12 : Cycle per degree.

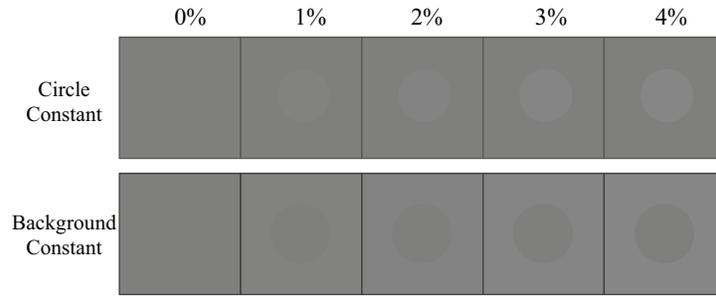


Figure 12.13 : Just Noticeable Difference at different percentage values.

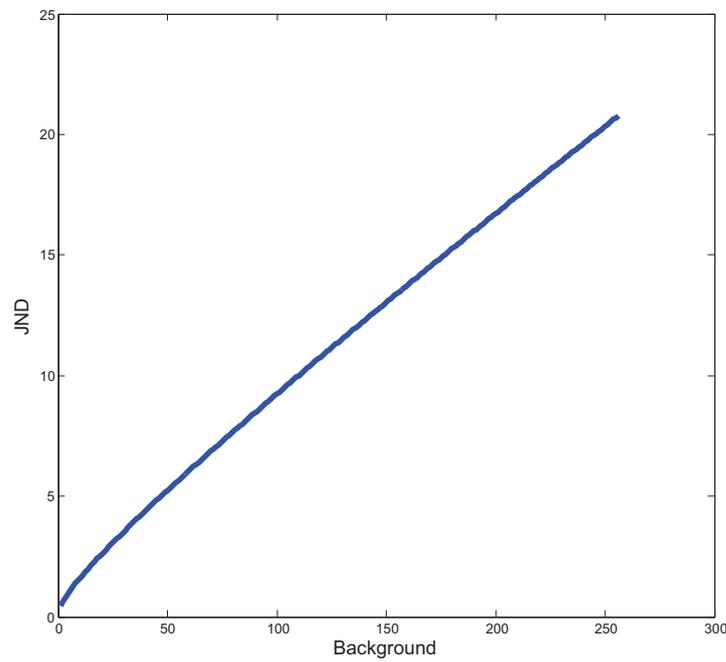


Figure 12.14 : Blackwell's *JND*.

The difference between L_a and the visible patch is the *JND*. Hence, a superimposed patch having $L_a + JND(L_a)$ luminance is discernible, but a patch of luminance $L_a + \epsilon$, where ϵ is lower than $JND(L_a)$, is not visible (Fig.(12.13)).

Many *JND* formulas have been proposed in literature. Blackwell [29] established the relationship between the adaptation luminance and the *JND* using a briefly flashing dot on a uniform background (Fig.(12.14)):

$$JND(L_a) = 0.0594 \cdot (1.219 + L_a^{0.4})^{2.5} \tag{12.14}$$

A more complex function (12.15), which accounts for both rod and cone responses

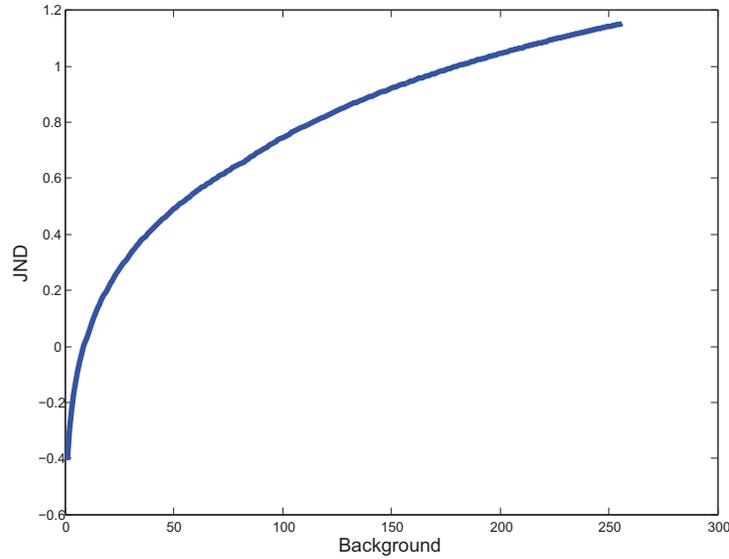


Figure 12.15 : Ferwerda's *JND*.

and tuned for the whole range of human vision, has been proposed by Ferwerda [30](Fig.(12.15)):

$$JND(L_a) = \begin{cases} -2.86 & \text{if } \log(L_a) < -3.94 \\ (0.405 \cdot \log(L_a) + 1.6)^{2.18} - 2.86 & \text{if } -3.94 \leq \log(L_a) < -1.44 \\ \log(L_a) - 0.395 & \text{if } -1.44 \leq \log(L_a) < -0.0184 \\ (0.249 \cdot \log(L_a) + 0.65)^{2.7} - 0.72 & \text{if } -0.0184 \leq \log(L_a) < 1.9 \\ \log(L_a) - 1.255 & \text{if } \log(L_a) \geq 1.9 \end{cases} \quad (12.15)$$

12.5.5 Masking

The *masking effect* is the visual system behavior according to which the perception of a signal A is somehow inhibited by a masking signal B. In the following is briefly discussed the interaction between a *signal* and a *masker*.

Signals and maskers can interact in different ways: two common masking effects are the contrast masking and the activity masking. The contrast masking regards the interaction of a signal and a masker which can have, or not, similar properties or appearance. The two examples of Fig.(12.16) show two sinusoidal patterns which are superimposed to create a third image. If the masker is not able to hide the signal, the third image is different to the first two. If the masker successfully hides the signal, the final third image appears like the masker itself. In the first example of Fig.(12.16(a)) the masker and the signal have the same orientation but different frequencies, in this case, the masking happens successfully because the final image appears equal to the masker image. In the second example

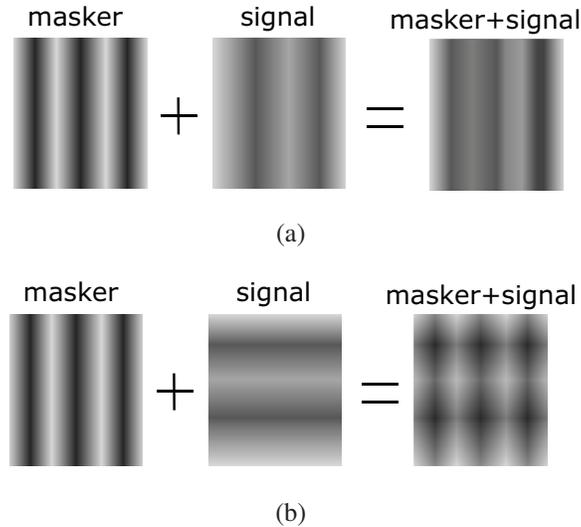


Figure 12.16 : In the first example (a) the masker successfully hides the signal; in the second example (b), where the signal and the masker have different frequencies and orientation, the masking has a bad result.

Fig.(12.16(b)) the masker and the signal have different orientations and frequencies, so the masking has a bad result.

The analysis of masking effect is bounded to the "multi-channel theory" according to which the HVS perceives visual information via various parallel channels. These channels differ in terms of their spatial and spectral frequencies as well as their orientation. The large variety of existing masking models can be distinguished by the way they combine the information of the different channels. Models that consider only masking effects within a channel are called intra-channel, while models that account also for masking between different channels are called multi-channel masking models.

The activity masking, also called entropy masking, is another phenomenon that is closely linked to contrast masking. The basic idea is that a concentrated distortion signal is easily recognized in a smooth and homogeneous zone, while it is somehow hidden in an active region. The Fig.(12.17) shows this effect. In both images (a-b) the same distortion signal was introduced, but, while it is easy to see it in the top image, it is hidden in the bottom one. The local surround of the flowers (see Fig.(12.17(b))) is very "active" and the HVS needs more time to pick up the distortion. Due to its property, the activity masking is a concept of great relevance for image compression applications. The difference between models for contrast and activity masking lies in its spatial support. Both explain the reduced sensitivity by the presence of strong contrasts. However, contrast masking is typically applied in a point-wise fashion. That means that the contrast at position (x,y) indicates how large the error (for example the quantization error) may be at the same location, so much to not be perceptible. The activity masking considers an extended local surround for the prediction of the visibility of the distortion signal. Sometimes the two effects can no longer be separated and are combined in the same masking model.



(a)



(b)

Figure 12.17 : The local surround of the flowers in the image (b) is very active and the HVS needs more time to pick up the distortion.

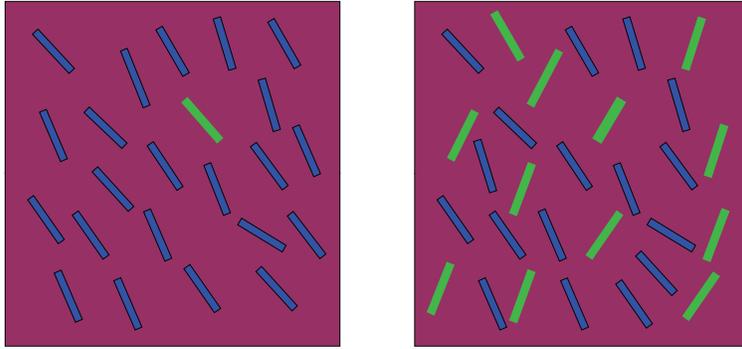


Figure 12.18 : Search arrays stimuli.

12.5.6 Visual Attention (Saliency)

Visual attention is the ability of HVS to filter out redundant visual information and to detect the most salient parts of the input visual stimulus [31]. This is a really complex process since the simultaneous identification of all the interesting targets in one's visual field involves many anatomical areas of the brain. The HVS capability to focus on the salient part of the observed image is supported by the fact that early stages of visual processing quickly distinct some particular stimuli from among other items or locations. Our brain is able to rapidly compute saliency in an automatic manner and in real-time over the entire visual field. The visual saliency is the capability to select a location sufficiently different from its surroundings to be worthy of your attention. Visual saliency is not a simple physical property of a visual system but is the consequence of an interaction of a stimulus with other stimuli, as well as with a visual system (biological or artificial). Below are reported two simple examples of so-called search arrays stimuli; these images contain many items, one of which should appear to the reader as highly visually salient.

In the left image there are many patterns all having the same direction and the same color, except the green one. The image is very simple and the HVS does not need to scan the picture to discern the unique green pattern from all the others called "distracters". The right image shows many blue and green bars, one of each is different from all the others. The visual attention mechanism, in this case, needs more time to individuate the unique bar called "conjunction-target" which is the only one green bar having the same direction of all blue bars. The essence of saliency lies in enhancing the neural and perceptual representation of locations whose visual statistics significantly differ from the broadly surrounding image statistics in some behaviorally relevant manner. This explains why for the first image in Fig.(12.18) the visual attention focuses on green bars immediately, this happens because all the other bars are totally different in color. Thus, a way for modeling the visual attention is to extract an image map which well represents the target for attention.

12.5.7 Pooling

The light stimulus is elaborated by HVS in many channels and its interpretation involves different areas of the brain. This multi-channel signal decoding mechanism can be emulated by collecting and combining each single channel response in a unique final response. This process is known as pooling. The human ability to handle all together the input signals and how it collects data in a final interpretation is not a totally known process. However, when a metric is planned, usually, after a first step oriented to extract some image distortion maps, is foreseen a final pooling step employed to combine the distortion maps into a single quality score. In this section the Minkowski summation [32], which is a mathematical formula widely used in literature to emulate the pooling step, is explained. In psychological tests, where the stimulus consists of two or more component sinusoidal gratings, the detectability of the compound stimulus is estimated by a nonlinear (weighted) summation of the detectability of its components. The summation rule known as Minkowski summation is expressed by the following:

$$S_c = \left(\sum_{i=1}^n S_i^m \right)^{1/m} \quad (12.16)$$

where S_c is the sensitivity for the compound stimulus, S_i is the sensitivity to each component stimulus, n is the number of components and m is the summing Minkowski exponent. Robson and Graham [33] showed that a Minkowski exponent of approximately 3.5 yielded the strongest predictions in a number of grating detection tasks. The useful applicability of Minkowski summation to predict thresholds for composite stimuli is clear, but the mechanism is only hypothesized. The summation rule has generally been interpreted as describing probability summation. The probability of detecting a composite stimulus (P_c) would be calculated from the probabilities of detecting the n components (P_i) independently.

$$P_c = 1 - \prod_{i=1}^n (1 - P_i) \quad (12.17)$$

Yet the question remains whether this straightforward but powerful model of detection processes can be extended to even more realistic viewing situations.

12.6 Perceptual Objective Metrics

Metrics which build a score emulating some psychophysical features of the HVS as the perception of blockiness, blurring, lack of dynamic range, loss of high frequencies etc., are referred as *perceptual objective metrics*. The first works present in literature propose model of some HVS behaviors based on formulas totally independent of subjective parameters coming from experiments. These first indexes are essentially based on some statistical properties which characterize the image. Today, the score formulated by *perceptual objective metrics* is typically based on parameters tuned by subjective experiments. In the following section some selected techniques which have provided significant

contributes to the image quality assessment are described. A brief description of one of the first proposed perceptual objective metric (UIQ) [3] and its evolutions (SSIM [1], LC-SSIM [12]) is reported. Also, one of the latest perceptual quality metrics [34], which is based on some of the HVS behaviors described in this chapter (CSF, masking, pooling etc.) is discussed.

12.6.1 Universal Index Quality

The universal quality index (UIQ) [3] is one of the first perceptual objective full reference metrics which tries to model some HVS properties by using simple mathematical formulas not subjectively tuned. The UIQ is based on the assumption that the HVS is highly sensitive to the image structural information, therefore, a measure of structural similarity should be a good approximation of the perceived image quality. To realize a "universal" metric, the authors designed the index assuming that any image distortion can be modeled by taking in consideration three factors: the loss of correlation, the luminance distortion and contrast distortion. The (12.18) shows the final UIQ score formula:

$$UIQ = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \cdot \frac{2\bar{x}\bar{y}}{(\bar{x})^2 + (\bar{y})^2} \cdot \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \quad (12.18)$$

In the equation (12.18) the x and the y are the original and the test image signals, while \bar{x} and \bar{y} are their mean values, σ_x , σ_y are the variances, and σ_{xy} is the covariance:

$$x = \{x_i \mid i = 1, 2, \dots, N\} \quad y = \{y_i \mid i = 1, 2, \dots, N\}$$

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad \bar{y} = \frac{1}{N} \sum_{i=1}^N y_i \quad (12.19)$$

$$\sigma_x^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \quad \sigma_y^2 = \frac{1}{N-1} \sum_{i=1}^N (y_i - \bar{y})^2$$

$$\sigma_{xy}^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$$

The first element in (12.18) measures the correlation degree between x and y . Its range in $[-1,1]$ and the best value is obtained when $\sigma_x = \sigma_y$. The second element, ranging in $[0,1]$, measures how similar are x and y in terms of luminance and it is equal to 1 only if $\bar{x} = \bar{y}$. The third element measures how similar are in terms of contrast the two images; in fact σ_x and σ_y can be considered an assessment of the contrast of x and y . Its range is defined in $[0,1]$ and its max value, equal to 1, is achieved if and only if $\sigma_x = \sigma_y$. The maximum value achieved by UIQ is equal to 1 and it is obtained when the two images,

x and y , are identical. The UIQ is a simple mathematical definition of structural image information and no human visual system model is explicitly employed. Also in (12.18) no subjectively tuned parameters are contemplated. Its consistency with subjective quality measurement is some time better than the widely used mean squared error, or the PSNR, but fails when tested on a wide range of impairments.

12.6.2 Quality Measure for Image Fusion

Since image signals are generally non-stationary, the authors of [13] suggest to measure the image quality index UIQ over local regions and then combine the different result into a single measure. This choice permits to weight the final score by taking into consideration the saliency information, which is a measure of the local window relevance, in terms of contrast, variance, entropy, etc. This metric is implemented for image fusion and is used to evaluate multi-resolution algorithms performance. In [13], authors work on local window w to compute saliencies $s(a|w)$ and $s(b|w)$ of the two input images (a and b). From local saliencies, the local weights for both images are then computed using the formula:

$$\lambda_a(w) = \frac{s(a|w)}{s(a|w) + s(b|w)} \quad (12.20)$$

The $\lambda_b(w)$ weights are computed similarly to (12.20), the weights (λ_a , λ_b) range in $[0,1]$. The quality score for the image fused f is defined as:

$$Q(a,b,f) = \frac{1}{|W|} \sum_{w \in W} (\lambda_a(w)Q_0(a,f|w) + \lambda_b(w)Q_0(b,f|w)) \quad (12.21)$$

It is evident that if, for example, λ_a is larger than λ_b , the image a will give a bigger contribution than b to the final score computation. The authors consider also the concept of salient information analyzing the entire image. So, a further coefficient $c(w)$ is introduced and the final score is give by the formula:

$$Q_w(a,b,f) = \sum_{w \in W} (c(w)\lambda_a(w)Q_0(a,f|w) + \lambda_b(w)Q_0(b,f|w)) \quad (12.22)$$

where:

$$c(w) = \frac{C(w)}{\sum_{w \in W} C(w')} \quad \text{and} \quad C(w) = \max(s(a|w), s(b|w)) \quad (12.23)$$

The $C(w)$ is the overall saliency of a window. The quality index has a dynamic range of $[-1,1]$. The closer the value is to 1, the higher is the quality of the composite image.

12.6.3 Structural Similarity Index Metric

An updated version of the UIQ is the Structural Similarity Index Measure (SSIM) [12] which is still a full FR based on the three components of the *UIQ* equation (12.18). The

SSIM has the following expression:

$$SSIM = \left[\frac{2\bar{x}\bar{y} + C_1}{(\bar{x})^2 + (\bar{y})^2 + C_1} \right]^\alpha \cdot \left[\frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \right]^\beta \cdot \left[\frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \right]^\gamma \quad (12.24)$$

The constant C_1 is calculated to avoid instability when $\bar{x}^2 + \bar{y}^2$ is very close to zero, it is calculated as $C_1 = (K_1L)^2$ where $K_1 \ll 1$ and L is the dynamic range of the pixel. The constant $C_2 = (K_2L)^2$ and $C_3 = (K_3L)^2$ are, as the C_1 , non negative small constant. The parameters α , β and γ are used to differently weight the three components (luminance, contrast and correlation). In case of $\alpha=\beta=\gamma=1$ then the SSIM corresponds to UIQ. The SSIM index ranges from [0,1], the closer is the SSIM to one the more similar is the test to the reference image.

12.6.4 Luminance and Contrast SSIM

In [12] a NR quality metric for the evaluation of image fusion algorithm's performances is proposed. It is based on the SSIM model and it handles the problem coming from the pixel dependency ignored by previous works as [13]. The problem of the leakage, according to which the spatial profile of a single pixel goes beyond borders of the cell in which it is conceptually localized, is an important issue to handle and, the assumption of pixel's independence oversimplifies the analysis related to the quality assessment. In [12] authors modify the second and the third elements of SSIM (12.24) introducing dependencies between pixels. The luminance pixel contribution to its neighbors has a Gaussian distribution centered in (i_0, j_0) :

$$\Theta(i, j) \propto \frac{1}{2\pi\sigma^2 e^{-\frac{(i-i_0)^2 + (j-j_0)^2}{2\sigma^2}}} \quad (12.25)$$

where σ varies from display to display. The global image luminance can be computed by summing each luminance relative to each local window (12.26).

$$L(i_0, j_0) = \sum_{i,j} l(i, j) \quad (12.26)$$

The contrast is estimated on a (3×3) block size window centered in (i_0, j_0) , and it is computed as the ratio between the difference of the local maximum and local minimum and the difference of the global maximum and global minimum (12.27).

$$C(i_0, j_0) = \frac{\max(block) - \min(block)}{\max(image) - \min(image)} \quad (12.27)$$

The luminance and contrast similarity between the source image a and the fused image f , are calculated as following:

$$l_n(i, j) = \frac{2L_a(i, j)L_f(i, j)}{C_a^2(i, j) + L_f^2(i, j)} \quad c_n(i, j) = \frac{2C_a(i, j)C_f(i, j)}{C_a^2(i, j) + C_f^2(i, j)} \quad (12.28)$$

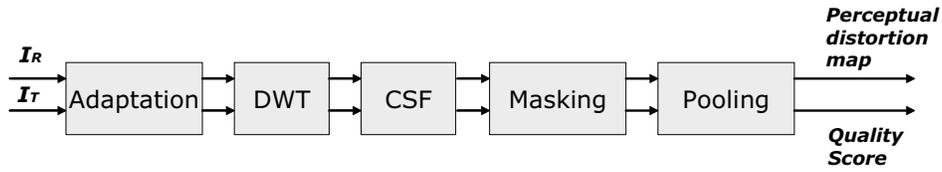


Figure 12.19 : Block-scheme of SLVM.

Substituting the l_n and the c_n in the equation (12.24), the SSIM formula becomes:

$$LCSSIM = [s(i, j)]^\alpha \cdot [l_n(i, j)]^\beta \cdot [c_n(i, j)]^\gamma \quad (12.29)$$

12.6.5 Wavelet Based Image Quality Metric

In latest years many perceptual objective metrics have been proposed, here is discussed a technique which focuses on the concept of semi-local masking which is quite different from the contrast masking. While the contrast masking regards the analysis of visibility dependence to the contrast value, the semi-local masking limits this analysis to the neighborhood characteristics. The semi-local masking is a more accurate model of masking which allows to analyze the visibility threshold behavior when very complex images, as natural ones, are treated. The method on [34] has been selected, first to discuss about the *semi-local* masking which is an accurate model of the contrast masking, and, second, to discuss about a quality index which is based on many of the HVS properties discussed in this chapter as the low-level perception, the spatial frequency sensitivity, the contrast masking and the semi-local masking and, finally, the pooling (Fig.(12.19)).

Firstly, the adaptation to human vision system process, which is activated when different levels of light hit the eye is taken into account. Then, a sub-band decomposition, by using the Discrete Wavelets Transform (DWT) (Cohen-Daubechies-Feauveau), is performed to evaluate the multi-channel HVS model. The spatial frequency sensitivity of the HVS is simulated using the contrast sensitivity function proposed by Daly. In particular, each coefficient $c(l, o)$ of the equation (12.30) is normalized by the CSF Daly's formula.

$$\bar{c}_{l,0}(m, n) = f(c_{l,0}(m, n) \cdot CSF) \quad (12.30)$$

In (12.30) l is the decomposition-level number and 0 is the orientation.

Authors test four masking functions:

1. Contrast masking by using Daly's model

$$T_{l,0}(m, n) = (1 + (k_1 \cdot (k_2 \cdot |\bar{c}_{l,0}(m, n)|)^s)^b)^{1/b} \quad (12.31)$$

2. Semi-local masking by modifying Daly's model

$$T_{l,0}(m, n) = (1 + (k_1 \cdot (k_2 \cdot |\bar{c}_{l,0}(m, n)|)^{s(m,n)})^b)^{1/b} \quad (12.32)$$

where

$$s(m, n) = S + \Delta s(m, n) \in [0.65; 1] \quad (12.33)$$

3. Intra-channel Nadenau's model

$$T_{l,0}(m, n) = \max(1, \bar{c}_{l,0}(m, n)^\epsilon) \quad (12.34)$$

4. Intra-channel Nadenau's model with semi-Local Masking

$$T_{l,0}(m, n) = \max(1, \bar{c}_{l,0}(m, n)^\epsilon) \cdot (1 + \omega_\Gamma) \quad (12.35)$$

where ω_Γ is the term which takes in consideration the influence of the neighborhood of the (m, n) position.

Being not the focus of the argument, the reader can find more details about the previous four formulas and about their specific parameters in [34].

The final step of pooling (Minkosky's formula) is performed on the data coming from the different decomposition levels of both the reference and the test images. Two are the pooling outputs: a perceptual distortion map and a final quality score. The performance of the four proposed metrics are evaluated by MOS obtained by conducting three subjective quality assessment experiments in normalized conditions as suggested by ITU-R BT 500.10. A psychometric function $f(Q)$ is used to transform the objective quality score Q in predicted MOS ($MOSp$) and, as recommended by the VQEG, the perceptual objective quality metrics are evaluated using three performance indexes: the linear correlation coefficient (CC), the Spearman Rank Order Correlation coefficient (SROCC), and the Root Mean Square Error (RMSE). For completeness, the authors performed also the PSNR and the SSIM. Final results highlighted that the perceptual multi-channel model outperforms the PSNR and the SSIM and, moreover, the use of the semi-local masking both for the Daly's and the Nadenau's formula, consistently increases the performance of the model in terms of CC, SROCC and RMSE. In conclusion, it can be observed that masking effect is a complex visual system phenomenon that should be accurately represented, and, as demonstrated by the discussed work, the semi-local masking has a really positive impact in image quality assessment.

12.7 Feature Trends and Conclusion

One of quality image assessment targets is to develop NR metrics. In latest years the trend is to plan NR indexes for measuring specific impairments, as ones related to the High Definition standard [7], other works regard the development of indexes to assess the robustness of digital watermarking systems [35]. With the increasing importance of the networks infrastructure, many works have faced the problem to develop automatic indexes for quality assessments of videos delivered over the network [36, 37]. Specifically, for evaluating the quality of compressed videos many works have been proposed [9, 38, 39]. Another trend, in quality assessment, regards the use of perceptual quality indexes for

tuning image processing algorithms or systems. For example, many studies try to improve the performance of image compression standard. These methods, defined "quality constrained quantization algorithms", optimize the quantization step by using the quality score provided by an objective perceptual metric [40]. Metrics to classify videos into groups have been proposed [41], other works regard the improvement of super-resolution algorithms [42]. The image quality assessment topic is object of interest and attention because it is considered a valid support to successfully promote image quality products and, more generally, it is a good instrument for delivering algorithms and systems able to satisfy customer expectations.

Bibliography

- [1] Z. Wang and A. C. Bovik, "Image quality assessment: from error visibility to structural similarity," *Transactions on Image Processing*, vol. 13, pp. 600–612, 2004.
- [2] I. Guarneri, M. Guarnera, A. Bosco, and G. Santoro, "A perceptual quality metric for color interpolated images," *SPIE proceedings on image quality and system performance. Conference*, vol. 5668, pp. 61–69, 2005.
- [3] Z. Wang and A. C. Bovik, "A universal image quality index," *Signal Processing Letters*, vol. 9, pp. 81–84, 2002.
- [4] T. M. Kusuma, H.-J. Zepernick, and M. Caldera, "On the development of a reduced-reference perceptual image quality metric," *International Conference on Multimedia Systems Communications (ICW)*, pp. 178–184, 2005.
- [5] Z. Wang and E. P. Simoncelli, "Reduced-reference image quality assessment using a wavelet-domain natural image statistic model," *Human Vision and Electronic Imaging (SPIE)*, pp. 149–159, 2005.
- [6] M. I. Iqbal, H. J. Zepernick, and U. Engelker, "Error sensitivity analysis for wireless JPEG2000 using perceptual quality metrics," in *International Conference on Signal Processing and Communication Systems*, pp. 1–9, 2008.
- [7] K. Christian, T. Oelbaum, and K. Diepold, "No reference video quality evaluation for high-definition video," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009.
- [8] S. Wang, H. R. Sheikh, and A. C. Bovik, "No-reference perceptual quality assessment of jpeg compressed images," *International Conference on Image Processing (ICIP)*, vol. 1, pp. 477–480, 2002.
- [9] A. Bhat, I. Richardson, and S. Kannangara, "A new perceptual quality metric for compressed video," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 933–936, 2009.
- [10] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi, "Perceptual blur and ringing metrics: Application to jpeg2000," *Signal Processing Image communication*, vol. 19, pp. 163–172, 2004.
- [11] S. Li, R. Hong, and X. Wu, "A no reference and reduced reference metric for detecting dropped video frames," *Video Processing and Quality Metrics for Consumer Electronics (VPQM)*, pp. 167–172, 2009.
- [12] S. Li, R. Hong, and X. Wu, "A novel similarity quality metric for image fusion," *International Conference on Audio, Language and Image Processing (ICALIP)*, vol. 9, pp. 156–160, 2008.

- [13] G. Piella and H. Heijmans, "A new quality metric for image fusion," in *International Conference on Image Processing (ICIP)*, pp. 173–176, 2003.
- [14] M. Jian, P. Ma, and J. Jia, "A perceptual quality metric for performance evaluation of image fusion," *Intelligent Information Technology and Security Informatics (IITSI)*, pp. 148–152, 2009.
- [15] Video Quality Expert Group, "Methodology for the subjective assessment of the quality of television pictures," *ITU-R BT.500-11*, www.itu.int, 2002.
- [16] I. T. U. (ITU), "<http://www.itu.int>."
- [17] Video Quality Expert Group, "Final report from the video quality expert group on the validation of objective models of video quality assessment, phase II," *ITU-R Recommendation*, www.itu.int, 2003.
- [18] Video Quality Expert Group, "<http://www.its.bldrdoc.gov/vqeg/>."
- [19] I. Guarneri, M. Guarnera, A. Bosco, and F. Vella, "A quality assessment metric based on perceptual HVS behaviors," *Advanced Concepts for Intelligent Vision Systems (ACIVS)*, 2004.
- [20] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, edition 3 ed., 2008.
- [21] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it?," *IEEE Signal Processing Magazine*, pp. 98–117, 2009.
- [22] S. Daly, *Digital images and human vision*, ch. The visible differences predictor: an algorithm for the assessment of image fidelity. MIT Press, 1993.
- [23] J. L. Mannos and D. J. Sakrison, "The effect of visual fidelity criterion on the encoding of images," *IEEE Transactions on Information Theory*, vol. 20, pp. 525–536, 1973.
- [24] L. T. Sharpe, A. Stockman, H. Jgle, H. Knau, G. Klausen, A. Reitner, and J. Nathans, "Red, green, and red-green hybrid photopigments in the human retina: correlations between deduced protein sequences and psychophysically-measured spectral sensitivities," *Journal of Neuroscience*, vol. 18, pp. 0053–10069, 1998.
- [25] P. L. Callet, A. Saadane, and D. Barba, "Interactions of chromatic components on the perceptual quantization of the achromatic component," *Human Vision and Electronic Imaging Conference*, vol. 3644, pp. 121–128, 1999.
- [26] P. L. Callet, A. Saadane, and D. Barba, "Orientation selectivity of opponent-colour channels," *European Conference on Visual Perception (ECVP)*, vol. 28, 1999.

- [27] X. Yanfang, W. Yu, and Z. Ming, "Color reproduction quality metric on printing images based on the S-CIELAB model," *Computer Science and Software Engineering (CSSE)*, pp. 294–297, 2008.
- [28] F. W. Campbell and J. G. Robson, "Application of Fourier analysis to the visibility of gratings," *Journal of Physiology*, vol. 197, pp. 551–566, 1968.
- [29] H. R. Blackwell, "Contrast thresholds of human eye," *Journal of Optical Society of America*, vol. 11, 1946.
- [30] F. W. Campbell and J. G. Robson, "A model of visual adaptation for realistic image synthesis," *Computer Graphics*, pp. 249–258, 1996.
- [31] A. Ninassi, O. L. Meur, P. L. Callet, and D. Barbba, "Does where you gaze on an image affect your perception of quality? applying visual attention to image quality metric," *International Conference on Image Processing (ICIP)*, vol. 2, pp. 169–172, 2007.
- [32] M. To, P. G. Lovell, T. Troscianko, and D. Tolhurst, "Minkowski summation of cues in complex visual discriminations using natural scene stimuli," *Journal of Vision*, vol. 7, p. 968a, 2007.
- [33] J. G. Robson and N. Graham, "Probability summation and regional variation in contrast sensitivity across the visual field," *Vision Research*, vol. 21, p. 409418, 1981.
- [34] A. Ninassi, O. L. Meur, P. L. Callet, and D. Barba, "Which semi-local visual masking model for wavelet based image quality metric?," in *International Conference on Image Processing (ICIP)*, pp. 1180–1183, 2008.
- [35] B. Giulia, V. Conotter, F. D. Natale, and C. Fontanari, "Watermarking robustness evaluation based on perceptual quality via genetic algorithms," in *IEEE Transaction on Information Forensics and Security*, 2009.
- [36] P. Brun, G. Hauske, and T. Stockhammer, "Subjective assessment of H.264/AVC video for low-bitrate multimedia messaging services," in *ICIP*, pp. 1145–1148, 2004.
- [37] G. Hauske, T. Stockhammer, and R. Hofmaier, "Subjective image quality of low-rate and low-resolution video sequences," in *MOMUC*, 2003.
- [38] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, "Study of subjective and objective quality assessment of video," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1427–1441, 2010.
- [39] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, "A subjective study to evaluate video quality assessment algorithms," *SPIE Proceedings Human Vision and Electronic Imaging XV*, vol. 7527, 2010.

- [40] J. Hu, S. Choudhury, and J. Gibson, "Video capacity of WLANs with a multiuser perceptual quality constraint," *IEEE Transaction on Multimedia*, vol. 10, 2008.
- [41] A. Khan, L. Sun, and E. Ifeachor, "Content clustering-based video quality prediction model for mpeg4 video streaming over wireless networks," *IEEE International Conference on Communications (ICC)*, vol. 28, 2009.
- [42] F. Liu, J. Wang, S. Zhu, M. Gleicher, and Y. Gong, "Visual-quality optimizing super resolution," *Computer Graphics Forum*, 2009.

Beyond Embedded Device

S. Battiato

Image Processing Lab, University of Catania, Italy.

A. Castorina, G. Messina

Advanced System Technology - Catania Lab, STMicroelectronics, Italy.

Abstract: In the recent years, technology advances in sensors and image processing have allowed significant improvements that encompass not only quality, but also image content analysis and understanding. Consumer devices are now being equipped with sophisticated algorithms allowing complex tasks such as face recognition, smile detection, automatic red eye removal, etc. In addition, topics related to image forensics oriented to verify the authenticity of digital images, such as camera identification and image manipulation detection are now available. This Chapter provides an overview of the main topics representing a challenge to innovation both from hardware and software point of view: Super Resolution, Temporal Demosaicing, Bracketing or High Dynamic Range Reconstruction. Also some aspects of Computational Photography and Forensics Camera Identification are briefly described. These topics represent the current trends and novel solutions for the next generation of imaging devices.

13.1 Super Resolution

Quality improvement is obtained by increasing the resolution of the sensor and/or by using more sophisticated image-processing algorithms. But for specific application (e.g., mobile imaging) there are too many constraints to manage in terms of power consumption, DSP capabilities, etc. Best quality image acquisition with low cost is thus necessary. The main objective is to obtain from a low cost image sensor a High-Resolution (HR) image in a relatively short time. Zooming algorithms usually interpolate "new artificial" intra-pixel information to expand the images resolution. Simple interpolation is not well suited to generate resolution-enhanced images. Super Resolution techniques aim to insert "real" intra-pixel information to obtain the true matching scene.

There are different mathematical approaches to solve this problem [1]. Almost all of them merge in a proper way non-redundant image information combining multiple frames. Obviously, if successive frames are exact copies, no new information is available. It is important to treat properly each particular feature (e.g., object boundaries that will not be at the same pixel locations in successive frames). In this way it is possible to extract more information with respect to the original data available in each single Low-Resolution (LR) image.

One of the LR input images is considered as reference frame; this is named the primary image and typically is the central image of the sequence. The reference frame is expanded in a high-resolution grid (e.g., a 640x480 pixel LR image is expanded to a 1280x960 pixel image). The complementary pixels, introduced by the expansion, could be derived using classical interpolation approaches (Bilinear, Bicubic), without changing

the effective "resolution". A more accurate HR image can be obtained properly using the data from the other LR frames.

Classical Super Resolution approaches are mainly based on frequency domain, bayesian methods and reconstruction methods.

13.1.1 Frequency Domain Methods

One of the first frequency domain based algorithm was presented by Tsai and Huang [2]. Such method is based on three fundamental principles:

- The shifting property of the Fourier Transform (FT);
- The aliasing relationship between the Continuous Fourier Transform (CFT) and the Discrete Fourier Transform (DFT);
- The original scene is band-limited.

Unlike all other methods, the data are first transformed to the frequency domain where they are combined together (see Fig.(13.1)). This data are then transformed back into spatial domain where the new image will have a higher resolution than the original frames. The frequency domain methods do not reach good results for real image sequences and moreover these methods are computationally expensive for embedded imaging devices.

Recently, some improvements have been introduced, taking into account a non-band-limited stationary spectral model to reconstruct the HR image [3].

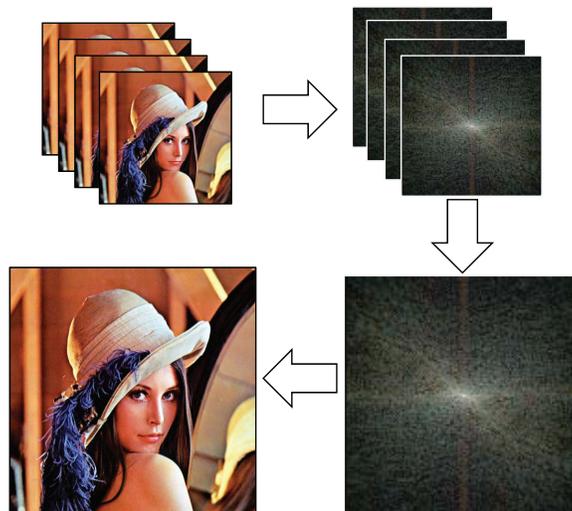


Figure 13.1 : Classical frequency domain Super Resolution approach.

13.1.2 Statistical Methods

Bayesian methods for HR image reconstruction use a statistical *a priori* model combined with maximum *a posteriori* (MAP) or maximum likelihood (ML) formulation for HR reconstruction. Bayes' theorem gives the rule for updating belief in a hypothesis A (i.e., the probability of A) given additional evidence B , and background information (context) c :

$$p(A/B, c) = \frac{p(A/c) * p(B/A, c)}{p(B/c)} \quad (13.1)$$

An example of Bayesian approach is given in [4]. NASA researchers use Bayesian theory for constructing super-resolved surface models by combining information from a set of LR images. The basic idea behind NASA approach is based on inverse graphics. That is, if it's known what the ground is like, the lighting conditions, the camera orientation and characteristics, etc., then it is possible to predict what the camera would see (an image). However, the problem is here inverted: it is known what the images are, and it must be found the most probable ground truth (surface) that would have generated them, assuming that the lighting conditions and camera characteristics are known. The most important (and difficult) part of this process is the recovering of the camera orientation and position for each image. To do this, all the images must be registered with respect to each other by accuracy of small fraction of pixel; this registration clarifies how an image map is built onto the ground truth model. The initial ground model is formed by consenting to each pixel "to vote" on how much that ground position contributed to that pixel. This initial ground model is then used to project what each image should be (i.e., to predict each pixel value). The difference between the predicted pixel value and the observed value is used to update the ground model until it cannot be further improved. This procedure increases both spatial and luminance resolution.

13.1.3 Reconstruction Methods

In the methods of this class, both observation model and reconstruction are accomplished in the spatial domain. These approaches use a model relating the LR images to the desired HR image by iterative reconstruction techniques useful to estimate the HR image. The main advantage of this approach is the ability of modeling more realistic video formulation process and the flexibility of applying different iterative methods to estimate HR image. Below some of the most popular approaches are described.

Projection Onto Convex Sets: The POCS [5,6] method requires the definition of closed convex constraint sets within a well-defined vector space that contains the actual HR image. An estimation of the HR image is then defined as a point in the intersection of these constraint sets, and is determined by successive projecting an arbitrary initial assessment onto the constraint sets.

A projection operator P is associated with each constraint set, mapping an arbitrary point within the space to the closest point within the set. The *Convex Sets* are defined in

the following way:

$$C_{t_r}(m_1, m_2, k) = \left\{ y(n_1, n_2, t_r) : |r^{(y)}(m_1, m_2, k)| \leq \delta_0(m_1, m_2, k) \right\} \quad (13.2)$$

For each pixel within the LR image sequence $g(m_1, m_2, k)$, where k is the fixed frame number at time t_r and

$$r^{(y)}(m_1, m_2, k) = g(m_1, m_2, k) - \sum_{(n_1, n_2)} y(n_1, n_2, t_r) \cdot h_{t_r}(n_1, n_2, m_1, m_2, k) \quad (13.3)$$

is the residual associated with an arbitrary member y of the constraint set. These sets are referred to as the *data consistency constraint sets*. Sets $C_{t_r}(m_1, m_2, k)$ can be defined only where the motion information is accurate. Therefore it is simple to incorporate occlusion and uncovered background knowledge by only defining sets for appropriate observations. The quantity $\delta_0(m_1, m_2, k)$ is a bound reflecting the statistical confidence.

Once *Convex Sets* are defined, for an arbitrary high resolution image, the *Projection operator* is constructed as follow:

$$Ptr(m_1, m_2, k)[y(n_1, n_2, t_r)] = y(n_1, n_2, t_r) + \begin{cases} \frac{(r^{(y)}(\star) - \delta_0(\star))h_{t_r}(n_1, n_2, \star)}{\sum_{o_1} \sum_{o_2} h_{t_r}^2(o_1, o_2, \star)}, & r^{(y)}(\star) > \delta_0(\star). \\ \frac{(r^{(y)}(\star) + \delta_0(\star))h_{t_r}(n_1, n_2, \star)}{\sum_{o_1} \sum_{o_2} h_{t_r}^2(o_1, o_2, \star)}, & r^{(y)}(\star) < -\delta_0(\star) \\ 0, & \text{else} \end{cases} \quad (13.4)$$

where “ \star ” function argument is interpreted as “ m_1, m_2, k ”, o_1 and o_2 denote the h_{t_r} blur function support in $y(n_1, n_2, t_r)$ referring to lower image frame k at pixel (m_1, m_2) . Additional constraint C_A (bounded amplitude and positivity) can be utilized to improve the results.

Given the above projections, an estimate, $\tilde{f}(n_1, n_2, t_r)$, of the HR image $f(n_1, n_2, t_r)$, is obtained iteratively from all LR images $g(m_1, m_2, k)$:

$$\tilde{f}_{l+1}(n_1, n_2, t_r) = T_A \tilde{T}[\tilde{f}_l(n_1, n_2, t_r)] \quad \text{for } l = 0, 1, 2, \dots \quad (13.5)$$

where T_A is the relaxed projection operator of C_A and \tilde{T} denotes the composition of the relaxed projection operators onto the family of sets $C_{t_r}(m_1, m_2, k)$. The initial estimation $\tilde{f}_0(n_1, n_2, t_r)$ is obtained by interpolating the LR images, through a bilinear approach, to HR image size, taking also into account the motion compensation among the LR images. In theory, iterations continue until an assessment lies within the intersection of all the constraint sets. In practice, however, iterations are generally terminated according to a certain stopping criterion such as visual inspection of the image quality. A visual example of a POCS Super Resolution result is shown in Fig.(13.2).

Simple Back-Projection This iterative process was introduced by Michal Irani and Samuel Peleg [7, 8]. This technique uses averaged projections in HR grid to iteratively



(a)



(b)

Figure 13.2 : Example of Super Resolution POCS based: (a) represents one of the low resolution frames (upsampled by using a simple nearest-neighbor approach) and (b) the resulting Super Resolution frame.

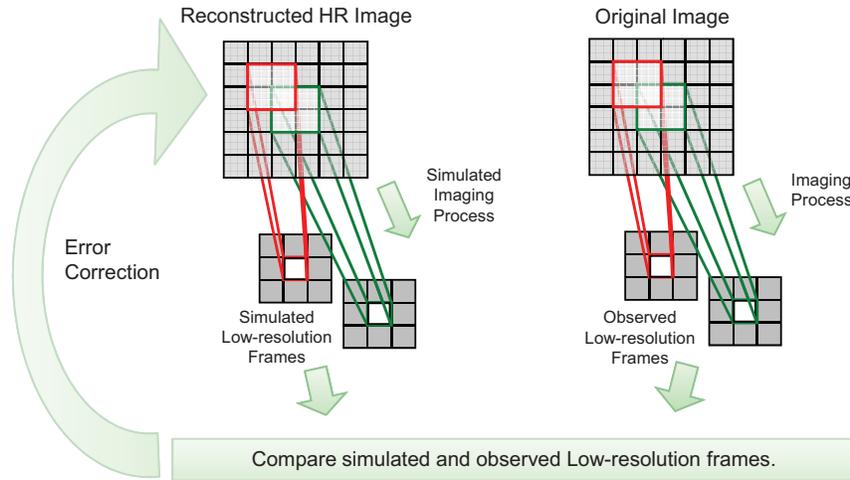


Figure 13.3 : Standard diagram of a Back-Projection approach.

estimate the HR image. This algorithm considers also translational and rotational motion among LR frames.

The approach of back projection method is based on the comparison between the known LR frames and the simulated LR frames, generated from the computed HR image (see Fig.(13.3)). The low-resolution pixel is a “*projection*”, in HR grid, of a region in the scene whose size is determined by the imaging blur.

The imaging process, yielding the observed image sequence $\{g_k\}$ is modeled by:

$$g_k(m,n) = \sigma_k(h(f(x,y)) + \eta_k(x,y)) \quad (13.6)$$

where g_k is the k -th observed image frame, f is the original scene, h is the blurring operator, η_k is an additive noise term, σ_k is a non-linear function that digitizes and decimates the image into pixels and quantizes the resulting pixels values from intensities into gray levels, σ_k also includes the displacement from the k -th frame, (x,y) is the center in the receptive field (in f) of the detector whose output is $g_k(m,n)$.

Starting from an initial guess $f^{(0)}$ of the HR image, the imaging process is simulated to obtain a set of LR images $\{g_k^{(0)}\}$ corresponding to the observed input image $\{g_k\}$. If $f^{(0)}$ is the correct high-resolution image, then simulated LR frames $\{g_k^{(0)}\}$ must be identical to the observed $\{g_k\}$. The difference images $\{g_k - g_k^{(0)}\}$ are then computed and used to improve the initial guess by “back-projecting” each value of the difference images onto its receptive field in $f^{(0)}$. The process is repeated iteratively to minimize the error function:

$$e^{(n)} = \sqrt{\sum_k \sum_{(x,y)} \left(g_k(x,y) - g_k^{(n)}(x,y) \right)^2} \quad (13.7)$$

The imaging process of g_k at the n -th iteration is simulated by:

$$g_k^{(n)} = T_k(f^{(n)} * h) \downarrow s \quad (13.8)$$

where T_k represents the degradation model, $\downarrow s$ denotes a downsampling operation by a factor s , and $*$ is the convolution operator. The iterative update scheme of the high-resolution images is expressed by:

$$f^{(n+1)} = f^{(n)} + \frac{1}{K} \sum_{k=1}^K T_k^{-1}(((g_k - g_k^{(n)}) \uparrow s) * h_{BP}) \quad (13.9)$$

where K is the number of LR frames, $\uparrow s$ is an upsampling operation by factor s , and h_{BP} is a “back-projection” kernel, determined by h and T_k . The mean value taken in this last equation reduces additive noise. More details are available in [9].

This technique is suitable for still picture Super Resolution. In other words, if no objects are moving into the scene and the frames represents the same scene with different translations and rotations. An example, which describes the increased resolution obtained from a sequence of LR images, heavily compressed, is shown in Fig.(13.4).

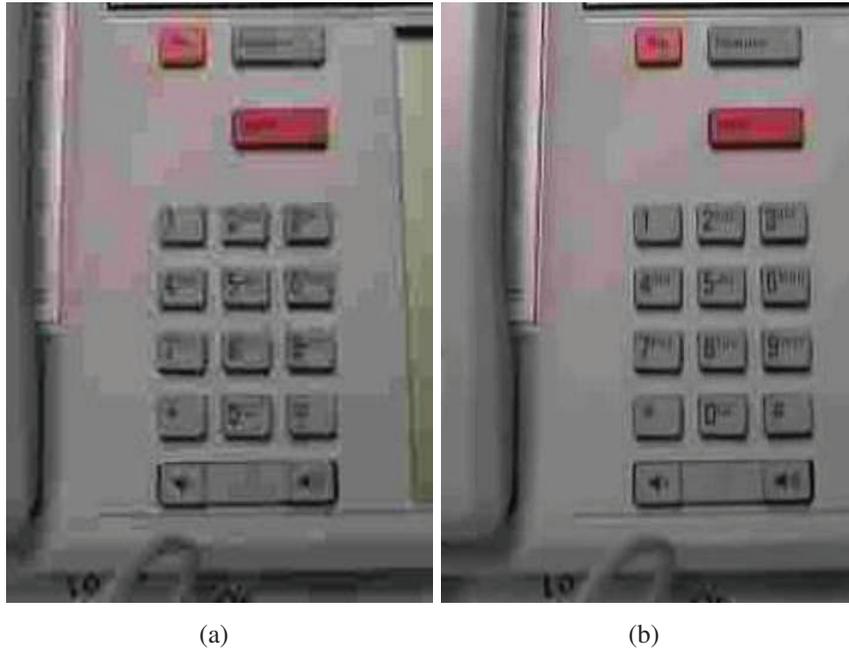


Figure 13.4 : Example of an image of a low resolution sequence (a) captured at a very low bit rate (here upsampled by using a simple nearest-neighbor approach to match the same resolution) and combined into a Super Resolution frame (b) using the simple back projection approach combined with a global motion estimator.

13.1.4 Temporal Demosaicing

As described in Chapter 7 the demosaicing is the process of color interpolation, which permits to reconstruct the missing RGB values that have not been captured by imaging sensor (i.e., usually acquires only Bayer CFA data). The temporal demosaicing is in a similar way a process of color interpolation, through spatial correlations, but also through temporal correlations coming from multiple frames. By taking into account multiple frames, to generate a single image, the temporal demosaicing is a particular case of Super Resolution.

There is a growing interest in multi-frame demosaicing, mainly because it can permit to increase the effective information (i.e., augmenting in some sense the sampling characteristic) of the device, without increasing its effective resolution. This means also less noise and more luminance sensitivity.

An interesting work from Farsiu *et al.* [10], has explained the reason of applying simultaneously demosaicing and Super Resolution. The main problems of performing Super Resolution on Bayer data are due to lacking values, the LR data missing from the merged HR Bayer pattern. Furthermore the problem of achieving by first demosaicing and then Super Resolution is clearly due to loss of real information from original Bayer data. The authors by first estimate the following image formation model (shown in Fig.(13.5)):

$$Y_i(k) = A_i(k)D_i(k)H(k)F(k)X_i + V_i(k) \quad k = 1, \dots, N \quad (13.10)$$

where $i \in R, G, B$, k represents the number of the LR frame, X_i is the intensity distribution of the scene, V_i is the additive noise, Y_i is the resulting color filtered LR image and the operators F , H , D and A are respectively the warping, blurring, down-sampling and color-filtering processes. The authors use a Maximum A Posteriori (MAP) model to reconstruct the HR demosaiced image. In details, they apply the global assumption about the correlation between color channels and spatial correlation, by using the following penalty function:

$$\Omega(X) = J_1(X, Y(k)) + P_1(X) + P_2(X) + P_3(X) \quad (13.11)$$

which means that the estimation Ω of the HR image X is obtained by minimizing a cost function of the Data Fidelity Term (J) and three Prior Information Terms P_1 (spatial luminance penalty term), P_2 (spatial chrominance penalty term) and P_3 (inter-color dependencies penalty term). With the Steepest Descent (SD) method, the $n + 1$ th estimate is obtained by updating the previous one utilizing the derivative of $\Omega(X)$. For further details see [10].

A simplest approach to temporal demosaicing has been presented by Wau and Zhang [11]. The authors estimate the motion by working on frames that have been already reconstructed by an intra-frame demosaicing method (on the G channel). Then a temporal enhancement is achieved on the resulting G channel by using the following weighting function:

$$\hat{G} = \sum_{i=0}^K w_i \tilde{G} \quad (13.12)$$



Figure 13.5 : The image formation model described by [10].

where \hat{G} is the estimate of G and $\tilde{G} = G + e_i$ for $i \in 0, \dots, K$ represents the sum of the unknown value G plus the error e_i . The weight used in the estimation is identified by minimizing a computed weighting function (see [11] for details). Once the Green channel has been completely estimated the Red and Blue channel are spatially demosaiced and thus a similar temporal enhancement is achieved on these channels.

13.1.5 Single Frame Super Resolution

Three complementary ways exist for increasing an image's apparent resolution:

- Aggregating from multiple frames. Extracting a single HR image from a sequence of LR frames adds value and is referred as multiple frame Super Resolution (as described in the previous Sections).
- Sharpening by amplifying existing image details. This is the change in the spatial frequency amplitude spectrum of an image associated with image sharpening. Existing high frequencies in the image are amplified. The main issues of such approach are the presence of ringing along edges and amplification of noise.
- Single-frame Super Resolution. The goal of such approach is to estimate missing high resolution details that are not present in the original image, and which can not make visible by simple sharpening.

Different approaches are present in literature [12–14] and are focused on specific field. For instance Backer and Kanade [14] focus their approach on "Hallucinating Faces", this is the specific terms they have used to illustrated their single image Super Resolution for face enlargement. In practice they use a dataset of registered and resampled faces, and through a super-resolution algorithm that uses the information contained in a collection of recognition decisions (in addition to the reconstruction constraints) they create an high resolution face. In effect such approach works only if the image that must be reconstructed is of the same nature of the information contained into the recognition decisions (see Fig.(13.6)).

Freeman *et al.* [12] use a training set of a not fixed nature to achieve their example-based Super Resolution. Such approach is characterized by a first training step which define rules to associate low resolution patches to high resolution ones. To perform such

step they use the bandpass and contrast normalized version of the low resolution and the high resolution images respectively. Such patches are stored as a dataset of rules and processed through a Markov network. The resulting images seem to be sharper, also if they do not reconstruct exactly high resolution details. This approach is less dependent to the nature of the image that must be super resolved, but the dimension of the dataset of patches could be an issue.

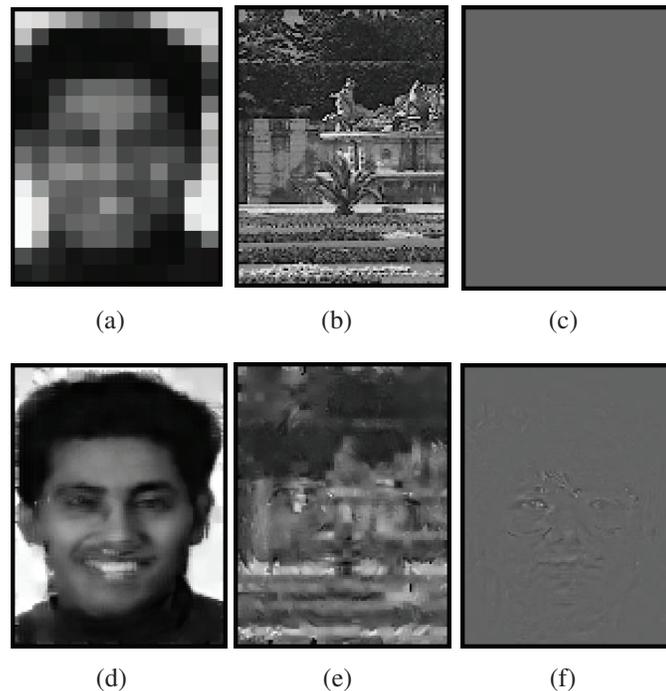


Figure 13.6 : Hallucination algorithm: **(a)** a face image of 12x16 pixels and **(d)** the resulting hallucination output; **(b)** original HR image not containing faces and **(e)** the resulting hallucinated image from its LR version; similar result for a neutral grayscale image **(c)** and **(f)** the resulting hallucinated output. As it is clearly visible a face is hallucinated by the algorithm even when none is present, hence the term "hallucination algorithm".

The more recent method presented by Glasner *et al.* [13] propose an unified framework for combining example-based and classical multi-image Super Resolution. This approach can be applied to obtain Super Resolution from a single image, with no database or prior examples. The work is based on the observation that patches in a natural image tend to redundantly recur many times inside the image, both within the same scale, as well as across different scales. Recurrence of patches within the same image scale (at subpixel misalignments) gives rise to the classical super-resolution, whereas recurrence of patches across different scales of the same image gives rise to example-based super-resolution.

The increasing interest of commercial Super Resolution solutions for HD television has allowed to develop furthermore such approaches. Some companies have manufactured hardware solutions to overcome the problem of compatibilities between old low resolution broadcasting and new full HD televisions (i.e., NEC Electronics [15]).

13.2 Bracketing and Advanced Applications

In order to attempt to recover or enhance a badly exposed image, even if some kind of post-processing is possible, there are situations where this strategy is not possible or leads to poor results. The problem comes from the fact that badly captured data can be enhanced, but if no data exists at all there's nothing to enhance. Today, despite the great advancements realized by digital photography, which has made available tremendous resolution even for mass market oriented products, almost all digital photo-cameras still deal with limited dynamic range and inadequate data representation, which make critical lighting situations, and the real world has tons of them, difficult to handle. This is where multiple exposure capture stands as a useful alternative to overpass actual technology limits. Even if the idea of combining multiple exposed data is just recently receiving great attention, the methodology itself is very old. In the early sixties, well before the advent of digital image processing Charles Wyckoff [16] was able to capture high dynamic range images by using photographic emulsion layers of different sensitivity to light. The information coming from each layer was printed on paper using different colors, thus obtaining a pseudo-color image depiction.

13.2.1 The Sensor Versus The World

Table 13.1 : Typical world luminance levels.

Scene	Illumination
<i>Starlight</i>	$10^{-3} cd/m^2$
<i>Moonlight</i>	$10^{-1} cd/m^2$
<i>Indoorlight</i>	$10^2 cd/m^2$
<i>Sunlight</i>	$10^5 cd/m^2$

Dynamic range refers to the ratio of the highest and lowest sensed level of light. For example, a scene where the quantity of light ranges from $1000 cd/m^2$ to $0.01 cd/m^2$, has a dynamic range of $1000/0.01=100,000$. The simultaneous presence in real world scenes poses great challenges on image capturing devices, where usually the available dynamic range is not capable to cope with that coming from the outside world. High dynamic range scenes are not uncommon; imagine a room with a sunlit window, environments presenting opaque and specular objects and so on. Table 13.1 shows typical luminance values for different scenes, spanning a very wide range from starlight to sunlight. On the other side dynamic range (*DR*) of an imaging device is defined as the ratio between the maximum charge that the sensor can collect (*Full Well Capacity, FWC*), and the minimum charge that is just above sensor noise (*Noise Floor, NF*).

$$DR = \log_{10} \left[\frac{FWC}{NF} \right]. \quad (13.13)$$

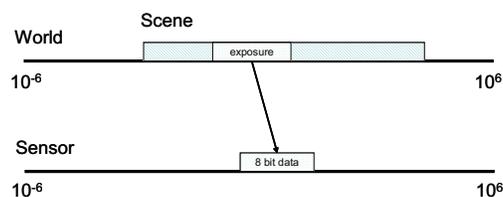


Figure 13.7 : Due to limited camera dynamic range, only a portion, depending of exposure settings, of the scene can be captured and digitized.

DR values are usually expressed in logarithmic units. This dynamic range, which is seldom in the same order of magnitude of those coming from real world scenes, is further affected by errors coming from analogue to digital conversion (*ADC*) of sensed light values. Once the light values are captured, they are properly quantized to produce digital codes, that usually for common 8-bit data fall in the $[0 : 255]$ range. This means that a sampled, coarse representation of the continuously varying light values is produced.

Limited dynamic range and quantization thus irremediably lead to loss of information and to inadequate data representation. This process is synthetically shown in Fig.(13.7), where the dynamic range of a scene is converted to the digital data of an imaging device: only part of the original range is captured, the remaining part is lost. The portion of the dynamic range where the loss occurs depends on employed exposure settings. Low exposure settings, by preventing information loss due to saturation of highlights, allow to capture highlight values, but lower values will be easily overridden by sensor noise. On the other side, high exposures settings allow a good representation of low light values, but the higher portion of the scene will be saturated. Once again a graphical representation gives a good explanation of the different scenarios.

Fig.(13.8(a)) shows a high exposure capture. Only the portion of the scene under the green area is sensed with a very fine quantization (for simplicity only 8 quantization levels, shown with dotted lines, are supposed), the other portion of the scene is lost due to saturation which happens at the luminance level corresponding to the end of the green area. Fig.(13.8(b)) shows a low exposure capture. This time since saturation, which happens at the light level corresponding to the end of the red area, is less severe due to low exposure settings and apparently all the scene is captured (the red area). Unfortunately, due to very widely spanned sampling intervals, quality of captured data is damaged by quantization noise and errors. To bring together data captured by different exposure settings allows to cover a wider range, and reveal more details than those that would have been possible by a single shot.

The process is usually conveyed by different steps:

1. camera response function estimation;
2. high dynamic range construction;
3. tone mapping to display or print medium.

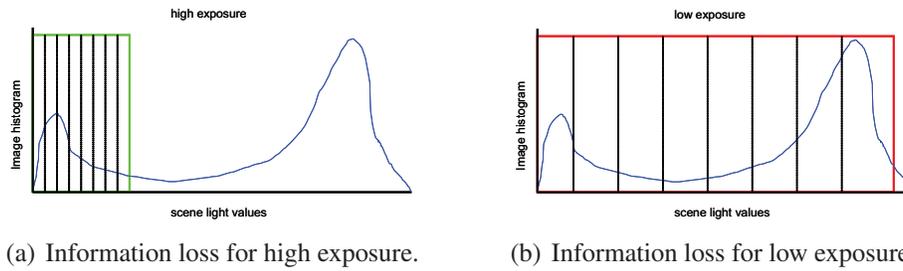


Figure 13.8 : Information loss for high and low exposure. In case of high exposure (a), a limited dynamic range is captured due to saturation. In case of low exposure (b), the captured data is coarsely quantized. For simplicity, only eight quantization levels are considered.

13.2.2 Camera Response Function

In order to properly compose a high dynamic range image, using information coming from multiple low dynamic range (*LDR*) images, the camera response function must be known. This function describes the way the camera reacts to changes in exposures, thus providing digital measurements.

Camera exposure X , which is the quantity of light accumulated by the sensor in a given time, can be defined as follows:

$$X = It \quad (13.14)$$

where I is the irradiance and t the integration time.

When a pixel value Z is produced, it is known that it comes from some scene radiance I sensed for a given time t , mapped into the digital domain through some function f . Even if most CCD and CMOS sensors are designed to produce electric charges that are strictly proportional to the incoming amount of light (up to the near saturation point, where values are likely to fluctuate), the final mapping is seldom linear. Nonlinearities can come from the *ADC* stage, sensor noise, gamma mapping and specific processing introduced by the manufacturer. In fact often DSC camera have a built-in nonlinear mapping to mimic a film-like response, which usually produces more appealing images when viewed on low dynamic displays.

The full pipeline, from the scene to the final pixel values is shown in Fig.(13.9) where prominent nonlinearities can be introduced in the final, generally unknown, processing.

The most obvious solution to estimate the camera response function, is to use a picture of uniformly lit different patches, such as the Macbeth Chart [17] and establish the relationship between known light values and recorded digital pixel codes. However this process requires expensive and controlled environment and equipment. This is why several chartless techniques have been investigated. One of the most flexible algorithms has been described in [18], which only requires an estimation of exposure ratios between the input images. Of course, exposure ratios can be derived from exposure times. Given N digitized *LDR* pictures, representing the same scene and acquired with timings $t_j : j = 1, \dots, N$,

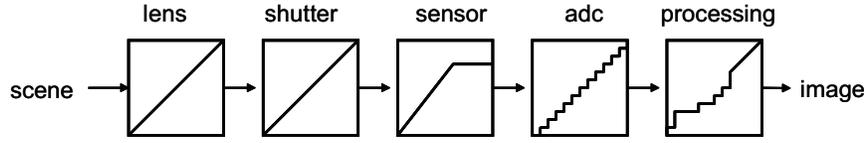


Figure 13.9 : The full pipeline from scene to final digital image. The main problem behind assembling the high dynamic range from multiple exposures, lies in recovering the function synthesizing the full process.

exposure ratios $R_{j,j+1}$ can be easily described as

$$R_{j,j+1} = \frac{t_j}{t_{j+1}}. \quad (13.15)$$

Thus, the following equation relates the i th pixel of the j th image, $Z_{i,j}$, to the underlying unknown radiance value I_i

$$Z_{i,j} = f(I_i t_j) \quad (13.16)$$

which is the aforementioned camera response function. The principle of high dynamic range compositing is the estimation for each pixel, of the radiance values behind it, in order to obtain a better and more faithful description of the scene that has originated the images. This means that we are interested in finding the inverse of (13.15): a mapping from pixel value to radiance value is needed:

$$g(Z_{i,j}) = f^{-1}(Z_{i,j}) = I_i t_j \quad (13.17)$$

The nature of the function g is unknown, the only assumption is that it must be monotonically increasing. That's why a polynomial function of order K is supposed.

$$I_i = g(Z) = \sum_{k=0}^K c_k Z^k \quad (13.18)$$

The problem thus becomes the estimation of the order K and the coefficients c_k appearing in (13.18). If the ratios between successive image pairs ($j, j+1$) are known, the following relation holds:

$$\frac{I_i t_j}{I_i t_{j+1}} = \frac{g(Z_{i,j})}{g(Z_{i,j+1})} = R_{j,j+1} \quad (13.19)$$

Using (13.19), parameters are estimated by minimizing the following objective function:

$$O = \sum_{j=1}^N \sum_{i=1}^P \left[\sum_{k=0}^K c_k Z_{i,j}^k - R_{j,j+1} \sum_{k=0}^K c_k Z_{i,j+1}^k \right]^2 \quad (13.20)$$

where N is the number of images and P the number of pixels. The system can be easily solved by using the least square method. The condition $g(1) = 1$ is enforced to fix the

scale of the solution, and different K orders are tested. The K value that better minimizes the system is retained.

To limit the number of equations to be considered not all pixels of the images should be used and some kind of selection is advised by respecting the following rules:

1. pixels should be well spatially distributed;
2. pixels should sample the input range;
3. pixels should be picked from low variance (homogenous) areas.

A different approach for feeding the linear system in (13.20) could be done by replacing pixel values correspondences by *comparagram* pairs. *Comparagrams* have been well described in [19] and provide an easy way to represent how pixels of one image are mapped to the same image with different exposure. This mapping is usually called brightness transfer function (*BTF*).

It's worth noting that if direct access to raw data is available, and known to be linear, the response curve estimation step could be avoided, since in this case the function equals a simple straight line normalized in the range $[0, \dots, 1]$. Fig.(13.10) shows 10 images captured at different exposure settings, from $\frac{1}{1600}$ sec to $\frac{1}{4}$ sec, while Fig.(13.11) shows the recovered response curve on both linear (Fig.(13.11(a))) and logarithmic units (Fig.(13.11(b))).

13.2.3 High Dynamic Range Image Construction

Once the response function, estimated or a priori known, is at hand the high dynamic range image, usually referred as *radiance map* and composed of floating point values having greater range and tonal resolution than usual *low dynamic range (LDR)* data, can be assembled. The principle is that each pixel in each image, provides a more or less accurate estimation of the radiance value of the scene in the specific position. For example, very low pixel values coming from low exposure images are usually noisy, and thus not reliable, but the same pixels are likely to be well exposed in images acquired with higher exposure settings.

Given N images, with exposure ratios $e_i : i = 1 : N$ and considering (13.17) the sequence $\left\{ \frac{g(Z_{i,1})}{t_1}, \frac{g(Z_{i,2})}{t_2}, \dots, \frac{g(Z_{i,N})}{t_N} \right\}$ of estimates for a pixel in position i is obtained. Different estimates should be assembled by means of a weighted average taking into account reliability of the pixel itself. Of course, the weight should completely discard pixels that appear as saturated and assign very low weight to pixels whose value is below some noise floor, since they are unable to provide decent estimation.

One possible weighting function could be a hat or Gaussian shaped function centered around mid-gray pixel values, which are far from noise and saturation. As a rule of thumb, for each pixel there should be at least one image providing a useful pixel (e.g., that is not saturated, nor excessively noisy). Given the weighting function $w(Z)$ the radiance

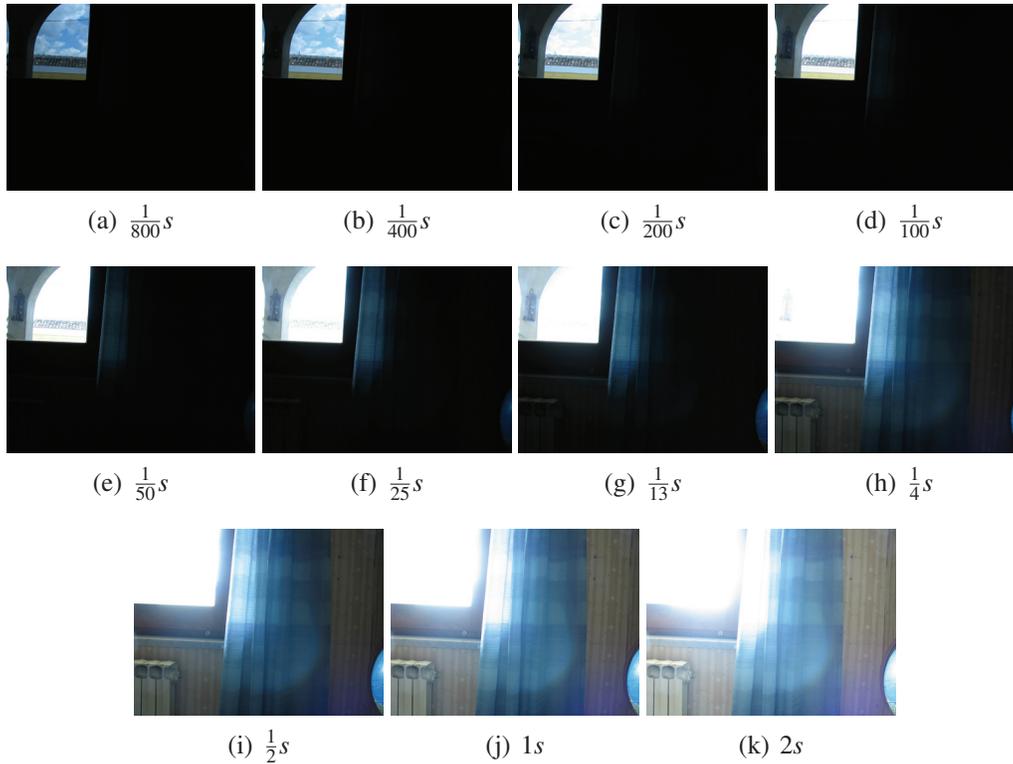


Figure 13.10 : A sequence of 11 images, captured at iso 50, f-4, and exposures ranging from $\frac{1}{800}$ to 2 sec.

estimate for a given position i is given by:

$$I_i = \frac{\sum_{j=1}^N w(Z_{i,j}) \frac{g(Z_{i,j})}{t_j}}{\sum_{j=1}^N w(Z_{i,j})} \quad (13.21)$$

13.2.4 The Scene Versus the Display Medium

Once the high dynamic range image has been assembled, what's usually required is a final rendering on the display medium, such as a CRT display or a printer. The human eye is capable of seeing a huge range of luminance intensities, thanks to its capability to adapt to different values. Unfortunately this is not the way most image rendering systems work. Hence they are usually not capable to deal with the full dynamic range contained into images that provide an approximation of real world scenes. Indeed most CRT displays have a useful dynamic range in the order of nearly 1:100. It's for sure that in the next future, high dynamic reproduction devices will be available, but for the moment they are well far from mass market consumers. Simply stated, *tone mapping* is the problem of converting an image containing a large range of numbers, usually expressed in floating point precision, into a meaningful number of discrete gray levels (usually in the range

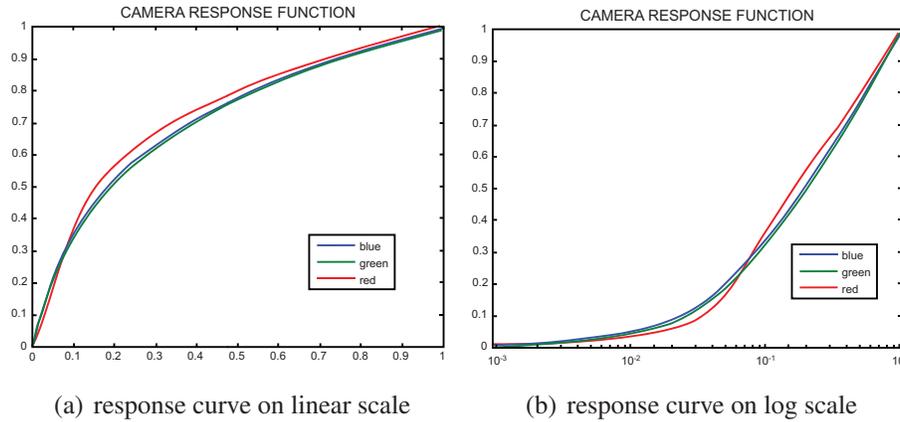


Figure 13.11 : Response curves derived from images depicted in Fig.(13.10).



Figure 13.12 : An HDR image built from the sequences of Fig.(13.10), linearly scaled in the $[0, \dots, 1]$ range and quantized to 8 bits.

$0, \dots, 255$), that can be used by any imaging device. So, we can formulate the topic as that of the following quantization problem:

$$Q(val) = \lfloor (N - 1) \cdot F(val) + 0.5 \rfloor \tag{13.22}$$

$$F : [L_{w_{min}} : L_{w_{max}}] \rightarrow [0 : 1]$$

where $[L_{w_{min}} : L_{w_{max}}]$ is the input range, N the number of allowed quantization levels, and F the tone mapping function. A simple linear scaling usually leads to the loss of a high amount of information on the reproduced image. Fig.(13.12), shows the result obtained by linearly scaling an high dynamic range image, constructed from the sequence of Fig.(13.10) using the techniques described above. As it can be seen, only a parts of the scene are clearly visible, so better alternatives for F are needed.

Two different categories of tone mapping exist:

1. Tone Reproduction Curve (TRC): the same function is applied for all pixels;
2. Tone Reproduction Operator (TRO): the function acts differently depending on the value of a specific pixel and its neighbors.

In what follows, several of such techniques will be briefly described and applied on the input HDR image, assembled from the sequence in Fig.(13.10). The recorded input was in the range of 0.00011 : 32.

Histogram Adjustment (TRC)

The algorithm described in [20], by G. Ward *et al.*, is based on ideas coming from image enhancement techniques, specifically histogram equalization. While histogram equalization is usually employed to expand contrast images, in this case it is adapted to map the high dynamic range of the input image within that of the display medium, while preserving the sensation of contrast. The process starts by computing a downsampled version of the image, with a resolution that equals to 1 degree of visual angle. Luminance values of this, so called *fovea* image, are then converted in the *brightness* domain, which can be approximated by computing logarithmic values. For the logarithmically valued image, an histogram is built, where values between minimum and maximum bounds $L_{w_{min}}$ and $L_{w_{max}}$ (of the input radiance map) are equally distributed on the logarithmic scale. Usually employing around 100 histogram bins each having a size of $\Delta b = \frac{\log(L_{w_{max}}) - \log(L_{w_{min}})}{100}$ provides sufficient resolution. The cumulative distribution function, normalized by the total number of pixels T , is defined as:

$$P(b) = \frac{\sum_{b_i < b} f(b_i)}{T} \quad (13.23)$$

$$T = \sum_{b_i} f(b_i)$$

where $f(b_i)$ is the frequency count for bin i . The derivative of this function can be expressed as

$$\frac{\partial P(b)}{\partial b} = \frac{f(b)}{T \Delta b}. \quad (13.24)$$

Applying a histogram equalization on the input, the result is an image where all brightness values have equal probability. The equalization formula, which provides a way to map luminance values to display values, can be expressed as:

$$\log(L_d(x, y)) = \log(L_{d_{min}}) + (\log(L_{d_{max}}) - \log(L_{d_{min}})) \cdot P(\log \cdot L_w(x, y)) \quad (13.25)$$

where $L_{d_{min}}$ and $L_{d_{max}}$ stay for minimum and maximum display values. This means that the equalized brightness is fit into the available display dynamic range. Unfortunately naive equalization, tends to over-exaggerate contrast in correspondence of highly populated bins (histogram peaks) leading to undesirable effects. To prevent this, a ceiling procedure is applied on the histogram, imposing that contrast should never exceed those obtained



Figure 13.13 : Histogram Adjustment mapping.

by a linear mapping. The ceiling can be written in terms of the derivative of the mapping (which is indicative of contrast):

$$\frac{\partial L_d}{\partial L_w} \leq \frac{L_d}{L_w}. \quad (13.26)$$

By putting together (13.24) and (13.25) the final histogram constraint is obtained:

$$f(b) \leq \frac{T\Delta b}{(\log(L_{d_{max}}) - \log(L_{d_{in}}))}. \quad (13.27)$$

Thus in order to prevent excessive contrast, histogram values are repetitively cut to satisfy (13.27). The operator has been further refined by the authors to include more sophisticated ceiling procedure, simulation of color and contrast sensitivity, according to some features of the human visual system (*HVS*). Fig.(13.13) shows the example radiance map, tonemapped to display using Ward's operator in its basic implementation.

Fast Tone Mapping for High Dynamic Range Image Visualization (TRC)

Another histogram based technique (and somehow similar to the Ward's approach) has been described in [21] by J. Duan *et al.*, which the authors deliberately describe as a visualization tool for high dynamic range images. The authors, correctly, claim that image reproduction is strictly device dependent (with display calibration being the main issue), and thus aim to provide a technique that allows the user to quickly display image content while allowing a good deal of flexibility in brightness and contrast parameters tuning, which can be intuitively changed "on the fly". The technique is a two-phase process, where the first one scales the range of the input image L_w within display allowed values and sets the amount of brightness, while the second one tunes the amount of final contrast:

1. luminance mapping;

2. adaptive histogram adjustment.

The first phase is carried out by means of the following logarithmic/linear scaling, where $\tau = \alpha(L_{w_{max}} - L_{w_{min}})$:

$$L_d(x,y) = (L_{d_{max}} - L_{d_{min}}) \cdot \frac{\log(L_w(x,y) + \tau) - \log(L_{w_{in}} + \tau)}{\log(L_{w_{max}} + \tau) - \log(L_{w_{min}} + \tau)} + L_{d_{min}}. \quad (13.28)$$

The parameter α tunes the brightness settings, with low parameters allocating more input values to the upper range of the display capacity, thus producing bright images, and vice versa.

In order to better distribute the final image values inside the display range a controlled equalization is applied by balancing between uniform quantization, no equalization at all, and full equalization. In the first case, given N available display digital values, the output range of (13.28) is divided into N equally spaced intervals, where all pixels falling inside $[l_{i-1}, l_i]$ are assigned the same value i . In the case of histogram equalization, instead, the quantization intervals are not uniform, but the number of pixels falling into each interval, $[e_{i-1}, e_i]$ is the same. Once again pixels inside $[e_{i-1}, e_i]$ are assigned to the digital value i . The balancing is achieved by dividing the output range of (13.28) into N $[el_{i-1}, el_i]$ intervals according to the following equation:

$$el_i = l_i + \beta e_i, \quad i = 1, 2, \dots, N. \quad (13.29)$$

By varying the β value in the range $[0, 1]$ the user can vary the image visualization between no equalization and full equalization. Fig.(13.14(a)) and Fig.(13.14(b)), show two visualizations with different α settings, while β remains unchanged.



(a) $\alpha = 0.25, \beta = 2.0$

(b) $\alpha = 0.75, \beta = 2.0$

Figure 13.14 : Image visualized with two different α values (0.25 and 0.75), and same β value, (2.0) to show the brightness effect.

Bilateral Filtering (TRO)

Durand *et al.* in [22] consider the input image as separable into two different layers: a *base layer* and a *detail layer*. The first is related to the low frequency content of the image and the second to the high frequency content. Thus an input image $L_w(x, y)$ can be expressed as the multiplication of its two layers: $L_w(x, y) = Base(x, y) \cdot Detail(x, y)$. In order to properly scale the high dynamic range data, the base layer is feed to a compressive function, while the detail layer is leaved unchanged. This helps the preservation of subtle local contrast, and is also related to the concept that the base layer represents the influence of lighting conditions (and thus the scene dynamic range). All the processing is done in the logarithmic domain, where the two layers are separated, processed and recombined. The basic steps of the algorithm are:

1. express the data in the logarithmic domain $l(x, y) = \log(L_w(x, y))$;
2. compute the base layer $Base(x, y)$;
3. compute the detail layer $Detail(x, y) = Base(x, y) - l(x, y)$;
4. compress the base layer obtaining $comp(Base(x, y))$;
5. recombine the layers and exponentiate the result, to produce the final image, $L_d(x, y) = \exp(comp(Base(x, y)) + Detail(x, y))$.

The compression of the base layer is simply done by scaling it by a multiplicative factor m , such that its range equals a desired contrast c :

$$comp(Base(x, y)) = m \cdot Base(x, y), \quad (13.30)$$

such that

$$m \cdot (\max(Base(x, y)) - \min(Base(x, y))) = c. \quad (13.31)$$

The most relevant feature of the algorithm is the way in which the base layer is computed, which should be a low pass filtered version of the image, but without the unwanted issues of the Gaussian filtering employed by the aforementioned Chiu's algorithm. In other words the low pass filtering process should not consider for each pixel, those luminance values that are far from the luminance of the pixel itself. To achieve this, a *bilateral filter* is considered. Bilateral filtering $bil(x, y)$ is obtained by adding to usual gaussian filter with a spatial kernel g , a further Gaussian weighting function w , whose weights decrease as the difference in luminance value between the central pixel and its neighbors in a surround Ω increase.

$$bil(x, y) = \frac{1}{k(x, y)} \sum_{(u, v) \in \Omega} g(x - u, y - v) \cdot l(u, v) \cdot w(d(l(u, v), l(x, y))),$$

$$d(u, v) = |l(u, v) - l(x, y)| \quad (13.32)$$

where $k(x, y)$ is the normalization term. Since bilateral filtering in the spatial domain can be computationally very slow, the authors have developed a very fast approximation in the frequency domain. Fig.(13.15) shows the result of the algorithm ($c = \log(50)$) on the input image.



Figure 13.15 : Image mapped with bilateral filtering.

Photographic Tone Reproduction (TRO)

Reinhard *et al.* [23] have developed an operator based on some simple photographic principles such as *automatic exposure* and *dodge and burning*, where the first provides a global mapping of the image and the latter exploits some local features. The global part of the operator analyzes the concept of *scene's key*, which is measure of how overall dark or bright the images is. This quantity is approximated with the log average value \bar{L}_w of the image luminance values. According to photographic principles, where the key is usually printed (or displayed) to have the 18% reflectance of the medium, an initial global mapping is performed using the following equation:

$$L_m(x,y) = \frac{0.18}{\bar{L}_w} \cdot L_w(x,y). \quad (13.33)$$

In this way a kind of automatic exposure setting is provided for the scene, even it is done *ex post facto*, since the scene has already been captured by the camera (but, since radiance maps provide a floating point description of the initial scene, this allow us to do such virtualizations of the photographic process). Even if in (13.33), the scene's key value is linearly mapped to the value of 0.18, different values could be used depending on the specific image content (e.g., a nightlife picture should be scaled to a very low value). No matter what the dynamic range of the initial scene is, the luminance values exposed by means of (13.33) are forced to fit inside the medium dynamic range (which is here supposed to vary within $[0, \dots, 1]$) using a compressive function, which is particularly effective on very high luminance values:

$$L_d(x,y) = \frac{L_m(x,y)}{1 + L_m(x,y)}. \quad (13.34)$$

This function scales input values differently, according to their magnitude: small values, usually $\ll 1$ are almost leaved unchanged, while very high values, usually $\gg 1$, are scaled by a very large amount (the quantity $\frac{1}{L_m(x,y)}$).

The scaling function, is further refined to include some local processing, similar to *dodge and burning* procedures, where a dark value in a bright surround is heavily compressed (burned), and a bright pixel on a dark surround is only mildly compressed (dodged). To exploit these local properties, filtered versions at different scales $s = 1, 2, \dots, S$ of L_m are produced as

$$L_{blur_s} = L_m * R_s \quad (13.35)$$

and in (13.34) the quantity $L_m(x, y)$ on the denominator, is replaced by

$$L_d(x, y) = \frac{L_m(x, y)}{1 + L_{blur_s}(x, y)} \quad (13.36)$$

where $*$ equals to the convolution operator, and R_s are different Gaussian kernels, having different pixel widths $w(s)$, for $s = 1, 2, \dots, S$

$$w(s) = e^{\log(w_{min}) + \frac{s}{S} \cdot (\log(w_{max}) - \log(w_{min}))}. \quad (13.37)$$

the term w_{max} and w_{min} are the maximum and minimum allowed pixel widths, and are fixed respectively to 1 and 43. Thus the smallest scale for a pixel in position (x, y) equals to the pixel itself. To avoid halo artifacts, for each pixel the largest scale $s_{max} : |V_{s_{max}}(x, y)| < \varepsilon$, where V_s is the difference between two successive scales, is computed as:

$$V_s(x, y) = \frac{L_{blur_s} - L_{blur_{s+1}}}{2^\Phi \cdot 0.18/s^2 + L_{blur_s}}. \quad (13.38)$$

According to the authors a value of $\Phi = 8$ is used. Practically (13.38) is the search of the largest surround across a pixel in position (x, y) whose value is reasonably similar to that of the pixel. This avoids the appearance of severe halo artifacts, similar to those seen by the application of Chiu's algorithm. Fig.(13.16) shows the result of the algorithm of Reinhard *et al.* on our example radiance map, where the parameters $S = 8, \Phi = 8$ have been used.

Gradient Compression (TRO)

The last technique, belonging to the family of TRO, that we are going to describe was developed by Fattal *et al.* [24], and it's far more sophisticated than those that have been described hereof. Even if sometimes output images can have an unnatural appearance, in most cases results can look very appealing. This algorithm doesn't operate directly on the spatial domain, but instead computes the *gradient field* of the input image and after manipulating it, reconstructs by means of Poisson integration the image having the new gradients. This derives from the observation that an image exhibiting an high dynamic range, will be characterized by gradients of large magnitude around zones of brightness transition. Hence attenuating those gradients seems like a viable way for building a LDR depiction of the scene, suitable to be viewed on a common display. Similarly to the



Figure 13.16 : Photographic Tone Reproduction mapping.

pipeline of the algorithm based on bilateral filtering, gradient compression works on logarithmic data, and so just before producing the output image, the result undergoes exponentiation. Indicating with $l(x,y)$ the data in the logarithmic domain, the gradient field $\nabla l(x,y)$ is computed as follows:

$$\nabla l(x,y) = (l(x+1,y) - l(x,y), l(x,y+1) - l(x,y)) \quad (13.39)$$

Attenuation of the gradient field is obtained by multiplication with a proper scaling function $\Phi(x,y)$:

$$G(x,y) = \nabla l(x,y) \cdot \Phi(x,y). \quad (13.40)$$

The attenuated gradient field $G(x,y)$ is then inverted by solving the Poisson equation

$$\nabla^2 \tilde{l}(x,y) = \text{div } G(x,y). \quad (13.41)$$

Since edges (and thus gradients) exist at multiple resolution levels, a Gaussian pyramid representation $\langle l_0, l_1, \dots, l_s \rangle$ is constructed, and for each level the gradient field is computed. The attenuation function is then computed on each level and reported to the upper level in *bottom to top* fashion. The attenuation function at the top level is the one that will be effectively used in (13.40). Attenuation function at each level s is computed as follows:

$$\Psi_s(x,y) = \frac{\alpha}{\|\nabla l_s(x,y)\|} \cdot \left(\frac{\|\nabla l_s(x,y)\|}{\alpha} \right)^\beta. \quad (13.42)$$

The α parameter in 13.42 determines which gradient magnitudes are leaved untouched, while the β exponent amplifies magnitudes greater than α . Suggested values are $\alpha = 0.1 \cdot (\text{average gradient magnitude})$ and $\beta = 0.9$. Since the attenuation function is computed for each resolution level s , the propagation to full resolution is done by scaling the attenuation function from level $s-1$ to s , and accumulating the values to obtain the full

resolution attenuation function $\Phi(x, y)$ that will be effectively used (authors claim that by using the attenuation function just at full resolution halo artifacts are mostly invisible). This can be expressed by the following equations:

$$\begin{aligned}\Phi_d(x, y) &= \Psi_d(x, y) \\ \Phi_k(x, y) &= L(\Phi_{k+1})(x, y) \cdot \Psi_d(x, y) \\ \Phi(x, y) &= \Phi_0(x, y)\end{aligned}\tag{13.43}$$

where d is the smallest resolution level and L is the bilinear up-sampling operator. Fig.(13.17) shows the result of applying the gradient compression operator on our sample HDR image. The operator looks computationally more complicated than others that have been described but as it can be seen the mapped image looks far more impressive, in terms of high-light and low-light visibility, than the previous renderings.



Figure 13.17 : Gradient Compression mapping.

13.3 Computational Photography

Traditional digital cameras coupled with the latest computational methods in digital imaging overcome the traditional limitations of a camera and enable novel imaging applications [25]. The so-called computational photography extends digital photography because it records much more information than before and offers the possibility to process it. The final output is yet an ordinary picture, but one that could not have been taken by a traditional camera. A novel way to exploit the fruition of the information coming from the real world is hence possible. The term was used in the current definition, from a 2004 course at Stanford University; it has rapidly evolved to cover a number of subject areas in Computer Graphics, Computer Vision, and Applied Optics. In particular several sub-topics can be identified according to the following list:

- Computational illumination that includes flash/no-flash imaging, multi-flash imaging, different exposures imaging, image-based relighting and other uses of structured illumination.
- Computational optics devoted to capture optically coded images, followed by computational decoding to produce new images. Coded aperture imaging is used to boost the image quality just making use of a pinhole pattern (not a single pin-hole), while deconvolution is performed to recover the image. In coded exposure imaging, the on/off state of the shutter is coded to modify the kernel of motion blur just to be able to make tractable the motion deblurring problem. Similarly, in a lens based coded aperture, the aperture can be modified by inserting a broadband mask. Thus, out of focus deblurring becomes a well-conditioned problem. The coded aperture can also improve the quality in light field acquisition using Hadamard transform optics. Some of the involved issues are: coded aperture imaging, coded exposure imaging, light field photography, catadioptric imaging, wavefront coding, compressive imaging.
- Computational processing of non-optically coded images to produce new images is mainly referred to the following arguments: panorama mosaicing, matte extraction, digital photomontage, high dynamic range imaging, all-focused imaging.
- Computational sensors treat detectors that combine sensing and processing, typically in hardware as: artificial retinas, high dynamic range sensors, retinex sensors.

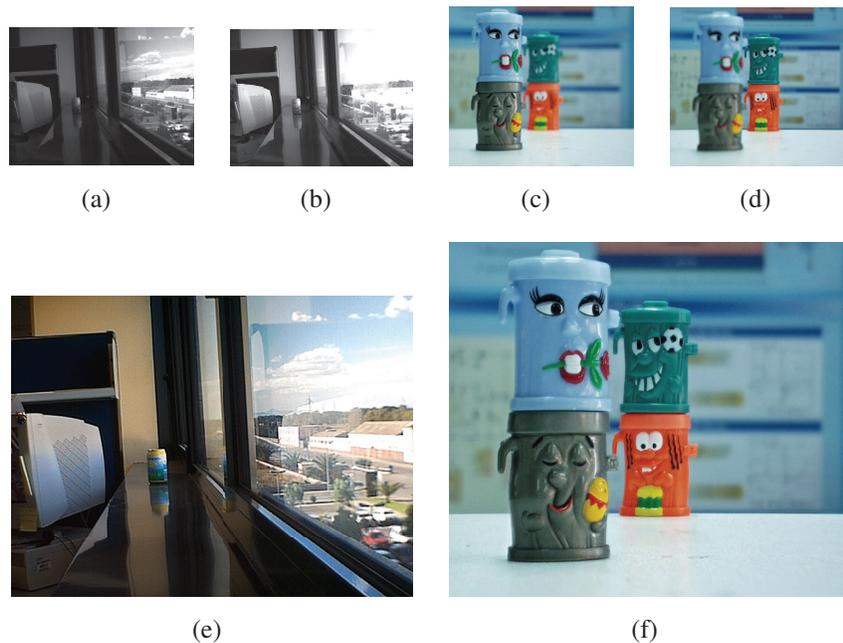


Figure 13.18 : Multiple picture fusion.

Recently a lot of material has been presented at major international conferences (e.g., SIGGRAPH07, SIGGRAPH08, etc.). For more details see [26,27]. Just to provide an example of the great expectation about this new exciting field we cite the work [28] where a framework is proposed for automatically enhancing the spatial and/or temporal resolution of videos (usually in low resolution) of a static scene using a few photographs of the same scene (usually in high resolution).

To conclude this Section we mention an interesting project [29] that has as main goal the realization of a new open source camera platform with sufficient power computing to implement some of the new trends in the field. The project is also devoted to exploit the possibility to develop effective (and profitable) computational photography applications for commercially available mobile phones.

In Fig.(13.18) two examples of multi picture acquisition of the same scene that allowed to obtain enhanced images in term of dynamic range [30] and depth of field are shown.

13.4 Camera Identification

In Chapter 6 a deep review about the various kinds of noise that affect images acquired by single-sensor imaging devices due to different causes and modalities has been presented. While almost all of them should be removed, in some way, to improve the image quality, there is a specific source of noise that can be used to univocally identify the camera manufacturer, model and, with some extents, also the specific camera that has acquired a given image. Such peculiarity is useful for forensics application devoted to address the camera source identification but also for tampering detection. Also different copyright related issues could be interested in using such technology (e.g., device identification, device linking, recovery of processing history and detection of digital forgeries). Although some other methods have been recently proposed for camera identification [31], the method based on the noise intrinsic signature related to the particular camera seems to be very promising.

In details, the property of imaging sensor used in this case is the photo-response non uniformity (PRNU), that is intrinsically related with the small variations among individual pixels during photons acquisition and subsequent conversion to electrons. The obtained pattern can be used as a sort of sensor fingerprint, that, acting as an unintentional stochastic spread-spectrum watermark, is able to resist also to heavy processing (i.e., lossy compression). See [31,32] for more details.

Given an input image the PRNU can be extracted just removing the other noise component of the pattern noise: the fixed pattern noise (FPN) (see Chapter 6). The FPN is caused by dark currents. It primarily refers to pixel-to-pixel differences when the sensor array is not exposed to light. Because the FPN is an additive noise, some middle-to high-end consumer cameras suppress this noise automatically by subtracting a dark frame from every image they take. FPN also depends on exposure and temperature.

Just having a suitable mathematical model for the overall noise it is possible to filter out the low frequencies content of the pattern noise maintaining only the pixel non unifor-

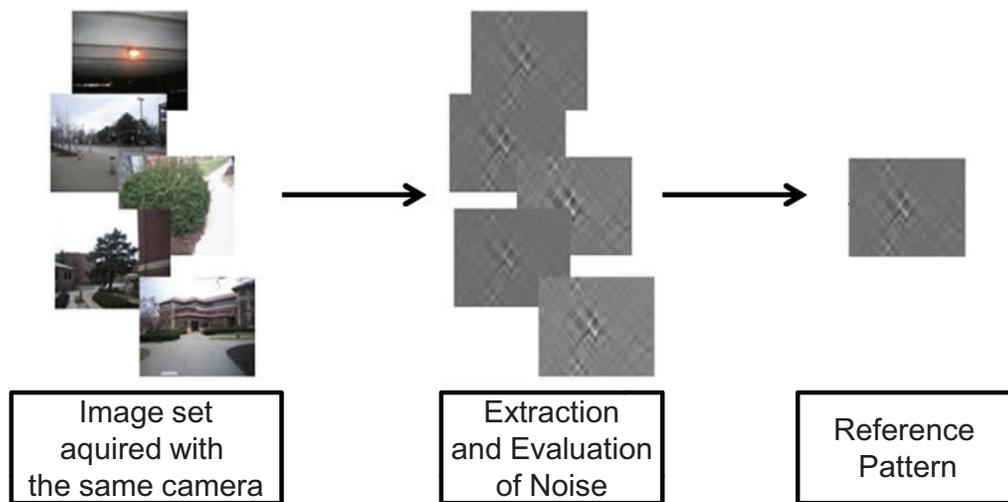


Figure 13.19 : Overall schema of the camera identification by noise signature analysis.

mity (PNU) noise component. Such values are intrinsically related with the manufacturing process of each single sensor and can be used as a signature for camera identification. Such intrinsic patterns can be estimated from images taken by the camera by averaging their noise components as shown in Fig.(13.19).

More specifically, it is possible to test the Netherlands Forensic Institute PRNU identifier [33] that is developed as open-source code.

More recently in [34] the overall process has been improved just introducing some pre-processing of the input images used in the training phase taking into account several factors related with weak artifacts of color interpolation, on sensor signal transfer and on sensor design. In the same paper the same idea is used to detect forgery [35] inside the image.

Bibliography

- [1] S. Chaudhuri, *Super-Resolution Imaging*. Kluwer, September 2001.
- [2] R. Y. Tsai and T. S. Huang, "Multipleframe image restoration and registration," *Advanced Computer Vision Image Processing*, no. 1, pp. 317–339, 1984.
- [3] R. S. Prendergast and T. Q. Nguyen, "Improving frequency domain super-resolution via undersampling model," in *IEEE International Conference on Image Processing (ICIP 2005)*, vol. 1, pp. I–853, 2005.
- [4] P. Cheesman, B. Kanefsky, R. Kraft, J. Stutz, and R. Hanson, "Super-resolved surface reconstruction from multiple images," in *Maximum Entropy and Bayesian Methods*, pp. 293–308, 1996.
- [5] M. Ozkan, A. Tekalp, and M. Sezan, "POCS-based restoration of space-varying blurred images," *IEEE Transactions on Image Processing*, vol. 3, pp. 450–454, Jul 1994.
- [6] A. Patti, M. I. Sezan, and A. M. Tekalp, "Super resolution video reconstruction with arbitrary sampling lattices and non-zero aperture time," *IEEE Transactions on Image Processing*, vol. 6, pp. 1064–1076, 1997.
- [7] M. Irani and S. Peleg, "Super resolution from image sequences," in *International Conference on Pattern Recognition (ICPR 1990)*, pp. 115–120, 1990.
- [8] M. Irani and S. Peleg, "Improving resolution by image registration," in *Graphics Models and Image Processing (CVGIP 1993)*, vol. 53, pp. 324–335, 1993.
- [9] G. Messina, S. Battiato, M. Mancuso, and A. Buemi, "Improving image resolution by adaptive back-projection correction techniques," *IEEE Transactions on Consumer Electronics*, vol. 48, no. 3, pp. 409–416, 2002.
- [10] S. Farsiu, M. Elad, and P. Milanfar, "Multi-frame demosaicing and super-resolution of color images," *IEEE Transaction on Image Processing*, vol. 15, no. 1, pp. 141–159, 2006.
- [11] X. Wu and L. Zhang, "Improvement of color video demosaicing in temporal domain," *IEEE Transaction on Image Processing*, vol. 15, no. 10, pp. 3138–3151, 2006.
- [12] W. Freeman, T. Jones, and E. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.
- [13] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *International Conference on Computer Vision (ICCV 2009)*, 2009.

- [14] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1167–1183, 2002.
- [15] Y. Ryoji, I. Hiroshi, A. Hiroaki, K. Takeshi, K. Hideki, and M. Yoshihiro, "Development of single-frame super-resolution system LSI," Tech. Rep. 4, NEC Electronics, 2009. <http://www.nec.co.jp/techrep/en/journal/g09/n01/090104.pdf>.
- [16] C. Wyckoff, "An experimental extended response film," *SPIE Newsletter*, July 1962.
- [17] Y.-C. Chang and J. F. Reid, "RGB calibration for analysis in machine vision," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 1414–1422, October 1996.
- [18] T. Mitsunaga and S. Nayar, "Radiometric self calibration," in *Computer Vision and Pattern Recognition*, vol. 2, pp. 374–380, 1997.
- [19] S. Mann, "Comparametric equations with practical applications in quantigraphic image processing," *IEEE Transactions on Image Processing*, vol. 9, pp. 1389–1406, August 2000.
- [20] G. W. Larso, H. Rushmeier, and C. Piatko, "A visibility matching tone reproduction operator for high dynamic range scenes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, pp. 291–306, 1997.
- [21] J. Duan, G. Qiu, and M. Chen, "Comprehensive fast tone mapping for high dynamic range image visualization," in *Pacific Graphics Conference on Computer Graphics and Applications*, 2005.
- [22] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high dynamic range images," in *Proceedings of SIGGRAPH 2002*, pp. 257–266, 2002.
- [23] E. Reinhar, M. Stark, P. Shirley, and J. Ferweda, "Photographic tone reproduction for digital images," in *Proceedings of SIGGRAPH 2002*, pp. 267–276, 2002.
- [24] R. Fattal, D. Lischinski, and M. Werman, "Gradient domain high dynamic range compression," in *Proceedings of SIGGRAPH 2002*, pp. 249–256, 2002.
- [25] R. Motta, "The future of digital photography," in *Digital Photography VII, vol.7537*, SPIE Electronic Imaging, 2010.
- [26] R. Raskar and J. Tumblin, *Computational Photography: Mastering New Techniques for Lenses, Lighting, and Sensors*. A K Peters, 2010. In press.
- [27] "Special Issue on Computational Photography," *IEEE Computer*, vol. 39, Aug. 2006.

- [28] A. Gupta, P. Bhat, M. Dontcheva, B. Curless, O. Deussen, and M. Cohen, “Enhancing and experiencing spacetime resolution with videos and stills,” in *International Conference on Computational Photography*, IEEE, 2009.
- [29] Stanford Computer Graphics Laboratory, “Camera 2.0: New computing platforms for computational photography,” 2010. <http://graphics.stanford.edu/projects/camera-2.0/>.
- [30] S. Battiato, A. Castorina, and M. Mancuso, “High dynamic range imaging for digital still camera: an overview,” *Journal of Electronic Imaging*, vol. 12, pp. 459–469, July 2003.
- [31] J. Fridrich, “Digital image forensics,” *IEEE Signal Processing Magazine*, vol. 26, pp. 26–37, March 2009.
- [32] J. Lukas, J. Fridrich, and M. Goljan, “Digital camera identification from sensor pattern noise,” *IEEE Transactions on Information Forensics and Security*, vol. 1, pp. 205–214, June 2006.
- [33] Netherlands Forensic Institute, “NFI PRNU Compare open-source code,” 2009. <http://sourceforge.net/projects/prnucompare/>.
- [34] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, “Determining image origin and integrity using sensor noise,” *IEEE Transactions on Information Forensics and Security*, vol. 3, pp. 74–90, March 2008.
- [35] C. Baron, *Adobe Photoshop Forensics*. Course Technology PTR, 1st ed., 2007. ISBN 1598634054.

Index

- XYZ, 101
 ΔE_{94} , 103
 χ^2 Distance, 253
- AF, Auto-focus, 55
 APS, Active Pixel Sensor, 19
 AutoExposure, 240, 244
 Autofocus, 240, 244
 AWB, 96
 AWGN, Additive White Gaussian Noise, 125, 126
- bag of visual words model, 244
 Bayes' theorem, 241
 Bhattacharyya coefficient, 253, 254
 Bilateral Filtering, 144
 binary classifiers, 241, 242, 260
 biological vision systems, 239
 Bit Rate, 211
 Block Matching, 221
 blooming, 17
- categorization, 239, 240
 CCD, Charge Coupled Device, 11, 19
 CCT, Correlated Color Temperature, 99
 Center-weighted Average Metering, 39
 chromatic aberration, 13
 Chrominance Noise, 136
 CIE, Commission International de l'Eclairage, 101
 CIELAB, 206, 208
 CIELab, 102
 Classification, 213
 Classifier, 205
 cluster-based LDA, 263–265
 clustering, 263, 264
 CMOS, Complementary MetalOxideSemiconductor, 11, 20
 Color Balancing, 1
 Color Cast, 197
 color constancy, 243, 266
 Color Encoding, 321
 Compression, 1
 Computer Vision, 239–241, 244, 266
 confusion matrix, 261, 265
 content-based image retrieval, 240
 Contrast, 43
- Dark Current Noise, 124
 Dark Flash, 214
 DCT, 245–248, 250, 260
 Demosaicing, 1
 Denoising, 1
- Depth of Field, 208
 Desaturation, 206, 207
 detection, 240
 Differential Mean Opinion Score (DMOS), 317
 dimensional reduction, 241, 265
 Discrete Cosine Transform, 240, 244
 discriminative classifiers, 242
 discriminative methods, 242, 263
 DoF, Depth of Field, 55, 59, 62, 83, 85
 dof, depth of focus, 59
 Double Stimulus Continuous Quality Scale (DSCQS), 316
 Double Stimulus Impairment Scale (DSIS), 316
 DPCM, 286, 289, 290
 DR, Dynamic Range, 121
 DRM, Dichromatic Reflection Model, 98
 DSC, Digital Still Camera, 55
 Dynamic range, 357
- EBCOT, 295
 Edge Detection, 198
 edge orientation, 248
 EDoF, Extended Depth of Field, 55, 85, 86
 Enhancement, 1
 error concealment, 275, 284, 297–299, 301
 error detection, 275, 284, 297–299
 Exposure Correction, 1
 Exposure Time, 208
- Face Detection, 200
 False Positive, 209
 Feature Based Motion Estimator, 227
 filter banks, 244
 Flash, 193, 198, 208, 214
 Flash/no-Flash, 208, 214
 Focus, 43
 Forward Mapping, 233
 FPN, Fixed Pattern Noise, 124
 Frame Warping, 219
 frequency domain, 244, 247
 FSWM, Frequency Selective Weighted Median filter, 73
 Full Reference Metrics (FR), 314
 Full Search, 76
- gamma, 95
 gamut, 94
 Gaussian, 207
 generative classifiers, 242
 generative methods, 242
 Golden Eyes, 195

- gray-world, 96
- H.263, 280, 281, 283
- H.264, 284–286
- HCS, Hill Climbing Search, 76
- HDR, High Dynamic Range, 11, 28
- Heavy Tailed Noise, 129
- Histogram Adjustment, 365
- histograms of oriented blocks, 244
- holistic cues, 239
- holistic representation, 240, 244, 246, 247, 250
- HSL, 198, 203
- Human Visual System, 239, 240
- Human Visual System (HVS), 313
- HVS, Human Visual System, 95
- Hyperfocal distance, 60
- Illuminant, 197
- image categorization, 239–241, 250
- Image Coding, 240, 244
- Image Compression, 211
- Image Enhancement, 240
- Image Generation Pipeline, 239, 242
- Image Quality, 313
- Imaging Pipeline, 1
- imaging pipeline, 239, 243
- Impulse Noise Model, spike pixel, dead pixel, 127
- Infinite Impulse Response Filter, 229
- Inpainting, 207, 211
- Inverse Mapping, 233
- Jeffrey Divergence, 253
- JPEG, 211, 239, 240, 244, 246, 248, 251, 274, 286, 292, 305, 306
- JPEG-LS, 286
- JPEG2000, 278, 295
- K-means, 264
- K-nearest neighbors, 244, 250, 252, 254
- Kalman Filter, 229
- KNN, 250, 252, 254–256
- LDA, 263–265
- LDR, Low Dynamic Range, 11, 28
- leave-one-out cross validation, 250, 252, 264, 265
- Lens Aperture, 208
- Lightness Perception, 320
- Likelihood, 241
- Linear Discriminant Analysis, 242, 263
- linear discriminant classifier, 240
- local dominant orientation, 245–250, 253, 257, 258, 260
- local invariant descriptors, 244
- Logistic Classification, 242, 252, 255–258
- logistic classifier, 240, 244, 255, 256, 260
- Luminance Contrast, 326
- Luminance Noise, 135
- Machine Learning, 239, 240, 244, 266
- machine vision systems, 239, 241
- Mahalanobis distance, 45
- Masking, 331
- Matrixing, 1
- maximum a posteriori, 256
- Maximum Likelihood Estimation, 255
- Mean Opinion Score (MOS), 315
- Morphological Operators, 198, 204, 208
- Motion Estimation, 219
- Motion Models, 220
- MPEG-4, 282–284, 296, 299, 300
- MSE, 318
- No Reference Metric (NR), 314
- Noise, 208, 211
- Noise Estimation, spike pixel, dead pixel, 138
- object detection, 240, 266
- object recognition, 240
- Objective Metric, 318
- one-against-all, 242, 244
- one-against-one, 242
- Opponent Color Space, 324
- Outlier Filtering, 226
- Pairing Verification, 200
- Pattern Recognition, 244
- PCA, 264, 265
- Peak Signal to Noise Ratio (PSNR), 318
- Pearson Correlation Coefficient, 253
- Perceptual Objective Metric, 320
- Perceptual Objective Metrics, 335
- phase detection system, 68
- Pipeline Modularity, 1
- Pixel, Picture Element, 11
- Pooling, 335
- Post-Processing, 193
- Pre-Flash, 193
- Predicted Differential Mean Opinion Score (pDMOS), 318
- Principal Component Analysis, 241, 264
- PRNU, Pixel Response Non Uniformity, 124
- PSF, Point Spread Function, 61, 85
- PSN, Photon Shot Noise, 122
- purple fringing, 16
- PVC, 289
- PVQ, 289
- Quantization Noise, 124

- Rate-Distortion, 295
- raw data, 277
- Readout Noise, 124
- Red Eyes, 193
- red-eye detection, 239, 262, 266
- Redness Map, 198
- Reduced Reference Metric (RR), 314
- reflectance, 93
- Reset Noise, 124
- Resolution, 198
- RGB, 197, 203
- RIP, Rotation In Plane, 83
- ROI, 198
- ROP, Rotation Out of Plane, 83

- Saliency, 334
- scene recognition, 239, 240, 242–244, 247, 266
- Segmentation, 210
- Semi Local Masking, 339
- Sigma-Filter, 143
- similarity measure, 252–254
- Simultaneous Double Stimulus for Continuous Evaluation, 317
- single sensor imaging devices, 239, 240, 242, 244, 246, 254, 266
- Single Stimulus Continuous Quality Evaluations (SS-CQE), 316
- Skin Detection, 45, 196, 198
- skin tone, 109
- SLR, Single Lens Reflex, 11, 55, 68
- SML, Sum-Modified Laplacian, 74
- SMLF, Sum-Modified Laplacian Focus measure, 74
- SNR, Signal to Noise Ratio, 121
- spatial constraints, 244
- spatial layout, 244
- spectral signatures, 244
- Square Difference Distance, 253
- sRGB, 95, 104
- statistics of natural images, 244
- Structural Similarity Index Metric (SSIM), 337
- Subjective Metric, 314
- supervised learning, 241

- Template, 198
- Temporal (Random) Noise, 124
- Temporal Filtering, 146
- training set, 241, 255, 257, 258, 261, 265

- Universal Index Quality (UIQ), 336
- Unwanted Movement Detection, 219

- Vector Quantization, 286
- Video Stabilization, 219

- Vision, 239
- Visual Attention, 334
- Visual Categorization, 239
- visual content, 239, 240
- Visual Quality Expert Group (VQEG), 315
- visual vocabulary, 244, 247

- WDR, Wide Dynamic Range, 11, 28
- Weighted Euclidean Distance, 253
- White Balance, 240, 244
- white patch, 97

- YCC, 198

- Zone System, 38