

M. Emre Celebi  
Bogdan Smolka *Editors*

# Advances in Low- Level Color Image Processing

# Lecture Notes in Computational Vision and Biomechanics

Volume 11

## *Series editors*

João Manuel R. S. Tavares, Porto, Portugal  
R. M. Natal Jorge, Porto, Portugal

## *Editorial Advisory Board*

Alejandro Frangi, Sheffield, UK  
Chandrajit Bajaj, Austin, USA  
Eugenio Oñate, Barcelona, Spain  
Francisco Perales, Palma de Mallorca, Spain  
Gerhard A. Holzapfel, Stockholm, Sweden  
J. Paulo Vilas-Boas, Porto, Portugal  
Jeffrey A. Weiss, Salt Lake City, USA  
John Middleton, Cardiff, UK  
Jose M. García Aznar, Zaragoza, Spain  
Perumal Nithiarasu, Swansea, UK  
Kumar K. Tamma, Minneapolis, USA  
Laurent Cohen, Paris, France  
Manuel Doblaré, Zaragoza, Spain  
Patrick J. Prendergast, Dublin, Ireland  
Rainald Löhner, Fairfax, USA  
Roger Kamm, Cambridge, USA  
Thomas J. R. Hughes, Austin, USA  
Yongjie Zhang, Pittsburgh, USA  
Yubo Fan, Beijing, China

For further volumes:

<http://www.springer.com/series/8910>

The research related to the analysis of living structures (Biomechanics) has been a source of recent research in several distinct areas of science, for example, Mathematics, Mechanical Engineering, Physics, Informatics, Medicine and Sport. However, for its successful achievement, numerous research topics should be considered, such as image processing and analysis, geometric and numerical modelling, biomechanics, experimental analysis, mechanobiology and enhanced visualization, and their application to real cases must be developed and more investigation is needed. Additionally, enhanced hardware solutions and less invasive devices are demanded.

On the other hand, Image Analysis (Computational Vision) is used for the extraction of high level information from static images or dynamic image sequences. Examples of applications involving image analysis can be the study of motion of structures from image sequences, shape reconstruction from images and medical diagnosis. As a multidisciplinary area, Computational Vision considers techniques and methods from other disciplines, such as Artificial Intelligence, Signal Processing, Mathematics, Physics and Informatics. Despite the many research projects in this area, more robust and efficient methods of Computational Imaging are still demanded in many application domains in Medicine, and their validation in real scenarios is matter of urgency.

These two important and predominant branches of Science are increasingly considered to be strongly connected and related. Hence, the main goal of the LNCV&B book series consists of the provision of a comprehensive forum for discussion on the current state-of-the-art in these fields by emphasizing their connection. The book series covers (but is not limited to):

- Applications of Computational Vision and Biomechanics
- Biometrics and Biomedical Pattern Analysis
- Cellular Imaging and Cellular Mechanics
- Clinical Biomechanics
- Computational Bioimaging and Visualization
- Computational Biology in Biomedical Imaging
- Development of Biomechanical Devices
- Device and Technique Development for Biomedical Imaging
- Experimental Biomechanics
- Gait & Posture Mechanics
- Grid and High Performance Computing for Computational Vision and Biomechanics
- Image Processing and Analysis
- Image Processing and Visualization in Biofluids
- Image Understanding
- Material Models
- Mechanobiology
- Medical Image Analysis
- Molecular Mechanics
- Multi-Modal Image Systems
- Multiscale Biosensors in Biomedical Imaging
- Multiscale Devices and Biomems for Biomedical Imaging
- Musculoskeletal Biomechanics
- Multiscale Analysis in Biomechanics
- Neuromuscular Biomechanics
- Numerical Methods for Living Tissues
- Numerical Simulation
- Software Development on Computational Vision and Biomechanics
- Sport Biomechanics
- Virtual Reality in Biomechanics
- Vision Systems

M. Emre Celebi · Bogdan Smolka  
Editors

# Advances in Low-Level Color Image Processing

 Springer

*Editors*

M. Emre Celebi  
Computer Science Department  
Louisiana State University  
Shreveport, LA  
USA

Bogdan Smolka  
Department of Automatic Control  
Silesian University of Technology  
Gliwice  
Poland

ISSN 2212-9391

ISSN 2212-9413 (electronic)

ISBN 978-94-007-7583-1

ISBN 978-94-007-7584-8 (eBook)

DOI 10.1007/978-94-007-7584-8

Springer Dordrecht Heidelberg New York London

Library of Congress Control Number: 2013953224

© Springer Science+Business Media Dordrecht 2014

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law. The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media ([www.springer.com](http://www.springer.com))

# Preface

Color perception plays an important role in object recognition and scene understanding both for humans and intelligent vision systems. Recent advances in digital color imaging and computer hardware technology have led to an explosion in the use of color images in a variety of applications including medical imaging, content-based image retrieval, biometrics, watermarking, digital inpainting, remote sensing, visual quality inspection, among many others. As a result, automated processing and analysis of color images has become an active area of research, which is witnessed by the large number of publications during the past two decades. The multivariate nature of color image data presents new challenges for researchers and practitioners as the numerous methods developed for single channel images are often not directly applicable to multichannel ones.

The goal of this volume is to summarize the state-of-the-art in the early stages of the color image processing pipeline. The intended audience includes researchers and practitioners, who are increasingly using color and, in general, multichannel images.

The volume opens with two chapters on image acquisition. In [Chap. 1](#) Chen et al. focus on the problem of color artifacts generated by line-scan cameras. They propose a method that enables automated correction of the color misalignment in multi-line CCD images for rotational and translational scans. The chapter presents the experimental results achieved using a close-range multi-line CCD imaging system for inspection applications and a long-range camera intended for surveillance tasks. The results confirm that the two imaging systems enable the acquisition of hyper-resolution images with effective color misalignment adjustment.

In [Chap. 2](#) Lee and Park propose a novel adaptive technique for color image demosaicking that exploits the characteristics of the CFA pattern. Comparative experiments performed on a large set of test images show the effectiveness of the proposed interpolation algorithm in terms of peak signal-to-noise ratio, structural similarity, and subjective visual quality. The new algorithm outperforms conventional algorithms especially in the case of natural images containing many image structures such as lines, edges, and corners.

The volume continues with two chapters on color constancy. In [Chap. 3](#) Lee and Plataniotis et al. present a comprehensive survey of color constancy and color invariance. The color of an object recorded in image data is not only a function of an intrinsic property of the object itself, but also a function of the acquisition

device and the prevailing illumination. When these factors are not properly controlled, the performance of color image processing applications can deteriorate substantially. The Authors review two common approaches to attain reliable color description of image data under varying imaging conditions, namely, color constancy and color invariance, where the former is based on scene illuminant estimation and image correction, while the latter is based on invariant feature extraction.

In [Chap. 4](#) Lecca describes applications of the von Kries model of chromatic adaptation to color correction, illuminant invariant image retrieval, estimation of color temperature and intensity of light, and photometric characterization of a device. The von Kries model describes the change in image colors due to illuminant variation. Lecca first illustrates the theoretical foundations of the von Kries model. The Author then presents a method for the model parameter estimation and derives a mathematical relationship between the parameters of the von Kries model and the color temperatures and intensities of the varied illuminants. The chapter concludes by showing a model relating the von Kries parameters to the photometric properties of the acquisition device. Through this model, it is possible to estimate the light wavelengths for which the camera sensors are maximally responsive. These wavelengths are used for finding an illuminant invariant image representation. The chapter reports various experiments carried out on publicly available real-world datasets.

In [Chap. 5](#) Baljovic et al. propose a novel algorithm for removing impulsive or mixed noise from color images based on the halfspace depth function. The resulting multichannel filter maintains the spectral correlation between the color channels and does not depend on the nature or distribution of the noise. The Authors compare the performance of their filter against a large number of state-of-the-art noise removal filters on a diverse set of images.

The volume continues with three chapters on mathematical morphology. In [Chap. 6](#) Debayle and Pinoli present a spatially adaptive image processing framework based on the General Adaptive Neighborhood Image Processing (GANIP) concept. The Authors extend the GANIP approach to color images and define a set of locally adaptive image processing operators. Special emphasis is given to adaptive fuzzy and morphological filters, which are compared to their classic counterparts in restoration, enhancement, and segmentation of color images.

In [Chap. 7](#) Velasco and Angulo investigate the applicability of recent multivariate ordering approaches to morphological analysis of color and multispectral images. The Authors survey supervised learning and anomaly based ordering approaches and present applications of each.

In [Chap. 8](#) Lefèvre et al. review morphological template matching using the Hit-or-Miss Transform (HMT) operator. The Authors review the application of HMT to binary, grayscale, and color images. They also discuss several case studies illustrating practical applications of HMT in different application domains.

The volume continues with two chapters on segmentation. In [Chap. 9](#) Moreno et al. propose two powerful color edge detection methods based on the tensor

voting concept, which extracts structures from a cloud of multidimensional points. The proposed edge detection techniques are evaluated based on measures of completeness, discriminability, precision, and robustness to noise. Experimental results on a database with ground-truth edges reveal useful properties of the new methods, especially in the case of color images distorted by a Gaussian noise process.

In [Chap. 10](#) Alarcon and Dalmau first review various discrete and fuzzy-based color categorization models and then they focus on a new framework which provides a probabilistic partition of a given color space. The proposed approach combines the color categorization model with a probabilistic segmentation algorithm and generalizes it including the interaction between categories. The effectiveness of the proposed approach is illustrated using various applications including color image segmentation, edge detection, video re-colorization, and object tracking.

In [Chap. 11](#) Kawulok et al. present an overview of skin detection. The Authors focus on approaches based on pixel classification and present a comparative study of various state-of-the-art methods. They give an overview of techniques which model the skin color using a set of fixed rules, as well as those based on machine learning. In the latter case, the Authors report an experimental study which shows the sensitivity of commonly used methods to the number of samples in the training set. Not only are the techniques for skin color modeling explored, but also important approaches toward reducing skin detection errors are presented and validated empirically. In particular, the Authors outline the possibilities of adapting the skin color models to specific lighting conditions or to an individual, whose skin regions are to be segmented. In addition, the Authors present how the textural features and spatial analysis of the skin probability maps can be employed for skin detection.

In [Chap. 12](#) Lee et al. address the issues connected with the employment of skin color as a feature in automatic face detection systems. After providing a general overview of face detection methods utilizing color information, the Authors discuss approaches for modeling skin color distribution in various color spaces, focusing on the influence of illumination conditions on the skin detection results and describe practical applications of skin color classification in high-level image processing systems. The effectiveness of color cues in terms of detection performance and computational efficiency is addressed using two distinct case studies.

[Chapter 13](#) by Jiang et al. completes the volume. The Authors describe a very interesting application of color-based visual saliency in the design of video games. They provide an overview of several state-of-the-art saliency estimation methods and propose novel methods which are evaluated and compared with previously published techniques on an image saliency dataset. The proposed saliency estimation frameworks are applied to the visual game design process. The results demonstrate that the incorporation of color saliency information improves the visual quality of the video games and substantially increases their attractiveness.

As Editors, we hope that this volume focused on low-level color image processing will demonstrate the significant progress that has occurred in this field in



recent years. We also hope that the developments reported in this volume will motivate further research in this exciting field.

M. Emre Celebi  
Bogdan Smolka

# Contents

<b>Automated Color Misalignment Correction for Close-Range and Long-Range Hyper-Resolution Multi-Line CCD Images</b> . . . . .	1
Zhiyu Chen, Andreas Koschan, Chung-Hao Chen and Mongi Abidi	
<b>Adaptive Demosaicing Algorithm Using Characteristics of the Color Filter Array Pattern</b> . . . . .	29
Ji Won Lee and Rae-Hong Park	
<b>A Taxonomy of Color Constancy and Invariance Algorithm</b> . . . . .	55
Dohyoung Lee and Konstantinos N. Plataniotis	
<b>On the von Kries Model: Estimation, Dependence on Light and Device, and Applications</b> . . . . .	95
Michela Lecca	
<b>Impulse and Mixed Multichannel Denoising Using Statistical Halfspace Depth Functions</b> . . . . .	137
Djordje Baljzović, Aleksandra Baljzović and Branko Kovačević	
<b>Spatially Adaptive Color Image Processing</b> . . . . .	195
Johan Debayle and Jean-Charles Pinoli	
<b>Vector Ordering and Multispectral Morphological Image Processing</b> . . . . .	223
Santiago Velasco-Forero and Jesus Angulo	
<b>Morphological Template Matching in Color Images</b> . . . . .	241
Sébastien Lefèvre, Erchan Aptoula, Benjamin Perret and Jonathan Weber	
<b>Tensor Voting for Robust Color Edge Detection</b> . . . . .	279
Rodrigo Moreno, Miguel Angel Garcia and Domenec Puig	
<b>Color Categorization Models for Color Image Segmentation</b> . . . . .	303
Teresa Alarcon and Oscar Dalmau	

**Skin Detection and Segmentation in Color Images** . . . . . 329  
Michal Kawulok, Jakub Nalepa and Jolanta Kawulok

**Contribution of Skin Color Cue in Face Detection Applications** . . . . . 367  
Dohyoung Lee, Jeaff Wang and Konstantinos N. Plataniotis

**Color Saliency Evaluation for Video Game Design** . . . . . 409  
Richard M. Jiang, Ahmed Bouridane and Abbes Amira

**Erratum to: On the von Kries Model: Estimation, Dependence  
on Light and Device, and Applications** . . . . . E1  
Michela Lecca

**Erratum to: Skin Detection and Segmentation in Color Images.** . . . . . E3  
Michal Kawulok, Jakub Nalepa and Jolanta Kawulok

# Automated Color Misalignment Correction for Close-Range and Long-Range Hyper-Resolution Multi-Line CCD Images

Zhiyu Chen, Andreas Koschan, Chung-Hao Chen and Mongi Abidi

**Abstract** Surveillance and inspection have an important role in security and industry applications and are often carried out with line-scan cameras. The advantages of line-scan cameras include hyper-resolution (larger than 50 Megapixels), continuous image generation, and low cost, to mention a few. However, due to the physical separation of line CCD sensors for the red (R), green (G), and blue (B) color channels, the color images acquired by multi-line CCD cameras intrinsically exhibit a color misalignment defect, such that the edges of objects in the scene are separated by a certain number of pixels in the R, G, B color planes in the scan direction. This defect, if not corrected properly, can severely degrade the quality of multi-line CCD images and hence impairs the functionality of the cameras. Current techniques for correcting such color misalignments are typically not fully automated, which is undesirable in applications such as inspection and surveillance that depend on fast unmanned responses. This chapter introduces an algorithm to automatically correct the color misalignments in multi-line CCD images for rotational scans as well as for translational scans. Results are presented for two different configurations of multi-line CCD imaging systems: (a) a close-range multi-line CCD imaging system for inspection applications and (b) a long-range imaging system for surveillance applications. Experimental results show that the two imaging systems are able to acquire hyper-resolution images and the color misalignment correction algorithm can automatically and accurately correct those images for their respective applications.

---

Z. Chen (✉)

Seagate Technology PLC, Bloomington, MN, USA

A. Koschan · M. Abidi

Imaging, Robotics, and Intelligent Systems Laboratory,  
Department of Electrical Engineering and Computer Science,  
The University of Tennessee, Knoxville, TN, USA

C.-H. Chen

Department of Electrical and Computer Engineering,  
Old Dominion University, Norfolk, VA, USA

**Keywords** Color correction · Color misalignment · Hyper-resolution color image · Multi-line CCD camera · Line-scan camera · Image acquisition · Imaging system

## 1 Introduction

Remote sensing, surveillance and inspectional scanning have an important role in security and industry applications and usually require acquiring hyper-resolution (larger than 50 Megapixels) images of a constant stream of materials or landscapes with relative motion. Such applications are often carried out with line-scan cameras because of their advantages, including: (i) hyper-resolution (at least thousands of pixels in one dimension and the size in the other dimension is limited only by the capacity of storage device); (ii) continuous image generation (video stream compared to discrete video frames generated from a frame-based camera); (iii) low cost (compared to still image cameras with a 2-D imaging sensor array that can achieve the same resolution).

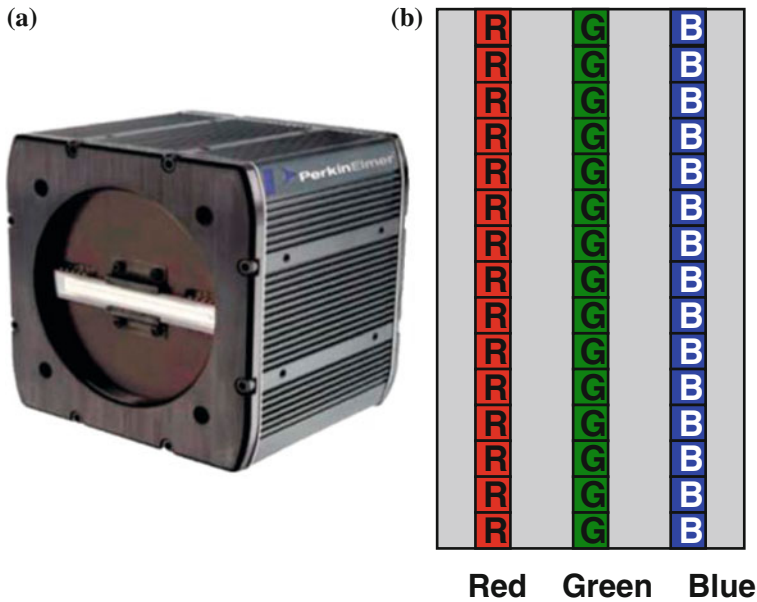
### 1.1 Line-Scan Imaging Sensor and Applications

A line-scan camera is an imaging device generally containing a line-scan imaging sensor chip, supporting electronics and an optical system that includes one or multiple lenses and a focusing mechanism. Gupta et al. introduced a simplified camera model for line-scan imaging sensors [1]. Unlike video cameras and regular still image cameras, a line-scan camera has a single row (1-D) of pixel sensors, instead of a 2-D array of them.

Figure 1a shows a picture of a line-scan camera without the lens and focusing mechanism. Figure 1b illustrates the configuration of the RGB-multi-line CCD sensor on that camera which is capable of capturing color images.

There are two major technologies for solid-state imaging sensors, i.e., CCD (Charge-Coupled Devices) sensors and CMOS (Complementary Metal Oxide Semiconductor) sensors. The charge-coupled device was invented in 1969 at AT&T Bell Laboratories by Willard Boyle and George E. Smith [2–5]. CCDs have been in existence for over four decades and the technology has matured to the point where very large, consistent devices can now be manufactured. Compared to CCD technology, the CMOS imaging sensor technology is not as mature, but it is set to develop rapidly and offer a number of advantages over CCDs in terms of low power, low cost and monolithic integration. Nowadays active pixel CMOS imaging sensors have been optimized for optical sensing and can rival CCD counterparts in most aspects [6, 7].

One image frame captured by a line-scan camera contains only one row of pixels. In order to capture a 2-D image with such a camera, the 1-D frames are continuously fed to a computer that joins them to each other. Due to the much shorter time of transferring 1-D frames instead of 2-D frames out of the imaging chip, line-scan



**Fig. 1** **a** A line-scan CCD camera with three line sensors for color image capture. **b** Illustration of RGB line-scan sensors

cameras are capable of capturing sharp hyper-resolution images of objects passing in front of the camera at high speeds. Therefore, this kind of camera is commonly used in sports events to acquire photo finishes, i.e. to determine the winner when multiple competitors cross the finishing line at nearly the same time. Line-scan CCD cameras have significant advantages and play important roles in many industrial, scientific and military applications. The areas in which line-scan cameras have important applications include, but are not limited to, remote sensing, surveillance, high speed document/film scanning, industrial quality control inspection, surface inspection, racing sports, etc. Common applications of line-scan cameras in the area of remote sensing and surveillance include satellite and aerial imaging, and their applications in the area of industrial quality control inspection include printing inspection, produce and food inspection, textile inspection, etc.

Line-scan technology is capable of capturing data extremely fast and at very high image resolutions. Under such conditions the acquired image data can quickly exceed 100 MB in a matter of seconds. Data coming from the line-scan camera has a frequency at which the camera scans a line, waits, and repeats. The one-dimensional line data from the line-scan camera is commonly collected by image acquisition electronics, e.g., a frame grabber card in a computer, and then processed by the computer to create a 2-D image. The raw 2-D image is then processed by image-processing techniques in order to meet application objectives. Figure 2 illustrates the data path of a line-scan imaging system.

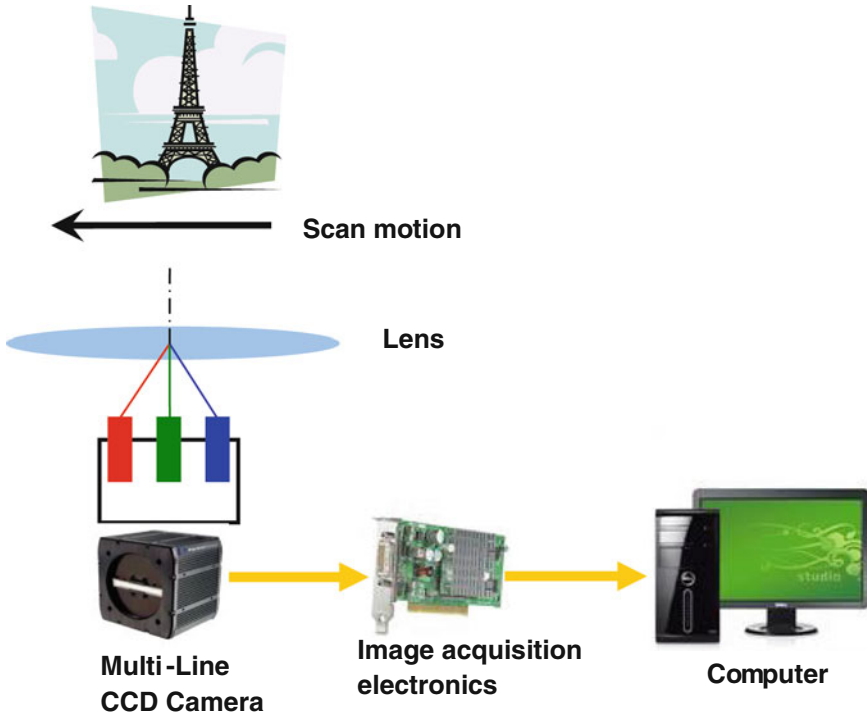


Fig. 2 The data path of a line-scan imaging system

Numerous novel applications of line-scan imaging systems have been reported in the literature. For example, Wilson reported an on-line surface inspection system designed to characterize the rotary screen-print process of applying up to 20 separate colors to a continuous textile web for the textile industry [8]. Reulke et al. developed a mapping method of combining high resolution images acquired by a line CCD camera with depth data acquired by a laser scanner. Application areas are city modeling, computer vision and documentation of cultural heritage [9–12]. Huang et al. developed a rotating-line-camera imaging system for stereo viewing and stereo reconstruction [13, 14]. Maresch et al. used three partially inclined and vertical linear CCD arrays and developed a vehicle-based 3-line CCD camera system for 3D city modeling [15]. Yoshioka et al. developed a vehicle lane change aid system (LCAS) with multi-line CCD sensors [16]. Bowden et al. designed a line-scanned micro Raman spectrometer using a cooled CCD imaging detector to obtain sequences of Raman spectra [17]. Ricny et al. developed an autonomous optoelectronic method of measuring the flying objects track velocity vector using two-line CCD sensors [18]. Kroll et al. developed a system using an 8-bit CCD line scanner for automatic determination of the brightness of star-like objects on a photographic plate [19]. Demircan et al. used a wide angle CCD line camera to measure the bi-directional reflectance distribution function of natural surfaces [20]. Kipman et al. developed a method of

measuring gloss mottle and micro-gloss using a line-scan CCD camera [21]. Rosen et al. used a line-scan CCD image sensor for on-line measurement of red blood cell velocity and microvascular diameter [22].

## ***1.2 Color Misalignment Defect of Multi-Line CCD Images and its Correction***

Hyper-resolution is one of the major advantages of line CCD images. However, due to the physical configuration and characteristics of multi-line CCD sensors, raw output images acquired by multi-line CCD cameras typically exhibit some defects, and may not be usable for desired application purposes. For example, because of the physical separation of line CCD sensors for the red (R), green (G), and blue (B) color channel, the color images acquired by multi-line CCD cameras intrinsically exhibit a color misalignment defect, such that the edges of objects in the scene are separated by a certain number of pixels in the R, G, B color planes in the scan direction. This defect, if not corrected properly, can severely degrade the quality of multi-line CCD images and hence impair the functionality of multi-line CCD cameras. Figure 3 illustrates the creation of color misalignment, and Fig. 4 shows a raw multi-line CCD image with such a defect. This misalignment can be considered a major problem in hyper-resolution multi-line CCD scans and is also called pixel lag [23] or video delay [24]. There are two commonly used methods for correcting color misalignment in multi-line CCD imaging:

- (1) Synchronize the CCD line acquisition rate to the object's moving speed and/or the camera's scan motion [23].
- (2) Set the video delay parameter of the multi-line CCD camera to compensate the target motion for the physical separation of color sensors. When the camera reconstructs the color image, the adjacent color planes are shifted by a certain number of lines that was specified by the video delay parameter [24].

The above methods have significant drawbacks and/or limitations. Both methods are not fully automated. Synchronizing the line acquisition rate to the object's motion is not an easy task. Furthermore, this method puts undesirable constraints on imaging parameters, e.g., line acquisition rate, exposure time, aspect ratio of acquired images, etc. The imaging parameters that synchronize the line acquisition rate to the scan motion speed may create images with undesirable or even unacceptable brightness and aspect ratios. Setting the video delay parameter can avoid putting undesirable constraints on imaging parameters; however, similar to the synchronization method, setting the correct video delay parameter before and/or when imaging is taking place is not an easy task. It is usually done by the user via visual inspection, which is subjective and not accurate, and the acquired images may still exhibit small color misalignment.

Since the current techniques for correcting color misalignments are not fully automated, and may put adverse constraints on imaging parameters, they are undesirable



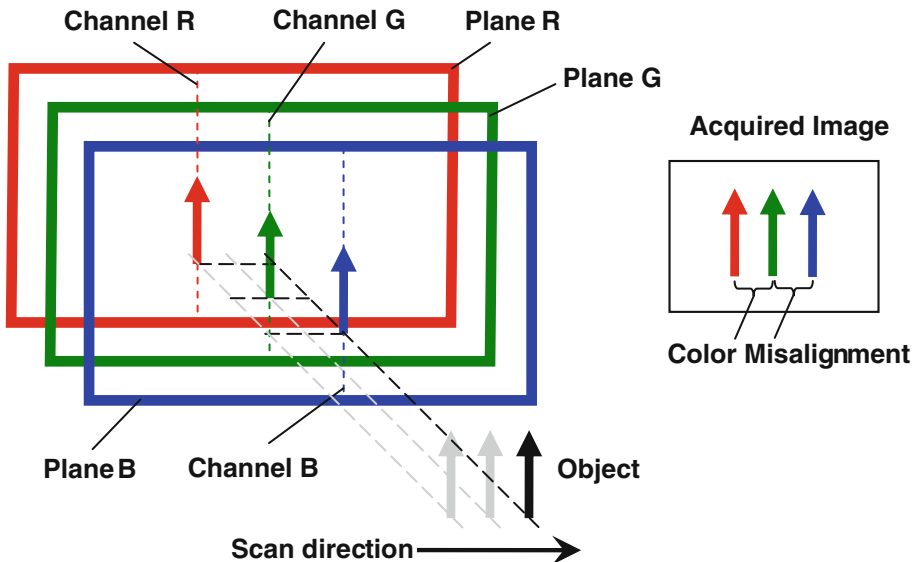


Fig. 3 Illustration of the creation of color misalignment

in applications such as inspection and surveillance that depend on fast unmanned responses. In order to greatly expand the applications of multi-line-scan cameras, it is important to develop a technique that can fully-automatically correct color misalignments in multi-line CCD images and does not put constraints on imaging parameters.

This chapter introduces an algorithm to automatically correct color misalignments in multi-line CCD images for rotational scans as well as for translational scans. Results are presented for two different configurations of multi-line CCD imaging systems: (a) a close-range multi-line CCD imaging system for inspection applications and (b) a long-range multi-line CCD imaging system for surveillance applications. This chapter is organized as follows: Section 2 presents the setup of two of our multi-line CCD imaging systems. Section 3 introduces a fully-automated color misalignment correction algorithm. Experimental results are presented in Sects. 4 and 5 concludes this chapter.

## 2 Multi-Line CCD Imaging Systems

We used a multi-line CCD camera to develop two hyper-resolution imaging systems, i.e., a close-range line-scan imaging system for inspection applications and a long-range system for surveillance applications. In the following, the configurations of the two imaging systems are presented.



Fig. 4 Color misalignment in a multi-line CCD image

## 2.1 Translational Scan and Rotational Scan

In order to create 2-D images, a relative scan motion between the line-scan camera and the scene is necessary during the imaging process. There are two kinds of scan schemes—translational scan and rotational scan. Figures 5 and 6 illustrate two translational scan schemes. Translational scan is more suitable for close-range imaging; since the distance between the object to be imaged and the lens is constant in translational scan, the focus does not need to be adjusted and the image remains focused during the imaging process. For close range imaging, due to narrow depth of field, the object is usually a flat surface or an object with small depth. In the scan scheme illustrated in Fig. 5, the object is located on a platform which can undergo a translational scan motion across the field of view of the line-scan camera, and the scan

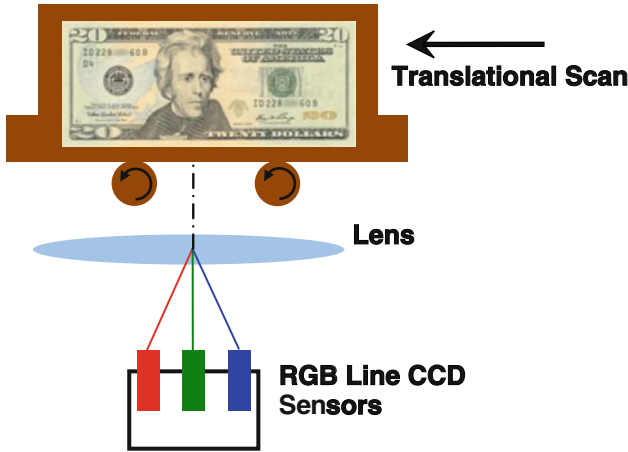


Fig. 5 A close-range imaging system with translational scan scheme

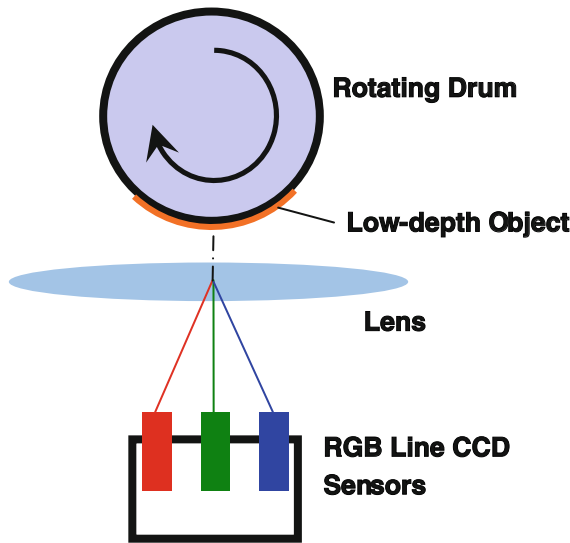


Fig. 6 A close-range imaging system with translational scan schem

speed can be adjusted by the user to change the aspect ratio of acquired images. In the scan scheme illustrated in Fig. 6, a rotating drum is placed in front of the camera lens to provide the scan motion. A flat or low-depth object is attached to the drum surface. Although the drum rotates, the scan motion seen by the line-scan camera is translational.

Figures 7 and 8 illustrate two rotational scan schemes. Rotational scan is more suitable for long-range imaging, since it enables the line-scan camera to scan a wide

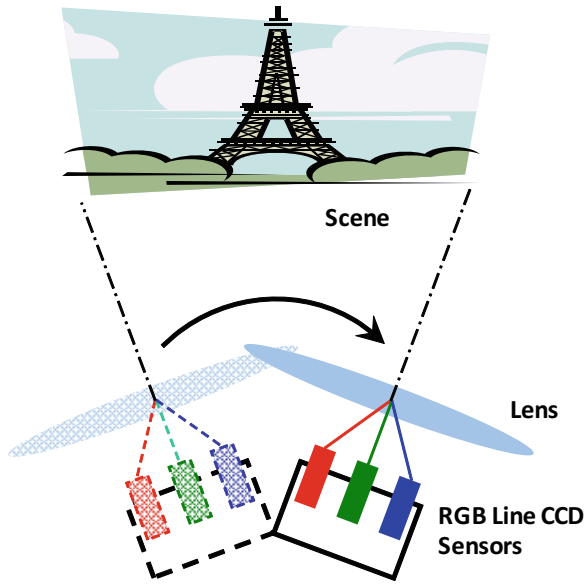


Fig. 7 A long-range imaging system with a rotating camera to provide scan motion

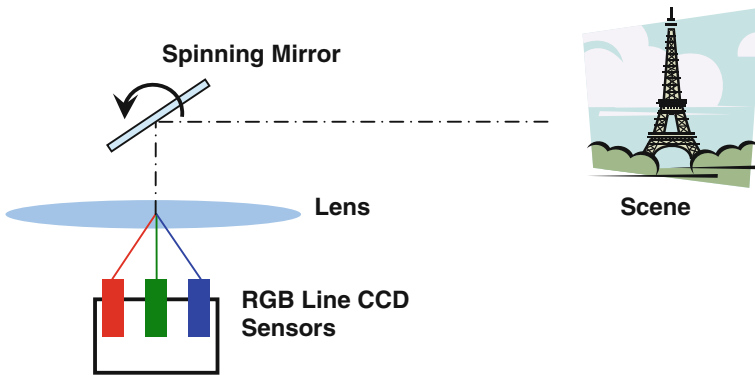


Fig. 8 A long-range imaging system with a rotating camera to provide scan motion

scene with the camera being fixed to one location. There are two ways to provide a rotational scan motion for a long-range imaging system:

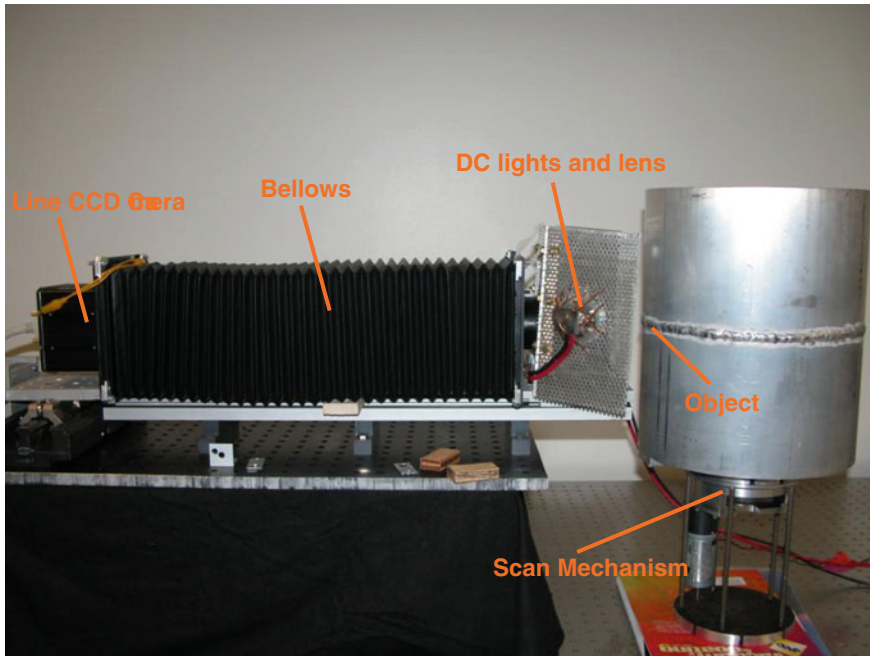
- (1) The camera rotates to sweep across the scene. Figure 7 shows the schematic of a long-range imaging system with a rotating camera to provide scan motion.
- (2) A spinning mirror is placed in front of the camera lens and reflects the moving scene into camera. Figure 8 shows the schematic of a long-range imaging system with a spinning mirror to provide scan motion.

The spinning mirror design has a significant advantage over the rotating camera design, because the mechanical system for spinning a long, narrow light-weight mirror is smaller, lighter and less expensive than that for rotating the entire imaging system; therefore, the system is more mobile and more suitable for outdoor applications. The disadvantage of the spinning mirror design is that it is more susceptible to wind when used for outdoor image acquisitions because of the mirror's light weight and relatively large area. Wind can cause the mirror to vibrate slightly, and slight vibrations of the spinning mirror can severely impair the quality of acquired images because of the hyper-resolution nature of these images. Therefore, in order to minimize the effect of wind and mechanical vibration, the rotation mechanism and the fixture for attaching the mirror to the mechanism need to be carefully designed. In addition, the wind effect can be significantly reduced by placing a glass windshield in front of the spinning mirror.

## ***2.2 Close-Range Multi-Line CCD Imaging System***

We developed a close-range multi-line CCD imaging system incorporating a Perkin-Elmer multi-line-scan camera for inspection applications. A picture of this system is shown in Fig. 9. This imaging system consists of a YD5060 tri-linear CCD color camera, a bellows, a short focal-length (90 mm) lens, DC diffusive illumination lights, and scanning mechanism. These system components are mounted on a metal plate platform for stabilization and easy transportation. The DC diffusive lights provide intense, constant and uniform illumination over the object to ensure that the image formed on sensor lines has sufficient brightness.

The bellows connects to the camera and the lens at two ends. The bellows length is adjustable for changing focusing and magnification. For close-range inspection applications, our goal is to image a small area of interest (e.g., a few centimeters or smaller in one dimension) on the inspected surface with a hyper-resolution of 6,144 pixels in one dimension. Therefore, high magnification is required in order for the lens to create an image of the small area of interest at the camera sensor plane, and the image should have the same number of lines as there are CCD sensor lines. High magnification is achieved with a long bellows length ( $\sim 60$  cm) and a short distance ( $\sim 10$  cm) between the camera lens and the surface to be inspected. The optical magnification of this system is  $6\times$ , resulting in that a pixel on the CCD sensor line corresponds to  $1.5\ \mu\text{m}$  on the object, and this  $1.5\ \mu\text{m}$  is close to the size limit of a feature that can be discerned by visible light because of diffraction limit. Therefore, this imaging system is capable of capturing an object's micron-scale fine details, and it has almost reached the size limit discernible by visible light. A microscope can achieve a similar scale; however, for microscopic imaging, the object to be imaged is often cut into thin slices or small pieces, and this destructive imaging method cannot be accepted for many applications such as weld inspection. Therefore, close-range line CCD imaging has an advantage over microscopic imaging for non-destructive inspection imaging applications. In addition, close-range line-scan imaging does not



**Fig. 9** Example of a close-range imaging system

put a limit on the object's length, and acquires images with higher resolution and faster speed.

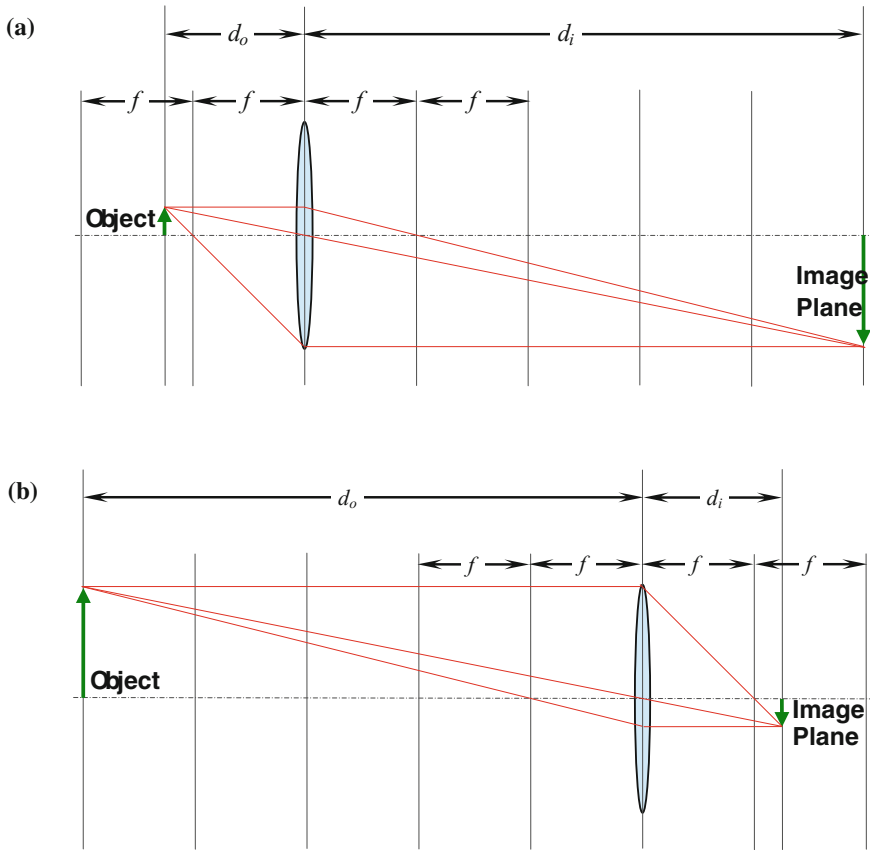
Figure 10a illustrates the optical path of image formation for close-range imaging. The distances between the object and lens, and the lens and image plane, are related by the thin lens formula,

$$\frac{1}{d_o} + \frac{1}{d_i} = \frac{1}{f}, \quad (1)$$

where  $d_o$  is the distance from the object plane to the center of the lens,  $d_i$  is the distance from the image plane to the center of the lens, and  $f$  is the focal length of the lens. Since  $d_o \gg d_i$ , the size of the formed image is much larger than that of the object; it makes this close-range imaging system more suitable for surface inspection applications.

For close-range imaging applications, due to the high magnification and short exposure time (a few milliseconds) used in line-scan CCD imaging, intense illumination is required in order to acquire images with proper brightness.

In our system, two DC projector light bulbs are used to provide this illumination. The light bulbs are located on each side of the camera lens, pointing to the object that is imaged. DC power is necessary in order to provide time-invariant constant illumination for line scan imaging. At one end of the metal base platform there is a small translational scan platform, which is connected to a gear box and driven by a



**Fig. 10** Optics of multi-line CCD imaging systems: **a** Close-range system and **b** Long-range system

computerized stepping motor. The stepping motor is powered and controlled by an electronic control card, which is connected to a PC computer through a serial port. Therefore, the scan direction and speed can be controlled by the user through the computer.

In one of the close-range imaging configurations, the object to be imaged is positioned on the translational scan platform at one end of the base platform, and the multi-line-scan camera is located on the other end. When being imaged, the object moves with the translational scan platform in the direction perpendicular to the optical axis of the line-scan camera. In another close-range imaging setup, the object to be imaged is placed on a spinning drum which is positioned in front of the camera lens, as shown in Fig. 9. When the drum rotates, it moves the object attached to it and provides the translational scan motion for the line-scan camera to take an image. The spin speed of the drum can be adjusted by adjusting the voltage applied on the driving motor.

### 2.3 Long-Range Multi-Line CCD Imaging System

We also developed a long-range multi-line CCD imaging system incorporating the PerkinElmer YD5060 line-scan camera for surveillance and security monitoring applications. A picture of this system is shown in Fig. 11. This imaging system consists of a tri-linear CCD color camera, a bellows, a long focal-length (508 mm) lens, and a geared spinning mirror mechanism for providing rotational scan motion. These system components are mounted on a metal plate platform for stabilization and easy transportation. The entire system can be placed on a cart and is easily transported for outdoor image acquisition. The mirror is mounted on a geared spinning mechanism, which is powered by a 12 V DC motor. The spinning speed can be adjusted by changing the gear ratio. The aspect ratio of acquired images can be changed by adjusting the gear ratio along with camera exposure time.

The bellows connects to the camera and the lens at two ends. The bellows length is adjustable for changing the focus and magnification. For long-range inspection applications, our goal is to image an area of interest in a remote scene with a hyper-resolution of 6,144 pixels in one dimension. This differs from the close-range imaging system, since the scene is far away from the camera; in order to form an image of the scene on the camera sensor plane, a long focal-length lens and a relatively short bellows length ( $\sim 20$  cm) should be used. Figure 10b illustrates the optical path of

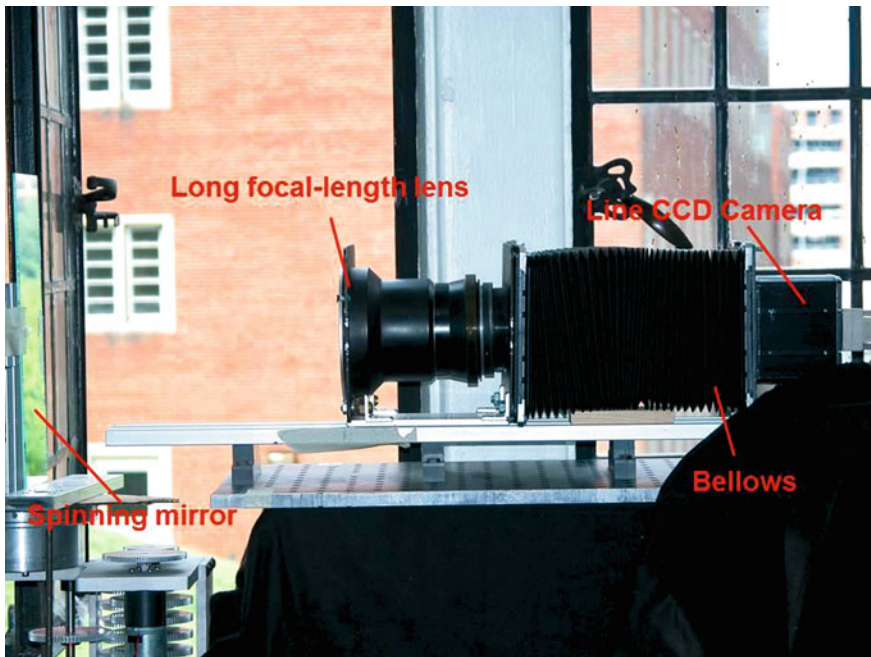


Fig. 11 Long-range imaging system



image formation for long-range imaging. The distances between the object and lens, the lens and image are also related by the thin lens formula, given by Eq.1.

### 3 Color Misalignment Correction

The hyper-resolution images acquired by both multi-line CCD imaging systems mentioned in the previous section intrinsically have a color misalignment defect, which must be fixed before the images can be useful. In this section, we present a technique to automatically correct this defect.

#### 3.1 Formulations of Color Misalignment

Color misalignment in multi-line CCD imaging is measured by the number of pixels and is determined by the relative scan motion between the camera and the object, the optical parameters, the imaging parameters, and the physical distance between adjacent color sensor lines. The pixel displacement between images acquired by different CCD channels is directly related to the CCD line rate (approximately the inverse of exposure time). For example, assume that we are taking images of a tiny point. The point is so small that its image can fall on only one color sensor line at one time. The amount of time that takes the point image to travel from one color sensor line to the next is determined by the relative motion between the camera and the point object, the optical parameters (distances, focus, etc.), and the physical distance between adjacent color channels. During this period of time, the line acquisition rate or the number of exposures of this color channel will determine the number of pixels between the two point images on adjacent color planes in the resulting image.

Figure 12 illustrates the factors that determine color misalignment in multi-line CCD images acquired by the translational scan scheme, and Fig. 13 illustrates the factors for the rotational scan scheme. The primary factors determining color misalignment include:

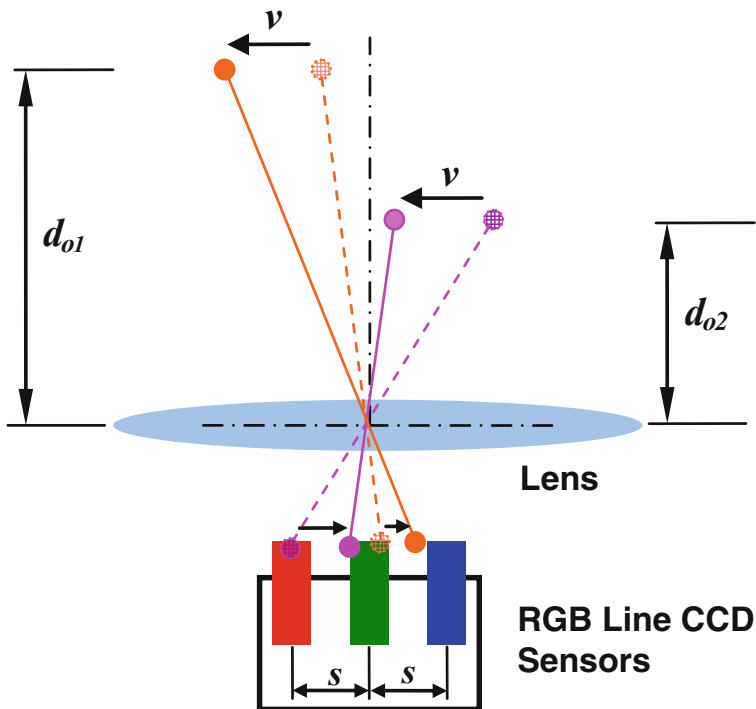
- (1)  $\tau$ , time that a point image in the image plane traverses from one color channel to the next;
- (2)  $R$ , CCD sensor line scan rate or the inverse of exposure time.

The pixel displacement in color misalignment,  $D$ , can be formulated as

$$D = R\tau. \quad (2)$$

The point image traverse time,  $\tau$ , can be determined by secondary factors which include:

- (1)  $d_o$ , distance from the object plane to the center of camera lens;



**Fig. 12** Factors that affect color misalignment in multi-line CCD images acquired by a translational scan scheme

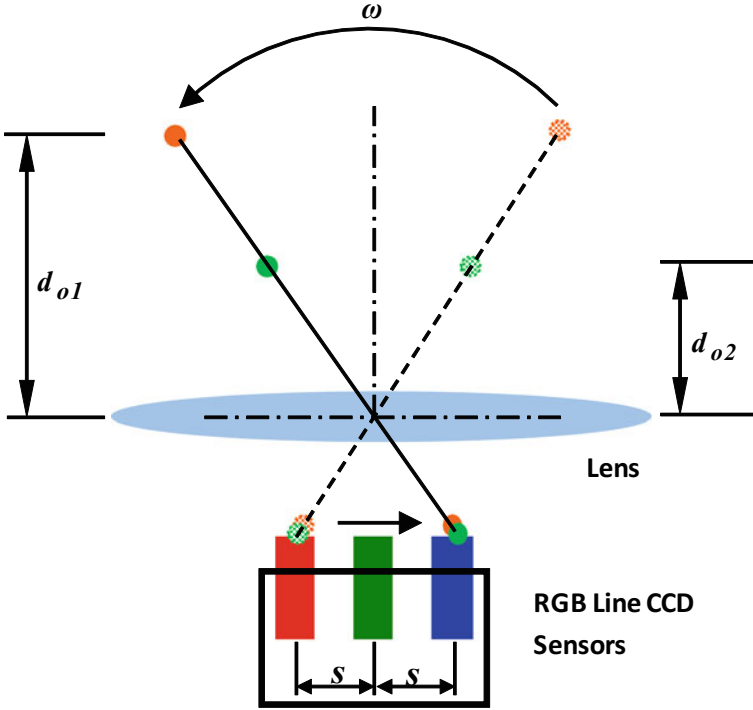
- (2)  $v$ , relative motion speed between the object and the line CCD camera;
- (3)  $f$ , focal length of the camera lens;
- (4)  $s$ , color channel separation.

In a translational scan scheme, the time  $\tau$  that takes a point image to travel from one color channel to the next can be formulated as follows,

$$\tau = \frac{d_i v}{d_o s}, \quad (3)$$

where  $d_o$  is the distance from the object plane to the center of the lens,  $d_i$  is the distance from the image plane to the center of the lens,  $v$  is the translational motion speed of the object, and  $s$  is the separation distance between the centers of adjacent color sensor lines. It can be derived from Eq. 1 that

$$d_i = \frac{d_o f}{d_o - f}. \quad (4)$$



**Fig. 13** Factors that affect color misalignment in multi-line CCD images acquired by rotational scan scheme

After substituting Eqs. 3 and 4 into Eq. 2, we obtain a new formulation of color misalignment which can be easily calculated from camera and imaging parameters as follows,

$$D_T = \frac{f v R}{(d_o - f)s}. \quad (5)$$

By examining Eq. 5, one can conclude that the pixel displacement in color misalignment is the same everywhere for an entire image acquired by the close-range imaging system with a translational scan scheme, because the object to be imaged in such scenario is usually a surface or an object with a shallow depth that is much smaller than  $d_o$ . Therefore, the distance between the object surface and the camera lens,  $d_o$ , is the same for the entire surface when imaging is taking place, so the color misalignment is also the same for different parts of the image.

Similarly, in a rotational scan scheme, the time  $\tau$  that takes a point image to travel from one color channel to the next can be formulated as follows,

$$\tau = \frac{\omega d_i}{s}, \quad (6)$$

where  $\omega$  is the angular speed of the rotational scan motion. Substituting Eqs. 4 and 6 into Eq. 2, we obtain a formulation of color misalignment in long-range multi-line CCD images acquired with a rotational scan scheme as

$$D_R = \frac{f\omega d_o R}{(d_o - f)s}. \quad (7)$$

It can be noted, that for a long-range imaging system the object distance is much larger than the focal length of the camera lens, and Eq. 7 can be approximated as

$$D_R \approx f\omega R/s, \text{ for } d_o \gg f. \quad (8)$$

Equation 8 indicates that color misalignment in long-range multi-line CCD images is independent of the object distance; therefore, different objects of different distances from the camera in a scene would have the same color misalignment in the same image. An experiment was conducted to test the validity of Eq. 8. Figure 14 shows a long-range multi-line CCD image, and the color misalignment values for objects of different distances in the image are listed in Table 1. It can be seen that those objects of different distances from the camera have exactly the same pixel displacement, and the observation results agree well with the above theoretical analysis.

Therefore, we can conclude that the pixel displacement in color misalignment is the same for an entire image in almost all circumstances for the two types of multi-line CCD imaging systems that we developed. This makes the task a lot easier to develop an algorithm to fully automatically and accurately detect and correct the color misalignment in acquired multi-line CCD images.

### 3.2 Correction of Color Misalignment

Since the pixel displacement in color misalignment is the same for an entire image, in order to automatically correct this misalignment in multi-line CCD images we must develop a method to automatically detect the amount of displacement and use this value to shift the R, G, B color planes to correct the color misalignment. This method is applied after the image acquisition process is completed, thereby putting

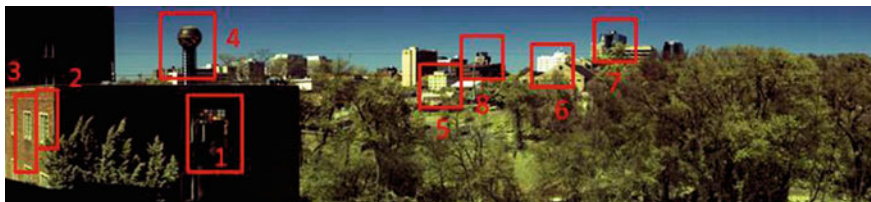


Fig. 14 Test image for studying the relationship between object distance and color misalignment

**Table 1** Color misalignment for objects of different distances from the camera

Patch #	Object distance (m)	Color misalignment (pixels)
1	5	8
2	15	8
3	20	8
4	900	8
5	1200	8
6	1300	8
7	1500	8
8	1600	8

no constraints on imaging parameters. In this way, desired imaging parameters can be used to obtain images with desirable aspect ratio, brightness, contrast, etc. An algorithm developed to automatically correct color misalignment in multi-line CCD images is described as follows:

1. Slice the multi-line-scan RGB image into three color planes – R, G, B.
2. Calculate an estimate of the pixel displacement of color misalignment,  $D$ , according to Eqs. 5 or 8. Then shift the R-plane and B-plane, with all possible displacements in the scan direction within a specified range  $[-(1 + \delta)D, -(1 - \delta)D]$  or  $[(1 - \delta)D, (1 + \delta)D]$ , in the anticipated direction of color misalignment, where  $\delta$  is a threshold chosen by user. The shift tends to realign the R-plane and B-plane with the G-plane.
3. For each displacement, calculate the gray-level distances between adjacent color planes for all displacements with the following formula

$$\begin{cases} D_{RG}(d_R) = \sqrt{\sum_{x,y} (I_R(d_R, x, y) - I_G(x, y))^2} \\ D_{GB}(d_B) = \sqrt{\sum_{x,y} (I_G(d_B, x, y) - I_B(x, y))^2} \end{cases} \quad (9)$$

$$\forall d_R, d_B \in [-(1 + \delta)D, -(1 - \delta)D] \text{ or } [(1 - \delta)D, (1 + \delta)D]$$

where  $I_*(x, y)$  is an original color-plane image,  $I_*(d_*, x, y)$  is a color-plane image shifted with a displacement of  $d_*$ ,  $*$  represents  $R$  or  $B$ , and  $x, y$  are pixel coordinates.

4. Find the minimum  $D_{RG}(d_R)$  and  $D_{GB}(d_B)$ . The corresponding displacements,  $d_{R-Opt}$  and  $d_{B-Opt}$ , are the correct color misalignment in the corresponding image.
5. Correct the color misalignment in the image by shifting the R-, B-plane by the corresponding detected pixel displacement,  $d_{R-Opt}$  and  $d_{B-Opt}$ , and superimpose them with the G-plane to reconstruct the color-misalignment-corrected image.

In Step 3 of the above procedure, instead of using gray-level distances between adjacent color planes as a criterion, other criteria can be applied, e.g., cross correlation of adjacent color channels. If cross correlation is used in Step 4, the correct

pixel displacement in color misalignment corresponds to the maximum cross correlation. Based on extensive tests, we found that cross correlation gives the same results as the gray-level distance between color planes. It is also worth noting that, in Step 2, the pixel displacement of the color misalignment does not have to be estimated according to Eqs. 5 or 8 as long as the search range of the pixel displacement is sufficiently large. However, a larger search range might significantly increase the processing time, given the hyper-resolution of multi-line CCD images. Figure 15 illustrates the proposed method of color misalignment correction

## 4 Experimental Results

The multi-line CCD camera used in our imaging systems is a high-performance PerkinElmer YD5060 tri-linear digital line-scan color camera. The output speeds are up to 90 MHz (30 MHz per output, each output corresponding to either the red, green, or blue color channel), pixel resolution of 6,144, and line scan rates up to 4.88 kHz. The camera features a geometrically precise photodiode CCD image sensor, with  $10\ \mu\text{m}$  square photo-elements. Line spacing between the color-filtered linear rows is  $40\ \mu\text{m}$  [24].

Color separation and imaging are accomplished through the tri-linear image sensor in the YD5060. However, given the  $40\ \mu\text{m}$  center-to-center spacing between the color lines, the image must be reconstructed to correct color misalignment and combine the colors into a usable image. The YD5060 provides a functionality that allows the user to set a delay in the camera, so as to synchronize the camera to its target. Delay can be set from +15 to -15 lines, allowing the camera to image in either direction,

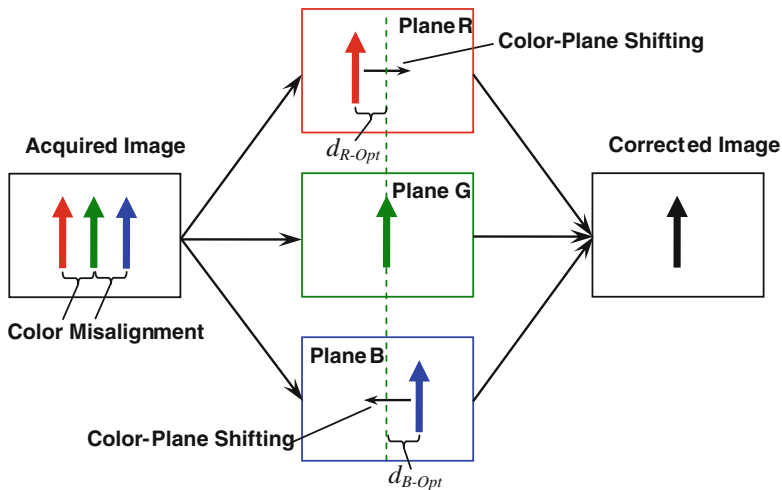
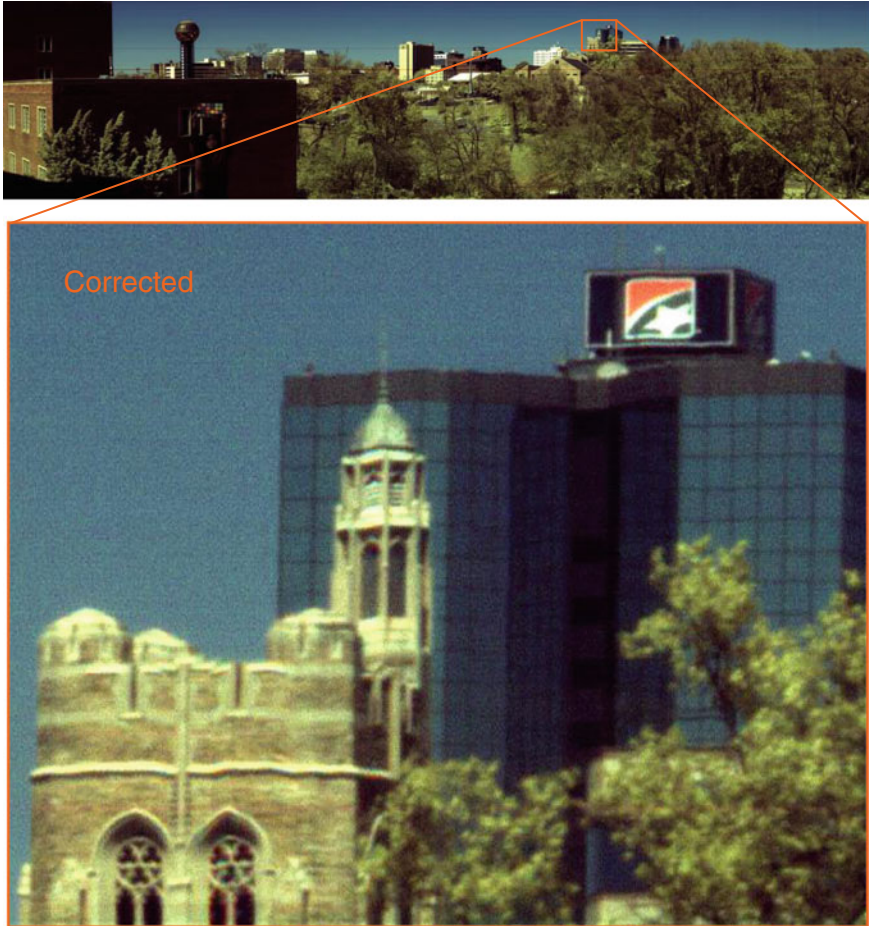


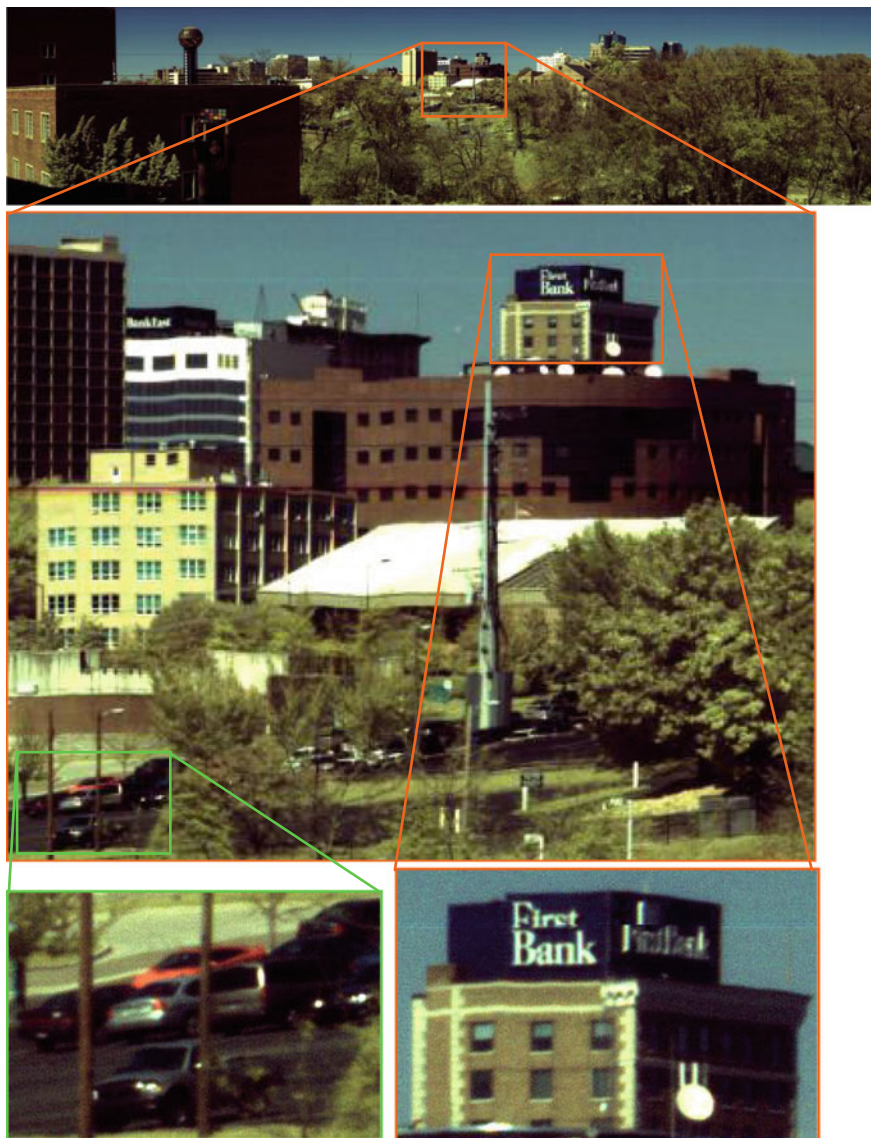
Fig. 15 Illustration of color misalignment correction



**Fig. 16** A multi-line CCD image with corrected color misalignments

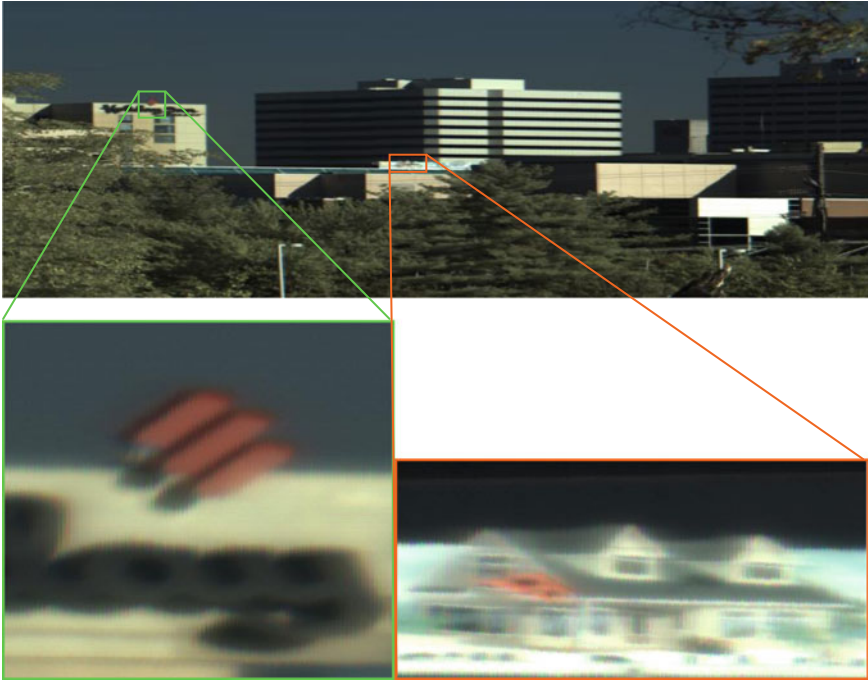
i.e., red-green-blue, or blue-green-red. As discussed in Sect. 1 of this Chapter, the method of setting a video delay parameter is not automated, and therefore not suitable for applications such as inspection and surveillance that require acquisition of vast quantity of hyper-resolution images at high speed with fast unmanned responses. In addition, the delay limit of  $\pm 15$  lines on the YD5060 would fail the correction if the pixel displacement in color misalignment exceeds the limit.

In this section, hyper-resolution images acquired by our two multi-line CCD imaging systems are presented after color misalignments are corrected by applying the automated technique introduced in the previous section.



**Fig. 17** Long-range multi-line scan color image of downtown Knoxville as seen from the University of Tennessee, Knoxville. The resolution of the image is  $4k \times 18k$  pixels

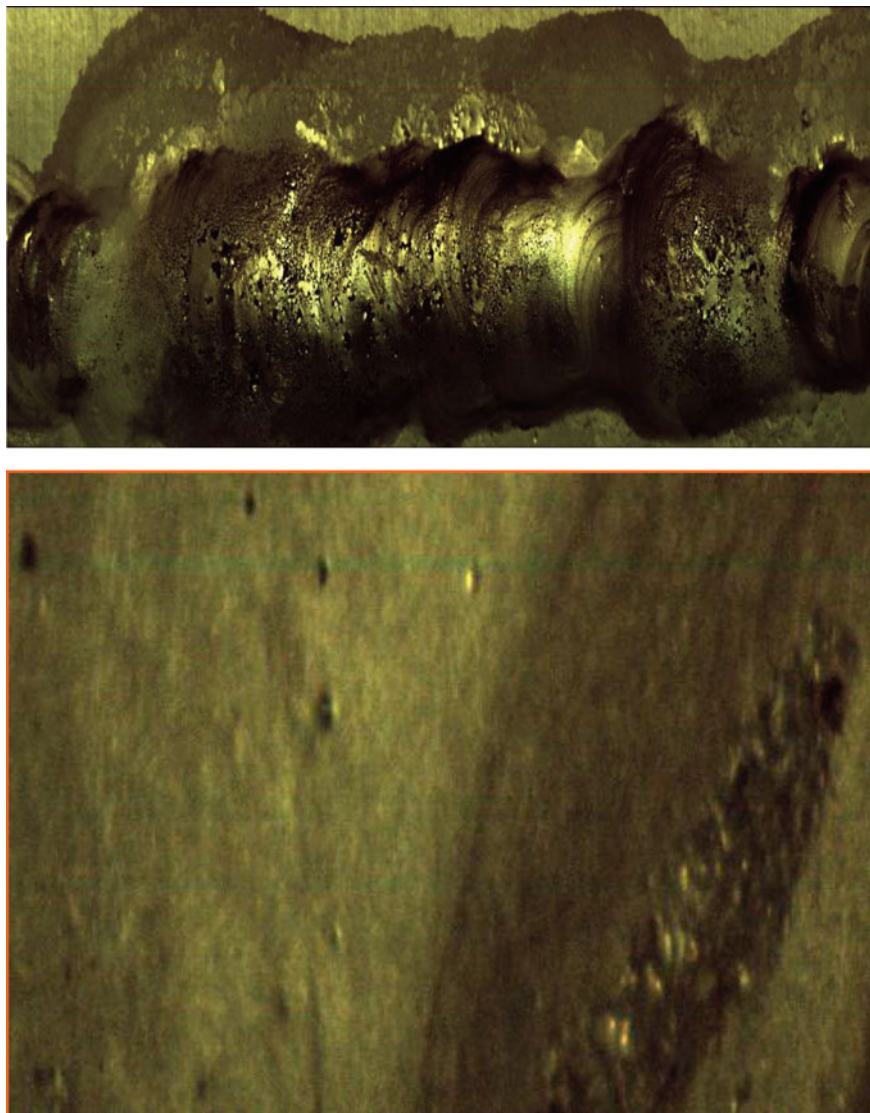




**Fig. 18** Long-range multi-line scan image of downtown Knoxville, acquired with a lens with a longer focal-length (508 mm). The resolution of the image is  $4\text{k} \times 18\text{k}$  pixels

#### ***4.1 Corrected Images Acquired by the Long-Range Multi-Line Imaging System***

Figure 16 shows a long-range multi-line scan color image of downtown Knoxville as seen from the University of Tennessee, Knoxville. The resolution of the image is  $4\text{k} \times 18\text{k}$  pixels. The raw uncorrected version of this image has been shown in Fig. 4, which demonstrates severe color misalignment. It can be seen that, in the same magnified section in Fig. 16, color misalignment in the image has been satisfactorily corrected. Figure 17 shows the same corrected long-range image with different zoom-in patches, in which observers can easily recognize the cars in the parking lot and the signs on top of the building, which are not discernable when looking at the full image. Such images are suitable for surveillance applications. Figure 18 shows another long-range multi-line scan image of downtown Knoxville. When acquiring this image, a lens with a longer focal-length (508 mm) was installed on the long-range imaging system. Therefore, the acquired image has a higher magnification and narrower field of view, and covers only a small portion of downtown Knoxville. The resolution of the image is  $6\text{k} \times 18\text{k}$  pixels, and the magnified portions of the image also reveal fine details.



**Fig. 19** Close-range multi-line CCD image of a weld with a shiny surface, color misalignment has been corrected. The resolution of the image is  $6\text{ k} \times 12\text{ k}$  pixels



**Fig. 20** Close-range multi-line CCD image of the tail of a silver dollar, color misalignment has been corrected. The resolution of the image is  $6k \times 8k$  pixels



**Fig. 21** Close-range multi-line CCD image of a two-dollar bill and several coins. **a** Before color calibration. **b** After color calibration. The resolution of the image is  $6k \times 10k$  pixels

#### ***4.2 Corrected Images Acquired by the Short-Range Multi-Line Imaging System***

Figure 19 shows a close-up image of a weld acquired by the close-range multi-line CCD imaging system. The resolution of the image is  $6k \times 12k$  pixels, and the magnified section reveals features or defects in the weld of size of only  $15 \mu\text{m}$ . Such images are suitable for surface inspection applications. Figure 20 shows a close-up image of a silver dollar coin acquired by the close-range imaging system. The

resolution of the image is  $6\text{ k} \times 8\text{ k}$  pixels, and the magnified section reveals defects on the surface of the coin. The width of the cracks is less than  $16\ \mu\text{m}$ .

### ***4.3 Color Calibration of Multi-Line CCD Images***

Due to the non-ideal responses of CCD sensors to illumination, the colors in acquired multi-line CCD images are generally inaccurate, and somewhat different from the real colors of imaged objects. The color deviations could be significant under extreme imaging conditions like those in close-range multi-line CCD imaging, where illumination is often low, and exposure time is short. Therefore, it is generally necessary to calibrate the colors of acquired CCD images. This can commonly be done by including a Macbeth color chart in the scene when imaging, and using the color calibration matrix generated from camera responses to Macbeth color chart to calibrate the colors in the acquired image.

We use the standard color calibration procedure to calibrate acquired multi-line CCD images, after they are first corrected for color misalignment. Figure 21 shows a close-range multi-line CCD image of a two-dollar bill and several coins before and after color calibration. It can be seen that the colors in the calibrated images are closer to the actual colors of the imaged objects.

## **5 Conclusion**

We introduced an algorithm that can fully automatically correct color misalignments in multi-line CCD images for rotational scans as well as for translational scans. Results were presented for two different configurations of multi-line CCD imaging systems: (a) a close-range multi-line CCD imaging system for inspection applications and (b) a long-range multi-line CCD imaging system for surveillance applications. Experimental results showed that the two imaging systems are able to acquire hyper-resolution images and the color misalignment correction algorithm can automatically and accurately correct those images for respective applications. The presented algorithm improves the suitability of multi-line-scan imaging technology for applications such as inspection and surveillance that require acquisition of vast quantity of hyper-resolution images at high speed with fast unmanned responses.

**Acknowledgments** This work was supported in part by the University Research Program in Robotics under Grant DOE-DE-FG52-2004NA25589 and in part by the U.S. Air Force under Grant FA8650-10-1-5902. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the U.S. Government.

## References

1. Gupta R, Hartley R (1997) Linear pushbroom cameras. *IEEE Trans Pattern Anal Mach Intell* 19(9):963–975
2. Boyle W, Smith G (1974) Three dimensional charge coupled devices. US Patent 3,796,927, 1 Mar 1974
3. Boyle W, Smith G (1974) Buried channel charge coupled devices. US Patent 3,792,322, 24 Oct 24 1974
4. Janesick J, (2001) Scientific charge-coupled devices. SPIE Press, pp. 4, ISBN 978-0-8194-3698-6
5. Boyle W, Smith G (1970) Charge coupled semiconductor devices. *Bell Sys Tech J* 49(4):587–593
6. Blanc N (2001) CCD versus CMOS—has CCD imaging come to an end? *Photogram Week* 1:131–137
7. Taylor S (1998) CCD and CMOS imaging array technologies: technology review. *Relatorio Tecnico EPC-1998-106*, Cambridge Laboratory
8. Wilson A (1999) Textile-inspection system measures colors accurately. *Vis Syst Des* 10(4)
9. Reulke R, Wehr A, Griesbach D (2003) High resolution mapping CCD-line camera and laser scanner with integrated position and orientation system. *Int Arch Photogrammetry, Remote Sens Spat Inf Sci (IAPRS)*, vol 35, part B3. pp 72–77
10. Reulke R, Wehr A (2004) Mobile panoramic mapping using CCD-line camera and laser scanner with integrated position and orientation system. *Int Arch Photogrammetry, Remote Sens Spat Inf Sci (IAPRS)*, vol 34, part 5/W16
11. Scheibe K, Korsitzky H, Reulke R (2001) Eyescan—a high resolution digital panoramic camera. In: *Proceedings of the robot vision*, Auckland, pp 77–83
12. Reulk R, Scheele M (1997) CCD-line digital imager for photogrammetry in architecture. *Int Arch Photogrammetry, Remote Sens Spat Inf Sci*, vol 32, part 5C1B, pp 195
13. Huang F, Wei S.K., Klette R (2006) Rotating line cameras: model and calibration. *IMA Preprint Series # 2104*, Minnesota (Mar 2006)
14. Huang F, Wei S.K, Klette R (2006) Rotating line cameras: epipolar geometry and spatial sampling. *IMA Preprint Series # 2105*, Minnesota, (Mar 2006)
15. Maresch M, Duracher P (1996) The geometric design of a vehicle based 3 line CCD camera system for data acquisition of 3D city models. *Int Arch Photogrammetry Remote Sens* 31:121–127
16. Yoshioka T, Nakaue H, Uemura H (1999) Development of detection algorithm for vehicles using multi-line CCD sensor. In: *Proceedings of the IEEE international conference image process (ICIP 1999)*, vol 4, pp 21–24
17. Bowden M, Gardiner DJ, Rice G, Gerrard DL (1990) Line-scanned micro Raman spectroscopy using a cooled CCD imaging detector. *J Raman Spectrosc* 21(1):37–41
18. Ricny V, Mikulec J (1994) Measuring flying object velocity with CCD sensors. *IEEE Aerosp Electron Syst Mag* 9(6):3–6
19. Kroll P, Neugebauer P (1993) Brightness determination on photographic plates using a CCD line scanner. *Astron Astrophys* 273(1):341–348
20. Demircan A, Schuster R, Radke M, Schönermark M, Röser HP (2000) Use of a wide angle CCD line camera for BRDF measurements. *Infrared Phys Tech* 41(1):11–19
21. Kipman Y, Mehta P, Johnson K, Wolin D (2001) A new method of measuring gloss mottle and micro-gloss using a line-scan CCD-camera based imaging system. In: *Proceedings of the international conference digital printing technologies*, pp 714–717
22. Rosen B, Paffhausen W (1993) On-line measurement of microvascular diameter and red blood cell velocity by a line-scan CCD image sensor. *Microvasc Res* 45(2):107–121
23. ALGE-Timing Co (2005) Introduction of the ALGE 3-line-CCD system. <http://www.alge-timing.com/alge/download/brochure/optic/optic-3-ccd-sensor-pe.pdf>
24. Perkin Elmer Optoelectronics Inc (2013) YD5000 trilinear CCD color camera datasheet. [http://alacron.com/clientuploads/directory/Cameras/PERKINELMER/yd5000series\\_data.pdf](http://alacron.com/clientuploads/directory/Cameras/PERKINELMER/yd5000series_data.pdf)

# Adaptive Demosaicing Algorithm Using Characteristics of the Color Filter Array Pattern

Ji Won Lee and Rae-Hong Park

**Abstract** Generally, the color filter array (CFA) image is interpolated considering the correlation between color channels. Previous works first interpolate the green (G) signal, and then obtain the differences between the R/B signal and the reference signal (the initial interpolated G signal). To determine the direction of interpolation, the proposed method computes the horizontal/vertical absolute inter-channel differences directly computed from the CFA image. Then, three color components (R/G/B) are interpolated along the estimated horizontal/vertical directions considering the differences of absolute inter-channel differences. Comparative experiments using 24 test images with six conventional demosaicing algorithms show the effectiveness of the proposed demosaicing algorithm in terms of the peak signal to noise ratio, structural similarity, and subjective visual quality.

**Keywords** Absolute inter-channel difference · Color filter array · Demosaicing · Real-time processing · Single-sensor

## 1 Introduction

Most of digital cameras and camcorders capture color images through a single-sensor of color charge coupled device (CCD) or complementary metal oxide semiconductor (CMOS). The acquired single-sensor color image is the mosaiced image according to a color filter array (CFA) pattern [1], which has a single known color value at each pixel with two unknown (missing) color values. Recently, for minimizing the

---

J. W. Lee · R.-H. Park (✉)

Department of Electronic Engineering, Sogang University, 35 Baekbeom-ro, Mapo-gu, Seoul 121-742, Korea

e-mail: rhpark@sogang.ac.kr

J. W. Lee

e-mail: jiwonleenk@gmail.com

demosaicing error such as false color, zipper effect, and so on, the new CFA pattern is developed based on the frequency structure [2]. The new CFA pattern leads to better image quality of the demosaiced images. However, this needs a specific demosaicing algorithm for the optimal CFA.

The conventional CFA pattern is composed of the red (R), green (G), and blue (B) signals, where each signal has different horizontal/vertical subsampling factor. The decimated G signal has twice as many pixels as the decimated R/B signals. The two R/B colors alternate with G color (G and R, or G and B) along each column or row. The demosaicing of the CFA pattern, single-sensor color image, is still an important topic in digital color imaging and fast-growing of single-sensor consumer electronic devices, such as digital still and video camera, mobile phone, and so on [3].

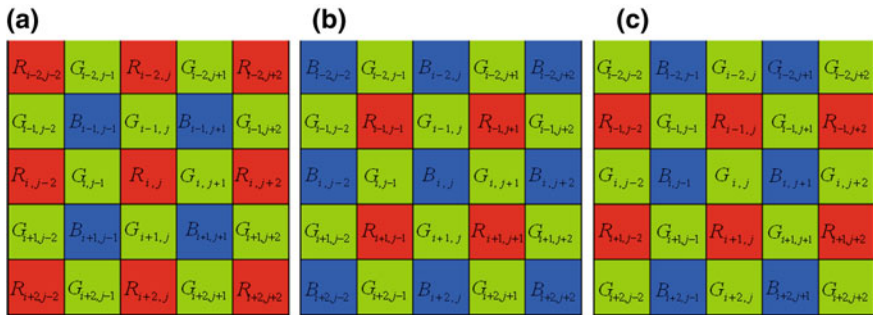
Demosaicing, the CFA interpolation, is the process that reconstructs the unknown  $R$ ,  $G$ , and  $B$  components to generate a full-color image. The two unknown colors at each pixel are estimated by various demosaicing algorithms. The demosaicing algorithms reconstruct a full-color image from the incomplete data (with two missing color data at each pixel). These reconstruction algorithms use the inter- and intra-channel correlation of the available data in  $R/G/B$  signals.

The previous adaptive demosaicing algorithms consider the color correlation and the local characteristics of an image. The adaptive algorithms such as the edge-directed interpolation [4, 5] and various adaptive weighted-sum interpolations [6–19] use adaptive weights for interpolation of unknown pixel values. In the high-frequency region, where the artifacts such as zipper effects, false color, and aliasing are shown, the color difference signals ( $R-G$  and  $B-G$ ) have the high correlation. For reducing the demosaicing artifacts, iterative algorithms [8–10] use the inter- and intra-channel correlation with the appropriate stopping criterion. The stopping criterion determines the quality of the demosaiced image and thus is to be appropriately chosen for reducing zipper artifacts and false color. Edge-sensing algorithms [11–17] using a color-difference model or color-ratio model preserve edges by interpolating along the edges, preventing the interpolation across the edges. To estimate the unknown color pixel values, these algorithms consider the spatial correlation between neighboring pixels and then choose a suitable direction of interpolation with the neighboring pixels. Several of them have a refining step as postprocessing to improve the demosaiced image.

Many conventional demosaicing algorithms require iterations and multi-level interpolations, complex processing, which are not suitable for real-time processing. Also, these conventional demosaicing algorithms generally produce high-quality demosaiced images, especially they are good at reconstructing high-frequency regions such as sharp edges. However, some artifacts such as zipper effect and color artifact are generated near sharp color edges of the demosaiced image.

The proposed algorithm simplifies the demosaicing process, which consists of three steps: edge direction detection, adaptive interpolation, and refinement. Each step is processed considering computational complexity to improve the image quality, speed, and memory usage. In the proposed algorithm, we interpolate three color components in the CFA image, with absolute differences of absolute inter-channel differences (ADAIDs) that are directly computed from the CFA image by considering





**Fig. 1**  $5 \times 5$  Bayer CFA patterns with center pixels of **a** red, **b** blue, and **c** green channels

the characteristics of the CFA pattern. The proposed algorithm reduces the zipper effect around vivid color edges and textures. To evaluate the performance of the proposed algorithm, six conventional demosaicing algorithms [8, 9, 11, 15, 16, 20] are simulated. Experimental results are compared in terms of the peak signal to noise ratio (PSNR), the structural similarity (SSIM) [21, 22], and subjective visual quality. The performance of the proposed algorithm reduces well the zipper effects around color edges and textures.

The rest of the paper is organized as follows. Section 2 describes conventional demosaicing algorithms. Section 3 proposes a demosaicing algorithm using ADAIDs, which are directly computed from the CFA image by considering the characteristics of the CFA pattern. Experimental results with 24 Kodak images are presented in Section 4, showing the effectiveness of the proposed demosaicing algorithm. Finally, Section 5 concludes the paper.

## 2 Conventional Demosaicing Algorithms

Figure 1 shows the Bayer CFA pattern, in which half of the total number of pixels is green and the remaining pixels are equally assigned to red or blue. Each pixel in this pattern is sensitive only to one color:  $R$ ,  $G$ , or  $B$ . Therefore, color images captured using this pattern are to be interpolated in three color channels to generate full-color images, which is called CFA demosaicing. A large number of demosaicing algorithms have been proposed [3–19].

The basic idea of the adaptive weighted-sum algorithms [4, 9, 15] is to estimate the local variance from a mosaiced image and then to utilize the local covariance for demosaicing. The conventional algorithms mostly obtain the demosaiced image using a weighted sum of the neighboring pixel intensities.

Figure 1a, b and c show the  $5 \times 5$  Bayer CFA patterns, which have an  $R$ ,  $B$ , and  $G$  pixel at the center of the pattern, respectively. With the CFA pattern of Fig. 1a, unknown  $G$  pixel value is estimated first by considering the direction of interpolation using horizontal and vertical gradients, which are respectively defined as

$$\Delta H_{i,j} = |G_{i,j-1} - G_{i,j+1}| + |2R_{i,j} - R_{i,j-2} - R_{i,j+2}| \quad (1)$$

$$\Delta V_{i,j} = |G_{i-1,j} - G_{i+1,j}| + |2R_{i,j} - R_{i-2,j} - R_{i+2,j}| \quad (2)$$

where  $G_{p,q}$  and  $R_{p,q}$  represent the known  $G$  and  $R$  pixel values at  $(p, q)$  in the CFA pattern, respectively. Using the horizontal gradient  $\Delta H_{i,j}$  and the vertical gradient  $\Delta V_{i,j}$ , the unknown  $G$  pixel value  $\hat{G}_{i,j}$  is computed as [4], [15]

$$\hat{G}_{i,j} = \begin{cases} \frac{G_{i,j-1} + G_{i,j+1}}{2} + \frac{2R_{i,j} - R_{i,j-2} - R_{i,j+2}}{4}, & \text{if } \Delta H_{i,j} < \Delta V_{i,j} \\ \frac{G_{i-1,j} + G_{i+1,j}}{2} + \frac{2R_{i,j} - R_{i-2,j} - R_{i+2,j}}{4}, & \text{if } \Delta H_{i,j} > \Delta V_{i,j} \\ \frac{G_{i-1,j} + G_{i+1,j} + G_{i,j-1} + G_{i,j+1}}{4} + \frac{4R_{i,j} - R_{i-2,j} - R_{i+2,j} - R_{i,j-2} - R_{i,j+2}}{8}, & \text{otherwise.} \end{cases} \quad (3)$$

Figure 1b is similar to Fig. 1a, only with  $R_{p,q}$  and  $B_{p,q}$  interchanged. Thus, with the CFA pattern of Fig. 1b, unknown  $G$  pixel value  $\hat{G}_{i,j}$  is estimated using (1)–(3), only with  $R_{p,q}$  replaced by  $B_{p,q}$ . Figure 1c shows the Bayer CFA pattern at  $G$  center pixel with unknown  $R$  and  $B$  pixel values.

The unknown  $R$  and  $B$  pixel values are estimated using the interpolated  $G$  pixel values, under the assumption that the high-frequency components have the similarity across three color components,  $R$ ,  $G$ , and  $B$ .

To evaluate the performance of the proposed algorithm, six conventional demosaicing algorithms [8, 9, 11, 15, 16, 20] are simulated.

Gunturk et al.'s algorithm [8] is a projection-onto-convex-set based method in the wavelet domain. This method is proposed to reconstruct the color channels by two constraint sets. The first constraint is defined based on the known pixel values and prior knowledge about the correlation between color channels. The second constraint set is to force the high-frequency components of the  $R$  and  $B$  channels to be similar to that of the  $G$  channel. This method gives high PSNRs for a set of Kodak images. However, the algorithm gives poor visual quality near line edges such as fences and window frames because it has difficulty in converging to a good solution in the feasibility set by iteratively projecting onto given constraint sets and by iteratively updating the interpolated values.

Li's algorithm [9] uses the modeling inter-channel correlation in the spatial domain. This algorithm successively interpolates the unknown  $R$ ,  $G$ , and  $B$  pixel value enforcing the color difference rule at each iteration. The spatially adaptive stopping criterion is used for demosaicing artifacts suppression. This algorithm gives high PSNRs for a set of Kodak images, however sometimes with poor visual quality near line edges such as fences and window frames like Gunturk et al.'s algorithm, because of difficulty in selecting the appropriate stopping criterion of updating procedures.

Lu and Tan's algorithm [11] improves the hybrid demosaicing algorithm that consists of two successive steps: an interpolation step to reconstruct a full-color image

and a refinement step to reduce visible demosaicing artifacts (such as false color and zipper artifacts). In the interpolation step, every unknown pixel value is interpolated by properly combining the estimates obtained from four edge directions, which are defined according to the nearest same color value in the CFA pattern. The refinement step is selectively applied to demosaicing artifact-prone region, which is decided by the discrete Laplacian operator. Also it proposes a new image measurement for quantifying the performance of demosaicing algorithms.

Chung and Chan's algorithm [15] is an adaptive demosaicing algorithm, which can effectively preserve the details in edge/texture regions and reduce the false color. To estimate the direction of interpolation for interpolating the unknown  $G$  pixel values, this algorithm uses the variance of the color differences in a local region along the horizontal and vertical directions. The interpolation direction is the direction that gives the minimum variance of color difference. After interpolation of  $G$  channel, the known  $R$  and  $B$  pixel values are estimated at  $G$  pixel sampling position in the CFA pattern.

Menon et al.'s algorithm [16] is based on directional filtering and a posteriori decision, where the edge-directed interpolation is applied to reconstruction of a full-resolution  $G$  component. This algorithm uses a five-tap FIR filter to reconstruct the  $G$  channel along horizontal and vertical directions. Then  $R$  and  $B$  components are interpolated using the reconstructed  $G$  component. In the refinement step, the low and high frequency components in each pixel are separated. The high frequency components of the unknown pixel value are replaced with the high frequency of the known components.

The proposed demosaicing algorithm gives better visual quality near line edges such as window frames and fences than conventional algorithms, because the edge direction is detected by absolute differences of absolute inter-channel differences (ADAID) between adjacent pixels in the CFA image.

In Sect. 3, we describe the proposed demosaicing algorithm using the ADAIDs. Performance of the conventional and proposed demosaicing algorithms is compared using the 24 natural Kodak images.

### 3 Proposed Demosaicing Algorithm

In the proposed demosaicing algorithm, the horizontal and vertical ADAIDs are computed directly from the CFA image to determine the direction of interpolation. The artifacts in the demosaiced image, which usually appear in high-frequency regions, are caused primarily by aliasing in the  $R/B$  channels, because the decimated  $R/B$  channels have half the number of pixels compared with the decimated  $G$  channel. Fortunately, high correlation among color signals ( $R$ ,  $G$ , and  $B$ ), i. e., inter-channel correlation, exists in high-frequency regions of color images.

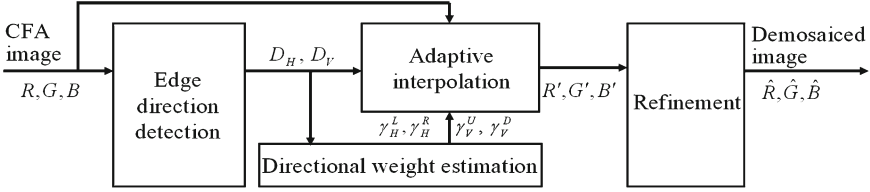


Fig. 2 Block diagram of the proposed adaptive demosaicing algorithm

### 3.1 Proposed Edge Direction Detection

The high-frequency components of the  $R$ ,  $G$ , and  $B$  channels are large at edge and texture regions. It is assumed that the positions of the edges are the same in the  $R$ ,  $G$ , and  $B$  channels. We use the absolute inter-channel difference, which is directly computed from the CFA pattern, to detect the edge direction and directional interpolation weights. The color components along the center row in Fig. 1a alternate between  $G$  and  $R$ . A similar argument can be applied to Fig. 1b and c. At each pixel, direction of interpolation is computed with neighboring pixel values along the horizontal and vertical directions, and the adaptive directional weights are estimated using the spatial correlation among the neighboring pixels along the detected direction of interpolation.

Figure 2 shows the block diagram of the proposed algorithm. The first block computes ADAIDs ( $D_H$  and  $D_V$ ) for edge direction detection, in which the absolute inter-channel differences ( $D_H^L$ ,  $D_H^R$ ,  $D_V^U$ , and  $D_V^D$ ) are used. For example, with the CFA pattern of Fig. 1a, the ADAIDs are defined along the horizontal and vertical directions, respectively, as

$$D_H = \left| D_H^L - D_H^R \right| = \left| |R_{i,j} - G_{i,j-1}| - |R_{i,j} - G_{i,j+1}| \right| \quad (4)$$

$$D_V = \left| D_V^U - D_V^D \right| = \left| |R_{i,j} - G_{i-1,j}| - |R_{i,j} - G_{i+1,j}| \right| \quad (5)$$

where  $R_{p,q}$  and  $G_{p,q}$  represent the color intensity values at  $(p, q)$  in  $R$  and  $G$  channels, respectively, and  $D_H^L$ ,  $D_H^R$ ,  $D_V^U$ , and  $D_V^D$  denote absolute inter-channel differences defined in the left, right, up, and down sides, respectively. For reliable edge direction detection, the quantity is defined:

$$\tau = \left| \frac{D_H}{\bar{m}_H} - \frac{D_V}{\bar{m}_V} \right| \quad (6)$$

where  $\bar{m}_H$  and  $\bar{m}_V$  represent local means of the horizontal ( $1 \times 3$ ) and vertical ( $3 \times 1$ ) masks, respectively.  $\tau$  is similar to the local contrast, which is related with the shape of contrast response function of the human eye [23]. If local mean is high, a small difference is negligible. Before classifying the edge direction at each pixel, we first

use  $\tau$  to separate flat regions, where dominant edge direction is difficult to define. Along with the difference of absolute differences  $D_H$  and  $D_V$ , the absolute difference of relative intensities to local means  $\tau$  is used. The fixed threshold  $th$  ( $=0.1$ ) is used for separation of the flat region, which gives robustness to local average intensity change of an image.

The proposed edge detection method uses two types of 1-D filters according to the pixel values as follows (for horizontal direction):

Case 1.

$$\begin{aligned} [10 - 1] : D_H &= G_{i,j+1} - G_{i,j-1}, \text{ if } R_{i,j} \geq G_{i,j-1} \geq G_{i,j+1} \\ [-101] : D_H &= G_{i,j-1} - G_{i,j+1}, \text{ if } R_{i,j} \geq G_{i,j+1} \geq G_{i,j-1} \end{aligned}$$

Case 2.

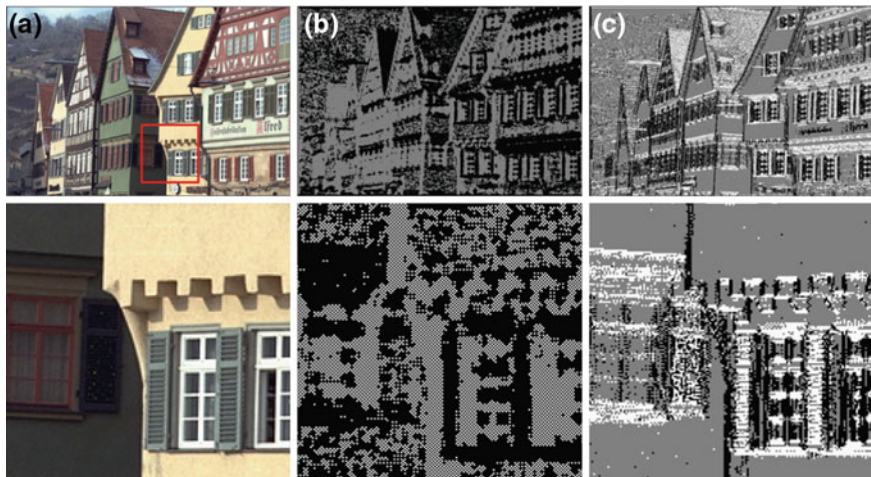
$$\begin{aligned} [10 - 1] : D_H &= G_{i,j+1} - G_{i,j-1}, \text{ if } G_{i,j-1} \geq G_{i,j+1} \geq R_{i,j} \\ [10 - 1] : D_H &= G_{i,j+1} - G_{i,j-1}, \text{ if } G_{i,j+1} \geq G_{i,j-1} \geq R_{i,j} \end{aligned}$$

Case 3.

$$\begin{aligned} [-12 - 1] : D_H &= 2R_{i,j} - G_{i,j+1} - G_{i,j-1}, \text{ if } G_{i,j+1} \geq R_{i,j} \geq G_{i,j-1} \\ [1 - 21] : D_H &= G_{i,j-1} + G_{i,j+1} - 2R_{i,j}, \text{ if } G_{i,j-1} \geq R_{i,j} \geq G_{i,j+1} \end{aligned}$$

where  $G_{i,j-1}$ ,  $R_{i,j}$ , and  $G_{i,j+1}$  are pixel values in the dashed line box of Fig. 1a and both cases can be represented by 1-D filters along the horizontal direction. If the center pixel value  $R_{i,j}$  is much different from both pixel values  $G_{i,j-1}$  and  $G_{i,j+1}$ , operation in Eq. (4) is equivalent to intra-channel difference (Cases 1 and 2). If the center pixel value  $R_{i,j}$  is between the values  $G_{i,j-1}$  and  $G_{i,j+1}$  in difference color channel, operation in Eq. (4) corresponds to inter-channel difference (Case 3) and the 1-D filter is similar to a Laplacian filter. That is, if the center pixel value  $R_{i,j}$  is similar to both pixel values  $G_{i,j-1}$  and  $G_{i,j+1}$ , our method detects the edge direction using adjacent pixels. Therefore, our method gives better performance for edge direction detection in the local region, which contains small structure, line and color edges.

Figure 3 shows the result of edge direction detection, where black, white, and gray represent the vertical edge, horizontal edge, and flat regions, respectively. The bottom row shows enlarged cropped images of the images in the top row. Figure 3a shows the original image of Kodak image no. 8, which contains a number of buildings with sharp edges of horizontal and vertical directions. Figure 3b shows the edge map of the conventional algorithm [15], which has two edge directions (vertical: black, horizontal: white). Figure 3c shows the edge map of the proposed algorithm (vertical: black, horizontal: white, flat: gray), which gives sharp edge lines and represents the



**Fig. 3** Edge direction detection (Kodak image no. 8). **a** original image, **b** edge map of the conventional algorithm [15] (*black vertical, white horizontal*), **c** edge map of the proposed algorithm (*black vertical, white horizontal, gray flat*)

structure of wall and windows well. Note that in our method the flat region (gray) is well separated, where the direction of edges is not dominant. Therefore, the proposed demosaicing algorithm has a good performance in images with a lot of structure, especially in regions with line edges such as window frames. In the flat region, the proposed algorithm simply uses the bilinear interpolation and does not perform refinement step (Sect. 3.3 Refinement), thus reducing the total computation time and the memory size.

Figure 1b is similar to Fig. 1a, only with  $R_{p,q}$  and  $B_{p,q}$  interchanged. Thus, with the CFA pattern of Fig. 1b, ADAIDs are calculated using (4) and (5), with  $R_{p,q}$  replaced by  $B_{p,q}$ . Figure 1c shows the Bayer CFA pattern with a  $G$  center pixel. With the CFA pattern of Fig. 1c, ADAIDs ( $D_H, D_V$ ) are similarly computed using (4) and (5), with  $R_{p,q}$  replaced by  $G_{p,q}$  and  $G_{p,q}$  replaced by  $B_{p,q}$  (in (4)) or  $R_{p,q}$  (in (5)).

### 3.2 Adaptive Interpolation

After edge direction detection, we derive the adaptive interpolation weights and interpolate the unknown pixel value using the directional interpolation weights and color difference values. The adaptive interpolation weights are defined as the same as those in Kimmel's algorithm [6]. Kimmel's algorithm uses the edge adaptive weights that are calculated using vertical and horizontal gradient magnitudes. The edge adaptive demosaicing algorithm, Kimmel's algorithm, is similar to the proposed algorithm and the interpolated image is smooth.

**Table 1** Performance comparison of the proposed and conventional algorithms (PSNR, unit: dB, 24 Kodak images)

Image	Gunturk et al.'s Algorithm [8]			Li's algorithm [9]			Lu and Tan's algorithm [11]			Chung and Chan's algorithm [15]			Menon et al.'s algorithm [16]			Proposed algorithm					
	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B			
1	25.3	29.6	25.4	<b>37.4</b>	40.1	37.1	36.6	<b>40.7</b>	<b>37.8</b>	32.4	36.8	35.5	34.5	36.0	34.7	36.1	38.1	36.4	35.7	39.3	36.4
2	31.9	36.3	32.4	<b>38.5</b>	40.6	39.0	35.4	39.9	39.0	32.5	40.6	39.0	37.2	41.3	39.9	38.1	<b>43.1</b>	<b>41.3</b>	38.4	43.3	41.1
3	33.5	37.2	33.8	41.6	43.5	39.8	38.7	41.8	38.8	34.0	41.4	38.8	40.6	43.1	39.9	41.3	<b>44.7</b>	40.7	<b>41.8</b>	44.5	<b>41.3</b>
4	32.6	36.5	32.9	<b>37.6</b>	42.3	41.4	36.0	42.4	40.8	37.3	41.0	34.4	36.8	41.5	40.9	37.0	43.0	<b>41.9</b>	<b>37.6</b>	<b>43.5</b>	41.6
5	25.8	29.3	25.9	<b>37.8</b>	<b>39.6</b>	35.5	35.1	37.6	34.7	32.8	37.9	35.5	35.0	36.5	34.5	36.8	39.4	36.1	36.2	39.3	<b>36.3</b>
6	26.7	31.1	27.0	38.6	<b>41.4</b>	37.3	37.2	41.3	36.6	32.8	37.8	35.4	37.0	38.4	36.5	<b>39.0</b>	40.7	<b>37.9</b>	36.5	40.0	36.4
7	32.6	36.5	32.6	42.2	43.7	39.6	39.9	42.3	38.9	34.3	41.6	38.8	40.4	42.2	39.7	41.2	44.0	40.6	<b>42.3</b>	<b>45.3</b>	<b>41.6</b>
8	22.5	27.4	22.5	<b>35.5</b>	38.2	34.3	34.3	<b>38.3</b>	<b>34.9</b>	31.4	35.7	33.4	32.4	34.5	32.6	34.5	37.1	34.6	32.1	36.3	32.4
9	31.6	35.7	31.4	<b>41.4</b>	44.0	40.7	40.2	42.7	40.6	39.8	41.6	34.3	40.0	42.3	40.9	40.9	<b>44.1</b>	<b>42.7</b>	40.7	43.9	41.1
10	31.8	35.4	31.2	<b>41.4</b>	44.4	40.6	39.9	43.5	40.4	39.6	41.8	34.3	39.5	42.0	40.3	40.7	44.4	<b>41.8</b>	40.9	<b>44.6</b>	41.4
11	28.2	32.2	28.3	<b>38.8</b>	41.1	38.4	37.3	<b>41.3</b>	38.4	33.3	38.7	37.3	36.5	38.4	37.1	38.0	40.6	<b>38.7</b>	37.4	40.9	38.2
12	32.7	36.8	32.4	<b>42.5</b>	44.8	41.4	40.8	44.3	41.1	34.4	41.7	39.3	40.7	43.2	41.3	41.7	<b>45.2</b>	<b>42.4</b>	41.3	<b>45.2</b>	41.6
13	23.1	26.5	23.0	34.5	36.9	33.0	<b>35.2</b>	<b>38.2</b>	<b>33.6</b>	30.8	33.2	31.6	31.3	32.2	30.8	33.1	34.2	32.2	32.2	35.1	32.1
14	28.2	32.0	28.6	<b>35.8</b>	37.8	34.0	32.5	35.9	33.4	32.2	38.2	<b>35.8</b>	34.7	37.4	34.4	35.3	39.1	35.6	35.4	<b>39.2</b>	35.4
15	31.1	35.4	32.3	<b>38.1</b>	40.3	38.8	35.2	40.3	38.7	32.4	39.7	37.8	36.7	40.6	39.4	37.0	42.0	<b>40.0</b>	37.2	<b>42.6</b>	39.7
16	30.3	34.7	30.4	41.9	<b>44.8</b>	40.6	38.7	43.4	39.4	33.5	39.7	37.7	40.9	42.2	40.3	<b>42.6</b>	44.3	<b>41.7</b>	39.7	43.3	39.7
17	31.6	34.5	30.9	<b>41.5</b>	<b>43.7</b>	<b>40.1</b>	40.5	42.8	39.7	40.4	41.1	39.7	38.8	40.1	38.7	40.1	41.9	39.9	39.9	42.6	39.8
18	27.3	30.5	26.8	<b>36.7</b>	<b>39.5</b>	35.9	35.4	38.4	36.4	34.7	36.4	35.1	34.4	36.1	34.7	35.8	38.2	<b>36.4</b>	35.7	39.0	36.3
19	26.8	31.7	27.0	<b>39.6</b>	<b>42.6</b>	39.0	38.3	42.0	39.3	37.9	40.0	38.6	37.3	39.4	38.3	38.8	41.3	<b>40.0</b>	38.3	42.1	39.3
20	30.8	34.6	30.6	<b>41.6</b>	<b>43.6</b>	38.2	38.6	41.5	37.0	33.7	39.2	31.3	38.8	40.2	37.5	40.5	42.4	<b>38.5</b>	40.3	43.3	38.9
21	27.6	31.5	27.5	<b>39.1</b>	<b>41.8</b>	<b>37.3</b>	37.9	41.6	37.0	33.8	38.1	36.0	36.0	37.4	35.3	37.5	39.4	36.6	37.3	40.5	36.9
22	29.8	33.3	29.2	<b>37.9</b>	39.9	36.3	36.7	39.2	36.5	34.0	38.9	36.8	35.2	37.9	35.7	36.6	39.5	36.7	36.6	<b>40.5</b>	<b>37.2</b>
23	34.4	38.0	34.1	<b>42.3</b>	43.7	40.0	38.0	41.6	40.5	33.3	41.7	39.7	40.5	43.6	41.6	41.2	44.8	41.9	41.5	<b>45.1</b>	<b>42.0</b>
24	26.4	<b>39.4</b>	25.3	<b>35.0</b>	37.4	32.6	34.6	37.5	32.9	33.8	35.4	31.9	32.8	34.8	32.3	34.4	36.8	<b>32.9</b>	34.1	37.5	<b>32.9</b>
Mean	29.3	33.6	29.2	<b>39.1</b>	<b>41.5</b>	38.0	37.2	40.8	37.8	34.4	39.1	36.2	37.0	39.2	37.4	38.3	41.2	<b>38.7</b>	37.9	<b>41.5</b>	38.3

In Fig. 1a where an  $R$  pixel is at the center of the  $5 \times 5$  Bayer CFA pattern, the unknown (missing)  $G$  pixel value  $G'_{i,j}$  is interpolated as

$$G'_{i,j} = \begin{cases} R_{i,j} + \frac{\gamma_H^L \Delta G_H^L + \gamma_H^R \Delta G_H^R}{\gamma_H^L + \gamma_H^R}, & \tau \geq th \& D_H < D_V \\ R_{i,j} + \frac{\gamma_V^U \Delta G_V^U + \gamma_V^D \Delta G_V^D}{\gamma_V^U + \gamma_V^D}, & \tau \geq th \& D_H > D_V \\ R_{i,j} + \frac{\gamma_H^L \Delta G_H^L + \gamma_H^R \Delta G_H^R + \gamma_V^U \Delta G_V^U + \gamma_V^D \Delta G_V^D}{\gamma_H^L + \gamma_H^R + \gamma_V^U + \gamma_V^D}, & \text{otherwise} \end{cases} \quad (7)$$

where  $\Delta G_H^L$ ,  $\Delta G_H^R$ ,  $\Delta G_V^U$ , and  $\Delta G_V^D$  are the color difference values which are defined in the left, right, up, and down sides, respectively, as

$$\Delta G_H^L = G_{i,j-1} - \frac{R_{i,j-2} + R_{i,j}}{2} \quad (8)$$

$$\Delta G_H^R = G_{i,j+1} - \frac{R_{i,j+2} + R_{i,j}}{2} \quad (9)$$

$$\Delta G_V^U = G_{i-1,j} - \frac{R_{i-2,j} + R_{i,j}}{2} \quad (10)$$

$$\Delta G_V^D = G_{i+1,j} - \frac{R_{i+2,j} + R_{i,j}}{2} \quad (11)$$

and  $\gamma_H^L$ ,  $\gamma_H^R$ ,  $\gamma_V^U$ , and  $\gamma_V^D$  represent the directional interpolation weights defined in the left, right, up, and down sides, respectively, as

$$\gamma_H^L = \frac{1}{1 + \left| \frac{G_{i,j-1} - G_{i,j+1}}{2} \right| + \left| \frac{R_{i,j-2} - R_{i,j}}{2} \right|} \quad (12)$$

$$\gamma_H^R = \frac{1}{1 + \left| \frac{G_{i,j-1} - G_{i,j+1}}{2} \right| + \left| \frac{R_{i,j+2} - R_{i,j}}{2} \right|} \quad (13)$$

$$\gamma_V^U = \frac{1}{1 + \left| \frac{G_{i-1,j} - G_{i+1,j}}{2} \right| + \left| \frac{R_{i-2,j} - R_{i,j}}{2} \right|} \quad (14)$$

$$\gamma_V^D = \frac{1}{1 + \left| \frac{G_{i-1,j} - G_{i+1,j}}{2} \right| + \left| \frac{R_{i+2,j} - R_{i,j}}{2} \right|}. \quad (15)$$

The interpolation weights are used to determine the directional weights to interpolate the unknown pixel values in the adaptive interpolation block in Fig. 2. The proposed



algorithm interpolates the  $G$  components with the directional interpolation weights (12)–(15).

Figure 1b is similar to Fig 1a, only with  $R_{p,q}$  and  $B_{p,q}$  interchanged. Thus, in Fig. 1b where a  $B$  pixel is at the center of the  $5 \times 5$  Bayer CFA pattern, unknown  $G$  pixel value  $G'_{i,j}$  is similarly estimated using (6)–(15), with  $R_{p,q}$  replaced by  $B_{p,q}$ .

After  $G$  interpolation, R and B interpolation follows. For example, in Fig. 1b where B pixel is at the center of the  $5 \times 5$  Bayer CFA pattern, the unknown (missing)  $R$  pixel value  $R'_{i,j}$  is interpolated as

$$R'_{i,j} = G'_{i,j} + \frac{\gamma_D^L \Delta R_D^L + \gamma_U^R \Delta R_U^R + \gamma_U^L \Delta R_U^L + \gamma_D^R \Delta R_D^R}{\gamma_D^L + \gamma_U^R + \gamma_U^L + \gamma_D^R} \quad (16)$$

where  $\Delta R_D^L$ ,  $\Delta R_U^R$ ,  $\Delta R_U^L$ , and  $\Delta R_D^R$  are the color difference values defined in the left-down, right-up, left-up, and right-down sides, respectively. They are expressed as

$$\Delta R_D^L = R_{i+1,j-1} - G'_{i+1,j-1} \quad (17)$$

$$\Delta R_U^R = R_{i-1,j+1} - G'_{i-1,j+1} \quad (18)$$

$$\Delta R_U^L = R_{i-1,j-1} - G'_{i-1,j-1} \quad (19)$$

$$\Delta R_D^R = R_{i+1,j+1} - G'_{i+1,j+1} \quad (20)$$

and  $\gamma_D^L$ ,  $\gamma_U^R$ ,  $\gamma_U^L$ , and  $\gamma_D^R$  represent the directional interpolation weights similarly defined in the left-down, right-up, left-up, and right-down sides, respectively, as

$$\gamma_D^L = \frac{1}{1 + \left| \frac{R_{i+1,j-1} - R_{i-1,j+1}}{2} \right| + \left| \frac{B_{i+2,j-2} - B_{i,j}}{2} \right|} \quad (21)$$

$$\gamma_U^R = \frac{1}{1 + \left| \frac{R_{i+1,j-1} - R_{i-1,j+1}}{2} \right| + \left| \frac{B_{i-2,j+2} - B_{i,j}}{2} \right|} \quad (22)$$

$$\gamma_U^L = \frac{1}{1 + \left| \frac{R_{i-1,j-1} - R_{i+1,j+1}}{2} \right| + \left| \frac{B_{i-2,j-2} - B_{i,j}}{2} \right|} \quad (23)$$

$$\gamma_D^R = \frac{1}{1 + \left| \frac{R_{i-1,j-1} - R_{i+1,j+1}}{2} \right| + \left| \frac{B_{i+2,j+2} - B_{i,j}}{2} \right|} \quad (24)$$

The proposed algorithm interpolates the  $R$  components with the directional interpolation weights (21)–(24) in Fig. 1b.

Figure 1a is similar to Fig. 1b, only with  $B_{p,q}$  and  $R_{p,q}$  interchanged. Thus, in Fig. 1a where  $R$  pixel is at the center of the  $5 \times 5$  Bayer CFA pattern, unknown  $B$  pixel value  $B'_{i,j}$  is estimated using (16)–(24), with  $B_{p,q}$  and  $R_{p,q}$  interchanged.

In Fig. 1c, for  $R/B$  components interpolation, the directional weights  $\gamma_H^L$ ,  $\gamma_H^R$ ,  $\gamma_V^U$ , and  $\gamma_V^D$  are estimated using (12)–(15), with  $R_{p,q}$  replaced by  $G_{p,q}$  and  $G_{p,q}$  replaced by  $B_{p,q}$  (in (12) and (13)) or  $R_{p,q}$  (in (14) and (15)). The unknown (missing)  $R$  pixel value  $R'_{i,j}$  is interpolated using (16), the color difference values, and the directional weights. The color difference values are defined as

$$\Delta R_H^L = R'_{i,j-1} - G'_{i,j-1} \quad (25)$$

$$\Delta R_H^R = R'_{i,j+1} - G'_{i,j+1} \quad (26)$$

$$\Delta R_V^U = R_{i-1,j} - G'_{i-1,j} \quad (27)$$

$$\Delta R_V^D = R_{i+1,j} - G'_{i+1,j}. \quad (28)$$

The unknown (missing)  $B$  pixel value  $B'_{p,q}$  is interpolated using (16), with red color difference values  $\Delta R_H^L$ ,  $\Delta R_H^R$ ,  $\Delta R_V^U$ , and  $\Delta R_V^D$  replaced by blue color difference values  $\Delta B_H^L$ ,  $\Delta B_H^R$ ,  $\Delta B_V^U$ , and  $\Delta B_V^D$ , respectively, and (25)–(28), with  $R'_{p,q}$  replaced by  $B_{p,q}$  (in (25) and (26)) and  $R_{p,q}$  replaced by  $B'_{p,q}$  (in (27) and (28)).

### 3.3 Directional Refinement

In the proposed demosaicing algorithm, a modified version of Menon et al.'s algorithm [16] is used, in which flat region and edge region (vertical and horizontal) are classified. Note that the flat region is not refined. The demosaiced image contains the noticeable artifacts, such as zipper artifact, blurring, and false color. These artifacts appear mainly in the high-frequency region. Therefore, the demosaiced image needs to be refined using the high-frequency components with inter-channel correlation of the  $R/G/B$  channels. For example,  $G$  components are updated by adding the high-frequency components in  $R$  ( $B$ ) channel, which is the known (measured) pixel value in the CFA pattern. For example, in Fig. 1a where an  $R$  pixel is at the center of the  $5 \times 5$  Bayer CFA pattern,  $G/B$  components are refined. The refined  $G$  component  $\hat{G}$  is expressed as

$$\hat{G}_{i,j} = G'_{i,j} + R_{i,j}^h \quad (29)$$

$$R_{i,j}^h = R_{i,j} - R'_{i,j} \quad (30)$$

where  $G'_{i,j}$  represents the interpolated  $G$  component at  $(i,j)$  at the adaptive interpolation step,  $R_{i,j}^h$  is the high-frequency component of the  $R$  channel, and  $R'_{i,j}$  denotes the

low-frequency component that is filtered by a 3-tap 1-D filter  $[1/3, 1/3, 1/3]$  along the detected horizontal or vertical edge direction. Similarly,  $B$  components are refined using (29) and (30), with  $\hat{G}_{i,j}$  replaced by  $\hat{B}_{i,j}$  and  $G'_{i,j}$  replaced by  $B'_{i,j}$ .

In Fig. 1b and c where  $B$  and  $G$  pixels are at the center of the  $5 \times 5$  Bayer CFA pattern, respectively, the  $R/G$  and  $R/B$  components are refined in a similar way. After this refinement step, the demosaiced image with  $\hat{R}$ ,  $\hat{G}$ , and  $\hat{B}$  is finally obtained, as shown in the last block of Fig. 2. The average PSNR improvements with the “refinement process” are 1.69, 1.17, and 1.89 dB for  $R$ ,  $G$ , and  $B$  channels, respectively.

## 4 Experimental Results and Discussions

Figure 4 shows the comparison of the demosaiced images of the six conventional algorithms: bilinear interpolation [20], Gunturk et al.’s algorithm [8], Li’s algorithm [9], Lu and Tan’s algorithm [11], Chung and Chan’s algorithm [15], and Menon et al.’s algorithm [16]. Figure 4a shows the  $30 \times 70$  Kodak image in which the sharp color edge is shown along the vertical line. Figure 4b shows the demosaiced image by bilinear interpolation, in which the color edge is blurred. Figure 4c and d show the demosaiced images by Gunturk et al.’s and Li’s algorithms, respectively, in which zipper effects are shown along the vertical color edge line. Figure 4e, f, g and h show the demosaiced images by Lu and Tan’s, Chung and Chan’s, Menon et al.’s, and proposed algorithms, respectively. The subjective visual quality of the demosaiced images by the Chung and Chan’s, Menon et al.’s, and proposed algorithms are satisfactory.

Figure 4i, j and k show the 1-D intensity profiles of  $R$ ,  $B$ , and  $G$  channels along the 50<sup>th</sup> horizontal line in the Kodak image, respectively. Color edge is sharp boundary between two distinct colors ( $R$  and  $B$ , with zero  $G$ ). To determine the direction of interpolation, the proposed algorithm computes the horizontal and vertical ADAIDs directly from the CFA image. Three color components ( $R/G/B$ ) are interpolated along the detected horizontal and vertical directions considering the ADAIDs. The sharp color edge in the result image of the proposed algorithm is preserved well without blurring and zipper artifacts.

Figure 5 shows the 24 Kodak images used in experiments. These Kodak images have the resolution of  $768 \times 512$  and in experiments the input images are sampled according to the Bayer pattern [1]. After demosaicing by conventional and proposed algorithms, we compare the demosaiced images with the original ones, in terms of the PSNR, SSIM, and subjective visual quality. The SSIM [21, 22] can be computed using local structural similarity measurement, which is derived from the luminance (mean), contrast (variance), and difference of the cross-correlation of the image patches to be compared.

Table 1 shows the PSNR comparison of the proposed and six conventional algorithms. We use bold type to highlight the largest PSNR value among seven demosaicing algorithms to be compared for each of  $R/G/B$  channels. The PSNRs of the

**Table 2** Performance comparison of the proposed and conventional algorithms (SSIM, 24 Kodak images)

Image	Bilinear [20]	Gunturk et al.'s Algorithm [8]	Li's algorithm [9]	Lu and Tan's algorithm [11]	Chung and Chan's algorithm [15]	Menon et al.'s algorithm [16]	Proposed algorithm
1	0.9022	0.9917	<b>0.9928</b>	0.9873	0.9867	0.9894	0.9916
2	0.9478	0.9817	0.9847	<b>0.9869</b>	0.9856	0.9866	0.9835
3	0.9661	0.9926	0.9921	0.9925	0.9930	<b>0.9932</b>	0.9921
4	0.9554	0.9883	0.9889	0.9894	0.9879	0.9890	<b>0.9895</b>
5	0.9359	0.9931	0.9903	<b>0.9934</b>	0.9932	0.9932	0.9914
6	0.9172	0.9924	0.9927	0.9890	0.9923	<b>0.9928</b>	0.9916
7	0.9756	0.9935	0.9930	0.9946	<b>0.9956</b>	0.9936	0.9926
8	0.9108	0.9919	<b>0.9922</b>	0.9883	0.9879	0.9903	0.9911
9	0.9621	<b>0.9914</b>	0.9911	0.9904	0.9904	0.9904	0.9906
10	0.9620	0.9923	0.9922	0.9917	0.9913	0.9913	<b>0.9924</b>
11	0.9326	0.9914	0.9916	0.9898	0.9908	<b>0.9919</b>	<b>0.9919</b>
12	0.9543	0.9922	0.9923	0.9912	<b>0.9925</b>	0.9923	0.9922
13	0.8771	0.9906	<b>0.9919</b>	0.9822	0.9830	0.9853	0.9893
14	0.9331	0.9893	0.9853	0.9898	0.9888	<b>0.9902</b>	0.9864
15	0.9577	0.9853	0.9870	0.9866	<b>0.9873</b>	0.9870	0.9862
16	0.9347	0.9933	0.9932	0.9899	0.9930	<b>0.9935</b>	0.9930
17	0.9632	0.9935	0.9930	0.9917	0.9915	0.9917	<b>0.9936</b>
18	0.9340	<b>0.9899</b>	0.9878	0.9876	0.9861	0.9870	0.9884
19	0.9362	0.9915	0.9909	0.9885	0.9889	0.9890	<b>0.9921</b>
20	0.9630	<b>0.9911</b>	0.9908	0.9910	<b>0.9911</b>	0.9894	0.9906
21	0.9429	<b>0.9914</b>	0.9911	0.9892	0.9900	0.9890	0.9904
22	0.9413	<b>0.9875</b>	0.9867	0.9874	0.9849	0.9855	0.9866
23	0.9783	0.9906	0.9898	0.9921	<b>0.9931</b>	0.9907	0.9899
24	0.9368	0.9918	0.9914	0.9905	0.9904	0.9905	<b>0.9920</b>
Mean	0.9425	<b>0.9908</b>	0.9905	0.9896	0.9898	0.9901	0.9903

proposed algorithm does not give the best performance for entire images that contain different types of regions with different characteristics: flat, edge, and detail regions. The PSNR does not reflect well the human visual perception. The SSIM has been used as a better image or video quality measure because it closely reflects the human visual perception. Thus, for performance comparison of different demosaicing algorithms, we use the SSIM as well as the PSNR.

Table 2 shows the performance comparison of the proposed and six conventional algorithms in terms of the SSIM, where the SSIM is in the range of 0 and 1 (the larger, the better). The SSIMs of the proposed algorithm for the images that contain color sharp edges and natural background (Kodak image nos. 4, 10, 11, 17, 19, and 24) are higher than those of the conventional algorithms. Similarly, bold type is used to highlight the largest SSIM value among seven demosaicing algorithms to be compared for the luminance.

Figure 6a shows the enlarged region A of the original image ( $768 \times 512$ , Kodak image no. 3), which contains a red cap with sharp color edges. Figure 6b, which is interpolated by bilinear interpolation, is blurred and shows color artifacts around color edges. In Fig. 6c, d, f and g, which are obtained by Gunturk et al.'s, Li's, Chung and Chan's, and Menon et al.'s algorithms, respectively, zipper effects are shown around the region with color edges. They are capable of producing better results in terms of the PSNR and SSIM with shaper reconstructed edges and less demosaicing artifacts in other regions. However their results around sharp color edges show zipper effects, which look like residual patterns of Bayer pattern. Figure 6e and h, which are reconstructed by Lu and Tan's and the proposed algorithms, respectively, can reconstruct sharp color edges with little zipper effect. Figure 6i and j show the edge maps of the proposed algorithm, of the entire image and cropped image, respectively, which show detected edge direction using ADAID.

In Table 1, the PSNR of the Gunturk et al.'s algorithm is the highest, but Fig. 6c shows zipper effects around the color edge on the cap. The PSNRs of the Kodak image no. 3 of three algorithms (Lu and Tan's, Menon et al.'s, and proposed algorithms) are lower than that of Gunturk et al.'s algorithm. In Table 2, the rank of the proposed algorithm is the third. The proposed algorithm does not give the best performance for entire images that contain different types of regions.

Figures 7 and 8 show the performance comparison of the conventional and proposed algorithms. Figures 7a and 8a show the enlarged regions B and C in the original image ( $768 \times 512$ , Kodak image nos. 21 and 22), which contain small structure with high frequency components such as roof, windows, and fence. Figures 7b and 8b obtained by the bilinear algorithm are blurred and have color artifacts and zipper effects in edges of roof, windows, and fence.

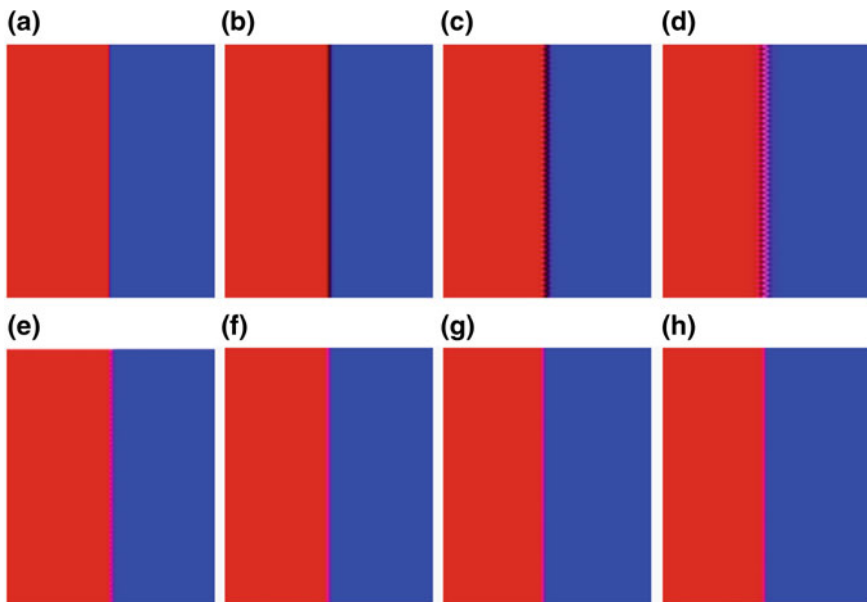
Figure 7c and d, which are obtained by Gunturk et al.'s algorithm and Li's algorithm, respectively, show color artifacts in thin line segments as in a fence. Figures 7e and 8e show the demosaiced images by Lu and Tan's algorithm, which is capable of reconstructing line segments with little zipper effects and color artifacts, however this algorithm requires long computation time. Figure 7f and g show the demosaiced images by Chung and Chan's, and Menon et al.'s algorithms, respectively, with a little color artifact in the fence. Figure 7h shows the demosaiced images by the proposed

**Table 3** Performance comparison of the proposed and conventional algorithms (PSNR, unit: dB, cropped images (Kodak image nos. 3, 21, and 22))

Image	Bilinear [20]			Gunturk et al.'s Algorithm [8]			Li's algorithm [9]			Lu and Tan's algorithm [11]			Chung and Chan's algorithm [15]			Menon et al.'s algorithm [16]			Proposed algorithm		
	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
Fig. 6a	29.2	32.8	32.0	31.5	33.5	34.0	30.3	34.6	36.3	30.8	39.2	38.3	31.6	36.2	37.2	31.8	36.9	36.3	32.6	39.1	39.0
Fig. 7a	22.7	26.7	23.8	34.7	38.8	34.0	35.1	38.2	33.6	35.0	38.4	34.1	35.0	37.1	34.4	36.0	38.5	34.8	35.8	38.6	34.3
Fig. 8a	25.7	27.7	23.5	29.4	32.5	28.2	29.6	32.7	28.7	31.7	35.4	30.4	31.7	33.8	29.2	29.9	34.3	29.1	32.1	34.6	30.0
Mean	25.9	29.1	26.4	31.9	35.0	32.0	31.7	35.2	32.9	32.5	37.7	34.3	32.8	35.7	33.6	32.6	36.6	33.4	33.5	37.4	34.4

**Table 4** Performance comparison of the proposed and conventional algorithms (SSIM, unit: dB, Y component of the cropped images (Kodak image nos. 3, 21, and 22))

Image	Bilinear [20]	Gunturk et al.'s Algorithm [8]	Li's algorithm [9]	Lu and Tan's algorithm [11]	Chung and Chan's algorithm [15]	Menon et al.'s algorithm [16]	Proposed algorithm
Fig. 6a	0.9714	0.9806	0.9812	0.9896	0.9901	<b>0.9913</b>	0.9900
Fig. 7a	0.9926	0.9932	0.9926	<b>0.9936</b>	0.9928	0.9913	0.9931
Fig. 8a	0.9451	0.9696	0.9705	<b>0.9792</b>	0.9762	0.9719	0.9785
Mean	0.9697	0.9811	0.9814	<b>0.9875</b>	0.9864	0.9848	0.9872



**Fig. 4** Kodak image ( $30 \times 70$ , sharp color edge). **a** original image, **b** demosaiced image by bilinear interpolation [20], **c** demosaiced image by Gunturk et al.'s algorithm [8], **d** demosaiced image by Li's algorithm [9], **e** demosaiced image by Lu and Tan's algorithm [11], **f** demosaiced image by Chung and Chan's algorithm [15], **g** demosaiced image by Menon et al.'s algorithm [16], **h** demosaiced image by the proposed algorithm, **i** comparison of the demosaiced images (1-D intensity profile along the 50th row, R channel), **j** comparison of the demosaiced images (1-D intensity profile along the 50th row, B channel), **k** comparison of the demosaiced images (1-D intensity profile along the 50th row, G channel)

algorithm, which does not show zipper effects, color artifact, and blurring. Figure 7i and j show the edge maps of the proposed algorithm, of the entire image and cropped image, respectively, which show detected edge direction using ADAID.

In Fig. 8c and d, which are obtained by Gunturk et al.'s and Li's algorithms, respectively, their PSNRs are higher than those of the other algorithms, however zipper effects are shown around color edges of roof and windows. Figure 8f and g show demosaiced images, which are obtained by Chung and Chan's algorithm and Menon et al.'s algorithm, respectively, with a little zipper effect around roof and windows. Figure 8h shows the demosaiced images by the proposed algorithm, which does not show zipper effects. Figures 8i and j show the edge maps of the proposed algorithm, of the entire image and cropped image, respectively, which show detected edge direction using ADAID.

Tables 3 and 4 show the comparison of the PSNR and SSIM of the proposed and six conventional algorithms for enlarged regions A, B, and C in Kodak images (nos. 3, 21, and 22, respectively). We use bold type to highlight the largest PSNR value among seven demosaicing algorithms compared for each of  $R/G/B$  channels.



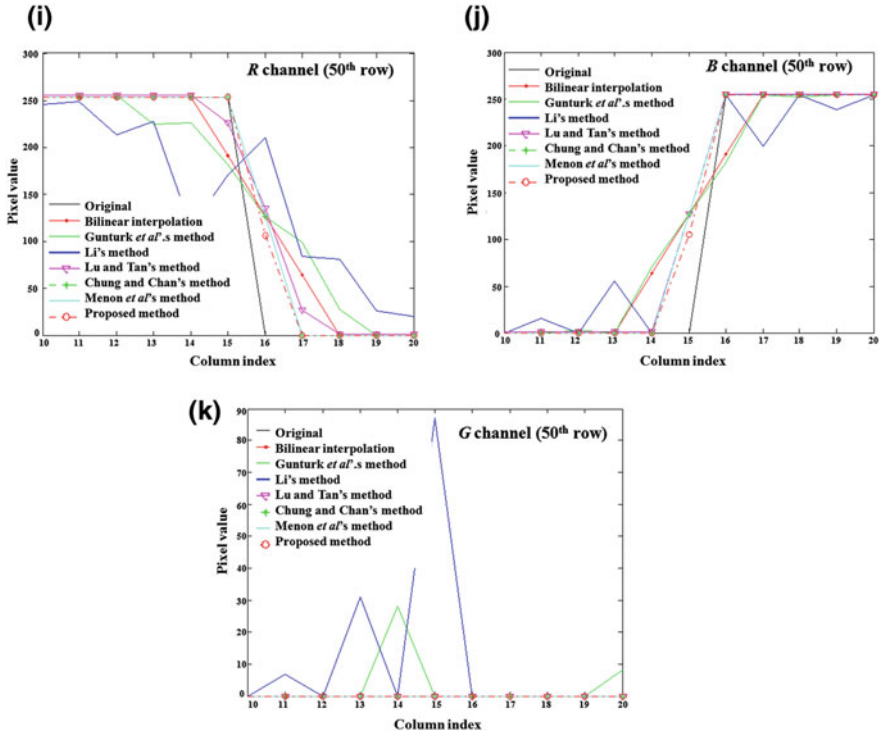
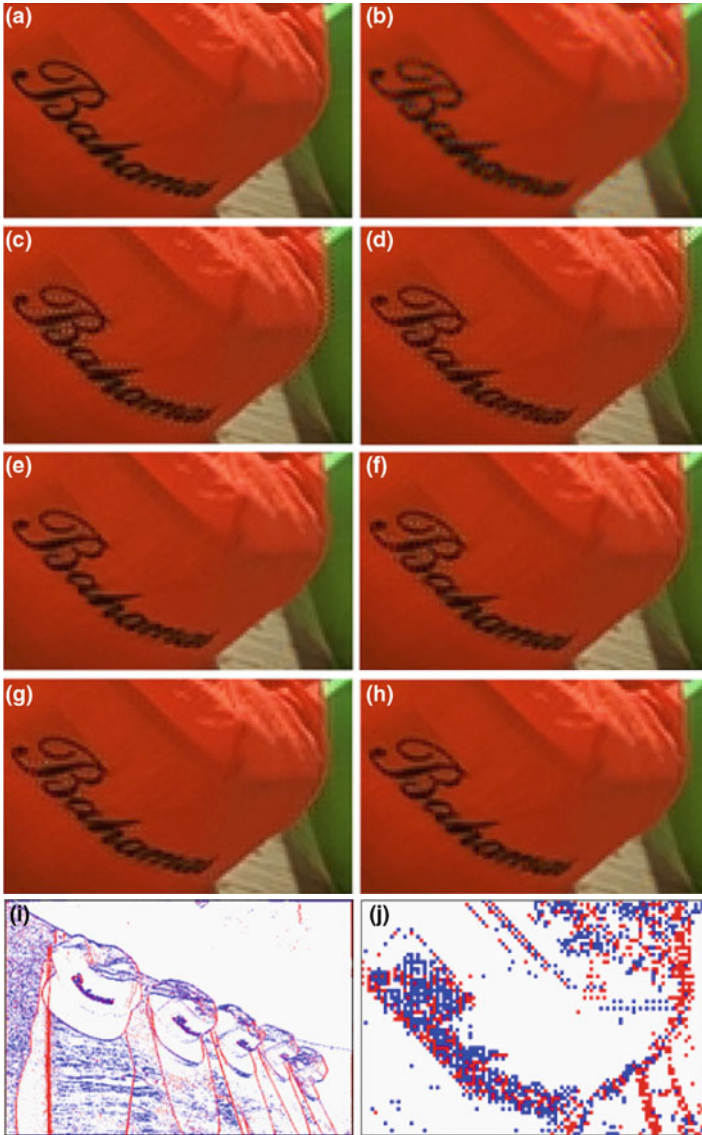


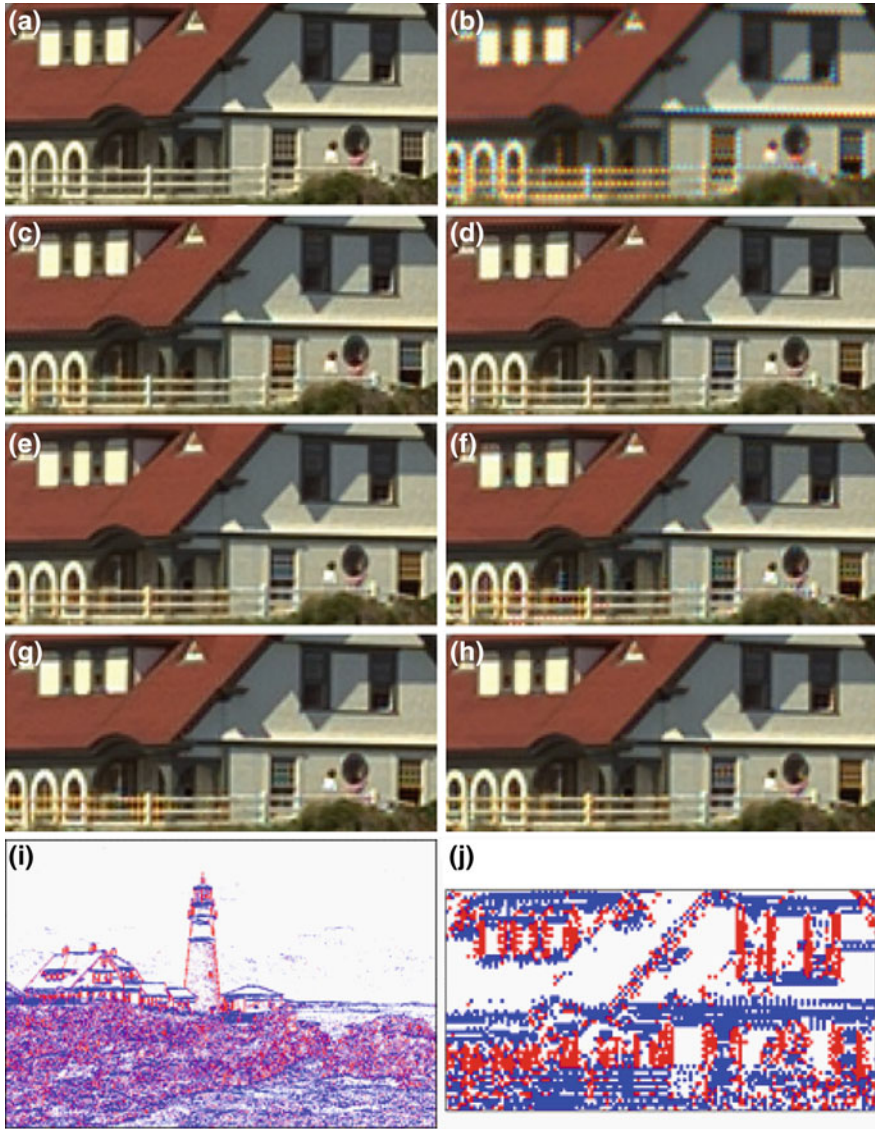
Fig. 4 Continued



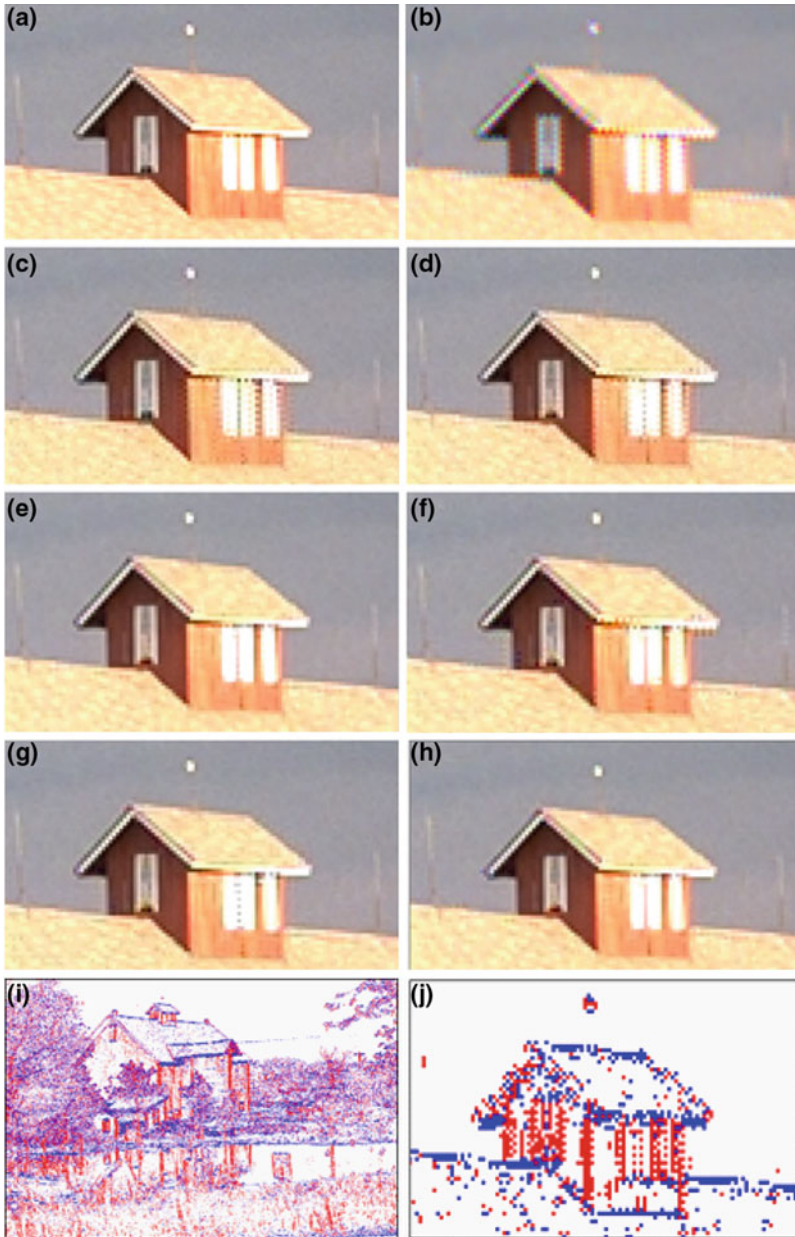
Fig. 5 Kodak images ( $768 \times 512$ ) used in experiments



**Fig. 6** Comparison of demosaicing algorithms (Kodak image no. 3, enlarged region A,  $110 \times 65$ ). **a** original image, **b** demosaiced image by bilinear interpolation [20], **c** demosaiced image by Gunturk et al.'s algorithm [8], **d** demosaiced image by Li's algorithm [9], **e** demosaiced image by Lu and Tan's algorithm [11], **f** demosaiced image by Chung and Chan's algorithm [15], **g** demosaiced image by Menon et al.'s algorithm [16], **h** demosaiced image by the proposed algorithm, **i** edge map of the proposed algorithm (Kodak image no. 3 in Fig. 5), **j** edge map of the proposed algorithm (Fig. 6i, enlarged region A)



**Fig. 7** Comparison of demosaicing algorithms (Kodak image no. 21, enlarged region B,  $140 \times 65$ ). **a** original image, **b** demosaiced image by bilinear interpolation [20], **c** demosaiced image by Gunturk et al.'s algorithm [8], **d** demosaiced image by Li's algorithm [9], **e** demosaiced image by Lu and Tan's algorithm [11], **f** demosaiced image by Chung and Chan's algorithm [15], **g** demosaiced image by Menon et al.'s algorithm [16], **h** demosaiced image by the proposed algorithm, **i** edge map of the proposed algorithm (Kodak image no. 21 in Fig. 5), **j** edge map of the proposed algorithm (Fig. 7i, enlarged region B)



**Fig. 8** Comparison of demosaicing algorithms (Kodak image no. 22, enlarged region C,  $120 \times 65$ ). **a** original image, **b** demosaiced image by bilinear interpolation [20], **c** demosaiced image by Gunturk et al.'s algorithm [8], **d** demosaiced image by Li's algorithm [9], **e** demosaiced image by Lu and Tan's algorithm [11], **f** demosaiced image by Chung and Chan's algorithm [15], **g** demosaiced image by Menon et al.'s algorithm [16], **h** demosaiced image by the proposed algorithm, **i** edge map of the proposed algorithm (Kodak image no. 22 in Fig. 5), **j** edge map of the proposed algorithm (Fig. 8i, enlarged region C)

The PSNR and SSIM of Lu and Tan's algorithm and the proposed algorithm do not give the best performances for the entire image containing different types of regions. However, they give higher PSNR and SSIM for the local region (Figs. 6a, 7a, and 8a), which contains small structure, line and color edges. In terms of the PSNR and SSIM, the rank of Lu and Tan's algorithm is the first highest and that of the proposed algorithm is the second highest. The abrupt changes of color values indicate low spatial correlation among neighboring pixels. Intuitively, the higher the spatial correlation among pixels along the interpolation direction, the more accurate the estimate of a missing color value can be obtained along that direction. Lu and Tan's algorithm and the proposed algorithm combine the estimates from four directions (right/left, up/down) by assigning them with weights that measure the spatial correlations among the neighboring pixels along the corresponding interpolation directions. For the entire image, Gunturk et al.'s and Li's algorithms, which reconstruct the missing pixel values iteratively, give the higher PSNR and SSIM than those of other algorithms. However, sometimes they give poor performance for the local region, which contains the small structure, line and color edges, due to excessive sharpening.

The conventional algorithms and the proposed algorithm have been implemented using Matlab code with MEX files. The edge directions and the directional weights computations of the proposed algorithm are also implemented using MEX files. In our simulations on a 2.8 GHz Pentium 4 PC with 2048 MB, the average computational time for the proposed algorithm is 1.05 s. The computational times of the bilinear interpolation (our implementation), Gunturk et al.'s [8] (publicly available implementation: <http://www.csee.wvu.edu/~xinl/>), Li's [9] (publicly available implementation: <http://www.csee.wvu.edu/~xinl/>), Lu and Tan's [11] (publicly available implementation: <http://www.csee.wvu.edu/~xinl/>), Chung and Chan's algorithm (our implementation), and Menon et al.'s [16] (publicly available implementation: <http://www.danielemenon.it>) algorithms are 0.21, 5.19, 1.62, 6.25, 5.28, and 2.54 s, respectively. The number of iterations of Gunturk et al.'s [8] and Li's [9] algorithms is set to 10.

In summary, the proposed algorithm outperforms conventional algorithms for Kodak images that contain small structure, line and color edges, in terms of the PSNR (Table 3) and SSIM (Table 4). Also it reduces well the zipper effect around color edges in terms of subjective visual quality, as shown in Figs. 6, 7 and 8.

## 5 Conclusions

This paper proposes an adaptive demosaicing algorithm using the characteristics of the CFA pattern. It is simple, because the edge direction and the directional weights are computed directly from the CFA image using the ADAIDs. Experimental results with synthetic and 24 Kodak images show the effectiveness of the proposed algorithm for zipper effects and color artifacts, in terms of the performance measures such as

the PSNR, SSIM, and subjective visual quality. Future research will focus on the false color suppression in the texture region.

**Acknowledgments** This work was supported in part by the Second Brain Korea 21 Project.

## References

1. Bayer BE (1976) Color imaging array. US Patent, 3,971,065, 12 May 1976
2. Hao P, Li Y, Lin Z, Dubois E (2011) A Geometric method for optimal design of color filter arrays. *IEEE Trans Image Process* 20(3):709–722
3. Lukac R (2006) Single-sensor imaging in consumer digital cameras: a survey of recent advances and future directions. *J Real-Time Image Process* 1(1):45–52
4. Adams JE (1998) Design of color filter array interpolation algorithms for digital cameras, Part 2. In: Proceedings of the IEEE international conference on image processing, Chicago, IL, pp 488–492 (1998)
5. Hamilton JF, Adams JE (1997) Adaptive color plane interpolation in single sensor color electronic camera. US Patent, 5,629,734, 13 May 1997
6. Kimmel R (1999) Demosaicing: image reconstruction from CCD samples. *IEEE Trans Image Process* 8(9):1221–1228
7. Lin T-N, Hsu C-L (2004) Directional weighting-based demosaicing algorithm. In: Proceedings of the Pacific Rim conference on multimedia, Tokyo, Japan, vol 3332, pp 849–857 (2004)
8. Gunturk BK, Altunbasak Y, Mersereau RM (2002) Color plane interpolation using alternating projections. *IEEE Trans Image Process* 11(9):997–1013
9. Li X (2005) Demosaicing by successive approximation. *IEEE Trans Image Process* 14(3):370–379
10. Hirakawa K, Park TW (2005) Adaptive homogeneity-directed demosaicing algorithm. *IEEE Trans Image Process* 14(3):360–369
11. Lu W, Tan Y (2003) Color filter array demosaicing: new method and performance measures. *IEEE Trans Image Process* 12(10):1194–1210
12. Lukac R, Plataniotis KN, Hatzinakos D, Aleksic M (2004) A novel cost effective demosaicing approach. *IEEE Trans Consum Electron* 50(1):256–261
13. Chang L, Tam YP (2004) Effective use of spatial and spectral correlations for color filter array demosaicing. *IEEE Trans Consum Electron* 50(1):355–365
14. Lukac R, Plataniotis KN (2005) Data-adaptive filters for demosaicing: a framework. *IEEE Trans Consum Electron* 50(2):560–570
15. Chung K-H, Chan Y-H (2006) Color demosaicing using variance of color differences. *IEEE Trans Image Process* 15(10):2944–2954
16. Menon D, Andriani S, Calvagno G (2007) Demosaicing with directional filtering and a posteriori decision. *IEEE Trans Image Process* 16(1):132–141
17. Tsai C-Y, Song K-T (2007) A new edge-adaptive demosaicing algorithm for color filter arrays. *Image Vis. Comput.* 25(9):1495–1508
18. Mairal J, Elad M, Sapiro G (2008) Sparse representation for color image restoration. *IEEE Trans Image Process* 17(1):53–69
19. Chung K-L, Yang W-J, Yan W-M, Wang C-C (2008) Demosaicing of color filter array captured images using gradient edge detection masks and adaptive heterogeneity-projection. *IEEE Trans Image Process* 17(12):2356–2367
20. Gonzalez RC, Woods RE (2010) Digital image processing, 3rd edn. Pearson Education, Inc., Upper Saddle River
21. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process* 13(4):600–612

22. Wang Z, Bovik AC (2002) A universal image quality index. *IEEE Signal Process Lett* 9(3):600–612
23. Frazor RA, Geisler WS (2006) Local luminance and contrast in natural images. *Vis. Res.* 48:1585–1598

# A Taxonomy of Color Constancy and Invariance Algorithm

Dohyoung Lee and Konstantinos N. Plataniotis

**Abstract** Color is an effective cue for identifying regions of interest or objects for a wide range of applications in computer vision and digital image processing research. However, color information in recorded image data, typically represented in RGB format, is not always an intrinsic property of an object itself, but rather it also depends on the illumination condition and sensor characteristic. When these factors are not properly taken into consideration, the performance of color analysis system can deteriorate significantly. This chapter investigates two common methodologies to attain reliable color description of recorded image data, color constancy and color invariance. Comprehensive overview of existing techniques are presented. Further, fundamental physical models of light reflection, and a color image formation process in typical imaging devices are discussed, which provide important underlying concepts for various color constancy and invariance algorithms. Finally, two experiments are demonstrated to evaluate the performance of representative color constancy and invariance algorithms.

**Keywords** Color · Color constancy · Color invariance · Color image processing · Illuminant estimation · Physical reflection model

## 1 Introduction

Over the last few decades, we have seen a rapid transition in the field of computer vision and digital image processing, that conventional theory initially developed in colorblind manner (i.e. algorithms based on grayscale domain) has been extended

---

D. Lee (✉) · K. N. Plataniotis  
Multimedia Lab, The Edward S. Rogers Department of Electrical and Computer Engineering,  
University of Toronto, 10 King's College Road, Toronto, Canada  
e-mail: dohyoung.lee@utoronto.ca

K. N. Plataniotis  
e-mail: kostas@comm.utoronto.ca



to incorporate color information. Especially, with the advancement in color imaging devices and multimedia technologies, color has become an essential discriminative property of objects for a wide range of applications, such as content-based image retrieval (CBIR), object tracking/recognition, and human-computer interaction (HCI) system. Consequently, we have seen inventions of color algorithms ranging from direct extensions of grayscale versions, where images are treated as three grayscale channels, to more sophisticated schemes that take into account the correlations among the color bands.

One of the critical issues associated with exploiting color in real-world applications is to infer a reliable color descriptor that is stable despite variations in imaging condition (e.g. illumination and camera characteristic). The color of an object observed in recorded image data is often not an intrinsic property of the object itself, but rather it depends also on the illumination condition under which the object is viewed. For instance, images captured under fluorescent light appear to be bluish, while images captured under tungsten filament light appear to be reddish. In addition, even under the same lighting condition, the color of objects may differ from one camera to another depending on camera characteristics. Human vision has a natural tendency which compensates such color variations and recognizes colors with high fidelity. This well known property of human vision is known as *color constancy* [15]. However, it is not trivial for computer algorithms to alleviate the influence of imaging conditions.

In general, there are two common methodologies to attain reliable color description of image data (Fig. 1): (i) *computational color constancy*, (ii) *color invariance*. The computational color constancy can be viewed as a two-stage operation where the first step is specialized on estimating the color of the scene illuminant (i.e. light source) from the image data, followed by the second step that applies correction on the image to generate a new image of the scene as if it was taken under a known, namely, canonical illuminant (i.e. reference light source).<sup>1</sup> On the other hand, color invariance methods represent images by features which remain unchanged with respect to specific imaging condition (e.g. illumination and sensor characteristic variations). A successful solution that enables stable color representation in image data is potentially useful for many applications such as:

1. **Color based segmentation:** Color is an effective cue in computer vision for identifying regions of interest/objects in the image. For example, skin color segmentation is one of the most widely used techniques in various vision applications due to the perceptual importance of skin color in human visual system. Most of existing skin color segmentation methods are derived from an assumption that the skin colors of different subjects form a small cluster in colorspace provided that the images are captured under controlled lighting setup [46]. In real-world scene, where illumination condition varies substantially depending on the prevailing illuminant, color constancy algorithm can be applied prior to executing segmentation solution to maintain stable performance [48].

---

<sup>1</sup> The choice of canonical illuminant is arbitrary, but for image reproduction application, it is commonly defined as an illuminant for which the camera sensor is balanced [4].

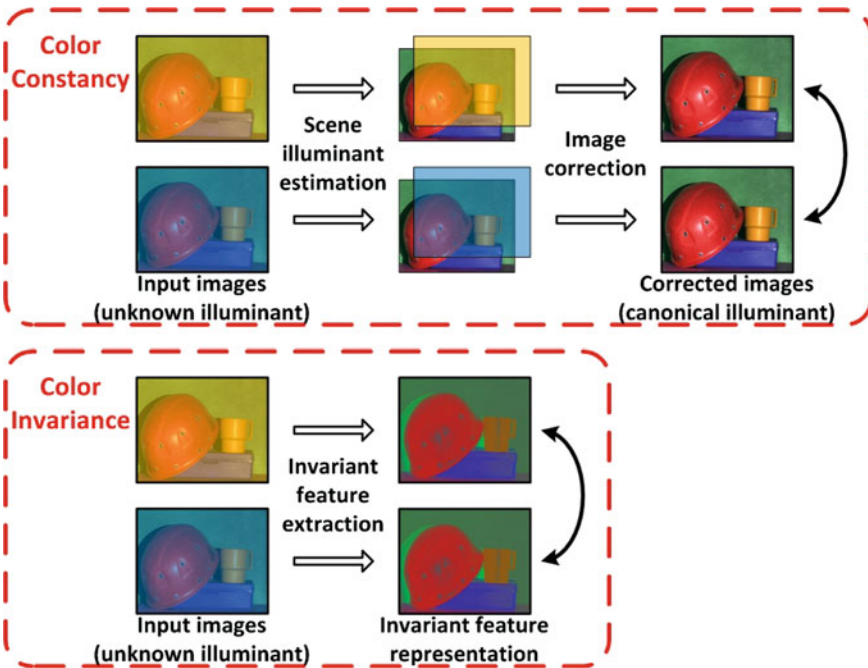


Fig. 1 Comparison of color constancy and color invariance (sample image is taken from [5])

2. **Face recognition:** Since faces play a crucial role in human social interaction, facial analysis is one of the most active areas of research in digital imaging. A key challenge in automatic face recognition technology lies in achieving high identification accuracy under uncontrolled imaging conditions. To this end, extraction of highly discriminative features which are invariant to illumination and sensor characteristic can significantly improve the machine recognition of faces [2].
3. **Novel consumer imaging devices:** With the rapid advancement of the consumer electronics, customers have higher requirements for the quality of the products. In digital photography, the visual quality of the captured image is a key criterion to judge the performance of the digital cameras. Many digital cameras provide the auto white balance (AWB) functionality, which helps less-experienced users to take a high quality photo under an unknown illumination condition that has the same chromatic representation as the ones taken under sunlight.

As shown above, maintaining stable object color appearances against varying imaging conditions is an essential requirement for many color image processing applications. In this chapter, we discuss several methodologies to achieve stable color representation with emphasis on computational color constancy and color invariance solutions. In Sect. 2, we provide background information related to fundamental principles of light reflection, as well as color image formation process in conventional imaging devices. In Sects. 3 and 4, comprehensive overview of existing color con-

stancy and color invariance solutions are provided. In Sect. 5, representative color constancy and invariance solutions are evaluated in two different experimental setups. Finally, conclusions are drawn in Sect. 6.

## 2 Color Image Formation in Digital Imaging Devices

### 2.1 Physical Reflection Model

There are three main factors that are closely related in the process of color image generation in imaging devices: (i) the underlying physical properties of the imaged surfaces (object), (ii) the nature of the light source incident upon those surfaces, (iii) the characteristics of the imaging system. A fundamental question related to stable color representation in digital image data is “how to derive the process of color image formulation in image acquisition devices using the physical laws of light?”. Especially we are interested in color images recorded in RGB system due to its dominant usage in digital imaging technology. Shafer has created a simple physical model of the reflection process, called the *dichromatic reflection model* (DRM) [61], to capture an important relationship between light sources, surface reflectances, and imaging devices. This model has become the foundation of various color constancy and color invariance methods.

According to the DRM, inhomogeneous dielectric material<sup>2</sup> consists of a clear substrate with embedded colorant particles. The model states that the light reflected from an object is caused by two types of reflections.

- **Body (diffuse) reflection:** The incident light enters the object surface where it is selectively absorbed and emitted by colorant particles within the material body. Afterwards, the light is reflected back into the air through the surface. Diffuse component of the incident light is spectrally altered depending on properties of the surface. The body reflection component is almost random in its direction.
- **Surface (specular) reflection:** The surface of an object acts like a mirror by simply reflecting any light which is incident upon it. For many inhomogeneous materials, specular component of the reflected light is spectrally similar to the incident light, as the light does not interact with the surface. Generally, surface reflection component is more directional than body component.

Thus, reflection of inhomogeneous objects can be modeled as a linear combination of diffuse and specular reflections (Fig. 2):

---

<sup>2</sup> Unlike optically homogeneous materials (e.g. metals, glasses, and crystals) which have a constant refraction index throughout the material, inhomogeneous materials (e.g. paints, ceramics, and plastics) are composed of a vehicle with many embedded colorant particles that differs optically from the vehicle. The DRM limits its discussion to optically inhomogeneous materials.

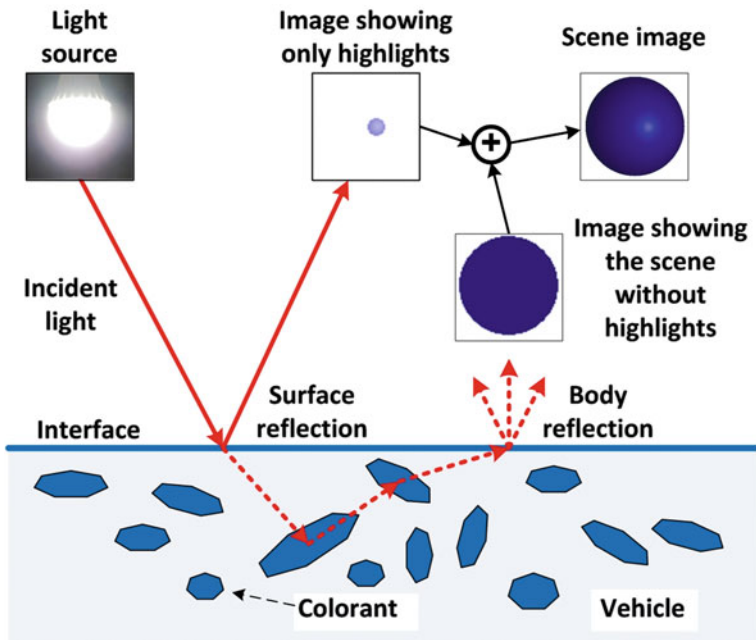


Fig. 2 The dichromatic reflection model of inhomogeneous dielectric material

$$I(\mathbf{x}, \lambda) = \underbrace{w_d(\mathbf{x})S_d(\mathbf{x}, \lambda)E(\mathbf{x}, \lambda)}_{\text{diffuse reflection term}} + \underbrace{w_s(\mathbf{x})S_s(\mathbf{x}, \lambda)E(\mathbf{x}, \lambda)}_{\text{specular reflection term}} \quad (1)$$

where  $\mathbf{x} \in \mathbb{Z}^2$  is the two-dimensional spatial position of a surface point, and  $\lambda \in \mathbb{R}$  is wavelength,  $w_d(\mathbf{x})$  and  $w_s(\mathbf{x})$  are the geometrical parameters for diffuse and specular reflection, respectively, whose values are depending on the geometric structure at location  $\mathbf{x}$  (e.g. the viewing angle, light source direction, and surface orientation).  $S_d(\mathbf{x}, \lambda)$  is the diffuse spectral reflectance function,  $S_s(\mathbf{x}, \lambda)$  is the specular spectral reflectance function, and  $E(\mathbf{x}, \lambda)$  is the spectral power distribution (SPD) function of the illumination. According to well-known Neutral Interface Reflection (NIR) assumption, for many dielectric inhomogeneous objects the index of refraction does not change significantly over the visible spectrum, thus it can be assume to be constant (i.e.  $S_s(\mathbf{x}, \lambda) \rightarrow S_s(\mathbf{x})$ ) [51]. As a result, (1) becomes:

$$I(\mathbf{x}, \lambda) = w_d(\mathbf{x})S_d(\mathbf{x}, \lambda)E(\mathbf{x}, \lambda) + w_s(\mathbf{x})E(\mathbf{x}, \lambda) \quad (2)$$

where  $w_s(\mathbf{x})$  now contains both the geometry dependent parameter and the constant reflectance of specular term.

In (2), light reflections are described using continuous spectra. However, digital cameras typically use  $k = 3$  samples to describe the light spectrum. Sample measure-

ments are obtained by filtering the input light spectrum and integrating over filtered spectrum. According to the DRM, an image taken by a digital color camera can be expressed as:

$$I_k(\mathbf{x}) = w_d(\mathbf{x}) \int_{\omega} \rho_k(\lambda) S_d(\mathbf{x}, \lambda) E(\mathbf{x}, \lambda) d\lambda + w_s(\mathbf{x}) \int_{\omega} \rho_k(\lambda) E(\mathbf{x}, \lambda) d\lambda \quad (3)$$

where  $I_k$  is the sensor response (RGB pixel values) of  $k$  color channel ( $k \in \{R, G, B\}$ ), and  $\rho_k$  is the sensor spectral sensitivity of  $k$  color channel. The integration is done over the visible spectrum  $\omega$ . It should be noted that camera noise and gain are ignored in this formulation for simplicity.

To simplify the problem, many researchers [12, 25, 49, 69] have considered a simplified scene in which all objects are flat, matte, Lambertian surfaces.<sup>3</sup> In this simplified imaging condition, the specular reflection term in (3) can be discarded and the *Lambertian reflection model* is obtained as follows:

$$I_k(\mathbf{x}) = w_d(\mathbf{x}) \int_{\omega} \rho_k(\lambda) S_d(\mathbf{x}, \lambda) E(\mathbf{x}, \lambda) d\lambda \quad (4)$$

Note that for many materials the Lambertian model does not hold in the strict sense. For example, materials with glossy surfaces often cause specularities at some spots on the material where omission of specular reflection term can be problematic. However, the Lambertian model is a good approximation since often the specularities only occupy a small part of the objects, and therefore, widely used in design of tractable color constancy and invariance solutions.

## 2.2 Linear and Non-linear RGB Representation

In the research of stable color representation, it is important to discriminate between linear and non-linear RGB representation, since unacceptable results may be obtained when they are used without caution. Linear RGB values are proportional to the intensity of the physical power radiated from an object around 700, 550, and 440 nm bands of the visible spectrum, respectively. Thus, RGB responses estimated by (4) is considered as the linear RGB. In practical image acquisition systems (e.g. digital cameras), the device RGB responses are often not linearly related to the values suggested by (4), but rather they are transformed to non-linear RGB signals through the gamma correction. Several gamma correction functions exist depending on applications, and the one that mostly related to our discussion is defined in International Electrote-

---

<sup>3</sup> The bi-directional reflectance distribution function (BRDF) describes that surface reflectance of a material  $S(\theta, \lambda)$  is a function of wavelength  $\lambda$  and imaging geometry  $\theta$ , where  $\theta$  contains information related to the directions of the incident and reflected radiance in the local coordinate system. According to the Lambertian surface model, the BRDF is a constant function of the imaging geometry so that  $S(\theta, \lambda) \rightarrow S(\lambda)$ .

chinal Commission (IEC) sRGB gamma transfer function (gamma correction used in most cameras, PCs, and printers):

$$I'_k = \begin{cases} 12.92I_k, & \text{if } I_k \leq 0.0031308 \\ 1.055I_k^{1/2.4} - 0.055, & \text{otherwise} \end{cases}, (I_k \in [0, 1]) \quad (5)$$

where  $I_k$  is the intensity of linear RGB,  $I'_k$  is the intensity of nonlinear RGB ( $k \in \{R, G, B\}$ ). It is worthwhile to note that even though the sRGB standard uses a power function with an exponent of 2.4, the transfer curve is better represented as a power function with an exponent of 2.2 [56]. The gamma correction is applied to correct a nonlinear characteristic of the cathode ray tube (CRT) display devices. The CRT display is nonlinear in a sense that the intensity of light reproduced at the screen is a nonlinear function of the input voltage. Thus, compensation for this nonlinearity is a main objective of the gamma correction process. For color images, the linear RGB values are converted into nonlinear voltages R' G' B' through the gamma correction and the CRT monitor will then convert corrected R' G' B' values into linear RGB values to reproduce the original color (See Fig. 3a). The transformation from gamma corrected R' G' B' to linear RGB values can be formulated as:

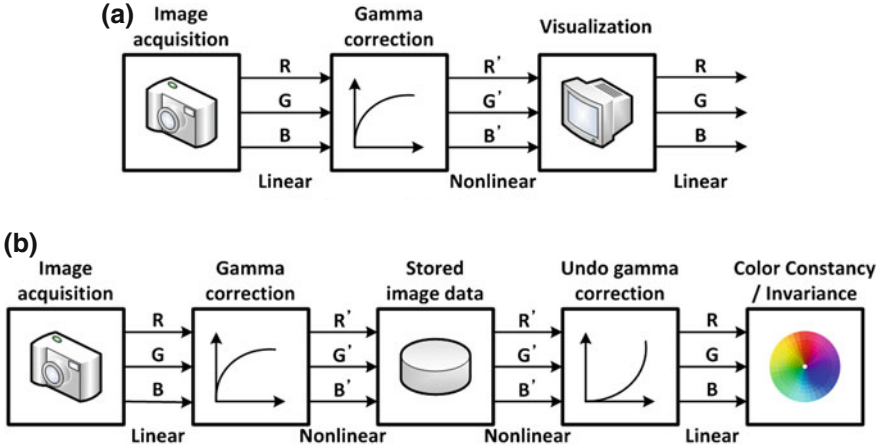
$$I_k = \begin{cases} \frac{I'_k}{12.92}, & \text{if } I'_k \leq 0.03928 \\ \left( \frac{I'_k + 0.055}{1.055} \right)^{2.4}, & \text{otherwise} \end{cases}, (I'_k \in [0, 1]) \quad (6)$$

Computer images are often stored with a gamma corrected value (e.g. stored images in JPG or TIFF file format). When we process stored nonlinear image signal, we need to linearize the intensity values by undoing gamma correction, since underlying physical reflectance models (e.g. (3) and (4)) for color constancy/invariance algorithms are derived using linear RGB values (See Fig. 3b).

### 3 Color Constancy

#### 3.1 General Formulation of Color Constancy

Essentially, the color constancy problem can be posed as estimating SPD of the scene illuminant  $E(\mathbf{x}, \lambda)$  from a given image, and then use this knowledge to recover an image which is independent of the scene illuminant. It is often unnecessary to recover the full spectra of illuminant, rather it is sufficient to represent it by the projection of  $E(\mathbf{x}, \lambda)$  on RGB domain, i.e.  $\mathbf{e}(\mathbf{x}) = [e_R(\mathbf{x}), e_G(\mathbf{x}), e_B(\mathbf{x})]^T$ , where:



**Fig. 3** Color image processing pipeline in general digital imaging devices. **a** General display pipeline. **b** General color processing pipeline

$$e_k(\mathbf{x}) = \int_{\omega} \rho_k(\lambda) E(\lambda, \mathbf{x}) d\lambda \quad (7)$$

Without prior knowledge, the estimation of  $\mathbf{e}$  is an under-constrained problem. In practice, color constancy algorithms rely on various assumptions on statistical properties of the illuminant, and surface reflectance properties to estimate  $\mathbf{e}$ . One of the widely used assumptions is that the scene illumination is constant over the entire image and thus  $E$  is simply a function of wavelength  $\lambda$  (i.e.  $E(\lambda, \mathbf{x}) \rightarrow E(\lambda)$ ).

Once the illuminant  $\mathbf{e}$  is estimated, then each RGB pixel value of the input image  $\mathbf{I}^i = [I_R^i, I_G^i, I_B^i]^T$  is mapped to the corresponding pixel of the output image  $\mathbf{I}^o = [I_R^o, I_G^o, I_B^o]^T$  (under the canonical illuminant) by a transform matrix,  $\mathcal{D}^{i,o} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ :

$$\mathbf{I}^o = \mathcal{D}^{i,o} \mathbf{I}^i \quad (8)$$

Further simplification can be done by restricting the transform matrix to a diagonal matrix. Assuming narrow-shaped sensor sensitivity functions (implying that sensor is sensitive only at a single wavelength, i.e.  $\rho_k(\lambda) = \delta(\lambda - \lambda_k)$ ), the RGB sensor response can be represented by  $I_k(\mathbf{x}) = S_d(\mathbf{x}, \lambda_k) E(\mathbf{x}, \lambda_k)$  under Lambertian reflection model in (4). Let  $E^c$  and  $E^u$  denote the canonical and the unknown illuminants, respectively, then the relationship between the RGB responses of two illuminants is:

$$\frac{I_k^c(\mathbf{x})}{I_k^u(\mathbf{x})} = \frac{S_d(\mathbf{x}, \lambda_k) E^c(\mathbf{x}, \lambda_k)}{S_d(\mathbf{x}, \lambda_k) E^u(\mathbf{x}, \lambda_k)} = \frac{E^c(\mathbf{x}, \lambda_k)}{E^u(\mathbf{x}, \lambda_k)} \quad (9)$$

Consequently, the diagonal model maps the image taken under an unknown illuminant to the canonical counterpart simply by treating each channel independently:

$$\mathbf{I}^c = \mathcal{D}^{u,c} \mathbf{I}^u \Rightarrow \begin{pmatrix} I_R^c \\ I_G^c \\ I_B^c \end{pmatrix} = \begin{pmatrix} d_R & 0 & 0 \\ 0 & d_B & 0 \\ 0 & 0 & d_G \end{pmatrix} \begin{pmatrix} I_R^u \\ I_G^u \\ I_B^u \end{pmatrix} \quad (10)$$

This model is closely related to the Von Kries hypothesis [53] which states that human color constancy is an independent gain regulation of the three cone photoreceptor signals through three different gain coefficients. Although (10) is derived for sensors with narrow spectral sensitivity function, it is a reasonably accurate model for general image sensors, and hence, this diagonal model is widely used in computational color constancy algorithms [41, 44].

### 3.2 Existing Color Constancy Solutions

As mentioned earlier, color constancy is an under-constrained problem that scene illuminant (i.e. color of the light) cannot be estimated without any assumptions. Over the last several decades, a plethora of methods have been proposed with different underlying assumptions and implementation details. Comprehensive overviews are provided by Agarwal et al. [1], Gijsenij et al. [41], and Hordley [44]. Generally, prior art solutions can be divided into two classes (Table 1): (i) static approach, (ii) learning-based approach. Static approaches predict the illumination information solely based on the content in a single image with certain assumptions about the general nature of color images, while learning-based approach requires training data in order to build a statistical model prior to estimation of illumination.

#### 3.2.1 Static Approaches

The static approaches can be further divided into two subclasses: (i) DRM based method, (ii) Lambertian model based method (low-level statistics method).

##### DRM Based Methods

DRM based method exploits physical knowledge of image formation to estimate the color of the scene illuminant. Commonly shared assumption by these methods is that RGB sensor values measured from different points on an inhomogeneous material (uniformly colored surface) will fall on a plane in RGB colorspace. In other words, the DRM in (3) can be restated as follows:

$$\mathbf{I}(\mathbf{x}) = w_d(\mathbf{x}) \mathbf{I}_D(\mathbf{x}) + w_s(\mathbf{x}) \mathbf{I}_E(\mathbf{x}) \quad (11)$$



**Table 1** Summary of representative color constancy algorithms

Group	Subcategory	Methods
Static approach	DRM based method	Specular highlights method (Lee [50]; Tominaga and Wandell [64]), Inverse-intensity chromaticity space (Tan et al. [63]), Constrained dichromatic reflection (Finlayson and Schaefer [24]), Log-relative chromaticity planar constraint (Drew et al. [16])
	Low-level statistics based method	Grayworld (Buchsbaum [12]), White Patch [30, 49], Shades of Gray (Finlayson and Trezzi [25]), Gray Edge (Weijer et al. [69]), Weighted Gray Edge (Gijssenji et al. [42])
Learning based approach	Probabilistic method	Bayesian color constancy (Brainard and Freeman [11, 32]), Bayesian method with Non-Gaussian Models (Rosenberg et al. [58]), Color by Correlation (Finlayson et al. [27])
	Gamut mapping method	Gamut mapping (Forsyth [28]), Improved gamut mapping (Barnard [3]), 2D chromaticity gamut mapping (Finlayson and Hordley [22]), Diagonal offset model gamut mapping (Finlayson et al. [21]), Color in perspective (Finlayson [19]), Edge based gamut mapping (Gijssenji et al. [40]), Gamut mapping using skin color (Bianco and Schettini [6])
	Fusion and selection based method	Combined physical and statistical (Schaefer et al. [60]), Committee-based (Cardei and Funt [13]), Consensus-based framework (Bianco et al. [8]), Natural image statistics (Gijssenji and Gevers [38]), Texture similarity (Bing et al. [52]), Indoor/outdoor classification (Bianco et al. [7]), Category correlation (Vazquez-Corral et al. [65]), Low-level feature combination (Bianco et al. [9])

where  $\mathbf{I}_D(\mathbf{x}) = [I_{R,D}(\mathbf{x}), I_{G,D}(\mathbf{x}), I_{B,D}(\mathbf{x})]^T$  is the RGB response corresponding to diffuse reflection and  $\mathbf{I}_E(\mathbf{x}) = [I_{R,E}(\mathbf{x}), I_{G,E}(\mathbf{x}), I_{B,E}(\mathbf{x})]^T$  is the one corresponding to scene illuminant.<sup>4</sup> According to (11), the RGB responses for a given surface reflectance and illuminant lie on a 2D plane (called a dichromatic plane), spanned by the two vectors  $\mathbf{I}_D(\mathbf{x})$  and  $\mathbf{I}_E(\mathbf{x})$ . If there are two distinct objects within the scene of constant illumination, two of such dichromatic planes can be obtained. Therefore, the vector representing the color of the illuminant can be estimated by intersecting two of these planes.

Lee [50] proposed a simple DRM based method to compute the illuminant chromaticity by exploiting specular highlights of multiple surfaces in the scene (Identification of highlights in the scene is important since highlight areas exhibit significant contribution of surface reflection component). Instead of using RGB domain, the

<sup>4</sup> Here, NIR hypothesis is used that the spectral reflectance distribution of the specular reflection term in the DRM is similar to the spectral energy distribution of the scene illuminant.

author identified pixels corresponds to highlight regions by projecting them into chromaticity space. From identified highlight pixels of more than two different colored surfaces (where each of pixel groups from a uniform surface yields a line segment in chromaticity space), the intersection point of the line segments is computed, which becomes the final estimate of the scene illuminant chromaticity. Similar approach is proposed by Tominaga and Wandell [64], where they exploited the singular-value decomposition (SVD) to determine a dichromatic plane from the spectral power distribution of inhomogeneous material. Although aforementioned methods provide simple solutions for characterizing the scene illuminant, they often yield suboptimal performance since the presence of small image noise dramatically affects the computation of the intersection in colorspace.

Finlayson and Schaefer [24] extended DRM based method by imposing a constraint on the colors of illumination. This method essentially adopts statistical knowledge that almost all natural and man-made illuminants fall close to the Planckian locus of black-body radiators in chromaticity space. Initially, highlight pixels are located from a uniformly colored surface to obtain a chromatic line segment in chromaticity space. Subsequently, the illuminant estimate is obtained by intersecting the dichromatic line with the Planckian locus. This method is beneficial over aforementioned methods since: (i) it present a novel scheme by combining physical model of image formation and statistical knowledge of plausible illuminants, (ii) the scene illuminant can be estimated even from a single surface, whereas aforementioned approaches requires at least two distinct surfaces.

Tan et al. [63] introduced the concept of inverse-intensity chromaticity (IIC) space for reliable estimation of scene illuminant chromaticity. Authors indicated that there is a linear relationship between the image chromaticity  $\sigma_k = \frac{I_k}{I_R+I_G+I_B}$  and the inverse-intensity  $\frac{1}{I_R+I_G+I_B}$  ( $k \in \{R, G, B\}$ ), if pixels are taken from a uniformly colored surface. Based on this linear correlation, a novel color constancy method exploiting IIC space is presented. Similar to other DRM based methods, highlight pixels are initially identified from the input image and projected into IIC space. In IIC space,<sup>5</sup> highlight pixels extracted from a uniformly colored object forms a line segment, and chromaticity of the scene illuminant can be estimated by finding the point where the line intersect with vertical axis of the IIC. One clear advantage of this method is that it is capable of handling both uniform and nonuniform surface color object in a single framework, without imposing strong constraints on illumination.

In general, DRM based solutions are advantageous since they are theoretically strong and require relatively fewer surfaces (i.e. reflectance statistics) than other methods to identify the scene illuminant. However, their performance typically depends on nontrivial preprocessing steps, such as identification of highlight pixels or segmentation, which are another challenging research questions. Moreover, pixel intensity of specular highlight content is often clipped due to limited dynamic range of sensor, turning it into an unreliable indicator of the scene illuminant. For these reasons, most of early works in this category focused on high-quality images

---

<sup>5</sup> where its horizontal axis represents the inverse-intensity and its vertical axis represents the image chromaticity  $\sigma_c$ .

of simple objects (i.e. no texture) taken under well-controlled laboratory conditions and rarely dealt with real world datasets of images [41, 44]. Recently, Drew et al. [16] presented a novel DRM based method which yields promising performance on real world datasets. Authors introduced the planar constraint that log-relative-chromaticity values<sup>6</sup> for highlight pixels are orthogonal to the light chromaticity. This constraint indicates that the geometric mean of the pixels in highlight region is an important indicator for the scene illuminant. Authors claimed that this method yields comparable performance with other training based methods on both laboratory and real world data sets (e.g. Ciurea’s grayball dataset [14] and Gehler’s color checker dataset [32]).

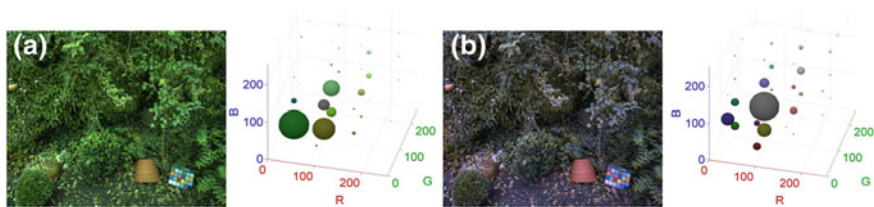
#### Low-level Statistics Methods

On the other hand, low-level statistics methods, which are based on Lambertian reflection model, provide more practical means to deal with general web contents type of input. Buchsbaum [12] proposed the Grayworld (GW) algorithm which assumes that the spatial average of surface reflectances in a scene is achromatic (i.e. gray) under the canonical illuminant. The underlying logic behind GW algorithm is that since the surface color is random and independent, it is reasonable to assume that given a sufficient large number of samples, the average surface color should converge to gray. Thus, any deviation from achromaticity in the average scene color is caused by the effect of the scene illuminant. The GW is one of the most popular algorithms for color constancy due to its simplicity. Another popular color constancy method in this category is the White Patch (WP) algorithm (also known as max RGB) [30, 49] which assumes that the scene contains a surface with perfect reflectance property. Since a surface with this property reflects the full range of light, the color of the illuminant can be estimated by identifying a perfect reflectance surface. WP algorithm estimates the scene illuminant by computing the maximum intensity of each color channel.

Although both GW and WP are well-known tractable solutions, their performances significantly deteriorate when the underlying assumptions are violated. The GW algorithm fails when the image contains large regions of uniform colors since it is unable to distinguish whether the dominant color in a scene is a superimposed cast due to the illuminant or an intrinsic color of the scene. For example, the average color of an image which contains an object in front of the dominant green background will be biased towards green rather than gray, even if it is taken under the canonical illuminant (Fig. 4). For such images, the dominant colors in a scene contribute to the shifted average color and GW algorithm will produce sub-optimal results (i.e. results in under or over-compensated output). Weijer et al. [70] proposed an extended GW hypothesis to address such issue: the average reflectance of semantic classes (such as sky, grass, road, and building) in an image is equal to the average reflectance color of that semantic class in the training dataset. Basically, this new hypothesis analyzes the input scene to assign a semantic class to it, and then, adaptively adjusts the target average color of the scene depending on the inferred semantic class. The drawback of WP algorithm is that a single maximum value may not be a reliable estimate of

---

<sup>6</sup> The log-relative-chromaticity value of a pixel is defined as the logarithm of the ratio between the chromaticity of the given pixel and the chromaticity of the scene illuminant.



**Fig. 4** An example illustrating the failure of Grayworld color constancy solution (sample image is taken from [32]). 3D RGB histograms are presented to demonstrate the distribution of color values. **a** Original image. **b** Image corrected by Grayworld

the scene illuminant, since the value can be contaminated by noise due to highlight. Instead of selecting the highest intensity directly, the stability of WP algorithm can be enhanced by choosing the intensity  $L$  such that all pixels containing intensity higher than the selected one account for specific percentage of total number of pixels in the scene (e.g. the number of pixels with higher intensity than  $L$  are less than 1 % of total pixels) [17]. Another well-known cause for the failure of WP is a potential clipping issue of maximum intensity due to limited dynamic range of image acquisition devices. Funt and Shi [31] investigated this issue by evaluating WP algorithm on 105 high dynamic range (HDR) images generated by standard multiple exposure approach. Authors demonstrated that WP algorithm yields comparable performance to other advanced methods, e.g. Gray Edge [69], provided that image data preserve full dynamic range of the original scene and they are properly preprocessed (either by a media filtering or bicubic interpolation).

Recently, several attempts have been made to generalize color constancy algorithms under a unified framework, such as *Shades of Gray* (SoG) [25] and *Gray Edge* (GE) [69]. Especially, GE algorithm by Weijer et al. [69] not only generalizes existing works but also extends color constancy methods to incorporate derivative information, i.e. edges and higher order statistics (Recall that aforementioned GW and WP algorithms use pixel intensity values to estimate the scene illuminant). GE is based on the hypothesis that the average of the reflectance differences in a scene is achromatic. This hypothesis is originated from the observation that the color derivative of images under white light sources are relatively densely distributed along with the axis coincides with the white light axis [68].

Under GE assumption, Weijer et al. [69] presented a single combined framework of color constancy technique to estimate the color of light source  $\mathbf{e}$  based on both RGB pixel value and low-level image features (i.e. derivatives) as follows:

$$\mathbf{e}(n, p, \sigma) = \frac{1}{k} \left( \int \left| \frac{\partial^n \mathbf{I}^\sigma(\mathbf{x})}{\partial \mathbf{x}^n} \right|^p d\mathbf{x} \right)^{1/p} \quad (12)$$

where  $\mathbf{I}(\mathbf{x})$  is the RGB sensor response of two-dimensional spatial coordinates  $\mathbf{x} \in \mathbb{Z}^2$ , and  $k$  is a constant so that the  $\mathbf{e}$  has a unit length. The framework of (12) covers a wide range of color constancy algorithms using three variables ( $n, p, \sigma$ ) in Table 2.

**Table 2** Description of parameters for Gray Edge framework in (12)

Parameters	Description
Order of structure $\mathbf{n}$	Defines if the method is based on the pixel intensity or based on the spatial derivatives of order $n$ . For example, $n = 0$ is for pixel based methods (e.g. GW and WP), while $n = 1, 2$ is for 1st/2nd order GE methods
Minkowski norm $\mathbf{p}$	Determines the relative weights of the multiple measurements from which the final illuminant color is estimated. For example, $(n, p) = (0, 1)$ is GW, $(n, p) = (0, \infty)$ is WP, and $(n, p) = (0, p)$ is SoG
Local averaging scale $\sigma$	Applying local averaging prior to illuminant estimation allows for the reduction of noise effect in illuminant estimation [37]. $\frac{\partial^n \mathbf{I}^\sigma}{\partial \mathbf{x}^n}$ is equal to $\mathbf{I} \otimes \frac{\partial^n G^\sigma}{\partial \mathbf{x}^n}$ , where $G^\sigma$ is a Gaussian filter with the standard deviation $\sigma$

Selection of proper parameter values for Gray Edge framework is highly important to yield good color constancy accuracy (See Sect. 5.1 for further detail).

In general, low-level statistics based methods are known to yield acceptable performance (inferior to methods based on extensive training data or complex assumptions) at low computational cost, and thus are suitable for practical applications dealing with real-world images.

### 3.2.2 Learning Based Approaches

The second type of color constancy algorithms estimate the scene illuminant using a model that is learned on training data. Three main subcategories include: (i) gamut mapping method, (ii) probabilistic method, (iii) fusion and selection based method.

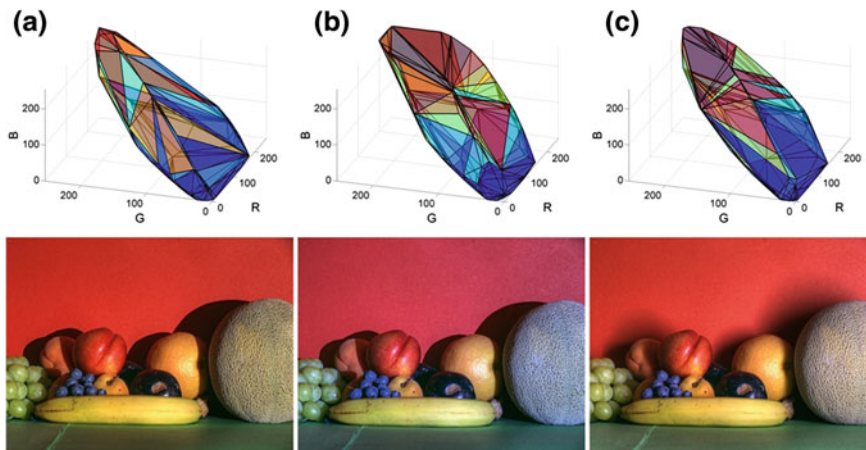
#### Gamut Mapping Methods

The gamut mapping color constancy algorithms, introduced by Forsyth [28], rely on an assumption that the range of color measurements in real-world images are restricted on a given illuminant. The limited set of colors that can occur under a given illuminant is called the *canonical gamut*  $\Gamma(\mathcal{C})$  which can be estimated by collecting a large selection of objects with different reflectances illuminated with one known light source (i.e canonical illuminant).

A key component of gamut mapping algorithm is the definition of a canonical gamut  $\Gamma(\mathcal{C})$ , which denotes the convex set of RGB sensor responses  $\mathcal{C} = \{\mathbf{I}^{c,1}, \dots, \mathbf{I}^{c,N}\}$  to  $N$  surface reflectances under a canonical illuminant  $c$ :

$$\Gamma(\mathcal{C}) = \left\{ \sum_{i=1}^N w_i \mathbf{I}^{c,i} \mid \mathbf{I}^{c,i} \in \mathcal{C}, w_i \geq 0, \sum_{i=1}^N w_i = 1 \right\} \quad (13)$$

The image gamut  $\Gamma(\mathcal{I})$  is defined in a similar way from the set of RGB values from input image. If  $I$  is the set of RGB responses recorded under the unknown



**Fig. 5** Variations of RGB image gamut of the sample scene [5] under three different illuminants. **a** Solux 3500 (spectra similar to daylight of 3500 K). **b** Sylvania 50MR16Q + blue filter (incandescent light). **c** Sylvania cool white (fluorescent light)

illumination, then all convex combinations of  $I$  denoted  $\Gamma(\mathcal{I})$  could occur under the unknown illuminant. Because only a limited number of surfaces is observed within a single image, the unknown gamut can only be approximated by the observed image gamut (Fig. 5). The next stage of gamut mapping is to find all transform matrices  $\mathcal{A} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  such that:

$$\forall \mathbf{I} \in \Gamma(\mathcal{I}), \quad \mathcal{A}\mathbf{I} \in \Gamma(\mathcal{C}) \quad (14)$$

where  $\mathbf{I}$  is a tri-vector RGB responses. A transform matrix  $\mathcal{A}$  (as shown in (10), a diagonal matrix is assumed) is a possible solution to 14 if it maps the  $m$ -th point in the image gamut  $\mathbf{I}^{i,m}$  to any point within the canonical gamut. For individual point on the convex hull of the image gamut, there exists a set of mappings taking this point to a point on the convex hull of the canonical gamut, denoted as  $\mathcal{C}/\mathbf{I}^{i,m}$ . Then the set of all convex combinations of  $\mathcal{C}/\mathbf{I}^{i,m}$ , denoted  $\Gamma(\mathcal{C}/\mathbf{I}^{i,m})$ , can be defined as the set of all mappings taking  $\mathbf{I}^{i,m}$  into point within the canonical gamut. A point in this set represents the diagonal components of a diagonal matrix  $\mathcal{A}$ , mapping  $\mathbf{I}^{i,m}$  to a point  $\mathbf{I}^{c,n}$  in the canonical gamut:

$$d_k = \mathbf{I}_k^{c,n} / \mathbf{I}_k^{i,m} \quad (15)$$

where  $d_k$  represents the diagonal components of a diagonal matrix  $\mathcal{A}$  (See (10)  $k \in \{R, G, B\}$ ). For each point in the image gamut, there is a corresponding set of diagonal mappings representing possible illuminants under which this surface could have been viewed. Therefore, the intersection of all these mapping sets, denoted  $\Gamma(\mathcal{C}/\mathcal{I})$ , is the set of mappings corresponds to a possible illuminant:

$$\Gamma(\mathcal{C}/\mathcal{I}) = \bigcap_{m=1}^N \Gamma(\mathcal{C}/\mathbf{I}^{i,m}) \quad (16)$$

Once set of all possible mapping from  $\Gamma(\mathcal{I})$  to  $\Gamma(\mathcal{C})$  is calculated then the mapping that transforms the given image gamut to a maximally large gamut (in volume) within the canonical gamut is selected as the final solution. Alternative approach is proposed in [3] by using the weighted average of the feasible set. A limitation of the original gamut mapping method is that it may produce an empty set of feasible mapping  $\Gamma(\mathcal{C}/\mathcal{I})$  under certain conditions. Finlayson et al. [21] proposed the *diagonal offset model* to address this null solution case by introducing an offset term  $\mathbf{O} = [O_R, O_G, O_B]^T$  to the diagonal model in (10):

$$\mathbf{I}^c = \mathcal{D}^{u,c} \mathbf{I}^u + \mathbf{O} \quad (17)$$

Although complexity has been increased by additional three parameters of offset term, this model is found to be robust to the failures of the diagonal model.

Instead of using three dimensional (3D) RGB values for gamut mapping, 2D approach is introduced by Finlayson and Hordley under the Color in Perspective algorithm [22]. In this method, gamut mapping is performed in 2D chromaticity space by converting input RGB response  $(I_R, I_G, I_B)$  to  $(I_R/I_B, I_G/I_B)$ .<sup>7</sup> This 2D gamut mapping solution is advantageous since computational complexity can be significantly reduced by evaluating convex hull in 2D than in full 3D.

Bianco and Schettini [6] adopted the gamut mapping approach to exploit skin memory color.<sup>8</sup> Two assumptions are used in this work: (i) skin colors form a compact cluster in the colorspace, (ii) the given input images contain human faces in the scene. The general workflow of gamut mapping approach is directly inherited in this method except that the canonical gamut is generated from a large set of skin pixels under canonical illuminant, whereas the image gamut is generated by extracting skin pixels in the input image by exploiting Viola-Jones face detector [66].

Aforementioned variations of the gamut mapping algorithm are restricted to the use of pixel values to estimate the illuminant. Gijsenij et al. [40] extended the gamut mapping to incorporate the derivative structure of an image, similar to the Gray edge algorithm. Their observations on large data sets of laboratory and real-world scene showed that: (i) in case of deviations of the diagonal model, e.g. existence of diffuse light or object reflection, the derivative-based method outperforms the pixel-based gamut mapping, (ii) the combination of the different orders of derivatives provides more accurate estimation performance.

Compared to the other algorithms, gamut mapping algorithms are computationally expensive because the convex hull must be computed from the input image [4, 41, 44]. However, a simplified realization of gamut mapping using a convex programming

<sup>7</sup> The instability of the method when  $I_B = 0$  can be addressed by exploiting histogram based method rather than using input RGB values directly [18].

<sup>8</sup> The memory color refers to a group of colors that has the most perceptive impact on the human visual system, such as skin color, blue sky, and green foliage [71].

[26] is now very common in this area. This implementation represents a canonical gamut as the intersection of a group of half-spaces<sup>9</sup> rather than as the set of convex hull points. Hence, gamut mapping problem can be simplified to find an optimal illuminant estimate subject to linear inequalities. Overall, the gamut mapping solution is one of the most successful solutions to the color constancy problem and yield good practical performance on real images.

### Probabilistic Methods

Probabilistic methods use a statistical model to quantify the probability that each of a set of plausible illuminants is the scene illuminant. The color-by-correlation by Finlayson et al. [27] is one of the most representative algorithms in this category, which is derived under an assumption that illumination condition is uniform over the scene. It makes use of a correlation matrix  $M$ , which encodes knowledge about the statistical relationship between plausible illuminants and observed image colors. The probability of set of colors (represented in 2D chromaticity vectors) generated by each illuminant is encoded in the columns of a correlation matrix  $M$ , where the number of columns in  $M$  is equal to the number of plausible illuminants. By changing the entries of the correlation matrix  $M$ , several methods can be reformulated in the color-by-correlation framework. In fact, color-by-correlation was originally a different implementation of a discrete version of 2D gamut mapping, wherein matrix entries were boolean. Then, it was improved by replacing the booleans with probability.

Formally, the essence of the color-by-correlation method [27] is to find the illuminant  $E$  from a set of feasible lights which correlates better with the colors presented in input image data  $C_{im}$ . The probability that the scene illuminant is  $E$ , given the image data  $C_{im}$ , can be written as:

$$P(E|C_{im}) = \frac{P(C_{im}|E)P(E)}{P(C_{im})} \quad (18)$$

Noting that for a given image  $P(C_{im})$  is constant and assuming that observed chromaticities are independent (i.e.  $P(C_{im}|E)$  itself becomes the product of the probabilities of observing the individual chromaticities  $c$ , given the illuminant  $E$ ), (18) reduces to:

$$P(E|C_{im}) = \left[ \prod_{\forall c \in C_{im}} P(c|E) \right] P(E) \quad (19)$$

If prior information about scene illuminant  $P(E)$  is not available (i.e. the range of occurrence of the illuminant is unknown), often it is further assumed that all illuminants are equally likely:

---

<sup>9</sup> A plane divides 3-dimensional space into two half-spaces. A half-space can be specified by a linear inequality.



$$P(E|C_{im}) = \prod_{\forall c \in C_{im}} P(c|E) \quad (20)$$

Finally, given the posteriori probability distribution, one light source is selected as scene illuminant, e.g. using maximum likelihood.

Other approaches in this category operate according to the same underlying principle but they differ both in terms of how they encode prior probability of illuminants/reflectances as well as their implementation details. For example, a simple Gaussian prior model is used to represent the distribution of surface reflectances by Brainard et al. [11], whereas a non-parametric model is used by Rosenberg et al. [58]. The strength of the probabilistic approaches is the fact that they incorporate as much prior information (related to plausible illuminants and observed image colors) as possible into the problem formulation and the illumination condition is obtained from elegant probabilistic knowledge. However, potential weaknesses of these methods are: (i) they are computationally expensive, (ii) the performance is limited by the degree to which real image data conform to the statistical priors assumed by the algorithms.

#### Fusion and Selection based Methods

As mentioned earlier, color constancy algorithms rely on specific assumptions on the scene reflectances/illuminants properties, which cannot always hold true for all types of input scene. Therefore, approaches in this category attempts to yield accurate estimate of scene illuminant on a wide range of test images by either selecting the most appropriate method or combining multiple outcomes for a given configuration. Earlier approach in this category simply computes the scene illuminant by obtaining the mean value of individual color constancy methods without considering the content of given scene [13].

Alternatively, Gijsenij and Gevers [38] introduced a strategy to select and combining color constancy algorithms based on the statistical contents of the input image. This method is derived from an observation that if a limited number of edges are present in an input scene, it is likely that pixel-based color constancy methods will outperform edge-based methods (since there is lack of information for edge-based methods to work with), even though edge-based methods are considered superior on average [40, 69]. To this end, the Weibull parameterization [33] is exploited to express image characteristics:

$$w(x) = C \exp\left(-\frac{1}{\gamma} \left|\frac{x}{\beta}\right|^\gamma\right) \quad (21)$$

where  $x$  is the gradient image obtained by filtering the image by Gaussian derivative filters,  $C$  is a normalization constant,  $\beta$  and  $\gamma$  are contrast and grain size parameter of the distribution, respectively. Two parameters provide essential information for selecting the optimal color constancy method for a given input image, such as the number of edges, the amount of texture, and the SNR ratio. For example,

Fig. 6 demonstrates that increased degree of contrast yields higher value for  $\beta$ , while increased amount of fine textures results in higher  $\gamma$  value.

In the training phase of this framework, Weibull parameters  $\theta = (\beta, \gamma)$  are obtained for all training images and the scene illuminants are estimated for each of the training image using several color constancy methods  $m \in \{GW, WP, SoG, GE\}$ . Then, each training image  $t$  is assigned with the optimal method  $m_t$  which gives the minimum difference between groundtruth illuminant and the estimated illuminant. Subsequently, a classifier is learned by representing  $p(\theta_t | m_t)$ , the likelihood of the observed weibull feature  $\theta_t$  given the color constancy algorithm  $m_t$ , as a linear combination of multiple Gaussian distributions. In the testing phase, the learned classifier is applied to assign the algorithm that maximizes the posterior probability to a given test image based on its Weibull parameters. The authors indicated that on the Ciurea’s grayball dataset [14], an increase in color constancy performance upto 20 % (median angular error) can be achieved compared to the best-performing single algorithm. Li et al. [52] indicated that the global Weibull distribution parameters alone are not enough to describe the texture characteristic of the entire scene and also exploits local Weibull features (four additional features corresponding to top, bottom, left, right halves of the image) to improve illumination estimation performance.

Instead of the Weibull parameters, Bianco et al. [9] made use of multiple low-level image features, including color, texture, and edge. Their multiple feature-based algorithm is based on five independent color constancy algorithms and a classification step that automatically selects which algorithm to use for given input image. The

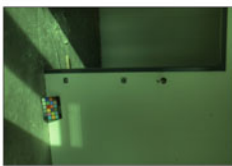
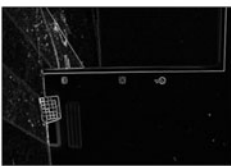
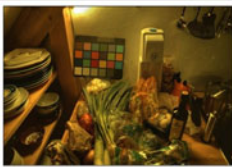



Image type	Sample image	Gradient image	Weibull distribution
Image with simple background			$\beta = 0.04, \gamma = 0.75$ $(\beta, \gamma) = (0.04, 0.75)$
Image with moderate contrast background			$\beta = 0.09, \gamma = 0.75$ $(\beta, \gamma) = (0.09, 0.75)$
Image with complex background			$\beta = 0.38, \gamma = 1.48$ $(\beta, \gamma) = (0.38, 1.48)$

Fig. 6 Three natural images of different visual appearance and associated Weibull distribution. Sample images are taken from [32]

classifier is trained on features automatically extracted from the images (e.g. color histogram, edge direction histogram, wavelet coefficients statistics, color moments, and so forth). Another combinational strategy was proposed by Bianco et al. [7] using an indoor-outdoor image classification. During the training stage, this framework divides images into indoor or outdoor scenes and finds the most suitable color constancy method for each category. Then, the appropriate method is determined for any given new image based on its indoor/outdoor category. This method has shown its limitation since the distinction between indoor and outdoor is rather arbitrary. Recently, Vazquez-Corral et al. [65] proposed the color category hypothesis based method, which weights the set of feasible illuminants according to their ability to anchor the colors of an image to basic color categories.

Overall, key idea of fusion and selection based methods is to yield enhanced color constancy accuracy by finding the weighting parameters for multiple algorithms using pre-defined combination rule or extracted information from input scene. Generally, methods in this class outperforms other methods in terms of accuracy, but there is inherited computational overheads compared to single algorithm approaches.

### 3.2.3 Color Constancy for Multiple Illuminants

Review of the color constancy solutions in previous section reveals that most existing algorithms assume the scene illumination to be uniform throughout the scene. Such an assumption significantly simplifies the design of an algorithm, but often is not justifiable for real-world scene due to the presence of multiple illuminants, e.g. indoor scene lit by both indoor illuminant and sunlight through windows. Recently several methods have been proposed to deal with the multiple illuminant scenario, such as Riess et al. [57], Bleier et al. [10], and Gijsenij et al. [43]. General workflow of multiple illuminant based color constancy solution is as follow:

1. Image partition: the given image is partitioned into local patches with an assumption that the color of the illuminant is uniform with a local patch.
2. Local illuminant estimation: the local illuminants are estimated using existing single solutions. For example, DRM based method is used by Riess et al. [57] while low-level statistics based solutions, e.g. GW, WP, and GE, are used by Gijsenij et al. [43].
3. Combination of local estimates: the local estimates are merged if adjacent regions exhibit illuminants of similar chromaticity characteristics. Additional smoothing filter can be applied to avoid abrupt transition of illumination between local regions.

The most important steps in the multiple illuminant based framework is the image partition since it has significant impact on the color constancy accuracy and complexity of the solution. Gijsenij et al. [43] compared various strategies to divide the input scene (e.g. color-based segmentation, fixed size grid segmentation), and concluded that segmenting the scene into fixed size rectangular grids (with a patch size of approximately 2–5 % of the entire image dimension) generally yields good perfor-

mance. Overall, aforementioned algorithms have shown that existing color constancy solutions can be extended to deal with more realistic scenes where there are multiple light sources present in an image. One practical problem involved in this line of research is that validation dataset which can facilitate multiple illuminant case is quite limited at this point. Most existing datasets, e.g. Ciurea’s grayball dataset [14] and Gehler’s color checker dataset [32] do not provide locally varying groundtruth of scene illumination to evaluate multiple illuminant solutions. Alternatively, some latest datasets, such as Bleier’s multiple illuminant groundtruth set [10], allow for the validation of such solutions since it provides pixel-level groundtruth information of image data.

## 4 Color Invariance

In Sect. 3, we mainly investigated color constancy solutions, which produce a transformed version of the input image as if the scene is rendered under the canonical illuminant. Essentially, the produced image by color constancy algorithms is in RGB representation, and thus they can be conveniently applied to any target applications designed to operate on RGB domain. Another class of solutions to attain stable object color appearance is to represent images by features which are invariant with respect to specific imaging condition. Unlike color constancy solutions, the invariant feature based methodologies neither require an estimation of the scene illuminant, nor produce output images of RGB domain. It is worthwhile to note that color invariant features can be calculated on the images that is pre-processed by color constancy algorithm.

As we discussed in Sect. 2.1, the interaction between light sources, imaging devices, and objects in the scene can be described by physical reflection models. On the basis of these models, algorithms have been proposed which are invariant to different imaging conditions. In this section, existing color invariance methods are classified into two categories: (i) illumination independent invariance, (ii) device independent invariance.

### 4.1 Illumination Independent Color Invariants

In this section, we review several representations which are invariant to varying illumination condition. Suppose we have a small surface patch of an object, then the DRM states that the RGB sensor values are given as:

$$I_k = w_d(\mathbf{n}, \mathbf{s}) \int_{\omega} \rho_k(\lambda) S_d(\lambda) E(\lambda) d\lambda + w_s(\mathbf{n}, \mathbf{s}, \mathbf{v}) \int_{\omega} \rho_k(\lambda) E(\lambda) d\lambda \quad (22)$$

where  $\mathbf{n}$  is the surface patch normal,  $\mathbf{s}$  is the direction of the light source, and  $\mathbf{v}$  is the direction of the viewer.<sup>10</sup> To simplify the problem, many color invariant features are derived using an assumption that objects in the scene is lit by white illumination (i.e. all wavelengths within the visible spectrum have consistent energy, i.e.  $E(\lambda) = E$ ). Sensor values under white illumination can be represented as:

$$I_k = w_d(\mathbf{n}, \mathbf{s})E \int_{\omega} \rho_k(\lambda)S_d(\lambda)d\lambda + w_s(\mathbf{n}, \mathbf{s}, \mathbf{v})E \int_{\omega} \rho_k(\lambda)d\lambda \quad (23)$$

Further, we can introduce a new variable representing the color of body reflectance,  $c_k = \int_{\omega} \rho_k(\lambda)S_d(\lambda)d\lambda$ , then RGB sensor values can be given by [35]<sup>11</sup>:

$$I_k = w_dc_kE + w_s\rho E \quad (24)$$

Equation (24) demonstrates that RGB representation is sensitive to not only sensor characteristics and surface reflectances, but also other factors, such as viewing directions, object positioning, illumination intensity, and sensor characteristics. In the following section, various invariant color features are summarized and their invariant characteristics towards varying imaging conditions are analyzed.

**Normalized RGB** One of the simplest and most widely used color invariance is normalized RGB, also known as a chromaticity representation. This invariants can be obtained by normalizing RGB values by their intensity as follows:

$$r = \frac{R}{R + G + B}, \quad g = \frac{G}{R + G + B}, \quad b = \frac{B}{R + G + B} \quad (25)$$

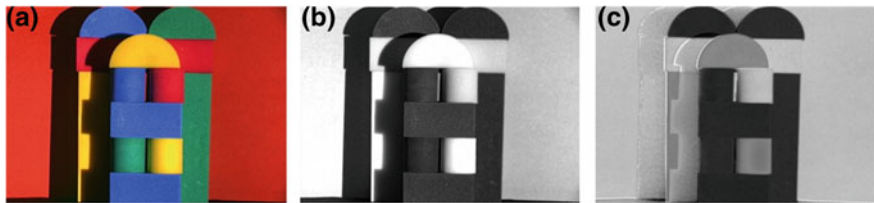
If we assume that the scene only contains matte, dull surfaces (thus, the specular reflection term can be discarded) under a white illuminant, the normalized red component can be expressed as follows:

$$r = \frac{w_dc_RE}{w_dc_RE + w_dc_GE + w_dc_BE} = \frac{c_R}{c_R + c_G + c_B} \quad (26)$$

The coefficient denoting the interaction between the white light source and surface reflectance (represented by  $w_d(\mathbf{n}, \mathbf{s})$ ) is cancelled out. Consequently, the normalized RGB values are invariant for the surface orientation, illumination direction, and illumination intensity. In other words, assuming Lambertian reflection model and white illumination, the normalized-rgb value is only dependent on the sensor characteristics  $\rho_k(\lambda)$  and the surface reflectance  $S_d(\lambda)$ . Figure 7 demonstrates that shadow boundaries are strongly correlated with original red band edges, while it

<sup>10</sup> Here, geometric terms  $w_d$  and  $w_s$  of DRM in (3) are restated by specifying their dependence on the surface normal, the direction of the light source, and the direction of the viewer.

<sup>11</sup> Here, we assume the integrated white light condition (i.e. the area under the sensor spectral sensitivity function is approximately the same for all three channels)[35] such that  $\int_{\omega} \rho_R(\lambda)d\lambda = \int_{\omega} \rho_G(\lambda)d\lambda = \int_{\omega} \rho_B(\lambda)d\lambda = \rho$ .



**Fig. 7** Visual comparison of Red and normalized Red components of sample image [5]. **a** Original RGB image. **b** Red. **c** Normalized Red

is weakly associated with normalized red boundaries. It is worthy to note that normalized RGB is sensitive to highlights since it is still dependent on the specular reflection term.

$C_1C_2C_3$  **Color Model** Color invariance  $C_1C_2C_3$  is defined as:

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} \arctan(R/(\max(G, B))) \\ \arctan(G/(\max(R, B))) \\ \arctan(B/(\max(R, G))) \end{pmatrix} \quad (27)$$

The  $C_1C_2C_3$  has similar invariant characteristic as the normalized RGB that for matte surfaces illuminated by white illuminant, it is independent for the changes of surface orientation, illumination direction, and illumination intensity:

$$C_1 = \arctan \left[ \frac{w_d c_R E}{\max(w_d c_G E, w_d c_B E)} \right] = \arctan \left[ \frac{c_R}{\max(c_G, c_B)} \right] \quad (28)$$

$O_1O_2O_3$  **Color Model** Unlike RGB colorspace, three channels in the opponent colorspace are well decorrelated and defined as follows:

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} (R - G)/\sqrt{2} \\ (R + G - 2B)/\sqrt{6} \\ (R + G + B)/\sqrt{3} \end{pmatrix} \quad (29)$$

In  $O_1O_2O_3$  colorspace, the chromaticity information is represented by  $O_1$  (red-green channel) and  $O_2$  (yellow-blue channel), while the intensity information is given by  $O_3$ . Gevers and Stokman [36] indicated that  $O_1$  and  $O_2$  components of  $O_1O_2O_3$  opponent colorspace have invariant properties. Assuming that the scene is under an white illumination, the channel  $O_1$  is independent of highlights as specular reflection term is cancelled out completely:

$$O_1 = \frac{\{(w_d c_R E + w_s \rho E) - (w_d c_G E + w_s \rho E)\}}{\sqrt{2}} = \frac{\{w_d c_R E - w_d c_G E\}}{\sqrt{2}} \quad (30)$$

(30) indicates that  $O_1$  is still sensitive to object geometry, illumination direction, and the illumination intensity. The channel  $O_2$  has identical invariant characteristic as  $O_1$ . Note that  $O_3$  has no invariant property at all.

**HSI Color Model** This color system corresponds more closely to the human perception of color than aforementioned systems.<sup>12</sup> Hue is the attribute of a visual sensation related to the dominant wavelength in a mixed light wave. In other words, hue refers to the property of color perception by which an object is judged to red, green, blue, and so forth. Hue is defined as follows:

$$H = \arctan \left[ \frac{\sqrt{3}(G - B)}{(R - G) + (R - B)} \right] \quad (31)$$

Saturation refers to purity of a color that varies from gray through pastel to saturated colors, and is defined as:

$$S = 1 - \frac{\min(R, G, B)}{R + G + B} \quad (32)$$

By substituting (24) into (31), it can be seen that hue is invariant to surface orientation, illumination direction, intensity, as well as highlights under white illumination with both matte and glossy materials:

$$\begin{aligned} H &= \arctan \left[ \frac{\sqrt{3}\{(w_d c_G E + w_s \rho E) - (w_d c_B E + w_s \rho E)\}}{2(w_d c_R E + w_s \rho E) - (w_d c_G E + w_s \rho E) - (w_d c_B E + w_s \rho E)} \right] \\ &= \arctan \left[ \frac{\sqrt{3}(c_G - c_B)}{2c_R - c_G - c_B} \right] \end{aligned} \quad (33)$$

On the other hand, saturation is an invariant feature for matte, dull surface illuminated by white illumination:

$$S = 1 - \frac{\min(w_d c_R E, w_d c_G E, w_d c_B E)}{w_d c_R E + w_d c_G E + w_d c_B E} = 1 - \frac{\min(c_R, c_G, c_B)}{c_R + c_G + c_B} \quad (34)$$

**Color Ratio Model** Aforementioned invariant features are all derived under an assumption that the scene is lit by white illuminant. Gevers and Smeulders [35] proposed a color invariance called  $m_1 m_2 m_3$ , which defines the color ratio between two neighboring pixels,  $\mathbf{x}_1$  and  $\mathbf{x}_2$  as follows:

$$\begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = \begin{pmatrix} \{R(\mathbf{x}_1)G(\mathbf{x}_2)\}/\{R(\mathbf{x}_2)G(\mathbf{x}_1)\} \\ \{R(\mathbf{x}_1)B(\mathbf{x}_2)\}/\{R(\mathbf{x}_2)B(\mathbf{x}_1)\} \\ \{G(\mathbf{x}_1)B(\mathbf{x}_2)\}/\{G(\mathbf{x}_2)B(\mathbf{x}_1)\} \end{pmatrix} \quad (35)$$

<sup>12</sup> Although there exist many different ways to compute HSI representation, here, we use a definition given in [55].

Assuming no glossy surfaces in the scene, and the narrow-shaped sensor sensitivity functions (i.e.  $\rho_k(\lambda) = \delta(\lambda - \lambda_k)$ ), the measured RGB sensor values at spatial location  $\mathbf{x}_1$  can be approximated as:

$$I_k(\mathbf{x}_1) = w_d(\mathbf{x}_1, \mathbf{n}, \mathbf{s}) S_d(\mathbf{x}_1, \lambda_k) E(\mathbf{x}_1, \lambda_k) \quad (36)$$

If we further assume that neighbor pixels have the same surface orientation (i.e.  $w_d(\mathbf{x}_1, \mathbf{n}, \mathbf{s}) = w_d(\mathbf{x}_2, \mathbf{n}, \mathbf{s})$ ), and the spectral characteristic of the illumination is locally constant (i.e.  $E(\mathbf{x}_1, \lambda_k) = E(\mathbf{x}_2, \lambda_k)$ ), the color ratio  $m_1$  is insensitive to variations in illumination intensity, color, direction, as well as surface orientation:

$$\begin{aligned} m_1 &= \frac{w_d(\mathbf{x}_1, \mathbf{n}, \mathbf{s}) S_d(\mathbf{x}_1, \lambda_R) E(\mathbf{x}_1, \lambda_R) \cdot w_d(\mathbf{x}_2, \mathbf{n}, \mathbf{s}) S_d(\mathbf{x}_2, \lambda_G) E(\mathbf{x}_2, \lambda_G)}{w_d(\mathbf{x}_2, \mathbf{n}, \mathbf{s}) S_d(\mathbf{x}_2, \lambda_R) E(\mathbf{x}_2, \lambda_R) \cdot w_d(\mathbf{x}_1, \mathbf{n}, \mathbf{s}) S_d(\mathbf{x}_1, \lambda_G) E(\mathbf{x}_1, \lambda_G)} \\ &= \frac{S_d(\mathbf{x}_1, \lambda_R) S_d(\mathbf{x}_2, \lambda_G)}{S_d(\mathbf{x}_2, \lambda_R) S_d(\mathbf{x}_1, \lambda_G)} \end{aligned} \quad (37)$$

Similarly,  $m_2$  and  $m_3$  hold the same invariant properties.

**SUV Color Model** The SUV color model, proposed by Mallick et al. [54] separates the highlight and diffuse components into S channel and UV channels, respectively, based on knowledge of the color of the illuminant. The SUV model is also known as data-dependent model since the transformation between RGB and SUV depends on the color of the scene illuminant.

The SUV representation can be obtained from linear transformation of RGB color space by rotating the coordinate axes in a way that one of the axes gets aligned with the illuminant color vector  $\mathbf{s} = [s_R, s_G, s_B]^T$ . Formally, the transformation can be defined according to  $\mathbf{I}_{SUV} = \mathcal{R} \mathbf{I}_{RGB}$  using transformation matrix  $\mathcal{R}$  that satisfies  $\mathcal{R} \mathbf{s} = [1, 0, 0]^T$  (assuming that we are aligning red axis of RGB to S axis of SUV). It is worthwhile to mention that if the color of the scene illuminant  $\mathbf{s}$  is unknown, then it can be estimated from color constancy algorithms prior to transformation.

Unlike other invariants which are designed to be invariant to geometry information (i.e. information encoded in geometric parameter  $w_d(n, s)$  of (22)) in order to isolate information related to material reflectance properties, the U and V components of SUV representation preserves geometry information while they are invariant to specular reflection. Therefore, practically, the UV invariants allows for an application of Lambertian model based algorithms to a broader class non-Lambertian surfaces, such as glossy materials [72].

Table 3 summarizes the characteristics and applicable conditions of various color invariance methods presented in this section. Aforementioned invariances are derived from RGB intensity values. Some researches extended color invariant features to incorporate higher-order derivative structure information. Gevers and Stokman [36] proposed a color edge classification rule by applying derivatives to pixel based invariant representation. For example, image edge map based on  $O_1 O_2$  feature is used to determine edges that are less sensitive to highlight, since  $O_1 O_2$  is independent of



**Table 3** Color invariant features and their properties to specific imaging conditions

Applicable condition	Color feature	Invariant characteristics
Image recorded under well-controlled illumination	RGB	No invariant properties
Image containing matte and dull surfaces under white illumination	I (intensity)	No invariant properties
	Normalized RGB	Invariant to illumination intensity, illumination direction, surface orientation
	$C_1C_2C_3$	Invariant to illumination intensity, illumination direction, surface orientation
Image containing both matte and shiny surfaces under white illumination	S (saturation)	Invariant to illumination intensity, illumination direction, surface orientation
	$O_1O_2$	Invariant to highlight
Image containing matte and dull surfaces under colored light. Sensor with narrow spectral sensitivity functions	H (hue)	Invariant to highlight, illumination intensity, illumination direction, surface orientation
	$m_1m_2m_3$	Invariant to illumination color, intensity, illumination direction, surface orientation
Image containing both matte and shiny surfaces under known illumination condition	UV	Invariant to highlight

highlight as shown in (30). Consequently, authors made use of derivative structures estimated from RGB,  $O_1O_2$ , and  $C_1C_2$  components to classify image edges into three types<sup>13</sup>: (i) shadow, (ii) highlight, (iii) material edges.

Alternatively, Weijer et al. [67] derived a set of invariant derivatives called *Quasi-invariants*. They raised a stability issue associated with directly deriving invariant derivatives from pixel-based invariances; normalized RGB is unstable near zero intensity and hue is undefined on the black-white axis, and thus taking the derivative of such invariants inherits the same stability issue. Quasi-invariant features address this stability issue by estimating the spatial derivative of an input color image, followed by decomposing a derivative image into three variant directions. The advantage of quasi-invariances are improved noise stability and discriminative power, and thus, they can be effectively used in shadow-edge independent image segmentation, and edge classification applications. Recently, quasi-invariant features are exploited by Gijssen et al. [42] to establish a novel color constancy solution, which extends Gray Edge algorithm. The authors indicated that different edge types (e.g. material, shadow, or highlight edges) have a distinctive influence on the performance of the

<sup>13</sup> Shadow edges indicate the shadow of the illuminated object. Highlight edges are generated by the specular property of the object surface. Material edges indicate the discontinuity due to a change of surface material

illuminant estimation, and consequently, weighted Gray Edge algorithm is introduced by assigning different weight for estimation of scene illuminant.

## 4.2 Device Independent Color Invariants

As mentioned in Sect. 4.1, most of color invariance methods are derived from physical image formation models, such as the DRM and the Lambertian reflection model. However, application of these models to *uncalibrated* real-world image/video data may leads to problems. This is especially true when image database are composed of images recorded by different acquisition devices (e.g. images randomly collected from web). In practice, typical digital cameras apply the gamma correction to the captured image/video data, introducing non-linearities in the sensor RGB output (See Sect. 2.2). The nonlinear RGB response of the camera can be represented as:

$$(I'_R, I'_G, I'_B) = (\alpha \cdot I_R^\gamma, \alpha \cdot I_G^\gamma, \alpha \cdot I_B^\gamma) \quad (38)$$

where  $I'_k$  indicates nonlinear response of  $k$  color channel,  $\gamma$  is the gamma parameter, and  $\alpha$  is the camera gain parameter (Here, we assumed that  $\gamma$  and  $\alpha$  are identical for all three color channels). A gamma different from 1 implies that there exists a power function relationship between physical intensities (i.e. linear RGB) and recorded sensor RGB values.<sup>14</sup> With non-unity gamma value, the invariant property of color invariance methods derived from Sect. 4.1 changes due to two factors ( $\alpha$ ,  $\gamma$ ). For example, the hue in (31) is no longer invariant to surface orientation and illumination direction with nonlinear response:

$$\begin{aligned} H &= \arctan \left[ \frac{\sqrt{3}(I'_G - I'_B)}{(I'_R - I'_G) + (I'_R - I'_B)} \right] = \arctan \left[ \frac{\sqrt{3}(\alpha I_G^\gamma - \alpha I_B^\gamma)}{(\alpha I_R^\gamma - \alpha I_G^\gamma) + (\alpha I_R^\gamma - \alpha I_B^\gamma)} \right] \\ &= \arctan \left[ \frac{\sqrt{3}\{(w_d c_G E + w_s \rho E)^\gamma - (w_d c_B E + w_s \rho E)^\gamma\}}{2(w_d c_R E + w_s \rho E)^\gamma - (w_d c_G E + w_s \rho E)^\gamma - (w_d c_B E + w_s \rho E)^\gamma} \right] \\ &= \arctan \left[ \frac{\sqrt{3}\{(w_d c_G + w_s \rho)^\gamma - (w_d c_B + w_s \rho)^\gamma\}}{2(w_d c_R + w_s \rho)^\gamma - (w_d c_G + w_s \rho)^\gamma - (w_d c_B + w_s \rho)^\gamma} \right] \end{aligned} \quad (39)$$

which is proven by substituting (24) into (39).<sup>15</sup> Moreover, it can be seen that the hue is dependent on  $\gamma$  parameter as well. In practice, the calibration data for acquisition devices are often unavailable (i.e.  $\alpha$  and  $\gamma$  values are unknown), and hence, we

<sup>14</sup> It is well known that Apple systems are calibrated to gamma value of 1.8, whereas most other systems are calibrated to gamma value of 2.2. It implies that images of same scene may appear differently dependent on the target systems.

<sup>15</sup> Since the hue is dependent on  $w_d(\mathbf{n}, \mathbf{s})$  and  $w_s(\mathbf{n}, \mathbf{s}, \mathbf{v})$ , it is no longer invariant to surface orientation and illumination direction with the nonlinear response. It is still invariant to illumination intensity as  $E$  term can be cancelled out.

should take into account camera specific parameters when we derive color invariance methods.

The *logarithmic hue*,  $H_{log}$  by Finlayson and Schaefer [23] proposed a simple method to retrieve stable descriptor towards unknown camera characteristic:

$$H_{log} = \arctan \left[ \frac{\log R' - \log G'}{\log R' + \log G' - 2 \log B'} \right] \quad (40)$$

where  $R'$ ,  $G'$ ,  $B'$  are nonlinear (i.e. gamma-corrected) red, green, and blue values obtained from camera. The logarithmic hue is invariant to intrinsic photometric camera parameters.

$$\begin{aligned} H_{log} &= \arctan \left[ \frac{\log(\alpha \cdot I_R^\gamma) - \log(\alpha \cdot I_G^\gamma)}{\log(\alpha \cdot I_R^\gamma) + \log(\alpha \cdot I_G^\gamma) - 2 \log(\alpha \cdot I_B^\gamma)} \right] \\ &= \arctan \left[ \frac{\log(\alpha) + \gamma \cdot \log(I_R) - \log(\alpha) - \gamma \cdot \log(I_G)}{\log(\alpha) + \gamma \cdot \log(I_R) + \log(\alpha) + \gamma \cdot \log(I_G) - 2(\log(\alpha) + \gamma \cdot \log(I_B))} \right] \\ &= \arctan \left[ \frac{\log I_R - \log I_G}{\log I_R + \log I_G - 2 \log I_B} \right] \end{aligned} \quad (41)$$

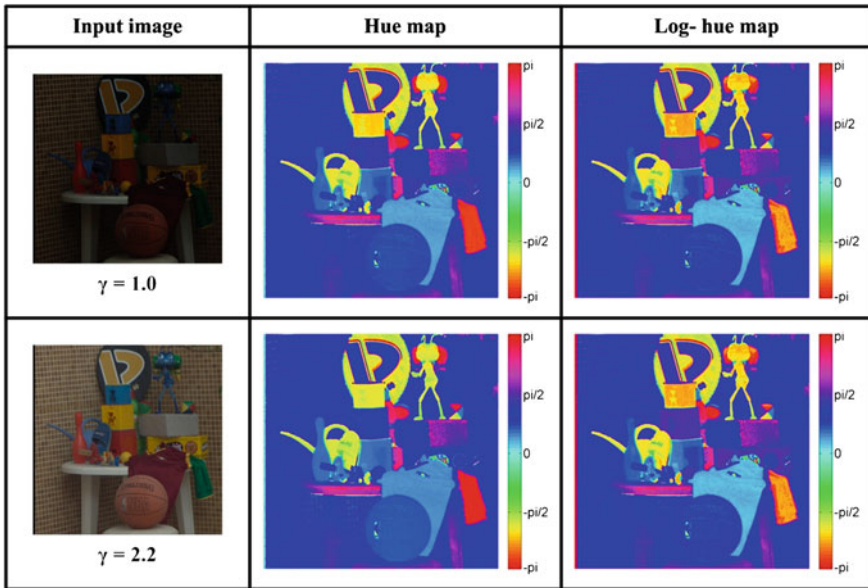
Figure 8 illustrates visual comparison of conventional hue and logarithmic hue, when the input scene is taken with unity gamma and non-unity gamma. The sample image is generated from hyperspectral data [29], which contains a collection of measured surface reflectances. The daylight illuminant spectra (CIE D65) is used to produce RGB pixel responses, and two distinct gamma values are applied to simulate the final RGB images with different gamma parameters. As can be seen, logarithmic hue feature remains unchanged for varying gamma parameters.

Similar line of work is proposed by Finlayson and Xu [20], introducing a color invariant feature that cancels out dependencies due to both illumination and device. The invariant feature is obtained in following sequence: (i) initially log is taken to the RGB image, and for each pixel, the pixel mean of the log RGB responses are subtracted individually, (ii) the mean of all the resulting red responses are subtracted from each pixel and the mean green and blue channel responses for the green and blue color channels, (iii) the resultant image is further divided by the standard deviation of the resultant image. Authors demonstrated the effectiveness of their color invariance by examining it on dataset containing nonlinear image data acquired from 6 different imaging devices.

Recently, Arandjelović [2] proposed device independent color invariances tailored for face recognition applications. The author proposed the photometric camera model of image formation as follows:

$$I_k(\mathbf{x}) = \max[\alpha \cdot \{S_k(\mathbf{x}) \cdot E(\Theta)\}^{\gamma_k}, 1.0] \quad (42)$$

where  $S_k(\mathbf{x})$  is the surface reflectance,  $E$  is the function of illuminant. Geometric and other parameters are not modelled explicitly and represented by  $\Theta \equiv \Theta(\mathbf{x})$ .



**Fig. 8** Comparison between conventional hue of (31) and logarithmic hue of (40) on sample image from [29] (for visual purpose, hue and log hue images are presented in *color* although they are 1D component)

In this formula, nonlinearities in the camera response is considered by including camera gamma parameter  $\gamma_k$ , linear camera gain  $\alpha$ , and the saturation function (for clipping). Several invariant features, namely, log- $\Delta$ -ratio, and adaptive log- $\Delta$ -ratio are derived from this model which have shown their effectiveness in face recognition application, especially where training and testing image data are acquired with different imaging devices. However, proposed invariant features are less applicable for general applications, since some underlying assumptions only hold true for frontal or nearly frontal face images.

## 5 Experiment

In this section, we evaluate the performance of representative color constancy and color invariance solutions introduced in previous sections. An important issue in performance evaluation is to select appropriate performance measures, which are closely related to the nature of target applications. For example, if color constancy or invariance methods are applied as a pre-processing operation for high-level vision applications, e.g. skin color detection [48] or object recognition [47], then the performance is typically evaluated by the overall performance of high-level system. On the other hand, if accurate color reproduction is required for consumer photography,

then algorithm that reliably estimates the scene illumination condition, and closely mimics the feature of human visual perception, is highly desirable. Another key issue is to select representative set of color images to judge the stability of algorithm over wide range of image variations. In Sects. 5.1 and 5.2, we provide examples of evaluation protocols for color constancy and color invariance solutions.

### 5.1 Evaluation of Color Constancy Solutions

Adopting general practice, the performance of color constancy solutions is measured by comparing the predefined groundtruth scene illuminant for a given input image to the estimated scene illuminant derived from algorithm. The performance of color constancy algorithms is dependent on the scene contents, and thus average performance over a large set of images should be assessed for reliable judgement of algorithm accuracy. The accuracy is commonly evaluated using a mathematical metric called *angular error*,  $d_{ang}$ , which measures the angular distance between the estimated light source  $\hat{\mathbf{e}} = [e_R^e, e_G^e, e_B^e]^T$  and the groundtruth light source  $\mathbf{e}_g = [e_R^g, e_G^g, e_B^g]^T$  as:

$$d_{ang}(\hat{\mathbf{e}}, \mathbf{e}_g) = \cos^{-1} \left( \frac{\hat{\mathbf{e}} \cdot \mathbf{e}_g}{\|\hat{\mathbf{e}}\| \cdot \|\mathbf{e}_g\|} \right) \quad (43)$$

where  $\hat{\mathbf{e}} \cdot \mathbf{e}_g$  is the dot product of the two illuminants and  $\|\cdot\|$  is the Euclidean norm of a vector. Gijssen et al. [39] performed psychophysical experiments on several image databases, and demonstrated that the angular error is a reasonably good indicator of the perceptual performance of color constancy algorithms.

To evaluate color constancy performance, we made use of the Ciurea’s grayball dataset [14], which consists of 11346 images of indoor and outdoor imaging conditions, extracted from 15 video sequences recorded by Sony VX-2000 3-CCD sensor video camera. All images in the dataset are  $360 \times 240$  in spatial resolution with 8-bit JPEG format. This dataset is particularly chosen since it is widely used dataset, allowing for the convenient validation of performance due to predefined groundtruth information (acquired by attaching a diffuse gray sphere to the camera, displayed in the bottom right corner of the image). In addition, compared to well-controlled images taken from laboratory environment, it contains real-world scenes with disturbances (i.e. image noise, and clipping due to saturation), thus more realistic assessment of performance can be achieved. Since images are extracted from video, there exist high correlations between consecutive images. Therefore, in our experiment, 1135 images of nearly no correlations were selected from the original set and used for evaluation. As discussed in Sect. 2.2, gamma corrected RGB values are linearized prior to applying color constancy solutions by assuming  $\gamma = 2.2$  (adopting the approach suggested in [41]) which roughly corresponds to sRGB format.

In this experiment, we report the angular error performance of individual algorithms using three statistics, including mean, median, and trimean. In color constancy

**Table 4** Performance of color constancy algorithms on Ciurea’s grayball dataset [14]

Color constancy Algorithm	Angular error		
	Mean	Median	Trimean
Nothing	15.66	14.06	14.56
Grayworld (GW) ( $\mathbf{e}(0, 1, 0)$ )	12.59	10.45	11.19
White Patch (WP) ( $\mathbf{e}(0, -1, 5)$ )	13.66	11.73	12.30
Shade of Gray (SoG) ( $\mathbf{e}(0, 2, 0)$ )	12.49	10.83	11.23
1st-order Gray Edge (GE1) ( $\mathbf{e}(1, 1, 3)$ ) [69]	11.17	9.63	9.95
2nd-order Gray Edge (GE2) ( $\mathbf{e}(2, 1, 2)$ ) [69]	11.23	9.67	10.04
Inverse Intensity Chromaticity (IIC) [63]	15.00	11.05	11.69
Gamut Mapping (GMP)	11.62	9.10	10.06
Edge based Gamut Mapping (GME) [40]	10.76	8.87	9.27
Natural Image Statistics (NIS) [38]	9.93	7.75	8.32

The  $(n, p, \sigma)$  parameter configurations are specified in the parenthesis for the low-level statistics based methods

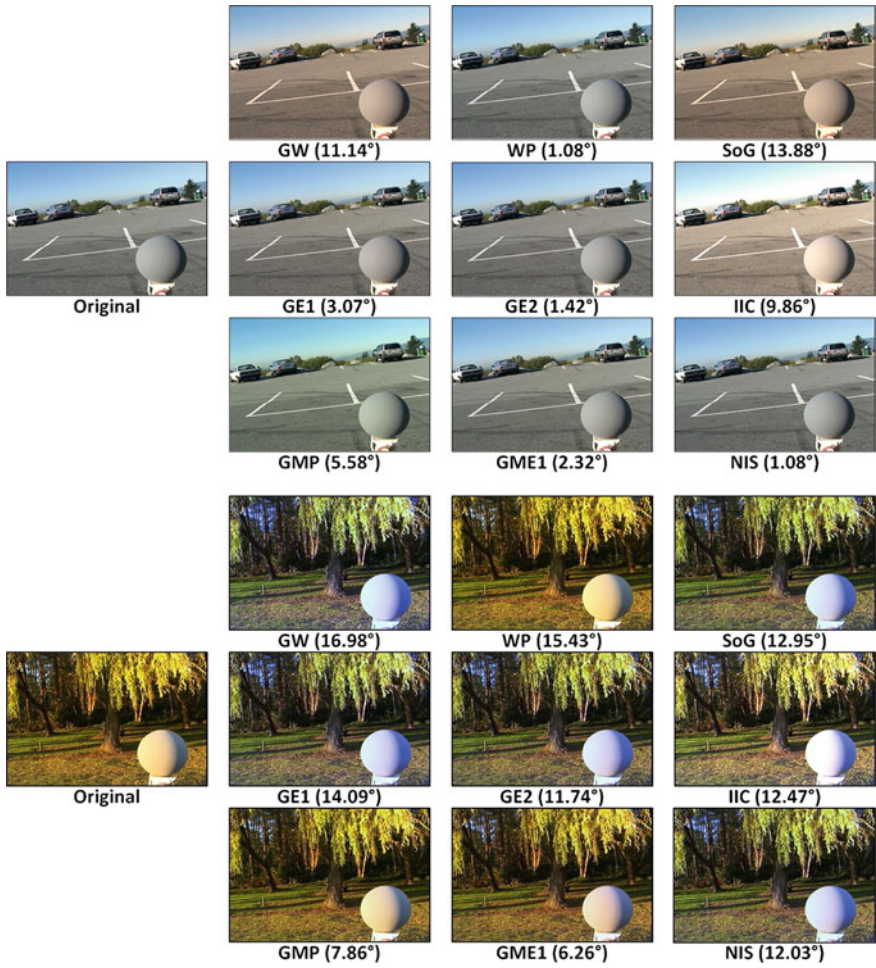
research, median angular error is considered to be more reliable than mean statistics since the error distributions tend to be significantly skewed rather than normal distribution [45]. In addition, the trimean which is the weighted average of the first, second, and third quantile  $Q_1$ ,  $Q_2$ , and  $Q_3$ :

$$Trimean = \frac{Q_1 + 2Q_2 + Q_3}{4} \quad (44)$$

is also widely used as a reliable summary statistic [39].

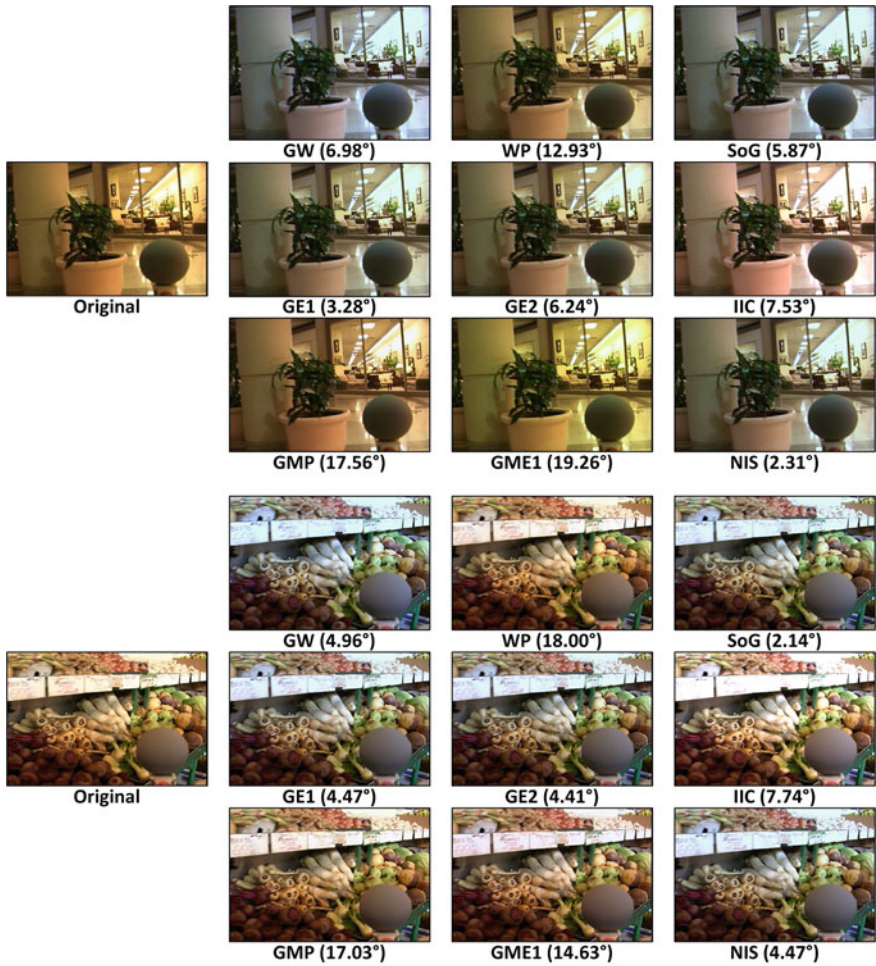
The performance of color constancy solutions are compared in Table 4 and examples of corrected images are presented in Figs. 9 and 10. Following observations are made from this experiment:

1. Learning based approaches generally outperform static approaches since incorporating prior knowledge (e.g. range of color measurements for a given illuminant, or optimal algorithm depending on scene edge response) enables more accurate estimation of scene illuminant. The best color constancy performance is achieved by fusion and selection based method (i.e. Natural Image Statistics [38]) at the cost of increased computational complexity. Instead of the angular error metric, one can also use a distance measure that is more closely correlated to the visual perception, such as the perceptual Euclidean distance (PED) [39] for comparative analysis.
2. Proper tuning of parameters for low-level statistic methods is highly important for stable performance. For example, WP algorithm tends to yield better result when local averaging scale  $\sigma > 0$  since it reduces the impact of noise during the illuminant estimation. The optimal configuration can be empirically estimated on a large set of image data.
3. All color constancy algorithms examined in this section operate under an assumption that illumination condition over the scene is constant which can be often vio-



**Fig. 9** Outdoor images from Ciurea's grayball dataset [14] corrected using various methods. Angular errors are indicated in *parenthesis*

lated in reality due to the presence of multiple illuminants. Alternatively, multiple illuminant based solutions in Sect. 3.2.3 can be applied to yield enhanced performance on real-world image dataset compared to conventional solutions relying on an uniform illumination hypothesis.



**Fig. 10** Indoor images from Ciurea’s grayball dataset [14] corrected using various methods. Angular errors are indicated in *parenthesis*

### 5.2 Evaluation of Color Invariance Solutions

Effective color invariance methods should maintain a good balance between constancy of the measurement regardless of the influence of the varying imaging condition and discriminative power between different states of the objects. In this section, we apply color histogram based object recognition to evaluate the effectiveness of the color invariance methods. Color histogram of a colored object is obtained by discretizing the image colors and counting the number of times each distinct color occurs in the image data [62].



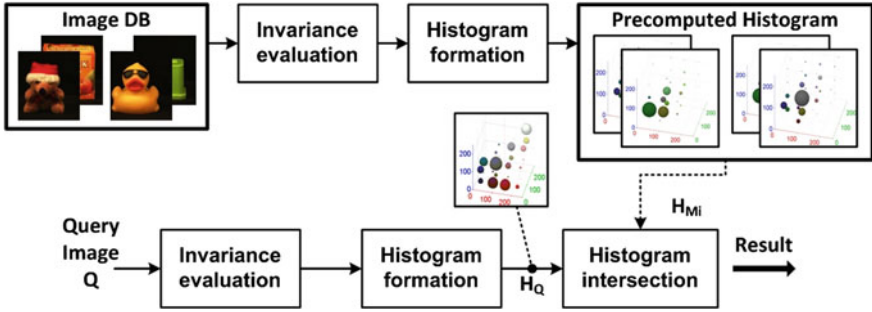


Fig. 11 Workflow of color indexing for evaluation of color invariance methods

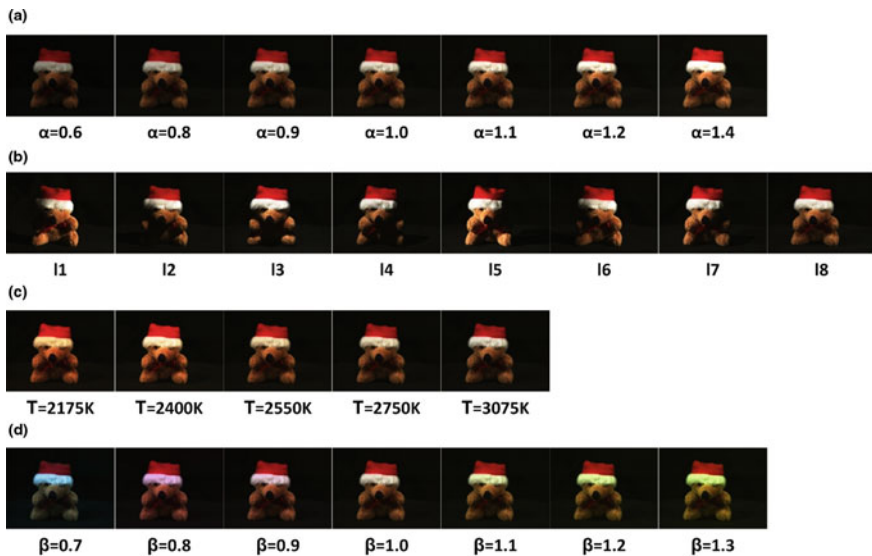
The experimental sequence is as follows (Fig. 11). Invariant features are initially computed from query image  $Q : \mathbb{R}^2 \rightarrow \mathbb{R}^3$  to obtain the color histogram  $H_Q$ . Then,  $H_Q$  is matched against the histogram computed from a set of  $N$  stored reference images  $M_i : \mathbb{R}^2 \rightarrow \mathbb{R}^3 (i = 1, \dots, N)$  in the database. Similarity between two images is measured using histogram intersection of two color distributions:

$$H_{\cap}(H_Q, H_{M_i}) = \frac{\sum_{\mathbf{c}} \min(H_Q(\mathbf{c}), H_{M_i}(\mathbf{c}))}{\sum_{\mathbf{c}} H_{M_i}(\mathbf{c})} \quad (45)$$

where  $H_{M_i}$  is the histogram of the reference image. For a query image of an object under unknown illumination conditions, if the reference image yielding the highest histogram intersection measure is equal to the original image, then it is regarded as successful recognition.

The histogram intersection measure are computed on the basis of different color invariant features in  $n$ -dimensional space, including RGB ( $n = 3$ ), normalized RGB ( $n = 3$ ),  $C_1C_2C_3$  ( $n = 3$ ),  $m_1m_2m_3$  ( $n = 3$ ),  $HS$  (hue and saturation,  $n = 2$ ),  $O_1O_2$  ( $n = 2$ ), and hue ( $n = 1$ ). For each color component, the range of pixel values is partitioned uniformly in fixed intervals. Following suggestions from literatures [35], the histogram bin size is set to 32. We used the Amsterdam Library of Object Images (ALOI) dataset [34], containing images of 1000 objects recorded under various imaging circumstances, to evaluate the object recognition performance. This dataset is chosen since: (i) it contains images of objects recorded under controlled laboratory setup with varying illumination angle, and illumination color for each object, facilitating a systematic evaluation of features, (ii) it is a publicly available dataset. In our experiment, we made use of randomly selected 100 objects from entire dataset.

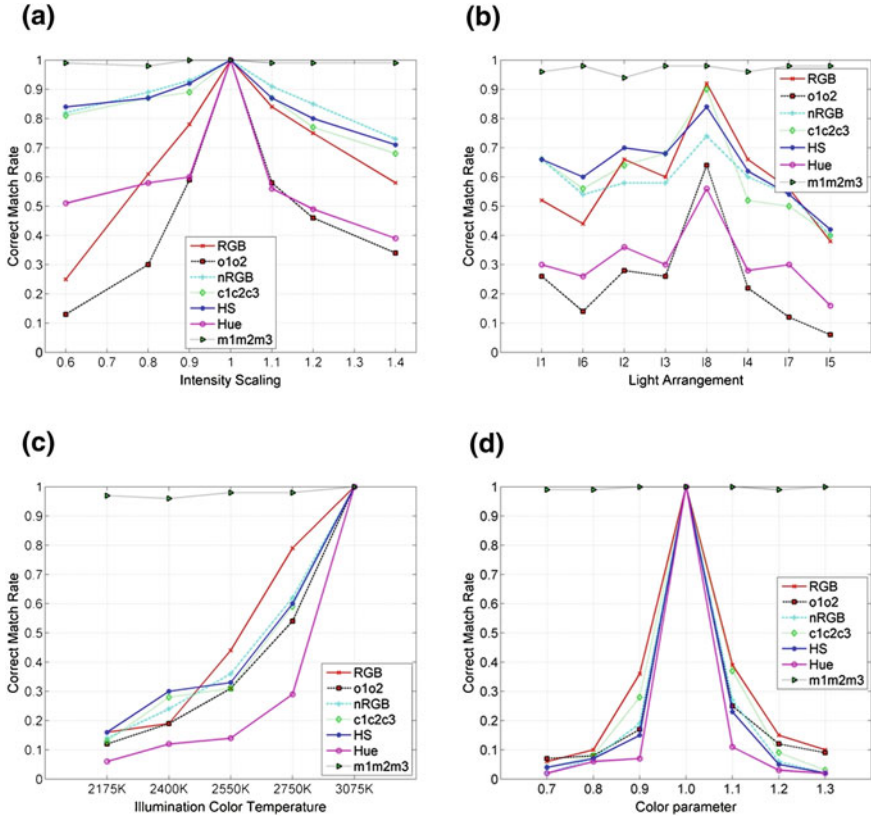
The performance of color invariant features are evaluated under changing illumination intensity, illumination direction, and illumination color. Illumination intensity variation is not provided in the original set; hence, we have artificially generated intensity sets by multiplying a constant factor  $\alpha$  to RGB values of the reference image (i.e.  $[R_{out}, G_{out}, B_{out}] = [\alpha R_{in}, \alpha G_{in}, \alpha B_{in}]$ ). Illumination direction variation set



**Fig. 12** Sample images [34] used in the evaluation of color invariance methods. **a** Illumination intensity variation (generated). **b** Illumination direction variation (original set). **c** Illumination color variation (original set). **d** Illumination color variation (generated)

is provided in the original set and directly used without modification. Illumination color set in ALOI is recorded with varying color temperature of light source in laboratory setup from 2175 to 3075 K, resulting in objects illuminated under a reddish to white illumination color. From the visual inspection, we noticed that given illumination color set does not cover wide range of colors, and therefore, we generated additional color sets covering bluish and greenish illumination. They are generated by multiplying a factor of  $1, \beta, 2 - \beta$  to RGB channels independently (i.e.  $[R_{out}, G_{out}, B_{out}] = [R_{in}, \beta G_{in}, (2 - \beta)B_{in}]$ ), where  $\beta > 1$  simulates the condition when the object is lit by greenish light, whereas  $\beta < 1$  simulates the condition when the object is lit by bluish light. The example of variation set is demonstrated in Fig. 12.

In Fig. 13, the evaluation results are summarized for different illumination conditions. For light intensity change, it is shown that descriptors such as normalized RGB, HS,  $C_1C_2C_3$ , and  $m_1m_2m_3$  outperform other descriptors with clear distinction. RGB and  $O_1O_2$  degrades significantly when scaling factor  $\alpha$  deviates further from 1, demonstrating their instability with respect to substantial light intensity change. It is worthwhile to mention that use of saturation component in conjunction with hue components greatly improve the recognition rate compared to the case when 1D hue histogram is used since saturation component provides additional discriminative power. Variation of illumination direction leads to shadows and partial invisibility on the object as shown in Fig. 12. Therefore, under varying illumination direction condition, not only invariant nature of color feature, but also discriminative power is



**Fig. 13** Evaluation of color invariance methods under different illumination condition. **a** Illumination intensity. **b** Illumination direction. **c** Illumination color. **d** Illumination color

considered to be important [59]. Experimental result indicates that  $m_1m_2m_3$  provides robust recognition performance, whereas  $O_1O_2$  and hue yield limited performance over illumination direction. For illumination color changes, it can be seen that most of invariant features are unstable even to moderate degree of variations. The exception is  $m_1m_2m_3$ , which maintained high recognition rate regardless of degree of variation. This outcome is consistent with the theory that invariant characteristic of other features such as normalized RGB, HS,  $C_1C_2C_3$  are valid only if object is rendered under white illumination (See Table 3).

## 6 Conclusion

Color is an essential discriminative property of objects in computer vision and digital image processing applications. Not only color is a cost-effective cue which can be processed regardless of the spatial context, but also it provides robust features against

geometric transformation such as scaling, rotation, and partial occlusion. However, RGB values recorded in image data are not only a function of object reflectance but also a function of acquisition device and illumination condition. When these factors are not properly controlled, or not take into consideration, the performance of color imaging task can deteriorate substantially. To alleviate this problem, one can pre-process images to remove bias due to illumination using color constancy approaches, or represent image with color invariant descriptors that remain unchanged over illumination and device variations. In this chapter, the underlying theory and application of computational color constancy and color invariance solutions are reviewed. Following conclusions can be drawn from this chapter:

1. In order to render the acquired image data as close as to human visual perception, the first stage of the computational color constancy is to estimate the scene illuminant. Then its effect is typically compensated based on the diagonal model. Since the only information available are RGB sensor values of the image, color constancy is an under-constrained problem, and hypotheses about statistical properties of the feasible scene illuminants and surface reflectances are generally imposed to make problem solvable. In general, existing solutions can be categorized into two classes. Static approaches estimate the illumination condition solely based on the content in a single image with assumptions about the general characteristic of color images, whereas learning-based approaches require training data to construct a statistical model prior to prediction of illumination. Most existing color constancy solutions rely on an assumption that the illumination condition is uniform over the scene. They can be further extended to deal with more realistic scenes where there are multiple light sources present in an image.
2. The interaction between light sources and objects in the scene can be described by a physical model of image formation, which explains how illumination changes influence the RGB values in image data. In the basis of this model, color features have been proposed which are independent to variation in illumination condition (e.g. illumination intensity, color, direction, and so forth). In many cases, color dependency due to device variations has been neglected and devices are often assumed to produce linear color signal. However, generally, images are stored in nonlinear format to reverse the gamma of image display devices, where this process is often achieved by applying a power function (i.e.  $(R, G, B) \rightarrow (R^{1/\gamma}, G^{1/\gamma}, B^{1/\gamma})$  where  $\gamma$  represents the gamma of the monitor). Problem arises when one deal with images acquired from different imaging devices, since they may use different  $\gamma$  values. In such case, device independent color invariance methods can be effectively used to remove color dependency due to variation in gamma factor.

## References

1. Agarwal V, Abidi BR, Koschan A, Abidi MA (2006) An overview of color constancy algorithms. *J Pattern Recognit Res* 1(1):42–54

2. Arandjelović O (2012) Colour invariants under a non-linear photometric camera model and their application to face recognition from video. *Pattern Recognit* 45(7):2499–2509
3. Barnard K (2000) Improvements to gamut mapping colour constancy algorithms. In: *Proceedings of the European conference on computer vision-Part I*. Springer, London, pp 390–403
4. Barnard K, Cardei V, Funt B (2002a) A comparison of computational color constancy algorithms. I: Methodology and experiments with synthesized data. *IEEE Trans Image Process* 11(9):972–984
5. Barnard K, Martin L, Funt B, Coath A (2002b) A data set for color research. *Color Res Appl* 27(3):147–151
6. Bianco S, Schettini R (2012) Color constancy using faces. *IEEE Conf. on Computer Vision and Pattern Recognition*, In, pp 65–72
7. Bianco S, Ciocca G, Cusano C, Schettini R (2008a) Improving color constancy using indoor-outdoor image classification. *IEEE Trans Image Process* 17(12):2381–2392
8. Bianco S, Gasparini F, Schettini R (2008b) Consensus-based framework for illuminant chromaticity estimation. *J Electron Imaging* 17(2):1–9
9. Bianco S, Ciocca G, Cusano C, Schettini R (2010) Automatic color constancy algorithm selection and combination. *Pattern Recognit* 43(3):695–705
10. Bleier M, Riess C, Beigpour S, Eibenberger E, Angelopoulou E, Troger T, Kaup A (2011) Color constancy and non-uniform illumination: can existing algorithms work? In: *IEEE international conference on computer vision workshops*, pp 774–781
11. Brainard DH, Freeman WT (1997) Bayesian color constancy. *J Opt Soc Am A* 14(7):1393–1411
12. Buchsbaum G (1980) A spatial processor model for object colour perception. *J Franklin Inst* 310(1):1–26
13. Cardei VC, Funt BV (1999) Committee-based color constancy. In: *Color imaging conference*, pp 311–313
14. Ciurea F, Funt BV (2003) A large image database for color constancy research. In: *Color imaging conference*, pp 160–164
15. Delahunt PB, Brainard DH (2004) Does human color constancy incorporate the statistical regularity of natural daylight. *J Vis* 4(2):57–81
16. Drew M, Joze H, Finlayson G (2012) Specularity, the Zeta-image, and information-theoretic illuminant estimation. In: *European conference on computer vision. Workshops and demonstrations*, vol 7584, pp 411–420
17. Ebner M (2007) *Color constancy*. Wiley-IS&T series in imaging science and technology. Wiley, West Sussex
18. Ebner M (2009) Color constancy based on local space average color. *Mach Vision Appl* 20(5):283–301
19. Finlayson G (1996) Color in perspective. *IEEE Trans Pattern Anal Mach Intell* 18(10):1034–1038
20. Finlayson G, Xu R (2003a) Illuminant and gamma comprehensive normalisation in log RGB space. *Pattern Recogn Lett* 24(11):1679–1690
21. Finlayson G, Hordley S, Xu R (2005) Convex programming colour constancy with a diagonal-offset model. In: *IEEE international conference on image processing*, vol 3, pp III - 948–951
22. Finlayson GD, Hordley SD (2000) Improving gamut mapping color constancy. *IEEE Trans Image Process* 9(10):1774–1783
23. Finlayson GD, Schaefer G (2001a) Hue that is invariant to brightness and gamma. In: *BMVC*
24. Finlayson GD, Schaefer G (2001b) Solving for colour constancy using a constrained dichromatic reflection model. *Int J Comput Vision* 42(3):127–144
25. Finlayson GD, Trezzi E (2004) Shades of Gray and Colour Constancy. In: *Twelfth color imaging conference: color science and engineering systems, technologies, and applications*, pp 37–41
26. Finlayson GD, Xu R (2003b) Convex programming color constancy. In: *IEEE workshop on color and photometric methods in computer vision*
27. Finlayson GD, Hordley SD, Hubel PM (2001) Color by correlation: A simple, unifying framework for color constancy. *IEEE Trans Pattern Anal Mach Intell* 23(11):1209–1221
28. Forsyth DA (1990) A novel algorithm for color constancy. *Int J Comput Vision* 5(1):5–36

29. Foster DH, Amano K, Nascimento SMC, Foster MJ (2006) Frequency of metamerism in natural scenes. *J Opt Soc Am A* 23(10):2359–2372
30. Funt B, Shi L (2010a) The effect of exposure on MaxRGB color constancy. In: Proceedings of the SPIE, vol 7527, pp 75,270Y–1–7
31. Funt B, Shi L (2010b) The rehabilitation of MaxRGB. In: Proceedings of IS&T color imaging conference, pp 256G–259G
32. Gehler P, Rother C, Blake A, Minka T, Sharp T (2008) Bayesian color constancy revisited. In: IEEE conference on computer vision and pattern recognition, pp 1–8
33. Geusebroek JM, Smeulders AWM (2005) A six-stimulus theory for stochastic texture. *Int J Comput Vision* 62(1–2):7–16
34. Geusebroek JM, Burghouts GJ, Smeulders AWM (2005) The Amsterdam library of object images. *Int J Comput Vision* 61(1):103–112
35. Gevers T, Smeulders AWM (1999) Color-based object recognition. *Pattern Recognit* 32(3):453–464
36. Gevers T, Stokman H (2003) Classifying color edges in video into shadow-geometry, highlight, or material transitions. *IEEE Trans Multimedia* 5(2):237–243
37. Gijsenij A, Gevers T (2007) Color constancy by local averaging. In: International conference on image analysis and processing workshops, pp 171–174
38. Gijsenij A, Gevers T (2011) Color constancy using natural image statistics and scene semantics. *IEEE Trans Pattern Anal Mach Intell* 33(4):687–698
39. Gijsenij A, Gevers T, Lucassen M (2009) A perceptual analysis of distance measures for color constancy algorithms. *J Opt Soc Am A* 26(10):2243–2256
40. Gijsenij A, Gevers T, van de Weijer J (2010) Generalized gamut mapping using image derivative structures for color constancy. *Int J Comput Vision* 86(2–3):127–139
41. Gijsenij A, Gevers T, van de Weijer J (2011) Computational color constancy: survey and experiments. *IEEE Trans Image Process* 20(9):2475–2489
42. Gijsenij A, Gevers T, van de Weijer J (2012a) Improving color constancy by photometric edge weighting. *IEEE Trans Pattern Anal Mach Intell* 34(5):918–929
43. Gijsenij A, Lu R, Gevers T (2012b) Color constancy for multiple light sources. *IEEE Trans Image Process* 21(2):697–707
44. Hordley SD (2006) Scene illuminant estimation: past, present, and future. *Color Res Appl* 31(4):303–314
45. Hordley SD, Finlayson GD (2006) Reevaluation of color constancy algorithm performance. *J Opt Soc Am A* 23(5):1008–1020
46. Kakumanu P, Makrogiannis S, Bourbakis N (2007) A survey of skin-color modeling and detection methods. *Pattern Recognit* 40(3):1106–1122
47. Kanan C, Flores A, Cottrell GW (2010) Color constancy algorithms for object and face recognition. In: Proceedings of the international conference on advances in visual computing—Volume Part I. Springer, Berlin, pp 199–210
48. Khan R, Hanbury A, Stttinger J, Bais A, (2012) Color based skin classification. *Pattern Recogn Lett* 33(2):157–163
49. Land EH (1977) The Retinex theory of color vision. *Scientific Am* 237(6):108–128
50. Lee HC (1986) Method for computing the scene-illuminant chromaticity from specular highlights. *J Opt Soc Am A* 3(10):1694–1699
51. Lee HC, Breneman E, Schulte C (1990) Modeling light reflection for computer color vision. *IEEE Trans Pattern Anal Mach Intell* 12(4):402–409
52. Li B, Xu D, Lang C (2009) Colour constancy based on texture similarity for natural images. *Coloration Technol* 125(6):328–333
53. MacAdam D (1970) Sources of color science. MIT Press, Cambridge
54. Mallick S, Zickler T, Kriegman D, Belhumeur P (2005) Beyond lambert: reconstructing specular surfaces using color. In: IEEE conference on computer vision and pattern recognition, vol 2, pp 619–626
55. Perez F, Koch C (1994) Toward color image segmentation in analog vlsi: algorithm and hardware. *Int J Comput Vision* 12:17–42

56. Poynton C (2003) Digital video and HDTV: algorithms and interfaces. Morgan Kaufmann series in computer graphics and geometric Mo. Morgan Kaufmann, San Francisco
57. Riess C, Eibenberger E, Angelopoulou E (2011) Illuminant color estimation for real-world mixed-illuminant scenes. In: IEEE international conference on computer vision workshops, pp 782–789
58. Rosenberg CR, Minka TP, Ladsariya A (2003) Bayesian color constancy with Non-Gaussian models. In: Neural information processing systems. MIT Press, Cambridge
59. van de Sande K, Gevers T, Snoek C (2010) Evaluating color descriptors for object and scene recognition. IEEE Trans Pattern Anal Mach Intell 32(9):1582–1596
60. Schaefer G, Hordley S, Finlayson G (2005) A combined physical and statistical approach to colour constancy. In: IEEE conference on computer vision and pattern recognition, vol 1, pp 148–153
61. Shafer SA (1992) Using color to separate reflection components. In: Healey GE, Shafer SA, Wolff LB (eds) Color. Jones and Bartlett Publishers, Inc., Boston, pp 43–51
62. Swain MJ, Ballard DH (1991) Color indexing. Int J Comput Vision 7(1):11–32
63. Tan RT, Nishino K, Ikeuchi K (2004) Color constancy through inverse-intensity chromaticity space. J Opt Soc Am A 21(3):321–334
64. Tominaga S, Wandell BA (1989) Standard surface-reflectance model and illuminant estimation. J Opt Soc Am A 6(4):576–584
65. Vazquez-Corral J, Vanrell M, Baldrich R, Tous F (2012) Color constancy by category correlation. IEEE Trans Image Process 21(4):1997–2007
66. Viola PA, Jones MJ (2004) Robust real-time face detection. Int J Comput Vision 57(2):137–154
67. van de Weijer J, Gevers T, Geusebroek JM (2005) Edge and corner detection by photometric quasi-invariants. IEEE Trans Pattern Anal Mach Intell 27(4):625–630
68. van de Weijer J, Gevers T, Bagdanov A (2006) Boosting color saliency in image feature detection. IEEE Trans Pattern Anal Mach Intell 28(1):150–156
69. van de Weijer J, Gevers T, Gijssenij A (2007a) Edge-based color constancy. IEEE Trans Image Process 16(9):2207–2214
70. van de Weijer J, Schmid C, Verbeek J (2007b) Using high-level visual information for color constancy. In: IEEE international conference on computer vision, pp 1–8
71. Yendrikhovskij SN, Blommaert FJJ, de Ridder H (1999) Color reproduction and the naturalness constraint. Color Res Appl 24(1):52–67
72. Zickler T, Mallick SP, Kriegman DJ, Belhumeur PN (2008) Color subspaces as photometric invariants. Int J Comput Vision 79(1):13–30

# On the von Kries Model: Estimation, Dependence on Light and Device, and Applications

Michela Lecca

**Abstract** The von Kries model is widely employed to describe the color variation between two pictures portraying the same scene but captured under two different lights. Simple but effective, this model has been proved to be a good approximation of such a color variation and it underpins several color constancy algorithms. Here we present three recent research results: an efficient histogram-based method to estimate the parameters of the von Kries model, and two theoretical advances, that clarify the dependency of these parameters on the physical cues of the varied lights and on the photometric properties of the camera used for the acquisition. We illustrate many applications of these results: color correction, illuminant invariant image retrieval, estimation of color temperature and intensity of a light, and photometric characterization of a device. We also include a wide set of experiments carried out on public datasets, in order to allow the reproducibility and the verification of the results, and to enable further comparisons with other approaches.

**Keywords** Color and light · von Kries model · Estimation of the von Kries coefficients · Dependence of the von Kries model on light and device · Planck's and Wien's lights · Color correction · Illuminant invariant image retrieval · Intensity and Color temperature of a light · Device photometric characterization

## 1 Introduction

Color is one of the most important features in many Computer Vision fields such as image retrieval and indexing [48], object and scene recognition [53], image segmentation [49], and object tracking [39]. Although color is robust to many image

---

An erratum to this chapter is available at [10.1007/978-94-007-7584-8\\_14](https://doi.org/10.1007/978-94-007-7584-8_14)

---

M. Lecca (✉)

Fondazione Bruno Kessler, via Sommarive 18, 38123 Trento, Italy  
e-mail: [lecca@fbk.eu](mailto:lecca@fbk.eu)



geometric distortions, e.g. changes of image size and/or orientation, and to noise, its use in practical applications is often limited by its strong sensitivity to the light. In fact, we experience that the same scene viewed under two different lights produces two different pictures. This is because the color of an image depends on the spectral profile of the light illuminating the scene, on the spectral reflectivity and geometry of the materials composing the scene, and on the device used for the acquisition. Features like colors, luminance, magnitude and orientation of the edges, which are commonly used to describe the visual appearance of a scene, remarkably change when the light varies. As a consequence, many algorithms for image classification and/or object recognition, that are based on these features (e.g. [38, 40]), do not work in case of illuminant variations [53]. Understanding how the colors of a picture vary across the illumination of the imaged scene is a crucial task to develop a recognition and/or retrieval system insensitive to light conditions.

In the human visual system, the color illuminant invariance is achieved by a chromatic adaptation mechanism named *color constancy*: it detects and removes possible chromatic dominants and illuminant incidents from the observed scene, so that the same scene under different illuminants is perceived as the same entity [54]. Although color constancy has been intensively investigated in the past decades, it remains still an unsolved problem [10, 24]. In the last years, many methods for simulating this human capability have been developed [2, 21–23, 25, 29, 33]. A recent survey on the main approaches is presented in [28], while a comparison of the most used color constancy algorithms can be found in [5, 6]. Advantages and disadvantages in using some of these methods are addressed in [1].

The *von Kries model* is widely used to describe the color variation between images or image regions due to an illuminant change. It relies on three main assumptions: (i) the lights under which the images (or regions) are acquired are spatially uniform across the scene; (ii) the materials composing the imaged scene are Lambertian; (iii) the device used for capturing the images is narrow-band, or the spectral sensitivities of its sensors do not overlap. Hypothesis (i) constraints the spectral power distribution of the light to be homogeneous across the scene. Hypothesis (ii) holds for a lot of wide matte materials, which appear equally bright from all viewing directions. Finally, hypothesis (iii) is generally satisfied by the most cameras. Otherwise, the camera sensitivities can be sharpened by a linear transform [7, 19], in order to fulfill the requirements of (iii). Under these conditions, the von Kries model approximates the color change between the input images with a linear diagonal map, that rescales independently the color responses of the two pictures. Usually, the von Kries map is defined over the RGB color space, where the parameters of this model, called the *von Kries coefficients*, are the three scale factors that interrelate the red, green and blue responses of the input images. Despite its simplicity, the von Kries model has been proved to be a successful approximation of the illuminant change [17, 18] and it underpins many color constancy and color correction algorithms, e.g. [4, 9, 23, 29, 34, 35]. This success justifies the attention we dedicate to the von Kries model in this chapter, where we present three research studies, which have been dealt with in [35–37].

The work in [35] proposes a novel efficient method to estimate the parameters of the von Kries map possibly relating two images or two image regions. The estimation of the von Kries coefficients is carried out by the most popular color constancy algorithms, e.g. [11, 14, 20, 23, 33] in two steps: first, they compute the illuminants under which the input images or regions have been captured and codify them as 3D vectors; second, they estimate the von Kries coefficients as the ratios between the components of the illuminants. The approach in [35] strongly differs from these methods because it does not require any estimation or knowledge of the illuminants. It namely defines a von Kries model based measure of dissimilarity between the color distributions of the inputs, and derives the von Kries coefficients by minimizing this dissimilarity. The technique in [35] is invariant to many image distortions, like image rescaling, in-plane rotation, translation and skew, and it shows impressive performances in terms of computational charge, execution time, and accuracy on color correction, also in comparison with other methods, as shown from the experiments reported here.

The works in [36, 37] present some theoretical advances, that clarify the relationships of the von Kries parameters with the physical cues of the light (i.e. color temperature and intensity) and with the photometric properties of the device used for the image acquisition (i.e. the camera spectral sensitivities). To the best of our knowledge, the mathematical results discussed in these papers have been never proposed before.

The paper in [37] presents an empirical analysis of pictures or regions depicting the same scene or object under different Planck's lights, i.e. lights behaving like a black-body radiator. The experiments carried out in [37] lead to two main results. First, a color change produced by varying two Planck's lights is well approximated by a von Kries map. Such a result is not surprising, as it is in line with [18]. Second, the coefficients of the von Kries approximation are not independent to each other: they namely form 3D points belonging to a ruled surface, named *von Kries surface*, and parametrized by the color temperatures and intensities of the varied lights. The mathematical equation of the von Kries surface puts on evidence the relationship between the von Kries coefficients and the physical cues of the varied lights, and it is used in [37] to estimate the color temperature and intensity of an illuminant, given a set of images captured under that illuminants and a von Kries surface. Since in [37] there are no geometric distortions between the differently illuminated images (or regions) considered in the experiments, estimation of the von Kries approximation is performed by a best-fit technique, that computes the von Kries coefficients as the slope of the *best* line interpolating the pairs of pixel-wise correspondent color responses.

The work in [36] deals with changes of Wien's illuminants, that are a special case of Planck's lights. As in [37], and according to [18], an empirical study shows that the von Kries model is a valid approximation for the color changes due to a variation of Wien's lights. Then the authors derive a mathematical equation interrelating the von Kries coefficients and the spectral sensitivities of the acquisition device. This equation reveals a direct proportionality between two 2D vectors: the first one is defined by the von Kries coefficients, while the second one by the wavelengths at which the RGB sensors of the device is maximally sensitive. This relationship allows to discover the relationship of the von Kries parameters with the photometric

**Table 1** Chapter outline

Topic	Short description	Sections
Generalities	Introduction;	1
	Derivation of the linear model for color variation;	2
	Derivation of the von Kries model	
Result 1	Estimation of the von Kries model	3
	with Application to Color Correction and Illuminant invariant image retrieval	4
Result 2	Dependency of the von Kries model on the physical cues of the illuminants; von Kries surfaces;	5
	Application to estimation of Color temperature and Intensity of an Illuminant.	
	Dependency of the von Kries model on the photometric properties of the acquisition device;	
Result 3	Applications to Device characterization and to Illuminant invariant image representation	6
Conclusions	Final remarks	7

properties of the acquiring camera, and thus to prime information about the sensor sensitivities from the von Kries maps relating pairs of images linked by a variation of Wien's illuminants. In the experiments reported in [36], estimation of the von Kries approximation was done by the approach of [35].

We notice that the hypotheses of Planck's and Wien's lights of the works [36, 37] do not compromise the generality of the results, because the most illuminants satisfy Planck's or Wien's law.

In the description of the works [35–37] provided in this chapter, we added more experiments and more comparisons with other approaches than those reported in the papers mentioned above. In particular, the method of [35] has been tested on three additional databases [8, 16, 45] and it has been compared with other techniques, listed in [13]. The experiments carried out in [37] have been repeated by estimating the von Kries approximation through the method [35], and a new measure for the accuracy on the von Kries approximation has been introduced (the Hilbert–Schmidt inner product of Sect. 5.2).

The overall organization of the Chapter is reported in Table 1.

## 2 Linear Color Changes

In the RGB color space, the response of a camera to the light reflected from a point  $x$  in a scene is coded in a triplet  $\mathbf{p}(x) = (p_0(x), p_1(x), p_2(x))$ , where

$$p_i(x) = \int_{\Omega} E(\lambda)S(\lambda, x)F_i(\lambda) d\lambda \quad i = 0, 1, 2. \quad (1)$$

In Eq. (1),  $\lambda$  is the wavelength of the light illuminating the scene,  $E$  its spectral power distribution,  $S$  the reflectance distribution function of the illuminated surface containing  $x$ , and  $F_i$  is the  $i$ -th spectral sensitivity function of the sensor. The integral ranges over the visible spectrum, i.e.  $\Omega = 4$  [380, 780] nm. The values of  $p_0(x)$ ,  $p_1(x)$ ,  $p_2(x)$  are the red, green and blue color responses of the camera sensors at point  $x$ .

For a wide range of matte surfaces, which appear equally bright from all viewing directions, the reflectance distribution function is well approximated by the Lambertian photometric reflection model [44]. In this case, the surface reflectance can be expressed by a linear combination of three basis functions  $S^k(\lambda)$  with weights  $\sigma_k(x)$ ,  $k = 0, 1, 2$ , so that Eq. (1) can be re-written as follows [42]:

$$\mathbf{p}(x)^T = W\sigma(x)^T \quad (2)$$

where  $\sigma(x) = (\sigma_0(x), \sigma_1(x), \sigma_2(x))$ , the superscript  $T$  indicates the transpose of the previous vector, and  $W$  is the  $3 \times 3$  matrix with entry

$$W_{ki} = \int_{\Omega} E(\lambda)S^k(\lambda)F_i(\lambda)d\lambda, \quad k, i = 0, 1, 2.$$

The response  $\mathbf{p}'(x) = (p'_0(x), p'_1(x), p'_2(x))$  captured under an illuminant with spectral power  $E'$  is then given by  $\mathbf{p}'(x)^T = W'\sigma(x)^T$ . Since the  $\sigma(x)$ 's do not depend on the illumination, the responses  $\mathbf{p}(x)$  and  $\mathbf{p}'(x)$  are related by the linear transform

$$\mathbf{p}(x)^T = W[W']^{-1}\mathbf{p}'(x)^T. \quad (3)$$

Here we assume that  $W'$  is not singular, so that Eq. (3) makes sense. In the following we indicate the  $ij$ -th element of  $W[W']^{-1}$  by  $\alpha_{ij}$ .

### 3 The von Kries Model

The *von Kries* (or *diagonal*) *model* approximates the color change in Eq. (3) by a linear diagonal map, that rescales independently the color channels by real strictly positive factors, named *von Kries coefficients*.

Despite its simplicity, the von Kries model has been proved to approximate well a color changes due to an illuminant variation [15, 17, 18], especially for narrow-band sensors and for cameras with non-overlapping spectral sensitivities. Moreover, when the device does not satisfy these requirements, its spectral sensitivities can be *sharpened* by a linear transform [7, 19], so that the von Kries model still holds.

In the following, we derive the von Kries approximation for a narrow-band camera (Sect. 3.1) and for a device with non-overlapping spectral sensitivities (Sect. 3.2). In addition, we discuss a case in which the von Kries model can approximate also a color change due to a device changing (Sect. 3.3).

### 3.1 Narrow-Band Sensors

The spectral sensitivity functions of a narrow-band camera can be approximated by the Dirac delta, i.e. for each  $i = 0, 1, 2$ ,  $F_i(\lambda) = f_i \delta(\lambda - \lambda_i)$ , where  $f_i$  is a strictly positive real number and  $\lambda_i$  is the wavelength at which the sensor maximally responds.

Under this assumption, from Eq. (1), for each  $i = 0, 1, 2$  we have

$$p_i(x) = E(\lambda_i)S(\lambda_i, x)F(\lambda_i) \quad \text{and} \quad p'_i(x) = E'(\lambda_i)S(\lambda_i, x)F(\lambda_i)$$

and thus

$$p_i(x) = \frac{E(\lambda_i)}{E'(\lambda_i)} p'_i(x) \quad \forall i = 0, 1, 2. \quad (4)$$

This means that the change of illuminant mapping  $\mathbf{p}(x)$  onto  $\mathbf{p}'(x)$  is a linear diagonal transform that rescales each channel independently. The von Kries coefficients are the rescaling factors  $\alpha_i$ , i.e. the non null elements  $\alpha_{ii}$  of  $W[W']^{-1}$ :

$$\alpha_i := \alpha_{ii} = \frac{E(\lambda_i)}{E'(\lambda_i)} \quad \forall i = 0, 1, 2. \quad (5)$$

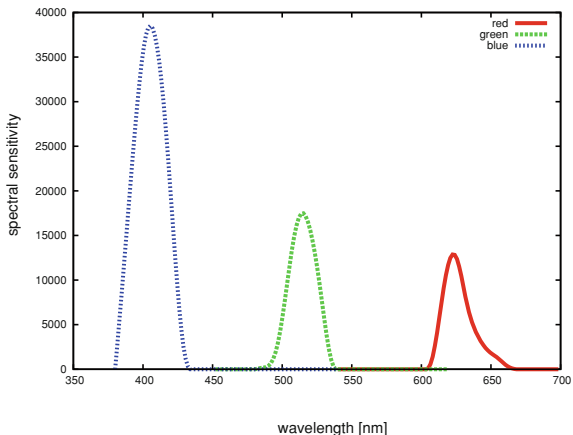
### 3.2 Non Overlapping Sensitivity Functions

Let  $I$  and  $I'$  be two pictures of a same scene imaged under different light conditions. Since the content of  $I$  and  $I'$  is the same, we assume a scene-independent illumination model [52] such that

$$E(\lambda)F_k(\lambda) = \sum_{j=0}^2 \alpha_{kj} E'(\lambda)F_j(\lambda). \quad (6)$$

Now, let us suppose that the device used for the image acquisition has *non overlapping sensitivity functions*. This means that for each  $i, j$  with  $i \neq j$ ,  $F_i(\lambda)F_j(\lambda) = 0$  for any  $\lambda$ . Generally, the spectral sensitivities are real-valued positive functions with a compact support in  $\Omega$  (see Fig. 1 for an example). Therefore non-overlapping sensitivities have non intersecting supports. We prove that under this assumption, the von Kries model still holds, i.e. matrix  $W[W']^{-1}$  is diagonal.

**Fig. 1** BARNARD2002: spectral sensitivities for the camera Sony DCX-930 used for the image acquisition of the database [8]



From Eq. (6) we have that

$$\int_{\Omega} E(\lambda)S(\lambda, x)F_k(\lambda) d\lambda = \sum_{j=0}^2 \alpha_{kj} \int_{\Omega} E'(\lambda)S(\lambda, x)F_j(\lambda) d\lambda. \quad (7)$$

i.e. the linear dependency between the responses of a camera under different illuminants is still described by Eq. (3). From Eq. (7) we have that

$$[E(\lambda)F_k(\lambda) - \sum_{j=0}^2 \alpha_{kj} E'(\lambda)F_j(\lambda)]^2 = 0. \quad (8)$$

By minimizing (8) with respect to  $\alpha_{kj}$  and by using the sensitivity non-overlap hypothesis we get the von Kries model. In fact, suppose that  $k = 0$ . The derivative of the Eq. (8) with respect to  $\alpha_{00}$  is

$$0 = -E'(\lambda)F_0(\lambda)[E(\lambda)F_0(\lambda) - \sum_{j=0}^2 \alpha_{0j} E'(\lambda)F_j(\lambda)].$$

Thanks to the non-overlapping hypothesis, and by supposing that  $E'(\lambda) \neq 0$  for each  $\lambda$  in the support of  $F_0$ , we have that

$$E(\lambda)F_0(\lambda) - \alpha_{00}E'(\lambda)F_0(\lambda) = 0. \quad (9)$$

By integrating Eq. (9) with respect to  $\lambda$  over  $\Omega$ , and by solving with respect to  $\alpha_{00}$ , we have that

$$\alpha_{00} = \frac{\int_{\Omega} E(\lambda) F_0(\lambda) d\lambda}{\int_{\Omega} E'(\lambda) F_0(\lambda) d\lambda}. \quad (10)$$

Since  $E$ ,  $E'$  and  $F_0$  are not identically null,  $\alpha_{00}$  is well defined add  $\alpha_{00} \neq 0$ . Now, we prove that  $\alpha_{0j} = 0$  for any  $j \neq 0$ . From Eq. (9) we have that

$$E(\lambda)F_0(\lambda) = \alpha_{00}E'(\lambda)F_0(\lambda).$$

Putting this expression of  $E(\lambda)F_0(\lambda)$  into Eq. (8) with  $k = 0$ , yields

$$0 = [\alpha_{01}E'(\lambda)F_1(\lambda)]^2 + [\alpha_{02}E'(\lambda)F_2(\lambda)]^2 + 2\alpha_{01}\alpha_{02}E'(\lambda)^2F_1(\lambda)F_2(\lambda).$$

Since the functions  $F_1$  and  $F_2$  do not overlap, the last term at left is null, and

$$[\alpha_{01}E'(\lambda)F_1(\lambda)]^2 + [\alpha_{02}E'(\lambda)F_2(\lambda)]^2 = 0.$$

By integrating this equation over  $\lambda$  we have that

$$\alpha_{01}^2 \int_{\Omega} [E'(\lambda)F_1(\lambda)]^2 d\lambda + \alpha_{02}^2 \int_{\Omega} [E'(\lambda)F_2(\lambda)]^2 d\lambda = 0, \quad (11)$$

and since  $E'$ ,  $E$ ,  $F_0$ ,  $F_1$ , are not identically zero, we have that  $\alpha_{01} = \alpha_{02} = 0$ . By repeating the same procedure for  $k = 1, 2$ , we obtain the von Kries model.

We remark that Eq. (9) has been derived by supposing that  $E'(\lambda)$  differs from zero for any  $\lambda$  in the compact support of  $F_0(\lambda)$ . This allows us to remove the multiplicative term  $E'(\lambda)F_0(\lambda)$  and leads us to Eq. (9). This hypothesis is reliable, because the spectral power distribution of the most illuminants is not null in the visible spectrum. However, in case of lights with null energy in some wavelengths of the support of  $F_0$ , Eq. (9) is replaced by

$$E'(\lambda)E(\lambda)[F_0(\lambda)]^2 - \alpha_{00}[E'(\lambda)]^2[F_0(\lambda)]^2 = 0.$$

The derivation of the von Kries model can be then carried out as before.

### 3.3 Changing Device

A color variation between two images of the same scene can be produced also by changing the acquisition device. Mathematically turns into changing the sensitivity function  $F_i$  in Eq. (1). Here we discuss a case in which the color variation generated by a device change can be described by the von Kries model.

Without loss of generality, we can assume that the sensors are narrow bands. Otherwise, we can apply the sharpening proposed in [18] or [7]. Under this assumption, the sensitivities of the cameras used for acquiring the images under exam are

approximated by the Dirac delta, i.e.

$$F_i(\lambda) = f_i^\gamma \delta(\lambda - \lambda_i) \quad (12)$$

where parameter  $f^\gamma$  is a characteristic of the cameras.

Here we model the change of the camera as a variation of the parameter  $\gamma$ , while we suppose that the wavelength  $\lambda_i$  remains the same. Therefore the sensitivity functions changes from Eq. (12) to the following Equation:

$$F'_i(\lambda) = f_i^{\gamma*} \delta(\lambda - \lambda_i). \quad (13)$$

Consequently, the camera responses are

$$p_i(x) = f_i^\gamma E(\lambda_i) S(\lambda_i, x) \quad \text{and} \quad p'_i(x) = f_i^{\gamma*} E(\lambda_i) S(\lambda_i, x)$$

and thus, therefore the diagonal linear model still holds, but in this case, the von Kries coefficients  $\alpha_i$  depends not only on the spectral power distribution, but also on the device photometric cues:

$$\alpha_i = \frac{f_i^\gamma E(\lambda_i)}{f_i^{\gamma*} E(\lambda_i)}, \quad \forall i = 0, 1, 2. \quad (14)$$

## 4 Estimating the von Kries Map

The *color correction* of an image onto another consists into borrow the colors of the first image on the second one. When the color variation is caused by a change of illuminant, and the hypotheses of the von Kries model are satisfied, the color transform between the two pictures is determined by the von Kries map. This equalizes their colors, so that the first picture appears as it would be taken under the illuminant of the second one. Estimating the von Kries coefficients is thus an important task to achieve color correction between images different by illuminants.

The most methods performing color correction between re-illuminated images or regions compute the von Kries coefficients by estimating the illuminants  $\sigma$  and  $\sigma'$  under which the images to be corrected have been taken. These illuminants are expressed as RGB vectors, and the von Kries coefficients are determined as the ratios between the components of the varied illuminants. Therefore, estimating the von Kries map turns into estimating the image illuminants. Some examples of these techniques are the low-level statistical based methods as Gray-World and Scale-by-Max [11, 33, 55], the gamut approaches [4, 14, 20, 23, 29, 56], and the Bayesian or statistical methods [26, 41, 47, 50].

The method proposed in [35], we investigate here, differs from these techniques, because it does not require the computation of the illuminants  $\sigma$  and  $\sigma'$ , but it esti-



mates the von Kries coefficients by matching the color histograms of the input images or regions, as explained in Sect. 4.1. Histograms provide a good compact representation of the image colors and, after normalization, they guarantee invariance with respect to affine distortions, like changes of size and/or in-plane orientation.

As matter as fact, the method in [35] is not the only one that computes the von Kries map by matching histograms. The histogram comparison is in fact adopted also by the methods described in [9, 34], but their computational complexities are higher than that of the method in [35]. In particular, the work in [9] considers the logarithms of the RGB responses, so that a change in illumination turns into a shift of these logarithmic responses. In this framework, the von Kries map becomes a translation, whose parameters are derived from the convolution between the distributions of the logarithmic responses, with computation complexity  $O(N \log(N))$ , where  $N$  is the color quantization of the histograms. The method in [34] derives the von Kries coefficients by a variational technique, that minimizes the Euclidean distance between the piecewise inverse of the cumulative color histograms of the input images or regions. This algorithm is linear with the quantizations  $N$  and  $M$  of the color histograms and of the piecewise inversions of the cumulative histograms respectively, so that its complexity is  $O(N + M)$ . Differently from the approach of [34], the algorithm in [35] requires the user just to set up the value of  $N$  and its complexity is  $O(N)$ .

The method presented in [35] is described in detail in Sect. 4.1. Experiments on the accuracy and an analysis of the algorithm complexity and dependency on color quantization are addressed in Sects. 4.2 and 4.3 respectively. Finally, Sect. 4.4 illustrates an application of this method to illuminant invariant image retrieval.

#### 4.1 Histogram-Based Estimation of the von Kries Map

As in [35], we assume that the illuminant varies uniformly over the pictures. We describe the color of an image  $I$  by the distributions of the values of the three channels red, green, blue. Each distribution is represented by a histogram of  $N$  bins, where  $N$  ranges over  $\{1, \dots, 256\}$ . Hence, the color feature of an image is represented by a triplet  $\mathbf{H} := (H^0, H^1, H^2)$  of histograms. We refer to  $\mathbf{H}$  as *color histograms*, whereas we name its components *channel histograms*.

Let  $I_0$  and  $I_1$  be two color images, with  $I_1$  being a possibly rescaled, rotated, translated, skewed, and differently illuminated version of  $I_0$ . Let  $\mathbf{H}_0$  and  $\mathbf{H}_1$  denote the color histograms of  $I_0$  and  $I_1$  respectively. Let  $H_0^i$  and  $H_1^i$  be the  $i$ -th component of  $\mathbf{H}_0$  and  $\mathbf{H}_1$  respectively. Here fter, to ensure invariance to image rescaling, we assume that each channel  $H_j^i$  histogram is normalized so that  $\sum_{x=1}^N H_j^i(x) = 1$ .

The channel histograms of two images which differ by illumination are stretched to each other by the von Kries model, so that

$$\sum_{k=1}^x H_1^i(k) = \sum_{k=1}^{\alpha_i x} H_0^i(k) \quad \forall i = 0, 1, 2. \quad (15)$$

We note that, as the data are discrete, the value  $\alpha_i x$  is cast to an integer ranging over  $\{1, \dots, 256\}$ .

The estimate of the parameters  $\alpha_i$ 's consists of two phases. First, for each  $x$  in  $\{1, \dots, 256\}$  we compute the point  $y$  in  $\{1, \dots, 256\}$  such that

$$\sum_{k=1}^x H_0^i(k) = \sum_{k=1}^y H_1^i(k). \quad (16)$$

Second, we compute the coefficient  $\alpha_i$  as the slope of the *best* line fitting the pairs  $(x, y)$ .

The procedure to compute the correspondences  $(x, y)$  satisfying Eq. (16) is implemented by the Algorithm 1 and more details are presented in [35]. To make the estimate robust to possible noise affecting the image and to color quantization, the contribution of each pair  $(x, y)$  is weighted by a positive real number  $M$ , that is defined as function of the difference  $\sum_{k=1}^x H_0^i(k) - \sum_{k=1}^y H_1^i(k)$ .

The estimate of the best line  $\mathcal{A} := y = \alpha x$  could be adversely affected by the pixel saturation, that occurs when the incident light at a pixel causes the maximum response (256) of a color channel. To overcome this problem, and to make our estimate robust as much as possible to saturation noise, the pairs  $(x, y)$  with  $x = 256$  or  $y = 256$  are discarded from the fitting procedure.

A least-square method is used to define the *best line* fitting the pairs  $(x, y)$ . More precisely, the value of  $\alpha_i$  is estimated by minimizing with respect to  $\alpha$  the following functional, that is called in [35] *divergence*:

$$d_\alpha(H_0^i, H_1^i) := \sum_k M_k d((x_k, y_k), \mathcal{A})^2 = \sum_k \frac{M_k}{\alpha^2 + 1} (\alpha x_k - y_k)^2. \quad (17)$$

Here  $(x_k, y_k)$  and  $M_k$  indicate the  $k$ -th pair satisfying Eq. (16) and its weight respectively, while  $d((x_k, y_k), \mathcal{A})$  is the Euclidean distance between the point  $(x_k, y_k)$  and the line  $\mathcal{A}$ .

We observe that:

1.  $d_\alpha(H_0^i, H_1^i) = 0 \Leftrightarrow H_0^i(\alpha z) = H_1^i(z)$ , for each  $z$  in  $\{1, \dots, N\}$ ;
2.  $d_\alpha(H_0^i, H_1^i) = d_{\frac{1}{\alpha}}(H_1^i, H_0^i)$ .

These properties imply that  $d_\alpha$  is a measure of dissimilarity (*divergence*) between the channel histograms stretched each to other. In particular, if  $d_\alpha$  is zero, than the two histograms are stretched to each other.

Finally we notice that, when no changes of size or in-plane orientation occur, the diagonal map between two images  $I_0$  and  $I$  can be estimated by finding, for each color channel, the best line fitting the pairs of sensory responses  $(p_i, p'_i)$  at the  $i$ -th

pixels of  $I_0$  and  $I$  respectively, as proposed in [37]. The histogram-based approach in [35] basically applies a least square method in the space of the color histograms. Using histograms makes the estimate of the illuminant change insensitive to image distortions, like rescaling, translating, skewing, and/or rotating.

---

**Algorithm 1** Computing the correspondences between the bins of two channel histograms, according to the von Kries Model

---

**Initialization:**

Define  $R_0 := \sum_{k=1}^x H_0^i(k)$ ,  $R_1 := \sum_{k=1}^y H_1^i(k)$ ;

Compute the bin  $x$  and  $y$  of  $H_0^i$  and  $H_1^i$  respectively, such that  $R_0 > 0$  and  $R_1 > 0$ ;

Set  $M := \min(R_0, R_1)$  and  $\mathcal{L} := \{\text{list of the pairs } (x, y, M)\}$ , initially empty.

**Iterations:**

**while** ( $x < 255$  or  $y < 255$ ) **do**

  Push  $(x, y, M)$  into  $\mathcal{L}$ ;

**if** ( $M = R_0$ ) **then**

**while** ( $M < R_1$ ) **do**

$x \leftarrow x + 1$ ;

$R_0 \leftarrow R_0 - M$  and  $R_1 \leftarrow R_1 - M$ ;

$M \leftarrow \min(R_0, R_1)$ ;

**end while**

**end if**

**if** ( $M = R_1$ ) **then**

**while** ( $M < R_0$ ) **do**

$y \leftarrow y + 1$ ;

$R_0 \leftarrow R_0 - M$  and  $R_1 \leftarrow R_1 - M$ ;

$M \leftarrow \min(R_0, R_1)$ ;

**end while**

**end if**

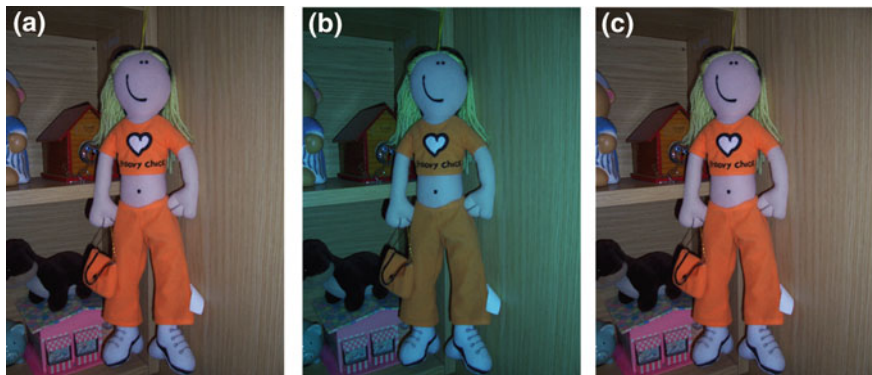
**end while**

---

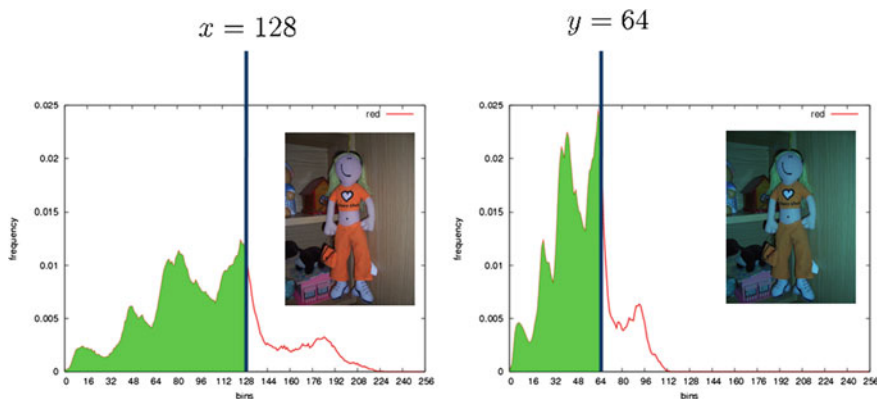
Figure 2 shows a synthetic example of pictures related by a von Kries map along with the color correction provided by the method described in [35]. The red channel of the image (a) has been rescaled by 0.5, while the other channels are unchanged. The re-illuminated image is shown in (b). Figure 3 shows the red histograms of (a) and (b) and highlights the correspondence between two bins. In particular, we note that the green regions in the two histograms have the same areas. The von Kries map estimated by [35] provides a very satisfactory color correction of image (b) onto image (a), as displayed in Fig. 2c.

## 4.2 Accuracy on the Estimate

The accuracy on the estimate of the von Kries map possibly relating two images or two image regions has been measured in [35] in terms of *color correction*. In the following, we report the experiments carried out on four real-world public databases



**Fig. 2** **a** A picture; **b** a re-illuminated version of **(a)**; **c** the color correction of **(b)** onto **(a)** provided by the method [35]. Pictures **(a)** and **(c)** are highly similar

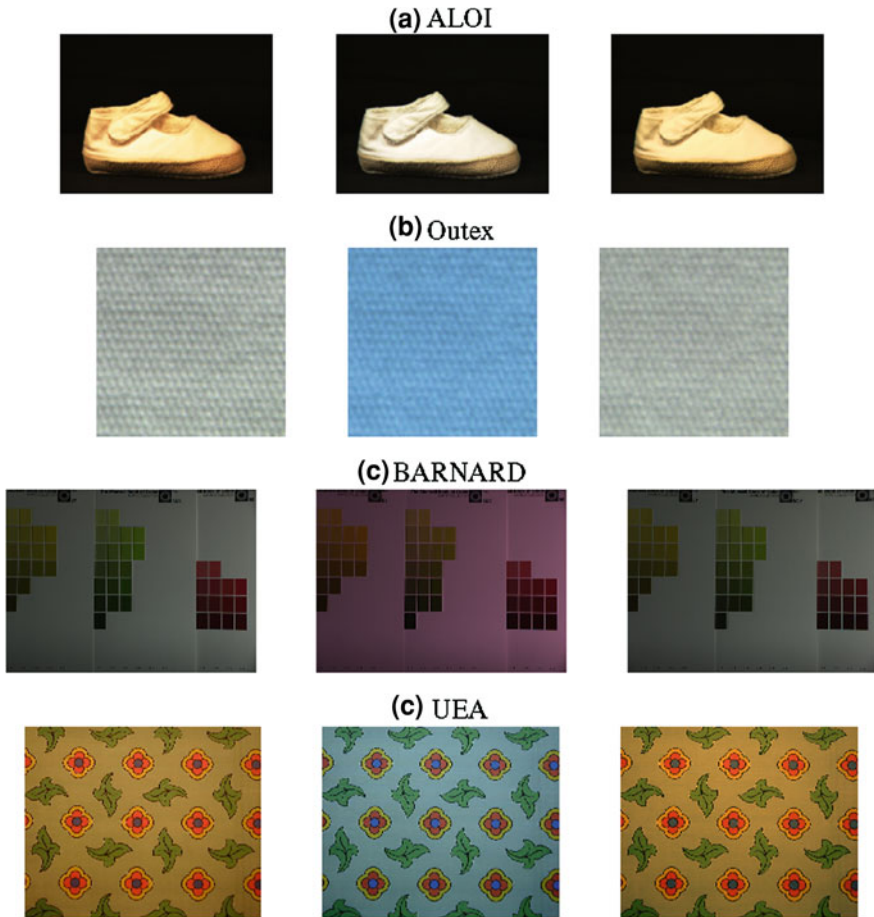


**Fig. 3** Histograms of the responses of the red channels of the pictures shown in Fig. 2a, b: the *red* channel of the first picture has been synthetically rescaled by 0.5. The two *red* histograms are thus stretched to each other. The method [35] allows to estimate the stretching parameters, and hence to correct the images as they would be taken under the same light. The *green parts* highlighted on the histograms have the same area, therefore the bin  $x = 128$  in the first histogram is mapped on the bin  $y = 64$  of the second one

(ALOI [27], Outex [45], BARNARD [8], UEA Dataset [16]). Some examples of pictures from these datasets are shown in Fig. 4 (first and second images in each row).

Each database consists of a set of images (*references*) taken under a reference illuminant and a set of re-illuminated versions of them (*test images*). For all the databases, we evaluate the accuracy on the estimation of the the von Kries map  $K$  by

$$A := 1 - L^1(I, K(I_0)). \tag{18}$$



**Fig. 4** Examples of color correction output by the approach in [35] for the four databases used in the experiments reported in Sect. 4.2: **a** ALOI, **b** Outex; **c** BARNARD; **d** UEA. In each row, from left to right: an image, a re-illuminated version of it, and the color correction of the second one onto the first one. The images in (d) have been captured by the same camera

Here  $I$  indicates a test image and  $I_0$  its correspondent reference, while  $L^1(I, K(I_0))$  is the  $L^1$  distance computed on the RGB space between  $I$  and the color correction  $K(I_0)$  of  $I_0$  determined by the estimated  $K$ . This distance has been normalized to range over  $[0,1]$ . Therefore, the closer to 1  $A$  is, the better our estimate is. To quantify the benefit of our estimate, we compare the accuracy in Eq. (18) with

$$A_0 := 1 - L^1(I, I_0). \quad (19)$$

The value of  $A_0$  measures the similarity of the reference to the test image when no color enhancement is applied.

The transform  $K$  gives the color correction of  $I_0$  with respect to the reference  $I$ : in fact,  $K(I_0)$  is the image  $I_0$  as it would be taken under the same illuminant of  $I$ .

We notice that this performance evaluation does not consider possible geometric image changes, like rescaling, in-plane rotation, or skew. In fact, the similarity between two color corrected images is defined as a pixel-wise distance between the image colors.

In case of a pair of images related by an illuminant change and by geometric distortions, we measure the accuracy of the color correction by the  $L^1$  distance between their color histograms. In particular, we compute the distance  $\mathcal{H}_0$  between the color histograms of  $I$  and  $I_0$  before the color correction

$$\mathcal{H}_0 = 1 - L^1(\mathbf{H}_0, \mathbf{H}), \quad (20)$$

and the distance  $\mathcal{H}$  between the color histograms  $\mathbf{H}$  and  $\mathbf{H}_K$  of  $I$  and  $K(I_0)$  respectively:

$$\mathcal{H} = 1 - L^1(\mathbf{H}, \mathbf{H}_K). \quad (21)$$

Examples of color correction output by the algorithm we described are shown in Fig. 4 for each database used here (third image in each row).

#### 4.2.1 ALOI

ALOI [27] (<http://staff.science.uva.nl/~aloi/>) collects 110,250 images of 1,000 objects acquired under different conditions. For each object, the frontal view has been taken under 12 different light conditions, produced by varying the color temperature of 5 lamps illuminating the scene. The lamp voltage was controlled to be  $V_j = j \times 0.047$  V with  $j \in \{110, 120, 130, 140, 150, 160, 170, 180, 190, 230, 250\}$ . For each pair of illuminants  $(V_j, V_k)$  with  $j \neq k$ , we consider the images captured with lamp voltage  $V_j$  as references and those captured with voltage  $V_k$  as tests.

Figure 5 shows the obtained results: for each pair  $(V_j, V_k)$ , the plot shows the accuracies (a)  $A_0$  and (b)  $A$  averaged over the test images.

We observe that, for  $j = 140$ , the accuracy  $A$  is lower than for the other lamp voltages. This is because the voltage  $V_{140}$  determines a high increment of the light intensity and therefore a large number of saturated pixels, making the performances worse.

The mean value of  $A_0$  averaged on all the pairs  $(V_j, V_k)$  is 0.9913, while that of  $A$  is 0.9961 by the approach in [35]. For each pair of images  $(I_i, I_j)$  representing a same scene taken under the illuminants with voltages  $V_i$  and  $V_j$  respectively, we compute the parameters  $(\alpha_0, \alpha_1, \alpha_2)$  of the illuminant change  $K$  mapping  $I_i$  onto  $I_j$ . In principle, these parameters should be equal to those of the map  $K'$  relating another pair  $(I'_i, I'_j)$  captured under the same pair of illuminants. In practice, since the von Kries model is only an approximation of the illuminant variation, the parameters of  $K$  and  $K'$  generally differ. In Fig. 6 we report the mean values of the von Kries

**Table 2** Outex: accuracies  $A_0$  and  $A$  for three different illuminant changes

Illuminant change	$A_0$	$A$
From INCA to HORIZON	0.94659	0.97221
From INCA to TL84	0.94494	0.98414
From TL84 to HORIZON	0.90718	0.97677
Mean	0.93290	0.97771

coefficients versus the reference set. The error bar is the standard deviation of the estimates from their mean value.

#### 4.2.2 Outex Dataset

The Outex database [45] (<http://www.outex.oulu.fi/>) includes different image sets for empirical evaluation of texture classification and segmentation algorithms. In this work we extract the test set named Outex\_TC\_00014: this consists of three sets of 1360 texture images viewed under the illuminants INCA, TL84 and HORIZON with color temperature 2856, 4100 and 2300 K respectively.

The accuracies  $A_0$  and  $A$  are stored in Table 2, where three changes of lights have been considered: from INCA to HORIZON, from INCA to TL84, from TL84 to HORIZON. As expected,  $A$  is greater than  $A_0$ .

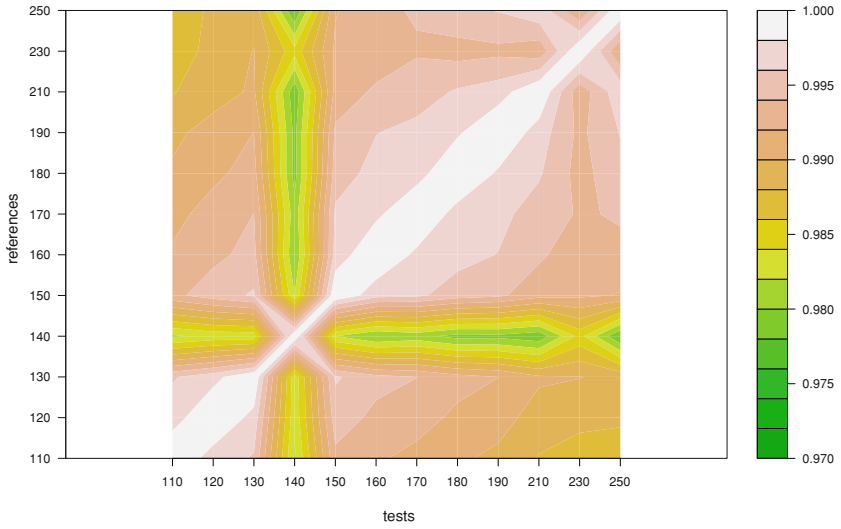
#### 4.2.3 BARNARD

The real-world image dataset [8] (<http://www.cs.sfu.ca/~colour/>), that we refer as BARNARD, is composed by 321 pictures grouped in 30 categories. Each category contains a reference image taken under an incandescent light Sylvania 50MR16Q (*reference illuminant*) and a number (from 2 to 11) of relighted versions of it (*test images*) under different lights. The mean values of the accuracies  $A_0$  and  $A$  are shown in Fig. 7. On average,  $A_0$  is 0.9447, and  $A$  is 0.9805.

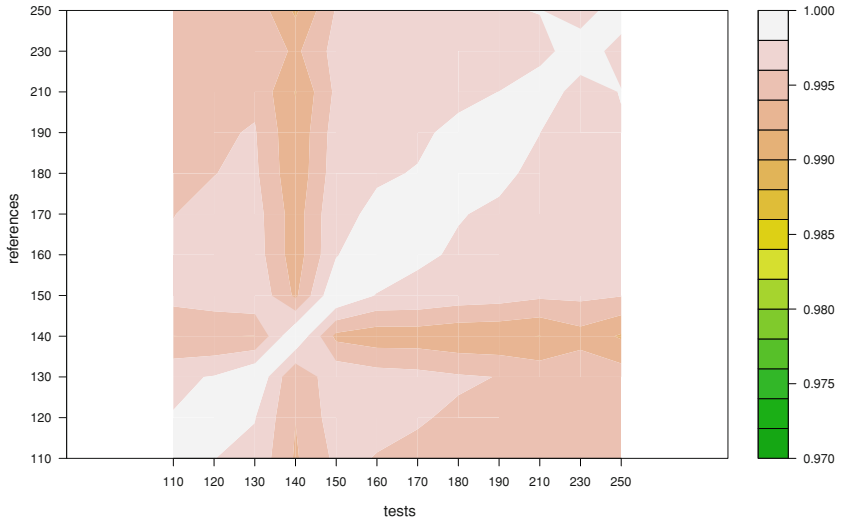
#### 4.2.4 UEA Dataset

The UEA Dataset [16] (<http://www.uea.ac.uk/cmp/research/>) comprises 28 design patterns, each captured under 3 illuminants with four different cameras. The illuminants are indicated by Ill A (tungsten filament light, with color temperature 2865 K), Ill D65 (simulated daylight, with color temperature 6500 K), and Ill TL84 (fluorescent tube, with color temperature 4100 K). We notice that the images taken by different cameras differ not only for their colors, but also for size and orientation. In fact, different sensors have different resolution and orientation. In this case, the

(a)



(b)



**Fig. 5** ALOI: accuracy **a**  $A_0$  and **b**  $A$  (see Eqs.(19) and (18)) for the different pairs of reference and test sets. The  $x$  and  $y$  axes display the lamp voltages ( $\times 0.047V$ ) of the illuminants used in ALOI. The *right axis* shows the correspondence between the colors of the plots and the values of the accuracies



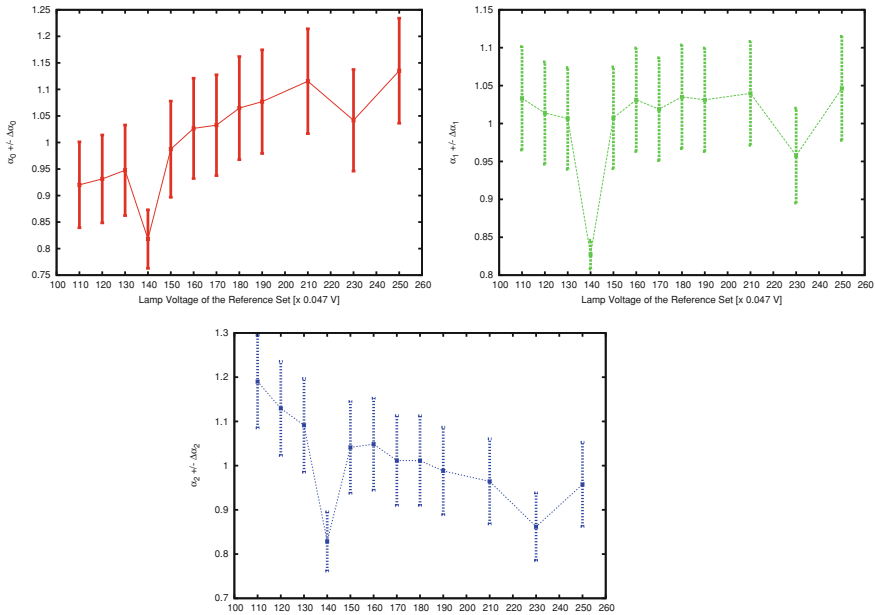
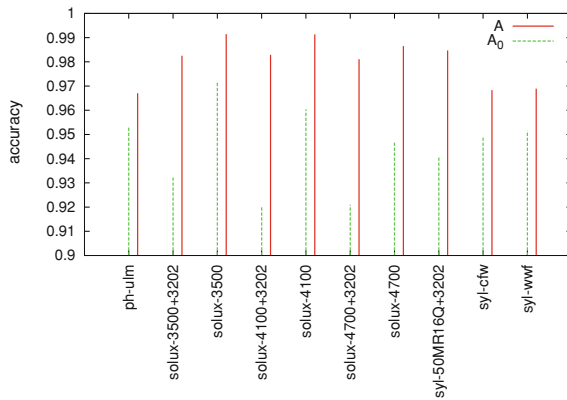


Fig. 6 ALOI: estimates of the von Kries coefficients

Fig. 7 BARNARD: accuracies  $A_0$  (Eq. 19) and  $A$  (Eq. 18) for the different illuminants



accuracy on the color correction cannot be measured by the Eqs. (18) and (19), but it is evaluated by the histogram distances defined in Eqs. (20) and (21).

The results are reported in Table 3. For each pair of cameras  $(i, j)$  and for each illuminant pair  $(\sigma, \sigma')$  we compute the von Kries map relating every image acquired by  $i$  under  $\sigma$  and the correspondent image acquired by  $j$  under  $\sigma'$ , and the accuracies  $\mathcal{H}_0$  and  $\mathcal{H}$  on the color correction of the first image onto the second one. On average, the  $L^1$  distance between the color histograms before the color correction is 0.0043, while it is 0.0029 after the color correction.

**Table 3** UEA Dataset: Accuracies  $\mathcal{H}_0$  and  $\mathcal{H}$  respectively (a) before and (b) after the color correction.

Camera	1	2	3	4
(a) Accuracy $\mathcal{H}_0$				
1	0.99756	0.99643	0.99487	0.99634
2	0.99643	0.99636	0.99459	0.99644
3	0.99487	0.99459	0.99463	0.99458
4	0.99634	0.99644	0.99458	0.99655
(b) Accuracy $\mathcal{H}$				
1	0.99859	0.99745	0.99588	0.99783
2	0.99742	0.99738	0.99565	0.99744
3	0.99691	0.99664	0.99669	0.99709
4	0.99712	0.99693	0.99541	0.99782

### 4.3 Algorithm Complexity and Accuracy versus Color Quantization

The approach presented in [35] has a linear complexity with respect to the number of image pixels, and to the color quantization  $N$ . Therefore, it is very efficient in terms of computational charge: the execution time requested to estimate the von Kries coefficients for a pair of images of size  $150 \times 200$  is less than 40 ms on a standard Pentium4 CPU Intel® Core™ i7-870 2.93 GHz, for  $N = 256$ .

The color quantization is a user input. The experiments carried out on both synthetic and real-world databases [35] show that the accuracy on the color correction decreases by decreasing the number of bins used to represent the color distributions. Figure 8 reports the mean value of the accuracy on color correction for the database ALOI for different values of  $N$  and for the twelve illuminants of the database: the best performances are obtained for  $N = 256$ .

### 4.4 An Application: Illuminant Invariant Image Retrieval

The illuminant invariant image recognition problem is stated as follows: let us consider a set of known images (*references*) and let  $I$  be an unknown image (*query*). The problem consists into find the reference  $I_0$  that displays the same (possibly re-illuminated) content of the query, i.e. the reference that visually is the most similar to the query.

The illuminant invariant recognition technique described in [35] relies on the estimate of the von Kries transform possibly linking a reference and a test image. Following [35], we compute the von Kries maps that borrow the colors of each reference onto those of the query, and we associate a dissimilarity score to each of

these transforms. The solution  $I_r$  is the image reference whose von Kries transform  $K(I_0)$  has the minimum score from  $I$ .

The dissimilarity score is defined in terms of the divergence in Eq. (17) between the color histograms  $\mathbf{H}$  and  $\mathbf{H}_r$  of  $I$  and  $I_r$  respectively:

$$\delta = \sum_i d_{\alpha_i}(H^i, H_r^i). \quad (22)$$

where  $\alpha_i$  are the von Kries coefficients we estimate. The solution of the image recognition problem is given by the image  $I_r$  of  $D$  such that the score defined by Eq. (22) is minimum.

As pointed out in [35], due to the dependency of the divergence in Eq. (17) on the von Kries coefficients,  $\delta$  is not a metric because it does not satisfy the triangular inequality. Nevertheless, it is a *query-sensitive* dissimilarity measure, because it depends on the query [3].

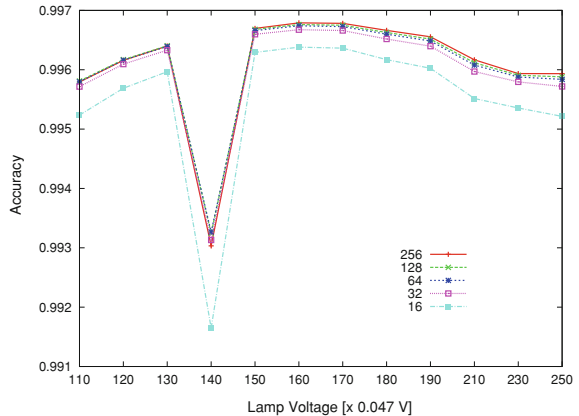
In the following, we say that a query  $I$  is *correctly* recognized if the reference image  $I_r$  of  $D$  minimizing the score in Eq. (22) is a re-illuminated version of  $I$ .

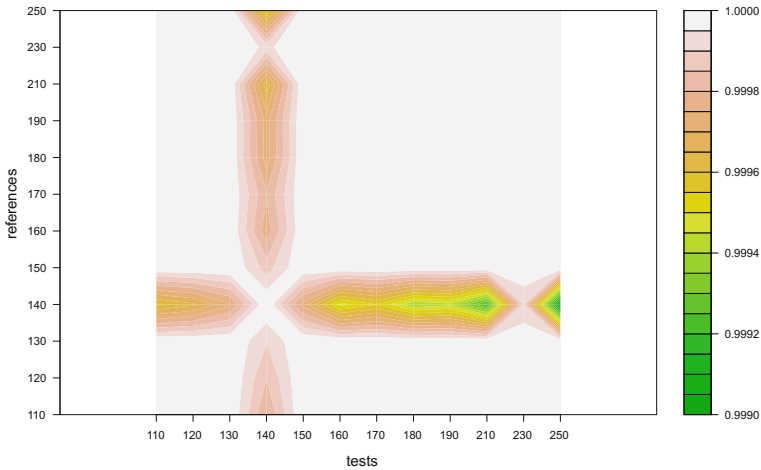
We test the accuracy of this algorithm for the illuminant invariant image retrieval on two real-world databases ALOI and UEA, that we have still described in Sects. 4.2.1 and 4.2.4. We choose the first one, because it contains a large number of images under different illuminants, and the second one because it includes images acquired under different lights and by different devices. Moreover, UEA database has been used in [13] to measure the performances of a novel illuminant- and device invariant image recognition algorithm, with which we compare our results.

We evaluate the recognition performances of our approach by the *average match percentile* (AMP) defined as the ratio (averaged over the test images)

$$\text{AMP} = \frac{Q - \text{rank}}{Q - 1}, \quad (23)$$

**Fig. 8** ALOI: mean accuracy versus the illuminants for different color quantizations





**Fig. 9** ALOI: AMP index for the different illuminants. The  $x$  and  $y$  axes display the lamp voltages ( $\times 0.047$  V) of the illuminants used in ALOI

where  $Q$  is the number of reference images and  $rank$  is the position of the correct object in the list of the models sorted by visual similarity.

In the experiments reported in the next subsections and carried out on ALOI [27] and UEA [16], we consider a color quantization with 256 bins. As well as in the case of color correction, a coarser quantization produces worse results [35] in image retrieval.

#### 4.4.1 ALOI

Each set of images captured under an illuminant with voltage  $V_j$  have been matched against each set of images captured under another illuminant with voltage  $V_k, k \neq j$ . The resulting AMP is shown in Fig. 9: in all the cases, the AMP is very high (0.9999 on average). The worst AMP has been obtained for the images taken under the lamps with voltage  $140 \times 0.047$  V: in fact, these pictures contain a high percentage of saturated pixels and, as highlighted in Sect. 4.2.1, the accuracy on the von Kries map estimate is lower than in the other cases.

#### 4.4.2 UEA Dataset

We use UEA Dataset to measure how the accuracy of the illuminant invariant image retrieval algorithm changes across (1) illuminant and device variations; (2) illuminant variations; and (3) device variations. The results have been also compared with the techniques described in [13].

**Table 4** UEA dataset: AMP by varying the cameras and the illuminants by using the method in [35]

	Cam. 1	Cam. 2	Cam. 3	Cam. 4
Cam. 1	0.9903	0.9830	0.9320	0.9725
Cam. 2	0.9784	0.9517	0.9178	0.9580
Cam. 3	0.8391	0.8363	0.9985	0.8727
Cam. 4	0.9688	0.9483	0.9606	0.9938

*Results across Illuminant and Device Variation*—For each pair  $(c, \sigma)$  of camera  $c$  and illuminant  $\sigma$  we take the images captured by  $c$  under  $\sigma$  as references and we match them with the images captured by a camera  $c^*$  under an illuminant  $\sigma^*$ . We repeat the experiments by varying all the cameras and the illuminants. When  $(c, \sigma)$  and  $(c^*, \sigma^*)$  coincide, we do not make any comparison, because in this case the references coincide with the tests. Table 4 reports the AMP averaged with respect the illuminants and the cameras.

*Results across Illuminant Variation*—Table 5 shows the AMP obtained by comparing images taken by the same camera under different illuminants. In these experiments, we fix a pair  $(c, \sigma)$ , we take the images captured by  $c$  under  $\sigma$  as references and we match them with the images captured by  $c$  under an illuminant  $\sigma^*$  different from  $\sigma$ .

In Table 6 these AMP's are broken down by illuminants and compared with the AMP's output by using the histogram equalization method (HE) and the Gray World color balancing (GW), both described in [13]. The same results are broken down by camera in Table 7.

**Table 5** UEA dataset—AMP for the four cameras and for the three illuminants (taken as references)

Camera	Ill A	Ill D65	Ill TL84	Mean AMP
1	0.9914	0.9848	0.9947	0.9903
2	0.9405	0.9550	0.9597	0.9517
3	0.9967	0.9987	1.0000	0.9985
4	0.9914	0.9921	0.9980	0.9938

The fifth column shows for each camera the AMP index averaged on the illuminants

**Table 6** UEA dataset—AMP broken down by illuminant

Method	Ill A	Ill D65	Ill TL84	Mean AMP
GW [13]	0.9008	0.9528	0.9653	0.9396
HE [13]	0.9525	0.9823	0.9873	0.9672
Method [35]	0.9800	0.9827	0.9881	0.9836

For instance the column Ill A reports the AMP averaged over the cameras when the illuminant Ill A is chosen as reference. In row “Method [35]” the AMP's are the mean values of the columns of Table 5

**Table 7** UEA dataset—AMP broken down by camera

Method	Camera 1	Camera 2	Camera 3	Camera 4	Mean AMP
GW [13]	0.9623	0.8159	0.9912	0.9890	0.9386
HE [13]	0.9925	0.9235	0.9691	0.9837	0.9672
Method [35]	0.9903	0.9517	0.9985	0.9938	0.9836

For instance the column ‘Camera 1’ reports the AMP averaged over the three illuminants when images captured by the same camera are matched. The values in the row “Method [35]” are those already reported in the last column of Table 5

**Table 8** UEA dataset—AMP across a change of device

Method	Camera 1	Camera 2	Camera 3	Camera 4	Mean AMP
GW [13]	0.9581	0.8992	0.9367	0.9750	0.9423
HE [13]	0.9816	0.9234	0.9362	0.9899	0.9578
Method [35]	0.9810	0.9652	0.8769	0.9586	0.9454

The images acquired with one camera under a fixed light are used as references, while the images captured under the same light by the other cameras are used as test images. The mean AMP averaged on the cameras is reported in the last column

*Results across Device Variation*—Table 8 shows the AMP’s for GW, HE, and for the method in [35] across a change of device: the images acquired by a camera  $c$  under a fixed illuminant  $\sigma$  are used as references, and matched with the images acquired by the other cameras under  $\sigma$ . The values reported in Table 8 are averaged over the reference illuminants. The results obtained by the approach in [35] are better than those achieved by GW and HE just for Camera 1 and Camera 2. For Camera 3 and Camera 4 HE performs better. On average, the results of our approach are very close to those output by GW, while HE outputs a higher AMP. More detailed results are reported in Table 9.

We note that when the illuminant changes, but the device is fixed, the method proposed in [35] outperforms GW and HE. On the contrary, HE provides better results than the approach in [35] and GW when the camera varies. This is because the von Kries model poorly describes a change of camera. In Sect. 3 we discussed a possible mathematical model for approximating a device variation, based on a change of the  $\gamma$  factor. However, this approximation does not take into account a possible shift of the wavelength at which a sensor maximally responds, resulting in a too coarse approximation of the variation of the sensitivity functions. On the contrary, HE does not consider any model for approximating the illuminant change, but it defines a new image representation that is illuminant- and (in many cases) device-independent. The main idea of HE relies on the following observation, empirically proved in [13] across a wide range of illuminant and devices: while changing the illuminant and the recording device leads to significant color variations, the rank orderings of the color responses is in general preserved. The desired invariant image description is hence obtained by a histogram equalization method that takes into account the rank ordering constancy. However, the gap between the results provided by [35] and HE is small.

**Table 9** UEA dataset—AMP across a change of device and illuminant when the von Kries model is used

Cam. 1 Cam. 1	III A	III D65	III TL84	Cam. 2 Cam. 1	III A	III D65	III TL84
III A	1.0000	0.9868	0.9960	III A	0.9947	0.9735	0.9868
III D65	0.9723	1.0000	0.9974	III D65	0.9458	1.0000	0.9960
III TL84	0.9947	0.9947	1.0000	III TL84	0.9723	0.9788	0.9987
Cam. 1 Cam. 3	III A	III D65	III TL84	Cam. 1 Cam. 4	III A	III D65	III TL84
III A	0.9206	0.9497	0.9286	III A	0.9788	0.9431	0.9484
III D65	0.9101	0.9577	0.9445	III D65	0.9894	0.9934	0.9828
III TL84	0.9008	0.9445	0.9312	III TL84	0.9775	0.9683	0.9709
Cam. 2 Cam. 1	III A	III D65	III TL84	Cam. 2 Cam. 2	III A	III D65	III TL84
III A	0.9828	0.9431	0.9616	III A	1.0000	0.9286	0.9524
III D65	0.9577	0.9987	0.9854	III D65	0.9101	1.0000	1.0000
III TL84	0.9828	0.9934	1.0000	III TL84	0.9193	1.0000	1.0000
Cam. 2 Cam. 3	III A	III D65	III TL84	Cam. 2 Cam. 4	III A	III D65	III TL84
III A	0.8297	0.8796	0.8664	III A	0.9140	0.8929	0.8915
III D65	0.9325	0.9537	0.9471	III D65	0.9828	0.9921	0.9696
III TL84	0.9352	0.9550	0.9616	III TL84	0.9974	0.9921	0.9894
Cam. 3 Cam. 1	III A	III D65	III TL84	Cam. 3 Cam. 2	III A	III D65	III TL84
III A	0.8320	0.8651	0.8505	III A	0.8003	0.8638	0.8783
III D65	0.8214	0.8942	0.8651	III D65	0.8095	0.8612	0.8704
III TL84	0.7500	0.8334	0.8399	III TL84	0.7593	0.8241	0.8598
Cam. 3 Cam. 3	III A	III D65	III TL84	Cam. 3 Cam. 4	III A	III D65	III TL84
III A	1.0000	0.9960	0.9974	III A	0.8796	0.8704	0.8730
III D65	0.9987	1.0000	0.9987	III D65	0.8928	0.8769	0.8796
III TL84	1.0000	1.0000	1.0000	III TL84	0.8505	0.8558	0.8743
Cam. 4 Cam. 1	III A	III D65	III TL84	Cam. 4 Cam. 2	III A	III D65	III TL84
III A	0.9696	0.9828	0.9934	III A	0.8690	0.9921	0.9974
III D65	0.9259	0.9788	0.9894	III D65	0.8638	0.9921	0.9947
III TL84	0.9272	0.9643	0.9881	III TL84	0.8466	0.9828	0.9960
Cam. 4 Cam. 3	III A	III D65	III TL84	Cam. 4 Cam. 4	III A	III D65	III TL84
III A	0.9259	0.9683	0.9445	III A	1.0000	0.9894	0.9934
III D65	0.9537	0.9775	0.9749	III D65	0.9868	1.0000	0.9974
III TL84	0.9577	0.9709	0.9723	III TL84	0.9974	0.9987	1.0000

“Cam.” stands for “Camera”

## 5 von Kries Model: Dependence on Light

In this Section, we analyze the dependence of the von Kries coefficients on the physical cues of the varied lights (i.e. color temperature and intensity). According to the work [37] result is obtained from an empirical analysis of pictures imaged under Planck's lights, i.e. illuminants satisfying Planck's law. As pointed out in Sect. 1, this assumption does not compromise the generality of the presented results, because many lights fulfill Planck's law.

The empirical analysis proposed in [37] lead us to two main results: (i) we verify that the von Kries map well approximates a change of colors due to a change of Planck's light; (ii) Planck's law constraints the 3D points whose components are the von Kries coefficients, to lie on a ruled surface, called *von Kries surface* and parametrized by the physical properties of the light. The approximated equation of the von Kries surface, we derive following [37], reveals the relationship of the von Kries coefficients with the color temperature and intensity of the illuminants.

In Sect. 5.1, we describe Planck's law and the Bradford transform, that is commonly used to model a color change due to a Planck's illuminant variation. In Sect. 5.2 we discuss the result (i), where—differently from [37]—the von Kries approximations have been computed by the method in [35]. In Sect. 5.3 we illustrate the result (ii), and finally in Sect. 5.4 we explain how the von Kries surfaces can be used for estimating the color temperature and intensity illuminant of an image.

### 5.1 Planck's Lights and Bradford Transform

A Planck's illuminant is a light satisfying Planck's law, i.e. its spectral power distribution is analytically expressed by the following formula:

$$E(\lambda, T, I) = J c_1 \lambda^{-5} \left( e^{\frac{c_2}{T\lambda}} - 1 \right)^{-1}. \quad (24)$$

In Eq. (24), variables  $\lambda$ ,  $J$  and  $T$  denote respectively the wavelength, the intensity and the color temperature of the illuminant. The terms  $c_1$  and  $c_2$  are constants ( $c_1 = 3.74183 \cdot 10^{-16} \text{ W m}^2$  and  $c_2 = 1.4388 \cdot 10^{-2} \text{ K m}$ , with  $\text{W} = \text{Watt}$ ,  $\text{m} = \text{meter}$ ,  $\text{K} = \text{Degree Kelvin}$ ).

The intensity  $J$  describes the illuminant brightness, while the color temperature measures the illuminant hue in Degrees Kelvin. For instance, the sun light at sunrise or at sunset has a color temperature between 2000 and 3000 K, while the color temperature of a candle flame ranges over [1850, 1930] K.

Usually, the color of a Planck's light is codified as the 2D vector of its chromaticities in the CIE XYZ color space. The chromaticities of the most Planck's illuminants have been tabulated empirically [30, 31], and approximated formulas are also available [43].



The *Bradford transform* [32, 57] is commonly used to model a color change due to a variation of two Planck's illuminants  $\sigma$  and  $\sigma'$ . This transform relates the XYZ coordinates  $[X, Y, Z]$  and  $[X', Y', Z']$  of the responses  $\mathbf{p}$  and  $\mathbf{p}'$  by the linear map

$$[X', Y', Z']^T = MDM^{-1}[X, Y, Z]^T, \quad (25)$$

where  $M$  is the *Bradford matrix* and  $D$  is a diagonal matrix encoding the relationship between the colorimetric properties (color temperatures and intensities) of  $\sigma$  and  $\sigma'$ .

Bradford matrix has been obtained empirically from Lam's experiments described in [32]:

$$M = \begin{bmatrix} 0.8951 & 0.2664 & -0.1614 \\ -0.7502 & 1.7135 & 0.0367 \\ 0.0389 & -0.0685 & 1.0296 \end{bmatrix} \text{ and } D = \frac{Y_\sigma}{Y_{\sigma'}} \begin{bmatrix} \frac{x_\sigma}{y_\sigma} & \frac{y_{\sigma'}}{x_{\sigma'}} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1-x_\sigma-y_\sigma}{y_\sigma} & \frac{y_{\sigma'}}{1-x_{\sigma'}-y_{\sigma'}} \end{bmatrix}$$

where  $[x_\sigma, y_\sigma]$  and  $[x_{\sigma'}, y_{\sigma'}]$  are the chromaticities of the color temperatures of  $\sigma$  and  $\sigma'$  respectively.  $Y_\sigma$  and  $Y_{\sigma'}$  are the  $Y$  coordinates of the white reference of the illuminants  $\sigma$  and  $\sigma'$  respectively.

In the RGB space, the Bradford transform at an image pixel  $x$  can be re-written as follows:

$$\mathbf{p}'^T(x) = C M D M^{-1} C^{-1} \mathbf{p}^T(x) := B \mathbf{p}^T(x), \quad (26)$$

where  $C$  is the  $3 \times 3$  matrix mapping the XYZ coordinates into the RGB coordinates and  $B := C M D M^{-1} C^{-1}$ .

## 5.2 von Kries Approximation of a Bradford Transform

The illuminants of the images in the databases Outex and UEA are Planck's lights. Therefore, the color variation relating correspondent pixels of re-illuminated images of these test sets can be expressed by the Eq. (26).

According to [17], the experiments reported in Sect. 4.2 on Outex and UEA Dataset showed that the von Kries model provides a good approximation of these illuminant changes, being the accuracy on the color correction very high. Therefore, the matrix  $B$  in Eq. (26) and the diagonal matrix  $K = \text{diag}(\alpha_0, \alpha_1, \alpha_2)$  representing the von Kries approximation of the Bradford transform described by  $B$  must be *similar*. We quantify the similarity between  $K$  and  $B$  on a synthetic dataset, as proposed in [37].

Our synthetic database is built up by re-illuminating the images of Outex by a set  $\mathcal{S}$  of Bradford transforms. More precisely, for each illuminant  $\sigma$  of the database (i.e.  $\sigma = \text{INCA, TL84, HORIZON}$ ), we considered 36 Bradford transforms, each mapping  $\sigma$  onto a Planck's illuminant with color temperature  $T_{\sigma'} = 2500 + t500$  K (and  $t = 0, \dots, 8$ ) and intensity  $J_{\sigma'} = (0.5 + i0.5)J_\sigma$  (and  $i = 0, \dots, 3$ ). Since for the database Outex there are no information about the intensity  $J_\sigma$  of the source

illuminants  $\sigma$ , the Bradford transforms of  $\mathcal{T}$  simply rescales  $J_\sigma$  by the parameters 0.5, 1.0, 1.5, 2.0.

We re-illuminated each image  $I$  of Outex by the Bradford transforms of  $\mathcal{T}$ , then for each re-lighted version  $I'$  of  $I$ , we estimate the von Kries map approximating the correspondent Planck's change of  $\mathcal{T}$ , and finally we correct the colors of  $I'$  onto the colors of  $I$ . Analogously to the results obtained in Sect. 4.2, the accuracy of the color correction on this synthetic database is very high: in fact, the mean values of  $A_0$  and  $A$  are 0.78423 and 0.97770 respectively.

These results imply that the matrix  $B$  and  $K$  are quite *similar*, i.e.

$$B := C M D M^{-1} C^{-1} \simeq K. \tag{27}$$

If  $C M D M^{-1} C^{-1} = K$ , then  $K$  and  $D$  represent the same endomorphism over the RGB color space with respect to two different bases of  $\mathbf{R}^3$ . However, since the von Kries model is just an approximation of a color variation, the equality  $B = K$  is never satisfied.

In this Chapter, we measure the difference between  $B$  and  $K$  by the *Hilbert–Schmidt inner product* between matrices, defined as

$$\langle K, B \rangle := \sum_i \langle K e_i, B e_i \rangle,$$

where  $\{e_i\}_{i=0,1,2}$  is the standard canonical basis for  $\mathbf{R}^3$ . It is easy to verify that  $\langle K, B \rangle$  is the trace of the matrix product  $K \cdot B$ . We normalize the Hilbert–Schmidt inner product as follows:

$$\langle K, B \rangle_{norm} = \frac{\langle K, B \rangle}{\left[ \sum_{i,j} K_{ij}^2 \sum_{ij} B_{ij}^2 \right]^{0.5}} \tag{28}$$

where  $K_{ij}$  and  $B_{ij}$  indicate the element  $ij$  of  $K$  and  $B$  respectively. Thus  $\langle K, B \rangle_{norm}$  ranges over  $[-1, 1]$ . The closer the module of the Hilbert–Schmidt distance (28) is to 1, more accurate the approximation represented by the diagonal matrix  $K$  is.

For the synthetic database considered here, the Hilbert–Schmidt distance averaged on the number of Bradford transforms is 0.9587. This value shows that the matrices  $B$  and  $K$  are really close to each other.

In [37], the authors model the non-perfect equality of the matrices  $K$  and  $B$  by the Equation

$$K = H B \tag{29}$$

where  $H$  is a  $3 \times 3$  non singular matrix. As matter as fact,  $H$  is really close to the identity matrix: in the synthetic experiments presented here, the mean values of the diagonal elements is 0.92, while that of the non diagonal entries is about 0.02. We also notice that in our synthetic test we do not control the pixel saturation phenomenon,

that some Bradford transforms of  $\mathcal{T}$  may generate, especially when the gap of color temperature and the intensity scale factor are greater than 3000 K and 1.5 respectively.

### 5.3 von Kries Surface

According to [37], the experiments reported in Sect. 5.2 show that, for a fixed camera, the coefficients of a von Kries approximation of a change of Planck’s illuminants are not independent to each other, but they belong to a ruled surface, parametrized by the color temperature and intensity of the varied lights. This surface is named *von Kries surface*, and its mathematical equation puts on evidence the relationship between the von Kries coefficients and the photometric cues of the light (see Fig. 10). Moreover, in order to understand how the von Kries coefficients vary across the devices, we considered the pairs of images from UEA dataset captured by different cameras, but related by the same Planck’s illuminant variation. As in [37], we discovered that different devices produce different von Kries surfaces. Therefore the von Kries surfaces can be used to characterize a device.

Let us give some mathematical details to prime the mathematical equation of a von Kries surface.

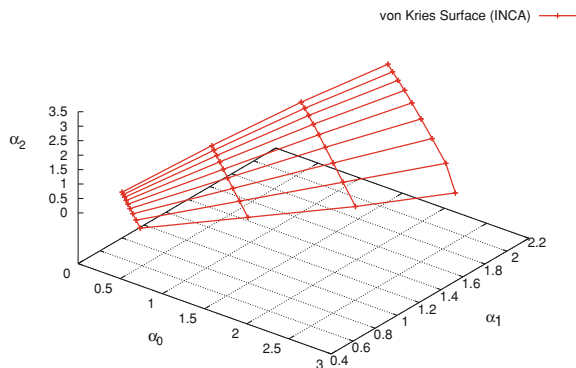
Let  $\sigma$  and  $\sigma'$  be two Planck’s lights, having respectively intensities  $J_\sigma$  and  $J_{\sigma'}$ , and color temperatures  $T_\sigma$  and  $T_{\sigma'}$ . Let  $\tau$  be the Planck illuminant change mapping  $\sigma$  onto  $\sigma'$  and let  $\mathcal{K}$  be its von Kries approximation. In particular, we have that  $\tau$  maps  $T_\sigma$  onto  $T_{\sigma'}$  and  $J_\sigma$  onto  $J_{\sigma'}$ .

The *von Kries surface relative to  $\sigma$*  is defined by the following Equations:

$$\alpha_i = \alpha_i(T_{\sigma'}, J_{\sigma'}), \quad i = 0, 1, 2 \quad (30)$$

where the  $\alpha_i$ ’s are the von Kries coefficients of  $\mathcal{K}$ .

**Fig. 10** Synthetic database (from Outex): von Kries surface for the illuminant INCA. The color temperature ranges over [2500, 6500 K], the intensity has been rescaled by 0.5, 1.0, 1.5, 2.0. The plot shows that the von Kries coefficients are non independent to each other



By combining Eqs. (26) and (29), we get  $\alpha_i = \frac{Y_\sigma}{Y_{\sigma'}} \alpha_i^*(T_{\sigma'})$ , for all  $i = 0, 1, 2$ , where  $\alpha_0^*$ ,  $\alpha_1^*$  and  $\alpha_2^*$  define the von Kries approximation of the transform  $\tau^*$  that maps the color temperature  $T_\sigma$  onto the color temperature  $T_{\sigma'}$ , while leaves unchanged the light intensity  $J_\sigma$ . By observing that  $\frac{Y_\sigma}{Y_{\sigma'}} = \frac{J_\sigma}{J_{\sigma'}}$ , we have that

$$\alpha_i = \frac{J_\sigma}{J_{\sigma'}} \alpha_i^*(T_{\sigma'}). \tag{31}$$

From the Eq. (29), we have that

$$\alpha_i^*(T_{\sigma'}) = \sum_{j=0}^2 h_{ij} b_{ji}^*(T_{\sigma'}), \quad i = 0, 1, 2 \tag{32}$$

where  $h_{ij}$  is the  $ij$ -th element of  $H$  and  $b_{ji}^*$  is the  $ij$ -th element of the matrix  $B^*$  associated to the linear transform  $\tau^*$ . Therefore we have that

$$\alpha_i(J_{\sigma'}, T_{\sigma'}) = \frac{J_\sigma}{J_{\sigma'}} \sum_{j=0}^2 h_{ij} b_{ji}^*(T_{\sigma'}). \tag{33}$$

Equation (33) makes evident the relationship between the von Kries coefficients and the photometric cues of the illuminants  $\sigma$  and  $\sigma'$ . Equation (33) describes a ruled surface depending on the intensity and on the color temperature of  $\sigma'$ . Varying discretely the intensity and the color temperature of the source illuminant  $\sigma$  produces a *sheaves* of such surfaces. Therefore, each illuminant  $\sigma$  defines a von Kries surface  $\mathcal{S}$ , that is completely determined by the Equation  $\alpha^*(T'_\sigma) = (\alpha_0^*(T'_\sigma), \alpha_1^*(T'_\sigma), \alpha_2^*(T'_\sigma))$ , as each point on  $\mathcal{S}$  is a rescaled version of a point onto  $\alpha^*(T'_\sigma)$ . As a consequence, a set of Bradford transforms changing  $\sigma$  to another Planck's light with different color temperature suffice to estimate the von Kries surface relative to  $\sigma$ , while changes of intensity are not needed.

As output from our experiments, since the von Kries coefficients depend on the device, the von Kries surface relative to any illuminant  $\sigma$  differs from device to device.

### 5.4 An Application: Estimating Color Temperature and Intensity from a von Kries Surface

Here we briefly discuss how the von Kries surfaces can be used to the estimate of the color temperature and the intensity of an illuminant. This estimation is a crucial task for many imaging applications, as for instance [46, 51].

Let  $\sigma$  be a Planck's illuminant with known color temperature  $T_\sigma$  and intensity  $J_\sigma$ . Let  $\mathcal{S}$  the von Kries surface of a camera  $c$  with respect to the source illuminant  $\sigma$ . Let

$I$  and  $I'$  two pictures depicting the same scene captured by  $c$  under the illuminants  $\sigma$  and  $\sigma'$  respectively, where  $\sigma'$  is supposed to be Planckian.

We determine the color temperature  $T_{\sigma'}$  and the intensity  $J_{\sigma'}$  of  $\sigma'$  from the von Kries surface  $\mathcal{S}$  as follows. First, we estimate the von Kries coefficients  $\beta_0, \beta_1, \beta_2$  of the von Kries map relating  $I$  and  $I'$ . Then, we compute the triplet  $(\alpha_0, \alpha_1, \alpha_2)$  on the von Kries map having the minimum Euclidean distance from  $(\beta_0, \beta_1, \beta_2)$ . Finally, we compute the color temperature and intensity correspondent to  $(\alpha_0, \alpha_1, \alpha_2)$  as the actual color temperature and intensity of  $\sigma'$ .

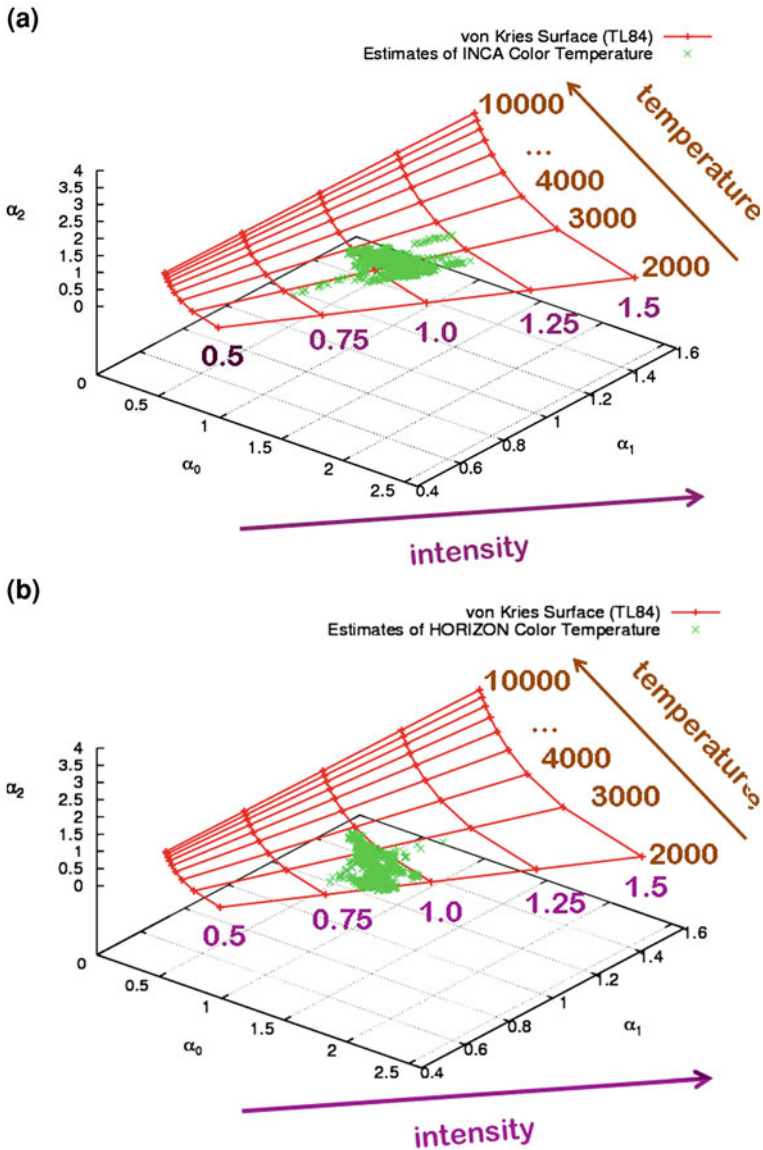
Here we report the case study presented in [37], where the authors considered the image pairs  $(I, I')$ , with  $I$  being an Outex picture imaged under TL84, and with  $I'$  being the same picture captured under INCA or HORIZON. Figures 11a, b show the von Kries surface  $\mathcal{S}$  relative to TL84. The coefficients of the von Kries map transforming (a) TL84 onto INCA and (b) TL84 onto HORIZON respectively are displayed over the von Kries surface in green: these estimates determine a range of color temperatures and intensities. The color temperature of INCA we estimate ranges over [3000, 4000] K, but the 70% about of the estimates are closer to 3000 K than to 4000 K. The variability range of the estimates of the color temperature of HORIZON is [2000, 4000] K, with the most part of the data (about the 90%) in [2000, 3000] K. Similarly, the estimated von Kries coefficients determine a variability range for the intensity, with the 99% of the estimates between 1.0 and 1.25 for INCA and between 0.75 and 1.0 for HORIZON.

The accuracy on these estimates is close to the actual value of the color temperatures and intensities of INCA and HORIZON, but it could be further refined by considering a finer grid of color temperatures in the computation of the von Kries surface and by restricting the search for the triplet minimizing the Euclidean distance with the surface to the ranges found before. Nevertheless, in general, obtaining an accurate estimate of these photometric parameters is a hard problem [51], also when calibrated images are used [22].

## 6 von Kries Model: Dependence on Device

In Sect. 5, we observed that von Kries surfaces can be used to characterize a sensor, but their equation does not reveal the relationships of the von Kries coefficients with the camera photometric cues. This dependence has been investigated in the work in [36], that we describe in this Section.

Wien's law holds for Planck's lights with color temperature varying over [2000, 10500] K, and their spectral power distribution is approximated by Wien's law. Following the work in [36], we combine the Wien's law with the von Kries model (Sect. 6.1), and we show that—as well as for Planck's lights—the coefficients of the von Kries approximation of a change of Wien's illuminants are not independent to each other (Sect. 6.2). Then we derive a mathematical equation linking the von Kries coefficients to the spectral sensitivities of the device. Since spectral sensitivities are often not reported in the technical manuals of the most cameras, and their



**Fig. 11** Outex: estimates of the color temperature and of the intensity for the illuminants **a** INCA and **b** HORIZON by using the von Kries surface with respect to TL84. Adapted from [37]

estimation requires that the camera and an image calibration target are available, this equation proposes an alternative way to estimate these data. In fact, it allows to recover the sensor spectral cues directly from the parameters of a Wien's illuminant variation, and thus to characterize a camera from a set of image pairs depicting the

same scene and related by a light change (Sect. 6.3). This von Kries model based camera characterization applies to the work in [19], where an illuminant invariant image representation, which requires to know the camera spectral sensitivities is calculated (Sect. 6.4).

### 6.1 von Kries Model under Wien's Law

Wien's law is a special case of Planck's law: it holds for Planck's light whose color temperature ranges over [2000, 10500] K. In this interval,  $e^{-\frac{c_2}{T\lambda}} \simeq (e^{\frac{c_2}{T\lambda}} - 1)^{-1}$ , so that the spectral power distribution of a Wien's light is

$$E(\lambda, T, I) = J c_1 \lambda^{-5} e^{-\frac{c_2}{T\lambda}}. \quad (34)$$

Variables  $\lambda$ ,  $J$  and  $T$ , and the constant terms  $c_1$  and  $c_2$  has the same meaning described in Sect. 5.1 for Eq. (24). Daylight, direct sunlight, candle lights and many fluorescent lights satisfy Wien's law in Eq. (34).

Similarly to the case of Planck's lights, any change of Wien's illuminant can be described by the Bradford transform in Eq. (25), that is well approximated by the von Kries model. In Sect. 5.2, we show that the accuracy on color correction provided by the method in [35] for Planck's illuminants is very high: the same holds for variations of Wien's lights. Further tests can be found in [36].

The mathematical expression of the spectral power distribution of a Wien's illuminant allows to recover an interesting relationship between the von Kries coefficients and the photometric properties of the camera used for the image acquisition. Here we describe the mathematical study presented in [36].

Without loss of generality, we assume that the camera is narrow-band. Therefore, by Eq. (34) we have that

$$p_i(x) = E(\lambda_i) S(\lambda_i, x) F(\lambda_i) = J c_1 \lambda_i^{-5} e^{-\frac{c_2}{T\lambda_i}} S(\lambda_i, x) F_i.$$

Let us now consider the *chromaticities* at an image pixel  $x$ , i.e. the ratios  $\frac{p_k(x)}{p_i(x)}$  for any  $k \neq i$ ,  $k, i = 0, 1, 2$ , and  $p_i(x) \neq 0$ . We have that

$$\frac{p_k(x)}{p_i(x)} = \left( \frac{\lambda_i}{\lambda_k} \right)^5 \frac{F_k S(\lambda_k, x)}{F_i S(\lambda_i, x)} e^{-\frac{c_2}{T} \left( \frac{1}{\lambda_k} - \frac{1}{\lambda_i} \right)}$$

and thus for each  $p_k(x) \neq 0$

$$\log \frac{p_k(x)}{p_i(x)} = \log \left[ \left( \frac{\lambda_i}{\lambda_k} \right)^5 \frac{F_k S(\lambda_k, x)}{F_i S(\lambda_i, x)} \right] - \frac{c_2}{T} \left( \frac{1}{\lambda_k} - \frac{1}{\lambda_i} \right). \quad (35)$$

By Eq. (35), the *log-chromaticity*  $\log \frac{p_k(x)}{p_i(x)}$  is expressed as the sum of two terms: the first one depends just on the reflectance function and on some properties of the camera, while the second one depends just on the color temperatures  $T$  and  $T'$ , and on the wavelength  $\lambda_k$  and  $\lambda_j$  at which the  $k$ th and the  $i$ th sensors maximally respond. If  $\mathbf{p}'(x)$  is the response of the same camera at a pixel  $x$  under a different Wien's illuminant with color temperature  $T'$ , we have that

$$\log \frac{p_k(x)}{p_i(x)} - \log \frac{p'_k(x)}{p'_i(x)} = c_2 \left( \frac{1}{T} - \frac{1}{T'} \right) \left( \frac{1}{\lambda_k} - \frac{1}{\lambda_i} \right). \quad (36)$$

By the von Kries model,  $p_k(x) = \alpha_k p'_k(x)$  ( $k = 0, 1, 2$ ), and thus

$$\left[ \log \frac{\alpha_k}{\alpha_i}, \log \frac{\alpha_j}{\alpha_i} \right] = c_2 \left( \frac{1}{T} - \frac{1}{T'} \right) \left[ \frac{1}{\lambda_k} - \frac{1}{\lambda_i}, \frac{1}{\lambda_j} - \frac{1}{\lambda_i} \right]. \quad (37)$$

Equation (37) leads us to two main issues, that we discuss in the next Subsections.

## 6.2 von Kries Coefficients are Constrained by Wien's Law

As for Planck's lights, the coefficients of a von Kries map approximating a Wien's illuminant change are not independent to each other. More precisely, from Eq. (37) we get the following Equations:

$$\begin{cases} \alpha_k = e^{c_2 \left( \frac{1}{T} - \frac{1}{T'} \right) \left( \frac{1}{\lambda_k} - \frac{1}{\lambda_i} \right)} \alpha_i \\ \alpha_j = e^{c_2 \left( \frac{1}{T} - \frac{1}{T'} \right) \left( \frac{1}{\lambda_j} - \frac{1}{\lambda_i} \right)} \alpha_i \end{cases} \quad (38)$$

Equation (38) show that the von Kries coefficients  $\alpha_k$  and  $\alpha_j$  are linearly proportional to  $\alpha_i$  through a constant that depends on the camera photometric cues and on the color temperatures of the varied illuminants.

Moreover, we observe that when the source and target illuminants have the same color temperature, i.e.  $T = T'$ , then  $\alpha_k = \alpha_i = \alpha_j$ . Of course, the left term of Eq. (37) is zero. This implies that the illuminant variation is just an intensity variation, and the von Kries coefficients correspond to the scale factors between the intensities of the source and target lights.

The linear dependency between the von Kries coefficients expressed by Eq. (38) is generally not perfectly satisfied in the real-world use cases, i.e. the ratios  $\frac{\alpha_k}{\alpha_i}$  and  $\frac{\alpha_j}{\alpha_i}$  are not constant. This is due to two main reasons. First of all, the images are affected by noise, as for instance pixel saturation. Secondly, the Dirac delta is just a rough approximation of the spectral sensitivities of a camera.



### 6.3 Device Characterization Through von Kries Maps

Equation (37) states that the *von Kries log-chromaticity vector*  $\underline{\alpha}$  and the *camera sensitivity vector*  $\underline{\lambda}$

$$\underline{\alpha} := \left[ \log \frac{\alpha_k}{\alpha_i}, \log \frac{\alpha_j}{\alpha_i} \right] \quad \underline{\lambda} := \left[ \frac{1}{\lambda_k} - \frac{1}{\lambda_i}, \frac{1}{\lambda_j} - \frac{1}{\lambda_i} \right]$$

are parallel. Following [36], we observe that the versor

$$\underline{u} := \frac{\underline{\alpha}}{\|\underline{\alpha}\|} \quad (39)$$

and the vector  $\underline{\lambda}$  defines the same line in  $\mathbf{R}^2$ , because

$$\underline{u} = \pm \frac{\underline{\lambda}}{\|\underline{\lambda}\|} \quad (40)$$

Versor  $\underline{u}$  does not depend on the physical properties of the light (i.e. intensity and color temperature), apart from its sign that is determined by the ratio

$$\frac{\frac{1}{T} - \frac{1}{T'}}{\left| \frac{1}{T} - \frac{1}{T'} \right|}.$$

The versors  $\underline{u}$  and  $\frac{\underline{\lambda}}{\|\underline{\lambda}\|}$  can be expressed as

$$\underline{u} = (\cos \theta, \sin \theta) \quad \text{and} \quad \frac{\underline{\lambda}}{\|\underline{\lambda}\|} = (\cos \theta^*, \sin \theta^*) \quad (41)$$

where  $\theta$  and  $\theta^*$  are respectively the angle between  $\underline{u}$  and the  $x$ -axis and the angle between  $\underline{\lambda}$  and the  $x$ -axis.

From Eq. (40), we have that

$$\theta = \theta^* + n\pi, \quad n \in \mathbf{Z}, \quad (42)$$

i.e. knowing the von Kries map that approximates a change of Wien's illuminants allows to recover the parameter  $\theta^*$  that describes a photometric characteristic of the acquisition device.

The variations of Planck's lights has been used in Sect. 5.3 to characterize a device through its von Kries surfaces. Here the approximation of the variations of Wien's illuminants with the von Kries model allows to characterize the camera through an estimate  $\theta$  of the parameters  $\theta^*$ . Different devices can have different values of  $\theta^*$  and, according to this fact, for any fixed illuminant change  $\tau$ , the von Kries approximations of  $\tau$  produced by these devices can differ to each other. However, we remark that the

estimate in Eq. (42) differs from the actual value of  $\theta^*$  by  $n\pi$ , where  $n$  is an integer number.

We observe that generally  $\theta^* \neq n\pi$  for any  $n$  in  $\mathbf{Z}$ . In fact, if  $\theta^* = n\pi$ ,  $\theta^*$  should be parallel to the  $x$ -axis, and consequently we should have a camera with  $\lambda_j = \lambda_k$ , that in practice does not happen for RGB devices. Analogously, we exclude the case  $\theta^* = 0.5\pi + n\pi$ , that implies  $\lambda_k = \lambda_j$ .

As pointed out in Sect. 6.2, for intensity changes the vector  $\underline{\alpha}$  is null and thus such light variations are inadequate to estimate  $\theta^*$ . For the same reason, slight changes of color temperature are also inappropriate. A general recommendation to get a reliable estimate, is to consider illuminant changes where at least one of the components of  $\underline{\alpha}$  is out of the range (0.90, 1.10) [36].

In principle, the estimate of  $\underline{u}$  can be done also when a single image  $I$  and the color temperature  $T$  of the illuminant under which it has been taken are known. Namely, in this case, we synthetically generate a second image  $I'$  by re-illuminating  $I$  with a Bradford transform that changes  $T$  and then estimate the vector  $\underline{\alpha}$ . Nevertheless, due to the presence of many sources of noise like a large number of saturated pixels and/or high percentages of black pixels, it is recommendable to consider a set of images and then estimate  $\underline{\alpha}$  by a statistical analysis of the data.

As matter as fact, since the Dirac delta are just a poor approximation of the spectral sensitivities of a camera, and since the images are often affected by noise (e.g. pixel saturation), the estimation of *the* angle  $\theta^*$  from the von Kries coefficients is generally intractable. What is feasible is to estimate a *variability range* of  $\theta^*$ , as shown in the experiments on the databases Outex [45] and FUNT1998 [24], illustrated in the next Subsections.

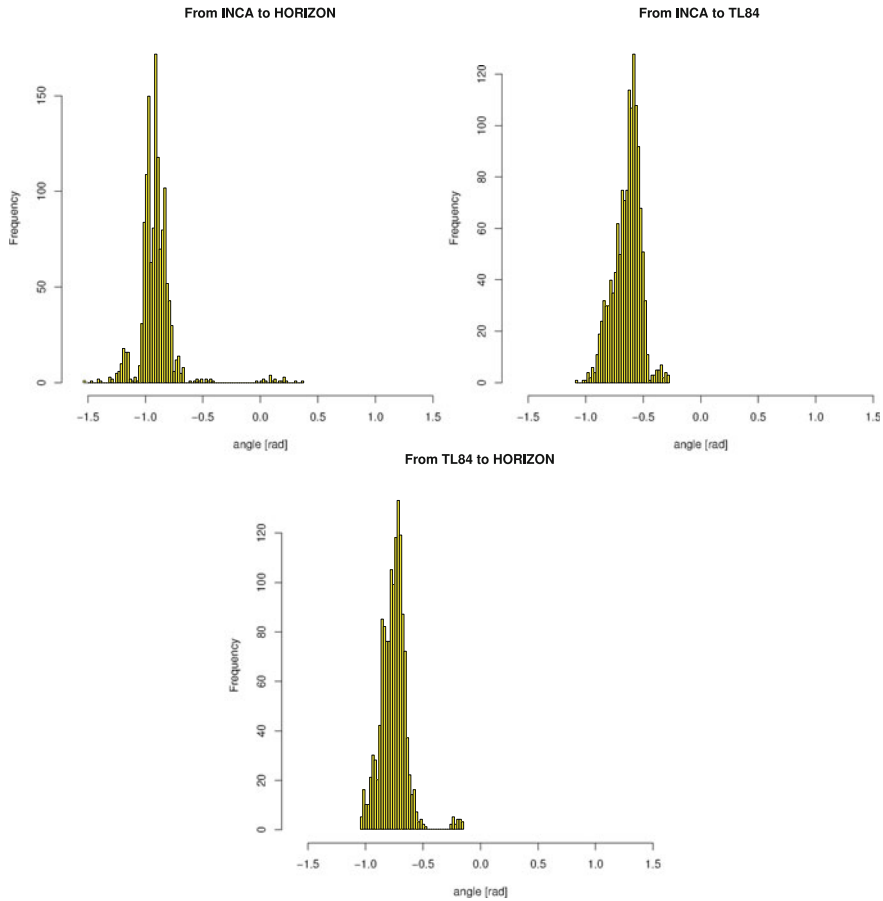
### 6.3.1 Outex

Here we consider the dataset Outex [45]. In particular, we estimate  $\theta^*$  from the image pairs related by the following three illuminant changes: (a) from INCA to HORIZON; (b) from INCA to TL84; (c) from TL84 to HORIZON. The distributions of the estimates  $\theta$  are displayed in Fig. 12: they show a peak close to their mean values, with a standard deviation equal to 0.22 (about  $10^\circ$ ). The mean values of  $\underline{u}$  are reported in Table 10.

### 6.3.2 FUNT1998

In this Section we describe the experiments carried out on the database [24] used in [36]. Here we refer to this database as FUNT1998.

The dataset FUNT1998 [24] contains 55 images of 11 objects taken under 5 different Wien's illuminants and the spectral sensitivities of the camera used for image acquisition (a Sony DXC-930) are available and displayed in Fig. 1. By approximating each spectral sensitivity with a Dirac delta centered in the wavelength at which the sensitivity is maximum, the direction of  $\underline{\lambda}$  is  $\theta^* = -0.9675$  rad.



**Fig. 12** Real-world experiments: distributions of the direction  $u$

**Table 10** Outex: versors  $\underline{u}$  for three illuminant changes

Illuminant change	$\underline{u}$
From INCA to HORIZON	(0.5974; -0.8019)
From INCA to TL84	(0.8183; -0.5747)
From TL84 to HORIZON	(0.7306; -0.6828)

Adapted from [36]

We compare this value with that output by the von Kries based estimation described before.

First of all, we consider the illuminant variation relating the image pairs taken under the following illuminant pairs: (halogen, syl-cwf), (halogen, mb-5000), (halogen, mb-5000+3202) and (halogen, ph-uhl). For each transform  $\tau$  interrelating two illuminants, we compute the versor  $\underline{u}$  from the von Kries approximation of  $\tau$ . The

**Table 11** FUNT1998: estimates of the von Kries coefficients for different illuminant pairs

Illuminant Change from halogen to	$\alpha_0$	$\alpha_1$	$\alpha_2$
mb-5000	$(2.274 \pm 0.231)$	$(1.259 \pm 0.129)$	$(0.651 \pm 0.268)$
mb-5000+3202	$(5.427 \pm 1.099)$	$(1.939 \pm 0.329)$	$(0.509 \pm 0.081)$
ph-ulm	$(1.196 \pm 0.233)$	$(0.952 \pm 0.184)$	$(0.931 \pm 0.181)$
syl-cwf	$(1.379 \pm 0.223)$	$(1.064 \pm 0.165)$	$(0.607 \pm 0.165)$

Adapted from [36]

**Table 12** FUNT1998: estimates of the direction of  $\underline{\lambda}$ 

Illuminant change from halogen to	$\theta \pm \Delta\theta$
mb-5000	$-0.8427 \pm 0.0668$
mb-5000+3202	$-0.9184 \pm 0.0589$
syl-cwf	$-1.1355 \pm 0.1099$

Adapted from [36]

mean values of the von Kries coefficients (averaged on the number of image pairs) along with their standard deviations as error are displayed in Table 11.

On FUNT1998, the error on the estimate of the von Kries coefficients is quite high. Nevertheless, the accuracy on the color correction is good: when no color correction is performed, the mean accuracy  $A_0$  is about 0.9682, while  $A$  is 0.9884. Since the color change from halogen to ph-ulm is close to the identity map, it cannot be employed to estimate  $\theta^*$ , because the log-chromaticities are null. Consequently, we just consider the illuminants syl-cwf, mb-5000 and mb-5000+3202. The estimates of  $\theta^*$  are listed in Table 12: the value of  $\theta$  is the mean value of the estimates of  $\underline{u}$  averaged over the 11 image pairs of FUNT1998, and the error is the standard deviation of the estimates from their mean value. The mean value of  $\theta$  across the changes of illuminants is  $-0.9655 \pm 0.1487$  rad.

## 6.4 An Application: Intrinsic Image Computation

In this Section, we explain how the estimate of  $\theta^*$  has been used in [36] to obtain the illuminant invariant image representation proposed in [19]. This *intrinsic* image is computed in [19] by observing that the log-chromaticity vector

$$\underline{\chi}(x) = \left[ \log \frac{p_k(x)}{p_i(x)}, \log \frac{p_j(x)}{p_i(x)} \right] \quad (43)$$

belongs to a line with direction

$$\underline{\lambda} = \left[ \frac{1}{\lambda_k} - \frac{1}{\lambda_i}, \frac{1}{\lambda_j} - \frac{1}{\lambda_i} \right]. \quad (44)$$

Let  $\underline{\lambda}^\perp$  be the 2D vector orthogonal to  $\underline{\lambda}$ . The intrinsic image of [19] is obtained by projecting the vector  $\underline{\chi}(x)$  onto  $\underline{\lambda}$ , for each pixel  $x$  of the input color picture. The intrinsic image (that is a one-channel image) is invariant to changes of Wien's illuminants. In fact, for each  $x$

$$\begin{aligned} \langle \underline{\chi}(x), \underline{\lambda}^\perp \rangle &= \log \left[ \left( \frac{\lambda_i}{\lambda_k} \right)^5 \frac{F_k S(\lambda_k, x)}{F_i S(\lambda_i, x)} \right] \left( \frac{1}{\lambda_k} - \frac{1}{\lambda_i} \right) \\ &+ \log \left[ \left( \frac{\lambda_i}{\lambda_j} \right)^5 \frac{F_j S(\lambda_j, x)}{F_i S(\lambda_i, x)} \right] \left( \frac{1}{\lambda_j} - \frac{1}{\lambda_i} \right) \end{aligned}$$

and this value does not depend on the illuminant.

Unfortunately, the value of  $\underline{\lambda}$  is generally unknown, because often the technical manuals of the most cameras do not report this information. When the camera or an image of a calibration target are available [12], the estimation of the camera sensitivity function is straightforward. However, in many applications, as for instance retrieving from the net pictures that are visually similar to an example, there are no information about the acquisition devices.

In order to overcome this lack, the work in [36] observes that the 2D vector  $\underline{u}$  is parallel to  $\underline{\lambda}$  and the scalar product  $\langle \underline{\alpha}^\perp, \underline{\chi}(x) \rangle$  is also invariant to variation of Wien's illuminants for each pixel  $x$ . To avoid possible ambiguities due to the sign of  $\underline{u}$ , the authors of [36] define a *canonical* orientation, for instance we require—as suggested above—that the basis  $\{\underline{u}^\tau, (\underline{u}^\tau)^\perp\}$  for  $\mathbf{R}^2$  is positively oriented.

#### 6.4.1 Quasi-Intrinsic Image: Tests

Vector  $\underline{u}$  differs from  $\underline{\lambda}$  for the module and for the sign. According to [36], we call *quasi-intrinsic image* the illuminant invariant image representation obtained by using  $\underline{u}$  instead of  $\underline{\lambda}$  to distinguish it from the *intrinsic image* of the work [19]. Thus, the quasi-intrinsic image differs from that proposed in [15] by a multiplicative factor. As the intrinsic image, quasi-intrinsic images of pictures related by a Wien's illuminant change are equal. Here we report the empirical analysis carried out in [36] to show that two quasi-intrinsic images computed from two images of the same scene captured under different Wien's lights are similar. Similarity between two quasi-intrinsic images is defined as the  $L^1$  distance between their values.

Table 13 reports the similarity measures for the database Outex. The values are averaged across the number of images and normalized to range over [0, 1]. Since the estimates of  $\underline{u}$  of Table 10 are very close to each other, we obtained similar distances for all the cases.

**Table 13** Outex: the first column reports three illuminant changes; for each of them, the second column reports the mean  $L^1$  RGB distances  $d_{RGB}$  between the re-illuminated images; the third, fourth, and fifth columns report the mean  $L^1$  distances  $d_{qi}$  between the quasi-intrinsic images, computed by using three different estimates of  $\underline{u}^\perp$ : (a)  $\underline{u}^\perp = (0.5747, -0.8183)$ , (b)  $\underline{u}^\perp = (0.8019, -0.5974)$ , (c)  $\underline{u}^\perp = (0.6828, -0.7306)$

Illuminant change	$d_{RGB}$	$d_{qi}$ (a)	$d_{qi}$ (b)	$d_{qi}$ (c)
From INCA to HORIZON	0.0534	0.0113	0.0108	0.0112
From INCA to TL84	0.0551	0.0131	0.0145	0.0143
From HORIZON to TL84	0.0928	0.0248	0.0251	0.0252

## 7 Conclusions

This Chapter investigated the *von Kries model*, that is commonly used to approximate the color change occurring between two images depicting the same scene, but captured under different lights. Three main issues, recently published in [35–37], have been addressed: estimation of the parameters of the von Kries model, and two theoretical studies, that reveals the dependency of the von Kries coefficients on the physical properties of the light and on the photometric cues of the camera. These results have been discussed in details, and many applications have been proposed. The experiments reported in the Chapter have been carried out on synthetic and real-world databases, all freely available, in order to allow the reproducibility and the verification of the results, and further comparisons with other approaches.

## References

1. Agarwal V, Abidi BR, Koschan A, Abidi MA (2006) An overview of color constancy algorithms. *J Pattern Recogn Res* 1:42–56
2. Agarwal V, Gribok AV, Abidi MA (2007) Neural networks letter: machine learning approach to color constancy. *Neural Netw.* 20(5):559–563
3. Athitsos V, Hadjieleftheriou M, Kollios G, Sclaroff S (2007) Query-sensitive embeddings. *ACM Trans Database Syst* 32(2):37
4. Barnard K (2000) Improvements to gamut mapping colour constancy algorithms. In: *Proceedings of the 6th European conference on computer vision-Part I, ECCV '00*, London, UK. Springer, New York, pp 390–403
5. Barnard K, Cardei V, Funt B (2002) A comparison of computational color constancy algorithms. Part I: Methodology and experiments with synthesized data. *IEEE Trans Image Process* 11(9):972–984
6. Barnard K, Cardei V, Funt B (2002) A comparison of computational color constancy algorithms. Part II: experiments with image data. *IEEE Trans Image Process* 11(9):985–996
7. Barnard K, Ciurea F, Funt B (2001) Sensor sharpening for computational color constancy. *J Opt Soc Am A* 18:2728–2743
8. Barnard K, Martin L, Funt B, Coath A (2002) A data set for color research. *Color Res Appl* 27(3):148–152
9. Berens J, Finlayson GD (2000) Log-opponent chromaticity coding of colour space. In: *Proceedings of 15th international conference on pattern recognition*, vol 1, pp 206–211

10. Brainard DH, Kraft JM, Longere P (2005) Color constancy: developing empirical tests of computational models. Oxford University Press, New York
11. Buchsbaum G (1980) A spatial processor model for object colour perception. *J Franklin Inst* 310(1):1–26
12. Ebner M (2007) Estimating the spectral sensitivity of a digital sensor using calibration targets. In: *GECCO '07: Proceedings of the 9th annual conference on genetic and evolutionary computation*, ACM, pp 642–649
13. Finlayson C, Hordley S, Schaefer G, Tian GY (2003) Illuminant and device invariance using histogram equalisation. In: *IS&T and SID's 11th color imaging conference*, pp 205–211
14. Finlayson G, Hordley S (1998) A theory of selection for gamut mapping color constancy. In: *Proceedings of the IEEE computer society conference on computer vision and pattern recognition, CVPR '98*, IEEE Computer Society, Washington, DC
15. Finlayson G, Hordley S (2001) Color constancy at a pixel. *J Opt Soc Am A* 18(2):253–264
16. Finlayson G, Schaefer G, Tian GY (2000) The UEA uncalibrated colour image database. Technical report SYS-C00-07, School of Information Systems, University of East Anglia, Norwich, United Kingdom
17. Finlayson GD, Drew MS, Funt BV (1993) Diagonal transforms suffice for color constancy. In: *Proceedings of international conference of computer vision*
18. Finlayson GD, Drew MS, Funt BV (1994) Color constancy: generalized diagonal transforms suffice. *J Opt Soc Am A* 11(11):3011–3019
19. Finlayson GD, Drew MS, Funt BV (1994) Spectral sharpening: sensor transformations for improved color constancy. *J Opt Soc Am A* 11(5):1553–1563
20. Finlayson GD, Hordley SD (2006) Gamut constrained illuminant estimation. *Int J Comput Vis* 67:2006
21. Finlayson GD, Schiele B, Crowley JL (1998) Comprehensive colour image normalization. In: *Proceedings of the 5th European conference on computer vision—volume I, ECCV '98*, Springer, London, pp 475–490
22. Finlayson GD, Hordley SD, HubeL PM (2001) Color by correlation: a simple, unifying framework for color constancy. *IEEE Trans Pattern Anal Mach Intell* 23(11):1209–1221
23. Forsyth DA (1990) A novel algorithm for color constancy. *Int J Comput Vis* 5(1):5–36
24. Funt B, Barnard K, Martin L (1998) Is machine colour constancy good enough? In: *Proceedings of the 5th European conference on computer vision*. Springer, New York, pp 445–459
25. Gatta C, Rizzi A, Marini D (2002) Ace: an automatic color equalization algorithm. In: *Proceedings of CGIV2002 IS T*, Poitiers, France
26. Gehler PV, Rother C, Blake A, Minka T, Sharp T (2008) Bayesian color constancy revisited. In: *Proceedings of computer vision and pattern recognition (CVPR)*, pp 1–8
27. Geusebroek JM, Burghouts GJ, Smeulders AWM (2005) The Amsterdam library of object images. *Int J Comput Vis* 61(1):103–112
28. Gijsenij A, Gevers T, van de Weijer J (2011) Computational color constancy: survey and experiments. *IEEE Trans Image Process* 20(9):2475–2489
29. Gijsenij A, Gevers T, Weijer J (2010) Generalized gamut mapping using image derivative structures for color constancy. *Int J Comput Vis* 86(2–3):127–139
30. Judd DB, MacAdam DL, Wyszecki G, Budde HW, Condit HR, Henderson ST, Simonds JL (1964) Spectral distribution of typical daylight as a function of correlated color temperature. *J Opt Soc Am* 54(8):1031–1036
31. Kelly KL (1963) Lines of constant correlated color temperature based on MacAdam's  $(u,v)$  uniform chromaticity transformation of the CIE diagram. *J Opt Soc Am* 53(8):999–1002
32. Lam KM (1985) *Metamerism and Colour Constancy*. University of Bradford, England
33. Land EH (1977) The Retinex theory of color vision. *Sci Am* 237(6):108–128
34. Lecca M, Messelodi S (2009) Computing von Kries illuminant changes by piecewise inversion of cumulative color histograms. *Electron Lett Comput Vis Image Anal* 8(2):1–17
35. Lecca M, Messelodi S (2009) Illuminant change estimation via minimization of color histogram divergence. In: *Computational color imaging workshop—CCIW. LNCS, vol 5646/2009*. S. Etienne, France, pp 41–50

36. Lecca M, Messelodi S (2011) Linking the von Kries model to Wien's law for the estimation of an illuminant invariant image. *Pattern Recogn Lett* 32(15):2086–2096
37. Lecca M, Messelodi S (2011) Von Kries model under Planckian illuminants. In: 16th International conference on image analysis and processing—ICIAIP. Lecture notes in computer science, vol 6978. Ravenna, Italy, pp 296–305
38. Lecca M, Messelodi S, Andreatta C (2007) An object recognition system for automatic image annotation and browsing of object catalogs. In: Proceedings of the 15th international conference on multimedia, MULTIMEDIA '07. ACM, pp 154–155
39. Leichter I, Lindenbaum M, Rivlin E (2010) Mean shift tracking with multiple reference color histograms. *Comput Vis Image Underst* 114(3):400–408
40. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 60(2):91–110
41. Lu R, Gijssenij A, Gevers T, Nedovic V, Xu D, Geusebroek JM (2009) Color constancy using 3D scene geometry. In: Proceedings of ICCV, pp 1749–1756
42. Marimont DH, Wandell BA (1992) Linear models of surface and illuminant spectra. *J Opt Soc Am A* 9:1905–1913
43. McCamy CS (1992) Correlated color temperature as an explicit function of chromaticity coordinates. *Color Res Appl* 17(2):142–144
44. Nayar SK, Ikeuchi K, Kanade T (1991) Surface reflection: physical and geometrical perspectives. *IEEE Trans Pattern Anal Mach Intell* 13:611–634
45. Ojala T, Topi M, Pietikäinen M, Viertola J, Kyllönen J, Huovinen S (2002) Outex: new framework for empirical evaluation of texture analysis algorithms. In: Proceedings of ICPR '02, vol 1, IEEE Computer Society
46. Reinhard E, Adhikhmin M, Gooch B, Shirley P (2001) Color transfer between images. *IEEE Comput Graph Appl* 21(5):34–41
47. Rosenberg C, Minka T, Ladsariya A (2004) Bayesian color constancy with non-gaussian models. In: Thrun S, Saul L, Schölkopf B (eds) *Advances in neural information processing systems*, vol 16. MIT Press, Cambridge
48. Schaefer G (2011) Colour for image retrieval and image browsing. In: Proceedings of ELMAR, 2011, pp 1–3
49. Siang Tan K, Mat I, Nor A (2011) Color image segmentation using histogram thresholding: fuzzy c-means hybrid approach. *Pattern Recogn* 44(1):1–15
50. Skaff S, Arbel T, Clark JJ (2002) Active Bayesian color constancy with non-uniform sensors. In: Proceedings of 16th international conference on pattern recognition, vol 2, pp 681–684
51. Tominaga S, Ishida A, Wandell BA (2007) Color temperature estimation of scene illumination by the sensor correlation method. *Syst Comput Jpn* 38(8):95–108
52. Unnikrishnan R, Hebert M (2006) Extracting scale and illuminant invariant regions through color. In: Proceedings of BMVC, pp 52.1-52.10. doi:[10.5244/C.20.52](https://doi.org/10.5244/C.20.52)
53. van de Sande K, Gevers T, Snoek C (2010) Evaluating color descriptors for object and scene recognition. *IEEE Trans Pattern Anal Mach Intell* 32(9):1582–1596
54. van de Weijer J, Gevers T, Geusebroek J-M (2005) Edge and corner detection by photometric quasi-invariants. *IEEE Trans Pattern Anal Mach Intell* 27(4):625–630
55. van de Weijer J, Gevers T, Gijssenij A (2007) Edge-based color constancy. *IEEE Trans Image Process* 16(9):2207–2214
56. Vergés-Llahí J, Sanfeliu A (2005) A new color constancy algorithm based on the histogram of feasible mappings. In: Proceedings of the 2nd international conference on image analysis and recognition, ICIAR'05. Springer, Berlin, pp 720–728
57. Takahama K, Nayatani Y, Sobagaki H (1981) Formulation of a nonlinear model of chromatic adaptation. *Color Res Appl* 6(3):161–171



# Impulse and Mixed Multichannel Denoising Using Statistical Halfspace Depth Functions

Djordje Baljzović, Aleksandra Baljzović and Branko Kovačević

**Abstract** Although the statistical depth functions have been studied in nonparametric inference for multivariate data for more than a decade, the results of these studies have thus far been mostly theoretical. Out of numerous statistical depth functions, the halfspace depth function behaves very well overall in comparison with various competitors, and is one of the few statistical depth functions for which a small number of algorithms for computation in real Euclidean spaces have been proposed. In this chapter a new approach for removal of impulse and mixed multichannel noise based on a modified version of the only proposed algorithm for higher dimensional computation of the deepest location, i.e. a set of points with maximal halfspace depth, is discussed. A survey of experimental results shows that even in its baseline nonlinear spatial domain form, this filtering method gives excellent results in comparison to currently used state-of-the-art filters in elimination of wide range of powers of impulse and mixed multichannel noise from various benchmark image datasets. Multivariate nature of the implemented algorithm ensures the preservation of spectral correlation between channels and consequently, fine image details. Also, since the presented filter is independent of the source or distribution of the noise, it can be potentially used for removal of other types of multichannel noise.

**Keywords** Colour image processing · Colour image noise removal · Multichannel mixed noise · Multichannel impulse noise · Statistical depth functions · Halfspace depth function · Multivariate median

---

D. Baljzović (✉) · B. Kovačević  
School of Electrical Engineering, University of Belgrade,  
Bulevar kralja Aleksandra 73, 11120 Belgrade, Serbia  
e-mail: djordje.baljzovic@gmail.com

A. Baljzović  
School of Computing, Teesside University Middlesbrough,  
Tees Valley TS1 3BA, UK

## 1 Introduction

Since elimination of monochromatic and multichannel (colour) noise from digital images is still one of the most important issues in digital image processing, various filtering methods have been proposed in the literature so far, each of them having its own underlying principles, advantages, and drawbacks. Primary aim of all filters is to reduce the level of noise whilst preserving image details, edges, colours, and other features.

Noise on digital images is usually classified into three major categories, namely impulse, additive and multiplicative noise, depending on its origin which can be diverse. Impulse noise is mostly produced by defective pixels in camera sensors, faulty hardware memory locations, electrical disturbances and operation of high-voltage machinery corrupting signal transmissions; it corrupts a portion of image pixels and leaves the rest of the pixels intact [9, 13, 41]. Additive noise is mostly generated in operational amplifiers and their resistive circuits, and adds values drawn from a certain probability distribution (such as Gaussian) to every image pixel [13, 41]. Multiplicative noise power depends on the signal intensity which makes it very difficult to remove [13, 15].

Rapid development of multichannel digital image technology imposed the necessity of development of multichannel noise filters. Unlike monochromatic noise, colour channels on digital images corrupted by multichannel noise are (spectrally) correlated, which means that there is a high probability that an image incoherence from one channel appears in at least one of the remaining channels [13, 41]. As creating an all-purpose denoising filter for removal of various types and powers of multichannel noise from digital images has proven to be virtually impossible, the filters proposed in the literature thus far are usually classified into two major categories: spatial domain and transform domain filters [13, 41].

Transform domain filters are based on discrete cosine transform (DCT), and primarily discrete wavelet transform (DWT). DWT based filters are highly successful in removing (additive) Gaussian types of multichannel noise (e.g. additive colour Gaussian noise (ACGN)) as DWT provides good time frequency signal analysis, with a non-redundant signal representation and an optimal representation of singularities [11, 14, 42, 46, 51, 52]. However, they have very limited use in elimination of multichannel impulse types of noise [13], and suffer from significant drawbacks such as oscillations, aliasing, absence of phase information, shift-variance, and directional selectivity. In particular, shift-invariance and directionality issues influence the quality of noise removal, hence leading to good DWT filter performances only for low ACGN powers [51, 52].

Spatial domain based filters give considerably better results in removal of multichannel noise with smaller amount of artefacts. Despite this and some other advantages, they require more memory resources and are computationally more demanding than transform domain filters. Both monochromatic and multichannel spatial domain filters are classified into two general groups: linear and nonlinear filters [13]. Linear and majority of nonlinear filtering techniques are based on a kernel (mask or filtering

window)  $M$  of size  $w_{mask} \times h_{mask}$  which successively shifts through an image  $I$  of size  $W \times H$ . This can be expressed as follows [13, 15]:

$$result\_I(x, y) = \sum_{s = -\frac{w_{mask}}{2}}^{\frac{w_{mask}}{2}} \sum_{t = -\frac{h_{mask}}{2}}^{\frac{h_{mask}}{2}} M(s, t) * I(x + s, y + t) \quad (1)$$

Linear spatial domain filters preserve the edges and perform well in removal of monochromatic Gaussian type noises (e.g. white or random), and in edge detection in images corrupted by impulse noise [13, 15]. However, they become less effective for higher noise powers, reduce edge sharpness, blur the lines and fine details on an image, and are not efficient in removing the signal-dependent noise [13, 15]. This led to development of improved linear spatial domain filters like Wiener filters, although Wiener filters require both noise and signal spectrum information, and can be applied only to smooth underlying signals [13].

Nonlinear spatial domain filters overcome many of the linear filters' drawbacks, and have proven their competence in elimination of impulse, uniform and fixed valued noises, without need for their explicit identification [13]. Baseline median filter implements the filtering technique expressed by Eq. (1), and is still very broadly used due to its speed and tendency to preserve the edges under certain conditions [13]. Primarily, it is applied in removal of monochromatic impulse noise with fixed noise power values (e.g. salt-and-pepper). As Eq. (1) shows, median filter like many other spatial domain filters, affects every pixel in an image, both those corrupted and uncorrupted by noise; hence, the resulting images are often blurred and have untraceable edges [13]. Many improved nonlinear spatial domain filters have been proposed in order to improve these shortcomings [13, 41]. Such filters are, for example, centre weighted median (CWM) and switching median (SM) filters which are widely implemented in removal of monochromatic impulse noise with fixed values [5, 13], and filters like adaptive CWM, adaptive SM, and progressive switching median which have been introduced for removal of random-valued monochromatic impulse noise [5, 9].

With increasing need for multichannel denoising, numerous spatial domain nonlinear filters have been proposed mostly for elimination of multichannel impulse noise [5, 26]. Early stage filters like marginal (component-wise) median filter (MMF) often applied the method of scalar filtering of each channel individually [4, 5, 16, 26]. In this way, the inherent spectral correlation between channels is ignored which commonly causes the appearance of colour artefacts in resulting images. To date, this issue has been mainly resolved using numerous vector filtering techniques which essentially consider multichannel images as vector fields, and process multichannel pixels as vectors. They have been very successfully applied in elimination of multichannel impulse noise, and can be classified into eight categories [5]:

- basic vector filters
- adaptive fuzzy vector filters
- hybrid vector filters

- adaptive centre-weighted vector filters
- entropy vector filters
- peer group vector filters
- vector sigma filters
- miscellaneous (other) vector filters

However, basic, adaptive fuzzy, and hybrid vector filters are applied to all pixels in an image, even those not corrupted by noise; this may lead to disappearance of fine image details, blurred edges and overall excessive smoothness of resulting images [5, 13]. As a response to these issues, intelligent filters have been proposed which attempt to discern the pixels corrupted by noise from those that are not [4, 5]. Once the noisy pixels are identified, some of the vector filters (e.g. based on robust order statistics) are used; in other words, these filters switch between the identity operation and a vector filtering method, depending on some predetermined criteria. Consequently, they have reduced computational costs since a time-consuming filtering process is only executed on selected pixels observed as noisy [4, 5].

In contrast to the advancement of multichannel impulse and additive denoising filters, only a small number of filtering methods (both spatial and transform domain) have been suggested so far for removal of mixed multichannel noise [9, 50, 62].

A completely novel concept of multichannel filtering based on statistical depth functions, or more precisely, statistical halfspace depth function is discussed in this chapter. Resulting halfspace deepest location filter (HSDLF) offers outstanding results in elimination of impulse and mixed multichannel noise [2, 3]. Even though HSDLF belongs to the class of spatial domain filters, its underlying algorithm is substantially different to nonlinear vector (as well as other) impulse noise filters, and transform domain filters. HSDLF filtering method is based on an adjusted version of the DEEPLC algorithm [60] which calculates the approximate value of multivariate median (i.e. deepest location or the most central point within a data cloud) in  $d$ -dimensional space. Its multivariate/multidimensional character ensures that HSDLF intrinsically considers all channels in a multichannel image simultaneously, thus preserving the spectral correlation between channels.

HSDLF does not depend on the source and/or distribution of multichannel noise, and its efficiency in removal of:

- multichannel (colour) impulse noise in both of its common variations: salt-and-pepper noise (with fixed values of pixel noise) and random-valued multichannel noise (with arbitrary values of pixel noise) [2], and
- multichannel (colour) mixed noise, i.e. mixture of impulse (precisely, salt-and-pepper) and additive Gaussian multichannel noise (ACGN) [3],

has been verified.

Denoising results of HSDLF are compared in terms of objective error metrics (effectiveness) criteria and visual quality to:

- marginal median multichannel filter (MMF) [16] and 25 most significant state-of-the-art multichannel impulse noise filters [1, 4, 6–8, 16–24, 27–38, 40, 43–45, 54–59, 61] for impulse multichannel noise removal

- two state-of-the-art DWT based filters for removing multichannel Gaussian type noise: Probshrink [42] and BM3D (image denoising by sparse 3D transform-domain collaborative filtering) [11], as well as 26 filters for removal of multichannel impulse noise [1, 4, 6–8, 16–24, 27–38, 40, 43–45, 54–59, 61] for mixed multichannel noise removal

## 2 Halfspace Depth Function and Computation of Deepest Location for Multivariate Data

Finding the centre set of points within a multidimensional data cloud and generalisation of notion of median in multivariate case have been widely discussed in the literature [10, 12, 25, 39, 47–49, 53, 60, 63–65]. Since these notions cannot be directly derived from their univariate equivalents, several approaches have been introduced to address these issues with theory of statistical depth functions being the most prolific of them all. With their emerging popularity, many statistical depth functions have been formulated; Zuo and Serfling made a thorough overview and classification of all significant statistical depth functions [65]. HSDLF applies the most important representative entitled halfspace (*Tukey's* or location) depth function [25, 60, 63, 65].

### 2.1 Definition and Properties of Halfspace Depth Function

Associated with a given probability distribution  $P$  on  $d$ -dimensional real space  $\mathbb{R}^d$ , statistical depth functions provide a  $P$ -based centre-outward ordering (and thus ranking) of points (or more precisely, Borel sets)  $\theta \in \mathbb{R}^d$ . The essential example of statistical depth functions is the halfspace (*Tukey's* or location) depth function  $HD(\theta; P)$  which can be defined through following equation [25, 60, 65]:

$$HD(\theta; P) = \inf \{P(H) : \theta \in H \text{ closed halfspace}\}, \theta \in \mathbb{R}^d \quad (2)$$

In other words, halfspace depth (HD) function denotes the minimal probability attached to any closed halfspace with  $\theta$  on the boundary. In a discrete case, the halfspace depth function  $HD(\theta; X_n)$  of a point  $\theta \in \mathbb{R}^d$  relative to a data set  $X_n = \{x_1, x_2, \dots, x_n\} \in \mathbb{R}^{d \times n}$  represents the smallest number of observations (elements of  $X_n \in \mathbb{R}^{d \times n}$ ) in any closed halfspace with boundary through point  $\theta$  [63]. Multivariate halfspace depth can be also expressed using its univariate case  $HD_1(\theta; X_n)$  [60]:

$$HD_1(\theta; X_n) = HD(\theta; X_n) |_{d=1} = \min(\#\{x_i \leq \theta\}, \#\{x_i \geq \theta\}) \quad (3)$$

where  $\#$  is a number of observations.

This definition interprets the halfspace depth  $HD(\boldsymbol{\theta}; \mathbf{X}_n)$  as the smallest univariate halfspace depth value of a point  $\boldsymbol{\theta}$  relative to any projection of the data set  $\mathbf{X}_n$  onto a direction  $\mathbf{u}$  [60]:

$$\begin{aligned} HD(\boldsymbol{\theta}; \mathbf{X}_n) &= \min_{\|\mathbf{u}\|=1} HD_1(\mathbf{u}'\boldsymbol{\theta}; \mathbf{u}'\mathbf{X}_n) = \min_{\|\mathbf{u}\|=1} (\min(\#\{\mathbf{u}'\mathbf{x}_i \leq \mathbf{u}'\boldsymbol{\theta}\}, \#\{\mathbf{u}'\mathbf{x}_i \geq \mathbf{u}'\boldsymbol{\theta}\})) \\ &= \min_{\|\mathbf{u}\|=1} \#(i; \mathbf{u}'\mathbf{x}_i \leq \mathbf{u}'\boldsymbol{\theta}) \end{aligned} \quad (4)$$

It means that halfspace depth  $HD(\boldsymbol{\theta}; \mathbf{X}_n)$  indicates how deep a point  $\boldsymbol{\theta}$  is centred within the data cloud (set). Additionally, halfspace depth is defined as multivariate ranking: points nearer the boundary of the data set have lower ranks, and the rank increases as one gets deeper inside the data cloud [60]. Ranking can be geometrically illustrated using the notion of halfspace depth regions  $D_l$  defined by equation:

$$D_l = \left\{ \boldsymbol{\theta} \in \mathbb{R}^d; HD(\boldsymbol{\theta}; \mathbf{X}_n) \geq l \right\} \quad (5)$$

Halfspace depth regions are convex sets, and for every HD and  $l$ ,  $D_l \subseteq D_{l-1}$  holds [60]. Boundary of a region with equivalent HD values is called halfspace depth contour. Unique centre of gravity of the smallest halfspace depth region  $D_{l_{max}}$  which contains the points with maximal HD values (equal to  $l_{max}$ ) can be interpreted as multivariate median, i.e. a generalisation of univariate median to multidimensional spaces [60]. It also called Tukey's median or the deepest location.

Zuo and Serfling have shown that halfspace depth function  $HD(\boldsymbol{\theta}; P)$  fulfils all useful and desirable properties required of a statistical depth function [65]:

- *Affine invariance.*  $HD(\boldsymbol{\theta}; P)$  is independent of the coordinate system.
- *Maximality at center.* If  $P$  is symmetric about  $\boldsymbol{\theta}$  in some sense, then  $HD(\boldsymbol{\theta}; P)$  is maximal at this point.
- *Symmetry.* If  $P$  is symmetric about  $\boldsymbol{\theta}$  in some sense, then so is  $HD(\boldsymbol{\theta}; P)$ .
- *Decreasing along rays.* The depth  $HD(\boldsymbol{\theta}; P)$  decreases along every ray from the deepest point.
- *Vanishing at infinity.*  $HD(\boldsymbol{\theta}; P) \rightarrow 0$ ,  $\|\boldsymbol{\theta}\| \rightarrow \infty$ .
- *Continuity of  $HD(\boldsymbol{\theta}; P)$  as a function of  $\boldsymbol{\theta}$  (upper semicontinuity).*
- *Continuity as of  $HD(\boldsymbol{\theta}; P)$  a functional of  $P$ .*
- *Quasi-concavity as a function of  $\boldsymbol{\theta}$ .* The set  $\{\boldsymbol{\theta} : HD(\boldsymbol{\theta}; P) \geq c\}$  is convex for each real  $c$ .

Affine invariance is a particularly useful property for calculation of deepest location; in case of HD, this means that for any affine transformation  $g : \mathbb{R}^d \rightarrow \mathbb{R}^d : \mathbf{x} \rightarrow A \cdot \mathbf{x} + \mathbf{b}$ , with  $\mathbf{b} \in \mathbb{R}^d$  and  $A \in \mathbb{R}^{d \times d}$  a nonsingular matrix, equality  $HD(g(\boldsymbol{\theta}); g(\mathbf{X}_n)) = HD(\boldsymbol{\theta}; \mathbf{X}_n)$  holds. Consequently, the deepest location (Tukey's median)  $\mathbf{M}_{DL}^{HD}$  is also affine equivariant:  $\mathbf{M}_{DL}^{HD}(g(\mathbf{X}_n)) = g(\mathbf{M}_{DL}^{HD}(\mathbf{X}_n))$ .

Furthermore, Tukey's median is a very robust multivariate location estimator, and its robustness can be evaluated by means of the breakdown value  $\varepsilon^*$ . As proven by Donoho and Gasko [12], Tukey's median  $\mathbf{M}_{DL}^{HD}$  satisfies the inequality

$\varepsilon^*(\mathbf{M}_{DL}^{HD}; \mathbf{X}_n) \geq \frac{n}{d+1}$  for any sample in general position. This means that the resulting Tukey's median  $\mathbf{M}_{DL}^{HD}$  will not move outside of a bounded region if  $\frac{n}{d+1}$  observations are replaced from the data set  $\mathbf{X}_n$ . If the original data set  $\mathbf{X}_n$  comes from any angularly symmetric distribution, and especially from an elliptically symmetric distribution, the breakdown value  $\varepsilon^*(\mathbf{M}_{DL}^{HD}; \mathbf{X}_n)$  tends to  $\frac{1}{3}$  in any dimension. Simply put, if at least 67 % of the data points in  $\mathbf{X}_n$  are drawn from such a distribution then the deepest location remains within reasonable limits, regardless of the other points in the data set.

In general, not many algorithms have been proposed for calculation of statistical depth functions. A number of algorithms for computation of halfspace depth have been proposed in the literature [10, 47–49, 60, 64]; however, only a few algorithms for finding multivariate halfspace deepest location (Tukey's median) have been proposed: [48] for bivariate data, and, [10, 47] and [60] for dimensions more than two. DEEPLOC algorithm [60] has been selected as a basis of HSDLF, primarily because it implements the only mathematically proven and exact algorithm for calculation of halfspace depth functions in  $\mathbb{R}^3$  [49, 60]. Most importantly, it has been shown that approximate location depth calculated with DEEPLOC is very close to the exact depth for real data sets [60].

## 2.2 Finding the Deepest Location Using DEEPLOC Algorithm

DEEPLOC algorithm computes an approximation of deepest location (Tukey's median) for any dimension  $d$ . It works with a subset of directions  $\mathbf{u}$  [see Eq. (4)] to approximate the halfspace depth, and is constructed to be the least time consuming [49, 60]. DEEPLOC algorithm and its stepwise pseudocode are presented in their full (and complex) details in the original work of Struyf and Rousseeuw [60]. In this section only the most significant steps in DEEPLOC are explained which are of importance for understanding of how HSDLF works. Also, the places where the original DEEPLOC algorithm is altered are indicated.

In short, DEEPLOC starts from an initial point and takes further steps in carefully selected directions in which the halfspace depth can be increased (unlike the solution proposed in [10]), so that the deepest location (Tukey's median) can be approached after several of these steps. In order to reduce the number of steps, a centrally located initial point like coordinate-wise median or (affine invariant) coordinate-wise mean is taken, since it gives a fairly good halfspace depth relative to the data set, and is easy to compute [60]:

$$\mathbf{M}_1 = (\text{Med}(x_{i1}; i = 1, \dots, n), \dots, \text{Med}(x_{id}; i = 1, \dots, n)) \quad (6)$$

After calculating the starting point,  $m$  directions  $\mathbf{u} \in \mathbb{R}^d$  are constructed with  $\|\mathbf{u}\| = 1$ . These directions are randomly drawn from following classes [60]:

- (a)  $d$  coordinate axes
- (b) vectors connecting an observation with  $\mathbf{M}_1$
- (c) vectors connecting two observations
- (d) vectors perpendicular to a  $d$ -subset of observations

Classes (a), (b) and (c) are included as they are easy to compute, and are used for detection of marginal and far outliers [60]. However, class (d) directions are of greatest importance due to their close relation to halfspace depth notion, thus the majority of directions are taken from it. As a basis, HSDLF uses the proportion of directions from the original DEEPLOC algorithm [60]:

- all directions from the class (a)
- at most  $m/4$  directions from each of the classes (b) and (c)
- at least  $m/2$  directions from the class (d)

However, an additional parameter which controls the distribution of directions is introduced in HSDLF, and is called threshold control parameter (see Sect. 3). The overall number of directions  $m$  can be given by the user and in case of HSDLF,  $m=500$  directions are computed by default (see Sect. 3).

Univariate halfspace depth of  $\mathbf{M}_1$  relative to the projection of  $\mathbf{X}_n$  on each of these  $m$  directions is then computed. The directions  $\mathbf{u}$  which lead to the same lowest value  $\# \{i; \mathbf{u}'\mathbf{x}_i \leq \mathbf{u}'\mathbf{M}_1\}$  are stored in set  $U_{move}$ , and these directions are considered as the ones in which the halfspace depth is able to improve [60]. The average direction  $\mathbf{u}_{move}$  of directions saved in set  $U_{move}$  is then calculated:

$$\mathbf{u}_{move} = \frac{1}{|U_{move}|} \sum_{\mathbf{u} \in U_{move}} \mathbf{u}, \text{ where } \mathbf{u}_{move} \neq \mathbf{0} \quad (7)$$

After that, a step is taken from  $\mathbf{M}_1$  in the direction of  $\mathbf{u}_{move}$ . As mentioned, the value of halfspace depth at the deepest location relative to  $\mathbf{X}_n$  has to be at least  $\left\lfloor \frac{n}{d+1} \right\rfloor$  (where square brackets represent the floor function). If this condition is not fulfilled for  $\mathbf{M}_1$ , i.e. if following inequality holds:

$$HD_1(\mathbf{M}_1; \mathbf{u}'_{move}\mathbf{X}_n) < \left\lfloor \frac{n}{d+1} \right\rfloor \quad (8)$$

a step in the direction  $\mathbf{u}_{move}$  is taken, which is large enough to reach a point  $\mathbf{M}_2$  which has univariate halfspace depth value of  $\left\lfloor \frac{n}{d+1} \right\rfloor$ . Otherwise, a step is taken in the direction  $\mathbf{u}_{move}$  to the point  $\mathbf{M}_2$  which has univariate halfspace depth larger for 1 unit than the halfspace depth of  $\mathbf{M}_1$  in the same direction  $\mathbf{u}_{move}$  [60].

Afterwards, the explained procedure is repeated with  $\mathbf{M}_2$  (or  $\mathbf{M}_{next}$  in iteration for  $next > 2$ ) taking the roll of the initial point ( $\mathbf{M}_1$  in previously described algorithm steps) until a point  $\mathbf{M}_{maxHD}$  is found with maximal halfspace depth of  $\frac{n}{2}$ , or until the algorithm stops showing any halfspace depth improvement in previous  $N_{try}$  iteration steps. If the algorithm begins to oscillate instead of moving towards the deepest



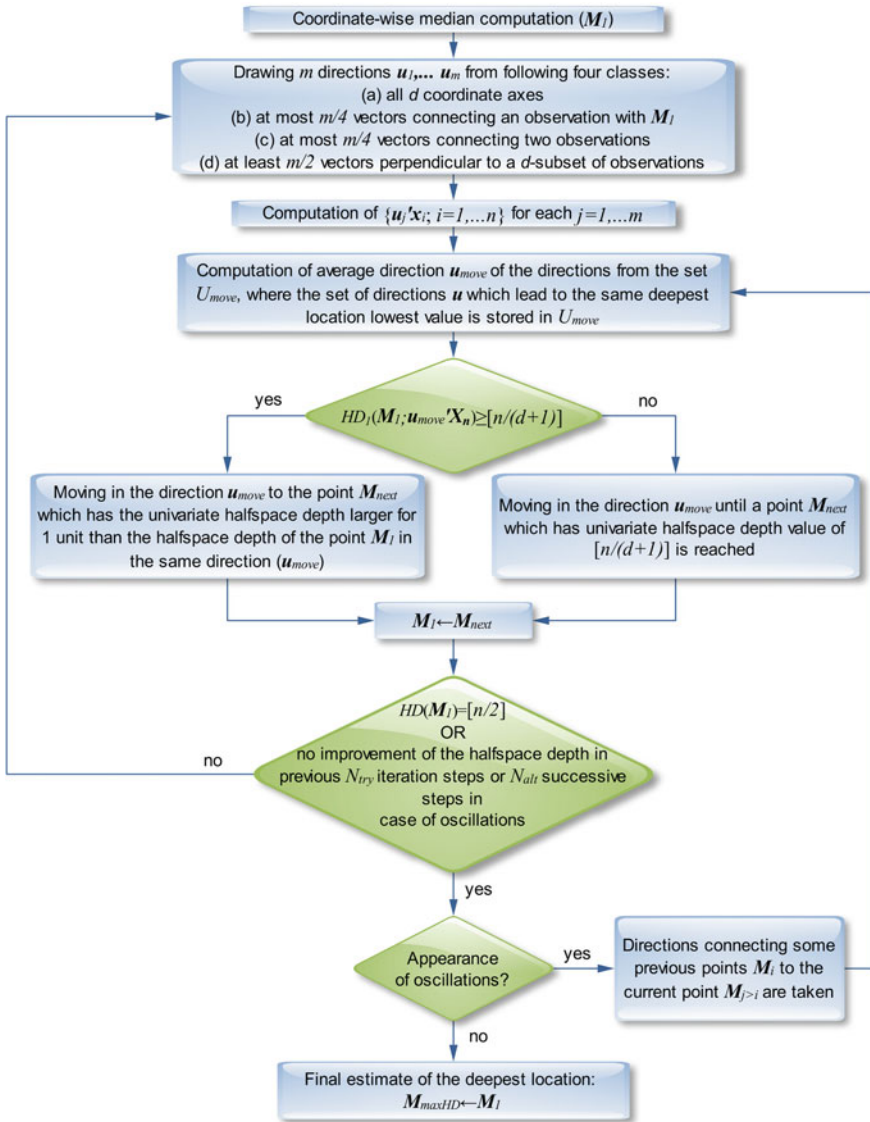


Fig. 1 Flowchart illustrating the most significant steps of the DEEPLOC algorithm

location, supplemental features are proposed for reduction of computational cost: if no halfspace depth improvement has been made after  $N_{alt}$  successive steps, then the directions connecting some previous points  $M_i$  to the current point  $M_{j>i}$  are considered [60]. It should be noted that parameters  $N_{try}$  and  $N_{alt}$  are also user defined.

Flowchart presented in Fig. 1 illustrates all relevant steps of the DEEPLOC algorithm. Detailed mathematical theorems and their proofs related to the DEEPLOC algorithm are given in [48, 49, 60].

### 3 Halfspace Deepest Location Filter

DEEPLC pseudocode served as a foundation of the HSDLF's programming code [2, 3]. During the experiments (see Sect. 4), it has been empirically established that a slight increase in number of directions from class (d) (defined in Sect. 2.2) improves filtering results. In order to give HSDLF an additional degree of freedom and even more flexibility, a scalar parameter entitled threshold control parameter  $\tau$  was introduced with values ranging from 0 to 1, which controls the percentage of class (d) directions<sup>1</sup>. As directions from class (d) play the vital role in DEEPLC algorithm, parameter  $\tau$  aims to provide fine-tuning of HSDLF's output results and precision. Experiments have shown that this small increase in number of class (d) directions has an insignificant impact on computational cost [2, 3].

HSDLF applies this adjusted version of the DEEPLC algorithm for calculating deepest location (Tukey's median) in three dimensional space ( $d=3$  in notation of Sects. 2.1 and 2.2), where R, G and B colour channels act as dimensions (coordinates). It was assumed that the multichannel images are 24-bit (8-bit per channel), so the values of R, G and B channels range from 0 to 255 for each channel.

HSDLF uses the standard spatial filtering technique based on a sliding convolution kernel (filtering window) described in Sect. 2 [2, 3]. Let  $r_{ij}$ ,  $g_{ij}$ , and  $b_{ij}$  denote the values of red, green, and blue channels, respectively, of a pixel at the position  $(i,j)$  in an image  $I$  of size  $W \times H$  contaminated by multichannel noise, i.e.  $I = \{(r_{ij}, g_{ij}, b_{ij}) | 1 \leq i \leq W, 1 \leq j \leq H\}$ . Convolution kernel  $M$  of size  $k \times k$ , where  $k$  is an odd positive integer, includes the pixel positioned at the centre of kernel  $M$ , i.e.  $(i,j)$  in the image  $I$ , and  $k^2 - 1$  of its neighbouring pixels. This means that the kernel  $M$  enfolds  $k^2$  ordered triplets of red, green and blue channel values, respectively, of corresponding pixels. HSDLF computes the deepest location within a data cloud which consists of these  $k^2$  three dimensional data using the adjusted DEEPLC algorithm, and replaces the channel values of pixel at the centre of kernel  $M$  ( $(i,j)$  in image  $I$ ) with estimated deepest location's channel values. Image edges are handled identically as in standard (marginal) median filters.

Number of directions is kept at a fixed value of  $m=500$  since this is the minimum value of  $m$  for which HSDLF gives excellent balance between high performance and computation time. It was shown that the algorithm accuracy gradually increases with an increase in number of directions  $m$ ; however, the computational cost grows rapidly [2, 3].

In early stages of HSDLF testing on 24-bit multichannel images, it was noted that deepest location channel values in very rare cases with far outliers can be negative (precisely, take value  $-1$ ), or can exceed the limit of 255 (precisely, take value of 256). In order to preserve the visual smoothness and consistency of filtered images without affecting the computational cost, these particular pixels are additionally filtered with MMF [5], which is very fast and calculated analogously to HSDLF. This correction

---

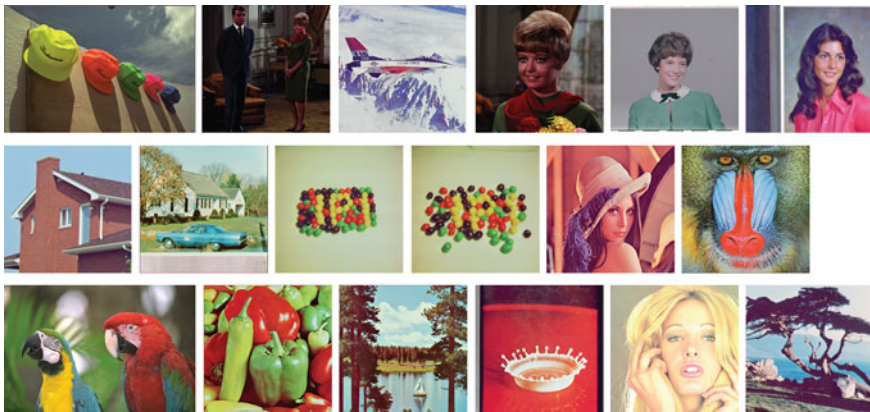
<sup>1</sup> The rest of the alterations in the algorithm are related to program code tweaking, and are not of interest for the scope of this chapter.

would not be needed if the number of directions  $m$  was set to value of more than 1,500 or if another colour system such as *CIE Lab* or *YCbCr* was used. However, in the latter case, a different range of channel values in these colour systems sometimes causes difficulties in calculation of location depth/deepest location.

It is important to point out that in case of the image set used in Sect. 4 for all observed types and/or powers of multichannel noise, and observed values of threshold control parameter  $\tau$ , the filtering results with and without this MMF correction are identical, and no cases of deepest location values going out of  $[0, 255]$  range appear on this image data set, even though the number of directions  $m$  is set to 500 [2, 3]. Also, HSDLF evidently does not depend on the nature or powers of applied multichannel noise; therefore, it can be used for elimination of many types of multichannel noise regardless of their origin [2, 3].

### 4 Experimental Results and Their Analysis

Performance of HSDLF has been tested on some of the most commonly used 24-bit multichannel (8-bit per colour channel) benchmark images: *Signal and Image Processing Institute (SIPI) Volume 3: Miscellaneous* image database set, with addition of two more images (“Parrots” and “Caps”) from the *Kodak Photo CD PCD0992* [2, 3]. These 24-bit (i.e. 8-bit-per-channel) multichannel benchmark images have fixed image sizes of  $256 \times 256$  pixels (“Couple”, “Girl1”, “Girl2”, “Girl3”, “House”, “Jellybeans1”, “Jellybeans2”, “Tree”),  $512 \times 512$  pixels (“F16”, “House With Car”, “Lena”, “Mandrill”, “Peppers”, “Sailboat”, “Splash”, “Tiffany”) and  $768 \times 512$  pixels (“Caps”, “Parrots”) (see Fig. 2).



**Fig. 2** Benchmark images used for comparison of all observed filters’ denoising performances, from left to right (image dimensions in pixels are given in brackets): Caps ( $768 \times 512$ ), Couple ( $256 \times 256$ ), F16 ( $512 \times 512$ ), Girl1 ( $256 \times 256$ ), Girl2 ( $256 \times 256$ ), Girl3 ( $256 \times 256$ ), House ( $256 \times 256$ ), House With Car ( $512 \times 512$ ), Jellybeans1 ( $256 \times 256$ ), Jellybeans2 ( $256 \times 256$ ), Lena ( $512 \times 512$ ), Mandrill ( $512 \times 512$ ), Parrots ( $768 \times 512$ ), Peppers ( $512 \times 512$ ), Sailboat ( $512 \times 512$ ), Splash ( $512 \times 512$ ), Tiffany ( $512 \times 512$ ), and Tree ( $256 \times 256$ )

However, due to very large amount of data, comparison of HSDLF's performance against 28 state-of-the-art filters will be presented for four essential benchmark images, namely "Lena" and "Peppers" (with sizes of  $512 \times 512$  pixels) from *Signal and Image Processing Institute (SIPI) Volume 3: Miscellaneous* image database, and "Parrots" and "Caps" (with sizes of  $768 \times 512$  pixels) from *Kodak Photo CD PCD0992*.

In all experiments, the convolution kernel (sliding filtering window) size is set to  $3 \times 3$  (see Sect. 3). HSDLF results have been produced using *Simple DirectMedia Layer* and *CImg* open source libraries within C++ programming language framework. Consequently, HSDLF does not require any specific digital image format, and can be successfully applied to nearly all digital image formats, including lossy compressed formats like JPEG. All types of noise have been generated using *MATLAB R2011a* software.

#### 4.1 Multichannel Impulse Noise Removal

As mentioned in Sect. 1, performances of HSDLF and other multichannel impulse noise filters are compared on images corrupted by salt-and-pepper as well as random-valued noise, with following power levels/densities applied:  $\xi = \{0.1, 0.2, 0.3, 0.4, 0.5\}$ .

Salt-and-pepper noise on image  $I$  is generated using *MATLAB* built-in *imnoise* ( $I, 'salt \& pepper', \xi$ ) function where  $\xi$  denotes the noise density. This means that approximately  $\xi \cdot \text{number of pixels}(I)$  randomly chosen pixels are replaced by pixels which have values of 0 or 255 on each channel (red, green and blue). Random-valued noise with density  $\xi$  on image  $I$  is generated indirectly since *MATLAB* does not have a built-in function for its production. Using the built-in *MATLAB* function *rand* for generation of uniformly distributed pseudorandom numbers,  $\xi \cdot \text{number of pixels}(I)$  pixels are selected randomly and then replaced by pixels with random values ranging from 0 to 255 on each colour channel (red, green and blue). Obviously, salt-and-pepper noise could be produced similarly using *MATLAB* function *rand*; experiments have shown that both of these methods for generating salt-and-pepper noise give identical results [2]. Salt-and-pepper and random-valued noise can be considered as instances of uncorrelated impulsive noise model [5]:

$$c_{ij}^{(ch)} = \begin{cases} r_{ij}^{(ch)} & \text{with probability } \xi \\ o_{ij}^{(ch)} & \text{with probability } 1-\xi \end{cases} \quad (9)$$

$$r_{ij}^{(ch)} \in \begin{cases} \{0, 255\} & \text{for salt-and-pepper noise} \\ [0, 255] & \text{for random-valued noise} \end{cases}$$

where  $c_{ij}^{(ch)}$  represents the pixel channel value (red, green or blue) of the output image corrupted by noise,  $o_{ij}^{(ch)}$  represents the pixel channel value (red, green or blue) of

the original image uncorrupted by noise, and  $r_{ij}^{(ch)}$  represents the pixel channel value (red, green or blue) of salt-and-pepper or random-valued noise;  $ch$  denotes the colour channel index:  $ch=1$  for red,  $ch=2$  green, and  $ch=3$  blue channel, and  $\xi$  symbolises the density of impulse noise that is applied to an image.

HSDLF performance in removal of multichannel impulse noise is compared to marginal median filter MMF [4] and following 25 state-of-the-art impulse noise filters [2, 5]:

- adaptive basic vector directional filter ABVDF [29]
- adaptive centre-weighted vector filters: ACWDDF [30], ACWVDF [38], ACWVMF [27]
- adaptive multichannel nonparametric filter with multivariate exponential kernel function AMNFE [43]
- adaptive vector sigma filters: ASBVDF [31–34, 37], ASDDF [31–34, 37], ASVMF [31–34, 37]
- adaptive vector median filter AVMF [28]
- basic vector directional filter BVDF [61]
- directional distance filter DDF [16, 17]
- entropy vector filters: EBVDF [35, 36], EDDF [35, 36], EVMF [35, 36]
- fast peer group filter FPGF [54]
- fuzzy vector median filter FVMF [8, 44, 45]
- fuzzy vector median-rational hybrid filter FVMRHF [21–23]
- kernel vector median filter KVMF [55–59]
- peer group filter PGF [18]
- order-statistics based switching vector filters: RSBVDF [7], RSDDF [7]
- robust switching vector median filter RSVMF [4]
- vector median filter VMF [1]
- vector median-rational hybrid filter VMRHF [19, 20, 24]
- vector signal-dependent rank order mean filter VSDROMF [40]

Filtering results are compared objectively by means of three error metrics criteria [5]:

- *peak signal-to-noise ratio* (PSNR) in decibels (dB) which measures the noise suppression capability of a filter:

$$PSNR = 10 \cdot \log_{10} \frac{255^2}{MSE}, \quad MSE = \frac{1}{3 \cdot W \cdot H} \sum_{ch=1}^3 \sum_{i=1}^W \sum_{j=1}^H (\hat{c}_{ij}^{(ch)} - c_{ij}^{(ch)})^2 \quad (10)$$

- *mean absolute error* (MAE) which measures the detail preservation capability of a filter:

$$MAE = \frac{1}{3 \cdot W \cdot H} \cdot \sum_{ch=1}^3 \sum_{i=1}^W \sum_{j=1}^H \left| \hat{c}_{ij}^{(ch)} - c_{ij}^{(ch)} \right| \quad (11)$$

where in Eqs. (10) and (11),  $\hat{c}_{ij}^{(ch)}$  and  $c_{ij}^{(ch)}$  represent the pixel channel values of denoised and original images, respectively, and  $ch$  symbolises the channel index:  $ch=1$  for red,  $ch=2$  green, and  $ch=3$  blue channel

- *normalized colour difference (NCD)* which measures the colour preservation capability of a filter:

$$NCD = \frac{\sum_{i=1}^W \sum_{j=1}^H \sqrt{\left( (\hat{L}_{ij} - L_{ij})^2 + (\hat{a}_{ij} - a_{ij})^2 + (\hat{b}_{ij} - b_{ij})^2 \right)}}{\sum_{i=1}^W \sum_{j=1}^H \sqrt{\left( (\hat{L}_{ij})^2 + (\hat{a}_{ij})^2 + (\hat{b}_{ij})^2 \right)}} \quad (12)$$

where  $\hat{L}_{ij}$  and  $L_{ij}$  represent lightness values, and  $(\hat{a}_{ij}, \hat{b}_{ij})$  and  $(a_{ij}, b_{ij})$  chrominance values of denoised and original images, respectively, expressed in *CIELAB* colour space [32]

Experimental results have shown that HSDLF provides optimal results in removal of multichannel impulse noise for two values of threshold control parameter  $\tau$  (0.0015 and 0.005) in terms of both objective error metrics criteria and visual quality [2]. Tested on *Signal and Image Processing Institute (SIPI) Volume 3: Miscellaneous* and *Kodak Photo CD PCD0992* image data sets, it has also been verified that HSDLF performances in multichannel impulse denoising deteriorate quickly for  $\tau > 0.006$ , and slightly for  $\tau < 0.0012$ .

Comparison of HSDLF performances in terms of used error metrics criteria for both observed values of the threshold control parameter  $\tau$ , calculated for all benchmark images is presented in Table 5. A detailed review of HSDLF performances for both observed types of impulse noise is given in Table 6.

Figures 3, 4 and 5 display the average PSNR, MAE and NCD gain values for all observed impulse denoising filters calculated over all benchmark images. Figures 6a–c, 7a–d, 8a–c, and 9a–d illustrate the visual quality of denoising results of all observed filters applied to image “Caps” corrupted by salt-and-pepper and random-valued multichannel noise, respectively.

The best filtering performances in Tables 1–4 are indicated by **boldface** font.

By closely observing the results given in Tables 1, 2, 3, 4, 5 and 6 and Figs. 3, 4, 5, 6a–c, 7a–d, 8a–c, and 9a–d, following conclusions can be drawn [2]:

- For all salt-and-pepper and random-valued impulse noise densities and calculated over all observed benchmark images, HSDLF performs better in more than 60% of cases for threshold control parameter value of  $\tau = 0.0015$  than for  $\tau = 0.005$  in terms of PSNR and MAE error metrics criteria, while NCD values are very similar for both values of  $\tau$ .
- HSDLF denoising results are marginally more sensitive to shifts in values of  $\tau$  in terms of all error metrics criteria when the filter is applied to images corrupted by random-valued impulse noise.
- PSNR, MAE and NCD gains steadily rise with the increase salt-and-pepper and random-valued noise powers for both HSDLF’s values of  $\tau$  on all observed images.

**Table 1** Performance results of observed multichannel impulse denoising filters applied to image “Caps” (with  $768 \times 512$  pixel image size) corrupted by various densities of salt-and-pepper and random-valued impulse noise. The best filtering performances are indicated by **boldface** font

Image: *Caps*  $768 \times 512$

	0.1			0.3			0.5		
	PSNR [dB]	MAE	NCD	PSNR [dB]	MAE	NCD	PSNR [dB]	MAE	NCD
None	17.66	21.36	0.392	12.89	42.83	0.708	10.74	57.86	0.907
ABVDF	18.09	18.23	0.321	12.39	44.38	0.691	9.95	63.48	0.917
ACWDDF	20.93	14.13	0.3	15.11	32.17	0.621	12.28	47.09	0.840
ACWVDF	18.33	18.05	0.332	12.7	42.67	0.692	10.25	60.81	0.917
ACWVMF	21.88	13.15	0.301	16.02	29.39	0.61	13.12	42.96	0.824
AMNFE	25.9	8.65	0.197	19.41	20.12	0.427	15.97	31.13	0.603
ASBVDF	18.31	17.66	0.312	12.91	41.93	0.686	10.56	58.69	0.905
ASDDF	20.72	14.14	0.29	14.26	35.69	0.64	11.52	51.99	0.865
ASVMF	22.5	12.37	0.282	16.16	29.18	0.594	13.01	43.71	0.816
AVMF	20.83	15.01	0.319	16.02	29.93	0.609	13.34	42.15	0.813
BVDF	18.2	17.62	0.279	12.39	44.48	0.668	9.97	63.49	0.902
DDF	24.43	9.6	0.213	17.27	24.96	0.525	13.75	39.58	0.753
EBVDF	17.95	18.56	0.332	12.68	42.98	0.701	10.37	59.96	0.921
EDDF	20.39	14.38	0.3	14.08	35.96	0.65	11.38	52.47	0.876
EVMF	22.36	12.56	0.29	16.25	28.83	0.599	13.25	42.54	0.819
FPGF	23.31	11.36	0.263	17.73	24.18	0.528	14.39	37.04	0.745
FVMF	25.55	8.74	0.2	18.77	21.27	0.471	15.14	33.81	0.681
FVMRHF	22.95	11.58	0.276	16.49	27.68	0.595	13.4	41.57	0.816
KVMF	24.59	9.56	0.223	17.91	23.55	0.517	14.42	36.84	0.742
MMF	25.26	8.98	0.202	19.13	20.77	0.435	15.7	31.83	0.625
PGF	22.88	11.77	0.275	17.24	25.38	0.548	14.1	38.21	0.76
RSBVDF	18.38	18.21	0.344	13.06	41.24	0.698	10.71	57.56	0.913
RSDDF	21.25	13.56	0.3	14.55	34.5	0.648	11.66	51.05	0.874
RSVMF	21.94	12.87	0.297	15.25	32.09	0.636	12.23	47.93	0.862
VMF	24.73	9.34	0.211	17.94	23.49	0.516	14.44	36.78	0.742
VMRHF	22.69	11.93	0.284	16.37	28.13	0.602	13.32	42	0.823
VSDROMF	24.21	10.14	0.235	17.88	23.7	0.52	14.43	36.83	0.742
HSDLF $\tau=0.005$	<b>27.29</b>	<b>7.58</b>	<b>0.157</b>	22.03	14.55	0.297	18.42	23.12	0.438
HSDLF $\tau=0.0015$	<b>27.29</b>	<b>7.58</b>	<b>0.157</b>	<b>22.06</b>	<b>14.49</b>	<b>0.296</b>	<b>18.46</b>	<b>22.99</b>	<b>0.437</b>

(continued)

**Table 1** (continued)

		Image: <i>Caps</i> 768 × 512								
		0.1			0.3			0.5		
	Random-valued noise density	PSNR	MAE	NCD	PSNR	MAE	NCD	PSNR	MAE	NCD
		[dB]			[dB]			[dB]		
	None	21.19	13.39	0.252	16.19	27.83	0.477	13.71	39.24	0.637
	ABVDF	21.6	11.72	0.211	15.83	28.53	0.463	13.12	42.11	0.636
	ACWDDF	23.56	9.99	0.203	17.82	22.93	0.431	14.83	34.27	0.597
	ACWVDF	21.71	11.71	0.217	16.04	27.84	0.464	13.31	41.03	0.636
	ACWVMF	24.7	9.07	0.201	18.95	20.38	0.419	15.84	30.58	0.579
	AMNFE	28.62	6.07	0.128	22.02	14.38	0.291	18.06	23.94	0.432
	ASBVDF	22.04	10.76	0.19	16.23	27.19	0.451	13.56	39.81	0.624
	ASDDF	24.53	8.81	0.182	17.72	22.98	0.421	14.52	35.39	0.595
	ASVMF	25.77	8.12	0.183	19.51	19.15	0.398	16.09	29.69	0.562
	AVMF	23.22	10.69	0.217	18.34	22.29	0.435	15.6	31.81	0.589
	BVDF	22.03	10.98	0.17	15.66	29.34	0.432	12.84	44.21	0.615
	DDF	27.42	6.6	0.136	20.46	16.88	0.347	16.67	27.7	0.513
	EBVDF	21.49	11.63	0.207	15.93	28.18	0.465	13.32	41.04	0.637
	EDDF	24.28	8.9	0.187	17.57	23.15	0.429	14.41	35.77	0.605
	EVMF	25.54	8.31	0.189	19.5	19.14	0.402	16.16	29.47	0.565
	FPGF	25.46	8.5	0.19	20.34	17.53	0.373	16.98	26.84	0.522
	FVMF	28.45	6.01	0.128	21.74	14.57	0.311	17.73	24.45	0.466
	FVMRHF	26.29	7.5	0.175	19.82	18.16	0.394	16.36	28.63	0.56
	KVMF	27.35	6.84	0.153	20.88	16.18	0.348	17.14	26.24	0.51
	MMF	28	6.32	0.134	21.71	14.95	0.299	17.9	24.32	0.439
	PGF	25.33	8.5	0.191	19.82	18.41	0.387	16.59	27.93	0.539
	RSBVDF	21.83	11.59	0.221	16.32	26.97	0.468	13.66	39.24	0.637
	RSDDF	24.86	8.63	0.191	18.09	22.09	0.429	14.77	34.37	0.603
	RSVMF	25.39	8.29	0.191	18.88	20.32	0.42	15.5	31.61	0.59
	VMF	27.63	6.44	0.135	20.91	16.12	0.345	17.15	26.26	0.51
	VMRHF	25.92	7.86	0.183	19.65	18.59	0.402	16.25	29.02	0.568
	VSDROMF	26.49	7.64	0.17	20.73	16.70	0.357	17.11	26.44	0.514
	HSDLF $\tau=0.005$	29.04	<b>5.86</b>	<b>0.112</b>	23.44	12.29	0.226	19.33	20.85	<b>0.342</b>
	HSDLF $\tau=0.0015$	<b>29.05</b>	<b>5.86</b>	<b>0.112</b>	<b>23.46</b>	<b>12.25</b>	<b>0.225</b>	<b>19.35</b>	<b>20.78</b>	<b>0.342</b>



**Table 2** Performance results of observed multichannel impulse denoising filters applied to image “Parrots” (with  $768 \times 512$  pixel image size) corrupted by various densities of salt-and-pepper and random-valued impulse noise. The best filtering performances are indicated by **boldface font**

Image: *Parrots*  $768 \times 512$

	Salt-and-pepper noise density	0.1			0.3			0.5		
		PSNR [dB]	MAE	NCD	PSNR [dB]	MAE	NCD	PSNR [dB]	MAE	NCD
None		17.51	21.61	0.341	12.72	43.45	0.624	10.53	58.84	0.809
ABVDF		18.28	17.66	0.273	12.59	43.13	0.594	10.03	62.44	0.799
ACWDDF		20.89	14	0.257	15.03	32.3	0.544	12.13	47.77	0.746
ACWVDF		18.43	17.69	0.283	12.86	41.7	0.598	10.29	60.15	0.803
ACWVMF		21.73	13.21	0.26	15.82	29.93	0.536	12.83	44.23	0.735
AMNFE		25.87	8.5	0.169	19.07	20.81	0.382	15.46	33.05	0.553
ASBVDF		18.55	17.01	0.263	12.99	41.31	0.594	10.53	58.51	0.795
ASDDF		20.68	14.05	0.25	14.28	35.44	0.557	11.41	52.37	0.764
ASVMF		22.36	12.4	0.243	15.98	29.67	0.522	12.75	44.84	0.728
AVMF		20.77	15.01	0.274	15.87	30.33	0.534	13.07	43.31	0.725
BVDF		18.77	16.07	0.23	12.78	42	0.565	10.14	61.63	0.78
DDF		24.49	9.29	0.181	17.14	25.31	0.464	13.52	40.62	0.674
EBVDF		18.2	17.92	0.283	12.8	42.18	0.607	10.38	59.59	0.809
EDDF		20.51	14.17	0.258	14.15	35.61	0.566	11.3	52.71	0.774
EVMF		22.23	12.59	0.251	16.04	29.39	0.528	12.95	43.82	0.731
FPGF		23.16	11.33	0.227	17.48	24.77	0.468	14.01	38.58	0.671
FVMF		25.54	8.53	0.171	18.48	21.87	0.418	14.72	35.41	0.617
FVMRHF		22.77	11.6	0.239	16.25	28.34	0.525	13.07	42.99	0.73
KVMF		24.53	9.4	0.191	17.64	24.14	0.459	14.05	38.37	0.668
MMF		25.21	8.81	0.174	18.8	21.47	0.389	15.22	33.69	0.572
PGF		22.92	11.58	0.233	17.06	25.8	0.482	13.77	39.57	0.682
RSBVDF		18.47	17.91	0.295	13.05	41.11	0.609	10.62	57.8	0.806
RSDDF		21.18	13.51	0.258	14.51	34.52	0.566	11.53	51.56	0.774
RSVMF		21.79	12.92	0.256	15.07	32.6	0.559	12	48.94	0.768
VMF		24.72	9.12	0.181	17.67	24.08	0.458	14.07	38.31	0.668
VMRHF		22.53	11.94	0.245	16.12	28.77	0.531	13	43.4	0.735
VSDROMF		24.21	9.94	0.2	17.62	24.29	0.461	14.06	38.37	0.668
HSDLF $\tau=0.005$		<b>27</b>	<b>7.64</b>	<b>0.14</b>	21.34	15.95	<b>0.283</b>	17.54	26.13	<b>0.428</b>
HSDLF $\tau=0.0015$		<b>27</b>	<b>7.64</b>	<b>0.14</b>	<b>21.36</b>	<b>15.89</b>	<b>0.283</b>	<b>17.57</b>	<b>26</b>	<b>0.428</b>

(continued)

**Table 2** (continued)

		Image: <i>Parrots</i> 768 × 512								
	Random-valued noise density	0.1			0.3			0.5		
		PSNR [dB]	MAE	NCD	PSNR [dB]	MAE	NCD	PSNR [dB]	MAE	NCD
Denoising method	None	20.85	13.99	0.226	15.78	29.34	0.436	13.28	41.48	0.59
	ABVDF	21.48	11.93	0.187	15.76	29	0.416	12.99	43.2	0.58
	ACWDDF	23.33	10.2	0.18	17.47	23.92	0.392	14.45	36.15	0.551
	ACWVDF	21.62	11.88	0.191	15.94	28.39	0.418	13.15	42.27	0.582
	ACWVMF	24.34	9.4	0.179	18.41	21.69	0.384	15.27	32.95	0.539
	AMNFE	28.46	6.07	0.115	21.29	15.78	0.275	17.27	27	0.418
	ASBVDF	22.12	10.72	0.166	16.11	27.76	0.405	13.33	41.27	0.571
	ASDDF	24.24	9.05	0.162	17.46	23.84	0.382	14.2	37.13	0.549
	ASVMF	25.48	8.31	0.162	18.97	20.39	0.365	15.51	32.04	0.524
	AVMF	22.91	11.06	0.193	17.89	23.45	0.396	15.09	33.93	0.546
	BVDF	22.46	10.31	0.146	15.98	28.28	0.38	12.93	43.99	0.553
	DDF	27.35	6.54	0.12	20	17.92	0.321	16.14	30.08	0.483
	EBVDF	21.69	11.46	0.182	15.88	28.58	0.418	13.16	42.24	0.583
	EDDF	24.08	9.13	0.168	17.4	23.89	0.39	14.15	37.37	0.557
	EVMF	25.22	8.53	0.169	18.93	20.44	0.37	15.56	31.94	0.528
	FPGF	25.15	8.75	0.168	19.79	18.65	0.342	16.33	29.35	0.49
	FVMF	28.31	<b>5.98</b>	0.114	21.06	15.79	0.291	16.99	27.24	0.446
	FVMRHF	25.91	7.74	0.157	19.2	19.53	0.364	15.72	31.23	0.525
	KVMF	27.07	6.93	0.136	20.27	17.38	0.323	16.46	28.88	0.482
	MMF	27.84	6.31	0.119	21.02	16.28	0.282	17.14	27.28	0.425
PGF	25.01	8.75	0.169	19.27	19.56	0.355	15.99	30.29	0.504	
RSBVDF	21.78	11.71	0.195	16.09	27.9	0.424	13.36	40.96	0.586	
RSDDF	24.56	8.86	0.17	17.73	23.14	0.391	14.4	36.26	0.557	
RSVMF	25.03	8.58	0.171	18.32	21.68	0.385	14.94	33.98	0.549	
VMF	27.5	6.44	0.121	20.3	17.32	0.32	16.47	28.91	0.482	
VMRHF	25.56	8.1	0.164	19.04	19.95	0.37	15.63	31.59	0.531	
VSDROMF	26.3	7.72	0.149	20.15	17.84	0.329	16.44	29.05	0.485	
HSDLF $\tau=0.005$	28.67	6.03	<b>0.104</b>	22.35	14.37	0.231	18.24	24.83	<b>0.358</b>	
HSDLF $\tau=0.0015$	<b>28.68</b>	6.02	<b>0.104</b>	<b>22.36</b>	<b>14.34</b>	<b>0.23</b>	<b>18.25</b>	<b>24.77</b>	<b>0.358</b>	

**Table 3** Performance results of observed multichannel impulse denoising filters applied to image “Lena” (with  $512 \times 512$  pixel image size) corrupted by various densities of salt-and-pepper and random-valued impulse noise. The best filtering performances are indicated by **boldface font**

Image: *Lena*  $512 \times 512$

	Salt-and-pepper noise density	0.1			0.3			0.5		
		PSNR [dB]	MAE	NCD	PSNR [dB]	MAE	NCD	PSNR [dB]	MAE	NCD
None		17.43	21.86	0.295	12.73	43.45	0.543	10.57	58.63	0.707
ABVDF		19.46	16.49	0.239	13.6	37.91	0.498	11.01	54.52	0.677
ACWDDF		20.96	14.57	0.233	15.27	31.63	0.474	12.45	45.96	0.647
ACWVDF		19.2	16.98	0.249	13.63	37.89	0.507	11.1	53.99	0.685
ACWVMF		21.76	13.84	0.233	15.91	29.85	0.464	12.93	43.73	0.635
AMNFE		25.66	9.23	0.156	19.23	20.73	0.333	15.66	32.29	0.485
ASBVDF		19.74	15.65	0.226	13.61	38.19	0.503	11.1	54.36	0.684
ASDDF		21.27	13.97	0.222	14.73	33.66	0.476	11.83	49.68	0.657
ASVMF		22.27	13.17	0.222	16.13	29.38	0.453	12.87	44.14	0.627
AVMF		21.11	15	0.241	16.03	29.98	0.461	13.2	42.7	0.623
BVDF		20.8	14.06	0.196	14.2	34.9	0.458	11.28	52.43	0.65
DDF		24.12	10.54	0.176	17.4	25.16	0.405	13.9	38.93	0.583
EBVDF		19.23	16.6	0.243	13.47	38.69	0.514	11.05	54.53	0.692
EDDF		21.46	13.86	0.228	14.84	33.09	0.483	11.87	49.2	0.667
EVMF		22.24	13.2	0.226	16.19	29.15	0.456	13.09	43.14	0.63
FPGF		23.04	12.16	0.207	17.57	24.94	0.405	14.2	37.84	0.575
FVMF		25.29	9.42	0.16	18.63	21.92	0.361	14.93	34.64	0.53
FVMRHF		22.75	12.28	0.215	16.34	28.28	0.454	13.18	42.42	0.631
KVMF		24.34	10.24	0.177	17.78	24.2	0.397	14.26	37.55	0.572
MMF		25.08	9.64	0.162	19	21.3	0.337	15.49	32.67	0.489
PGF		22.4	12.89	0.218	16.94	26.5	0.423	13.82	39.37	0.589
RSBVDF		18.86	17.59	0.259	13.43	39.23	0.525	11.07	54.66	0.697
RSDDF		21.41	13.93	0.232	14.79	33.47	0.49	11.85	49.53	0.671
RSVMF		21.7	13.71	0.232	15.09	32.65	0.486	12.02	48.74	0.665
VMF		24.42	10.22	0.172	17.81	24.15	0.396	14.27	37.49	0.572
VMRHF		22.5	12.64	0.221	16.22	28.7	0.46	13.1	42.85	0.635
VSDROMF		23.92	10.96	0.187	17.75	24.4	0.399	14.26	37.57	0.572
HSDLF $\tau=0.005$		<b>26.33</b>	<b>8.78</b>	<b>0.135</b>	<b>21.14</b>	<b>16.9</b>	<b>0.252</b>	<b>17.63</b>	26.15	<b>0.371</b>
HSDLF $\tau=0.0015$		<b>26.33</b>	<b>8.78</b>	<b>0.135</b>	<b>21.14</b>	<b>16.9</b>	<b>0.252</b>	17.62	<b>26.14</b>	0.372

(continued)

**Table 3** (continued)

		Image: <i>Caps</i> 512 × 512								
	Random-valued noise density	0.1			0.3			0.5		
		PSNR [dB]	MAE	NCD	PSNR [dB]	MAE	NCD	PSNR [dB]	MAE	NCD
Denoising method	None	20.65	14.49	0.199	15.78	29.53	0.382	13.36	41.26	0.524
	ABVDF	22.01	12.11	0.171	16.31	27.31	0.358	13.55	40.07	0.507
	ACWDDF	23.31	10.78	0.164	17.72	23.49	0.342	14.75	34.91	0.487
	ACWVDF	22.07	11.98	0.172	16.46	26.74	0.359	13.69	39.29	0.508
	ACWVMF	24.42	9.89	0.162	18.67	21.45	0.332	15.52	32.16	0.474
	AMNFE	<b>28.08</b>	6.99	0.112	21.65	15.65	0.245	17.66	25.83	0.381
	ASBVDF	23.04	10.54	0.15	16.72	25.86	0.344	13.8	38.81	0.499
	ASDDF	24.55	9.47	0.15	17.93	22.81	0.329	14.61	35.35	0.481
	ASVMF	25.27	9.12	0.152	19.2	20.34	0.319	15.73	31.44	0.462
	AVMF	22.99	11.45	0.173	18.1	23.19	0.342	15.32	33.16	0.479
	BVDF	23.77	10.17	0.135	17.39	23.92	0.313	14.01	37.76	0.471
	DDF	26.79	7.78	0.122	20.26	18	0.286	16.52	28.87	0.433
	EBVDF	22.58	11.12	0.162	16.56	26.36	0.355	13.74	39.1	0.507
	EDDF	24.64	9.5	0.154	18.18	22.24	0.332	14.77	34.69	0.485
	EVMF	25.12	9.24	0.155	19.19	20.31	0.321	15.83	31.15	0.464
	FPGF	25.34	9.17	0.152	20.15	18.44	0.296	16.67	28.44	0.431
	FVMF	27.93	<b>6.97</b>	0.112	21.43	15.83	0.255	17.37	26.28	0.397
	FVMRHF	25.8	8.5	0.145	19.48	19.4	0.316	16.01	30.36	0.463
	KVMF	26.96	7.6	0.127	20.59	17.36	0.281	16.8	27.99	0.426
	MMF	27.5	7.31	0.117	21.4	16.11	0.248	17.55	26.07	0.38
	PGF	25.01	9.37	0.155	19.56	19.44	0.307	16.27	29.52	0.443
	RSBVDF	22.07	11.84	0.175	16.38	26.99	0.367	13.73	39.16	0.515
	RSDDF	24.6	9.49	0.157	18.07	22.51	0.339	14.74	34.81	0.49
	RSVMF	24.9	9.33	0.157	18.52	21.6	0.335	15.13	33.43	0.484
VMF	27.05	7.58	0.121	20.61	17.36	0.28	16.81	28	0.426	
VMRHF	25.46	8.84	0.151	19.31	19.8	0.321	15.91	30.73	0.467	
VSDROMF	26.21	8.37	0.139	20.46	17.77	0.286	16.78	28.12	0.427	
HSDLF $\tau=0.005$	27.9	7.33	<b>0.107</b>	<b>22.32</b>	15	<b>0.211</b>	<b>18.43</b>	24.36	<b>0.332</b>	
HSDLF $\tau=0.0015$	27.9	7.33	<b>0.107</b>	<b>22.32</b>	<b>14.99</b>	<b>0.211</b>	<b>18.43</b>	<b>24.35</b>	<b>0.332</b>	

**Table 4** Performance results of observed multichannel impulse denoising filters applied to image “Peppers” (with  $512 \times 512$  pixel image size) corrupted by various densities of salt-and-pepper and random-valued impulse noise. The best filtering performances are indicated by **boldface font**

Image: *Peppers*  $512 \times 512$

	Salt-and-pepper noise density	0.1			0.3			0.5		
		PSNR [dB]	MAE	NCD	PSNR [dB]	MAE	NCD	PSNR [dB]	MAE	NCD
None		17.19	22.1	0.261	12.48	43.91	0.494	10.25	59.77	0.667
ABVDF		19.57	16.25	0.206	13.43	38.24	0.443	10.38	58.05	0.63
ACWDDF		20.86	14.74	0.205	15.08	32.1	0.425	12.04	47.66	0.604
ACWVDF		19.45	16.55	0.216	13.48	38.06	0.452	10.54	56.83	0.636
ACWVMF		21.26	14.31	0.206	15.64	30.48	0.417	12.6	44.86	0.594
AMNFE		25.06	9.75	0.142	18.71	21.87	0.311	15.1	34.54	0.474
ASBVDF		19.74	15.81	0.201	13.48	38.26	0.449	10.61	56.65	0.637
ASDDF		21.07	14.38	0.197	14.68	33.67	0.424	11.5	50.82	0.61
ASVMF		21.8	13.58	0.195	15.8	30.23	0.407	12.58	45.12	0.586
AVMF		20.48	15.86	0.215	15.74	30.72	0.413	12.91	43.64	0.579
BVDF		20.66	14.16	0.175	13.84	36.34	0.415	10.6	56.53	0.609
DDF		23.63	10.94	0.158	16.96	26.2	0.369	13.35	41.31	0.552
EBVDF		19.09	16.9	0.217	13.14	39.7	0.464	10.44	57.82	0.648
EDDF		20.84	14.65	0.205	14.51	34.16	0.435	11.36	51.63	0.621
EVMF		21.76	13.65	0.2	15.91	29.83	0.41	12.77	44.28	0.588
FPGF		22.7	12.34	0.182	17.26	25.54	0.366	13.81	39.28	0.542
FVMF		24.73	9.85	0.145	18.23	22.72	0.33	14.5	36.21	0.504
FVMRHF		22.24	12.72	0.193	16.06	28.86	0.411	12.81	43.8	0.593
KVMF		23.82	10.63	0.159	17.44	24.94	0.360	13.86	39.08	0.541
MMF		24.61	10.02	0.145	18.55	22.22	0.311	14.98	34.54	0.47
PGF		22.15	12.98	0.191	16.74	26.77	0.379	13.51	40.37	0.552
RSBVDF		19.14	17.12	0.225	13.3	39.13	0.468	10.64	56.37	0.648
RSDDF		21.1	14.24	0.204	14.65	33.66	0.437	11.49	50.77	0.622
RSVMF		21.31	14.06	0.205	14.94	32.81	0.435	11.76	49.42	0.622
VMF		23.88	10.7	0.156	17.46	24.91	0.36	13.87	39.04	0.541
VMRHF		22.02	13.06	0.198	15.94	29.28	0.415	12.74	44.19	0.597
VSDROMF		23.55	11.14	0.165	17.41	25.08	0.362	13.86	39.09	0.541
HSDLF $\tau=0.005$		<b>25.99</b>	9.07	<b>0.126</b>	20.59	18.03	<b>0.249</b>	17.02	28.58	<b>0.384</b>
HSDLF $\tau=0.0015$		<b>25.99</b>	<b>9.06</b>	<b>0.126</b>	<b>20.61</b>	<b>17.99</b>	<b>0.249</b>	<b>17.04</b>	<b>28.52</b>	<b>0.384</b>

(continued)

**Table 4** (continued)

		Image: <i>Caps</i> 512 × 512								
	Random-valued noise density	0.1			0.3			0.5		
		PSNR [dB]	MAE	NCD	PSNR [dB]	MAE	NCD	PSNR [dB]	MAE	NCD
Denoising method	None	20.65	14.49	0.199	15.78	29.53	0.382	13.36	41.26	0.524
	ABVDF	21.95	12.24	0.157	15.91	28.45	0.341	12.78	43.68	0.503
	ACWDDF	23.17	11.14	0.154	17.32	24.6	0.328	14.13	37.59	0.485
	ACWVDF	22.02	12.15	0.16	15.99	28.07	0.345	12.95	42.67	0.505
	ACWVMF	23.63	10.76	0.155	17.98	23.08	0.322	14.8	34.97	0.474
	AMNFE	27.07	7.84	0.111	20.61	17.81	0.252	16.67	29.64	0.401
	ASBVDF	22.73	11.09	0.144	16.18	27.42	0.333	13.06	42.13	0.497
	ASDDF	23.86	10.3	0.144	17.36	24.32	0.319	13.9	38.26	0.48
	ASVMF	24.41	9.94	0.145	18.4	22.07	0.309	14.98	34.28	0.464
	AVMF	22.55	12.1	0.163	17.63	24.36	0.328	14.72	35.58	0.476
	BVDF	23.4	10.56	0.131	16.7	25.94	0.308	13.21	41.54	0.471
	DDF	25.8	8.63	0.121	19.3	19.99	0.283	15.55	32.51	0.441
	EBVDF	22.12	11.81	0.156	15.82	28.54	0.347	12.89	42.99	0.507
	EDDF	23.66	10.48	0.15	17.25	24.59	0.327	13.87	38.5	0.488
	EVMF	24.29	10.05	0.148	18.4	22.04	0.312	15.06	34.09	0.466
	FPGF	24.5	9.97	0.144	19.29	20.13	0.289	15.82	31.55	0.437
	FVMF	26.93	<b>7.77</b>	0.111	20.43	17.71	0.257	16.43	29.68	0.411
	FVMRHF	24.89	9.34	0.141	18.65	21.21	0.309	15.19	33.51	0.467
	KVMF	26.01	8.36	0.123	19.69	19.12	0.278	15.93	31.18	0.434
	MMF	26.69	8.02	0.114	20.43	18.04	0.252	16.57	29.66	0.399
PGF	24.18	10.18	0.148	18.82	21	0.299	15.5	32.43	0.446	
RSBVDF	21.99	12.11	0.163	15.89	28.34	0.353	13.04	42.14	0.511	
RSDDF	23.9	10.25	0.149	17.4	24.17	0.329	13.99	37.87	0.489	
RSVMF	24.09	10.17	0.15	17.81	23.27	0.324	14.42	36.21	0.483	
VMF	26.09	8.41	0.12	19.71	19.14	0.277	15.93	31.21	0.433	
VMRHF	24.61	9.66	0.146	18.51	21.58	0.314	15.11	33.83	0.47	
VSDROMF	25.43	9.04	0.132	19.59	19.47	0.281	15.91	31.3	0.435	
HSDLF $\tau=0.005$	27.41	7.79	<b>0.105</b>	21.51	16.88	<b>0.226</b>	17.51	28.15	<b>0.363</b>	
HSDLF $\tau=0.0015$	<b>27.42</b>	<b>7.77</b>	<b>0.105</b>	<b>21.53</b>	<b>16.85</b>	<b>0.226</b>	<b>17.52</b>	<b>28.11</b>	<b>0.363</b>	

- For both salt-and-pepper and random-valued impulse noise, HSDLF reaches maximum PSNR, MAE and NCD gains for both values of  $\tau$  on image “Caps”, and minimum PSNR and NCD gains on image “Peppers”. Minimum MAE gain is attained on image “Peppers” corrupted by salt-and-pepper noise, and “Lena” corrupted by random-valued noise. All minimum error metrics parameters’ gains appear for the lowest noise power of both types of impulse noise ( $\xi = 0.1$ )<sup>2</sup>.

<sup>2</sup> Unlike positive PSNR gains, MAE and NCD gains have negative values; still, the term gain (not loss) is used for MAE and NCD since their larger absolute values indicate better denoising performance results of a filter.

**Table 5** General comparison of error metrics criteria (PSNR, MAE and NCD) for HSDLF calculated for all used benchmark images, and all salt-and-pepper and random-valued noise densities

Impulse noise type	$\tau$	Effectiveness criterion					
		PSNR		MAE		NCD	
		Better (%)	Ties (%)	Better (%)	Ties (%)	Better (%)	Ties (%)
Salt-and-pepper	0.0015	60	35	70	30	25	65
	0.005	5		0		10	
Random-valued	0.0015	75	25	90	10	20	70
	0.005	0		0		10	

$\tau$  symbolises HSDLF threshold control parameter

- HSDLF offers slightly better denoising results for  $\tau=0.005$  in terms of visual quality for images with lower contrasts and fewer details.
- HSDLF outperforms all other observed filters for both values of  $\tau$  in terms of average PSNR, MAE and NCD gains, as shown in Figs. 3, 4 and 5. As noted, average gains are calculated over all benchmark images and for all observed impulse noise densities, whilst maximal and minimal gains are calculated for each of the benchmark images individually.
- In terms of PSNR, HSDLF gives better denoising results than all other filters for all random-valued noise densities with the exception of image “Lena” corrupted by the lowest observed noise density  $\xi=0.1$ , where AMNFE and FVMF give slightly better results.
- HSDLF also outperforms all other filters for all random-valued noise densities in terms of MAE, with the exemption of images “Lena” corrupted by the lowest observed noise density  $\xi=0.1$ , where AMNFE and FVMF give marginally better results, and “Parrots” corrupted by the lowest observed noise density  $\xi=0.1$ , where FVMF provides very slightly better results. Even in these scarce exceptions, the differences in PSNR and MAE values between HSDLF and AMNFE/FVMF are negligible as presented in Tables 1, 2, 3 and 4.
- In terms of NCD, HSDLF steadily outperforms all other observed impulse noise filters for all random-valued noise densities.
- HSDLF consistently outperforms all other compared denoising filters for all densities of salt-and-pepper noise in terms of all error metrics criteria for both values of parameter  $\tau$ .
- As evidenced in Figs. 3, 4 and 5, HSDLF is particularly effective in elimination of both observed types of impulse noise with medium and heavy powers as it is the only filter whose performances do not decline for noise densities above  $\xi=0.1$ .
- Figures 6a–c, 7a–d, 8a–c and 9a–d clearly show that HSDLF tends to successfully preserve all image edges and details, and that it has considerably less artefacts than other compared impulse noise filters for both observed values of  $\tau$  and types of multichannel impulse noise.

**Table 6** Detailed HSDLF performance summary in terms of error metrics criteria (PSNR, MAE and NCD). Overall average error metrics’ gains are calculated over “Lena”, “Peppers”, “Parrots” and “Caps” benchmark images and for all observed impulse noise densities; maximal and minimal error metrics’ gains are calculated for each of the benchmark images individually

HSDLF threshold		Total AG	Max. AG	Min. AG	Max. G	Min. G			
control parameter $\tau$									
Salt-and-pepper noise	PSNR	0.005	+8.39	+8.93 dB	+7.97 dB	+9.7 dB	+6.77 dB		
				<i>Caps</i>	<i>Peppers</i>	<i>Caps</i>	<i>Peppers</i>		
				768 × 512	512 × 512	768 × 512	512 × 512	S&P ND: 0.2	S&P ND: 0.5
		0.0015	+8.4	+8.95 dB	+7.99 dB	+9.71 dB	+6.79 dB		
				<i>Caps</i>	<i>Peppers</i>	<i>Caps</i>	<i>Peppers</i>		
				768 × 512	512 × 512	768 × 512	512 × 512	S&P ND: 0.2	S&P ND: 0.5
	MAE	0.005	−25.1	−26.3	−24.02	−34.74	−13.03		
				<i>Caps</i>	<i>Peppers</i>	<i>Caps</i>	<i>Peppers</i>		
				768 × 512	512 × 512	768 × 512	512 × 512	S&P ND: 0.5	S&P ND: 0.1
		0.0015	−25.14	−26.36	−24.06	−34.87	−13.04		
				<i>Caps</i>	<i>Peppers</i>	<i>Caps</i>	<i>Peppers</i>		
				768 × 512	512 × 512	768 × 512	512 × 512	S&P ND: 0.5	S&P ND: 0.1
NCD	0.005	−0.298	−0.382	−0.227	−0.469	−0.135			
			<i>Caps</i>	<i>Peppers</i>	<i>Caps</i>	<i>Peppers</i>			
			768 × 512	512 × 512	768 × 512	512 × 512	S&P ND: 0.5	S&P ND: 0.1	
	0.0015	−0.299	−0.382	−0.227	−0.47	−0.135			
			<i>Caps</i>	<i>Peppers</i>	<i>Caps</i>	<i>Peppers</i>			
			768 × 512	512 × 512	768 × 512	512 × 512	S&P ND: 0.5	S&P ND: 0.1	

(Continued)

## 4.2 Multichannel Mixed Noise Removal

Performances of HSDLF and other multichannel denoising filters in removal of mixed multichannel noise are compared on images corrupted by mixture of salt-and-pepper impulse noise and additive Gaussian colour noise (ACGN), with wide range of noise powers, i.e. densities of impulse noise and variances of ACGN (see Tables 7, 8, 9 and 10). It is considered that ACGN distribution has zero means and known variances for each colour channel, that is  $n_{r,g,b}^{ACGN} \sim \mathcal{N}(0, \sigma_{r,g,b}^2)$ . Filtering results for mixed



**Table 6** (continued)

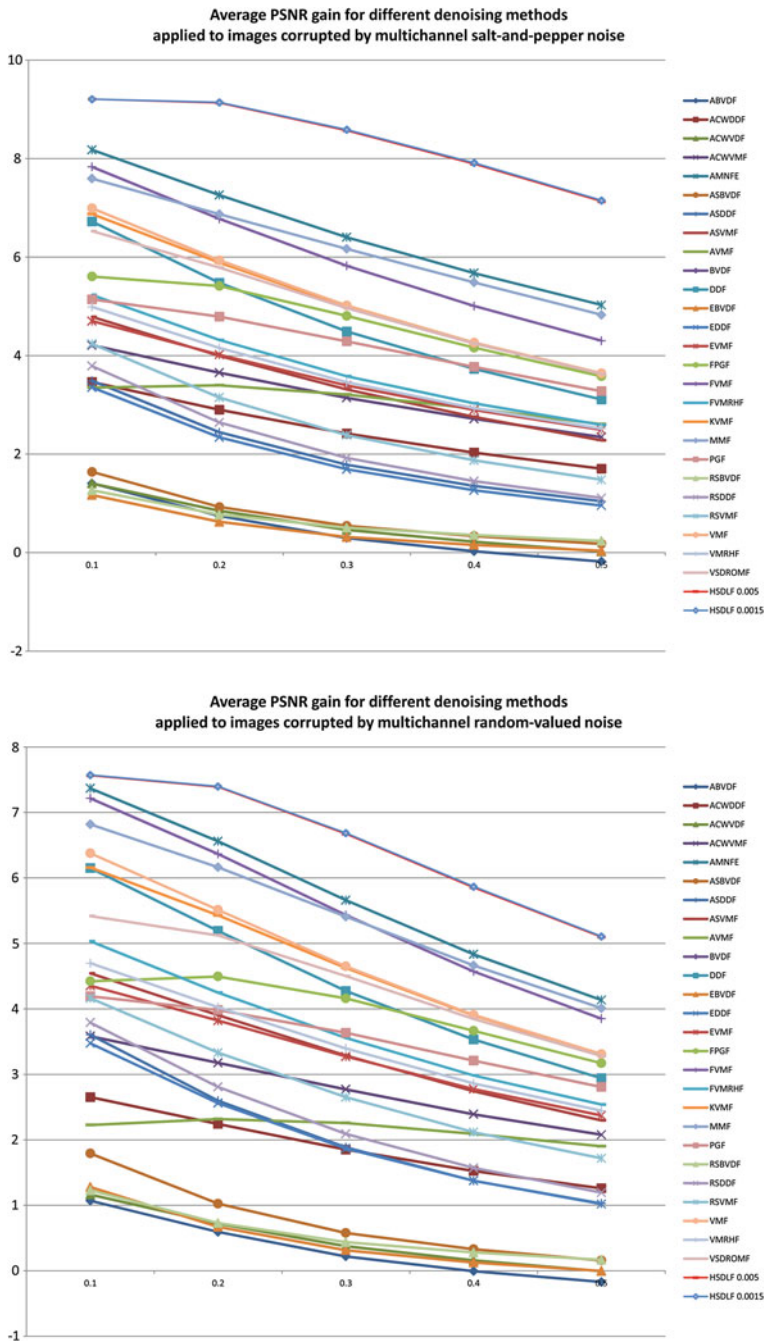
HSDLF threshold control parameter $\tau$		Total AG	Max. AG	Min. AG	Max. G	Min. G
Random-valued noise	PSNR 0.005	+6.52	+7 dB <i>Caps</i> 768 × 512	+6.19 dB <i>Peppers</i> 512 × 512	+7.9 dB <i>Caps</i> 768 × 512 RV ND: 0.2	+4.74 dB <i>Peppers</i> 512 × 512 RV ND: 0.5
		0.0015	+6.53	+7.02 dB <i>Caps</i> 768 × 512	+6.21 dB <i>Peppers</i> 512 × 512	+7.91 dB <i>Caps</i> 768 × 512 RV ND: 0.2
	MAE 0.005	-13.58	-14.28 <i>Caps</i> 768 × 512	-13.09 <i>Peppers</i> 512 × 512	-18.39 <i>Caps</i> 768 × 512 RV ND: 0.5	-7.16 <i>Lena</i> 512 × 512 RV ND: 0.1
		0.0015	-13.61	-14.31 <i>Caps</i> 768 × 512	-13.11 <i>Peppers</i> 512 × 512	-18.46 <i>Caps</i> 768 × 512 RV ND: 0.5
	NCD 0.005	-0.179	-0.235 <i>Caps</i> 768 × 512	-0.131 <i>Peppers</i> 512 × 512	-0.295 <i>Caps</i> 768 × 512 RV ND: 0.5	-0.081 <i>Peppers</i> 512 × 512 RV ND: 0.1
		0.0015	-0.179	-0.235 <i>Caps</i> 768 × 512	-0.131 <i>Peppers</i> 512 × 512	-0.295 <i>Caps</i> 768 × 512 RV ND: 0.5

*Total AG* overall average gain, *Max. AG* maximum average gain, *Min. AG* minimum average gain, *Max. G* maximum gain, *Min. G* minimum gain, *S&P ND* salt-and-pepper noise density, *RV ND* random-valued noise density

multichannel noise consisting of random-valued impulse noise and ACGN are very similar, so there is no loss in generality of presented results [3].

As noted in Sect. 1, HSDLF's results are compared to MMF, all 25 spatial domain impulse noise filters observed in Sect. 4.1, and two wavelet-domain filters: BM3D [11] and ProbShrink [42], where wavelets for the latter filter are selected to give the best possible denoising results in terms of PSNR gains.

Salt-and-pepper noise is generated identically as in Sect. 4.1, while ACGN is generated using built-in *MATLAB* function *imnoise* ( $I, 'Gaussian', 0, \sigma^2$ ), where (third parameter) 0 is the value of expectation and (fourth parameter)  $\sigma^2$  is the variance of Gaussian noise distribution.



**Fig. 3** Average PSNR gains for all compared impulse noise filters applied to all observed benchmark images corrupted by various densities of salt-and-pepper and random-valued impulse noise (indicated on *horizontal axis*)

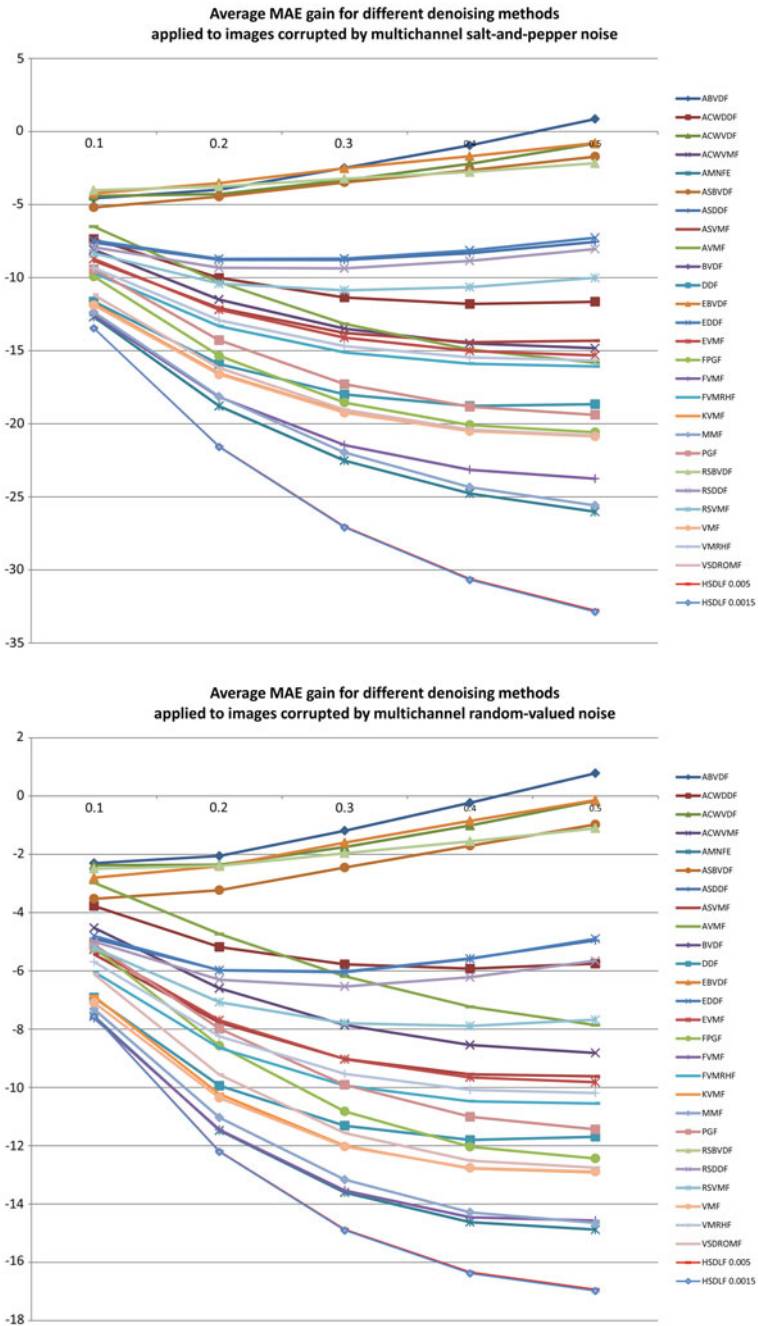
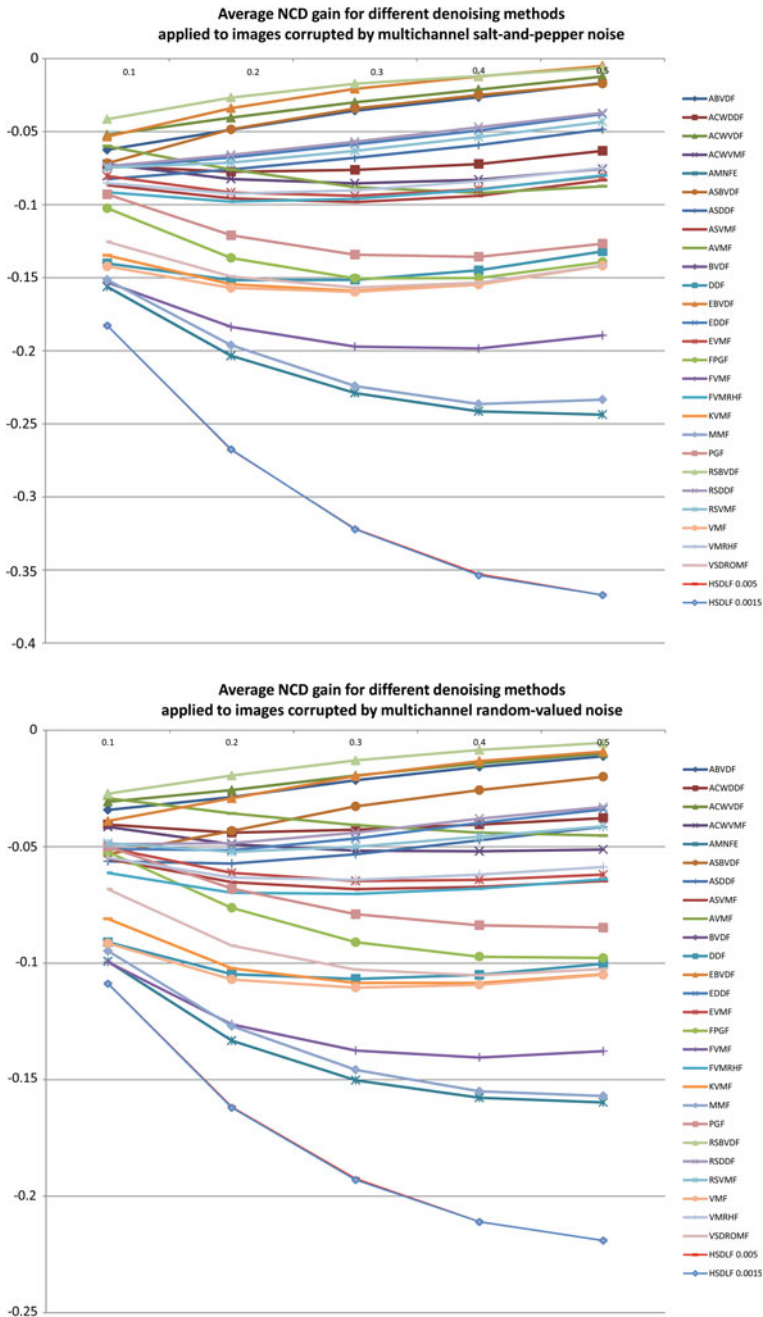
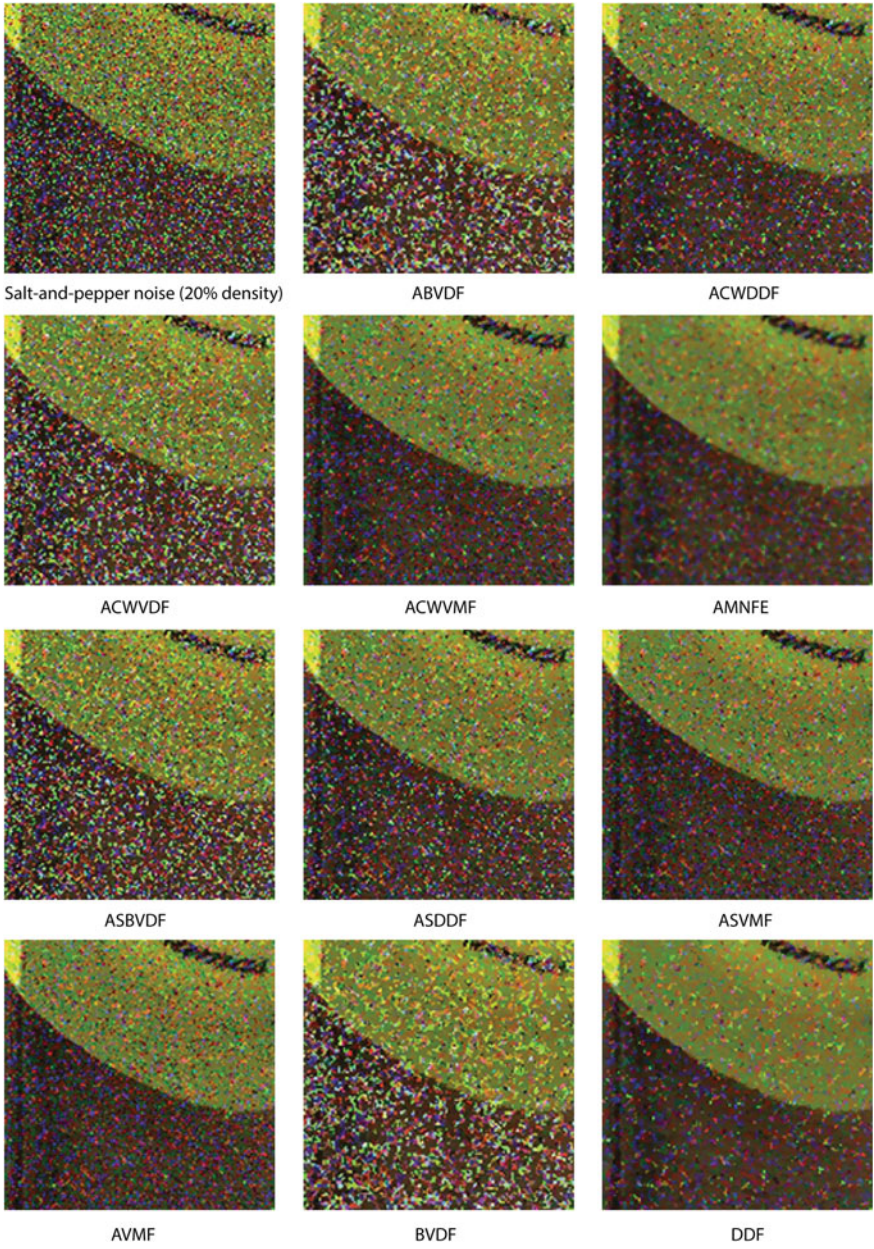


Fig. 4 Average MAE gains for all compared impulse noise filters applied to all observed benchmark images corrupted by various densities of salt-and-pepper and random-valued impulse noise (indicated on horizontal axis)



**Fig. 5** Average NCD gains for all compared impulse filters applied to all observed benchmark images corrupted by various densities of salt-and-pepper and random-valued impulse noise (indicated on *horizontal axis*)



**Fig. 6** (a)–(c) Denoising results of all compared impulse noise filters applied to a segment of image “Caps” ( $768 \times 512$ ) corrupted by salt-and-pepper noise with 0.2 (20%) density

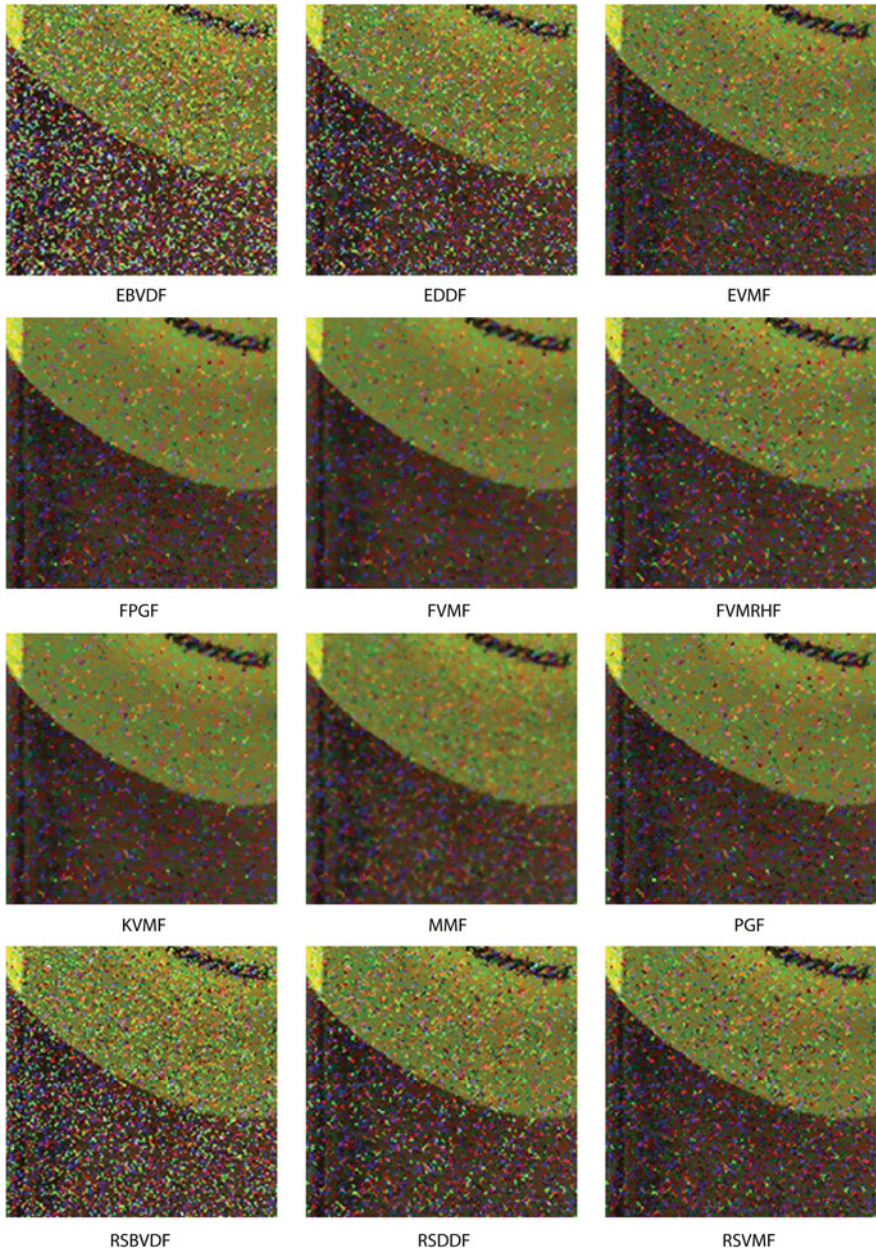


Fig. 6 (continued)

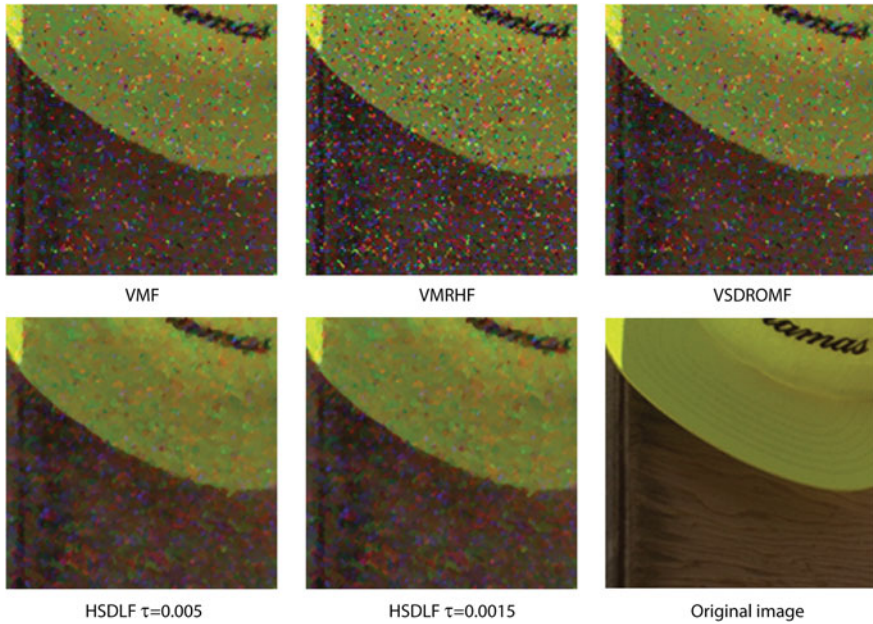
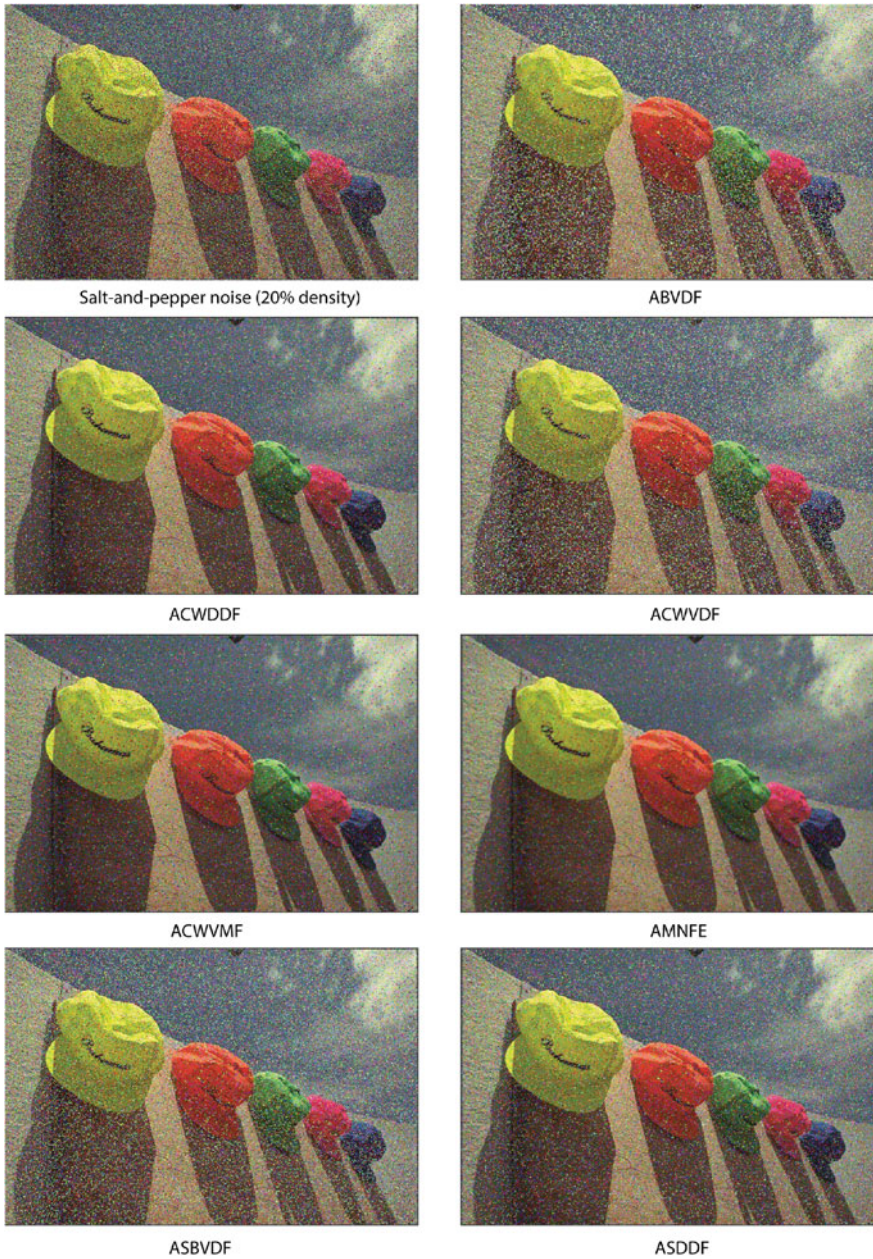


Fig. 6 (continued)

Like with impulse noise, HSDLF provides optimal denoising results in terms of visual quality and objective error metric criteria for certain values of threshold control parameter  $\tau$  (see Sect. 4.1); this time three values of  $\tau$  are considered: 0.05, 0.025 and 0.015. Experiments on *Signal and Image Processing Institute (SIPI) Volume 3: Miscellaneous* and *Kodak Photo CD PCD0992* image data sets have confirmed that HSDLF performances in removal of mixed multichannel noise deteriorate rapidly for  $\tau > 0.075$  and  $\tau < 0.01$ . This shows that the ranges of optimal values of parameter  $\tau$  depend on the type of multichannel noise (see Sect. 4.1) [2, 3].

PSNR gain is the most significant of all error metrics criterion related to removal of mixed multichannel noise, since it measures the capability of a filter to suppress noise (see Sect. 4.1) [5]; therefore, the denoising results are compared only in terms of this criterion. Tables 7, 8, 9, 10 and 11 and Figs. 10, 11 and 12 give a detailed summary of obtained results [3]. The best filtering performances in Tables 7–10 are indicated by **boldface** font.

A detailed overview of HSDLF performances in terms of PSNR gains for all three observed values of  $\tau$  is given in Table 11. HSDLF is compared in Table 11 and Figs. 10, 11 and 12 to both observed wavelet-based filters (ProbShrink [42] and BM3D [11]), and three spatial domain filters which give the best results in removal of mixed multichannel noise: MMF [4], AMNFE [43] and FVMF [8, 44, 45].



**Fig. 7** (a)–(d) Denoising results of all compared impulse noise filters applied to image “Caps” ( $768 \times 512$ ) corrupted by salt-and-pepper noise with 0.2 (20%) density



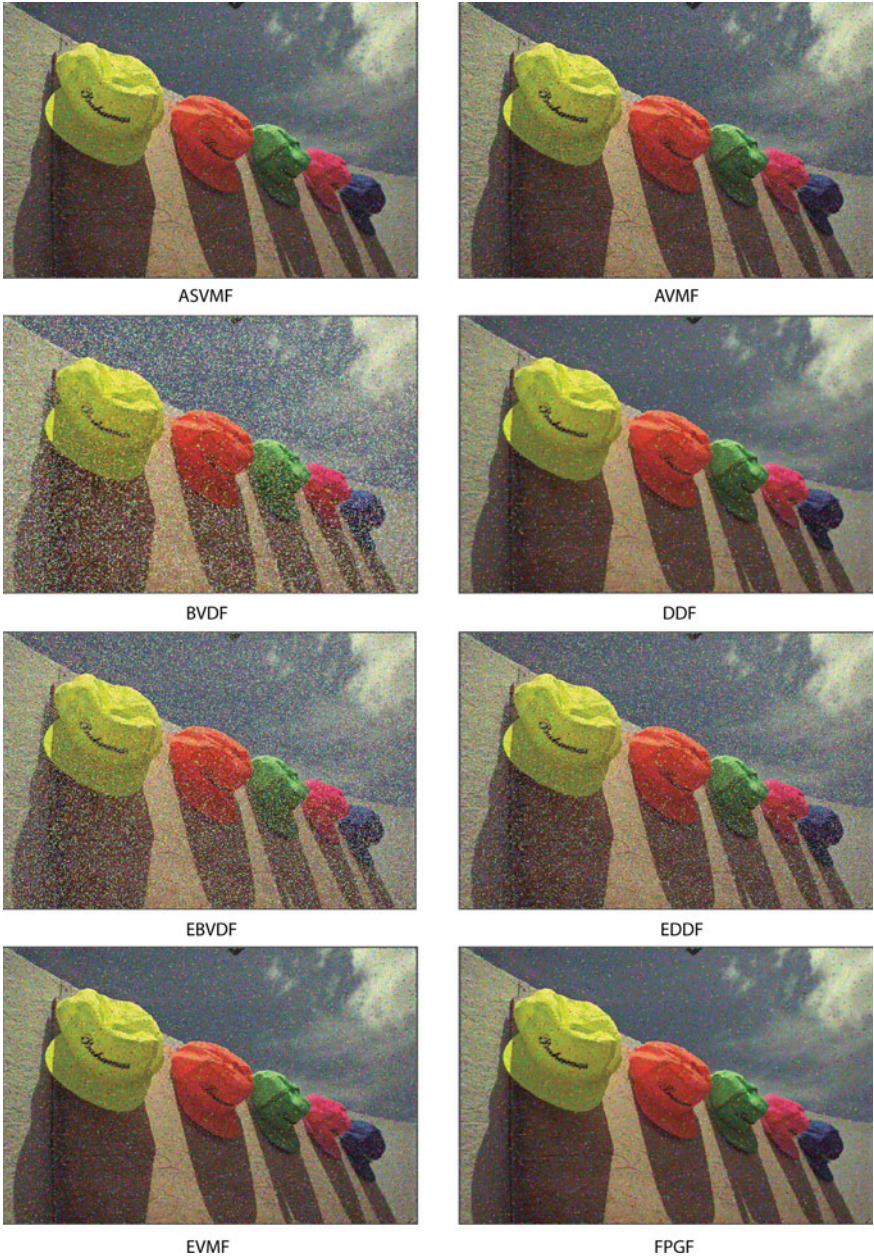


Fig. 7 (continued)

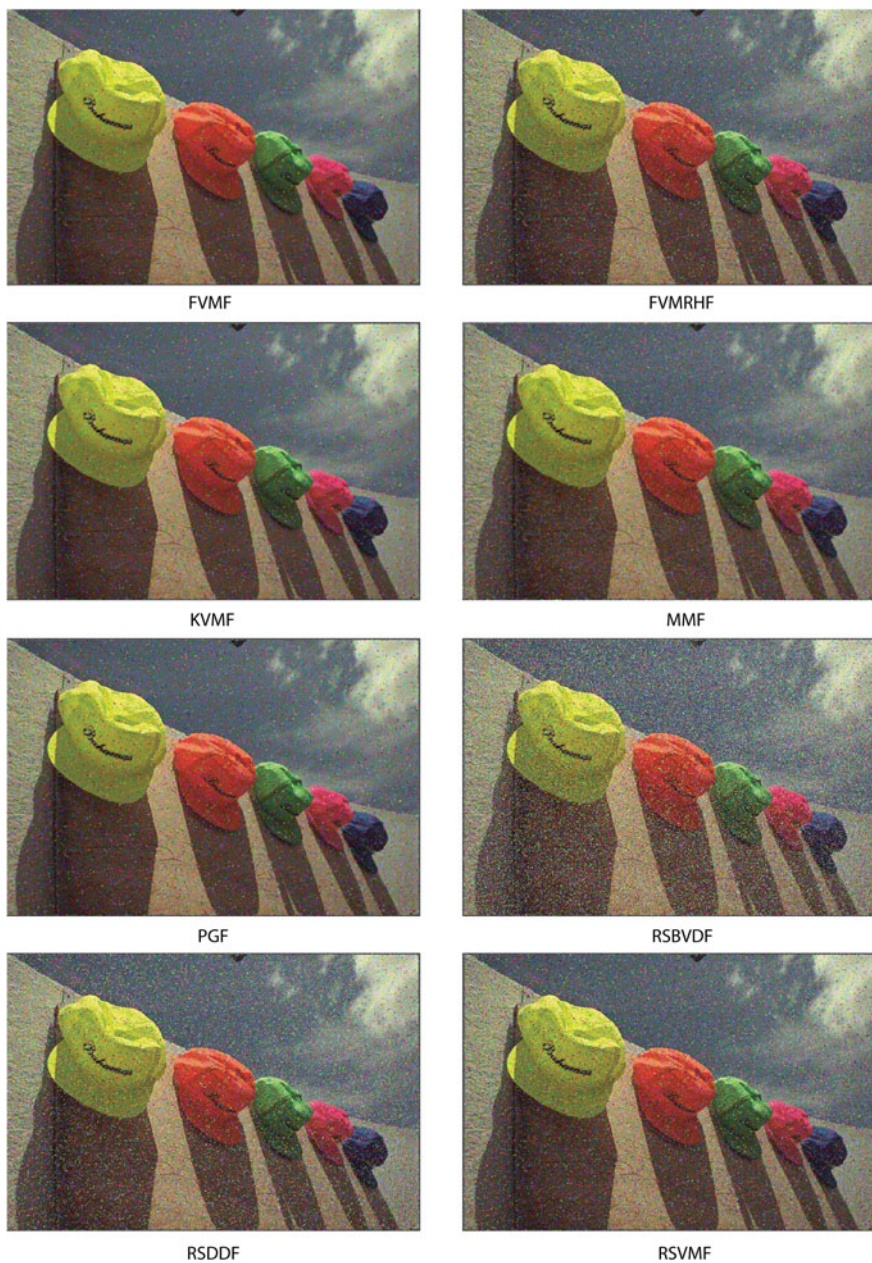


Fig. 7 (continued)

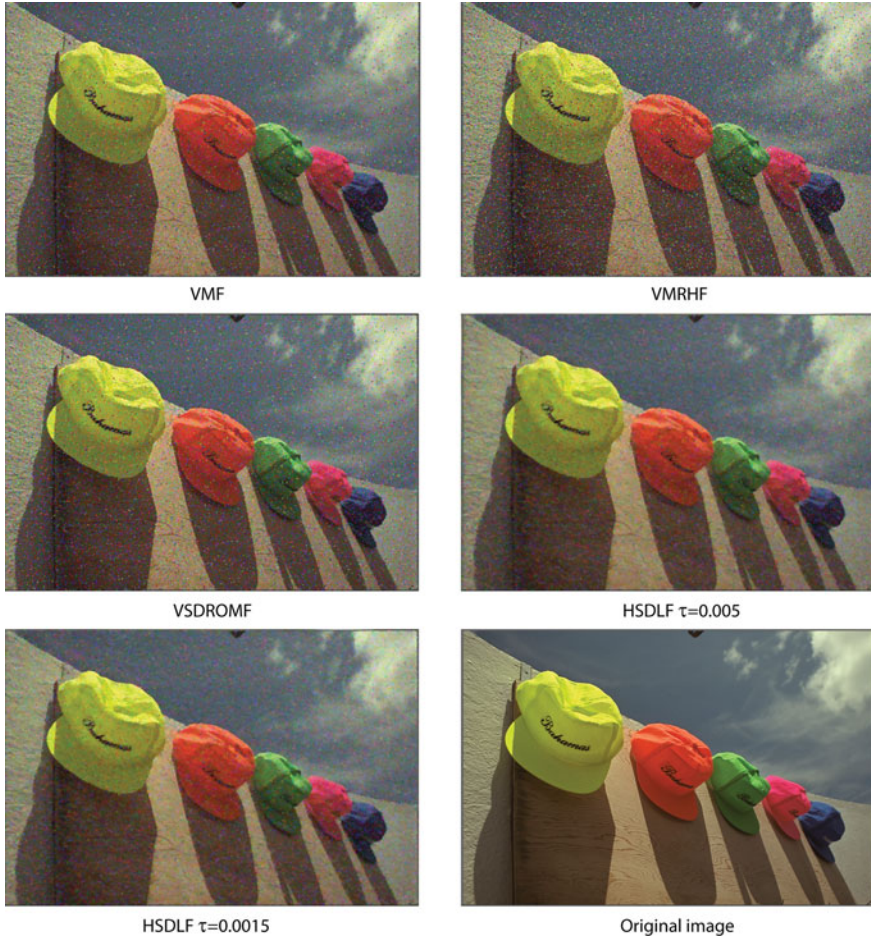


Fig. 7 (continued)

Figure 10 displays the average PSNR gain values for HSDLF [3], ProbShrink [42], BM3D [11], MMF [4], AMNFE [43] and FVMF [8, 44, 45] calculated over all benchmark images. Figures 11 and 12 illustrate the visual quality of denoising results of HSDLF [3], ProbShrink [42], BM3D [11], MMF [4], AMNFE [43] and FVMF [8, 44, 45] applied to image “Peppers” corrupted by mixed multichannel noise.

By closely observing the results presented in Tables 7, 8, 9, 10 and 11 and Figs. 10, 11, and 12, following conclusions can be drawn [3]:

- For all observed mixed noise powers and calculated over all observed benchmark images, HSDLF gives the best results in terms of PSNR gain for  $\tau=0.015$  in 56%, for  $\tau=0.025$  in 18%, and for  $\tau=0.05$  in 26% of the cases (excluding ties rounded to third decimal place).

**Table 7** Performance results of all observed multichannel denoising filters applied to image “Caps” (with  $768 \times 512$  pixel image size) corrupted by various powers of mixed noise. The best filtering performances are indicated by **boldface** font

Noise Power	Denoising method	PSNR [dB]																
		Gaussian colour noise variance	0.05	0.05	0.1	0.15	0.15	0.1	0.15	0.15	0.2	0.2	0.3	0.2	0.3			
	Salt-and-pepper noise density	0.05	0.1	0.1	0.05	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.2	0.2	0.3	0.2
	None	14.95	14.07	12.67	12.27	12.27	12.19	11.5	11.34	10.3	10.28							
	ABVDF	14.94	13.86	12.27	11.81	11.69	11.69	10.90	10.72	9.50	9.49							
	ACWDDF	16.71	15.94	14.40	13.92	13.90	13.90	13.09	12.91	11.63	11.60							
	ACWVDF	15.10	14.08	12.52	12.06	11.95	11.95	11.17	10.99	9.79	9.77							
	ACWVMF	17.32	16.59	15.08	14.62	14.60	14.60	13.81	13.62	12.37	12.34							
	AMNFE	21.34	20.43	18.66	18.09	18.05	18.05	17.08	16.86	15.25	15.23							
	ASBVDF	15.20	14.22	12.72	12.27	12.18	12.18	11.42	11.25	10.09	10.07							
	ASDDF	16.17	15.27	13.63	13.16	13.12	13.12	12.32	12.12	10.90	10.88							
	ASVMF	17.50	16.67	14.99	14.50	14.50	14.50	13.65	13.46	12.17	12.14							
	AVMF	16.84	16.29	15.02	14.62	14.62	14.62	13.92	13.75	12.57	12.54							
	BVDF	14.93	13.87	12.30	11.83	11.71	11.71	10.93	10.75	9.53	9.53							
	DDF	19.00	18.09	16.22	15.66	15.64	15.64	14.67	14.45	12.93	12.91							
	EBVDF	15.12	14.12	12.61	12.17	12.06	12.06	11.30	11.13	9.96	9.95							
	EDDF	16.30	15.31	13.66	13.18	13.10	13.10	12.28	12.09	10.84	10.82							
	EVMF	17.86	16.99	15.35	14.87	14.83	14.83	14.01	13.81	12.51	12.49							
	FPGF	18.99	18.24	16.60	16.07	16.07	16.07	15.18	14.96	13.52	13.49							
	FVMF	20.28	19.38	17.69	17	17	17	16.96	15.76	14.11	14.08							
	FVMRHF	18.36	17.45	15.70	15.19	15.19	15.14	14.26	14.06	12.68	12.65							
	KVMF	19.37	18.49	16.72	16.16	16.16	16.15	15.23	15.02	13.56	13.53							
	MMF	20.97	20.05	18.22	17.61	17.61	17.6	16.59	16.36	14.75	14.71							
	PGF	18.30	17.62	16.11	15.61	15.61	15.62	14.79	14.58	13.22	13.19							
	RSBVDF	15.22	14.28	12.78	12.35	12.27	12.27	11.52	11.34	10.20	10.19							

(Continued)

**Table 7** (continued)

PSNR [dB]	Image: <i>Caps768</i> × 512													
Noise Power	Gaussian colour noise variance			Salt-and-pepper noise density			0.05			0.1				
	0.05	0.1	0.05	0.1	0.15	13.30	0.15	0.1	0.15	0.1	0.15	0.2	0.2	0.3
RSDDF	16.45	15.54	13.84	13.34	13.34	13.30	12.47	12.26	11.00	10.97				
RSVMF	16.83	15.97	14.26	13.76	13.76	13.75	12.91	12.71	11.45	11.41				
VMF	19.42	18.53	16.75	16.20	16.20	16.18	15.26	15.04	13.58	13.55				
VMRHF	18.19	17.30	15.58	15.07	15.07	15.03	14.16	13.96	12.60	12.57				
VSDROMF	19.35	18.48	16.73	16.18	16.18	16.17	15.25	15.03	13.57	13.54				
BM3D	17.29	16.74	16.15	15.7	15.7	16.14	15.91	15.68	16.47	16.42				
ProbShrink	–	–	20.15	19.5	19.5	19.8	18.57	18.39	16.61	16.58				
HSDLF $\tau=0.05$	23.52	22.49	20.49	19.82	19.82	19.72	18.56	18.33	16.43	16.38				
HSDLF $\tau=0.025$	24.06	23.11	21.19	20.53	20.53	20.43	19.28	19.06	17.1	17.06				
HSDLF $\tau=0.015$	<b>24.19</b>	<b>23.28</b>	<b>21.42</b>	<b>20.78</b>	<b>20.78</b>	<b>20.67</b>	<b>19.54</b>	<b>19.33</b>	<b>17.37</b>	<b>17.33</b>				

**Table 8** Performance results of all observed multichannel denoising filters applied to image “Parrots” (with  $768 \times 512$  pixel image size) corrupted by various powers of mixed noise. The best filtering performances are indicated by **boldface** font

PSNR [dB]		Image: <i>Parrots</i> $768 \times 512$											
Noise Power	Denoising method	Gaussian colour noise variance					Salt-and-pepper noise density						
		0.05	0.05	0.05	0.1	0.15	0.15	0.15	0.15	0.2	0.2	0.2	0.2
	None	15.05	14.1	12.7	12.31	12.17	11.47	11.32	10.15	10.16			
	ABVDF	15.22	14.15	12.50	12.03	11.89	11.07	10.88	9.57	9.57			
	ACWDDF	16.85	16.03	14.47	13.99	13.92	13.12	12.91	11.52	11.52			
	ACWVDF	15.38	14.35	12.73	12.27	12.13	11.33	11.14	9.83	9.83			
	ACWVMF	17.37	16.60	15.08	14.61	14.55	13.75	13.56	12.18	12.18			
	AMNFE	21.38	20.4	18.55	17.97	17.86	16.86	16.64	14.84	14.84			
	ASBVDF	15.47	14.46	12.89	12.45	12.32	11.54	11.36	10.09	10.09			
	ASDDF	16.39	15.42	13.76	13.28	13.18	12.37	12.18	10.85	10.84			
	ASVMF	17.61	16.71	15.04	14.53	14.48	13.63	13.43	11.99	12.01			
	AVMF	16.89	16.30	15.03	14.62	14.58	13.87	13.69	12.38	12.39			
	BVDF	15.48	14.40	12.70	12.22	12.08	11.23	11.03	9.67	9.68			
	DDF	19.13	18.14	16.27	15.70	15.61	14.65	14.42	12.77	12.79			
	EBVDF	15.41	14.37	12.80	12.35	12.21	11.43	11.25	9.97	9.97			
	EDDF	16.56	15.54	13.83	13.34	13.22	12.38	12.19	10.80	10.80			
	EVMF	17.94	17.01	15.36	14.87	14.78	13.94	13.75	12.31	12.32			
	FPGF	19.06	18.24	16.57	16.04	15.97	15.06	14.84	13.26	13.27			
	FVMF	20.46	19.52	17.63	17.04	16.98	15.99	15.76	14.02	14.04			
	FVMRHF	18.42	17.44	15.68	15.17	15.07	14.18	13.97	12.46	12.46			
	KVMF	19.45	18.50	16.68	16.14	16.06	15.12	14.90	13.29	13.30			
	MMF	21.04	20.01	18.11	17.52	17.43	16.41	16.17	14.39	14.41			
	PGF	18.34	17.62	16.08	15.58	15.53	14.68	14.48	12.97	12.99			
	RSBVDF	15.41	14.43	12.90	12.45	12.34	11.58	11.40	10.16	10.16			
	RSDDF	16.63	15.68	13.95	13.44	13.36	12.52	12.31	10.93	10.92			

(Continued)



**Table 9** Performance results of all observed multichannel denoising filters applied to image "Lena" (with  $512 \times 512$  pixel image size) corrupted by various powers of mixed noise. The best filtering performances are indicated by **boldface** font

PSNR [dB]		Image: <i>Lena</i> $512 \times 512$										
Noise Power	Denoising method	Gaussian colour noise variance					Salt-and-pepper noise density					
		0.05	0.05	0.05	0.1	0.15	0.1	0.15	0.15	0.2	0.2	0.3
	None	15.06	14.12	12.73	12.35	12.23	11.53	11.37	10.26	10.25		
	ABVDF	15.95	15.00	13.47	13.04	12.93	12.11	11.94	10.69	10.67		
	ACWDDF	16.94	16.18	14.65	14.19	14.18	13.36	13.17	11.93	11.89		
	ACWVDF	15.99	15.03	13.51	13.09	12.98	12.18	12.01	10.77	10.75		
	ACWVMF	17.35	16.63	15.07	14.59	14.58	13.76	13.56	12.27	12.24		
	AMNFE	21.3	20.43	18.54	17.97	17.95	16.91	16.68	15.05	15		
	ASBVDF	15.98	15.00	13.44	13.02	12.92	12.12	11.95	10.74	10.73		
	ASDDF	16.71	15.79	14.13	13.66	13.60	12.76	12.56	11.31	11.28		
	ASVMF	17.54	16.70	15.01	14.50	14.50	13.62	13.41	12.08	12.04		
	AVMF	16.86	16.32	15.01	14.58	14.60	13.86	13.67	12.47	12.44		
	BVDF	16.78	15.74	14.00	13.52	13.39	12.48	12.28	10.90	10.87		
	DDF	19.16	18.29	16.46	15.91	15.90	14.92	14.68	13.21	13.16		
	EBVDF	15.96	14.94	13.43	13.02	12.90	12.13	11.96	10.75	10.74		
	EDDF	17.10	16.12	14.39	13.89	13.82	12.94	12.74	11.44	11.42		
	EVMF	17.88	17.03	15.35	14.85	14.82	13.96	13.76	12.42	12.39		
	FPGF	19.00	18.27	16.57	16.04	16.05	15.11	14.88	13.42	13.37		
	FVMF	20.38	19.55	17.64	17.05	17.08	16.04	15.78	14.21	14.16		
	FVMRHF	18.37	17.47	15.69	15.16	15.11	14.20	13.99	12.58	12.54		
	KVMF	19.39	18.54	16.69	16.14	16.15	15.18	14.94	13.46	13.41		
	MMF	20.97	20.08	18.18	17.58	17.59	16.53	16.28	14.65	14.6		
	PGF	18.30	17.66	16.06	15.57	15.59	14.71	14.50	13.12	13.07		
	RSBVDF	15.73	14.78	13.29	12.88	12.78	12.04	11.87	10.70	10.70		
	RSDDF	16.83	15.93	14.22	13.72	13.69	12.83	12.62	11.34	11.32		

(Continued)





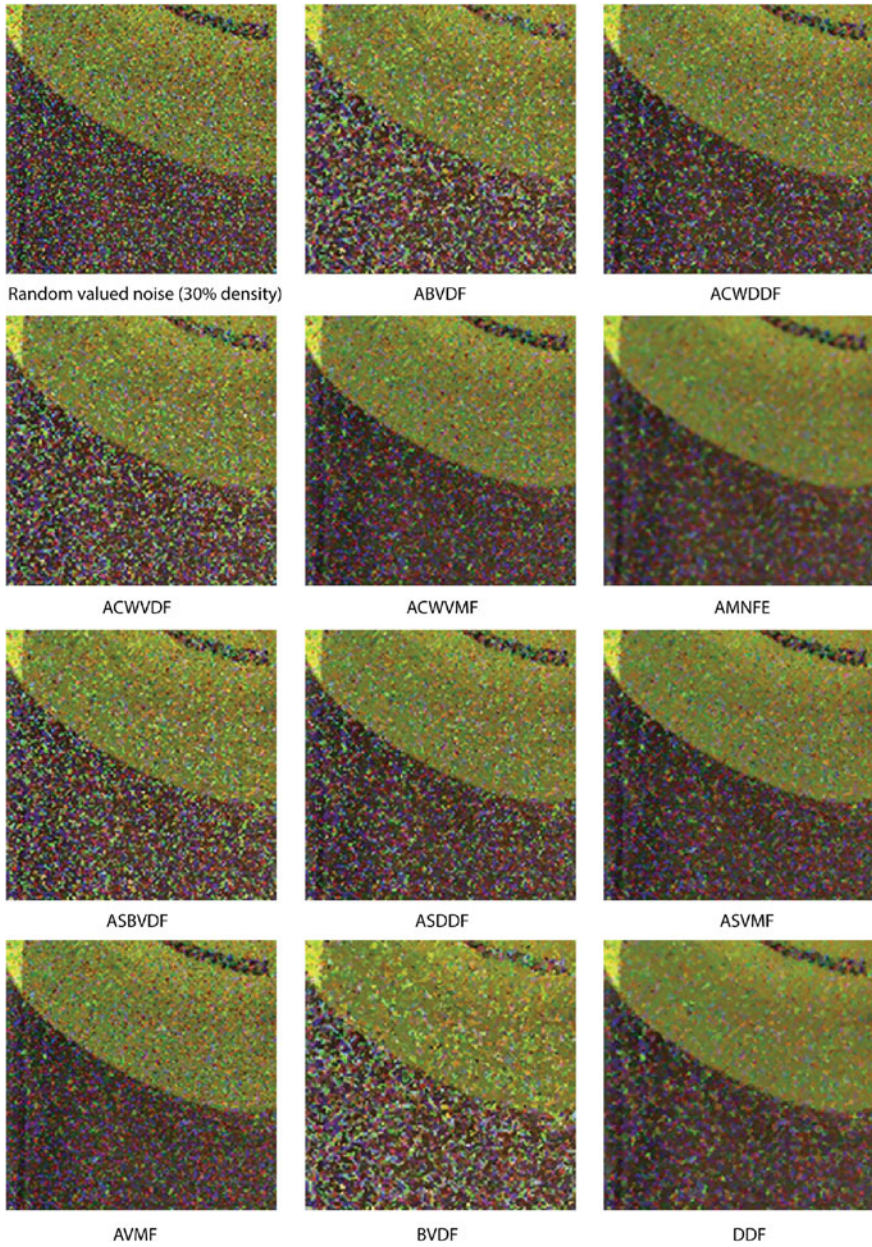
**Table 10** Performance results of all observed multichannel denoising filters applied to image “Peppers” (with  $512 \times 512$  pixel image size) corrupted by various powers of mixed noise. The best filtering performances are indicated by **boldface** font

Noise Power	PSNR [dB]											
	Gaussian colour noise variance		Salt-and-pepper noise density		0.1		0.15		0.2		0.3	
	0.05	0.05	0.05	0.1	0.1	0.15	0.15	0.15	0.2	0.2	0.3	0.3
Image: Peppers $512 \times 512$												
Denoising method	None	15.13	14.1	12.73	12.37	12.18	11.46	11.33	10.05	10.09	10.13	10.13
	ABVDF	16.02	15.03	13.36	12.91	12.72	11.86	11.68	10.09	10.09	10.13	10.13
	ACWDDF	17.04	16.23	14.67	14.19	14.13	13.29	13.08	11.59	11.59	11.62	11.62
	ACWVDF	16.04	15.05	13.42	12.97	12.80	11.96	11.78	10.24	10.24	10.28	10.28
	ACWVMF	17.40	16.62	15.12	14.64	14.60	13.78	13.55	12.09	12.09	12.11	12.11
	AMNFE	21.16	20.19	18.38	17.8	17.7	16.69	16.42	14.58	14.58	14.6	14.6
	ASBVDF	16.04	15.03	13.40	12.97	12.79	11.97	11.80	10.32	10.32	10.35	10.35
	ASDDF	16.84	15.89	14.19	13.69	13.59	12.72	12.52	11.02	11.02	11.04	11.04
	ASVMF	17.63	16.72	15.10	14.59	14.54	13.68	13.44	11.95	11.95	11.97	11.97
	AVMF	16.94	16.33	15.07	14.64	14.63	13.90	13.68	12.32	12.32	12.34	12.34
	BVDF	16.62	15.57	13.76	13.28	13.09	12.17	11.98	10.28	10.28	10.33	10.33
	DDF	19.11	18.15	16.34	15.77	15.69	14.75	14.50	12.79	12.79	12.82	12.82
	EBVDF	15.92	14.86	13.26	12.85	12.63	11.84	11.68	10.21	10.21	10.25	10.25
	EDDF	17.03	16.02	14.30	13.80	13.66	12.77	12.56	11.00	11.00	11.03	11.03
	EVMF	17.93	17.01	15.39	14.90	14.82	13.96	13.73	12.23	12.23	12.25	12.25
	FPGF	18.98	18.18	16.56	16.03	15.98	15.06	14.82	13.17	13.17	13.19	13.19
	FVMF	20.35	19.42	17.6	17.01	16.97	15.97	15.69	13.92	13.92	13.94	13.94
	FVMRHF	18.39	17.43	15.69	15.16	15.08	14.17	13.94	12.34	12.34	12.37	12.37
	KVMF	19.39	18.45	16.69	16.13	16.07	15.12	14.87	13.20	13.20	13.23	13.23
	MMF	20.86	19.89	18.01	18.18	17.37	17.46	16.09	14.27	14.27	14.29	14.29
	PGF	18.33	17.60	16.10	15.60	15.57	14.70	14.46	12.88	12.88	12.92	12.92
	RSBVDF	15.81	14.82	13.26	12.84	12.69	11.89	11.72	10.33	10.33	10.36	10.36

(Continued)

**Table 10** (continued)

PSNR [dB]		Image: <i>Peppers</i> $512 \times 512$															
Noise Power		Gaussian colour noise variance				Salt-and-pepper noise density											
		0.05	0.05	0.1	0.1	0.05	0.05	0.1	0.1	0.15	0.15	0.1	0.1	0.2	0.2	0.3	0.2
	RSDDF	16.90	15.97	14.27	13.76	13.67	12.78	12.57	11.04	11.07							
	RSVMF	17.00	16.08	14.40	13.89	13.84	12.95	12.73	11.23	11.25							
	VMF	19.43	18.48	16.72	16.15	16.09	15.14	14.89	13.22	13.24							
	VMRHF	18.23	17.28	15.58	15.06	14.98	14.08	13.85	12.27	12.30							
	VSDROMF	19.35	18.43	16.69	16.13	16.07	15.13	14.88	13.21	13.23							
	BM3D	17.49	16.73	16.22	15.84	16.02	15.71	15.58	15.33	15.42							
	ProbShrink	–	19.52	18.49	16.37	17.91	17.04	16.95	14.91	14.98							
	HSDLF $\tau = 0.05$	22.67	21.65	19.78	19.22	19.06	18	17.74	15.77	15.81							
	HSDLF $\tau = 0.025$	22.98	22.01	20.15	19.6	19.47	18.4	18.12	16.13	16.18							
	HSDLF $\tau = 0.015$	<b>23.04</b>	<b>22.09</b>	<b>20.26</b>	<b>19.71</b>	<b>19.59</b>	<b>18.52</b>	<b>18.24</b>	<b>16.26</b>	<b>16.3</b>							



**Fig. 8** (a)–(c) Denoising results of all compared impulse noise filters applied to a segment of image “Caps” ( $768 \times 512$ ) corrupted by random-valued noise with 0.3 (30%) density

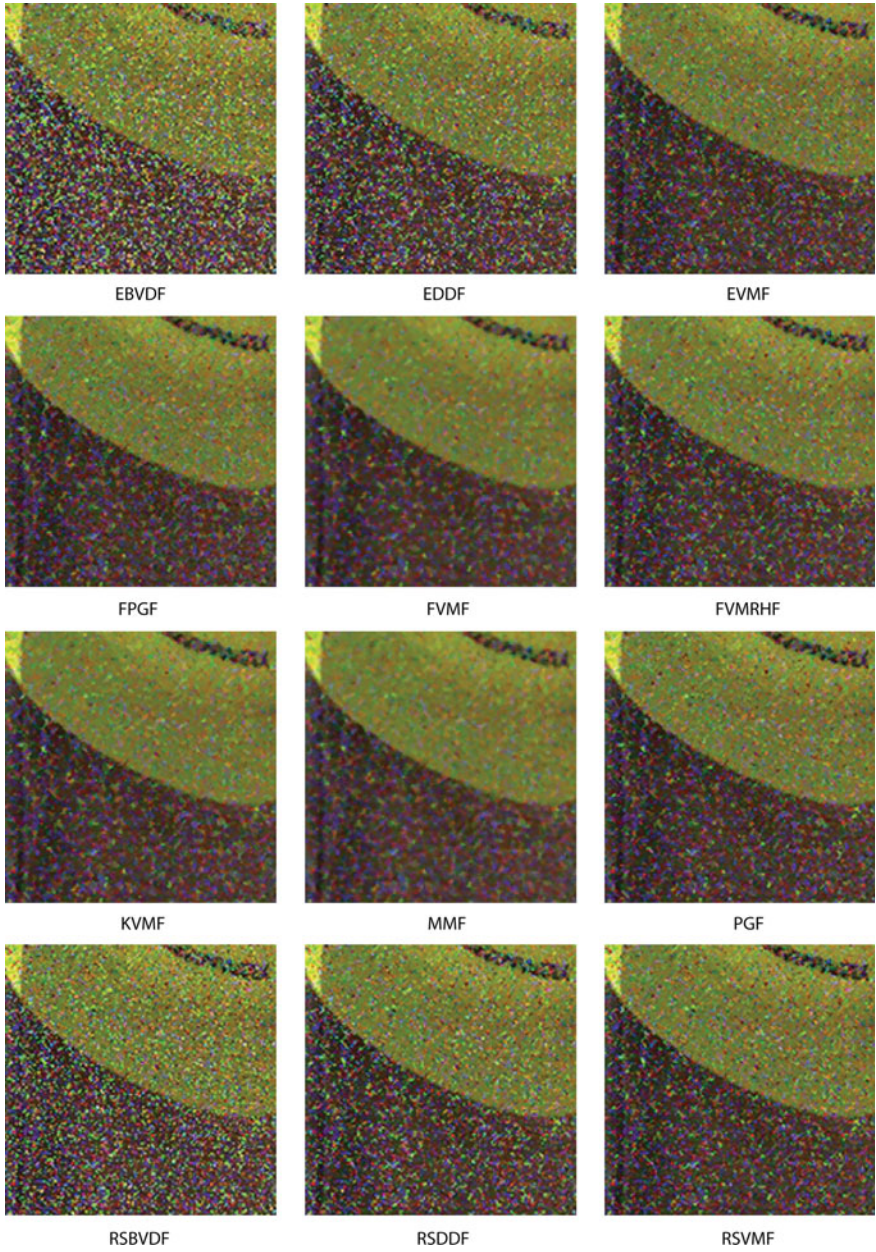


Fig. 8 (continued)

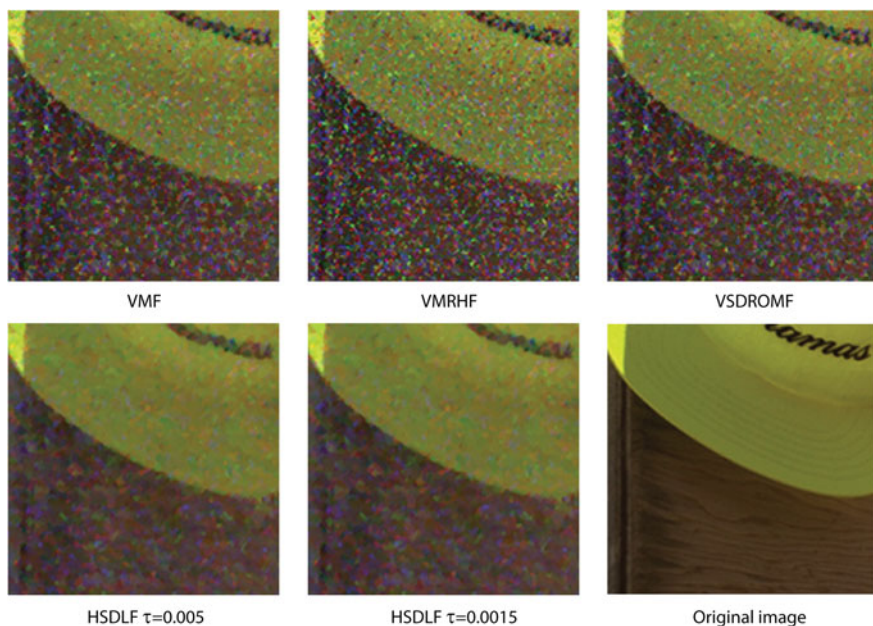


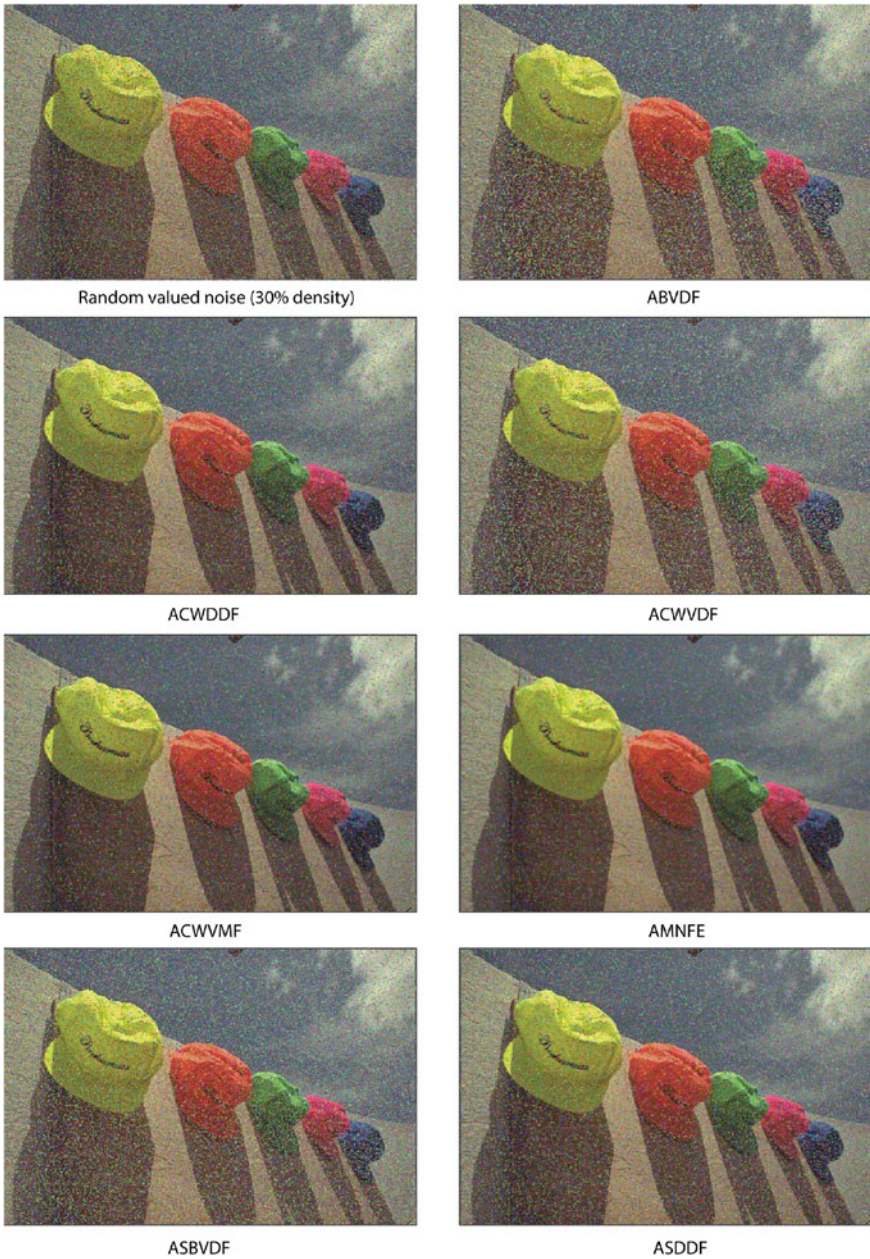
Fig. 8 (continued)

- PSNR gains tend to slightly rise with the increase of mixed noise powers for all three observed HSDLF's values of  $\tau$  on all benchmark images.
- HSDLF reaches maximum PSNR gain on image "Jellybeans1" for  $\tau=0.05$ , and on image "Girl2" for  $\tau=0.025$  and  $\tau=0.015$ . Minimum PSNR gains are attained on image "Mandrill" for all three values of  $\tau$ , however they appear for the lowest power of mixed noise ( $\xi=0.025$  and  $\sigma^2=0.025$ ).
- HSDLF gives better results in terms of overall, maximal and minimal average PSNR gains than all observed filters for all threshold control parameter values  $\tau$  (0.05, 0.025 and 0.015).
- In terms of both PSNR gain and visual quality, HSDLF consistently outperforms all other spatial domain filters as well as BM3D filter for all observed mixed noise powers and for all values of parameter  $\tau$  [3]. This is evidenced in Tables 7, 8, 9, 10 and 11 in an abbreviated form.
- Abbreviated Tables 7, 8, 9, 10 and 11 show that HSDLF steadily outperforms ProbShrink filter in terms of PSNR gains for observed images "Lena", "Peppers", "Parrots" and "Caps" benchmark images. In Baljovic, Kovacevic and Baljovic [3], it is shown that HSDLF outperforms ProbShrink filter in terms of PSNR in 98.4% of all cases, i.e. calculated on all benchmark images for all observed values of the threshold control parameter  $\tau$  and noise powers. The exceptions appear only for the lowest mixed noise powers applied to image "Mandrill", and in two instances on image "Girl2"; however, even in these very rare cases, HSDLF and ProbShrink PSNR values have insignificant differences.

**Table 11** Detailed performance summary in terms of PSNR gains for HSDLF, MMF, AMNFE, FVMF, BM3D and ProbShrink; overall average PSNR gain is calculated over “Lena”, “Peppers”, “Parrots”, and “Caps” benchmark images and for all observed mixed noise powers; maximal and minimal PSNR gains are calculated for each of the benchmark images individually

Denoising method	Overall average PSNR gain [dB]	Maximal average PSNR gain [dB]	Minimal average PSNR gain [dB]	Maximal PSNR gain [dB]	Minimal PSNR gain [dB]
MMF	+5.27	+5.35	+5.21	+6.21	+4.19
		<i>Caps</i> 768 × 512	<i>Peppers</i> 512 × 512	<i>Parrots</i> 768 × 512	<i>Peppers</i> 512 × 512
				G NV: 0.25	G NV: 0.25
AMNFE	+5.63	+5.8	+5.43	+6.5	+4.5
		<i>Caps</i> 768 × 512	<i>Peppers</i> 512 × 512	<i>Parrots</i> 768 × 512	<i>Peppers</i> 512 × 512
				G NV: 0.25	G NV: 0.25
FVMF	+4.73	+4.76	+4.68	+5.57	+3.8
		<i>Lena</i> 512 × 512	<i>Peppers</i> 512 × 512	<i>Parrots</i> 768 × 512	<i>Caps</i> 768 × 512
				G NV: 0.025	G NV: 0.025
BM3D	+3.79	+3.86	+3.68	+6.18	+1.77
		<i>Lena</i> 512 × 512	<i>Couple</i> 256 × 256	<i>Caps</i> 768 × 512	<i>Peppers</i> 512 × 512
				G NV: 0.025	G NV: 0.025
ProbShrink	+6.05	+7.15	+5.35	+8.07	+4
		<i>Caps</i> 768 × 512	<i>Peppers</i> 512 × 512	<i>Caps</i> 768 × 512	<i>Peppers</i> 512 × 512
				G NV: 0.15	G NV: 0.15
HSDLF $\tau = 0.05$	+7.24	+7.51	+6.79	+8.61	+5.71
		<i>Lena</i> 512 × 512	<i>Peppers</i> 512 × 512	<i>Caps</i> 768 × 512	<i>Peppers</i> 512 × 512
				G NV: 0.25	G NV: 0.25
HSDLF $\tau = 0.025$	+7.66	+8.16	+7.15	+9.11	+6.07
		<i>Caps</i> 768 × 512	<i>Peppers</i> 512 × 512	<i>Caps</i> 768 × 512	<i>Peppers</i> 512 × 512
				G NV: 0.05	G NV: 0.05
HSDLF $\tau = 0.015$	+7.78	+8.38	+7.25	+9.24	+6.21
		<i>Caps</i> 768 × 512	<i>Peppers</i> 512 × 512	<i>Caps</i> 768 × 512	<i>Peppers</i> 512 × 512
				G NV: 0.25	G NV: 0.25

G NV Gaussian noise variance, S&P ND salt-and-pepper noise density



**Fig. 9** (a)–(d) Denoising results of all compared impulse noise filters applied to image “Caps” ( $768 \times 512$ ) corrupted by random-valued noise with 0.3 (30%) density



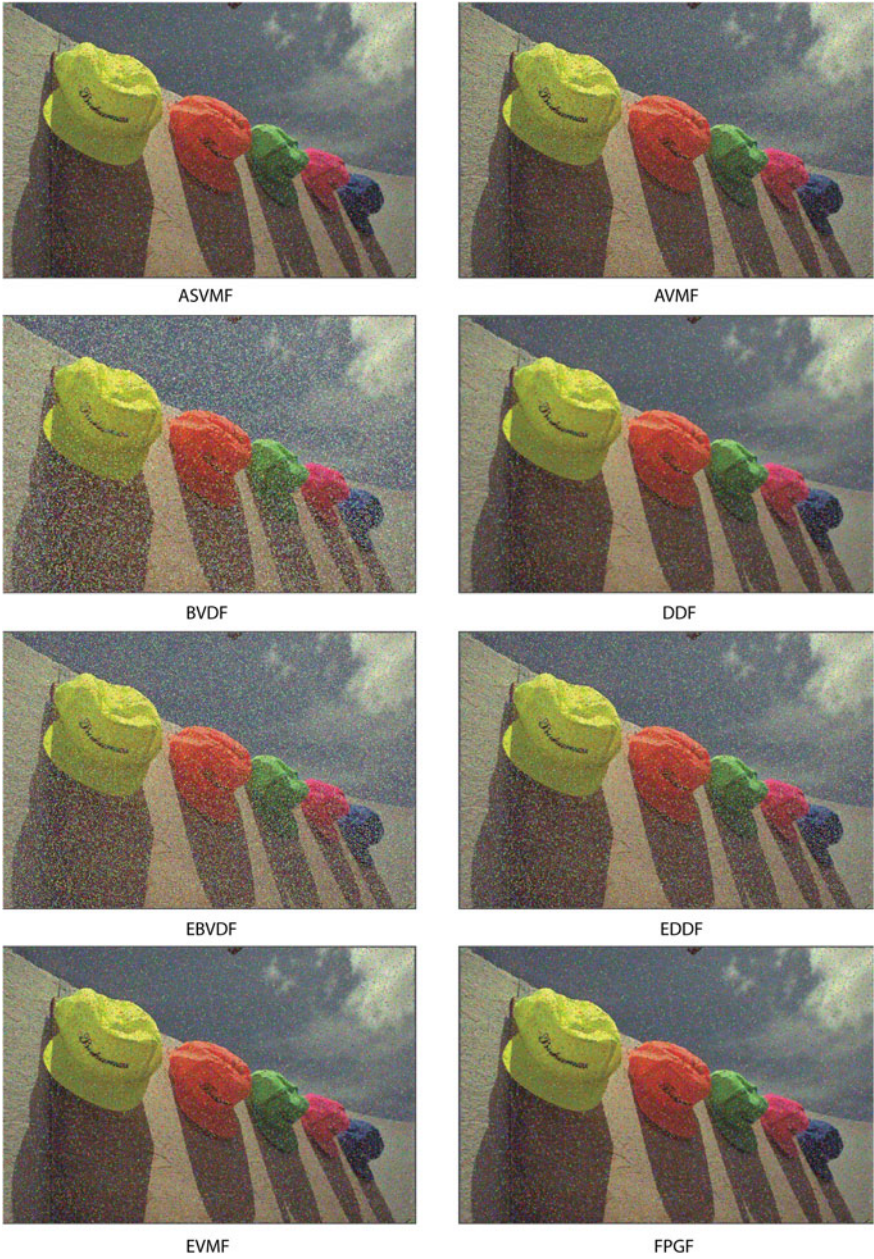


Fig. 9 (continued)



FVMF



FVMRHF



KVMF



MMF



PGF



RSBVDF



RSDDF



RSVMF

Fig. 9 (continued)

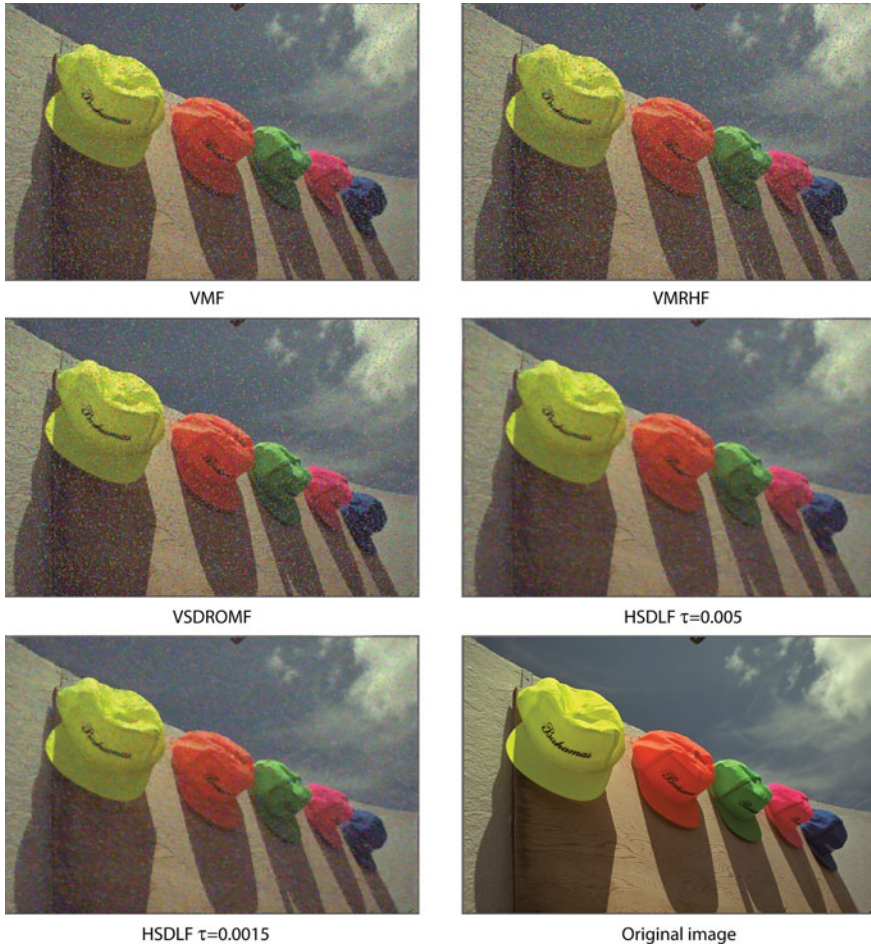
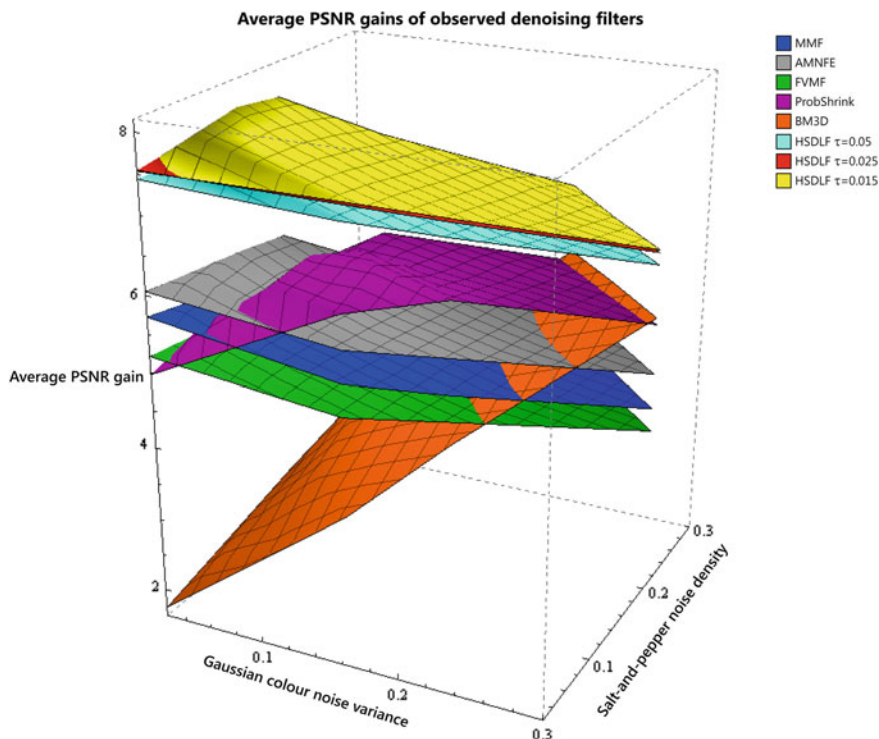


Fig. 9 (continued)

- Analogously to impulse noise, Figs. 11 and 12 show that HSDLF effectively preserves all image edges and details, and that it has noticeably less artefacts than other compared noise filters for all three observed values of  $\tau$ .

### 4.3 Computational Cost

In terms of practical use, it is very important to address the issue of HSDLF’s computational cost in comparison to other filtering methods. As mentioned in Sect. 3 and Sect. 4, HSDLF is neither dependent on the nature of the noise nor on the digital

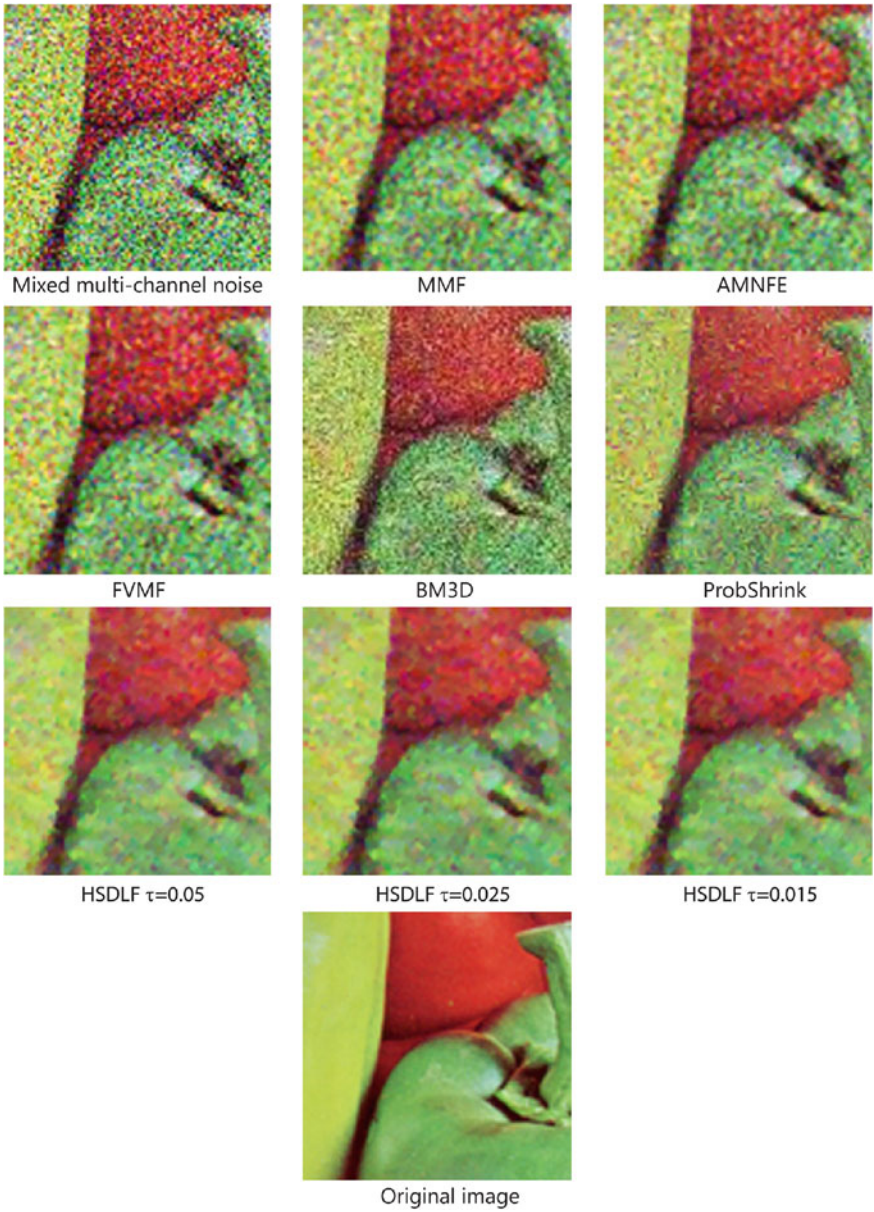


**Fig. 10** Average PSNR gains of HSDLF, MMF, AMNFE, FVMF, BM3D and ProbShrink applied to all benchmark images corrupted by various mixed noise powers (indicated on *horizontal axes*)

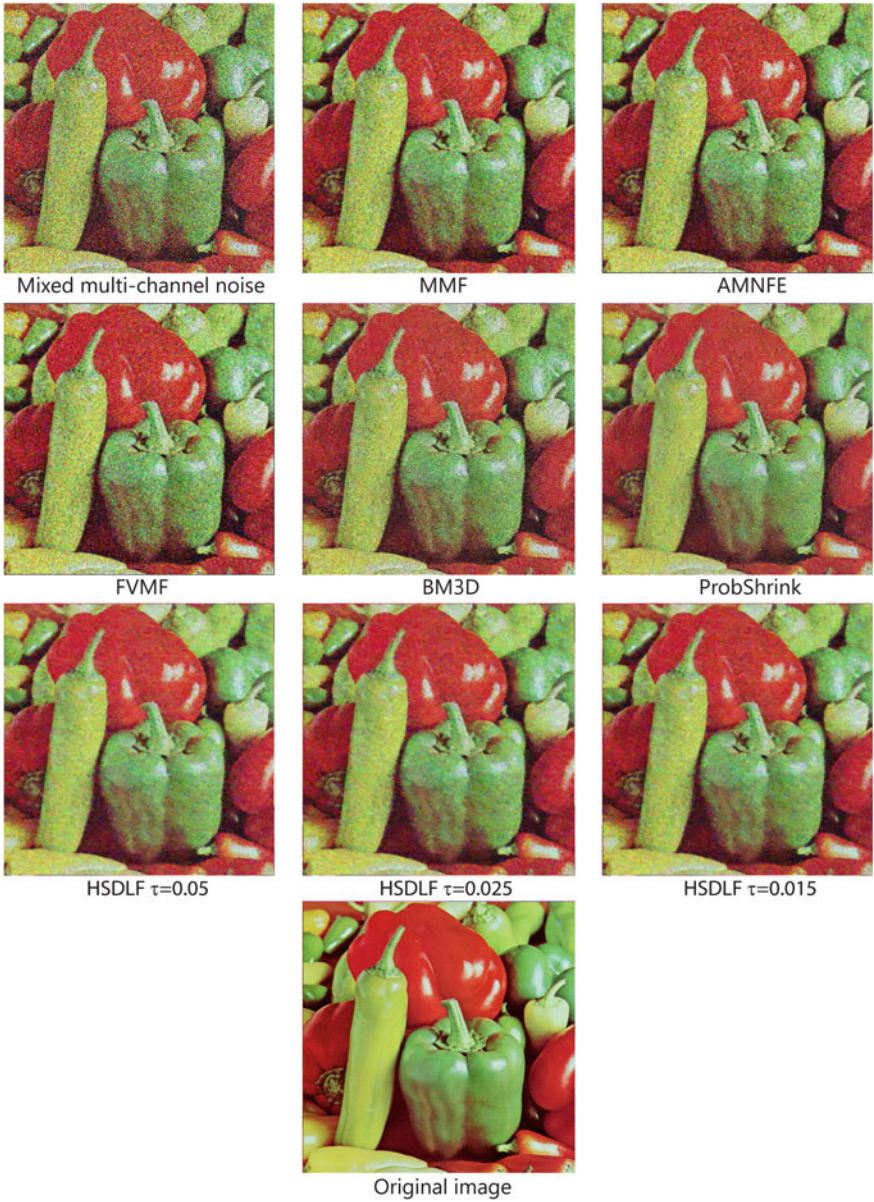
image format, so the HSDLF's computational cost is basically influenced only by the image size (apart from technical issues that affects all filters, like operating system, hardware etc.). Table 12 illustrates average computation costs of HSDLF (for all three observed values of  $\tau$ ) [3], ProbShrink [42], BM3D [11], MMF [4], AMNFE [43] and FVMF [8, 44, 45] applied to all benchmark images (classified by image size) corrupted by mixed multichannel noise.

Although HSDLF in its baseline form presented here affects every pixel in a processed image, its computational cost is lower than all observed DWT-based filters and some spatial domain filters (not presented in Table 12) for all observed values of  $\tau$ . HSDLF's computational cost can be additionally improved in a number of ways: by using fewer spatial directions  $m$  in DEEPLOC algorithm (see Sect. 3), or by further code optimisation or multi-thread programming [3]. Taken as a whole, the computational cost of HSDLF in its present form can be considered low with respect to the filter's high performances [2, 3].

It should be noted that all other compared filters have been used in their original formats. Thus, their computational costs have been calculated on their corresponding original programming platforms [2, 3].



**Fig. 11** Denoising results of HSDLF, MMF, AMNFE, FVMF, BM3D and ProbShrink applied to a segment of image “Peppers” ( $512 \times 512$ ) corrupted by mixture of salt-and-pepper noise with density 0.05 and AGCN with variance 0.1



**Fig. 12** Denoising results of HSDLF, MMF, AMNFE, FVMF, BM3D and ProbShrink applied to image “Peppers” ( $512 \times 512$ ) corrupted by mixture of salt-and-pepper noise with density 0.05 and AGCN with variance 0.1

**Table 12** Average computational cost of HSDLF, MMF, AMNFE, FVMF, BM3D and ProbShrink applied to all benchmark images corrupted by various mixed noise powers

Denoising method	Programming platform	Average computation cost (in seconds)		
		256 × 256 resolution images	512 × 512 resolution images	768 × 512 resolution images
MMF	C++	1.52	2.7	2.92
AMNFE	C	1.94	4.16	5.74
FVMF	C	2.1	4.23	8.18
BM3D	MATLAB	17.27	72.2	111.07
ProbShrink	MATLAB	12.62	39.36	53.86
HSDLF $\tau=0.05$	C++	7.04	19.91	28.31
HSDLF $\tau=0.025$	C++	6.97	19.87	28.23
HSDLF $\tau=0.015$	C++	6.94	19.75	28.1

Specifications of PC used in experiments: Pentium Prescott s478 32-bit 2.80 GHz processor (with enabled *HyperThreading*), 2GB of RAM, Microsoft Windows 7 Operating system

## 5 Conclusion

While depth functions have been extensively studied in nonparametric multivariate statistical inference, very few of these theoretical results have been practically used. DEEPLC algorithm is one of the very few methods for finding the deepest location within a  $d$ -dimensional data cloud, and accordingly halfspace depth function, which is an essential and most studied depth function in literature [49, 60]. Since HSDLF is based on a slightly modified version of the DEEPLC algorithm, it presents a novel approach in removal of impulse [2] and mixed [3] multichannel noise, and possibly, other types of multichannel noise. It inherently maintains spectral correlation between channels due to its multidimensional character, and does not depend on nature or distribution of noise, and/or any specific digital image format.

Compared to numerous observed spatial and transform domain state-of-the-art multichannel filters, HSDLF has shown outstanding results in terms of both objective error metrics criteria and visual quality of filtered images [2, 3]. As evidenced in the experiments on a comprehensive corpus of benchmark images corrupted by wide range of impulse and mixed multichannel noise powers, HSDLF effectively preserves most of the fine image details and edges without artefacts.

Potential enhancements of HSDLF could include further tweaking of the DEEPLC algorithm, which might improve HSDLF's precision and computational cost. These advances may be theoretical, e.g. selection of different spatial directions and/or their ratio, or programming, e.g. use of multi-thread or cloud computing.

**Acknowledgments** The authors would like to express their sincere gratitude to Dr. M. Emre Celebi for provision of source codes and programs of 25 state-of-the-art multichannel impulse noise filters. Without his support this research would be significantly slower and incomplete.

## References

1. Astola J, Haavisto P, Neuvo Y (1990) Vector median filters. *Proc IEEE* 78(4):678–689
2. Baljžović Dj, Kovačević B, Baljžović A (2012) Novel method for removal of multichannel impulse noise based on half-space deepest location. *J Electron Imaging* 21(1):013025. doi:[10.1117/1.JEI.21.1.013025](https://doi.org/10.1117/1.JEI.21.1.013025)
3. Baljžović Dj, Kovačević B, and Baljžović A (2013) Mixed noise removal filter for multichannel images based on halfspace deepest location. *IET Image Process* 7(4):310–323
4. Celebi ME, Aslandogan YA (2008) Robust switching vector median filter for impulsive noise removal. *J Electron Imaging* 17(4):043006
5. Celebi ME, Kingravi H, Aslandogan YA (2007) Non-linear vector filtering for impulsive noise removal from color images. *J Electron Imaging* 16(3):033008
6. Celebi ME, Kingravi H, Uddin B (2007) Fast switching filter for impulsive noise removal from color images. *Int J Imaging Syst Technol* 51(2):155–165
7. Celebi ME, Schaefer G, Zhou H (2010) A new family of order-statistics based switching vector filters. In: *Proceedings of IEEE international conference on image processing (ICIP 2010)*, pp 97–100, 26–29 Sept 2010
8. Chatzis V, Pitas I (1999) Fuzzy scalar and vector median filters based on fuzzy distances. *IEEE Trans Image Process* 8(5):731–734
9. Chen T, Ma K-K, Chen LH (1999) Tri-state median filter for image denoising. *IEEE Trans Image Process* 8(12):1834–1838
10. Cuesta-Albertos JA, Nieto-Reyes A (2008) The random Tukey depth. *Comput Stat Data Anal* 52:4979–4988
11. Dabov K, Foi A, Katkovnik V, Egiazarian K (2007) Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Trans Image Process* 16(8):2080–2095
12. Donoho DL, Gasko M (1992) Breakdown properties of location estimates based on halfspace depth and projected outlyingness. *Ann Stat* 20:1803–1827
13. González RC, Woods RE (2008) *Digital image processing*, 3rd edn. Prentice Hall, Trenton
14. Hancheng YLiZ, Wang H, (2009) Image denoising using trivariate shrinkage filter in the wavelet domain and joint bilateral filter in the spatial domain. *IEEE Trans Image Process* 18(10):2364–2369
15. Kannan K, Kanna BR, Aravindan C (2010) Root mean square filter for noisy images based on hyper graph model. *Image Vis Comput* 28:1329–1338
16. Karakos DG, Trahanias PE (1995) Combining vector median and vector directional filters: the directional distance filters. In: *Proceedings of IEEE ICIP conference*, pp 171–174
17. Karakos DG, Trahanias PE (1997) Generalized multichannel image filtering structures. *IEEE Trans Image Process* 6(7):1038–1045
18. Kenney C, Deng Y, Manjunath BS, Hewer G (2001) Peer group image enhancement. *IEEE Trans Image Process* 10(2):326–334
19. Khriji L, Gabbouj M (1999) A class of multichannel image processing filters. *Electron Lett* 35(4):285–287
20. Khriji L, Gabbouj M (1999) A new class of multichannel image processing filters: vector median-rational hybrid filters. *IEICE Trans Inf Syst* E82(12):1589–1596
21. Khriji L, Gabbouj M (2002) Adaptive fuzzy order statistics-rational hybrid filters for color image processing. *Fuzzy Sets Syst* 128(1):35–46
22. Khriji L, Gabbouj M (2000) Multichannel image processing using fuzzy vector median-rational hybrid filters. In: *Proceedings of EUSIPCO'00 conference*, pp 1345–1348
23. Khriji L, Gabbouj M (2004) Rational-based adaptive fuzzy filters. *Int J Comput Cognition* 2(1):113–132
24. Khriji L, Gabbouj M (1999) Vector median-rational hybrid filters for multichannel image processing. *IEEE Signal Process Lett* 6(7):186–190
25. Liu RY, Parelius J, Singh K (1999) Multivariate analysis by data depth: descriptive statistics, graphics and inference. *Ann Stat* 27:783–858 (with discussions)



26. Lukac R, Plataniotis KN (2006) A taxonomy of color image filtering and enhancement solutions. In: Hawkes PW (ed) *Advances in imaging and electron physics*, vol 140, Academic Press, San Diego, pp 187–264
27. Lukac R (2004) Adaptive color image filtering based on center-weighted vector directional filters. *Multidimens Syst Sign Process* 15(2):169–196
28. Lukac R (2003) Adaptive vector median filtering. *Pattern Recogn Lett* 24(12):1889–1899
29. Lukac R (2002) Color image filtering by vector directional order-statistics. *Patt Recog Image Anal* 12(3):279–285
30. Lukac R (2002) Optimised directional distance filter. *Mach Graphics Vision* 11(2/3):311–326
31. Lukac R, Plataniotis KN, Venetsanopoulos AN, Smolka B (2005) A statistically-switched adaptive vector median filter. *J Intell Robotic Syst* 42(4):361–391
32. Lukac R, Smolka B, Plataniotis KN, Venetsanopoulos AN (2006) Vector sigma filters for noise detection and removal in color images. *J Visual Commun Image Represent* 17(1):1–26
33. Lukac R, Smolka B, Plataniotis KN, Venetsanopoulos AN (2003) A variety of multichannel sigma filters. In: Salimbeni R (ed) *Optical metrology for arts and multimedia*. Proceedings of the SPIE, vol. 5146, pp 244–253
34. Lukac R, Smolka B, Plataniotis KN, Venetsanopoulos AN (2003) Generalized adaptive vector sigma filters. In: Proceedings of international conference on multimedia and expo (ICME'03), vol 1, pp 537–540
35. Lukac R, Smolka B, Plataniotis KN, Venetsanopoulos AN (2003) Entropy vector median filter. In: Proceedings of 1st Iberian conference on pattern recognition and image analysis (IbPRIA). Lecture notes in computer science, vol 2652, pp 1117–1125
36. Lukac R, Smolka B, Plataniotis KN, Venetsanopoulos AN (2003) Generalized entropy vector filters. In: Proceedings of 4th EURASIP EC-VIP-MC video image processing and multimedia communications conference, pp 239–244
37. Lukac R, Smolka B, Plataniotis KN, Venetsanopoulos AN, Zavarsky P (2003) Angular multichannel sigma filter. In: Proceedings of IEEE international conference on acoustics, speech, and signal processing (ICASSP'03), vol 3, pp 745–748
38. Ma Z, Wu HR, Feng D (2006) Partition-based vector filtering technique for suppression of noise in digital color images. *IEEE Trans Image Process* 15(8):2324–2342
39. Mizera I (2002) On depth and deep points: a calculus. *Ann Stat* 30:1681–1736
40. Moore MS, Gabbouj M, Mitra SK (1999) Vector SD-ROM filter for removal of impulse noise from color images. In: Proceedings of 2nd EURASIP conference focused on DSP for multimedia communication services (ECMCS'99)
41. Motwani MC, Gadiya MC, Motwani RC (2004) Survey of image denoising techniques. In: Proceedings of global signal processing expo and conference (GSPx '04), Santa Clara, 27 Sept 2004
42. Pizurica A, Philips W (2006) Estimating the probability of the presence of a signal of interest in multiresolution single- and multiband image denoising. *IEEE Trans Image Process* 15:654–665
43. Plataniotis KN, Androutsos D, Venetsanopoulos AN (1998) Adaptive multichannel filters for colour image processing. *Signal Process Image Commun* 11(3):171–177
44. Plataniotis KN, Androutsos D, Venetsanopoulos AN (1999) Adaptive fuzzy systems for multichannel signal processing. *Proc IEEE* 87(9):1601–1622
45. Plataniotis KN, Androutsos D, Venetsanopoulos AN (1996) Fuzzy adaptive filters for multichannel image processing. *Sign Process* 55(1):93–106
46. Portilla J, Strela V, Wainwright M, Simoncelli E (2003) Image denoising using Gaussian scale mixtures in the wavelet domain. *IEEE Trans Image Process* 12(11):1338–1351
47. Romanazzi M (2009) Data depth, random simplices and multivariate dispersion. *Stat Prob Lett* 79:1473–1479
48. Rousseeuw PJ, Ruts I (1998) Constructing the bivariate Tukey median. *Stat Sinica* 8:827–839
49. Rousseeuw PJ, Struyf A (1998) Computing location depth and regression depth in higher dimensions. *Statist Comput* 8:193–203
50. Safari MS, Aghagolzadeh A (2007) FIR filter based fuzzy-genetic mixed noise removal. In: Proceedings of international conference of information sciences, signal processing and their applications (ISSPA '07), vol 1–4, pp 1006–1289

51. Schulte S, Witte VD, Kerre EE (2007) A fuzzy noise reduction method for color images. *IEEE Trans Image Process* 16(5):1425–1436
52. Selesnick IW, Baraniuk RG, Kingsbury NG (2005) The dual-tree complex wavelet transform. *IEEE Signal Proc Mag* 22(6):123–151
53. Small CG (1990) A survey of multidimensional medians. *Int Stat Rev* 58:263–277
54. Smolka B, Chydzinski A (2005) Fast detection and impulsive noise removal in color images. *Real-Time Imag* 11(5/6):389–402
55. Smolka B, Plataniotis KN (2005) Soft-switching adaptive technique of impulsive noise removal in color images. In: Proceedings of 2nd international conference on image analysis and recognition (ICIAR 2005). Lecture notes on computer science, vol 3656, pp 686–693
56. Smolka B, Bieda R, Plataniotis KN, Lukac R (2005) Adaptive soft-switching filter for impulsive noise suppression in color images. In: Proceedings of European signal processing conference (EUSIPCO'05)
57. Smolka B, Lukac R, Plataniotis KN, Venetsanopoulos AN (2003) Application of kernel density estimation for color image filtering. In: Proceedings of visual communication and image processing conference (VCIP'03). SPIE, vol 5150, pp 1650–1656
58. Smolka B, Plataniotis KN, Lukac R, Venetsanopoulos AN (2003) New class of impulsive noise reduction filters based on kernel density estimation. In: Proceedings of the 28th IEEE international conference on acoustics, speech and signal process (ICASSP'03), vol 3, pp 721–724
59. Smolka B, Plataniotis KN, Lukac R, Venetsanopoulos AN (2003) Kernel density estimation based impulsive noise reduction filter. In: Proceedings of IEEE international conference on image processing (ICIP'03), vol 2, pp 137–140
60. Struyf A, Rousseeuw PJ (2000) High-dimensional computation of the deepest location. *Comput Stat Data Anal* 34:415–426
61. Trahanias PE, Venetsanopoulos AN (1993) Vector directional filters: a new class of multichannel image processing filters. *IEEE Trans Image Process* 2(4):528–534
62. Tu Y, Li S, Wang M (2008) Mixed-noise removal for color images using modified PCNN Model. In: Proceedings of international symposium on intelligent information technology application (IITA '08), vol 3, pp 347–351
63. Tukey JW (1975) Mathematics and the picturing of data. In: James RD (ed) Proceedings of international congress of mathematics, Vancouver 1974, vol 2, pp 523–531
64. Zhang J (2002) Some extensions of Tukey's depth function. *J Multivariate Anal* 82:134–165
65. Zuo Y, Serfling R (2000) General notions of statistical depth function. *Ann Stat* 28(2):461–482

# Spatially Adaptive Color Image Processing

Johan Debayle and Jean-Charles Pinoli

**Abstract** This chapter is focused on spatially adaptive image processing for color images in the context of the General Adaptive Neighborhood Image Processing (GANIP) approach. The GANIP was first defined for gray-tone images and is here extended to color images. A set of local adaptive neighborhoods is defined for each image point, depending on the color intensity function of the image. These adaptive neighborhoods are then used as spatially adaptive operational windows for defining adaptive Choquet filters and adaptive morphological filters. The resulting adaptive operators are successfully applied and compared with the classical operators for image restoration, enhancement and segmentation of color images.

**Keywords** Color adaptive neighborhood · Choquet filtering · General adaptive neighborhood · Image enhancement · Image restoration · Image segmentation · Mathematical morphology

## 1 Introduction

### *1.1 The General Adaptive Neighborhood Image Processing (GANIP) Approach*

The General Adaptive Neighborhood paradigm has been introduced [11] in order to propose an original image representation for adaptive processing and analysis of

---

J. Debayle (✉) · J.-C. Pinoli  
Ecole Nationale Supérieure des Mines, LGF UMR CNRS 5307, 158 cours Fauriel,  
42023 Saint-Etienne, France  
e-mail: debayle@emse.fr

J.-C. Pinoli  
e-mail: pinoli@emse.fr

gray-tone images. The central idea is the notion of adaptivity which is simultaneously associated to the analyzing scales, the spatial structures and the intensity values of the image to be addressed.

In the so-called General Adaptive Neighborhood Image Processing (GANIP) approach [11, 12], a set of General Adaptive Neighborhoods (GANs set) is identified around each point in the image to be analyzed. A GAN is a subset of the spatial support constituted by connected points whose measurement values, in relation to a selected criterion (such as luminance, contrast or thickness), fit within a specified homogeneity tolerance. These GANs are used as adaptive windows for further gray-tone image transformations [15, 30] such as Choquet filters [13, 14] or morphological filters [11, 29, 31].

The aim of this chapter is to extend these neighborhoods to color images for further defining spatially adaptive color image transformations.

## 1.2 Color Spaces

The most simple way to manipulate color images is to work in the RGB color space but it presents some drawbacks such as correlated components or non-uniformity [21, 28]. In this way, the HSL representation [20] with the hue, saturation and luminance or the CIE  $L^*a^*b^*$  representation have been introduced to overcome these limitations [8].

In this chapter, both the RGB,  $L^*a^*b^*$  and HSL (hue/saturation/luminance) color space representations will be used for defining spatially adaptive color image filters. The CIE  $L^*a^*b^*$  color space is deduced from the CIE XYZ color space:

$$\begin{cases} L^* = 116f(Y/Y_n) - 16, \\ a^* = 500(f(X/X_n) - f(Y/Y_n)), \\ b^* = 200(f(Y/Y_n) - f(Z/Z_n)). \end{cases} \quad (1)$$

where:

$$f(t) = \begin{cases} t^{\frac{1}{3}} & \text{if } t > \left(\frac{6}{29}\right)^3, \\ \frac{1}{3} \left(\frac{29}{6}\right)^2 t + \frac{4}{29} & \text{otherwise.} \end{cases} \quad (2)$$

Here  $X_n$ ,  $Y_n$  and  $Z_n$  are the CIE XYZ tristimulus values of the reference white point corresponding to the illuminant  $D65$ .

The HSL space is derived from the RGB space by the following equations:

$$\left\{ \begin{array}{l} H = 60^\circ \times \left\{ \begin{array}{ll} \frac{G - B}{\max(R, G, B) - \min(R, G, B)} & \text{if } R = \max(R, G, B), \\ \frac{B - R}{\max(R, G, B) - \min(R, G, B)} + 2 & \text{if } G = \max(R, G, B), \\ \frac{R - G}{\max(R, G, B) - \min(R, G, B)} + 4 & \text{if } B = \max(R, G, B). \end{array} \right. \\ S = \left\{ \begin{array}{ll} \frac{\max(R, G, B) - \min(R, G, B)}{\max(R, G, B) + \min(R, G, B)} & \text{if } L \leq 0.5, \\ \frac{\max(R, G, B) + \min(R, G, B)}{\max(R, G, B) - \min(R, G, B)} & \text{if } L > 0.5. \end{array} \right. \\ L = \frac{2 - \max(R, G, B) - \min(R, G, B)}{\max(R, G, B) + \min(R, G, B)} \end{array} \right. \quad (3)$$

### 1.3 Need of Vector Order Relations

Rank-order filters (such as morphological filters or Choquet filters) need the use of an order relation between the intensities to be processed. The application of rank-order filtering to color images is not straightforward due to the vectorial nature of the color data. Several vector order relations have been proposed in the literature such as marginal ordering, lexicographical ordering, partial ordering and reduced ordering [2, 6]. For example, let  $E = A \times B \times C$  be a color space where  $A, B, C$  are three sets of scalar values. The lexicographical order on  $E$ , denoted  $<_E$ , with the component ordering  $A \rightarrow B \rightarrow C$  is defined as following for two color vectors  $c_1 = (c_1^A, c_1^B, c_1^C)$  and  $c_2 = (c_2^A, c_2^B, c_2^C)$ :

$$c_1 <_E c_2 \Leftrightarrow \left\{ \begin{array}{l} c_1^A < c_2^A \text{ or,} \\ c_1^A = c_2^A \text{ and } c_1^B < c_2^B \text{ or,} \\ c_1^A = c_2^A \text{ and } c_1^B = c_2^B \text{ and } c_1^C < c_2^C. \end{array} \right. \quad (4)$$

Concerning the vectorial ordering, the lexicographical one will be used in this chapter by fixing the ordering of the components for the three color space representations as follows:

- in RGB:  $R \rightarrow G \rightarrow B$  or  $G \rightarrow R \rightarrow B$
- in  $L^*a^*b^*$ :  $L^* \rightarrow a^* \rightarrow b^*$
- in HSL:  $(H \div h_0) \rightarrow S \rightarrow L$  or  $L \rightarrow S \rightarrow (H \div h_0)$  where  $h_0$  corresponds to the origin of hues and  $\div$  denotes the angular difference defined as:

$$h_1 \div h_2 = \left\{ \begin{array}{ll} |h_1 - h_2| & \text{if } |h_1 - h_2| \leq 180^\circ, \\ 360^\circ - |h_1 - h_2| & \text{if } |h_1 - h_2| > 180^\circ. \end{array} \right. \quad (5)$$

In the literature, several approaches combining color representations and order relations have been proposed for defining nonlinear color image filters [2, 4, 5, 19, 27, 32, 37–39].

## 2 Color Adaptive Neighborhoods (CANs)

This chapter deals with 2D color images, that is to say image mappings defined on a spatial support  $D$  in the Euclidean space  $\mathbb{R}^2$  and valued onto a color space  $E \subseteq \mathbb{R}^3$  (such as RGB,  $L^*a^*b^*$ , HSL...). The set of color images is denoted  $\mathcal{I}$ .

### 2.1 Definition

In order to process color images within the GANIP framework, it is first necessary to define Color Adaptive Neighborhoods (CANs) in the most simple way.

For each point  $x \in D$  and for a color image  $f_0 \in \mathcal{I}$ , called the *pilot image*, the CANs denoted  $V_m^{f_0}(x)$  are subsets in  $D$ . They are built upon  $f_0$  in relation with a *homogeneity tolerance*  $m$  belonging to the positive real value range  $\mathbb{R}^+$ . More precisely,  $V_m^{f_0}(x)$  is a subset of  $D$  which fulfills two conditions:

1. its points have a color value close to that of the point  $x$ :

$\forall y \in V_m^{f_0}(x) \quad \|f_0(y) - f_0(x)\|_E \leq m$ , where  $\|\cdot\|_E$  denotes a functional on the color space  $E$ . By using the RGB,  $L^*a^*b^*$  and HSL color spaces, the functionals are defined as [2]:

$$\|c_1 - c_2\|_{RGB} = \sqrt{(c_1^R - c_2^R)^2 + (c_1^G - c_2^G)^2 + (c_1^B - c_2^B)^2} \quad (6)$$

$$\|c_1 - c_2\|_{L^*a^*b^*} = \sqrt{(c_1^{L^*} - c_2^{L^*})^2 + (c_1^{a^*} - c_2^{a^*})^2 + (c_1^{b^*} - c_2^{b^*})^2} \quad (7)$$

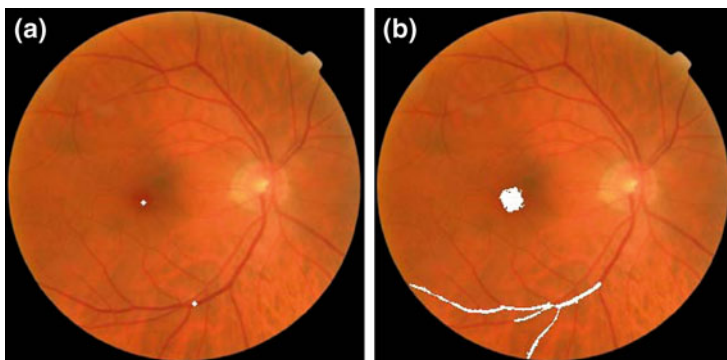
$$\|c_1 - c_2\|_{HSL} = \sqrt{(c_1^L - c_2^L)^2 + (c_1^S)^2 + (c_2^S)^2 - 2c_1^S c_2^S \cos(c_1^H - c_2^H)} \quad (8)$$

2. the set is path-connected with the usual Euclidean topology on  $D \subseteq \mathbb{R}^2$  (a set  $X$  is connected if for all  $x_1, x_2 \in X$  there exists a continuous mapping  $\tau : [0, 1] \rightarrow X$  such that  $\tau(0) = x_1$  and  $\tau(1) = x_2$ ).

The CANs are thus mathematically defined as following:

$$\forall(m, f_0, x) \in \mathbb{R}^+ \times \mathcal{I} \times D$$

$$V_m^{f_0}(x) = C_{\{y \in D; \|f_0(y) - f_0(x)\|_E \leq m\}}(x) \quad (9)$$



**Fig. 1** **a** Original image  $f_0$  with two seed points  $x$  and  $y$  (white dots). **b** CANs  $V_{25}^{f_0}(x)$  and  $V_{25}^{f_0}(y)$ . The CANs of the two selected points of the original color image  $f_0$  are homogeneous with the tolerance  $m = 10$  in the RGB color space

where  $C_X(x)$  denotes the path-connected component (with the usual Euclidean topology on  $D \subseteq \mathbb{R}^2$ ) of  $X \subseteq D$  containing  $x \in D$ .

The definition of  $C_X(x)$  ensures that  $x \in V_m^{f_0}(x)$  for all  $x \in D$ .

## 2.2 Illustration

Figure 1 illustrates the CANs of two points computed on a human retina image (used as the pilot image  $f_0$ ). The figure highlights the homogeneity and the correspondence of the CANs with the spatial structures.

## 2.3 Properties

These color adaptive neighborhoods satisfy several properties:

1. reflexivity:

$$x \in V_m^{f_0}(x) \quad (10)$$

2. increasing with respect to  $m$ :

$$\left( \begin{array}{l} (m_1, m_2) \in \mathbb{R}^+ \times \mathbb{R}^+ \\ m_1 \leq m_2 \end{array} \right) \Rightarrow V_{m_1}^{f_0}(x) \subseteq V_{m_2}^{f_0}(x) \quad (11)$$

3. equality between iso-valued points:

$$\left( \begin{array}{l} (x, y) \in D^2 \\ x \in V_m^{f_0}(y) \\ f_0(x) = f_0(y) \end{array} \right) \Rightarrow V_m^{f_0}(x) = V_m^{f_0}(y) \tag{12}$$

The proofs of these properties are similar to those stated for gray-tone images [11].

### 3 CAN Choquet Filtering

Fuzzy integrals [9, 36] provide a general representation of image filters. A large class of operators can be represented by those integrals such as linear filters, morphological filters, rank filters, order statistic filters or stack filters. The main fuzzy integrals are Choquet integral [9] and Sugeno integral [36]. Fuzzy integrals integrate a real function with respect to a fuzzy measure.

#### 3.1 Fuzzy Integrals

Let  $X$  be a finite set. In discrete image processing applications,  $X$  represents the  $K$  pixels within a subset of the spatial support of the image (an image window). A fuzzy measure,  $\mu$ , over  $X = \{x_0, \dots, x_{K-1}\}$  is a function  $\mu : 2^X \rightarrow [0, 1]$  such that:

- $\mu(\emptyset) = 0; \mu(X) = 1$
- $\mu(A) \leq \mu(B)$  if  $A \subseteq B$

Fuzzy measures are generalizations of probability measures for which the probability of the union of two disjoint events is equal to the sum of the individual probabilities.

The discrete Choquet integral of a function  $f : X = \{x_0, \dots, x_{K-1}\} \rightarrow E \subseteq \mathbb{R}$  with respect to the fuzzy measure  $\mu$  is [26]:

$$C_\mu(f) = \sum_{i=0}^{K-1} (f(x_{(i)}) - f(x_{(i-1)})) \mu(A_{(i)}) = \sum_{i=0}^{K-1} (\mu(A_{(i)}) - \mu(A_{(i+1)})) f(x_{(i)}) \tag{13}$$

where the subsymbol  $(.)$  indicates that the indices have been permuted so that:  $0 = f(x_{(-1)}) \leq f(x_{(0)}) \leq f(x_{(1)}) \leq \dots, \leq f(x_{(K-1)})$ ,  $A_{(i)} = \{x_{(i)}, \dots, x_{(K-1)}\}$  and  $A_{(K)} = \emptyset$ .

An interesting property of the Choquet fuzzy integral is that if  $\mu$  is a probability measure, the fuzzy integral is equivalent to the classical Lebesgue integral and simply computes the expectation of  $f$  with respect to  $\mu$  in the usual probability framework.



The fuzzy integral is a form of averaging operator in the sense that the value of a fuzzy integral is between the minimum and maximum values of the function  $f$  to be integrated.

### 3.2 Classical Choquet Filters

Let  $f$  be a color image in  $\mathcal{I}$ ,  $W$  a window of  $K$  points and  $\mu$  a fuzzy measure defined on  $W$ . This measure could be extended to all translated window  $W_y$  associated to a pixel  $y$ :  $\forall A \subseteq W_y \mu(A) = \mu(A_{-y})$ ,  $A_{-y} \subseteq W$ . In this way, the Choquet filter associated to  $f$  is defined by:

$$\forall y \in D \quad CF_{\mu}^W(f)(y) = \sum_{x_i \in W_y} (\mu(A_{(i)}) - \mu(A_{(i+1)}))f(x_{(i)}) \quad (14)$$

where the subsymbol  $(.)$  indicates that the indices have been permuted so that:  $0 = f(x_{(-1)}) \prec_E f(x_{(0)}) \prec_E f(x_{(1)}) \prec_E \dots \prec_E f(x_{(K-1)})$ ,  $A_{(i)} = \{x_{(i)}, \dots, x_{(K-1)}\}$  and  $A_{(K)} = \emptyset$ .

Note that  $(\mu(A_{(i)}) - \mu(A_{(i+1)}))f(x_{(i)})$  corresponds to the multiplication by  $(\mu(A_{(i)}) - \mu(A_{(i+1)}))$  of each component of  $f(x_{(i)})$  (which is an element of the color space  $E$ ). The same mapping is realized for the sum  $\sum$ . In this way, the resulting value  $CF_{\mu}^W(f)(y)$  is guaranteed to be in  $E$ .

The Choquet filters generalize [18] several classical filters:

- linear filters (mean, Gaussian, ...):  
 $LF_W^{\alpha}(f)(y) = \sum_{x_i \in W_y} \alpha_i f(x_i)$  where  $\alpha \in [0, 1]^K$ ,  $\sum_{i=0}^{K-1} \alpha_i = 1$
- rank filters (median, min, max, ...):  
 $RF_W^d(f)(y) = f(x_{(d)})$  where  $d \in [0, K - 1] \cap \mathbb{N}$
- order filters ( $n$ -power,  $\alpha$ -trimmed mean, quasi midrange, ...):  
 $OF_W^{\alpha}(f)(y) = \sum_{x_i \in W_y} \alpha_i f(x_{(i)})$  where  $\alpha \in [0, 1]^K$ ,  $\sum_{i=0}^{K-1} \alpha_i = 1$

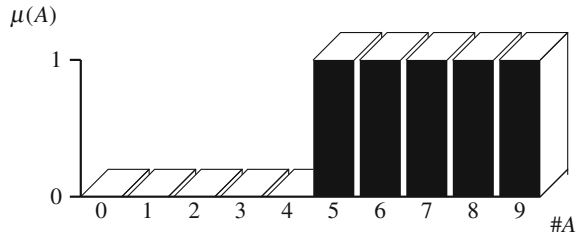
The mean, rank and order filters are Choquet filters with respect to the so-called cardinal measures:  $\forall A, B \subseteq W \#A = \#B \Rightarrow \mu(A) = \mu(B)$  ( $\#B$  denoting the cardinal of  $B$ ). Those filters, using an operational window  $W$ , could be characterized with the application:  $\#A \mapsto \mu(A)$ ,  $A \subseteq W$ . Indeed, different cardinal measures could be defined for each class of filters:

- mean filter:  $\mu$  is the fuzzy measure on  $W$  defined by  $\mu(A) = \#A/\#W$
- rank filters: (of order  $d$ ):  $\mu$  is the fuzzy measure on  $W$  defined by:  

$$\mu(A) = \begin{cases} 0 & \text{if } \#A \leq \#W - d, \\ 1 & \text{otherwise.} \end{cases}$$
- order filters:  $\mu$  is the fuzzy measure on  $W$  defined by  $\mu(A) = \sum_{j=0}^{\#A-1} \alpha_{\#W-j}$

In this way, there is a natural link between the weights  $\alpha_i$  of the general order filters and the fuzzy cardinal measures:  $\alpha_{\#W-i+1} = \mu_i - \mu_{i-1}$  where  $\mu_i$  corresponds to the fuzzy measure of the set with cardinal  $\#i$ .

**Fig. 2** Fuzzy measure of the classical median filter on a  $3 \times 3$  operational window [16]. The corresponding weights of this order filter are equal to  $\alpha_5 = \mu_5 - \mu_4 = 1$  and 0 otherwise



For example, the median filter (using a  $3 \times 3$  window) is characterized by the following cardinal measure (Fig. 2):

$$\forall A \subseteq W \quad \mu(A) = \begin{cases} 0 & \text{if } \#A \leq \lfloor \#W/2 \rfloor, \\ 1 & \text{otherwise.} \end{cases}$$

where  $\lfloor z \rfloor$  denotes the largest integer not greater than  $z$  (floor).

The Fig. 3 shows an illustration of several classical filters within the RGB color space, using the square of size  $7 \times 7$  as operational window and the lexicographical order  $R \rightarrow G \rightarrow B$ . The filters are performed on a painting image of the artist Gamze Aktan.

### 3.3 Adaptive Choquet Filters

In order to extend the Choquet filters with the use of CANs, the neighborhoods  $V_m^{f_0}(y)$  are used as operational windows  $W$ . Since the CANs are spatially-variant, a set of fuzzy measures has to be locally determined:  $\{\mu_y : V_m^{f_0}(y) \rightarrow [0, 1]\}_{y \in D}$ . In this way, the CAN-based Choquet filter is defined as follows:

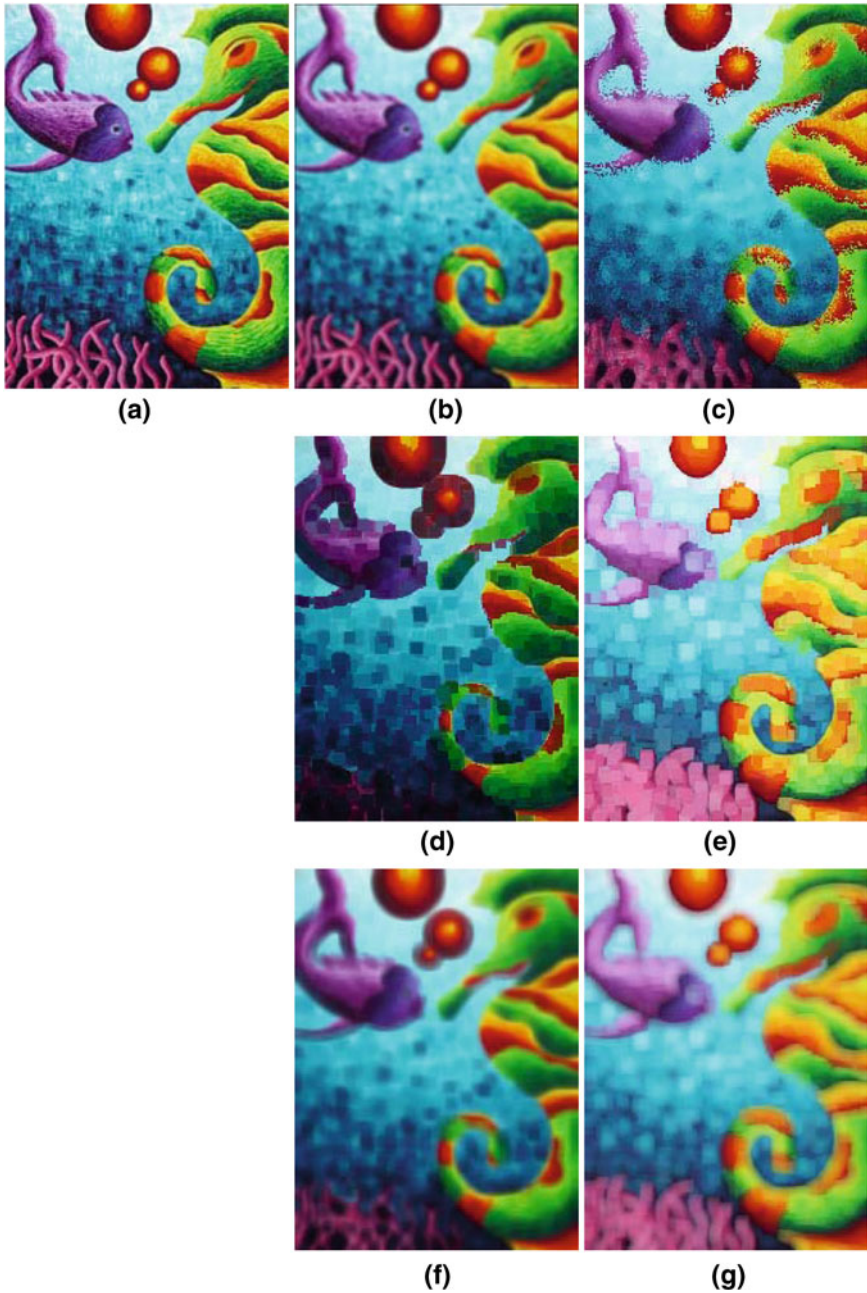
$$\forall (m, f_0, f, y) \in \mathbb{R}^+ \times \mathcal{S} \times \mathcal{S} \times D$$

$$CF_m^{f_0}(f)(y) = \sum_{x_i \in V_m^{f_0}(y)} (\mu_y(A_{(i)}) - \mu_y(A_{(i+1)})) f(x_{(i)}) \quad (15)$$

Several filters [18], such as the mean filter, the median filter, the min filter, the max filter, the  $\alpha$ -trimmed mean filter, the  $n$ -power filter, the  $\alpha$ -quasi-midrange filter and so on, could consequently be extended to CAN-based Choquet filters [1]. In the following (Fig. 4-9), a few fuzzy measures  $\mu_y$  attached to the CAN  $V_m^{f_0}(y)$  are illustrated with respect to specific CAN-based filters.

- CAN-based mean filter:

$$\forall A \subseteq V_m^{f_0}(y) \quad \mu_y(A) = \frac{\#A}{K}, \quad \text{where } K = \#V_m^{f_0}(y) \quad (16)$$



**Fig. 3** **a** Original image. **b** Classical mean filtering. **c** Classical median filtering. **d** Classical min filtering. **e** Classical max filtering. **f** Classical 3-power filtering. **g** Classical  $\frac{1}{3}$ -power filtering. Several classical Choquet filters within the RGB color space, using the square of size  $7 \times 7$  as operational window and the lexicographical order  $R \rightarrow G \rightarrow B$

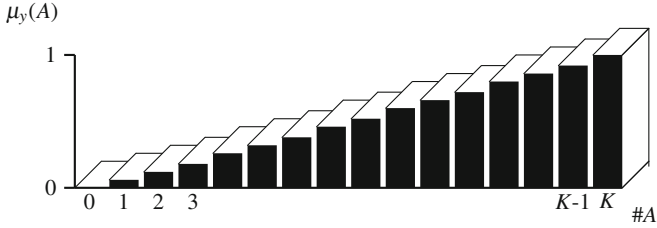


Fig. 4 Fuzzy measure of the adaptive mean filter

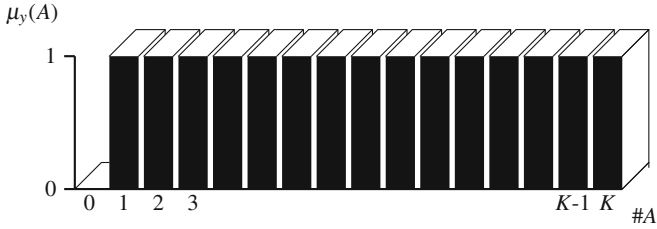


Fig. 5 Fuzzy measure of the adaptive max filter

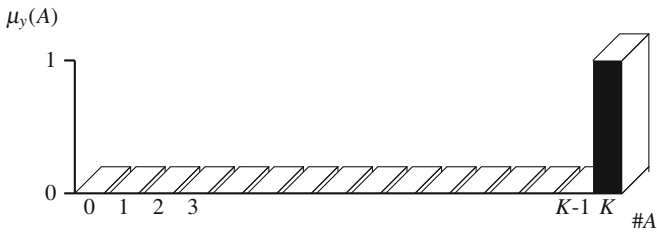


Fig. 6 Fuzzy measure of the adaptive min filter.

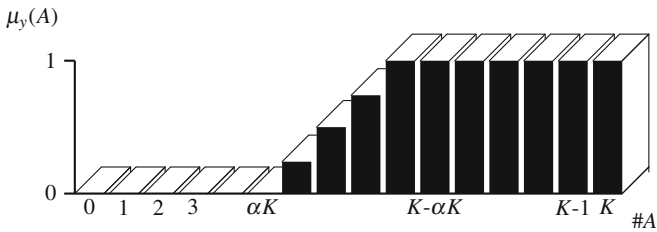


Fig. 7 Fuzzy measure of the adaptive  $\alpha$ -trimmed mean filter

- CAN-based max filter:

$$\forall A \subseteq V_m^{f_0}(y) \quad \mu_y(A) = \begin{cases} 0 & \text{if } \#A = 0, \\ 1 & \text{otherwise.} \end{cases}, \quad \text{where } K = \#V_m^{f_0}(y) \quad (17)$$

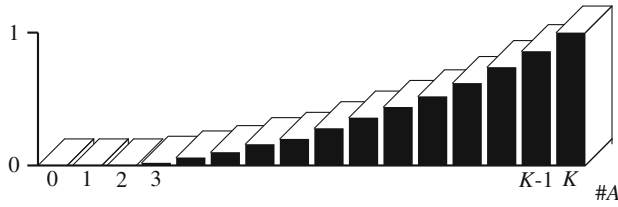


Fig. 8 Fuzzy measure of the adaptive  $n$ -power filter

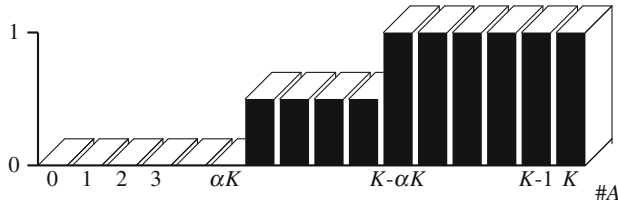


Fig. 9 Fuzzy measure of the adaptive  $\alpha$ -quasi-midrange filter

- CAN-based min filter:

$$\forall A \subseteq V_m^{f_0}(y) \quad \mu_y(A) = \begin{cases} 1 & \text{if } \#A = K, \\ 0 & \text{otherwise.} \end{cases}, \quad \text{where } K = \#V_m^{f_0}(y) \quad (18)$$

- CAN-based  $\alpha$ -trimmed mean filter:

$$\forall A \subseteq V_m^{f_0}(y), \forall \alpha \in [0, 0.5]$$

$$\mu_y(A) = \begin{cases} 0 & \text{if } \#A \leq \alpha K, \\ 1 & \text{if } \#A \geq K - \alpha K, \\ \frac{\#A - \alpha K + 1}{K(1 - 2\alpha)} & \text{otherwise.} \end{cases}, \quad \text{where } K = \#V_m^{f_0}(y) \quad (19)$$

- CAN-based  $n$ -power filter:

$$\forall A \subseteq V_m^{f_0}(y), \forall n \in [1, +\infty[ \quad \mu_y(A) = \left(\frac{\#A}{K}\right)^n, \quad \text{where } K = \#V_m^{f_0}(y) \quad (20)$$

- CAN-based  $\alpha$ -quasi-midrange filter

$$\forall A \subseteq V_m^{f_0}(y), \forall \alpha \in [0, 0.5]$$

$$\mu_y(A) = \begin{cases} 0 & \text{if } |A| \leq \alpha K, \\ 0.5 & \text{if } \alpha K < |A| < K - \alpha K, \\ 1 & \text{if } |A| \geq K - \alpha K. \end{cases} \quad (21)$$

where  $K = |V_m^{f_0}(y)|$ .

Some details on the spatially adaptive Choquet filters for gray-tone images can be found in [13, 14].

Those CAN-based Choquet filters provide image processing operators, in a well-defined mathematical framework.

A practical illustration of such filters is proposed in the Fig. 10. They are applied within the RGB color space, using the CANs with the homogeneity tolerance  $m = 80$  as operational window and the lexicographical order  $R \rightarrow G \rightarrow B$ . The proposed filters are performed on a painting image from the artist Gamze Aktan. This figure can be compared with Fig. 3 for viewing the differences between the classical filters and the proposed spatially adaptive filters.

### 3.4 Illustration Examples

Figure 11 shows a comparison of classical and adaptive mean filtering in RGB, L\*a\*b\* and HSL color spaces using the square of size  $7 \times 7$  and the CANs with the homogeneity tolerance  $m = 100$  as operational window.

The CAN-based morphological filters are more efficient than the classical morphological filters. Indeed, the spatial structures are rapidly damaged with blurring effects by using the classical morphology. On the contrary, the resulting images are smoothed using adaptive filters while preserving some transitions and details.

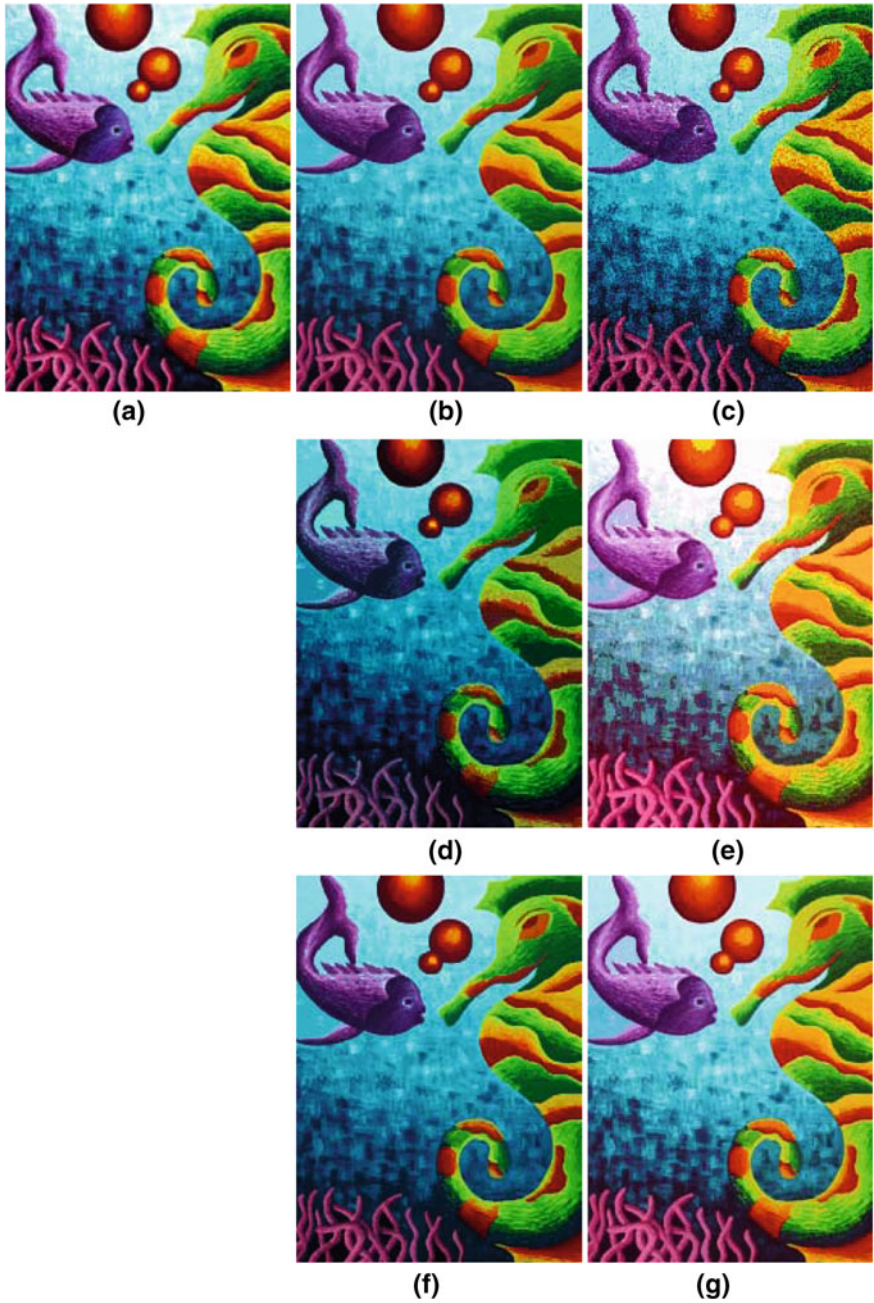
Figure 12 shows the impact of the ordering relation of the components for the lexicographical order. Adaptive  $n$ -power filtering is performed on a painting image of the artist Gamze Aktan in HSL using different ordering relations:  $L \rightarrow S \rightarrow H$  and  $H \rightarrow S \rightarrow L$  with  $h_0 = 0$ ,  $H \rightarrow S \rightarrow L$  with  $h_0 = 0.5$ . The CANs are calculated with the homogeneity tolerance  $m = 100$ .

The resulting images are indeed very different according to the ordering relation of the color components.

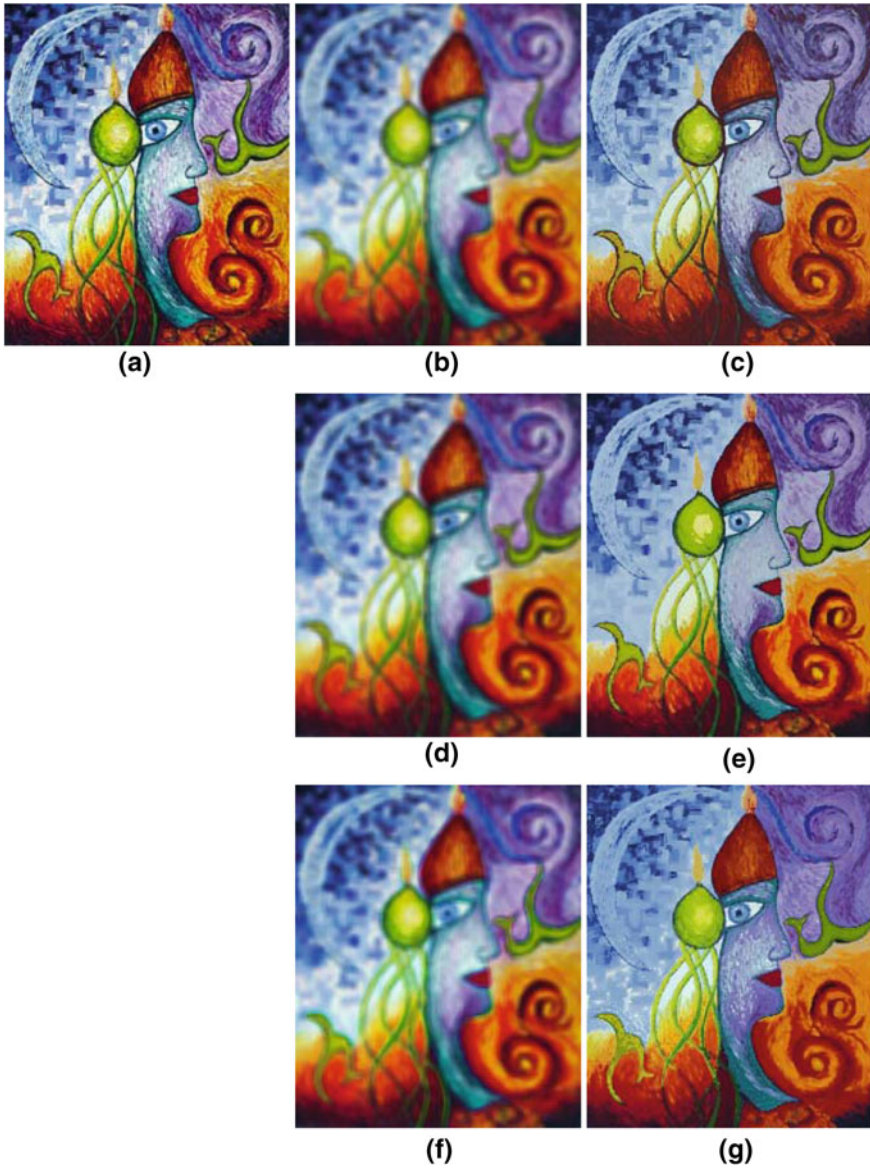
## 4 CAN Morphological Filtering

The origin of Mathematical Morphology stems from the study of the geometry of porous media by Matheron [25] who proposed the first morphological transformations for investigating the geometry of the objects of a binary image. MM can be defined as a theoretical framework for the analysis of spatial structures [34] characterized by a cross-fertilization among applications, methodologies, theories, and algorithms. It leads to several processing tools with the goal of image filtering, image segmentation and classification, image measurement, pattern recognition, or texture analysis and synthesis [35].

In the literature, several approaches have been proposed for color mathematical morphology [2, 4, 19, 27].

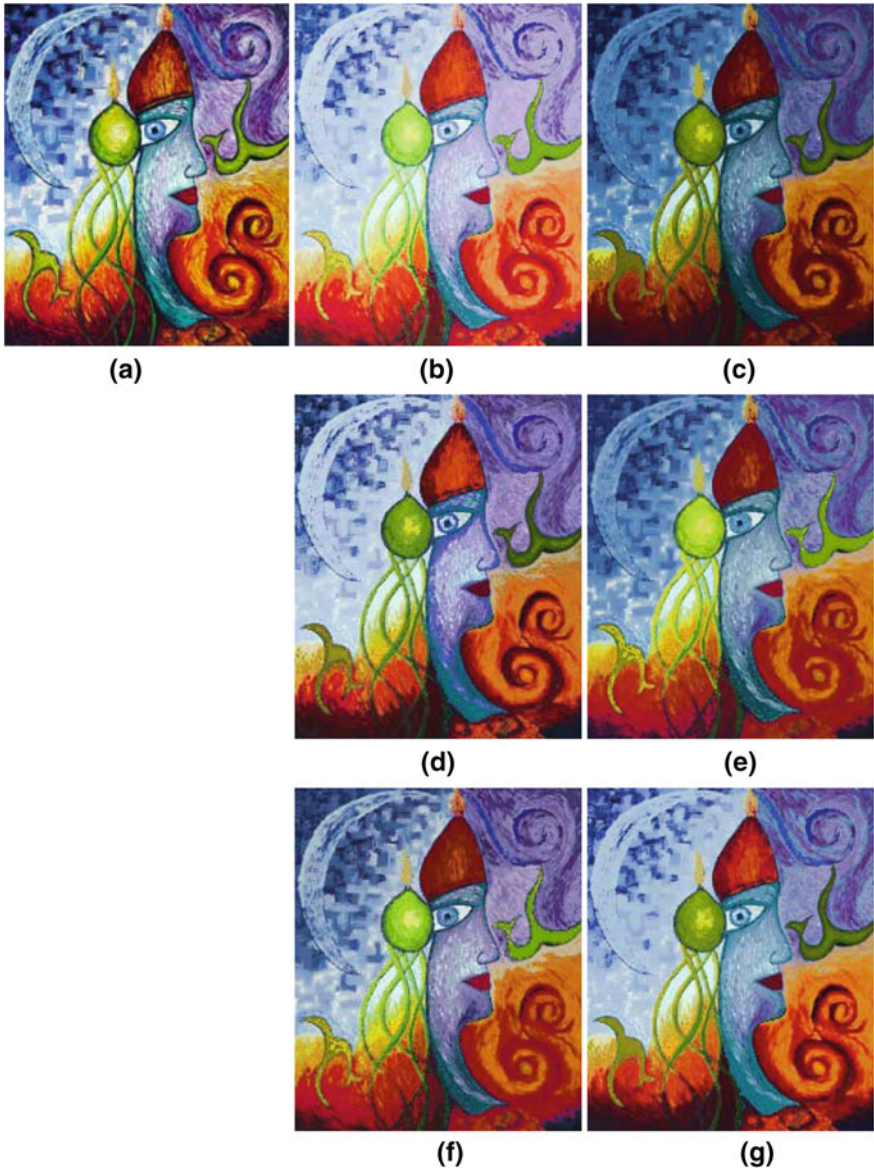


**Fig. 10** **a** Original image. **b** Adaptive mean filtering. **c** Adaptive median filtering. **d** adaptive min filtering. **e** Adaptive max filtering. **f** Adaptive 3-power filtering. **g** Adaptive  $\frac{1}{3}$ -power filtering. Several adaptive Choquet filters within the RGB color space, using the CANs with the homogeneity tolerance  $m = 80$  as operational window and the lexicographical order  $R \rightarrow G \rightarrow B$ .



**Fig. 11** **a** Original image. **b** RGB, classical mean. **c** RGB, adaptive mean. **d**  $L^*a^*b^*$ , classical mean. **e**  $L^*a^*b^*$ , adaptive mean. **f** HSL, classical mean. **g** HSL, adaptive mean. Classical vs. adaptive mean filtering in RGB,  $L^*a^*b^*$  and HSL color spaces using the square of size  $7 \times 7$  and the CANS with the homogeneity tolerance  $m = 100$  as operational window





**Fig. 12** **a** Original image. **b**  $H \rightarrow S \rightarrow L, h_0 = 0, n = 1/5$  **c**  $H \rightarrow S \rightarrow L, h_0 = 0, n = 5$ . **d**  $H \rightarrow S \rightarrow L, h_0 = 0, n = 1/5$ . **(e)**  $H \rightarrow S \rightarrow L, h_0 = 0, n = 5$ . **f**  $H \rightarrow S \rightarrow L, h_0 = \pi, n = 1/5$ . **g**  $H \rightarrow S \rightarrow L, h_0 = \pi, n = 5$ . Adaptive  $n$ -power filtering in HSL using different ordering relations of the color components:  $L \rightarrow S \rightarrow H$  and  $H \rightarrow S \rightarrow L$  with  $h_0 = 0$ ,  $H \rightarrow S \rightarrow L$  with  $h_0 = 0.5$ . The CANs are calculated with the homogeneity tolerance  $m = 100$

## 4.1 Classical Mathematical Morphology

The two fundamental operators of Mathematical Morphology [34] are mappings that commute with the infimum and supremum operations, called respectively erosion and dilation. To each morphological dilation there corresponds a unique morphological erosion, through a duality relation, and vice versa. For color images, two operators  $\psi$  and  $\phi$  defines an adjunction or a morphological duality [34] if and only if:  $\forall(f, g) \in \mathcal{S} \quad \psi(f) <_E g \Leftrightarrow f <_E \phi(g)$ .

The classical dilation and classical erosion of a color image  $f \in \mathcal{S}$  by a Structuring Element (SE) of size  $r$ , denoted  $B_r$ , are respectively defined as:

$$D_r(f) : \begin{cases} D \rightarrow E \\ x \mapsto \sup_E \{f(w); w \in \check{B}_r(x)\} \end{cases} \quad (22)$$

$$E_r(f) : \begin{cases} D \rightarrow E \\ x \mapsto \inf_E \{f(w); w \in B_r(x)\} \end{cases} \quad (23)$$

where  $B_r(x)$  denotes the SE located at point  $x$ , and  $\check{B}_r(x)$  is the reflected set of  $B_r(x)$ .  $\sup_E$  and  $\inf_E$  denote the supremum and infimum on the color space  $E$  using the ordering relation  $<_E$ .

The composition of erosion and dilation defines the elementary morphological filters of opening and closing respectively denoted  $O_r$  and  $C_r$ :

$$O_r(f) = D_r \circ E_r(f) \quad (24)$$

$$C_r(f) = E_r \circ D_r(f) \quad (25)$$

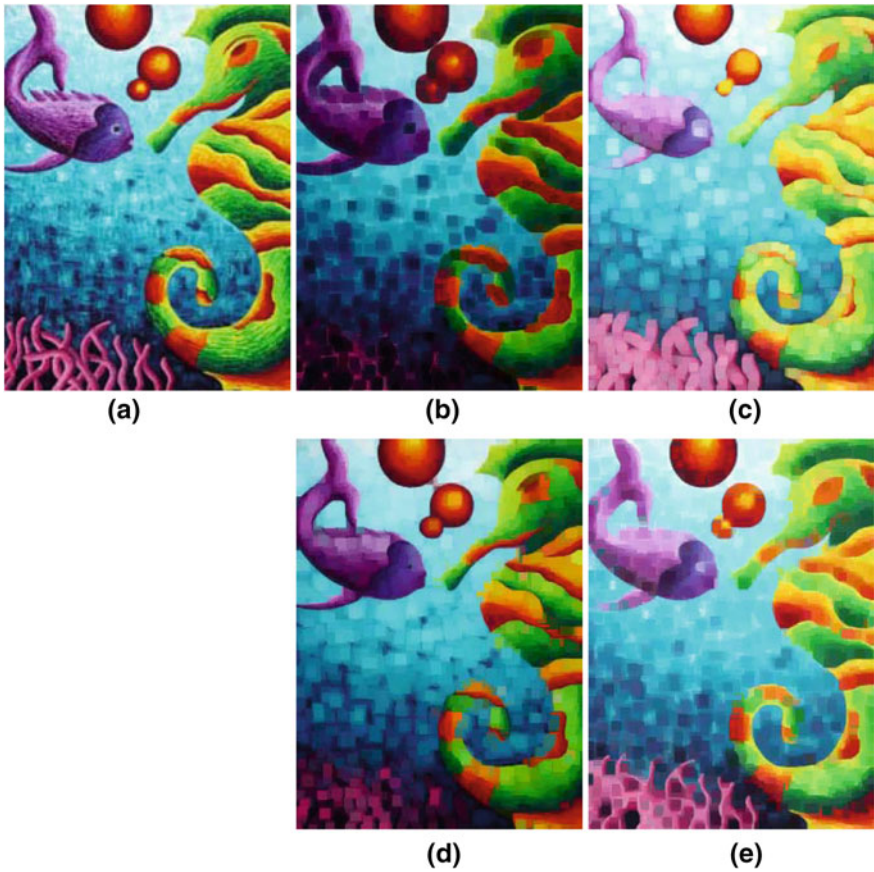
The opening and closing filters are idempotent and increasing operators.

Thereafter, more advanced morphological filters can be defined such as alternate (sequential) filters, filters by reconstruction, levelings...[34, 35].

The Fig. 13 shows an illustration of several classical morphological operators within the  $L^*a^*b^*$  color space, using the square of size  $7 \times 7$  as structuring element and the lexicographical order  $L \rightarrow A \rightarrow B$ . The operators are performed on a painting image from the artist Gamze Aktan.

## 4.2 Adaptive Mathematical Morphology

In the specialized literature, several approaches have been investigated for defining spatially adaptive morphological operators [3, 7, 10, 22, 24]. The basic idea of the Color Adaptive Neighborhood Mathematical Morphology (CANMM) is to replace the usual Structuring Elements (SEs) by CANs.



**Fig. 13** **a** Original image. **b** Classical erosion. **c** Classical dilation. **d** Classical opening. **e** Classical closing . Several classical morphological operators within the  $L^*a^*b^*$  color space, using the square of size  $7 \times 7$  as structuring element and the lexicographical order  $L^* \rightarrow a^* \rightarrow b^*$

### 4.2.1 Adaptive Structuring Elements

In CANMM, the Adaptive Structuring Elements (ASEs), denoted  $R_m^{f_0}(x)$  are defined as following:

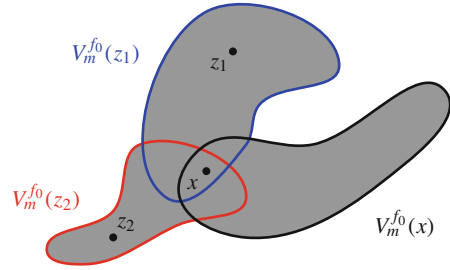
$$\forall(m, f_0, x) \in \mathbb{R}^+ \times \mathcal{S} \times D \quad R_m^{f_0}(x) = \bigcup_{z \in D} \{V_m^{f_0}(z) | x \in V_m^{f_0}(z)\} \quad (26)$$

Obviously:

$$V_m^{f_0}(x) \subseteq R_m^{f_0}(x) \quad (27)$$

Figure 14 illustrates the definition of an adaptive structuring element as union of CANs.

**Fig. 14** Representation of an ASE  $R_m^{f_0}(x)$



The CANs  $V_m^{f_0}(x)$  are not directly used as ASEs, because they do not satisfy the symmetry property contrary to the  $R_m^{f_0}(x)$ :

$$x \in R_m^{f_0}(y) \Leftrightarrow y \in R_m^{f_0}(x) \quad (28)$$

This symmetry condition is relevant for visual, topological, morphological and practical reasons [11]. In addition the ASEs also satisfy the properties of reflexivity, increasing with respect to  $m$ , translation invariance and multiplication compatibility.

#### 4.2.2 Adaptive Morphological Filters

The elementary operators of adaptive color dilation and adaptive color erosion are respectively defined accordingly to the ASEs:

$$\forall(m, f_0, f, x) \in \mathbb{R}^+ \times \mathcal{I} \times \mathcal{I} \times D$$

$$D_m^{f_0}(f)(x) = \sup_E \{f(w); w \in R_m^{f_0}(x)\} \quad (29)$$

$$E_m^{f_0}(f)(x) = \inf_E \{f(w); w \in R_m^{f_0}(x)\} \quad (30)$$

where  $\sup_E$  and  $\inf_E$  denote the supremum and infimum on the color space  $E$  using the ordering relation  $<_E$ .

These two CAN-based operators ( $D_m^{f_0}$ ,  $E_m^{f_0}$ ), using adaptive structuring elements that are computed on the input or pilot image  $f_0$ , defines input-adaptive adjunct morphological operators in the sense of [33]. The proof is given below :

$$\begin{aligned} D_m^{f_0}(f) <_E g &\Leftrightarrow D_m^{f_0}(f)(x) <_E g(x), \forall x \in D \\ &\Leftrightarrow \sup_E \{f(w); w \in R_m^{f_0}(x)\} <_E g(x), \forall x \in D \\ &\Leftrightarrow f(w) <_E g(x), \forall w \in R_m^{f_0}(x), \forall x \in D \\ &\Leftrightarrow f(w) <_E g(x), \forall x \in R_m^{f_0}(w), \forall w \in D \\ &\Leftrightarrow f(w) <_E \inf_E \{g(x); x \in R_m^{f_0}(w)\}, \forall w \in D \\ &\Leftrightarrow f(w) <_E E_m^{f_0}(g)(w), \forall w \in D \\ &\Leftrightarrow f <_E E_m^{f_0}(g) \end{aligned}$$

Using this input image  $f_0$  means that the ASEs in two successive runs will have the same shape, which results in the idempotence property for adaptive openings and closings defined as:

$$O_m^{f_0}(f) = D_m^{f_0} \circ E_m^{f_0}(f) \quad (31)$$

$$C_m^{f_0}(f) = E_m^{f_0} \circ D_m^{f_0}(f) \quad (32)$$

Thereafter, several advanced filters can be defined such as alternate (sequential) filters, morphological centre or toggle contrast [35].

The Fig. 15 shows an illustration of several adaptive morphological operators within the  $L^*a^*b^*$  color space, using the CANs computed on the original image (i.e.  $f_0 = f$ ) with the homogeneity tolerance  $m = 40$  as structuring element and the lexicographical order  $L^* \rightarrow a^* \rightarrow b^*$ . The operators are performed on a painting image from the artist Gamze Aktan. This figure can be compared with Fig. 13 for viewing the differences between the classical filters and the proposed spatially adaptive filters.

### 4.2.3 Properties

As stated for the gray-tone images [11], the proposed CAN-based morphological operators satisfy the following properties:

Let  $(m, f_0, f, f_1, f_2) \in \mathbb{R}^+ \times \mathcal{S}^4$ .

1. increasing:

$$f_1 <_E f_2 \Rightarrow \begin{cases} D_m^{f_0}(f_1) <_E D_m^{f_0}(f_2) \\ E_m^{f_0}(f_1) <_E E_m^{f_0}(f_2) \\ C_m^{f_0}(f_1) <_E C_m^{f_0}(f_2) \\ O_m^{f_0}(f_1) <_E O_m^{f_0}(f_2) \end{cases} \quad (33)$$

2. adjunction (morphological duality):

$$D_m^{f_0}(f_1) <_E f_2 \Leftrightarrow f_1 <_E E_m^{f_0}(f_2) \quad (34)$$

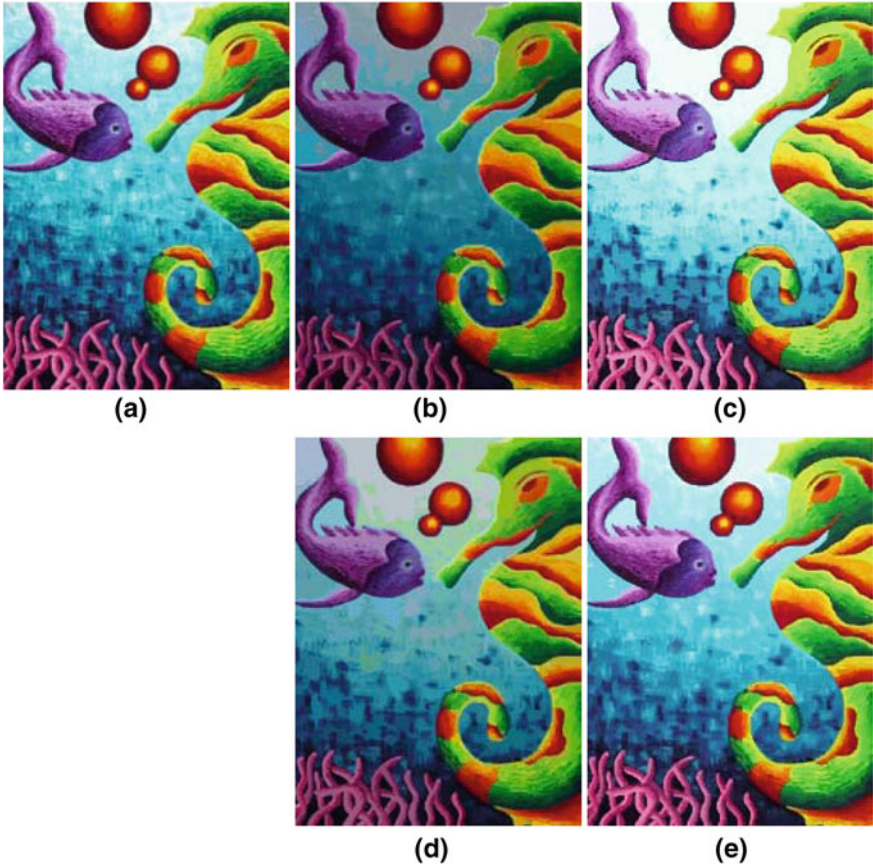
3. extensiveness, anti-extensiveness:

$$O_m^{f_0}(f) <_E f <_E C_m^{f_0}(f) \quad (35)$$

4. distributivity with  $\sup_E, \inf_E$ :

$$\forall (f_i) \in \mathcal{S}^I \quad \begin{cases} \sup_{i \in I} [D_m^{f_0}(f_i)] = D_m^{f_0}(\sup_{i \in I} [f_i]) \\ \inf_{i \in I} [E_m^{f_0}(f_i)] = E_m^{f_0}(\inf_{i \in I} [f_i]) \end{cases} \quad (36)$$

where  $I$  is an index set (finite or not).



**Fig. 15** **a** Original image. **b** Adaptive erosion. **c** Adaptive dilation. **d** Adaptive opening. **e** Adaptive closing. Several adaptive morphological operators within the  $L^*a^*b^*$  color space, using the CANs computed on the original image (i.e.  $f_0 = f$ ) with the homogeneity tolerance  $m = 40$  as structuring elements and the lexicographical order  $L^* \rightarrow a^* \rightarrow b^*$

5. idempotence:

$$\begin{cases} C_m^{f_0} \circ C_m^{f_0}(f) = C_m^{f_0}(f) \\ O_m^{f_0} \circ O_m^{f_0}(f) = O_m^{f_0}(f) \end{cases} \quad (37)$$

6. increasing, decreasing with respect to  $m$ :

$$\left( \begin{matrix} (m_1, m_2) \in \mathbb{R}^+ \times \mathbb{R}^+ \\ m_1 \leq m_2 \end{matrix} \right) \Rightarrow \begin{cases} D_{m_1}^{f_0}(f) <_E D_{m_2}^{f_0}(f) \\ E_{m_2}^{f_0}(f) <_E E_{m_1}^{f_0}(f) \end{cases} \quad (38)$$

The proofs of these properties are similar to those given in [11]. They are inferred from the adjunction (proved in Paragraph 4.2.2), the lattice theory of increasing mappings and the properties of the CANs (Sect. 2.3).

## 5 Application Examples

In the following examples, if nothing is mentioned, the original image is used as the pilot image (i.e.  $f_0 = f$ ).

### 5.1 Image Restoration

Morphological openings and closings or more advanced alternate filters, such as the morphological centre [35] (also known as automedian filter), are often used for image restoration.

In the Fig. 16, the adaptive opening and closing are both compared to the classical opening and closing and to the opening and closing by reconstruction (within the HSL color space). The processing is performed on a painting image from the artist Gamze Aktan.

The CAN-based morphological filters and filters by reconstruction are more efficient than the classical morphological filters. Indeed, the spatial structures are rapidly damaged with blurring effects by using the classical morphology. On the contrary, the resulting images are smoothed using reconstruction and adaptive filters while preserving some transitions and details. Nevertheless, there are significant differences around the eye and on the right of the nose, where the adaptive filters better preserve the color contrasts of these spatial structures.

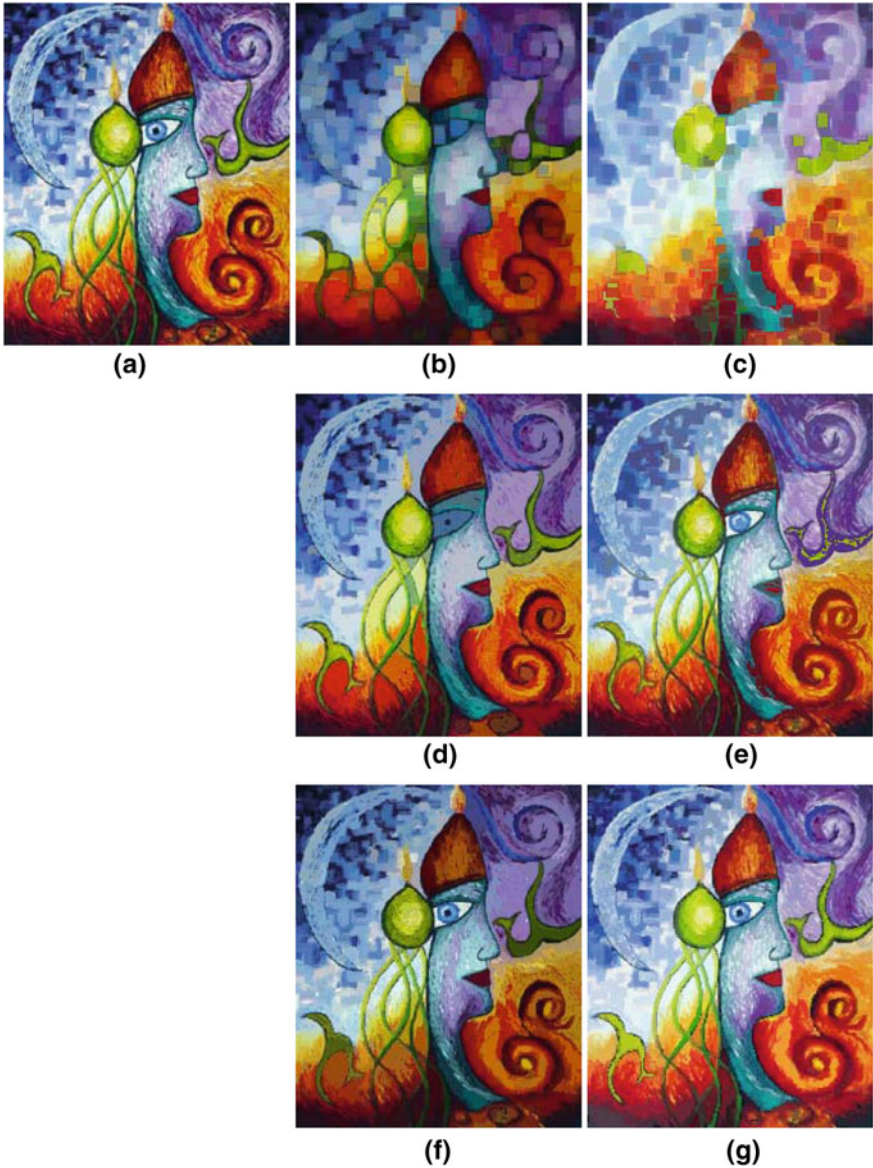
A second example of image restoration (Fig. 17) is given by the morphological centre [35] (a toggle mapping also known as automedian filter) which is defined in the adaptive way as:

$$\zeta_m^{f_0}(f) = \inf_E(\sup_E(f, \inf_E(\psi_m^{f_0}(f), \phi_m^{f_0}(f))), \sup_E(\psi_m^{f_0}(f), \phi_m^{f_0}(f))) \quad (39)$$

where:  $\psi_m^{f_0}(f) = O_m^{f_0} \circ C_m^{f_0} \circ O_m^{f_0}(f)$  and  $\phi_m^{f_0}(f) = C_m^{f_0} \circ O_m^{f_0} \circ C_m^{f_0}(f)$ .

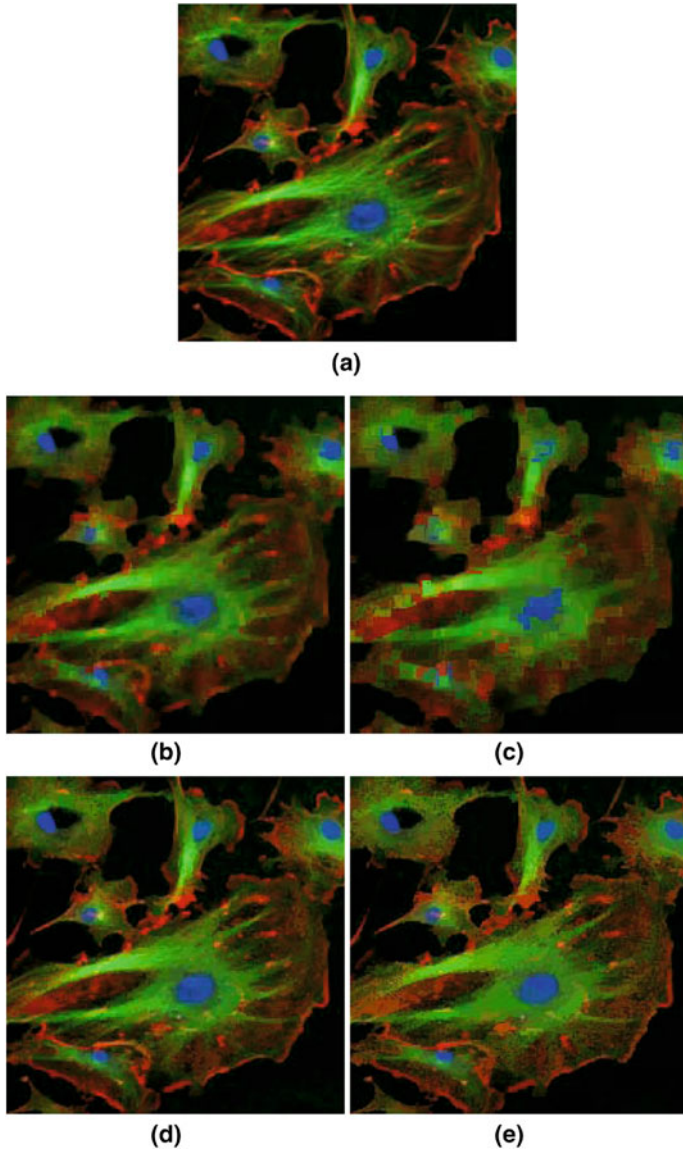
This operator is applied on an image of cells acquired by fluorescence microscopy (sample image of ImageJ).

The resulting images highlight the efficiency of the proposed adaptive morphological filtering where the restoration process is suitable for further image processes (such as image segmentation), contrary to the classical morphological filtering.

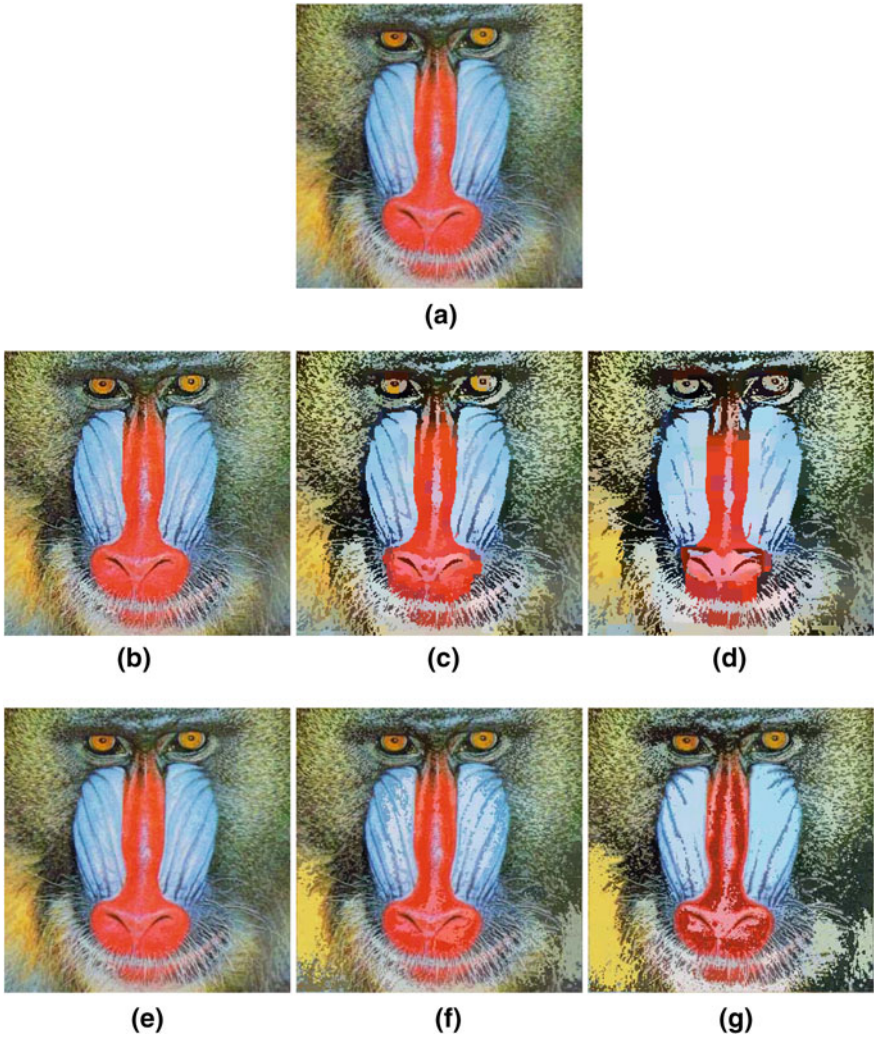


**Fig. 16** **a** Original image. **b** Classical opening. **c** Classical closing. **d** Opening by reconstruction. **e** Closing by reconstruction. **f** Adaptive opening. **g** Adaptive closing. Image restoration of the original painting image **(a)** from the artist Gamze Aktan. The morphological classical filtering **(b-c)** and the morphological filtering by reconstruction **(d-e)** are performed with the square of size  $7 \times 7$  as structuring element. The adaptive morphological filtering **(f-g)** is performed with CANs computed on the original image (i.e.  $f_0 = f$ ) with the homogeneity tolerance value  $m = 40$ . All these morphological operators are achieved within the HSL color space using the lexicographical order with the component ordering  $L \rightarrow S \rightarrow H$





**Fig. 17** **a** Original image **b** classical morphological centre,  $r = 1$  **c** classical morphological centre,  $r = 2$  **d** adaptive morphological centre,  $m = 20$  **e** adaptive morphological centre,  $m = 30$  . Image restoration of the original image **(a)** of cells acquired by fluorescence microscopy (image proposed in ImageJ). The classical morphological centre operator **(b-c)** is performed with centered squares of width  $2r + 1$  as structuring elements, and the adaptive morphological centre operator **(d-e)** is performed with CANs computed on the original image (i.e.  $f_0 = f$ ) with the homogeneity tolerance values  $m$ . All these morphological operators are achieved within the HSL color space using the lexicographical order with the component ordering  $L \rightarrow S \rightarrow H$



**Fig. 18** **a** Original image. **b** Classical toggle contrast,  $r = 1$ . **c** Classical toggle contrast,  $r = 5$ . **d** Classical toggle contrast,  $r = 10$ . **e** Adaptive toggle contrast,  $m = 25$ . **f** Adaptive toggle contrast,  $m = 50$ . **g** Adaptive toggle contrast,  $m = 75$ . Image enhancement performed on the original 'baboon' image (**h**) by the classical toggle contrast operator (**b-d**) using centered squares of width  $2r + 1$  as structuring elements, and by the adaptive adaptive toggle contrast operator (**e-g**) using the CANs computed on the color gradient of the original image with the homogeneity tolerance values  $m$ . All these morphological operators are achieved within the RGB color space using the lexicographical order with the component ordering  $G \rightarrow R \rightarrow B$

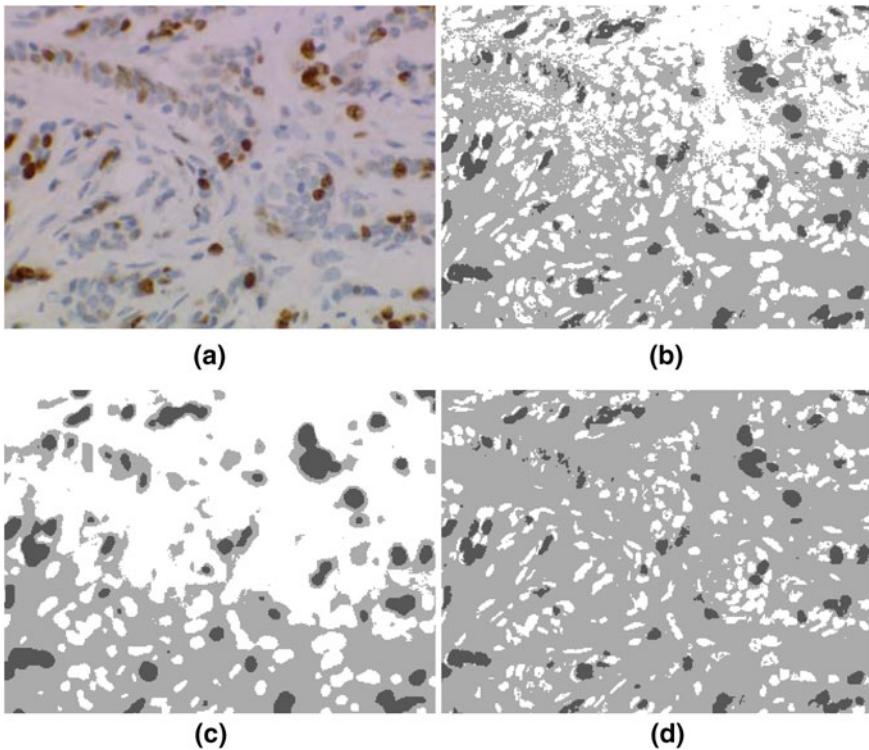
### 5.2 Image Enhancement

In the following example, the original image ‘baboon’ is enhanced using the morphological toggle contrast operator [35]. The CAN-based operator is defined as:

$$\tau_m^{f_0} f(x) = \begin{cases} D_m^{f_0}(f)(x) & \text{if } D_m^{f_0}(f)(x) - f(x) <_E f(x) - E_m^{f_0}(f)(x), \\ E_m^{f_0}(f)(x) & \text{otherwise.} \end{cases} \quad (40)$$

where the CANs are calculated on the color gradient using the  $R, G, B$  components:  $f_0 = (G_R, G_G, G_B)$  where  $G_Z$  denotes the Sobel gradient of the  $Z$  component ( $Z = R, G, \text{ or } B$ ).

The resulting images show large differences between the classical and adaptive approaches, especially when the filter becomes strong. On the one hand, the spatial



**Fig. 19** a Original image b segmented image without filtering c segmented image with classical mean filtering d segmented image with adaptive mean filtering. Segmentation of the original image (a) of cells (sample image of ADCIS Apheleon) by ‘kmeans’ clustering. The ‘kmeans’ operator (with  $k = 3$  clusters) is either directly performed on the original image (b) or performed after a classical mean filtering (c) or adaptive mean filtering (d) within the  $L^*a^*b^*$  color space. A square of size  $3 \times 3$  and the CANs computed on the original image (i.e.  $f_0 = f$ ) with the homogeneity tolerance  $m = 5$  are used as operational windows

structures such as the eyes are rapidly damaged by the classical filtering contrary to the adaptive one. On the other hand, the color contrast of both the nose and the region on the left of the face is strongly increased by the adaptive filtering (with  $m = 75$ ) in comparison with the other spatial structures (eyes, mouth). This effect is certainly a consequence of the choice of the component ordering.

### 5.3 Image Segmentation

In Fig. 19, a real application of cell image segmentation is proposed. The objective is to identify and separate the cells stained in brow and purple from the background. The segmentation process is based on the  $k$ -means clustering [23] that is applied on the  $a^*$  and  $b^*$  chromatic components of the  $L^*a^*b^*$  color space. The ‘kmeans’ operator (with  $k = 3$  clusters) is either directly performed on the original image or performed after a classical Choquet mean filtering or adaptive Choquet mean filtering within the  $L^*a^*b^*$  color space.

The resulting images shows that the CAN-based filtering gives the expected segmentation contrary to the classical filtering and the process without filtering.

## 6 Conclusion and Perspectives

In this chapter, spatially adaptive color image processing has been introduced in the context of the General Adaptive Neighborhood Image Processing (GANIP) approach. The proposed adaptive Choquet and morphological operators have been defined by using a lexicographical order on the RGB,  $L^*a^*b^*$  and HSL color spaces. The theoretical advantages of these adaptive operators have been practically highlighted on several examples for image restoration, image enhancement and image segmentation. The resulting filters show a good performance by processing an image while preserving its contrast details without damaging its transitions. The computational cost of the proposed adaptive filters is relatively low (the algorithms are not yet optimized). For example, the computation time of the CAN-based morphological dilation or erosion is about 5 s (respectively 120 and 700) for a  $128 \times 128$  (resp.  $256 \times 256$  and  $512 \times 512$ ) image using a Pentium IV (3 GHz/2 GB RAM) and the Matlab software.

Currently, the authors are working on the Color Logarithmic Image Processing (CoLIP) [17] framework for generalizing the proposed spatially adaptive filters.

## References

1. Amattouch M (2005) Théorie de la Mesure et Analyse d’Image. Master’s thesis, Ecole Nationale Supérieure des Mines, Saint-Etienne, France
2. Angulo J (2007) Morphological colour operators in totally ordered lattices based on distances: application to image filtering, enhancement and analysis. *Comput Vis Image Underst* 107:56–73

3. Angulo J, Velasco-Forero S (2011) Structurally adaptive mathematical morphology based on nonlinear scale-space decompositions. *Image Anal Stereology* 30(2):111–122
4. Aptoula E, Lefèvre S (2007) A comparative study on multivariate mathematical morphology. *Pattern Recogn* 40(11):2914–2929
5. Astola J, Haavisto P, Neuvo Y (1990) Vector median filtering. *Proc IEEE* 78(4):678–689
6. Barnett V (1976) The ordering of multivariate data. *J Roy Stat Soc A* 139(3):318–354
7. Bouaynaya N, Schonfeld D (2008) Theoretical foundations of spatially-variant mathematical morphology. Part II: Gray-Level Images. *IEEE Trans Pattern Anal Mach Intell* 30(5):837–850
8. Busin L, Vandenbroucke N, Macaire L (2008) Advances in imaging and electron physics. chapter Color spaces and image segmentation, vol 151. Elsevier, Orlando, pp 65–168
9. Choquet G (2000) Cours de Topologie, chap. Espaces topologiques et espaces métriques. Dunod, Paris, pp 45–51
10. Curic V, Luengo C, Borgfors G (2012) Saliency adaptive structuring elements. *IEEE J Sel Top Signal Process, Spec Issue Filtering and Segmentation Math Morphol* 6(7):809–819
11. Debayle J, Pinoli JC (2006) General adaptive neighborhood image processing - Part I: introduction and theoretical aspects. *J Math Imaging Vis* 25(2):245–266
12. Debayle J, Pinoli JC (2006) General adaptive neighborhood image processing - Part II: practical application examples. *J Math Imaging Vis* 25(2):267–284
13. Debayle J, Pinoli JC (2009) General adaptive neighborhood choquet image filtering. *J Math Imaging Vis* 35(3):173–185
14. Debayle J, Pinoli JC (2009) General adaptive neighborhood representation for adaptive choquet image filtering. 10th European congress of stereology and image analysis. Milan, pp 431–436
15. Debayle J, Pinoli JC (2011) Advances in imaging and electron physics. chapter Theory and applications of general adaptive neighborhood image processing, vol 167. Elsevier, pp 121–183
16. Debayle J, Pinoli JC (2011) Applied biomedical engineering. chapter General adaptive neighborhood image processing for biomedical applications. InTech, pp 481–500
17. Gouinaud H, Gavet Y, Debayle J, Pinoli JC (2011) Color correction in the framework of color logarithmic image processing. 7th international symposium on image and signal processing and analysis. Dubrovnik, Croatia, pp 129–133
18. Grabisch M (1994) Fuzzy integrals as a generalized class of order filters. In: *Proceedings of the SPIE*, vol 2315. pp 128–136
19. Hanbury A, Serra J (2001) Morphological operators on the unit circle. *IEEE Trans Image Process* 10(12):1842–1850
20. Joblove GH, Greenberg D (1978) Color spaces for computer graphics. *Comput Graphics* 12(3):20–25
21. Lee JH, Chang BH, Kim SD (1994) Comparison of colour transformations for image segmentation. *Electron Lett* 30(20):1660–1661
22. Lerallut R, Decencièrre E, Meyer F (2007) Image filtering using morphological amoebas. *Image Vis Comput* 25(4):395–404
23. MacQueen JB (1967) Some methods for classification and analysis of multivariate observations. In: *5th Berkeley symposium on mathematical statistics and probability*, pp 281–297
24. Maragos P, Vachier C (2009) Overview of adaptive morphology: trends and perspectives. *IEEE Int Conf Image Process*. Cairo, pp 2241–2244
25. Matheron G (1967) *Éléments pour une théorie des milieux poreux*. Masson, Paris
26. Murofushi T, Sugeno M (1989) An interpretation of fuzzy measure and the choquet integral as an integral with respect to a fuzzy measure. *Fuzzy Sets Syst* 29:201–227
27. Ortiz F, Torres F, Gil P, Pomares J, Puente S, Candelas F (2001) Comparative study of vectorial morphological operations in different color spaces. *Proceedings of the SPIE conference on intelligent robots and computer vision XX*, vol 4572. Boston, MA, pp 259–268
28. Otha YI, Kanade T, Sakai T (1980) Color information for region segmentation. *Comput Graphics Image Process* 13:222–241
29. Pinoli JC, Debayle J (2009) General adaptive neighborhood mathematical morphology. *IEEE international conference on image processing (ICIP)*. Cairo, pp 2249–2252

30. Pinoli JC, Debayle J (2012) Adaptive generalized metrics, distance maps and nearest neighbor transforms on gray tone images. *Pattern Recognit* 45:2758–2768
31. Pinoli JC, Debayle J (2012) Spatially and intensity adaptive morphology. *IEEE J Selected Top Sig Process, Special Issue Filtering Segmentation Math Morphol* 6(7):820–829
32. Pitas I, Tsakalides P (1991) Multivariate ordering in color image filtering. *IEEE Trans Circ Syst Video Technol* 1:247–259
33. Roerdink JBTM (2009) Adaptivity and group invariance in mathematical morphology. *IEEE international conference on image processing*, Cairo, pp 2253–2256
34. Serra J (1982) *Image Anal Math Morphol*. Academic Press, London
35. Soille P (2003) *Morphological image analysis. Principles and applications*. Springer, New York
36. Sugeno M (1974) *Theory of fuzzy integrals and its applications*. Ph.D. thesis, Tokyo Institute of Technology, Japan
37. Tang K, Astola J, Neuvo Y (1995) Nonlinear multivariate image filtering techniques. *IEEE Trans Image Process* 4:788–798
38. Trahanias PE, Venetsanopoulos AN (1993) Vector directional filters: a new class of multichannel image processing filters. *IEEE Trans Image Process* 2:528–534
39. Trémeau A, Tominaga S, Plataniotis KN (2008) Color in image and video processing. *EURASIP J Image Video Process* 581371:1–26

# Vector Ordering and Multispectral Morphological Image Processing

Santiago Velasco-Forero and Jesus Angulo

**Abstract** This chapter illustrates the suitability of recent multivariate ordering approaches to morphological analysis of colour and multispectral images working on their vector representation. On the one hand, *supervised ordering* renders machine learning notions and image processing techniques, through a learning stage to provide a total ordering in the colour/multispectral vector space. On the other hand, *anomaly-based ordering*, automatically detects spectral diversity over a majority background, allowing an adaptive processing of salient parts of a colour/multispectral image. These two multivariate ordering paradigms allow the definition of morphological operators for multivariate images, from algebraic dilation and erosion to more advanced techniques as morphological simplification, decomposition and segmentation. A number of applications are reviewed and implementation issues are discussed in detail.

**Keywords** Multivariate mathematical morphology · Supervised ordering · Segmentation · Complete lattice · Statistical depth function

## 1 Introduction

Problems on defining *total order* arise naturally in different aspects of science and engineering and their applications in daily life. In our context, “order” denotes an ordering principle: a pattern by which the elements of a given set may be arranged

---

S. Velasco-Forero (✉)  
ITWM - Fraunhofer Institute, Kaiserslautern, Germany  
e-mail: velascoforero@itwm.fraunhofer.de

J. Angulo  
Mathématiques et Systèmes, CMM-Centre de Morphologie Mathématique,  
MINES ParisTech; 35, rue Saint-Honoré, 77305 Fontainebleau CEDEX, France  
e-mail: angulo@cmm.ensmp.fr

[13], and “total” means that the order is a binary relation antisymmetric, transitive and reflexive. Another name for a total order is *linear order*. It expresses the intuitive idea that you can picture a total order on a set  $A$  as arranging the elements of  $A$  in a line. Not surprisingly, the task of defining a total order for a given set depends greatly on the prior knowledge provided. To illustrate this point, imagine a scenario where two researchers want to order a set of people. The first one defines minimum as the youngest person and the maximum as the oldest person, the second declares the minimum as the fattest one and the maximum as the thinnest one. Clearly, two persons that are similar according to the first researcher’s setting might be dissimilar according to the second’s. Accordingly, the complete list of ordered people can be totally different from one researcher to another. In the field of image processing, the definition of a total ordering among pixels of the image is the main ingredient of *mathematical morphology* techniques [20]. In this first part of this chapter, we study the case where “prior” knowledge about the spectral information of the background and the foreground on the image is available. We define a *supervised ordering* as a particular case of reduced ordering where the minimum (resp. maximum) value should be a pixel in the background (resp. foreground). This restriction can be included in the computation of the supervised ordering by using classical machine learning techniques, for instance, by support vector machines (SVM) [25]. Another possibility for known structure in the total ordering problem is to assume that the image is composed by two main components: background and foreground. Additionally, we include the assumption that the background is larger than the foreground. We uncover an interesting application of randomised approximation schemes in multivariate analysis [26]. To summarise, in this chapter a multispectral image is represented through a total ordering and it is analysed by mathematical morphology transformations. Prior information about the spectral information in the image is incorporated into the workflow of mathematical morphology transformations in two scenarios:

1. Spectral information about the background and the object of interest are available, i.e., “background/foreground training pixels”.
2. Image can be considered as objects (foreground) over a majority background.

The remainder of this chapter is organised as follows. In Sect. 2, we present the fundamental definitions of mathematical morphology in a lattice formulation. The approach involving a preordering function is presented in Sect. 3. This section also contains examples specialising the general approach to more specific settings. Section 4 explains implementation issues for any adjunction based morphological transformation. Finally, Sect. 5 concludes the chapter.



## 2 Complete Lattices and Mathematical Morphology

### 2.1 Mathematical Morphology

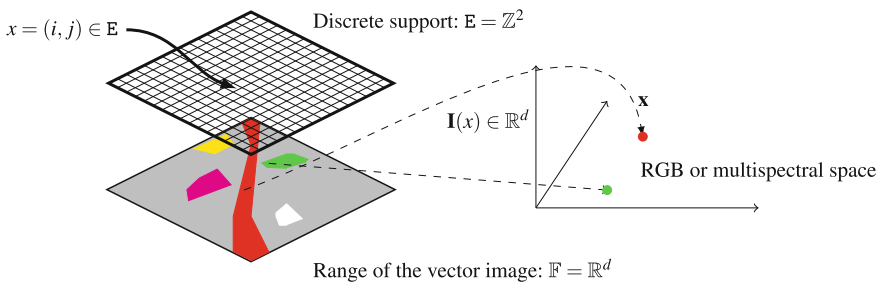
Basically, there are two points of view about mathematical morphological transformations: (1) *connection based* and (2) *adjunction based*. The first strategy deals with simplification of a given image in the partition space induced by its connected components [17, 18, 21]. The second perspective analyses an image by composition of two basic transformations, dilation and erosion, which form a Galois connection [9]. In this section we provide the theoretical background of mathematical morphology in its formulation based on adjunction, i.e. by using dilation/erosion operators. Our approach does not include the “connectivity approach”. We refer keen readers to [21] for a comprehensive review of connective morphology.

### 2.2 Fundamental Definitions

Let us introduce the notation for a multidimensional image, as it is illustrated in Fig. 1, where the object of interest is a  $d$ -dimensional image (denoted by  $\mathbf{I}$ ) which maps the spatial support  $\mathbb{E}$  to the vector support  $\mathbb{F}$ , i.e.,

$$\begin{aligned} \mathbf{I} : \mathbb{E} &\rightarrow \mathbb{F} = \mathbb{R}^d \\ x &\rightarrow \mathbf{x} \end{aligned}$$

Given a vector image  $\mathbf{I} \in \mathcal{F}(\mathbb{E}, \mathbb{F})$ , i.e. is a mapping from the spatial support to the vector space of dimensions  $d$ . Theoretical formulation of mathematical morphology is nowadays phrased in terms of complete lattices and operators defined on them. For a detailed exposition on complete lattice theory in mathematical morphology, we refer to J. Serra and C. Ronse in [16, Chap. 2].



**Fig. 1** Notation for a  $d$ -variate image,  $\mathbf{I} : \mathbb{E} \rightarrow \mathbb{F}$ . Note that the image  $\mathbf{I}$  maps each spatial point  $x$  to a vector  $\mathbf{x}$  in three dimension for a RGB image or in dimension  $d$  for the case of a multispectral image

**Definition 1 (Complete Lattice)** A space  $\mathcal{L}$  endowed with a partial order  $\leq$  is called a complete lattice, denoted  $(\mathcal{L}, \leq)$  if every subset  $\mathcal{M} \subseteq \mathcal{L}$  has both supremum (join)  $\bigvee \mathcal{M}$  and infimum (meet)  $\bigwedge \mathcal{M}$ .

A minimum (or least)  $\perp \in \mathcal{M}$  is an element which is least than or equal to any other element of  $\mathcal{M}$ , that is,  $r \in \mathcal{M} \Rightarrow \perp \leq r$ . We denote the minimum of  $\mathcal{L}$  by  $\perp$ . Equivalently, a maximum (largest)  $\top$  in  $\mathcal{M}$  is the greatest element of  $\mathcal{M}$ , that is,  $r \in \mathcal{M} \Rightarrow r \leq \top$ . We denote the maximum of  $\mathcal{L}$  by  $\top$ .

**Definition 2 (Dilation/Erosion)** A mapping  $f : \mathcal{L}_1 \rightarrow \mathcal{L}_2$  of a complete lattice  $\mathcal{L}_1$  into a complete lattice  $\mathcal{L}_2$  is said to be a dilation if  $f(\bigvee_{j \in J} r_j) = \bigvee_{j \in J} f(r_j)$  for all families  $(r_j)_{j \in J}$  of elements in  $\mathcal{L}_1$ . A mapping is said to be an erosion if  $f(\bigwedge_{j \in J} r_j) = \bigwedge_{j \in J} f(r_j)$  for all families  $(r_j)_{j \in J}$  of elements in  $\mathcal{L}_1$ .

The important relationship between dilation and erosion is that they are dual concepts from the lattice point of view. [9] showed that for any complete lattice  $\mathcal{L}$ , we always have a dual isomorphism between the complete lattice of dilation on  $\mathcal{L}$  and the complete lattice of erosions on  $\mathcal{L}$ . This dual isomorphism is called by Serra [20, Chap. 1] the *morphological duality*. In fact it is linked to what one calls *Galois connections* in lattice theory, as we will see at the end of this section.

**Definition 3 (Adjunction)** Let  $\delta, \varepsilon \in \mathcal{L} \rightarrow \mathcal{L}$ . Then we say that  $(\varepsilon, \delta)$  is an adjunction of every  $r, s \in \mathcal{L}$ , we have

$$\delta(r) \leq s \iff r \leq \varepsilon(s) \quad (1)$$

In an adjunction  $(\varepsilon, \delta)$ ,  $\varepsilon$  is called the *upper adjoint* and  $\delta$  the *lower adjoint*.

**Proposition 1** [9, p. 264] Let  $\delta, \varepsilon \in \mathcal{L} \rightarrow \mathcal{L}$ . If  $(\varepsilon, \delta)$  is an adjunction, then  $\delta$  is a dilation and  $\varepsilon$  is an erosion.

**Definition 4 (Galois connection)** Let  $\mathcal{L}_1$  and  $\mathcal{L}_2$  be lattices and let  $\alpha : \mathcal{L}_1 \rightarrow \mathcal{L}_2$  and  $\beta : \mathcal{L}_2 \rightarrow \mathcal{L}_1$  satisfy the following conditions.

1. For  $r, s \in \mathcal{L}_1$ , if  $r \leq s$ , then  $\alpha(r) \leq \alpha(s)$ .
2. For  $r, s \in \mathcal{L}_1$ , if  $r \leq s$ , then  $\beta(r) \leq \beta(s)$ .
3. For  $r \in \mathcal{L}_1$ ,  $\beta\alpha(r) \leq r$ .
4. For  $r \in \mathcal{L}_2$ ,  $\alpha\beta(r) \leq r$ .

Then  $(\alpha, \beta)$  is a Galois connection between  $\mathcal{L}_1$  and  $\mathcal{L}_2$ .

**Proposition 2** Let the lattices  $\mathcal{L}_1$  and  $\mathcal{L}_2$ , maps  $\alpha : \mathcal{L}_1 \rightarrow \mathcal{L}_2$  and  $\beta : \mathcal{L}_2 \rightarrow \mathcal{L}_1$  a Galois connection. Then the following condition holds for all  $r \in \mathcal{L}_1$  and  $s \in \mathcal{L}_2$ :

$$s \leq \alpha(r) \iff r \leq \beta(s) \quad (2)$$

Clearly an adjunction in  $\mathcal{L}$  is a Galois connection between the dual  $(\mathcal{L}, \geq)$  and  $(\mathcal{L}, \leq)$  (indeed, compare Definition 3 and Proposition 2).

At this point, we can see that definition of erosion/dilation on a image requires a complete lattice structure, i.e., a total ordering<sup>1</sup> among the pixels to be analysed. However, there is not difficult to see that the idea of order is entirely absent from multivariate scene, i.e., there is no unambiguous means of defining the minimum and maximum values between two vectors of more than one dimension. Accordingly, the extension of mathematical morphology to vector spaces, for instance, colour/multi/hyper/ultraspectral images, is neither direct nor trivial because the pixels in the images are vectors. We refer keen readers to [1, 2] for a comprehensive review of vector morphology.

### 2.3 Preorder by $h$ -Function

Let  $E$  be a nonempty set and assume that  $\mathcal{L}$  is a complete lattice. Let  $h : E \rightarrow \mathcal{L}$  be a surjective mapping. Define an equivalence relation  $=_h$  on  $E$  as follows:  $x =_h y \Leftrightarrow h(x) = h(y) \quad \forall x, y \in E$ . As it was defined in [8], we refer by  $\leq_h$  the  $h$ -ordering given by the following relation on  $E$

$$\forall x, y \in E, \quad x \leq_h y \Leftrightarrow h(x) \leq h(y)$$

Note that  $\leq_h$  preserves reflexivity ( $x \leq_h x$ ) and transitivity ( $x_1 \leq_h x_2$  and  $x_2 \leq_h x_3 \Rightarrow x_1 \leq_h x_3$ ). However,  $\leq_h$  is not a partial ordering because  $x \leq_h y$  and  $y \leq_h x$  implies only that  $x =_h y$  but not  $x = y$ . Note that  $h$ -ordering is a preorder in  $E$ .

An operator  $\psi : E \rightarrow E$  is  $h$ -increasing if  $x \leq_h y$  implies that  $\psi(x) \leq_h \psi(y)$ . Additionally, since  $h$  is surjective, an equivalence class is defined by  $\mathcal{L}[r] = \{y \in E | h(y) = r\}$ . The Axiom of Choice [8] implies that there exist mappings  $h^\leftarrow : \mathcal{L} \rightarrow E$  such that  $hh^\leftarrow(r) = r$ , for  $r \in \mathcal{L}$ . Unless  $h$  is injective, there exist more than one such  $h^\leftarrow$  mappings:  $h^\leftarrow$  is called the semi-inverse of  $h$ . Note that  $h^\leftarrow h =_h \text{id}$ . However, we have that for any  $h$ -increasing  $\psi : E \rightarrow E$  the result  $\psi h^\leftarrow h =_h \psi$  and hence  $h\psi h^\leftarrow h = h\psi$ . Let us introduce  $\tilde{\psi}$  the operator associated to  $\psi$  in the lattice  $\mathcal{L}$ . A mapping  $\psi : E \rightarrow E$  is  $h$ -increasing if and only if there exists an increasing mapping  $\tilde{\psi} : \mathcal{L} \rightarrow \mathcal{L}$  such that  $\tilde{\psi}h = h\psi$ . The mapping  $\tilde{\psi}$  is uniquely determined by  $\psi$  and can be computed from

$$\tilde{\psi} = h\psi h^\leftarrow$$

We can now define the  $h$ -erosion and  $h$ -dilation. Let  $\varepsilon, \delta : E \rightarrow E$  be two mappings with the property

$$\delta(x) \leq_h y \Leftrightarrow x \leq_h \varepsilon(y), \quad \forall x, y \in E$$

---

<sup>1</sup> Theoretically, a partial ordering is enough but to make easier the presentation we analyse the case of total ordering.

then the pair  $(\varepsilon, \delta)$  is called an  $h$ -adjunction. Moreover, let  $(\varepsilon, \delta)$  be  $h$ -increasing mappings on  $\mathbb{E}$ , and let  $\varepsilon \mapsto^h \tilde{\varepsilon}, \delta \mapsto^h \tilde{\delta}$ . Then  $(\varepsilon, \delta)$  is an  $h$ -adjunction on  $\mathbb{E}$  if and only if  $(\tilde{\varepsilon}, \tilde{\delta})$  is an adjunction on the lattice  $\mathcal{L}$ . Therefore a mapping  $\delta$  (resp.  $\varepsilon$ ) on  $\mathbb{E}$  is called  $h$ -dilation (resp.  $h$ -erosion) if  $\tilde{\delta}$  (resp.  $\tilde{\varepsilon}$ ) is a dilation (resp. erosion) on  $\mathcal{L}$ .  $h$ -adjunctions inherit a large number of properties from ordinary adjunctions between complete lattices. Assume that  $(\varepsilon, \delta)$  is an  $h$ -adjunction then

$$\gamma = \delta\varepsilon \leq_h \text{id} \leq_h \varphi = \varepsilon\delta.$$

Hence,  $\gamma$  is  $h$ -anti-extensive and  $\varphi$  is  $h$ -extensive. The operator  $\gamma$  on  $\mathbb{E}$  is called  $h$ -opening if the operator  $\tilde{\gamma}$  on  $\mathcal{L}$  determined by  $\gamma \mapsto^h \tilde{\gamma}$  is an opening. The operator  $\gamma$  is also  $h$ -increasing and satisfies  $\gamma\gamma =_h \gamma$  ( $h$ -idempotency). The  $h$ -closing is similarly defined.

### 2.4 Morphological Analysis on the $h$ -Function

From the preliminary section we have the ingredients to define morphological colour ( $\mathbb{F} = \mathbb{R}^3$ ) and multispectral ( $\mathbb{F} = \mathbb{R}^d$ ) erosion and dilation. We limit here our developments to the flat operators, i.e., the structuring elements are planar shapes. The non-planar structuring functions are defined by weighting values on their support [19]. Let us assume that we have an adaptive mapping<sup>2</sup>  $h : \mathbb{R}^d \rightarrow \mathbb{R}$ . The  $h$ -erosion  $\varepsilon_{SE,h}(\mathbf{I})$  and  $h$ -dilation  $\delta_{SE,h}(\mathbf{I})$  of an image  $\mathbf{I}$  at pixel  $x \in \mathbb{E}$  by the structuring element  $SE \subset \mathbb{E}$  are the two mappings  $\mathcal{F}(\mathbb{E}, \mathbb{F}) \rightarrow \mathcal{F}(\mathbb{E}, \mathbb{F})$  defined respectively by

$$h(\varepsilon_{SE,h}(\mathbf{I})(x)) = \tilde{\varepsilon}_{SE}(h(\mathbf{I}))(x), \tag{3}$$

and

$$h(\delta_{SE,h}(\mathbf{I})(x)) = \tilde{\delta}_{SE}(h(\mathbf{I}))(x), \tag{4}$$

where  $\tilde{\varepsilon}_{SE}(I)$  and  $\tilde{\delta}_{SE}(I)$  are the standard numerical flat erosion and dilation of image  $I \in \mathcal{F}(\mathbb{E}, \mathcal{L})$ :

$$\tilde{\varepsilon}_{SE}(I)(x) = \left\{ I(y) : I(y) = \bigwedge [I(z)], z \in SE_x \right\} \tag{5}$$

$$\tilde{\delta}_{SE}(I)(x) = \left\{ I(y) : I(y) = \bigvee [I(z)], z \in \check{SE}_x \right\} \tag{6}$$

with  $SE_x$  being the structuring element centred at point  $x$  and  $\check{SE}$  is the reflected structuring element. If the inverse mapping  $h^{-1}$  is defined, the  $h$ -erosion and dilation can be explicitly written as:

---

<sup>2</sup> Adaptive in the sense that the mapping depend on the information contained in a multivariate image  $\mathbf{I}$ . The correct notation should be  $h(\cdot; \mathbf{I})$ . However, in order to make easier the understanding of the section we use  $h$  for adaptive mapping.

$$\varepsilon_{SE,h}(\mathbf{I})(x) = h^{-1}(\widetilde{\varepsilon}_{SE}(h(\mathbf{I}))(x)),$$

and

$$\delta_{SE,h}(\mathbf{I})(x) = h^{-1}(\widetilde{\delta}_{SE}(h(\mathbf{I}))(x)).$$

Of course, the inverse  $h^{-1}$  only exists if  $h$  is injective. In practice, we can impose the invertibility of  $h$  by considering a lexicographic ordering for equivalence class  $\mathcal{L}[\mathbf{x}]$ . In fact, this solution involves a structure of total ordering which allows to compute directly the  $h$ -erosion and dilation without using the inverse mapping, i.e.,

$$\varepsilon_{SE,h}(\mathbf{I})(x) = \left\{ \mathbf{I}(y) : \mathbf{I}(y) = \bigwedge_h [\mathbf{I}(z)], z \in SE_x \right\}, \quad (7)$$

and

$$\delta_{SE,h}(\mathbf{I})(x) = \left\{ \mathbf{I}(y) : \mathbf{I}(y) = \bigvee_h [\mathbf{I}(z)], z \in \check{SE}_x \right\}, \quad (8)$$

where  $\bigwedge_h$  and  $\bigvee_h$  are respectively the infimum and supremum according to the ordering  $\leq_h$ . Starting from the  $h$ -adjunction  $(\varepsilon_{SE,h}(\mathbf{I}), \delta_{SE,h}(\mathbf{I}))$ , all the morphological

filters such as the opening and closing have their  $h$ -counterpart, e.g., the  $h$  opening and closing are defined as

$$\gamma_{SE,h}(\mathbf{I}) = \delta_{SE,h}(\varepsilon_{SE,h}(\mathbf{I})), \quad \varphi_{SE,h}(\mathbf{I}) = \varepsilon_{SE,h}(\delta_{SE,h}(\mathbf{I})) \quad (9)$$

Similarly, any other mathematical morphology operator based on adjunction operators can be also extended to multivariate images. For instance, geodesic operators as opening by reconstruction[22], levelings [14], additive morphological decompositions [27] and so on.

### 3 Pre-Ordering a Vector Space

Let  $\mathbf{X}_I$  be the set of vector values of a given image  $\mathbf{I}$ , which can be viewed as a cloud of points in  $\mathbb{F}$ . Figure 3 shows an example of colour image  $\mathbf{I}$ , and its spectral representation as points  $\mathbf{X}_I$ . In general, pixel values in multispectral images are vectors defined in  $\mathbb{F} = \mathbb{R}^d$ . From previous section, for a given multivariate image  $\mathbf{I} : \mathbb{E} \rightarrow \mathbb{R}^d$ , the challenge to build complete lattice structures is to define a mapping  $h : \mathbb{R}^d \rightarrow \mathcal{L}$ , to obtain a mapping  $\mathbb{E} \rightarrow \mathcal{L}$ , where  $\mathcal{L}$  is a lattice. In this chapter, we consider the lattice  $\mathcal{L}$  of the extended real line  $(\overline{\mathbb{R}}, \leq)$  using  $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$  and  $\leq$  as the “less than or equal to” relation (the natural partial ordering). Many authors have already worked in this idea [1–3, 25]. Basically, three family of reduced

**Table 1** Different adaptive multivariate orderings implemented by  $h$ -mapping based reduced ordering. Note that  $\mathbf{x}$  is a vector which  $d$  components,  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d$ ,  $\mathbf{K} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$  is a kernel-induced distance, the sets  $B = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{|B|}\}$ ,  $F = \{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_{|F|}\}$  and  $T = \{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_{|T|}\}$  are the background, foreground and training, respectively. The matrix  $\mathbf{X}_I$  is an array containing the pixel information of a multidimensional image denoted by  $\mathbf{I}$ , i.e.,  $\mathbf{X}_I = [\mathbf{x}_1 \mathbf{x}_2 \dots \mathbf{x}_n]$  where  $\mathbf{X}_I$  has  $d$  rows and  $n$  columns. The orders considered here depend on an input image  $\mathbf{I}$ . A schematic representation of these orders is given in Fig. 2

Type of reduced mapping	$h(\mathbf{x}; \cdot)$
Unsupervised	
Linear dimensionality reduction [10]	$h_{\text{PCA}}(\mathbf{x}) = \sum_{i=1}^d \lambda^i \mathbf{x}_i$
Local unsupervised [11]	$h_{\text{LPCA}}(\mathbf{x}) = \sum_{i=1}^d \lambda_{\mathbf{x}}^i \mathbf{x}_i$
Distance based	
Referenced ordering [1]	$h_{\text{REF}}(\mathbf{x}; T) = \sum_{i=1}^{ T } \lambda_{\mathbf{x}}^i \mathbf{K}(\mathbf{t}_i, \mathbf{x})$
Supervised [25]	$h_{\text{SUPER}}(\mathbf{x}; B, F) = \sum_{k=1}^{ B } \lambda_{\mathbf{x}}^k \mathbf{K}(\mathbf{b}_k, \mathbf{x}) + \sum_{j=1}^{ F } \lambda_{\mathbf{x}}^j \mathbf{K}(\mathbf{f}_j, \mathbf{x})$
Anomaly based	
Projection Depth [26]	$h_{\text{ANOM}}(\mathbf{x}; \mathbf{I}) = \max_{\ \mathbf{u}\ =1} \frac{ \mathbf{u}^T \mathbf{x} - \text{med}(\mathbf{u}^T \mathbf{X}_I) }{\text{mad}(\mathbf{u}^T \mathbf{X}_I)}$

mappings  $h$  for a given  $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d) \in \mathbb{R}^d$  can be defined as it is illustrated in Table 1.

### 3.1 Unsupervised Ordering

That can be obtained by using the more representative projection in a statistical dimensional reduction technique, for example a linear approach as PCA [10] or some non-linear projections approach [12]. To illustrate, we consider the first projection to induce the ordering, i.e.,  $\mathbf{x}_1 \leq \mathbf{x}_2 \iff h_{\text{PCA}}(\mathbf{x}) \leq h_{\text{PCA}}(\mathbf{x}_2)$ , where  $h_{\text{PCA}}$  is the first eigenvector of the centred covariance matrix  $\mathbf{X}_I^T \mathbf{X}_I$ . The intuition behind this approach is simple and clear: pixels are ordered according to their representation in the projection with greatest variance. An example is illustrated in Fig. 4b. In this example, we can see that the induced minimum and maximum have no practical interpretation. A second disadvantage is that in this case, the minimum or maximum can drastically change by altering “a pixel” or a “limited number of pixels” in the original image  $\mathbf{I}$ .

### 3.2 Distance Based Ordering

Let us focus on the case of  $h$ -ordering based on distances. This approach is motivated by the intuition that order computation should be adaptive to prior information given by application interests.

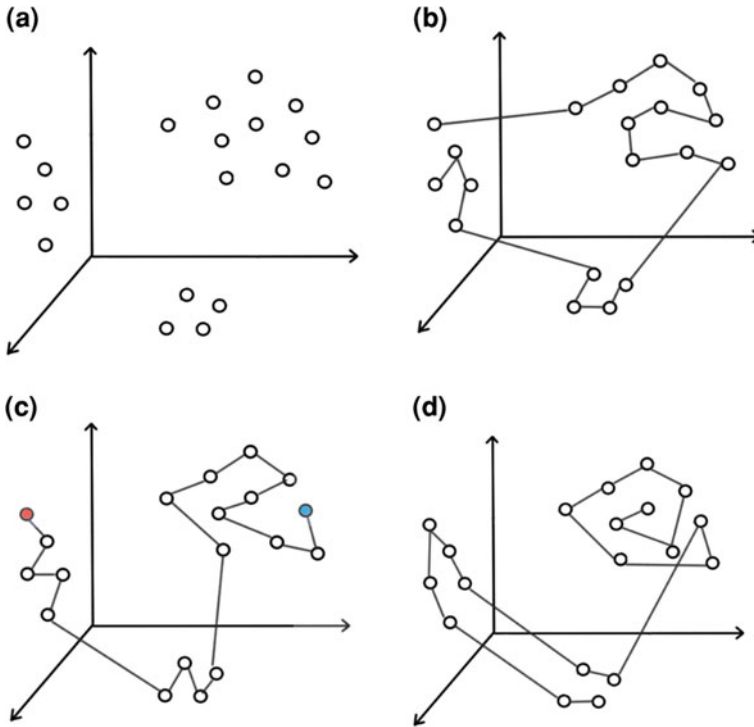
### 3.2.1 Referenced Ordering

As a starting point for distance based ordering, we consider the work of Angulo [1], who defines a function  $h_{\text{REF}}(\cdot, \mathbf{t})$  that computes the similarity for a given pixel  $\mathbf{x}$  to a colour reference  $\mathbf{t}$  by measuring its spectral distance, i.e.,  $\mathbf{x}_1 \leq_{h_{\text{REF}}} \mathbf{x}_2 \iff \mathbf{K}(\mathbf{x}_1, \mathbf{t}) \leq \mathbf{K}(\mathbf{x}_2, \mathbf{t})$ , where  $\mathbf{K} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$  is a kernel-induced distance[15]. The original formulation in [1] uses the case of Euclidean distance in the colour space as kernel-induced distance<sup>3</sup>. Thus, the ordering based on a reference spectrum exhibits a lattice where the minimum has been fixed. However, that maximum is associated with the “farthest” vector but that does not have a simple interpretation. To illustrate the result of this approach, we generalise the definition of a referenced order for a training set  $T$  as follows,  $\mathbf{x}_1 \leq_{h_{\text{REF}}} \mathbf{x}_2 \iff \min_i \|\mathbf{x}_1 - \mathbf{t}_i\| \geq \min_i \|\mathbf{x}_2 - \mathbf{t}_i\|$  for all  $\mathbf{t}_i \in T$ . The geometric interpretation is that  $h_{\text{REF}}(\mathbf{x}; T)$  is basically the distance in  $L_\infty$  of  $\mathbf{x}$  to the convex hull of vectors in  $T$  (if  $\mathbf{x}$  is not in the convex hull). Thus, is not so difficult to see that  $h_{\text{REF}}$  can be expressed as  $h_{\text{REF}}(\mathbf{x}; T) = \sum_{i=1}^{|T|} \lambda_x^i \mathbf{K}(\mathbf{t}_i, \mathbf{x})$  where  $\lambda_x^i \neq 0$  only for  $\arg \min_i \|\mathbf{x} - \mathbf{t}_i\|$ . Figure 4e shows the referenced mapping for the colour image in Fig. 4a. The training set are the pixel in the red region of Fig. 4d. Note that  $h_{\text{REF}}$  “detects” the girl but at the same time the border of the swimming-pool. Associated morphological adjunction and gradient are illustrated in Fig. 5g–i.

### 3.2.2 Supervised Ordering

A most general formulation for distance based ordering has been introduced in [25]. It defines a *h-supervised ordering* for every vector  $\mathbf{x} \in \mathbb{R}^d$  based on the subsets  $B = \{\mathbf{b}_1, \dots, \mathbf{b}_{|B|}\}$  and  $F = \{\mathbf{f}_1, \dots, \mathbf{f}_{|F|}\}$ , as a *h-ordering* that satisfies the following conditions:  $h(\mathbf{b}) = \perp$  then  $\mathbf{b} \in B$ , and  $h(\mathbf{f}) = \top$  then  $\mathbf{f} \in F$ . Note that  $\perp, \top$  are the smallest and largest element in the lattice  $\mathcal{L}$ . Such an *h-supervised ordering* is denoted by  $h_{\text{SUPER}}(\cdot; B, F)$ . Figure 2c illustrates the main intuition for a *h-supervised ordering* function: it is a linear ordering from the pixels in background set ( $B \subseteq \mathcal{L}(\perp)$ ) to the ones in foreground set ( $F \subseteq \mathcal{L}(\top)$ ). The main motivation of defining this new supervised ordering schema is to obtain maximum and minimum in the lattice  $\mathcal{L}$  interpretable with respect to sets  $B$  and  $F$ . It is important to remind that max and min are the basic words in the construction of all mathematical morphology operators. At this point, the problem is how to define an adequate supervised ordering for a given vector space  $\mathbb{F}$  and two pixel sets  $B, F$ . The approach introduced by [25] involves the computation of standard support vector machine (SVM) to solve a supervised classification problem to define the function  $h_{\text{SUPER}}(\mathbf{x}; B, F)$ . An amusing geometrical interpretation is based on results from [4], in where the ordering induced by  $h_{\text{SUPER}}$ , corresponds to the signed distance to the separating plane between the convex hull associated to  $F$  and the one containing the  $B$ . From [7], the solution of the classification case of SVM can be expressed as follows:

<sup>3</sup> In this case the sense of the inequality change, i.e.,  $\mathbf{x}_1 \leq_{h_{\text{REF}}} \mathbf{x}_2 \iff \|\mathbf{x}_1 - \mathbf{t}\|^2 \geq \|\mathbf{x}_2 - \mathbf{t}\|^2$ .

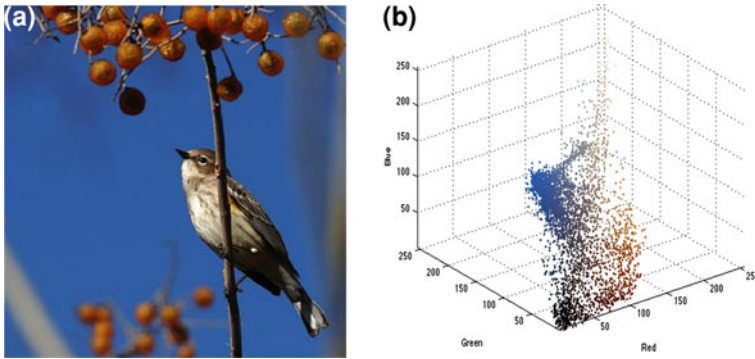


**Fig. 2** In the complete lattice representation, the set of pixels are analysed through a linear order relation denoted by  $h : \mathbb{R}^d \rightarrow \mathcal{L}$  in their spectral representation. In (c) the linear order starts from the background pixel (in red) and ends at the foreground pixel (in blue). In (d) the linear order starts from the centre of the greatest cluster in  $\mathbb{R}^d$ . (a) Spectral information in  $\mathbb{R}^d$ , (b) Example of linear ordering on (a), (c) Supervised ordering, the set of pixels are analysed through a total order relation, (d) Anomaly based ordering, the set of pixels are analysed through a total order relation

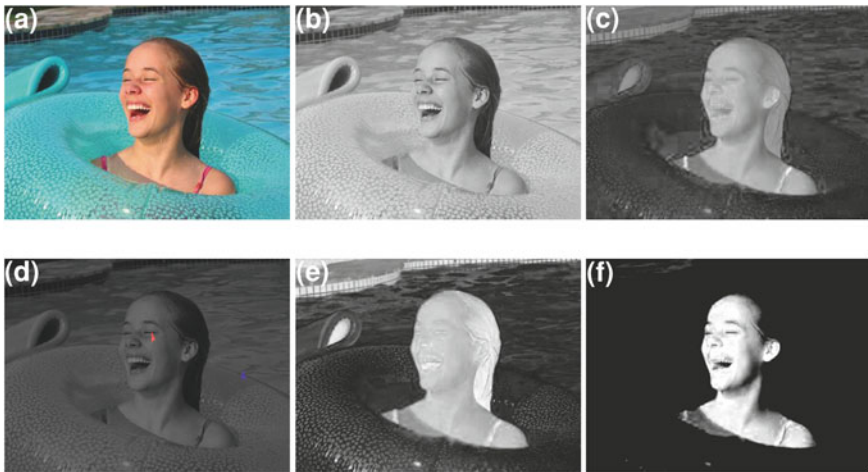
$$h_{\text{SUPER}}(\mathbf{x}; B, F) = \sum_{k=1}^{|B|} \lambda^k \mathbf{K}(\mathbf{b}_k, \mathbf{x}) + \sum_{j=1}^{|F|} \lambda^j \mathbf{K}(\mathbf{f}_j, \mathbf{x}) \quad (10)$$

where  $\lambda^k$  are computed simultaneous as a quadratic programming optimisation problem [7]. For all the examples, given in this chapter we have used a Gaussian Kernel, with the Euclidean distance between colour or spectra, i.e.  $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-c\|\mathbf{x}_i - \mathbf{x}_j\|^2)$ , where the constant  $c$  is obtained by cross-validation on the training set [7]. Results of this supervised ordering are illustrated in Fig. 4f. The  $h_{\text{SUPER}}$  matches our intuition of what should be maximum and what should be minimum in the image according to the couple  $\{B, F\}$  in Fig. 4d. The supervised adjunction is shown in Fig. 5j, k. Note that the supervised gradient in Fig. 5l is better defined on the contour of the girl in comparison to unsupervised and referenced orders. A second example is presented from the RGB image in Fig. 3 considering the training sets in



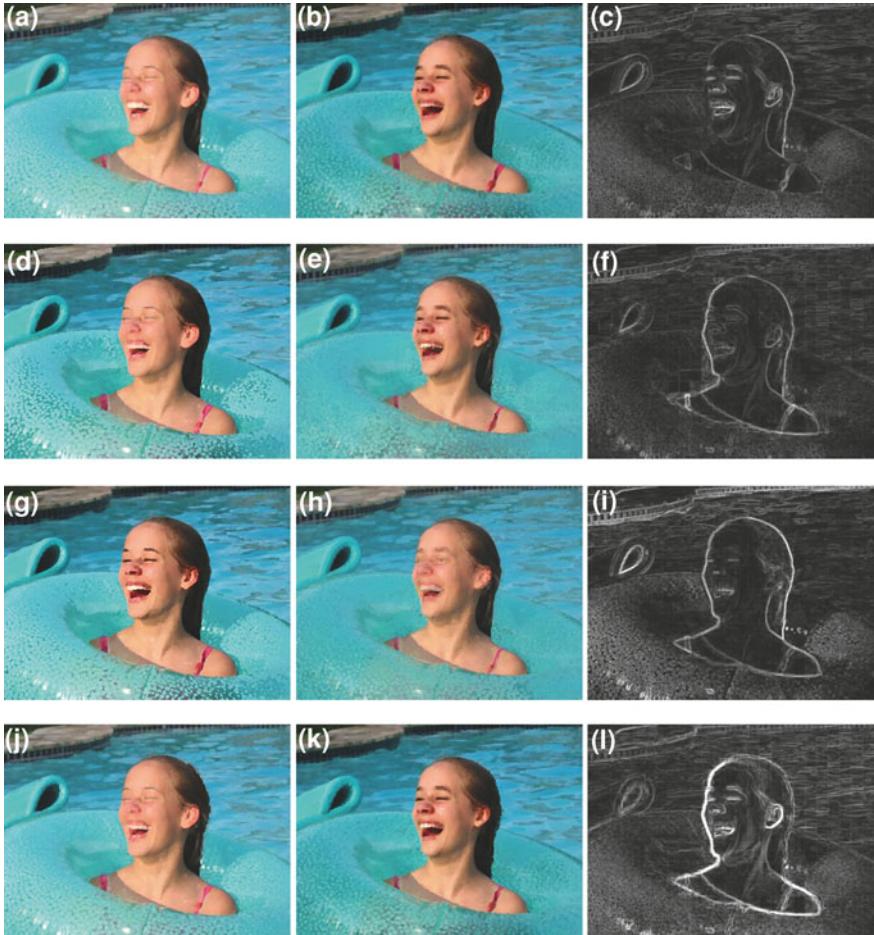


**Fig. 3** Spectral representation of a colour image in the RGB space. A spatial position  $x$  in the image  $\mathbf{I}$  contains three coordinates in the RGB-space represented by  $\mathbf{x}$ . (a) Original colour image denoted by  $\mathbf{I}$ , (b) Scatterplot of the three-channel image  $\mathbf{X}_{\mathbf{I}}$



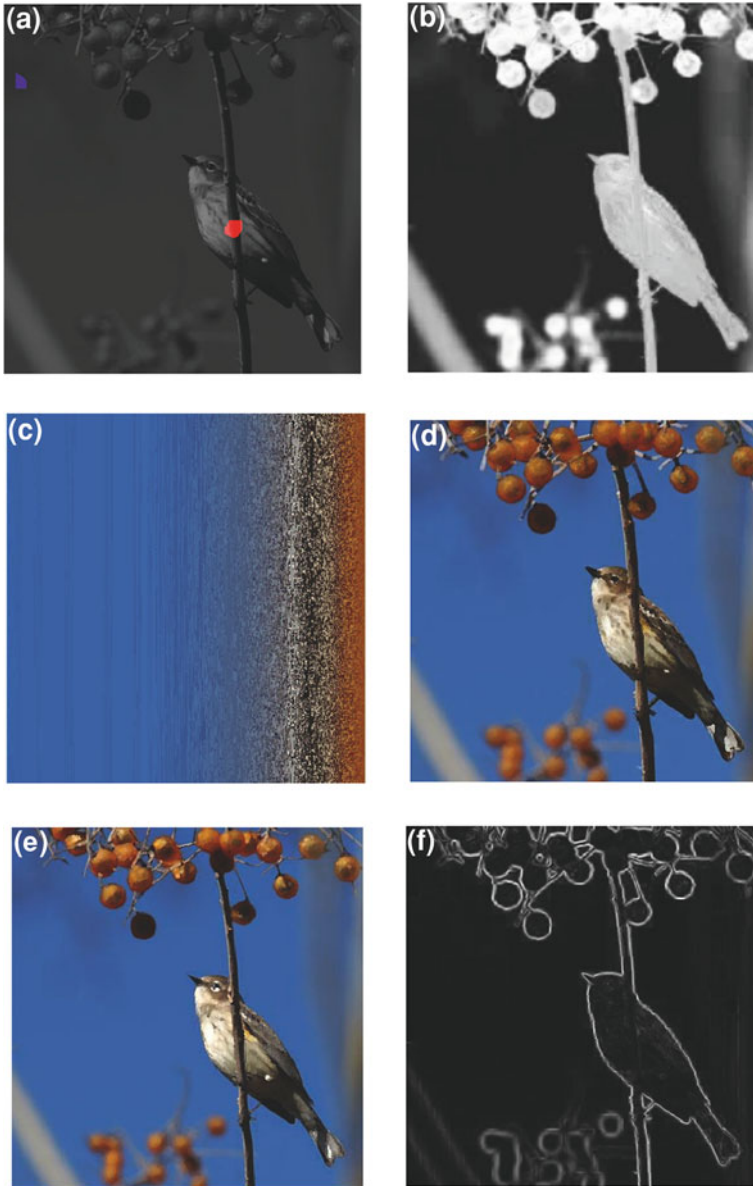
**Fig. 4** Comparison of different  $h$ -mappings considered in this chapter for a given colour image. Referenced and supervised  $h$ -mappings requires prior information given by the sets  $B$  and  $F$ . Anomaly based ordering is intrinsically adapted to the image. (a) Colour image:  $\mathbf{I}$ , (b)  $h_{PCA}$ , (c)  $h_{ANOM}$ , (d)  $B$  and  $F$  sets, (e)  $h_{REF}$ , (f)  $h_{SUPER}$

Fig. 6a. Note that the supervised lattice in Fig. 6c, is a mapping from the spectral information to a linear ordering (from top-left corner to bottom right corner). One advantage of the definition of  $h$ -ordering on vector space is that it can be applied directly to multispectral or even hyperspectral images. In order to illustrate this flexibility, we present the case of a RGB and Near-infrared (NIR) image in Fig. 7a, b from [6]. The spectral information is considered on  $\mathbb{R}^4$  and background and foreground sets are the spectra information contained in the marked regions in Fig. 7c. For purposes such as segmentation, we would use inner/outer markers-driven watershed

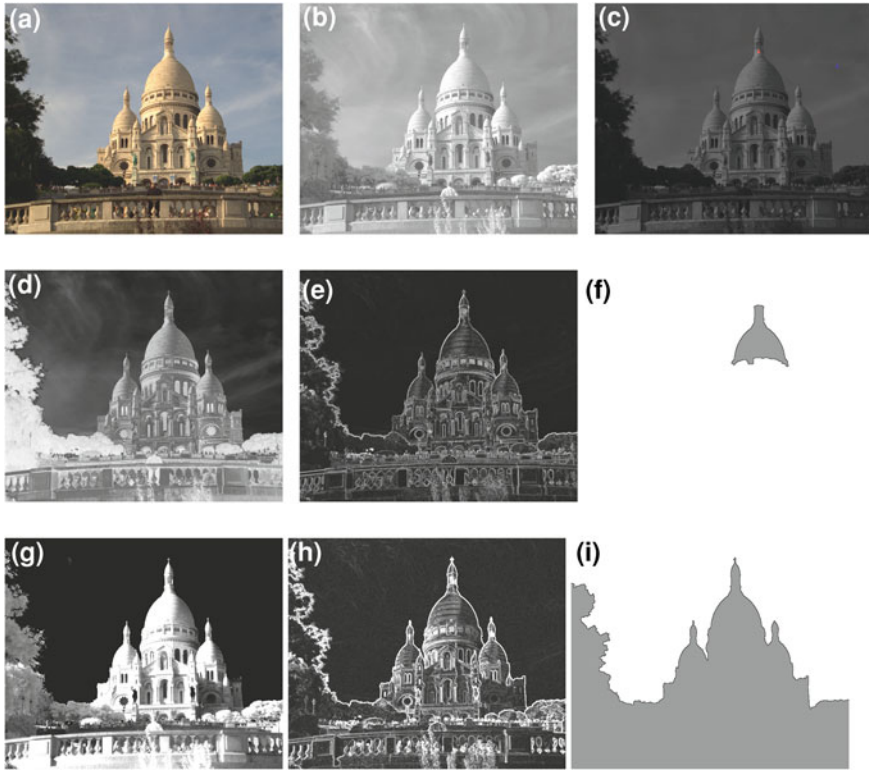


**Fig. 5** Comparison of colour dilation, erosion and associated gradient using different  $h$ -orderings (see Table 1). Gradients have been normalised from 0 to 1 to make easier the visual comparison. (a)  $\delta_{SE, h_{PCA}}(\mathbf{I})$ , (b)  $\varepsilon_{SE, h_{PCA}}(\mathbf{I})$ , (c) Gradient by  $h_{PCA}$ , (d)  $\delta_{SE, h_{ANOM}}(\mathbf{I})$ , (e)  $\varepsilon_{SE, h_{ANOM}}(\mathbf{I})$ , (f) Gradient by  $h_{ANOM}$ , (g)  $\delta_{SE, h_{REF}}(\mathbf{I})$ , (h)  $\varepsilon_{SE, h_{REF}}(\mathbf{I})$ , (i) Gradient by  $h_{REF}$ , (j)  $\delta_{SE, h_{SUPER}}(\mathbf{I})$ , (k)  $\varepsilon_{SE, h_{SUPER}}(\mathbf{I})$ , (l) Gradient by  $h_{SUPER}$

transformation [5]. Figure 7i depicts the results of the segmentation from the same set of markers Fig. 7c in both orders: referenced (f) and supervised (i). Notice that, supervised approach matches better the general structure of the original multispectral image than referenced one.



**Fig. 6** Background pixels are in blue, and foreground ones in red. The minimum in the supervised ordering is placed at the top left corner and the maximum at the bottom right corner. Morphological operators are computed by using a square of side 3 pixels as SE. **(a)** Background/foreground training set, **(b)**  $h_{SUPER}(\cdot; B, F)$ , **(c)** Learned order from **(b)**, **(d)**  $\delta_{SE, h_{SUPER}}(\mathbf{I})$ , **(e)**  $\varepsilon_{SE, h_{SUPER}}(\mathbf{I})$ , **(f)** Gradient by  $h_{SUPER}$  in **(b)**



**Fig. 7** Effect of the inclusion of supervised ordering in marked based segmentation. The spectra information of RGB+NIR image (a) are considered as vectors in  $\mathbb{R}^4$ . Background (resp. foreground) set are *blue* (resp. *red*) pixels in (c). (a) Original image and training sets, (b) NIR channel, (c) Training sets ( $F, B$ ), (d)  $h_{\text{REF}}(\cdot; F)$ , (e) Gradient of  $h_{\text{REF}}$ , (f) Marked watershed of  $h_{\text{REF}}$ , (g)  $h_{\text{SUPER}}(\cdot; B, F)$ , (h) Gradient of  $h_{\text{SUPER}}$ , (i) Marked watershed on  $h_{\text{SUPER}}$

### 3.3 Ordering Based on Anomalies

Distance based ordering approaches discussed above are valid if the pair set ( $B, F$ ) is available. Obviously, one cannot realistically believe that for every application the exact spectral information about the background of the image is available. Thus, if one gives up this paradigm, no other option different to unsupervised ordering remains. Therefore, in order to take advantage of the physical structure of an image, it was introduced in [26] an ordering based on “anomalies” with respect to a background associated to a majority of points. It is called *depth ordering* and is maximal in the “centre” of the spectral representation of a image  $\mathbf{I}$  and it produces a vector ordering “centre-outward” to the outliers in the vector space  $\mathbb{R}^d$ . In this paradigm, the assumption of existence of an intrinsic background/foreground representation is required, i.e., given a vector image  $\mathbf{I} : \mathbb{E} \rightarrow \mathbb{R}^d$ ,  $\mathbf{X}_{\mathbf{I}}$  has can be decomposed as

$\mathbf{X}_{\mathbf{I}} = \{\mathbf{X}_{B(\mathbf{I})}, \mathbf{X}_{F(\mathbf{I})}\}$  such that  $\mathbf{X}_{B(\mathbf{I})} \cap \mathbf{X}_{F(\mathbf{I})} = \emptyset$  and  $\mathbf{card}\{\mathbf{X}_{B(\mathbf{I})}\} > \mathbf{card}\{\mathbf{X}_{F(\mathbf{I})}\}$ . Roughly speaking, the assumption means: (1) the image has two main components: the background and the foreground; (2) There are more pixels in the background than in the foreground. Several examples of these kind of functionals have been analysed in [24]. However, we limited ourselves to the *statistical projection depth* case presented in [26] and defined by

$$h_{\text{ANOM}}(\mathbf{x}; \mathbf{I}) = \sup_{\|\mathbf{u}\|=1} \frac{|\mathbf{u}^T \mathbf{x} - \text{med}(\mathbf{u}^T \mathbf{X}_{\mathbf{I}})|}{\text{mad}(\mathbf{u}^T \mathbf{X}_{\mathbf{I}})} \quad (11)$$

where  $\text{med}$  denoted the *univariate median* and  $\text{mad}$  the *median absolute deviation*, i.e., the median of the differences with respect to the median. Note that the superscript  $T$  denotes matrix transposition. Let us now point out some aspects of (Eq. 11) in order to better characterise it. First, it is an anomaly based ordering, due to the fact that if  $\mathbf{X}_{\mathbf{I}} \sim \mathbb{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  a Gaussian distribution with mean vector  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$  then  $h_{\text{ANOM}}(\mathbf{x}; \mathbf{I})^2 \propto (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$ , the Mahalanobis distance (see [26] to details). Secondly, (Eq. 11) is invariant to affine transformations in the vector space  $\mathbb{R}^d$ . Third, unfortunately, the *exact computation* of (Eq. 11) is computationally intensive except when the number of pixels  $n$  is very small. However, we can compute a stochastic approximation by using a large number of random projections  $\mathbf{u}$  and computing the maximum for a given  $\mathbf{x}$  [26].

To summarise the above, the statistical projection depth function in (Eq. 11) induces an anomaly based ordering for images with background/foreground representation. That is an ordering based on a data-adapted function and in such a way that the interpretation of supremum and infimum operations is known a priori, because max values can be associated with “outlier” pixels in the high-dimensional space and min are “central” pixels in  $\mathbb{R}^d$  space. A simple example is illustrated in Fig. 4c where (Eq. 11) “detects” the girl thanks to the fact that her spectral information is unusual in comparison to the one from the swimming pool.

## 4 Implementation

Once a  $h$ -ordering has been defined, it becomes easy in practice to implement morphological transformations on multidimensional images such as colour or multispectral ones. Actually, we can use a scalar to code each pixel on the image, and the standard morphological transformations for grayscale images can be used directly. The result is deciphered by mapping back the total ordering in to the vector space. An effective implementation using a look-up table has been presented in [23]. A pseudo-code for a multivariate erosion<sup>4</sup> is shown in Algorithm 1 in Matlab notation.

<sup>4</sup> It is important to note that any adjunction based morphological transformations as openings, closings, levelings and so on, can be implemented in similar way, i.e., by changing the function  $\text{Erode}$  by another grey scale morphological transformation.

The index image and the sorted vector look-up table constructed above are used to generate an ordered table. At this point, any morphological transformation can be performed on the lattice image, which can be considered as a grey scale image. The output of the morphological transformation is converted back to the original vector space by replacing each pixel by its corresponding vector using a look-up table.

---

**Algorithm 1** Multivariate morphological Erosion: `h_Erode(im,h_function,se)`

---

**Require:** Multivariate image (`im`) of size  $n_1 n_2 \times d$ , the preorder (`h_function`) is a vector with  $n_1 n_2$  components and structuring element (`se`).

```
[·,b]=sort(h_function);
im_latt(b)=1:(n1n2);
im_latt=reshape(im_latt,n1,n2);
im_ero=Erode(im_latt,SE);
im_out=reshape(im(b(im_ero(:)),:),n1,n2,d);
```

---

## 5 Conclusions

Mathematical morphology is a non-linear methodology for image processing based on a pair of adjoint and dual operators, dilation and erosion, used to compute sup/inf-convolutions in local neighbourhoods. The extension of morphological operators to colour images has been the object of many works in the past; however, the generalisation of such colour approaches to multispectral images is not straightforward. In this chapter, we illustrated how kernel-based learning techniques and multivariate statistics can be exploited to design vector ordering and to include results of morphological operators in the pipeline of colour and multispectral image analysis. Two main families of ordering have been described. Firstly, we focused on the notion of supervised vector ordering which is based on a supervised learning formulation. A training set for the background and another training set for the foreground are needed as well as a supervised method to construct the ordering mapping. Secondly, we considered an (unsupervised) anomaly-based vector ordering based on statistical depth function computed by random projections. This led us to an intrinsic processing based on a background/foreground representation. We have illustrated in the examples the interest of morphological gradients (from pairs of multispectral dilation/erosion) for watershed segmentation. From a theoretical viewpoint, our framework is based on the theory of h-mapping adjunctions.

## References

1. Angulo J (2007) Morphological colour operators in totally ordered lattices based on distances: application to image filtering, enhancement and analysis. *Comput Vis Image Underst* 107(1–2):56–73

2. Aptoula E, Lefèvre S (2007) A comparative study on multivariate mathematical morphology. *Pattern Recogn* 40(11):2914–2929
3. Barnett V (1976) The ordering of multivariate data (with discussion). *J Roy Stat Soc: Ser A* 139(3):318–354
4. Bennett KP, Bredensteiner EJ (2000) Duality and geometry in svm classifiers. In: *Proceedings 17th International Conference on Machine Learning*, Morgan Kaufmann, pp 57–64
5. Beucher S, Meyer F (1993) The morphological approach to segmentation: the watershed transformation. *Mathematical morphology in image processing. Opt Eng* 34:433–481
6. Brown M, Süsstrunk S (2011) Multispectral SIFT for scene category recognition. *Computer Vision and Pattern Recognition (CVPR11)*. Colorado Springs, June, pp 177–184
7. Cristianini N, Shawe-Taylor J (2000) *An introduction to support vector machines and other kernel based learning methods*. Cambridge University Press, Cambridge
8. Goutsias J, Heijmans HJAM, Sivakumar K (1995) Morphological operators for image sequences. *Comput Vis Image Underst* 62(3):326–346
9. Heijmans HJAM, Ronse C (1990) The algebraic basis of mathematical morphology—part I: dilations and erosions. *Comput Vision Graph Image Process* 50:245–295
10. Jolliffe IT (1986) *Principal Component Analysis*. Springer-Verlag, New York
11. Kambhatla N, Leen TK (1997) Dimension reduction by local principal component analysis. *Neural Comp* 9(7):1493–1516
12. Lezoray O, Charrier C, Elmoataz A (2009) Learning complete lattices for manifold mathematical morphology. In: *Proceedings of the ISMM'09*, pp 1–4
13. Lorand R (2000) *Aesthetic order: a philosophy of order, beauty and art*, Routledge studies in twentieth century philosophy. Routledge, London
14. Meyer F (1998) The levelings. In: *Proceedings of the ISMM'98*, Kluwer Academic Publishers, USA, pp 199–206
15. Muller K, Mika S, Ritsch G, Tsuda K, Scholkopf B (2001) An introduction to kernel-based learning algorithms. *IEEE Trans on Neural Networks* 12:181–201
16. Najman L, Talbot H (2010) *Mathematical morphology: from theory to applications*. ISTE-Wiley, London
17. Ronse C (2011) Idempotent block splitting on partial partitions, I: isotone operators. *Order* 28:273–306
18. Salembier P, Serra J (1995) Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Trans Image Process* 4(8):1153–1160
19. Serra J (1982) *Image analysis and mathematical morphology*. Academic Press, USA
20. Serra J (1988) *Image analysis and mathematical morphology*. In: *Theoretical advances*, Vol. 2. Academic Press, USA
21. Serra J (2012) Tutorial on connective morphology. *IEEE J Sel Top Sign Proces* 6(7):739–752
22. Soille P (2003) *Morphological image analysis*. Springer-Verlag, New York
23. Talbot H, Evans C, Jones R (1998) Complete ordering and multivariate mathematical morphology. In: *Proceedings of the ISMM'98*, Kluwer Academic Publishers, USA, pp 27–34
24. Velasco-Forero S, Angulo J (2011) Mathematical morphology for vector images using statistical depth. *Mathematical Morphology and Its Applications to Image and Signal Processing*, volume 6671 of *Lecture Notes in Computer Science*. Springer, Berlin / Heidelberg, pp 355–366
25. Velasco-Forero S, Angulo J (2011) Supervised ordering in  $\mathbb{R}^p$ : application to morphological processing of hyperspectral images. *IEEE Trans Image Process* 20(11):3301–3308
26. Velasco-Forero S, Angulo J (2012) Random projection depth for multivariate mathematical morphology. *J Sel Top Sign Proces* 6(7):753–763
27. Velasco-Forero S, Angulo J (2013) Classification of hyperspectral images by tensor modeling and additive morphological decomposition. *Pattern Recogn* 46(2):566–577

# Morphological Template Matching in Color Images

Sébastien Lefèvre, Erchan Aptoula, Benjamin Perret  
and Jonathan Weber

**Abstract** Template matching is a fundamental problem in image analysis and computer vision. It has been addressed very early by Mathematical Morphology, through the well-known Hit-or-Miss Transform. In this chapter, we review most of the existing works on this morphological template matching operator, from the standard case of binary images to the (not so standard) case of grayscale images and the very recent extensions to color and multivariate data. We also discuss the issues raised by the application of the HMT operator to the context of template matching and provide guidelines to the interested reader. Various use cases in different application domains have been provided to illustrate the potential impact of this operator.

**Keywords** Mathematical morphology · Hit-or-miss transform · Template matching · Color image

---

S. Lefèvre (✉)  
Université de Bretagne-Sud, IRISA, Vannes, France  
e-mail: sebastien.lefevre@univ-ubs.fr

E. Aptoula  
Okan University, Istanbul, Turkey  
e-mail: erchan.aptoula@okan.edu.tr

B. Perret  
Laboratoire d'Informatique Gaspard-Monge, Université Paris-Est,  
Equipe A3SI, ESIEE Paris, France  
e-mail: b.perret@esiee.fr

J. Weber  
Université de Lorraine, LORIA-UMR 7503, Nancy, France  
e-mail: jonathan.weber@univ-lorraine.fr



# 1 Introduction

Mathematical Morphology is a very popular toolbox dating back to the 1960s and has achieved great successes. At its early years, mathematical morphology was dedicated to binary images. Binary morphological operators, while still very common in any digital image processing pipeline, are often applied in a second stage (*e.g.*, after a first thresholding step). The extension of mathematical morphology to grayscale images, both from theoretical and practical sides, is now mature after 20 years of developments in the field. It has broadened the possible uses of morphological operators and makes mathematical morphology a first-choice solution for digital image processing [36]. Multivariate images, such as color or multispectral images however constitute a more challenging task. Which is why multivariate mathematical morphology has been addressed only recently, mainly during the last decade. Despite being a recent research field, color morphology has been extensively explored and has led to many advances. As a matter of fact, several review papers or book chapters [1, 2, 4, 5] have already been published on this hot topic.

Our goal here is not to provide *yet another review on color morphology*. We rather focus on a specific problem fairly well addressed by mathematical morphology, namely template matching. Indeed, from the early years of mathematical morphology, this problem has been tackled with a morphological operator called the Hit-or-Miss Transform (HMT). The HMT has been widely used with binary images [42], relying on a simple pair of patterns to fit the foreground (object) and the background. Its use with grayscale images is more recent [8, 23, 31, 34] and has not yet reached the general image processing community. The various existing approaches for graylevel HMT have been recently reviewed by Murray and Marshall in [22]. The extension of the HMT to color or multispectral images has been achieved only very recently [2, 19, 37, 39] and is still at a preliminary stage of dissemination, while offering a great potential for color image processing. In this chapter, we thus aim to explore in a comprehensive way morphological template matching through the HMT, from the standard binary case to the most recent color extensions.

The organization of the chapter is the following. We first recall the initial definition of the HMT in the binary case (Sec. 2), before addressing the case of grayscale images (Sec. 3). We then further proceed to color images, and review the different HMT definitions proposed in the literature so far (Sec. 4). Each of these parts is provided with the necessary background in order to make the chapter self-contained. In Sec. 5, we discuss the implementation issues to be solved when using HMT for practical template matching applications. Such existing applications are then reviewed in Sec. 6. Section 7 concludes this chapter and provides suggestions for future research directions.

## 2 Template Matching in Binary Images

In this section, we recall the basics of Mathematical Morphology in the binary case, with adequate definitions and notations. The HMT on binary images will then be presented and discussed, with comprehensive examples.

### 2.1 Binary Mathematical Morphology

Let  $E$  be an Euclidean or digital space (*i.e.*  $E = \mathbb{R}^n$  or  $E = \mathbb{Z}^n$  with  $n \in \mathbb{N}^*$ ). Let  $X$  be a subset of  $E$ . Let  $\mathcal{P}(X)$  denote the class of all subsets of  $X$ :  $\mathcal{P}(X) = \{Y \subseteq X\}$ . Let  $X^c$  denote the complement of  $X$ , *i.e.* the set of all points of  $E$  that do not belong to  $X$ :  $X^c = \{y \in E \mid y \notin X\}$ . Let  $\check{X}$  denote the reflection of  $X$ , *i.e.* the set  $X$  transformed by a central symmetry:  $\check{X} = \{-x \mid x \in X\}$ . Finally, we will denote by  $X_p$  the translate of  $X$  by  $p \in E$  defined by  $X_p = \{x + p \mid x \in X\}$ .

The basic operators of erosion and dilation of  $X$  by the Structuring Element (SE)  $B \in \mathcal{P}(E)$  are written respectively  $\varepsilon_B(X)$  and  $\delta_B(X)$ . They are defined using the Minkowski subtraction ( $\ominus$ ) and addition ( $\oplus$ ):

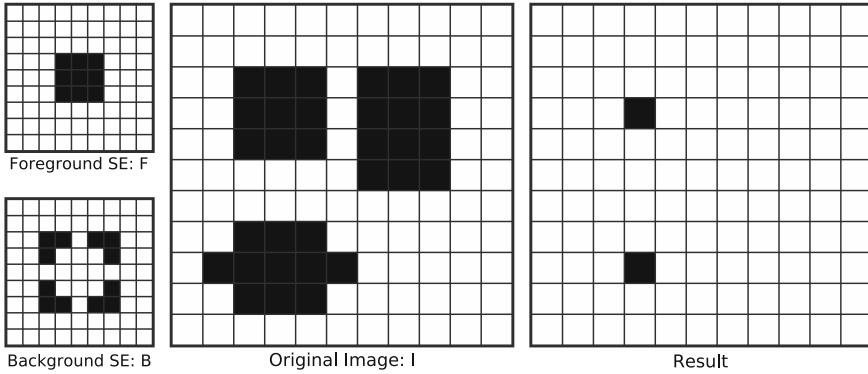
$$\varepsilon_B(X) = X \ominus B = \bigcap_{b \in B} X_{-b} \quad (1)$$

$$\delta_B(X) = X \oplus B = \bigcup_{b \in B} X_b \quad (2)$$

### 2.2 Binary HMT

Being given a subset  $X$  of  $E$ , the principle of the HMT is to look for all positions  $p$  in  $X$  where a structuring element  $F$  called the *foreground* fits in the shape ( $F_p \subseteq X$ ) while another structuring element  $B$  called the *background* fits in the complement of  $X$  ( $B_p \subseteq X^c$ ). The HMT of  $X$  by the structuring elements  $(F, B)$ , noted  $\text{HMT}_{F,B}(X)$ , can be defined in terms of Minkowski subtraction and addition (structuring erosion and dilation):

$$\begin{aligned} \text{HMT}_{F,B}(X) &= \{p \in X \mid F_p \subseteq X \text{ and } B_p \subseteq X^c\} \\ &= \left\{ p \in X \mid F_p \subseteq X \subseteq B_p^c \right\} \\ &= (X \ominus F) \cap (X^c \ominus B) \\ &= \varepsilon_F(X) \setminus \delta_{\check{B}}(X) \end{aligned} \quad (3)$$



**Fig. 1** Example of application of binary HMT to detect  $3 \times 3$  squares with possible extensions along edges [27]. The pixels belonging to the images are in black while those belonging to the complement are in white. The top left image represents the foreground SE:  $F$ , origin of the image is at the centre of the square. The bottom left image is the background SE  $B$ , with same origin as  $F$ . The middle image is the original image  $I$ . And the right image is the result of the HMT applied on  $I$  with SEs  $F$  and  $B$

A direct consequence of this definition is that if  $F$  and  $B$  have a non-empty intersection, then  $\text{HMT}_{F,B}(X)$  is empty for all  $X$  in  $\mathcal{P}(E)$  (a given pixel cannot simultaneously belongs to the foreground and to the background). One can also note that: first, the second formulation of (Eq. 3) is known as the *interval operator* from [15] and second, the complementation does not appear in the last formulation which allows to naturally extend the HMT to gray-level images where complementation is not defined.

Figure 1 illustrates the application of the hit-or-miss transform on a binary image. Our goal here is to detect  $3 \times 3$  squares with possible extensions along edges. It is noteworthy that the two structuring elements do not need to cover the whole neighborhood of a pixel and some pixels, as the middle of the edges of the square, can be excluded from the decision process by neither putting them in  $F$  nor in  $B$ . This is a first attempt to ensure robustness to uncertainty or noise. Some more advanced solutions, not specific to binary images will be reviewed in Sec. 5.

### 3 Template Matching in Grayscale Images

We recall here basics of Mathematical Morphology in the grayscale case, with adequate definitions and notations. The various definitions of the HMT on grayscale images (e.g., [8, 23, 31, 34]) will then be presented and discussed, with comprehensive examples.

### 3.1 Grayscale Mathematical Morphology

We now consider the case of grayscale images, *i.e.* of mappings from  $E$  to the set of values  $T$  ( $T^E$ ). Moreover, we assume that  $T$  is a complete lattice, *i.e.* a non empty set equipped with a partial ordering  $\leq$ , an infimum  $\bigwedge$  and a supremum  $\bigvee$  such that the infimum and supremum of any non empty subset of  $T$  belongs to  $T$  (for all  $A \subseteq T$ ,  $\bigwedge A \in T$  and  $\bigvee A \in T$ ). The lattice  $T$  is thus bounded by its least element  $\perp = \bigwedge T$  and its greatest element  $\top = \bigvee T$ . For grayscale images we usually take  $T = \mathbb{R} = \mathbb{R} \cup \{-\infty, \infty\}$  or  $T = \mathbb{Z} = \mathbb{Z} \cup \{-\infty, \infty\}$ . The notion of structuring element naturally extends to the one of structuring function (SF) which is simply an element of  $T^E$ .

The definitions of the binary erosion (Eq. 1) and binary dilation (Eq. 2) then naturally extend to grayscale images. Let  $F$  in  $T^E$  be a grayscale image and  $V$  in  $T^E$  be a structuring function. The grayscale erosion  $\varepsilon_V(F)$  and dilation  $\delta_V(F)$  of  $F$  by  $V$  are then defined for all  $p \in E$  by:

$$\varepsilon_V(F)(p) = \bigwedge_{x \in \text{supp}(V)} \{F(p+x) - V(x)\} \tag{4}$$

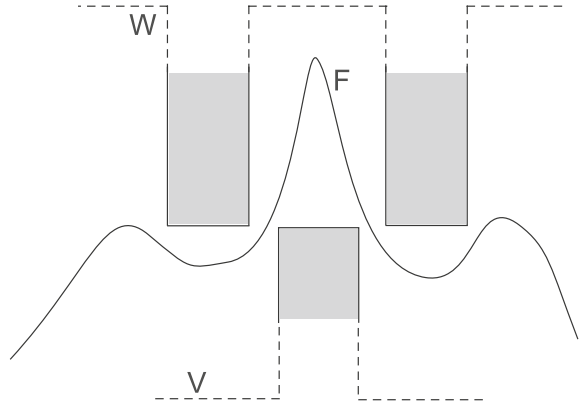
$$\delta_V(F)(p) = \bigvee_{x \in \text{supp}(V)} \{F(p-x) + V(x)\} \tag{5}$$

where  $\text{supp}(V)$  is the *support* of  $V$ , *i.e.* the set of points of  $E$  where  $V$  is greater than the least element  $\perp$ :  $\text{supp}(V) = \{p \in E \mid V(p) \neq \perp\}$ . These formulas can lead to disinclination like  $+\infty$  plus  $-\infty$ . To keep consistency  $+\infty$  plus  $-\infty$  must be valued as  $-\infty$  in Eq. 5 and as  $+\infty$  in Eq. 4.

### 3.2 Grayscale HMT

The binary HMT was first extended to grayscale images by Ronse [31] and then many other definitions followed: by Shaefer [32], Khosravi [18], Raducanu [29], Soille [36], Barat [7], Perret [27]. A recent survey on grayscale HMT is given in [22]. These various definitions have been recently unified into a common theoretical framework for graylevel *interval operators* [23]. In the binary case, the interval operator (second line of Eq. 3) looks for each translation  $p \in E$  if the image fits between the background SE translated at  $p$  and the complement of the background SE also translated at  $p$ . In the grayscale case, the sought template is translated not only horizontally (by a point  $p \in E$ ), but vertically as well (by a finite graylevel  $t \in T$ ) in an attempt to detect the positions where it fits (Fig. 2). Specifically, we will note  $V_{(p,t)}$  the translation of  $V \in T^E$  by a couple  $(p,t) \in E \times T$ : for all  $x \in E$ ,  $V_{(p,t)}(x) = V(x-p) + t$ .

**Fig. 2** The integral interval operator, (Eq. 13) [6]



According to the unified theory for grayscale HMT [23], a grayscale HMT is decomposed into two stages:

- the *fitting*, where the locations fitting the given structuring functions, describing the sought template, are computed,
- and *valuation*, where the resulting image containing the previously detected locations is constructed.

Let us observe that in the binary case, the fitted pixels are the ones retained by the HMT, and their valuation is simply equal to one (or foreground).

### 3.2.1 Fitting

We first describe the different fittings. Let  $F \in T^E$  be a grayscale image and  $V, W \in T^E$  be a couple of structuring functions describing the sought template such that  $V \leq W$ . Formally a fitting is a mapping from  $T^E$  into  $\mathcal{P}(E \times T')$  where  $T'$  can be a subset of  $T$  or any set of values. A first fitting involved in Ronse’s HMT is given by:

$$H_{V,W}(F) = \{(p, t) \in E \times T \mid V_{(p,t)} \leq F \leq W_{(p,t)}\} \tag{6}$$

$$= \{(p, t) \in E \times T \mid \varepsilon_V(F)(p) \leq t \leq \delta_{W^*}(F)(p)\} \tag{7}$$

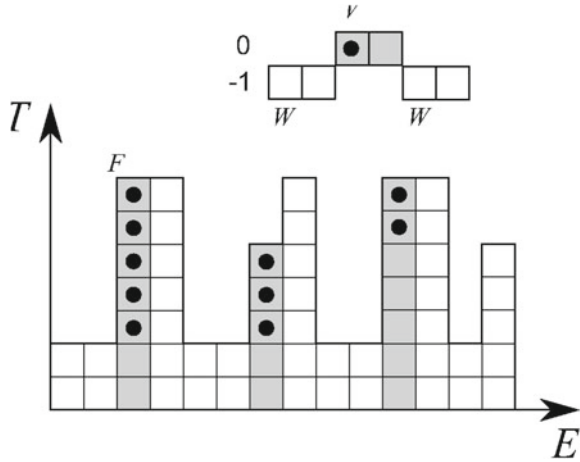
where  $W^* : x \rightarrow -W(-x)$ , is the *dual* of  $W$ .  $H_{V,W}(F)$  is thus the set of all translations where the image lies between both structuring functions (Fig. 3).

A second fitting, involved in Soille’s and Barat’s HMT is:

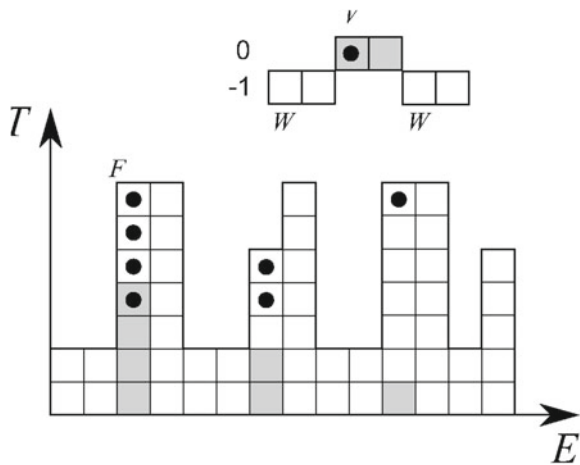
$$K_{V,W}(F) = \{(p, t) \in E \times T \mid V_{(p,t)} \leq F \ll W_{(p,t)}\} \tag{8}$$

$$= \{(p, t) \in E \times T \mid \varepsilon_V(F)(p) \leq t < \delta_{W^*}(F)(p)\} \tag{9}$$

**Fig. 3** Illustration of  $H_{V,W}$  (Eq. (6)). We consider a function  $F$  from  $E = \mathbb{Z}$  into  $T = \overline{\mathbb{Z}}$  and the two structuring functions  $V$  and  $W$  (the origin is on the left pixel of  $V$ ). The result of  $H_{V,W}(F)$  is the set of points of  $E \times T$  marked by a black circle inside  $F$ . We also give the result of the application of the supremal valuation (Eq. (12)) of  $H_{V,W}(F)$  in gray, *i.e.* the result of  $RHMT_{V,W}(F)$  (Eq. (14)).



**Fig. 4** Illustration of  $K_{V,W}$  (Eq. 8). We consider a function  $F$  from  $E = \mathbb{Z}$  into  $T = \overline{\mathbb{Z}}$  and the two structuring functions  $V$  and  $W$  (the origin is on the left pixel of  $V$ ). The result of  $K_{V,W}(F)$  is the set of points of  $E \times T$  marked by a black circle inside  $F$ . We also give the result of the application of the integral valuation (Eq. 13) of  $K_{V,W}(F)$  in gray, *i.e.* the result of  $SHMT_{V,W}(F)$  (Eq. 15)

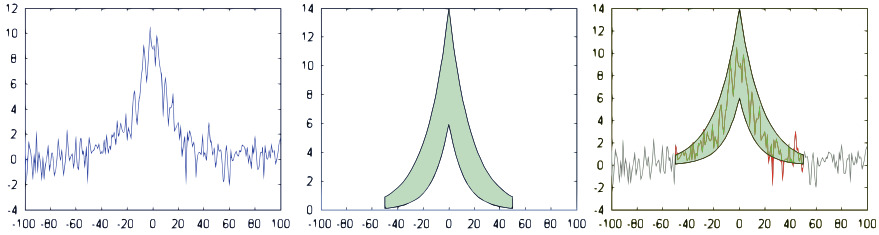


where  $F \ll W$  means that there is some  $h > 0$  such that for every  $p \in E$  we have  $F(p) \leq W(p) - h$ .  $K_{V,W}(F)$  is thus the set of all translations where the image lies between both structuring functions and does not touch  $W$  (Fig. 4).

Finally, a third fitting involved in Perret’s HMT is given by:

$$P_{V,W}(F) = \left\{ \left( p, \frac{| \{ q \in S \mid V(q) + t \leq F(p+q) \leq W(q) + t \} |}{|S|} \right) \in E \times [0, 1] \mid t \in T \right\} \quad (10)$$

where  $S = \{x \in E \mid V(x) \neq \perp \text{ or } W(x) \neq \top\}$  and  $|X|$  is the cardinal of the set  $X$ . Hence,  $P_{V,W}(F)$  does not look for locations where both structuring functions completely fit the image. Instead, it measures at each location how well both structuring functions fit the image by counting, among the points of the image that lies in the support of one of the SF, the ratio of points that fits between both SF (Fig. 5).



**Fig. 5** Example of application of the  $P_{V,W}(F)$  (Eq. 10) to detect an exponential profile with Gaussian noise [27]. The first image represents a 1D noisy signal. The second image represents the uncertainty area defined by the 2 SFs  $V$  (lower function) and  $W$  (upper function). The third image shows how well the pattern can fit the signal

### 3.2.2 Valuation

Likewise, there are several types of valuations. Formally, a valuation  $\eta$  is a mapping that associates a (grayscale) image  $(T^E)$  to any result of a fitting  $(\mathcal{P}(E \times T'))$ . Let  $Y \in \mathcal{P}(E \times T')$ , the simplest valuation  $\eta^B(Y)$ , the *binary* valuation, consists in taking the set of points  $p \in E$  for which there is at least one  $t \in T$  such that  $(p, t)$  belongs to  $Y$ .

$$\eta^B(Y)(p) = \begin{cases} 1 & \text{if } \exists t \in T', (p, t) \in Y \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

Another valuation is the *supremal* valuation  $\eta^S(Y)$ , which for every point  $p \in E$  of fit couples  $\{(p, t)\} \subseteq Y$ , takes the supremum of  $t$  (Fig. 3):

$$\eta^S(Y)(p) = \sup \{t \mid (p, t) \in Y\} \quad (12)$$

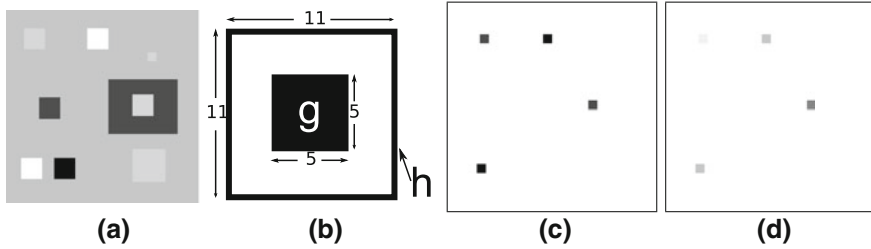
Finally there is the *integral* valuation  $\eta^I(Y)$  which instead for every point  $p$  of fit couples  $\{(p, t)\} \subseteq Y$ , uses the length of the interval of  $t$  for which the couples  $(p, t)$  fit (Fig. 4).

$$\eta^I(Y)(p) = \text{mes} \{t \mid (p, t) \in Y\} \quad (13)$$

where *mes* is an appropriate measure of intervals in  $T'$  (simply the Lebesgue measure if  $T'$  is an interval of  $\mathbb{R}$  or  $\mathbb{Z}$ ).

### 3.3 HMT Definitions

The different combinations of fittings and valuations allow to recover the different definitions of HMT. Let  $F, V, W \in T^E$  with  $V \leq W$  and  $p \in E$ , among others, we mention:



**Fig. 6** Grayscale HMT : **a** original image ( $64 \times 64$  pixels); **b** couple of structuring functions used within the HMT; **c** result with Ronse's, **d** and Soille's definition (results are displayed in inverse gray levels and for the sake of clarity,  $\perp$  value was replaced with 0) [39]

$$\begin{aligned}
 (\text{Ronse}) \quad RHMT_{V,W}(F)(p) &= \eta^S(H_{V,W}(F))(p) \\
 &= \begin{cases} \varepsilon_V(F)(p) & \text{if } \varepsilon_V(F)(p) \geq \delta_{W^*}(F)(p), \\ \perp & \text{otherwise} \end{cases} \quad (14)
 \end{aligned}$$

$$\begin{aligned}
 (\text{Soille}) \quad SHMT_{V,W}(F)(p) &= \eta^I(K_{V,W}(F))(p) \\
 &= \max \{ \varepsilon_V(F)(p) - \delta_{W^*}(F)(p), 0 \} \quad (15)
 \end{aligned}$$

$$\begin{aligned}
 (\text{Perret}) \quad PHMT_{V,W}(F)(p) &= \eta^S(P_{V,W}(F))(p) \\
 &= \max_{t \in T} \frac{|\{q \in S \mid V(q) + t \leq F(p+q) \leq W(q) + t\}|}{|S|} \quad (16)
 \end{aligned}$$

Ronse's and Soille's HMT are illustrated in Fig. 3 and 4 and further compared in Fig. 6 while an application of Perret's HMT is given in Sect. 6.1.

## 4 Template Matching in Color and Multivariate Images

The extension of the HMT to color and more generally to multivariate images such as multi- and hyper-spectral data, can amplify the application potential of the operator at a significant level. Of particular importance in this context are applications regarding color object detection from color data, target detection from remote sensing data, as well as the detection of challenging celestial objects from astronomical imaging sources.

However, given the multitude of existing approaches for the definition of the grayscale HMT, it is no surprise that the situation is no less clearer with multivariate images. In fact, conversely to grayscale morphological image processing, there is not even a generally accepted color mathematical morphology framework, let alone a standardized multivariate HMT. Nevertheless, the increasingly widespread availability of multivariate images, especially in the context of remote sensing, as well as recent advances in color morphology, have provided the missing impetus that has



led to the intensification of the research work on extending the HMT to color, multi- and hyper-spectral data.

We will start this section by first recalling the basic theoretical implications of using multivariate images to the mathematical morphology theory, and elaborate on the difficulties of its extension to this type of images. Then, we will present the different methods that have been developed especially in the past few years for applying the HMT to multivariate images.

#### 4.1 Multivariate Mathematical Morphology

According to the lattice based approach to mathematical morphology [30], digital images are represented as mappings  $F : E \rightarrow T$  between the discrete coordinate grid  $E$  and the set of pixel values  $T$ , with  $T$  being a complete lattice. More precisely, imposing a complete lattice structure on an arbitrary set of pixel values  $T$  is possible, if  $T$  is equipped with at least a partial ordering relation which enables the computation of the infimum and supremum of any non-empty subset of  $T$ . Consequently, the morphological operators can be in fact applied to any type of image data, as long as the set of pixel values  $T$  possesses a complete lattice structure. For a detailed account of multivariate morphology the reader is referred to Refs. [14, 33].

For instance, in the case of continuous multidimensional grayscale images where  $F : \mathbb{R}^d \rightarrow \overline{\mathbb{R}}$ , it suffices to employ the usual comparison operator  $\leq$ , in order to induce a complete lattice structure on  $\overline{\mathbb{R}}$ . Likewise, the inclusion operator  $\subseteq$  can be used with binary images  $F : \mathbb{R}^d \rightarrow \{0, 1\}$ . However, if we now consider multivariate images  $\mathbf{F} : \mathbb{R}^d \rightarrow \overline{\mathbb{R}}^n$ ,  $n > 1$ , where  $n = 3$  for the specific case of color images, it becomes problematic to find an ordering relation for the vectors of  $\overline{\mathbb{R}}^n$ , due to the fact that there is no universal method for ordering multivariate data. As a result, in the last 15 years several ordering approaches have been explored with the end of extending mathematical morphology to multivariate images, for a detailed survey of which the reader is referred to Ref. [2].

We briefly recall the main properties of an ordering, *i.e.* a binary relation  $\leq$  on a set  $\mathcal{T}$  being reflexive ( $\forall x \in \mathcal{T}, x \leq x$ ), anti-symmetric ( $\forall x, y \in \mathcal{T}, x \leq y$  and  $y \leq x \Rightarrow x = y$ ), and transitive ( $\forall x, y, w \in \mathcal{T}, x \leq y$  and  $y \leq w \Rightarrow x \leq w$ ). An ordering is total if the totality statement ( $\forall x, y \in \mathcal{T}, x \leq y$  or  $y \leq x$ ) holds, partial otherwise. It is a pre-ordering if the anti-symmetry statement does not hold.

Hence, the HMT being a morphological tool, its extension to multivariate images also requires the implication of a vector ordering. We will now proceed to examine various solutions developed for applying the HMT to color or to multivariate data in general.

## 4.2 HMT Based on a Vector Ordering

Aptoula et al. [6] provide the first study on the theoretical requirements of a vector HMT (VHMT). In detail, the initial step for extending the HMT to multivariate data, consists in defining the erosion and dilation operators for multivariate images in combination with multivariate structuring functions (SF). More precisely, these operators are based on horizontal translations (by a point  $p \in E$ ) as well as on vertical ones (by a finite pixel value  $\mathbf{t} \in T$ ) as in the grayscale case, the difference is however that pixel values are now multi-dimensional; in particular, given a multivariate image  $\mathbf{F} : E \rightarrow T$ :

$$\forall (p, \mathbf{t}) \in E \times T, \mathbf{F}_{(p, \mathbf{t})}(x) = \mathbf{F}(x - p) + \mathbf{t} \quad (17)$$

Furthermore, according to the fundamental Refs. [16, 17] of Heijmans and Ronse, translations need to be complete lattice automorphisms (*i.e.* bijections  $T \rightarrow T$  that preserve order, and whose inverse also preserve order). Consequently, the vector ordering ( $\leq_v$ ) from which the complete lattice is derived, must be translation invariant. In other words:

$$\forall \mathbf{w}, \mathbf{w}', \mathbf{t} \in T, \mathbf{w} \leq_v \mathbf{w}' \Leftrightarrow \mathbf{w} + \mathbf{t} \leq_v \mathbf{w}' + \mathbf{t} \quad (18)$$

Thus one can give the definition of the erosion and dilation respectively of a multivariate image  $\mathbf{F}$  by a multivariate SF  $\mathbf{B}$ :

$$\varepsilon_{\mathbf{B}}(\mathbf{F})(p) = \inf_{x \in \text{supp}(\mathbf{B})} \{\mathbf{F}(p + x) - \mathbf{B}(x)\} \quad (19)$$

$$\delta_{\mathbf{B}}(\mathbf{F})(p) = \sup_{x \in \text{supp}(\mathbf{B})} \{\mathbf{F}(p - x) + \mathbf{B}(x)\} \quad (20)$$

where  $\text{supp}(\mathbf{B}) = \{p \in E \mid \mathbf{B}(p) > \perp\}$ , while  $\inf_v$  and  $\sup_v$  denote respectively the infimum and supremum based on the vector ordering ( $\leq_v$ ) under consideration. Hence, these formulations form an adjunction as demanded by Refs. [16, 17] and besides, with a flat SE (*i.e.*  $\forall x, \mathbf{B}(x) = \mathbf{0}$ ) they are reduced to the flat multivariate erosion and dilation formulations. Furthermore, thanks to (Eq. 18), both fitting equivalences between (Eqs. 6 and 7) as well as between (Eqs. 8 and 9) become directly extendable to this case by replacing the grayscale operators with their multivariate counterparts. As to valuation, the same options as before are available, however the supremum is of course now computed among vectors through the vector ordering in use. In the case of integral valuation, a vector distance now can be used in order to measure the distance among the vectors that have fit. Consequently one can express the multivariate versions of the integral and supremal interval operators respectively as follows:

$$\eta^I(K_{\mathbf{V},\mathbf{W}}(\mathbf{F}))(p) = \begin{cases} \|\varepsilon_{\mathbf{V}}(\mathbf{F})(p) - \delta_{\mathbf{W}^*}(\mathbf{F})(p)\| & \text{if } \varepsilon_{\mathbf{V}}(\mathbf{F})(p) >_v \delta_{\mathbf{W}^*}(\mathbf{F})(p) \\ 0 & \text{otherwise.} \end{cases} \quad (21)$$

$$\eta^S(H_{\mathbf{V},\mathbf{W}}(\mathbf{F}))(p) = \begin{cases} \varepsilon_{\mathbf{V}}(\mathbf{F})(p) & \text{if } \varepsilon_{\mathbf{V}}(\mathbf{F})(p) \geq_v \delta_{\mathbf{W}^*}(\mathbf{F})(p) \\ \perp & \text{otherwise.} \end{cases} \quad (22)$$

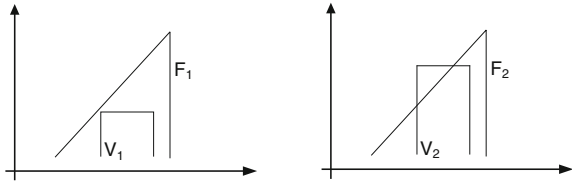
where  $\mathbf{V} \leq_v \mathbf{W}$ . As far as  $\eta^I$  is concerned, it provides a non-zero output at positions where  $\mathbf{V} \leq_v \mathbf{F} \ll_v \mathbf{W}$  according to the ordering in use. It should also be noted that the grayscale valuation choice by means of the Euclidean norm ( $\|\cdot\|$ ) is arbitrary, and a multi-dimensional valuation is of course possible. As to  $\eta^S$ , it produces a non-zero output at positions where  $\mathbf{V} \leq_v \mathbf{F} \leq_v \mathbf{W}$ .

Therefore, the only obstacle preventing the definition of a VHMT is a translation preserving vector ordering. This useful property, among others, is provided by the standard *lexicographical ordering* which is frequently employed in the context of multivariate morphology [3]:

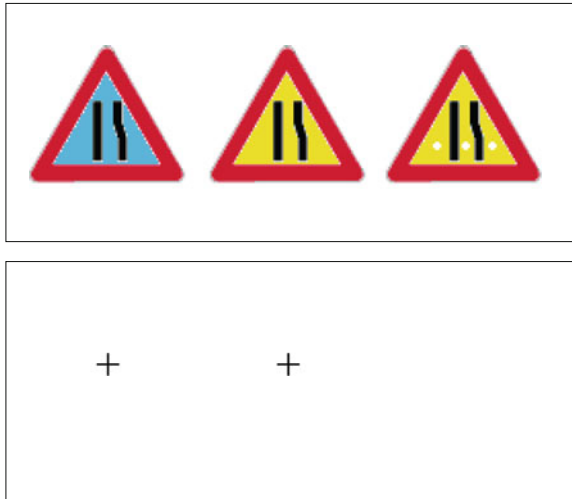
$$\forall \mathbf{v}, \mathbf{v}' \in \mathbb{R}^n, \mathbf{v} <_L \mathbf{v}' \Leftrightarrow \exists i \in \{1, \dots, n\}, (\forall j < i, v_j = v'_j) \wedge (v_i < v'_i) \quad (23)$$

while for instance its recent variation  $\alpha$ -modulus lexicographical [1], does not provide it. Moreover, the chosen ordering directly affects the behavior of VHMT. For instance, in the case of lexicographical ordering, which is known for its tendency of prioritising the first vector component [3], this property can be observed in the detection process of VHMT. In particular, during the fitting stage where the erosion and dilation outputs are computed, since it is the first vector component that decides the outcome of the majority of lexicographical comparisons, fitting the first channel of the vector structuring function becomes more important with respect to the rest. This example is illustrated in Fig. 7, where although only  $V_1 \leq F_1$ , according to the lexicographical ordering,  $\mathbf{V} <_L \mathbf{F}$ . This property would allow for a prioritised detection, where a color template is searched with more emphasis on its brightness than its saturation. A more practical example of VHMT is given in Fig. 8, where the yellow sign of the middle is sought using a lexicographical (pre-)ordering in the LSH color space where saturation (S) is compared after luminance (L) and hue does not participate in comparisons due to its periodicity. More precisely, the SF positioned under the object ( $\mathbf{V}$ ) is formed by decreasing the pixel values of the template by a fixed amount (*e.g.*, 3, if pixel values are in  $[0, 255]$ ), whereas the background SF ( $\mathbf{W}$ ) is formed by increasing it. Hence, the operator looks for all objects that fit between the upper and lower SF based on the lexicographical principle. In this particular case, as the hue is not taken into account, it detects the left sign despite its different hue value, while it misses the right sign, even though its only difference from the template are a few white points; a result that asserts the sensitivity of the operator. Robustness to noise will be specifically addressed in Sec. 5.

**Fig. 7** A two-channel image  $\mathbf{F} = (F_1, F_2)$  and a vector structuring function  $\mathbf{V} = (V_1, V_2)$  [6]



**Fig. 8** In the first row are three images, of which the middle is the sought pattern. The second row shows the locations where it was detected by (Eq. 21) based on a lexicographical ordering of luminance and saturation [6]



### 4.3 HMT Based on Multiple Structuring Elements

In multivariate images, template matching can benefit from user knowledge both on spatial and spectral point of views. The main difficulty faced by template matching operators (such as the morphological hit-or-miss transform) is how to combine these two kinds of information. While the spatial information such as the shape and the size of the sought object can be easily provided by a user, its combination with spectral information is not trivial. Moreover, when dealing with complex patterns, defining a single pair of structuring elements or even functions may be very challenging for the user.

Thus Weber and Lefèvre [39] propose another strategy to design the structuring elements. When seeking for a predefined complex template, the user is then assumed to be able to describe this template by a set of elementary units. Each of these units describes a particular feature of the template, combining some spatial and spectral information. More precisely, it consists of an expected spectral response in some spatial area. To represent such knowledge, each particular feature is defined by an extended structuring element combining spatial properties (shape and size of the area where spectral knowledge is available) provided by the structuring element similarly to existing HMT definitions, and spectral information consisting of an expected

intensity or value in a given spectral band. Thus it can be either lower or higher bounded by a predefined threshold (*i.e.* definition as a background or foreground template), resulting in three spectral properties for each structuring element: the spectral band it is related to, the kind of threshold used (either low or high threshold) and the threshold value.

Contrary to the standard definition of the HMT, a set of extended structuring elements (not necessarily only two) to be involved in the matching process is considered in this approach. While this method deals with spatial information similarly to previous approaches, a particular attention is given to the spectral information. Indeed, contrary to the previous vector definition, each extended structuring element used here is dedicated to a single spectral band. By this way, the user can more easily design the set of structuring elements based on prior knowledge on the sought template. The use of low and high thresholds helps to ensure the robustness of the template matching process, and provides a more practical and realistic way to formulate prior spectral knowledge (compared to previous definitions which are rather contrast-based operators) and may be seen somehow as a generalization of the initial HMT.

For a given extended structuring element  $k$  from the set  $K$ , the spatial pattern (combining shape and size information) is written  $F_k$ , the spectral band  $b_k$ , the threshold or bound  $t_k$  and the related operator (which can be either dilation  $\delta$  or erosion  $\varepsilon$ , corresponding respectively to a high or low threshold, or in other words to foreground or background SE)  $\phi_k$ . The spatial pattern is not expected to be constant over the different image bands. Of course several extended structuring elements may consider the same spectral band, under the assumption that they are consistent together. Conversely, some image bands might be of no interest for a given template matching task and thus not be related to any SE. The fitting step consists of checking, for each analyzed pixel, if its neighbourhood matches the set of extended structuring elements. A pixel will be matched if and only if its neighbourhood fits all the structuring elements, *i.e.* the following condition holds:

$$\text{KHMT}_K(\mathbf{X})(p) \text{ fits iff } \forall k \in K, \begin{cases} \varepsilon_{F_k}(X_{b_k})(p) \geq t_k, & \text{if } \phi_k = \varepsilon \\ \delta_{F_k}(X_{b_k})(p) < t_k, & \text{otherwise} \end{cases} \quad (24)$$

Other fusion options are available to merge the individual fitting results, but the conjunction is of course to be preferred since it ensures that the proposed operator possesses a consistent behavior with common morphological transforms. Similarly to existing definitions, the fitting is followed by a valuation step which aims at giving a resulting value to all matched pixels (the unmatched pixels are set to  $\perp$ ). But contrary to previous works assuming a single pair of foreground/background (or erosion/dilation) structuring elements, here a whole set of extended structuring elements with various properties (shape and size but also spectral band and threshold value, as well as the threshold or operator type) has to be considered. In order to measure how well a pixel (and its neighbourhood) fits a complete set of extended structuring elements, the proposed solution is to first perform a valuation for each individual

structuring element. The quality of each individual fitting procedure is then measured by relying on the erosion or dilation result and the considered threshold, instead of both erosion and dilation as in existing HMT definitions. Thus the difference between the morphologically processed pixel and the threshold is computed:

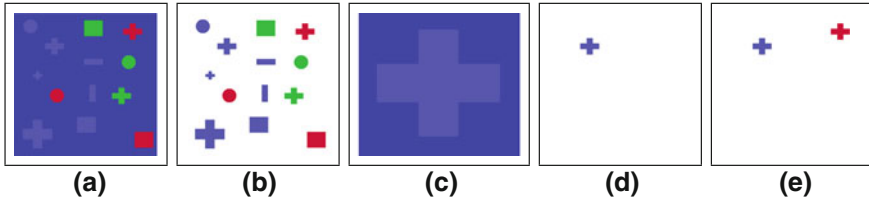
$$\text{KHMT}_k(\mathbf{X})(p) = \begin{cases} \varepsilon_{F_k}(X_{b_k})(p) - t_k, & \text{if } \phi_k = \varepsilon \\ t_k - \delta_{F_k}(X_{b_k})(p), & \text{otherwise} \end{cases} \quad (25)$$

which ensures a strictly positive result for each fit pixel. However, no assumption can be made that in practice multivariate images will always contain comparable spectral bands. In other words, the different spectral components of a multivariate image may not share the same value ranges. Thus, a normalization step is further introduced, resulting in a new definition for the individual valuation steps:

$$\begin{aligned} \text{KHMT}_k(\mathbf{X})(p) &= \begin{cases} (\varepsilon_{F_k}(X_{b_k})(p) - t_k) / (X_{b_k}^+ - t_k), & \text{if } \phi_k = \varepsilon \\ (t_k - \delta_{F_k}(X_{b_k})(p)) / (t_k - X_{b_k}^-), & \text{otherwise} \end{cases} \\ &= \begin{cases} (\varepsilon_{F_k}(X_{b_k})(p) - t_k) / (X_{b_k}^+ - t_k), & \text{if } \phi_k = \varepsilon \\ (\delta_{F_k}(X_{b_k})(p) - t_k) / (X_{b_k}^- - t_k), & \text{otherwise} \end{cases} \end{aligned} \quad (26)$$

where  $[X_i^-, X_i^+]$  is the predefined value range of the spectral band  $X_i$  (and of course the assumptions  $t_k \neq X_{b_k}^-$  and  $t_k \neq X_{b_k}^+$ ). The normalization is achieved by  $(X_{b_k}^+ - t_k)$  or  $(X_{b_k}^- - t_k)$  in order to obtain a valuation in  $[0, 1]$ . Once the individual valuations have been computed, it is then necessary to assign a unique value to each matched pixel. In addition, there is no unique valuation scheme for this multivariate HMT. Indeed, one can even keep the set of individual valuations as the final result if interested in the quality of the fit for each individual pattern. But usually, a single scalar value is “expected” and it is here built with a fusion rule. To ensure coherence with the previous fitting step (where a conjunction rule has been used to merge individual fitting results), the method relies on a T-norm, *e.g.*, either the product or the minimum, leading respectively to the following definitions:

$$\begin{aligned} \text{KHMT}_K^{\text{prod}}(\mathbf{X})(p) &= \begin{cases} \prod_{k \in K} (\text{KHMT}_k(X)(p)) & \text{if } \forall k \in K, \text{KHMT}_k(X)(p) > 0 \\ 0 & \text{otherwise} \end{cases} \\ &= \prod_{k \in K} (\max(\text{KHMT}_k(X)(p), 0)) \end{aligned} \quad (27)$$



**Fig. 9** **a** Original image; **b** Image without background; **c** Template to detect (magnified); **d** KHMT after reconstruction; **e** VHMT with lexicographical ordering after reconstruction [39]

$$\begin{aligned}
 \text{KHMT}_K^{\min}(\mathbf{X})(p) &= \begin{cases} \min_{k \in K}(\text{KHMT}_k(X)(p)) & \text{if } \forall k \in K, \text{KHMT}_k(X)(p) > 0 \\ 0 & \text{otherwise} \end{cases} \\
 &= \min_{k \in K}(\max(\text{KHMT}_k(X)(p), 0)) \tag{28}
 \end{aligned}$$

Figure 9 illustrates the relevance of this approach in the context of an RGB image containing objects of various shapes (crosses, rectangles, circles and squares) and different colors. The sought template is a cross with a color close to the background color, leading here to three structuring elements (related to the three color bands) to be used by the HMT. For the sake of comparison, the result with the vector HMT (based on a lexicographical ordering) is also given.

### 4.4 HMT Based on Supervised Ordering

Velasco-Forero and Angulo [37, 38] have developed an alternative approach for applying the HMT on multivariate images, using a supervised ordering which is not a total ordering (contrary to lexicographical ordering discussed previously).

As far as the lack of ordering from color vectors is concerned, they rely on the principle of reduced orderings. According to this technique, given a non-empty set  $R$  which lacks a complete lattice structure, one can impose a such structure by means of a mapping  $h : R \rightarrow T$  where  $T$  is complete lattice, which leads to an  $h$ -ordering  $\leq_h$  [14]:

$$\forall r, r' \in R, \quad r \leq_h r' \Leftrightarrow h(r) \leq h(r') \tag{29}$$

An  $h$ -supervised ordering on the other hand is based on subsets  $B, F \subset R$ , such that  $B \cap F = \emptyset$  and is defined as an  $h$ -ordering that satisfies  $h(\mathbf{b}) = \perp$  if  $\mathbf{b} \in B$  and  $h(\mathbf{f}) = \top$  if  $\mathbf{f} \in F$ ; where  $\perp$  and  $\top$  represent respectively the minimum and maximum of  $T$ . In which case the resulting  $h$ -supervised ordering is denoted by  $h_{\{B, F\}}$ . Given the lack of extremal coordinates within the vector space containing the color vectors, such as black and white with grayscale values, the use of arbitrarily selected  $B$  and  $F$  sets enables the construction of an ordering based on custom extremal coordinates. Ref. [37] relies on Support Vector Machines in this regard, in

order to construct a hyperplane separating the vectors that emanate from  $B$  and  $F$ . Thus the vectors are ordered w.r.t. their distance to the maximum margin hyperplane.

Consequently, they redefine the binary HMT using their h-supervised ordering, with  $\{1 = \top, 0 = \perp\}$  and the image complement obtained by exchanging the subsets  $B$  and  $F$ :

$$\text{HMT}(\mathbf{F}; \{B, F\}, S_1, S_2) = \{x \in E \mid \forall i, \varepsilon_{h_i; S_i}(\mathbf{F}(x)) = \top_i\} \quad (30)$$

where

$$h_i = \begin{cases} h_{\{B, F\}} | h(b) = \perp, h(f) = \top & \text{if } i = 1; \\ h_{\{F, B\}} | h(f) = \perp, h(b) = \top & \text{if } i = 2; \end{cases} \quad (31)$$

and  $\varepsilon_{h_i; S_i}$  denotes the erosion operator with SE  $S_i$  and based on the h-supervised ordering  $h_i$ .

At which stage they extend the binary HMT of (Eq. 30) to the multivariate case, by associating each vector value set within the sought template SE with a particular  $B_i \subset \mathbb{R}^n$  and using sets of  $\{B_i, S_i\}_{i=1, \dots, k}$  couples such that  $\forall i \neq j, S_i \in E, S_i \cap S_j = \emptyset$ :

$$\text{HMT}(\mathbf{F}; \{B_i, S_i\}) = \{x \in E \mid \forall i, \varepsilon_{\{B_i, B_{-i}\}; S_i}(\mathbf{F}(x)) = \top_i\} \quad (32)$$

where  $B_{-i} = \bigcup_{j \neq i} B_j$ ,  $\{S_i\}_{i=1, \dots, k}$  is the family of structuring elements and  $\{B_i\}_{i=1, \dots, k}$  is the family of vector pixel values associated with  $\{S_i\}_{i=1, \dots, k}$ .

Finally, in order to achieve robustness against noise the authors have also implemented a more practical version of (Eq. 32) by using a threshold  $\epsilon$  controlling the “level” of detection:

$$\text{HMT}_\epsilon(\mathbf{F}; \{B_i, S_i\}) = \{x \in E \mid \forall i, \text{dist}(\varepsilon_{\{B_i, B_{-i}\}; S_i}(\mathbf{F}(x)), \top_i) \leq \epsilon\} \quad (33)$$

with  $\text{dist}$  being an arbitrary metric.

#### 4.5 HMT Based on Perceptual Color Distance

A further approach focusing on color template matching by means of morphological methods has been recently introduced by Ledoux et al. [19]. The authors base their work on the gray HMT formulation of Barat et al. [7] and propose an extension designed specifically for color images.

In particular, in order to overcome the lack of ordering among color vectors, they propose to use two arbitrary reference color coordinates A and B within the CIELAB color space. In which case given a finite set of color vectors, they suggest computing their maximum as the closest color vector to reference A and the minimum



as the closest to reference B. And as far as the metric in place is concerned, they employ the perceptual distance of CIELAB. Consequently, they implement erosion and dilation based on the aforementioned extrema. Thus they employ two distinct pre-orderings; since two unequal color vectors may very well possess the same distance to a reference color, they can both end up being considered equivalent; meaning that distance based ordering approaches of this type lack anti-symmetry.

This theoretical inconvenience however has not hindered the authors from using their color dilation and erosion formulations with template matching purposes. Specifically, they have introduced the CMOMP (Color Multiple Object Matching using Probing) [19] tool, which given two structuring functions  $g'$  and  $g''$  representing the lower and higher bounds of the sought template respectively, they associate each pixel  $x$  of the color input image  $\mathbf{F}$  with the perceptual distance ( $\Delta E$ ) computed at CIELAB between the dilation and erosion outputs at the same coordinate.

$$\text{CMOMP}_{g',g''}(\mathbf{F})(x) = \Delta E(\delta_{g''}(\mathbf{F}), \varepsilon_{g'}(\mathbf{F}))(x) \quad (34)$$

All in all, the CMOMP adopts a similar approach to that in Ref. [37] by employing user specified extremal coordinates for the multi-dimensional color space, with the additional advantage of perceptual color different computation.

In conclusion, although all four color/multivariate HMT approaches presented in this section possess a variety of properties, none of them stands out as the “ultimate” solution to the problem of morphological color template matching; since such an approach would have to be both effective, robust against noise, computation-wise efficient, easy to use/customize and of course possess as many forms of invariance (w.r.t. rotation, scale, illumination, etc.) as possible. All the same, all of the presented approaches have appeared in the past few years, a fact highlighting both the attention that this topic enjoys at the moment as well as asserting future developments. And now, let us focus on the implementation issues related to the HMT.

## 5 Implementation of Template Matching Solutions Based on HMT

In the previous sections, we have given a theoretical presentation of the HMT, for binary, grayscale, and color or multivariate images. In order for these definitions to be of practical interest in template matching, several implementation issues have to be considered. We review here the main problems faced by the HMT when applied to template matching. Equipped with the solutions reviewed in this section, the HMT is then able to deal with real template matching use cases, which will be presented next.

## 5.1 Robustness to Image Transforms

When a template matching operator is applied in a real context, it is far from an artificial situation and easy to solve problem such as those given in the figures of previous sections. Indeed, it has often to face a great variation both of the template to be matched and its environment. These variations mainly occur due to global, or worse local, image transforms.

Greyscale or spectral image transforms lead to (linear or not) modifications of the template / image pixel values. Most of the HMT definitions reviewed so far are contrast-based operators, *i.e.* they are robust to a global change in pixel values. For methods which are not (*e.g.*, [39]), it is possible to normalize the input image in order to avoid brightness changes. Similarly, in case of varying image contrast, usual techniques in image processing such as histogram equalization or specification become necessary. Since these image dynamics may have different properties over an image (*i.e.* have an effect different in some parts of the image w.r.t. the others), spatially-variant morphological operators are certainly worth being exploited in this context.

Spatial image transforms are various. We distinguish here between translation and scale/orientation. Translation is inherently addressed by Mathematical Morphology which provides translation invariant operators, and thus does not need further discussion. Of course this assumes that the HMT is applied on every single pixel of the image.

The easiest solution to ensure orientation or scale robustness is to apply the template in various configurations corresponding to the possible orientations and scales. This means applying the HMT with a complete set of SE in various orientations and sizes related to the considered configurations. The fitting and valuation steps may be defined as follows: a pixel is fit as soon as it is kept by the HMT with at least one SE / configuration. The value assigned to it by the HMT is then computed as the highest one provided by the valuation step for all configurations that make the pixel fit.

Unfortunately, ensuring invariance to scale and rotation by applying HMT with a complete set of SE representing the various configurations is not efficient. Indeed, in the case of  $N$  possible configurations (or SE), HMT-based template matching will require  $N$  times the processing time needed for a single SE. This is a strong bottleneck, which may be overcome by using several optimizations to be discussed later in this section. Moreover, in case of larger SE to deal with closer range images, the computation time will greatly increase. In this case, another solution consists in subsampling / interpolating the image instead of the template, before applying the HMT. Finally, particular attention should be given to the new templates generated from the input one, to avoid unexpected effects brought by discretization for instance.

## 5.2 Dealing with Noise and Uncertainties

Furthermore, as the HMT attempts to perform an exact match of the given pattern, it becomes sensitive to noise, occlusions, and even to the slightest variations of the template shape. Consequently a series of approaches have been imagined, with the purpose of countering this drawback and increasing its practical interest.

Bloomberg and Maragos [9] proposed several solutions to improve robustness to noise and shape uncertainties. The first idea is to perform a subsampling of both image and SEs. This solution is attractive as it reduces small shape variations and saves computation time. Nevertheless it requires the resolution to be good enough so the subsampling will preserve major features of the shape. Another possibility is to perform a spatial decimation of the support of structural functions following a regular grid. The advantages of this method are the same as subsampling's ones, but it can be applied even at low resolution.

It is also possible to improve noise resistance of the HMT by first dilating both foreground and background before performing erosions, as proposed in the binary case by Bloomberg and Maragos [9]. In this case the foreground and the background overlap, thus it becomes easier to fit the SEs. However the extension to the greyscale and color cases is not trivial since one has to define the structuring functions to be used.

Another way to improve robustness against noise and shape uncertainties is to provide less informative SEs (i.e. increase the distance between the foreground and the background). This is nearly equivalent to previous method and can be the easiest way to work with low level signal-to-noise ratio. Nevertheless this method fails at very low SNR as the SEs become so blurred that almost every configurations can provide a good match.

A simple solution to improve the tolerance to impulsive noise of all methods is to use rank operators [31, 35] instead of traditional erosion and dilation [9, 18] (i.e. replace min and max by quantiles). This solution is described in detailed in Sec. 5.2.1 in the case of VHMT. However this solution implies to determine an appropriate rank for both dilation and erosion operations. Khosravi and Schafer [18] have shown that a lower and upper bounds for the rank are given respectively by impulsive and Gaussian noise, but they did not provide a general formula to obtain these bounds. They also concluded that the effect of rank operator on Gaussian or Poisson noise is limited. Later, Murray et al. [21] have proposed a general approach to automatically determine an optimal rank in the context of the HMT without any assumption on the noise distribution. In case of very noisy image, the Perret et al. fuzzy HMT [27] allows to recover good detection results.

Finally, some authors [7, 13] proposed to *learn* the foreground and background structuring elements from examples. This will be addressed in Sec. 5.4 but is also a way to incorporate the observed uncertainties in the definition of the templates. This latter approach called *synthetic* SE is described in Sec. ??.

### 5.2.1 Practical Solutions in Color Images

While the methods proposed so far to deal with noise and uncertainties are mainly related to the general case of binary or greyscale images, we recall here the results from [6], namely rank order based VHMT and synthetic multivariate structuring functions.

#### Rank order based VHMT

A rank order filter of  $k^{th}$  rank is a transformation, where given a gray-level image  $F$  and a SE  $B$ , the result becomes:

$$(F \square_k B)(p) = k^{th} \text{largest of } F(p+x), x \in B \tag{35}$$

with  $k \in \{1, \dots, |B|\}$ . Obviously,  $F \square_1 \check{B} = \delta_B(F)$  and  $F \square_{|B|} B = \varepsilon_B(F)$ . Moreover, always in the context of gray-level data, a rank order filter of  $k^{th}$  rank, is equivalent to the supremum of erosions using all possible SE with  $k$  points, and respectively to the infimum of dilations using all possible SE with  $|B| - k + 1$  points [36]. Due to this property, the binary HMT of (Eq. 3) has been reformulated in the literature, by replacing the erosion in its expression with a rank order filter of rank  $k < |B|$ , hence making it possible to detect binary templates even in conditions of partial occlusion.

In order to achieve the same additional robustness in the case of a multivariate image  $\mathbf{F}$  along with a multivariate SF  $\mathbf{B}$  and a vector ordering  $\leq_v$ , one can redefine the rank order filter of  $k^{th}$  rank as follows:

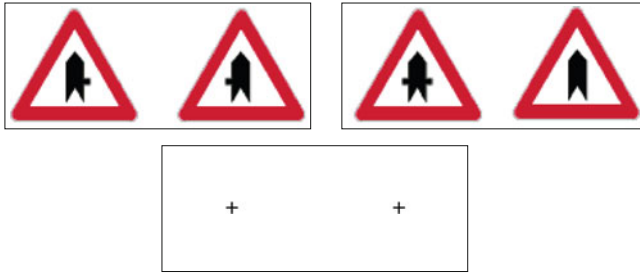
$$\zeta_{\mathbf{B}}^k(\mathbf{F})(p) = k^{th} \text{ largest of } \mathbf{F}(p+x) - \mathbf{B}(x), x \in \text{supp}(\mathbf{B}) \tag{36}$$

$$\theta_{\mathbf{B}}^k(\mathbf{F})(p) = k^{th} \text{ largest of } \mathbf{F}(p-x) + \mathbf{B}(x), x \in \text{supp}(\mathbf{B}) \tag{37}$$

where  $k \in \{1, \dots, |\text{supp}(\mathbf{B})|\}$ . Naturally, the vectors are sorted using  $\leq_v$ . Thus,  $\varepsilon_{\mathbf{B}} = \zeta_{\mathbf{B}}^{|\text{supp}(\mathbf{B})|}$  and  $\delta_{\mathbf{B}} = \theta_{\mathbf{B}}^1$ . Consequently, one can now formulate an approximative VHMT, capable of detecting the sought template  $(\mathbf{V}, \mathbf{W})$  even if  $m$  and  $n$  pixels do not match respectively the foreground and the background:

**Fig. 10** On the left is a couple of images, of which the leftmost is the sought pattern, and on the right is the output of  $\eta_{[\mathbf{V}, \mathbf{W}], 750, 750}^f$  (Eq. 38) [6]





**Fig. 11** In the first row, from left to right the two images to detect, and the corresponding lower and upper synthetic SF computed through (Eqs. 39 and 40); the second row contains the result of VHMT, (Eq. 21) [6]

$$\eta^l_{[\mathbf{V}, \mathbf{W}], m, n}(\mathbf{F})(p) = \begin{cases} \|\zeta^m_{\mathbf{V}}(\mathbf{F})(p) - \theta^n_{\mathbf{W}^*}(\mathbf{F})(p)\| & \text{if } \zeta^m_{\mathbf{V}}(\mathbf{F})(p) >_v \theta^n_{\mathbf{W}^*}(\mathbf{F})(p) \\ 0 & \text{otherwise.} \end{cases} \tag{38}$$

where  $m \in \{1, \dots, |\text{supp}(\mathbf{V})|\}$  and  $n \in \{1, \dots, |\text{supp}(\mathbf{W})|\}$ . It should also be noted that  $\eta^l_{[\mathbf{V}, \mathbf{W}], |\text{supp}(\mathbf{W})|, 1}(\mathbf{F}) = \eta^l(K_{\mathbf{V}, \mathbf{W}}(\mathbf{F}))$ . Figure 10 contains an example of the result given by this operator, where the leftmost image is sought under the same conditions as in Fig. 8. However, this time even though the right example has a red/brown stripe, it is still successfully detected. This is due to the use of the 750<sup>th</sup> rank, a number equal to the amount of different pixels between the two images. Thus the rank based operator can allow a flexibility margin large enough to realise the detection in case of pixel value variations, due to reasons such as noise.

### Synthetic Multivariate Structuring Functions

Although multivariate rank order filters make it possible to detect partial matches, (Eq. 38) still hardly satisfies practical needs, since the objects corresponding to the sought template may vary considerably. Consider for instance the case illustrated in Fig. 11 (top-left). This situation is of course present in the context of detection from gray-level images as well. One way of countering it, as explained in [7, 13], is to employ a set of example images, from which a common template is formed, or as defined in Ref. [13], “synthetic”.

More precisely, the foreground is represented by the minimum and the background by the maximum of the given set of examples ( $\{\mathbf{V}_i\}, \{\mathbf{W}_j\}$ ). Thus, in the multivariate case the same technique may be employed merely by using the chosen vector ordering  $\leq_v$ :

$$\mathbf{V}(x) = \inf_i \{\mathbf{V}_i(x)\} \tag{39}$$

$$\mathbf{W}(x) = \sup_j \{\mathbf{W}_j(x)\} \tag{40}$$

Returning to Fig. 11, it suffices to compute the templates corresponding to the images at the top-left by means of this operation, and the VHMT of (Eq. 21) detects both successfully.

### 5.3 Performance

Relying on a naive implementation of the previous HMT definitions will lead to a template matching process of poor performance (*i.e.* very long processing time). We describe here various methods which are available to reduce the computation time of the HMT when used in template matching.

#### SE/SF decimation

One simple way to speed-up the HMT is to use decimated structuring elements or functions [10]. The idea here is to subsample in a regular manner the structuring elements/functions. Indeed, the HMT precision is robust to such subsampling while this greatly reduces the computation cost as the computation cost is directly linked to the number of SE/SF pixels.

#### Hard fitting

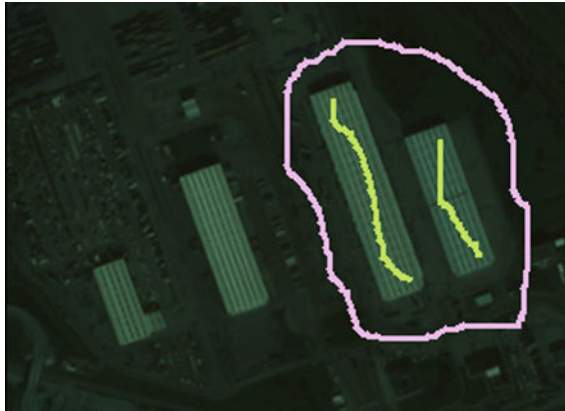
Binary HMT but also grayscale/color HMTs that rely on a hard fitting step can easily be speeded-up by taking advantage of this strong constraint. Such HMTs lead to an efficient implementation of the fitting step (which is in fact the most important part of HMT based template matching). In the case the fitting process requires a pixel to fit all the structuring elements/functions to be kept, it is relevant to stop the process as soon as one of the SE/SF pixel is not matched by the HMT. Only an incomplete processing of the set of SEs/SFs is then performed, thus greatly reducing the computing time.

### 5.4 Parameters Settings

The settings of the SE parameters may sometimes be tricky. In a given domain, it is expected to be made by an expert based on his domain knowledge. We consider here the context of remote sensing, for which several template matching attempts have been made with the HMT operator. In this context, and more precisely for coastline extraction, the expert knows (at least partially) the spectral signature of the desired object and how it can be distinguished from its environment. This spectral information depends on the type of sensor and eventually external factors (*e.g.*, season for remote sensing imagery). To avoid computational inefficiency, it is better not

**Table 1** Spectral ranges from sea and land obtained from a QuickBird image [39]

Band	Sea	Land
Blue	173–193	134–165
Green	215–269	147–241
Red	119–167	66–177
Near Infrared	55–84	126–575

**Fig. 12** Scribbles drawn on some warehouses and background to learn SE parameters (©Digital-globe) [39]

to use all possible constraints, but to rather consider only the most discriminant spectral information of the object under study when compared to its environment. For instance, if we consider spectral knowledge given in Table 1, we can observe an important overlap of spectral signatures of sea and land in bands *Green* and *Red*. Thus, the expert will most probably not use these spectral bands as discriminative information in the template matching process. Setting parameters w.r.t. bands *Blue* and *Near-Infrared* looks much more relevant since the related information is more reliable to distinguish between sea and land.

To ease the SE parameter setting step, intuitive and interactive approaches may be considered, *e.g.*, drawing scribbles on objects of interest [12, 40]. It then allows to learn the spectral range of the desired features and to automatically set the SE parameters in order to achieve the highest discriminative power of the template matching method. This principle is illustrated in Fig. 12 where the user has drawn marker on two warehouses and on the background: from this user input, the system is able to obtain the spectral range of the templates defined from the scribbles (Table 2). This can then help to feed a subsequent HMT template matching solution, such as the knowledge-based multivariate HMT [39]. In this case, either the user analyzes the provided spectral ranges to select appropriate bands and thresholds, or a machine learning algorithm is involved to identify the best decision functions.

**Table 2** Spectral range of the scribbles drawn by the user on figure 12 [39].

Band	Warehouses	Background
Blue	336 – 556	142 – 323
Green	511 – 857	161 – 476
Red	381 – 635	82 – 344
Near Infrared	362 – 612	93 – 431

**Fig. 13** Example of SEs for an oriented discontinuity extraction [39]



Besides the learning or definition of the spectral information of a multivariate/color template, the shape has also to be set and once again, interactive methods may be useful to get some rough ideas about the template sought by the user.

### 5.5 From HMT to Template Matching

We have already addressed a set of issues raised when considering HMT on real images for template matching. To be successfully applied, morphological template matching also requires to be adapted to the specificities of the desired pattern. We can distinguish between two types of patterns which will now be discussed.

#### 5.5.1 Discontinuity

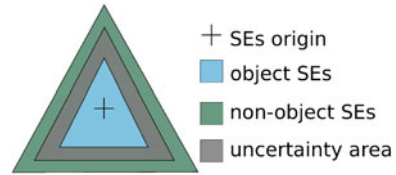
Discontinuity is somehow an abstract feature: indeed, it is not considered as a straight visual feature but rather denotes the limit between two specific areas. Discontinuity extraction may be observed in general cases (*e.g.*, edge detection) but also in very specific ones (*e.g.*, coastline delineation). Here a strong assumption is made about the existence of two opposite SEs, each of them defining one given area. HMT based template matching dedicated to particular discontinuity extraction is usually built in two steps:

- The two areas have to be fully defined (with spatial and color/spectral information), but the spatial definition may be limited to the depth or width of the feature only.
- Apart for the detection of a specific directional discontinuity (*e.g.*, only vertical discontinuity), it is necessary to apply the HMT with different SE orientations.

We notice that nothing prevents the two SEs to be unconnected. In this favorable case, the HMT operator is able to handle an uncertainty area.



**Fig. 14** Example of SEs for object extraction [39]



### 5.5.2 Object

Object may be seen as the classical feature of template matching. Relying on any HMT operator, specific object extraction is achieved using two SEs: one is defining the object (being possibly as small as a single point) while the other is defining what is not the object (*e.g.*, the background, the object surrounding, etc.). The following steps are here necessary:

- The SE representing the object is fully defined, as well as the SE dedicated to background
- Depending of the object and its properties (fixed size or not, fixed orientation or not, etc.), the HMT has to be applied with SEs at various orientations and/or scales.

In order to be able to match the object even with discretization artefacts (*e.g.*, stairing effect), an uncertainty area located between the two SEs might be necessary.

## 6 Applications

The relevance of the HMT as a solution for template matching is illustrated by several practical applications to help the reader understand the benefit of this operator when dealing with real-life problems. In this section, we present in particular some examples related to the fields of astronomical imaging [27], earth observation [20, 37, 39], document processing [41] and medical imagery [24]. Only a part of these examples are related to color images, since color HMT has been addressed only very recently. Nevertheless, nothing prevents the extension of the presented use cases to color data.

### 6.1 Astronomy: Low Surface Brightness Galaxy Detection

Galaxies come in various shapes but have long been expected to all have the same surface brightness. It's only recently that low surface brightness (LSB) galaxies (Fig. 15) have been discovered which leads to the problem of the automatic detection of these objects among the huge astronomical image datasets.

In this subsection we describe a method based on PHMT (Eq. 16) to automatically build a segmentation map of potential LSB galaxies. This method has been presented in more details in [27]. The algorithm is decomposed into several steps. First it is

necessary to build several patterns corresponding to LSB galaxies of various shapes and orientations. Because the sought objects are very close to the background in terms of photometry, a precise map of the background has also to be computed (*i.e.* evaluate the intrinsic luminosity of the sky at all points.) Next the original image is preprocessed with a median filter to reduce noise. Filtered image, background map, and pattern set are then used to calculate PHMT. The result is thresholded and the original shape of LSB galaxies is reconstructed.

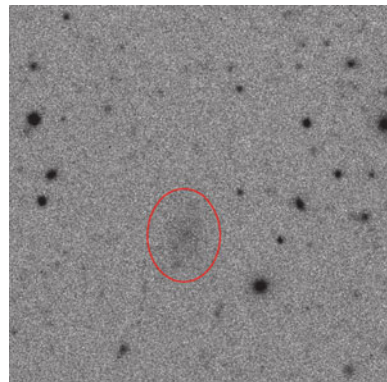
### Description of Patterns

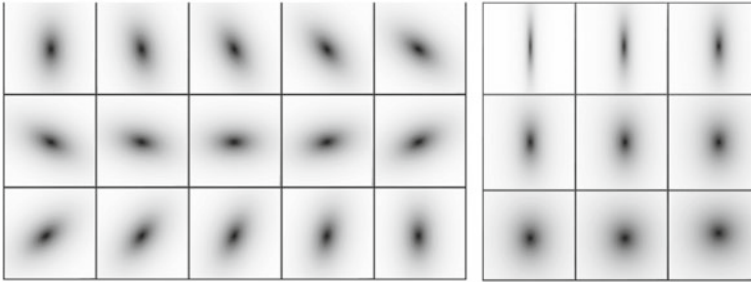
LSB galaxies can be modeled by a quite simple function that can be used to generate patterns of LSB galaxies. This model involves four parameters: the central brightness, the elongation, the orientation and the scale length. 10 possible scale-lengths and 14 orientations between 0 and  $\pi$  are considered. The final set of SEs obtained by combining all possible scale lengths, elongations and orientations while avoiding identical cases induced by symmetries, is composed of 640 templates (Fig. 16). Finally each pattern is composed of two SEs of same orientation and elongation.

### Computation of PHMT

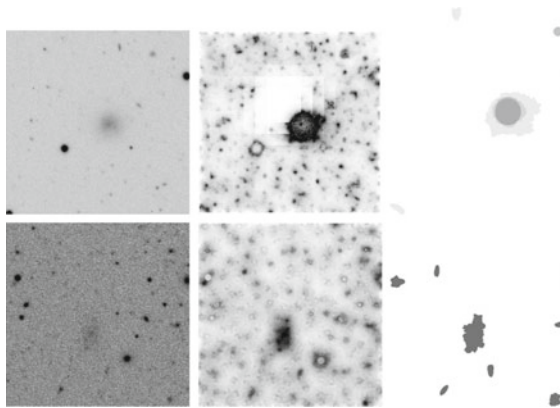
The PHMT of each pair of structuring elements is computed for each pixel and the best matching score is stored in a so called *score map* (Fig. 17). In the next step, the score map is thresholded. As all parameters of the algorithm are set automatically according to observation parameters and statistics, the score gives an absolute measure which is independent of the observation and the use of the same threshold for every observation is not a problem. Finally, each pixel of the binary map is dilated by the support of the pattern that gives the maximum score in this position (Fig. 17).

**Fig. 15** Example of low surface brightness galaxy (inside the ellipse) [27]





**Fig. 16** Example of SEs obtained with a variation of orientations (left) and elongation (right) [27]



**Fig. 17** Application of the PHMT to LSB galaxy detection [27]. First column: original image. Second column: score map obtained after application of the PHMT (the blocky aspect comes from the background and noise estimation procedure that is done on a subgrid of the original image). Third column: thresholded and reconstructed map. The final map contains different classes proportional to the object brightness. This allows deblending capabilities for overlapping objects having different brightnesses

## Experiments

The robustness of the method has been assessed by extensive testing on simulated and real datasets. The evaluation obtained on more than a thousand simulated images have shown that the method is able to detect LSB galaxies down to a peak signal-to-noise ratio of 0 dB even in relatively crowded environment (inducing overlapping). These good results have been reproduced on two real images ( $2048 \times 4096$  pixels) where the algorithm proposed 23 candidates, among which 6 were already known LSB galaxies, 8 were new LSB galaxies and the others were false positives.

## 6.2 Earth Observation: Natural and Human-Made Object Extraction

### 6.2.1 Coastline Extraction

As a first application example in earth observation through remote sensing, we consider the case of satellite images with a very high spatial resolution (VHR). Template matching on such images can focus on some predefined borders, coastline being one of the most representative examples when dealing with coastal remote sensing.

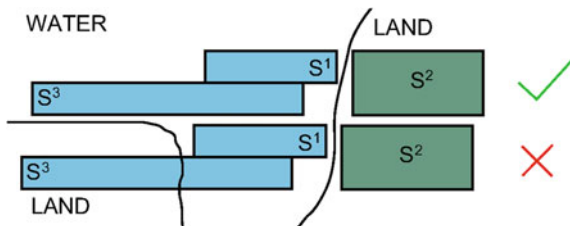
The sought template considered here is rather particular since it is indeed a discontinuity, as coastline is typically defined as the border between sea and land. Its automatic extraction from digital image processing is a very topical issue in remote sensing imagery [11]. Even if some methods have been proposed for low or medium spatial resolution, none are relevant on very high spatial resolution (VHR) satellite imagery where a pixel represents an area lower than  $5 \times 5 \text{ m}^2$ .

Since a coastline is well defined both with spatial and spectral information, the KHMT can be seen as a relevant tool to perform its extraction. A basic assumption would be that only two SEs are necessary to identify the two parts around the border (*i.e.* sea and land) as shown by  $S^1$  and  $S^2$  in Fig. 18. However, to be able to distinguish between coastlines and other water-land borders (*e.g.*, lake border, river bank, etc), an additional SE is involved to represent deep sea (or at least water further from the coastline) as illustrated by  $S^3$  in Fig. 18.

The relevance of KHMT for the extraction of coastline in VHR imagery can be observed in Fig. 19. The interested reader will find more details in [28, 39].

### 6.2.2 Tank Extraction

In [39], another application in earth observation is presented to deal with petroleum tank extraction. On a study site located in the harbor of Le Havre in France, such petroleum tanks are cylindric (like other tanks) but are also white (which make them distinguishable from other tanks). In this context, spatial information needs to be



**Fig. 18** Spatial definition of SEs used for coastline extraction with matching and unmatching conditions [39]

combined with spectral information to ensure an accurate detection of these objects. KHMT then appears as a relevant solution, using two sets of SEs: one defining the petroleum tanks, and the other defining their neighbourhood. Their shapes are shown in Fig. 20.

Since there is not a unique size for petroleum tanks, KHMT has to be applied with various SE sizes. The spectral parameters are defined by an expert and given in Fig. 20. The spatial parameters are the following: the circular SE is of radius  $r$  varying from 5 to 20 pixels, while the surrounding ring SE radius is a little bit larger (*i.e.* equal to  $r + 2$ ) and ring width is 1 pixel.

Shape *circle* represents object set of SEs while *ring* represents non-object set of SEs. NIR means Near Infra-Red band. In this example, thresholds have been easily set by the expert from a manual study of the multispectral image histogram. Indeed, tanks might be distinguished from the other objects and the image background based on their spectral signature. Peak values in the histogram are used to set adequate threshold values.

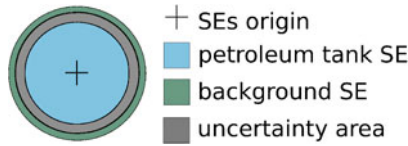
Applying KHMT with these parameters gives promising results as shown in Fig. 21. This satellite image contains 33 petroleum tanks, 32 were extracted, one was missed due to the presence of dark colors on its roof and there is no false positive. KHMT appears here as an efficient solution to deal with the problem of specific tank extraction. Let us note that the problem of the missed tank can be further solved by introducing some robustness in the KHMT operator, as it has been discussed previously with some other HMT definitions.

### 6.2.3 Building Extraction

The last example in the field of earth observation focuses on automatic building extraction, which is helpful to optimize the management of urban space by local politics. The overall approach presented here [20] is composed of three main steps: (1) perform binarization/clustering of the input grayscale image (here a panchromatic



**Fig. 19** Coastline extraction on Normandy coast from a QuickBird image (©Digitalglobe), extracted coastline in green and reference coastline in blue (results are dilated for better visualization) [39]



SE	spatial pattern ( $F$ )	band ( $b$ )	threshold ( $t$ )	related operator ( $\varphi$ )
1	circle	BLUE	450	$\varepsilon$
2	circle	GREEN	450	$\varepsilon$
3	circle	RED	450	$\varepsilon$
4	circle	NIR	450	$\varepsilon$
5	ring	GREEN	716	$\delta$

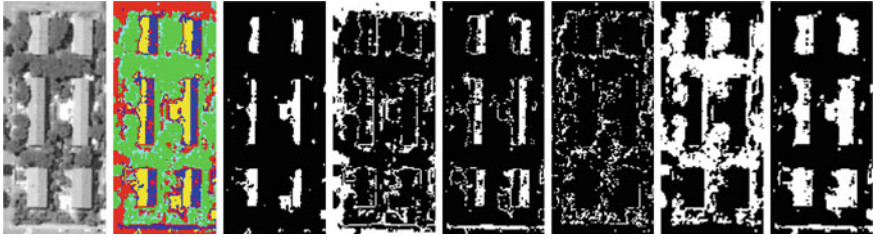
**Fig. 20** Spatial definition of SEs used for petroleum tank extraction and corresponding SEs definition [39]. Pixel values are encoded with 11 bits per band (range is [0; 2047]) and extracted from Quickbird sensor



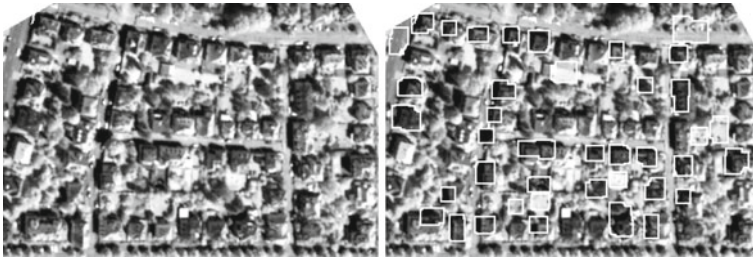
**Fig. 21** Petroleum tank extraction on Le Havre harbor QuickBird image(©Digitalglobe) [39]. Correct detections are surrounded by white boundary, false negative are given in cyan (there is no false positive)

Quickbird image) to obtain a map of relevant features and make the data compatible with a binary HMT operator; (2) prefilter the binary images using morphological filters for which parameters (SE properties) are obtained automatically from morphological image analysis step; (3) proceed to building extraction using an adaptive HMT.

This approach assumes that building roofs are of rectangular shape, and made of homogeneous content (or at least of several homogeneous parts). In this context, it is expected that binary images produced from the two first steps either contain entire building roofs, or significant parts of them, which are further recombined with a “combination and fusion of clusters” step. This last case can be explained by the variations brought by different roof materials and sunlight illuminations, as illustrated



**Fig. 22** Processing with the HMT a clustering result as a set of binary images [20]: (from left to right) input image, clustering, binary images corresponding to the 5 clusters, and the binary image obtained by merging clusters 1 and 3



**Fig. 23** Illustration of building extraction [20]: input image (left), and final result (right)

by Fig. 22 where we can observe the relevance of combining various clusters to build meaningful binary images to be processed by the HMT. The HMT is then adapted to various sizes (width and length of the rectangular SE, see previous section) to deal with the different configurations of rectangular-shaped buildings observed from remote sensing imagery. Similarly to previous approach, the last step consists of a (geodesic) reconstruction to obtain the identified buildings from the fitted pixels.

An illustration of this building extraction scheme is given in Fig. 23. We can observe the relevance of the HMT operator w.r.t assumptions made (building roof is made of a single homogeneous part or of an heterogeneous set of different homogeneous roof parts), as well as the limitations of such assumptions (in case of highly heterogeneous roofs).

#### 6.2.4 Ship Detection

Velasco-Forero and Angulo use their color HMT based on supervised ordering to detect ships in color satellite image [37]. The input RGB image is taken from the WorldView-2 satellite, with a spatial resolution of 50cm per pixel. Ships are made of grayish pixels on a blue background representing the sea. In this context, learning an ordering to distinguish between blue and grey pixels is relevant, and help the method to reach high detection rates.

### 6.3 Document : *Symbol Spotting in Architectural Plan*

Symbol spotting is an emerging topic and few works haven been proposed so far. Although well-known symbol descriptors work well to describe isolated symbols, their performance in real applications drop away when symbols are embedded in documents. In the context of monosize symbol (*e.g.*, architectural plan), HMT-based methods are particularly adapted to symbol spotting and we recall here results presented in [41].

The major symbol spotting difficulty when dealing with plan images is the overlapping/occlusion of symbols which are embedded into the document and make harder their spotting. To achieve the robustness to information overlapping, Weber and Tabbone do not rely on a fitting step but rather provide for every pixel two different matching scores. These scores are respectively dedicated to the foreground and the background, assuming a perfect match with the foreground while being less strict with the background due to information overlap. More precisely, the foreground (*resp.* background) matching score is defined as the mean of foreground (*resp.* background) pixel matching scores. This foreground (*resp.* background) pixel matching score is set to one if the pixel value is greater or equal (*resp.* lower or equal) to the corresponding foreground (*resp.* background) pixel, and to a value in  $[0; 1]$  otherwise (to ensure robustness to small value differences). As with other applications, geometric robustness (*w.r.t.* orientation, symmetry) is achieved through the application of the HMT in various configurations.

Figure 24 illustrates the spotting efficiency of this approach in complex overlapping situations. We can observe that: (1) HMT is able to spot a symbol even if it is connected to another symbol; (2) HMT is able to extract a symbol even if it is highly overlapped by other information; and (3) HMT comes with a high discrimination power since it can spot the queried symbol and not the other but very similar symbols.

### 6.4 Medical: *Feature Extraction for Pathology Detection*

#### 6.4.1 Vessel Segmentation

The spread of medical images from different types (*e.g.* Magnetic Resonance Imaging (MRI), Computed Tomography (CT), etc.) had led to the need of segmentation techniques adapted to the particular content of these data. In this context, vessels have been a successful target of HMT-based template matching. Indeed, using foreground and background structuring functions is particularly adapted to the invariant vessel properties in terms of shape and intensity. HMT can then be used directly as a vessel extractor or as voxel “vesselness” indicator [24].

The structuring functions to be used aim to match the “cylindrical” shape of vessels (both the vessel itself and its close neighborhood). In order to deal with



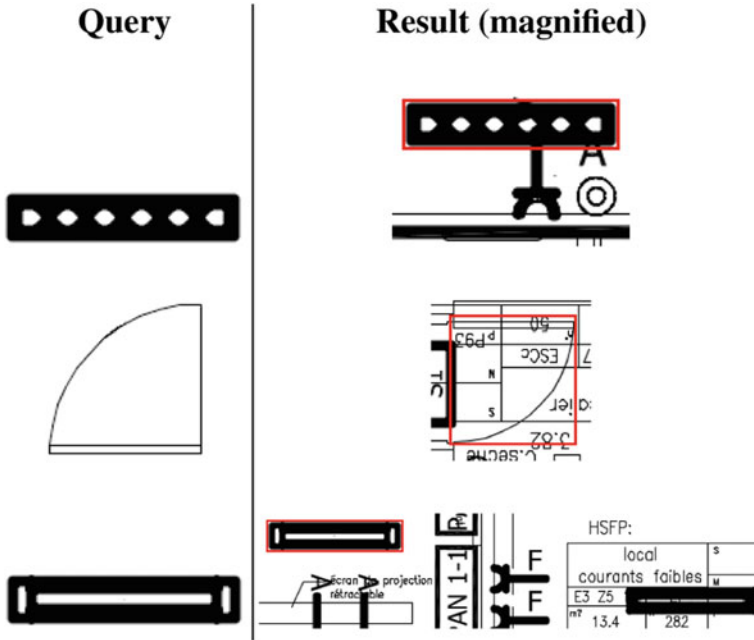


Fig. 24 Difficult spotting cases, from top to bottom: connected symbols, overlapped symbols, similar symbols [41]. The sought template is given on the left, the magnified result provided by the HMT on the right

vessels with different sizes and orientations, a whole set of SFs should be used. Moreover, computation time is reduced by using decimated SF (cf. Sec. 5).

This approach gives excellent results. It obtains a detection rate of 100% for the entrance of the liver portal vein [25] and better results than other methods on brain vessels extraction [26].

### 6.4.2 Rosacea Detection

The color HMT elaborated by Ledoux et al. [19] has been applied to skin image analysis. The goal is to detect specific lesions (rosacea). To do so, the reference colors have been set statistically and the sought template's bounds have been set manually. The preliminary results obtained by the authors call for further exploration of the HMT to extract complex color shapes in dermatology.

## 7 Conclusion

Template matching is a fundamental problem in image analysis and computer vision. It has been addressed very early by Mathematical Morphology, through the well-known Hit-or-Miss Transform. In this chapter, we review most of the existing works on this morphological template matching operator, from the standard case of binary image to the (not so standard) case of grayscale images and the very recent extensions to color and multivariate data. We also discuss the issues raised by the application of the HMT operator in the context of template matching and provide guidelines to the interested reader. Various use cases in different application domains have been provided to illustrate the potential impact of this operator.

While having successfully addressed various real template matching problems, HMT still suffers from some drawbacks which are calling for future work in the field. Among the main issues, we would like to underline the computational complexity of the HMT operator, especially when dealing with multiple templates (SE) to ensure orientation and scale invariance. Besides, adequately defining templates as structuring elements is often tricky and not enough intuitive. We believe that machine learning can be of great help to determine the template or set of templates to be used in the matching process. Finally, while having already benefited from several attempts to increase robustness to uncertainties, too many HMT based template matching methods are limited by the strong constraints brought by the underlying HMT definition. This has prevented the wide dissemination of the HMT as a powerful, reliable and theoretically sound template matching operator.

## References

1. Angulo J, Lefèvre S, Lézoray O (2012) Color representation and processing in polar color spaces. In: C. Fernandez-Maloigne, F. Robert-Inacio, L. Macaire (eds.) Numerical color imaging, ISTE—Wiley pp 1–40
2. Aptoula E, Lefèvre S (2007) A comparative study on multivariate mathematical morphology. *Pattern Recogn* 40(11):2914–2929
3. Aptoula E, Lefèvre S (2008) On lexicographical ordering in multivariate mathematical morphology. *Pattern Recogn Lett* 29(2):109–118
4. Aptoula E, Lefèvre S (2009) Multivariate mathematical morphology applied to colour image analysis. In: C. Collet, J. Chanussot, K. Chehdi (eds.) Multivariate image processing: methods and applications, ISTE—Wiley, pp 303–337
5. Aptoula E, Lefèvre S (2011) Morphological texture description of grayscale and color images. In: P. Hawkes (ed.) *Advances in imaging and electron physics*, vol. 169, Elsevier, pp. 1–74
6. Aptoula E, Lefèvre S, Ronse C (2009) A hit-or-miss transform for multivariate images. *Pattern Recogn Lett* 30(8):760–764
7. Barat C, Ducottet C, Jourlin M (2003) Pattern matching using morphological probing. In: *Proceedings of the 10th international conference on image processing*, vol 1. Barcelona, Spain, pp 369–372
8. Barat C, Ducottet C, Jourlin M (2010) Virtual double-sided image probing: a unifying framework for non-linear grayscale pattern matching. *Pattern Recogn* 43:3433–3447

9. Bloomberg DS, Maragos P (1990) Generalized hit-miss operations. In: SPIE conference 1350, image Algebra and morphological image processing. San Diego, pp 116–128
10. Bloomberg DS, Vincent L (2000) Pattern matching using the blur hit-miss transform. *J Electron Imaging* 9:140–150
11. Boak E, Turner I (2005) Shoreline definition and detection: a review. *J Coastal Res* 21(4):688–703
12. Boykov YY, Jolly MP (2001) Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: Proceedings of the IEEE international conference on computer vision, vol. 1, pp 105–112
13. Doh Y, Kim J, Kim J, Kim S, Alam M (2002) New morphological detection algorithm based on the hit-miss transform. *Opt Eng* 41(1):26–31
14. Goutsias J, Heijmans H, Sivakumar K (1995) Morphological operators for image sequences. *Comput Vis Image Underst* 62(3):326–346
15. Heijmans H, Serra J (1992) Convergence, continuity and iteration in mathematical morphology. *J Vis Commun Image Represent* 3(1):84–102
16. Heijmans H (1994) Morphological image operators: advances in electronics and electron physics series. Academic Press, Boston
17. Heijmans H, Ronse C (1990) The algebraic basis of mathematical morphology, part I: dilations and erosions. *Comput Vision, Graph Image Proc* 50(3):245–295
18. Khosravi M, Schafer R (1996) Template matching based on a grayscale hit-or-miss transform. *IEEE Trans Image Process* 5(5):1060–1066
19. Ledoux A, Richard N, Capelle-Laizé A (2012) Color hit-or-miss transform (CMOMP). In: Proceedings of the EURASIP European conference on signal processing (EUSIPCO). Bucharest, Romania, pp 2248–2252
20. Lefèvre S, Weber J, Sheeren D (2007) Automatic building extraction in VHR images using advanced morphological operators. In: IEEE/ISPRS joint workshop on remote sensing and data fusion over urban areas. Paris, France
21. Murray P, Marshall S (2011) A new design tool for feature extraction in noisy images based on grayscale hit-or-miss transforms. *IEEE Trans Image Process* 20(7):1938–1948
22. Murray P, Marshall S (2013) A review of recent advances in the hit-or-miss transform. In: P.W. Hawkes (ed.) *Advances in imaging and electron physics*, vol. 175, Elsevier, pp 221–282
23. Naegel B, Passat N, Ronse C (2007) Grey-level hit-or-miss transforms—part I: unified theory. *Pattern Recognit* 40(2):635–647
24. Naegel B, Passat N, Ronse C (2007) Grey-level hit-or-miss transforms—part II: application to angiographic image processing. *Pattern Recogn* 40(2):648–658
25. Naegel B, Ronse C, Soler L (2005) Using grey-scale hit-or-miss transform for segmenting the portal network of the liver. In: Proceedings of the ISMM 2005–7th international symposium on mathematical morphology, computational imaging and vision, vol. 30, Springer SBM, pp 429–440
26. Passat N, Ronse C, Baruthio J, Armspach JP, Maillot C (2006) Magnetic resonance angiography: from anatomical knowledge modeling to vessel segmentation. *Med Image Anal* 10(2):259–274
27. Perret B, Lefèvre S, Collet C (2009) A robust hit-or-miss transform for template matching in very noisy astronomical images. *Pattern Recogn* 42(11):2470–2480
28. Puissant A, Lefèvre S, Weber J (2008) Coastline extraction in vhr imagery using mathematical morphology with spatial and spectral knowledge. In: Proceedings of the XXIIth ISPRS congress. Beijing
29. Raducanu B, Grana M (2000) A grayscale hit-or-miss transform based on levet sets. In: Proceedings of the 7th international conference on image processing, vol 2. Vancouver, Canada, pp 931–933
30. Ronse C (1990) Why mathematical morphology needs complete lattices. *Sig Proc* 21(2):129–154
31. Ronse C (1996) A lattice-theoretical morphological view on template extraction in images. *J Vis Commun Image Represent* 7(3):273–295

32. Schaefer R, Casasent D (1995) Nonlinear optical hit-miss transform for detection. *Appl Opt* 34(20):3869–3882
33. Serra J (1993) Anamorphoses and function lattices. In: E.R. Dougherty (ed.) *Mathematical morphology in image processing*, chap. 13, Marcel Dekker, New York, pp 483–523
34. Soille P (2002) Advances in the analysis of topographic features on discrete images. In: *Proceedings of the 10th international conference on discrete geometry for computer imagery DGCI'02*, lecture notes in computer sciences, vol 2301. pp. 175–186
35. Soille P (2002) On morphological operators based on rank filters. *Pattern Recognit* 35(2):527–535
36. Soille P (2003) *Morphological image analysis: principles and applications*. Springer, Berlin
37. Velasco-Forero S, Angulo J (2010) Hit-or-miss transform in multivariate images. In: *proceedings of the advanced concepts for intelligent vision systems*, lecture notes in computer sciences, vol 6474. Springer Verlag, pp 452–463
38. Velasco-Forero S, Angulo J (2011) Supervised ordering in  $r^p$ : application to morphological processing of hyperspectral images. *IEEE Trans Image Process* 20(11):3301–3308
39. Weber J, Lefèvre S (2012) Spatial and spectral morphological template matching. *Image Vis Comput* 30(12):934–945
40. Weber J, Lefèvre S, Gançarski P (2011) Interactive video segmentation based on quasi-flat zones. In: *Proceedings of IEEE international symposium on image and signal processing and analysis*. pp 265–270
41. Weber J, Tabbone S (2012) Symbol spotting for technical documents: an efficient template-matching approach. In: *International conference on pattern recognition (ICPR)*. Tsukuba, Japan
42. Zhao D, Daut DG (1991) Morphological hit-or-miss transformation for shape recognition. *J Vis Commun Image Represent* 2(3):230–243

# Tensor Voting for Robust Color Edge Detection

Rodrigo Moreno, Miguel Angel Garcia and Domenec Puig

**Abstract** This chapter proposes two robust color edge detection methods based on tensor voting. The first method is a direct adaptation of the classical tensor voting to color images where tensors are initialized with either the gradient or the local color structure tensor. The second method is based on an extension of tensor voting in which the encoding and voting processes are specifically tailored to robust edge detection in color images. In this case, three tensors are used to encode local CIELAB color channels and edginess, while the voting process propagates both color and edginess by applying perception-based rules. Unlike the classical tensor voting, the second method considers the context in the voting process. Recall, discriminability, precision, false alarm rejection and robustness measurements with respect to three different ground-truths have been used to compare the proposed methods with the state-of-the-art. Experimental results show that the proposed methods are competitive, especially in robustness. Moreover, these experiments evidence the difficulty of proposing an edge detector with a perfect performance with respect to all features and fields of application.

**Keywords** Edge detection · Perceptual methods · Tensor voting · Perceptual grouping · Non-linear approximation · Curveness and junctionness propagation · Evaluation of edge detectors

---

R. Moreno

Center for Medical Image Science and Visualization and Department of Medical and Health Sciences, Linköping University, Campus US, 58185 Linköping, Sweden  
e-mail: rodrigo.moreno@liu.se

M.A. Garcia

Department of Electronic and Communications Technology,  
Autonomous University of Madrid, Francisco Tomas y Valiente 11, 28049 Madrid, Spain  
e-mail: miguelangel.garcia@uam.es

D. Puig

Intelligent Robotics and Computer Vision Group, Rovira i Virgili University,  
Av. Paisos Catalans 26, 43007 Tarragona, Spain  
e-mail: domenec.puig@urv.cat

## 1 Introduction

Edge detection is an important problem in computer vision, since the performance of many computer vision applications directly depends on the effectiveness of a previous edge detection process. The final goal of edge detection is to identify the locations at which the image has “meaningful” discontinuities. The inherent difficulty in defining what a meaningful discontinuity is has fostered this research area during the last decades. However, in spite of all the efforts, the problem has not completely been solved yet and problems such as automatic tuning of parameters, edge detection in multiscale analysis or noise robustness are still under active research.

The raw output of a general purpose edge detector can be seen as an edginess map, that is, a map of the probability of every pixel being an edge. Since most applications require binary edge maps instead of edginess maps, post-processing steps, such as non-maximum suppression and thresholding with or without hysteresis, are applied to the edginess maps in order to generate such binary maps [5].

In gray-scale images, the Canny’s edge detector [5] has consistently been reported as the best method in many comparisons, e.g., [4, 10, 23, 29, 31]. On the other hand, edge detectors specifically devised for color images usually outperform gray-scale edge detectors. For instance, [3, 28] and [1] have reported a better performance than Canny’s edge detector applied to gray-scale images. Complete reviews of strategies on color edge detection are presented in [11, 12, 25, 30, 33].

Although a number of edge detectors have been proposed during the last years, only a few have been devised to deal with noise. This can be due to the fact that edges of noisy images can be extracted from denoised versions of the input images [6]. This strategy is followed, for example, in Ref. [32]. However, image denoising is not a trivial problem and is still one of the most active research areas in image processing. In addition, the application of image denoising before extracting edges makes it difficult to measure how good the edge detector is, since its performance will be directly related to the performance of the applied filtering stage.

Since the human visual system is able to detect edges in noisy scenarios, the use of perceptual techniques for robust edge detection appears promising. In this context, this paper explores a new approach to extract edges from noisy color images by applying tensor voting, a perceptual technique proposed by Medioni and collaborators [16] as a robust means of extracting perceptual structures from noisy clouds of points. Unfortunately, tensor voting cannot be directly applied to images, since it was devised for dealing with noise in clouds of points instead of in images. Thus, adaptations of this technique are mandatory in order to make it suitable to the problem of edge detection in images. In this chapter, we present two different adaptations of tensor voting to robust color edge detection. The proposed methods take advantage of the robustness of tensor voting to improve the performance in noisy scenarios. These methods are summarized in Sects. 2 and 3.

Recently, we have also introduced a general methodology to evaluate edge detectors directly in gray-scale [22]. This methodology avoids possible biases generated

by post-processing of edginess maps by directly comparing the algorithms in gray-scale. This methodology is summarized in Sect. 4.

The chapter is organized as follows. Sections 2 and 3 detail the two proposed methods for color edge detection based on tensor voting. Section 4 summarizes the methodology of evaluation. Section 5 shows a comparative analysis of the proposed methods against some of the state-of-the-art color edge detection algorithms. Finally, Sect. 6 discusses the obtained results and makes some final remarks.

## 2 Color Edge Detection Through the Classical Tensor Voting

The first adaptation of tensor voting to robust color edge detection is based on using appropriate initialization and post-processing steps of the method proposed by Medioni and collaborators in [16], hereafter referred to as classical tensor voting, which is summarized in the following subsection.

### 2.1 Classical Tensor Voting

Tensor voting is a technique for extracting structure from a cloud of points, in particular in 3D. The method estimates saliency measurements of how likely a point lies on a surface, a curve, a junction, or it is noisy. It is based on the propagation and aggregation of the most likely normal(s) encoded by means of tensors. In a first stage, a tensor is initialized at every point in the cloud either with a first estimation of the normal, or with a ball-shaped tensor if a priori information is not available. Afterwards, every tensor is decomposed into its three components: a *stick*, a *plate* and a *ball*. Every component casts votes, which are tensors that encode the most likely direction(s) of the normal at a neighboring point by taking into account the information encoded by the voter in that component. Finally, the votes are summed up and analyzed in order to estimate surfaceness, curviness and junctionness measurements at every point. Points with low saliency are assumed to be noisy. More formally, the tensor voting at  $\mathbf{p}$ ,  $\text{TV}(\mathbf{p})$  is given by:

$$\text{TV}(\mathbf{p}) = \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p})} \text{SV}(\mathbf{v}, \mathbf{S}_{\mathbf{q}}) + \text{PV}(\mathbf{v}, \mathbf{P}_{\mathbf{q}}) + \text{BV}(\mathbf{v}, \mathbf{B}_{\mathbf{q}}), \quad (1)$$

where  $\mathbf{q}$  represents each of the points in the neighborhood of  $\mathbf{p}$ ,  $\mathcal{N}(\mathbf{p})$ ,  $\text{SV}$ ,  $\text{PV}$  and  $\text{BV}$  are the *stick*, *plate* and *ball* tensor votes cast to  $\mathbf{p}$  by every component of  $\mathbf{q}$ ,  $\mathbf{v} = \mathbf{p} - \mathbf{q}$ , and  $\mathbf{S}_{\mathbf{q}}$ ,  $\mathbf{P}_{\mathbf{q}}$  and  $\mathbf{B}_{\mathbf{q}}$  are the *stick*, *plate* and *ball* components of the tensor at  $\mathbf{q}$  respectively. These components are given by:

$$S_{\mathbf{q}} = (\lambda_1 - \lambda_2) (\mathbf{e}_1 \mathbf{e}_1^T), \tag{2}$$

$$P_{\mathbf{q}} = (\lambda_2 - \lambda_3) (\mathbf{e}_1 \mathbf{e}_1^T + \mathbf{e}_2 \mathbf{e}_2^T), \tag{3}$$

$$B_{\mathbf{q}} = \lambda_3 (\mathbf{e}_1 \mathbf{e}_1^T + \mathbf{e}_2 \mathbf{e}_2^T + \mathbf{e}_3 \mathbf{e}_3^T), \tag{4}$$

where  $\lambda_i$  and  $\mathbf{e}_i$  are the  $i$ th largest eigenvalue and its corresponding eigenvector of the tensor at  $\mathbf{q}$ . Saliency measurements can be estimated from an analysis of the eigenvalues of the resulting tensors. Thus,  $s_1 = (\lambda_1 - \lambda_2)$ ,  $s_2 = (\lambda_2 - \lambda_3)$ , and  $s_3 = \lambda_3$  can be used as measurements of surfaceness, curviness and junctionness respectively.

A *stick* tensor is a tensor with only a single eigenvalue greater than zero. *Stick* tensors are processed through the so-called *stick* tensor voting. The process is illustrated in Fig. 1. Given a known *stick* tensor  $S_{\mathbf{q}}$  at  $\mathbf{q}$ , the orientation of the vote cast by  $\mathbf{q}$  to  $\mathbf{p}$  can be estimated by tensorizing the normal of a circumference at  $\mathbf{p}$  that joins  $\mathbf{q}$  and  $\mathbf{p}$ . This vote is then weighted by a decaying scalar function,  $w_s$ . The *stick* tensor vote is given by [20]:

$$SV(\mathbf{v}, S_{\mathbf{q}}) = w_s R_{2\theta} S_{\mathbf{q}} R_{2\theta}^T, \tag{5}$$

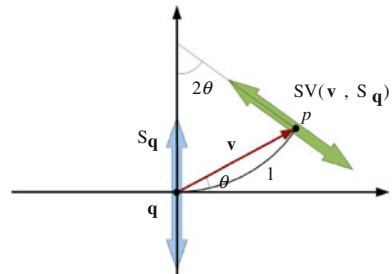
where  $\theta$  is shown in Fig. 1 and  $R_{2\theta}$  represents a rotation with respect to the axis  $\mathbf{v} \times (S_{\mathbf{q}} \mathbf{v})$ , which is perpendicular to the plane that contains  $\mathbf{v}$  and  $S_{\mathbf{q}} \mathbf{v}$ ; and  $w_s$  is an exponential decaying function that penalizes the arc-length  $l$ , and the curvature of the circumference,  $\kappa$ :

$$w_s = \begin{cases} e^{-\frac{l^2}{\sigma^2} + b\kappa^2} & \text{if } \theta \leq \pi/4 \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where  $\sigma$  and  $b$  are parameters to weight the scale and the curvature respectively.

In turn, a *plate* tensor is a tensor with  $\lambda_1 = \lambda_2 \geq 0$  and  $\lambda_3 = 0$ . *Plate* tensors are processed through the so-called *plate* tensor voting. The *plate* tensor voting uses the fact that any *plate* tensor  $P$ , can be decomposed into all possible *stick* tensors inside the *plate*. Let  $S_P(\beta) = R_{\beta} \mathbf{e}_1 \mathbf{e}_1^T R_{\beta}^T$  be a *stick* inside the *plate*  $P$ , with  $\mathbf{e}_1$  being its principal eigenvector, and  $R_{\beta}$  being a rotation with respect to an axis perpendicular to  $\mathbf{e}_1$  and  $\mathbf{e}_2$ . Thus, the *plate* vote is defined as [20]:

**Fig. 1** *Stick* tensor voting. A *stick*  $S_{\mathbf{q}}$  casts a *stick* vote  $SV(\mathbf{v}, S_{\mathbf{q}})$  to  $\mathbf{p}$ , which corresponds to the most likely tensorized normal at  $\mathbf{p}$





$$PV(\mathbf{v}, P_{\mathbf{q}}) = \frac{\lambda_{1P_{\mathbf{q}}}}{\pi} \int_0^{2\pi} SV(\mathbf{v}, S_{P_{\mathbf{q}}}(\beta)) d\beta, \quad (7)$$

where  $\lambda_{1P_{\mathbf{q}}}$  is the largest eigenvalue of  $P_{\mathbf{q}}$ .

Finally, a *ball* tensor is a tensor with  $\lambda_1 = \lambda_2 = \lambda_3 \geq 0$ . The *ball* tensor voting is defined in a similar way as the *plate* tensor voting. Let  $S_B(\phi, \psi)$  be a unitary *stick* tensor oriented in the direction  $(1, \phi, \psi)$  in spherical coordinates. Then, any *ball* tensor  $B$  can be written as [20]:

$$BV(\mathbf{v}, B_{\mathbf{q}}) = \frac{3\lambda_{1B_{\mathbf{q}}}}{4\pi} \int_{\Gamma} SV(\mathbf{v}, S_{B_{\mathbf{q}}}(\phi, \psi)) d\Gamma, \quad (8)$$

where  $\Gamma$  represents the surface of the unitary sphere, and  $\lambda_{1B_{\mathbf{q}}}$  is the largest eigenvalue of  $B_{\mathbf{q}}$ .

## 2.2 Color Edge Detection Through the Classical Tensor Voting

In Ref. [21], we showed that the classical tensor voting and the well-known structure tensor [8] are closely related. These similarities were used in [21] to extend classical tensor voting to different types of images, especially color images. This extension can be used to extract edges. This subsection summarizes that method for gray-scale and color images.

### 2.2.1 Gray-Scale Images

Tensor voting can be adapted in order to robustly detect edges in gray-scale images by following three steps. First, the tensorized gradient,  $\nabla u \nabla u^T$ , is used to initialize a tensor at every pixel. Second, the *stick* tensor voting is applied in order to propagate the information encoded in the tensors. In this case, it is not necessary to apply the *plate* and *ball* voting processes since the *plate* and *ball* components are zero at every pixel. Thus, tensor voting is reduced to:

$$TV(\mathbf{p}) = \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p})} SV(\mathbf{v}, \nabla u_{\mathbf{q}} \nabla u_{\mathbf{q}}^T). \quad (9)$$

Finally, the resulting tensors are rescaled by the factor:

$$\xi = \frac{\sum_{\mathbf{p} \in \Omega} \text{trace}(\nabla u_{\mathbf{p}} \nabla u_{\mathbf{p}}^T)}{\sum_{\mathbf{p} \in \Omega} \text{trace}(TV(\mathbf{p}))}, \quad (10)$$

in order to renormalize the total energy of the tensorized gradient, where  $\Omega$  refers to the given image.

After having applied tensor voting and the energy normalization step, the principal eigenvalue  $\lambda_1$  of the resulting tensors can be used to detect edges, since it attains high values not only at boundaries but also at corners.

### 2.2.2 Color Images

Figure 2 shows the two possible options to extend tensor voting to color images using the adaptation proposed in the previous subsection. The first option is to apply the *stick* tensor voting independently to every channel and then adding the individual results, that is:

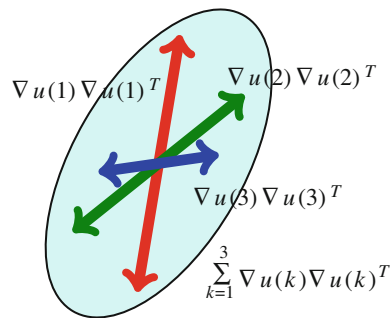
$$TV(\mathbf{p}) = \sum_{k=1}^3 \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p})} \alpha_k SV(\mathbf{v}, \nabla u_{\mathbf{q}}(k) \nabla u_{\mathbf{q}}(k)^T), \quad (11)$$

where  $\nabla u(k)$  is the gradient at color channel  $k$ , and  $\alpha_k$  are weights used to give different relevance to every channel.

The second option is to apply (1) to the sum of tensorized gradients, with  $S_{\mathbf{q}}$ ,  $P_{\mathbf{q}}$  and  $B_{\mathbf{q}}$  being the *stick*, *plate* and *ball* components of  $T_{\mathbf{q}} = \sum_{k=1}^3 \alpha_k \nabla u_{\mathbf{q}}(k) \nabla u_{\mathbf{q}}(k)^T$ . For two-dimensional images, the computation of *plate* votes can be avoided since  $P_{\mathbf{q}} = 0$ . Thus, the first option has the advantage that only the application of *stick* tensor voting is necessary, whereas the second option requires *stick* and *ball* tensor voting.

In practice, both strategies are very similar since  $T_{\mathbf{q}} \approx S_{\mathbf{q}}$  in most pixels of images of natural scenes [21]. Thus, in the experiments of Section 5, the first option has been used for the majority of pixels, whereas the second one only in those pixels in which the aforementioned approximation is not valid. In practice, the first option can be applied when the angle between any pair of gradients is below a threshold.

**Fig. 2** Tensor voting can be applied to the color channels independently (the *red*, *green* and *blue sticks*) or to the sum of the tensorized gradients (the *ellipse*)



Similarly to the case of gray-scale images, the classical tensor voting can be used to detect edges by means of the principal eigenvalue  $\lambda_1$  of the resulting tensor, after an energy normalization step similar to the one of (10).

Since this method does not apply any pre-processing step, its robustness must completely rely on the robustness of the classical tensor voting. This could not be sufficient in highly noisy scenarios. Thus, in order to improve the results it is necessary to iterate the method. By iterating tensor voting, the most significant edges can be reinforced at the expense of discarding small ridges. According to our experiments, a few iterations (two or three) usually give good results for both noisy and noiseless images.

### 3 Color Edge Detection Through an Adapted Tensor Voting

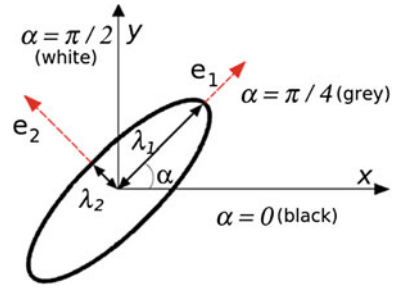
It is important to remark that tensor voting is a methodology in which information encoded through tensors is propagated and aggregated in a local neighborhood. Thus, it is possible to devise more appropriate methods for specific applications by tailoring the way in which tensors are encoded, propagated and aggregated, while maintaining the tensor voting spirit. In this line, we introduced a method for image denoising [17, 19] that can also be applied to robust color edge detection, since both problems can be tackled at the same time [18]. The next subsections detail the edge detector.

#### 3.1 Encoding of Color Information

Before applying the proposed method, color is converted to the CIELAB space. Every CIELAB channel is then normalized to the range  $[0, \pi/2]$ . In the first step of the method, the information of every pixel is encoded through three second-order 2D tensors, one for each normalized CIELAB color channel.

Three perceptual measures are encoded in the tensors associated with every input pixel, namely: the normalized color at the pixel (in the specific channel), a measure of local uniformity (how edgeless its neighborhood is), and an estimation of edginess. Figure 3 shows the graphical interpretation of a tensor for channel  $L$ . The normalized color is encoded by the angle  $\alpha$  between the  $x$  axis, which represents the lowest possible color value in the corresponding channel, and the eigenvector corresponding to the largest eigenvalue. For example, in channel  $L$ , a tensor with  $\alpha = 0$  encodes black, whereas a tensor with  $\alpha = \frac{\pi}{2}$  encodes white. In addition, local uniformity and edginess are encoded by means of the normalized  $\hat{s}_1 = (\lambda_1 - \lambda_2)/\lambda_1$  and  $\hat{s}_2 = \lambda_2/\lambda_1$  saliencies respectively. Thus, a pixel located at a completely uniform region is represented by means of three *stick tensors*, one for each color channel. In contrast, a pixel located at an ideal edge is represented by means of three *ball tensors*, one for every color channel.

**Fig. 3** Encoding process for channel  $L$ . Color, uniformity and edginess are encoded by means of  $\alpha$  and the normalized saliencies  $\hat{s}_1 = (\lambda_1 - \lambda_2)/\lambda_1$  and  $\hat{s}_2 = \lambda_2/\lambda_1$  respectively



Before applying the voting process, it is necessary to initialize the tensors associated with every pixel. The colors of the noisy image can be easily encoded by means of the angle  $\alpha$  between the  $x$  axis and the principal eigenvector, as described above. However, since metrics of uniformity and edginess are usually unavailable at the beginning of the voting process, normalized saliency  $\hat{s}_1$  is initialized to one and normalized saliency  $\hat{s}_2$  to zero. These initializations allow the method to estimate more appropriate values of the normalized saliencies for the next stages, as described in the next subsection. Thus, the initial color information of a pixel is encoded through three *stick tensors* oriented along the directions that represent that color in the normalized CIELAB channels:

$$\mathbf{T}_k(\mathbf{p}) = \mathbf{t}_k(\mathbf{p}) \mathbf{t}_k(\mathbf{p})^T, \quad (12)$$

where  $\mathbf{T}_k(\mathbf{p})$  is the tensor of the  $k$ th color channel ( $L$ ,  $a$  and  $b$ ) at pixel  $\mathbf{p}$ ,  $\mathbf{t}_k(\mathbf{p}) = [\cos(C_k(\mathbf{p})) \quad \sin(C_k(\mathbf{p}))]^T$ , and  $C_k(\mathbf{p})$  is the normalized value of the  $k$ -th color channel at  $\mathbf{p}$ .

### 3.2 Voting Process

The voting process requires three measurements for every pair of pixels  $\mathbf{p}$  and  $\mathbf{q}$ : the perceptual color difference,  $\Delta E_{\mathbf{p}\mathbf{q}}$ ; the joint uniformity measurement,  $U_k(\mathbf{p}, \mathbf{q})$ , used to determine if both pixels belong to the same region; and the likelihood of a pixel being impulse noise,  $\eta_k(\mathbf{p})$ .  $\Delta E_{\mathbf{p}\mathbf{q}}$  is calculated through CIEDE2000 [13], while

$$U_k(\mathbf{p}, \mathbf{q}) = \hat{s}_{1k}(\mathbf{p}) \hat{s}_{1k}(\mathbf{q}), \quad (13)$$

and

$$\eta_k(\mathbf{p}) = \begin{cases} \hat{s}_{2c}(\mathbf{p}) - \mu_{\hat{s}_{2c}}(\mathbf{p}) & \text{if } \mathbf{p} \text{ is located at a local maximum} \\ 0 & \text{otherwise} \end{cases}, \quad (14)$$

where  $\mu_{\hat{s}_{2c}}(\mathbf{p})$  represents the mean of  $\hat{s}_{2c}$  over the neighborhood of  $\mathbf{p}$ .

In the second step of the method, the tensors associated with every pixel are propagated to their neighbors through a convolution-like process. This step is independently applied to the tensors of every channel ( $L$ ,  $a$  and  $b$ ). The voting process is carried out by means of specially designed tensorial functions referred to as *propagation functions*, which take into account not only the information encoded in the tensors but also the local relations between neighbors. Two propagation functions are proposed for edge detection: a *stick* and a *ball* propagation function. The *stick* propagation function is used to propagate the most likely noiseless color of a pixel, while the *ball* propagation function is used to increase edginess where required. The application of the first function leads to *stick* votes, while the application of the second function produces *ball* votes. *Stick* votes are used to eliminate noise and increase the edginess where the color of the voter and the voted pixels are different. *Ball* votes are used to increase the relevance of the most important edges.

A *stick* vote can be seen as a *stick*-shaped tensor,  $ST_k(\mathbf{p})$ , with a strength modulated by three scalar factors. The proposed *stick* propagation function,  $S_k(\mathbf{p}, \mathbf{q})$ , which allows a pixel  $\mathbf{p}$  to cast a *stick* vote to a neighboring pixel  $\mathbf{q}$  for channel  $k$  is given by:

$$S_k(\mathbf{p}, \mathbf{q}) = GS(\mathbf{p}, \mathbf{q}) \overline{\eta}_k(\mathbf{p}) SV'_k(\mathbf{p}, \mathbf{q}) ST_k(\mathbf{p}), \quad (15)$$

with  $ST_k(\mathbf{p})$ ,  $GS(\mathbf{p}, \mathbf{q})$ ,  $\overline{\eta}_k(\mathbf{p})$  and  $SV'_k(\mathbf{p}, \mathbf{q})$  being defined as follows. First, the tensor  $ST_k(\mathbf{p})$  encodes the most likely normalized noiseless color at  $\mathbf{p}$ . Thus,  $ST_k(\mathbf{p})$  is defined as the tensorized eigenvector corresponding to the largest eigenvalue of the voter pixel, that is:

$$ST_k(\mathbf{p}) = \mathbf{e}_{1k}(\mathbf{p}) \mathbf{e}_{1k}(\mathbf{p})^T, \quad (16)$$

being  $\mathbf{e}_{1k}(\mathbf{p})$  the eigenvector with the largest eigenvalue of the tensor associated with channel  $k$  at  $\mathbf{p}$ . Second, the three scalar factors in (15), each ranging between zero and one, are defined as follows. The first factor,  $GS(\mathbf{p}, \mathbf{q})$ , models the influence of the distance between  $\mathbf{p}$  and  $\mathbf{q}$  in the vote strength. Thus,  $GS(\mathbf{p}, \mathbf{q}) = G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|)$ , where  $G_{\sigma_s}(\cdot)$  is a decaying Gaussian function with zero mean and a user-defined standard deviation  $\sigma_s$ . The second factor,  $\overline{\eta}_k(\mathbf{p})$  defined as  $\overline{\eta}_k(\mathbf{p}) = 1 - \eta_k(\mathbf{p})$ , is introduced in order to prevent a pixel  $\mathbf{p}$  previously classified as impulse noise from propagating its information. The third factor,  $SV'_k$ , takes into account the influence of the perceptual color difference, the uniformity and the noisiness of the voted pixel. This factor is given by:

$$SV'_k(\mathbf{p}, \mathbf{q}) = \overline{\eta}_k(\mathbf{q}) SV_k(\mathbf{p}, \mathbf{q}) + \eta_k(\mathbf{q}), \quad (17)$$

where:  $SV_k(\mathbf{p}, \mathbf{q}) = [G_{\sigma_d}(\Delta E_{\mathbf{p}\mathbf{q}}) + U_k(\mathbf{p}, \mathbf{q})]/2$ , and  $\overline{\eta}_k(\mathbf{q}) = 1 - \eta_k(\mathbf{q})$ .  $SV_k(\mathbf{p}, \mathbf{q})$  allows a pixel  $\mathbf{p}$  to cast a stronger *stick* vote to  $\mathbf{q}$  either if both pixels belong to the same uniform region, or if the perceptual color difference between them is small. That behavior is achieved by means of the factors  $U_k(\mathbf{p}, \mathbf{q})$  and the decaying Gaussian function on  $\Delta E_{\mathbf{p}\mathbf{q}}$  with a user-defined standard deviation  $\sigma_d$ . A normalizing factor of two is used in order to make  $SV_k(\mathbf{p}, \mathbf{q})$  vary from zero to one. The term  $\eta_k(\mathbf{q})$  in (17) makes noisy voted pixels,  $\mathbf{q}$ , to adopt the color of their voting neighbors,  $\mathbf{p}$ , dis-

regarding local uniformity measurements and perceptual color differences between  $\mathbf{p}$  and  $\mathbf{q}$ . The term  $\overline{\eta}_k(\mathbf{q})$  in (17) makes  $SV'_k$  vary from zero to one. The effect of  $\eta_k(\mathbf{q})$  and  $\overline{\eta}_k(\mathbf{q})$  on the strength of the *stick* vote received at a noiseless pixel  $\mathbf{q}$  is null.

In turn, a *ball* vote can be seen as a *ball*-shaped tensor,  $\mathbf{BT}(\mathbf{p})$ , with a strength controlled by the scalar factors  $GS(\mathbf{p}, \mathbf{q})$ ,  $\overline{\eta}_k(\mathbf{p})$  and  $BV_k(\mathbf{p}, \mathbf{q})$ , each varying between zero and one. The *ball* propagation function,  $\mathbf{B}_k(\mathbf{p}, \mathbf{q})$ , which allows a pixel  $\mathbf{p}$  to cast a *ball* vote to a neighboring pixel  $\mathbf{q}$  for channel  $k$  is given by:

$$\mathbf{B}_k(\mathbf{p}, \mathbf{q}) = GS(\mathbf{p}, \mathbf{q}) \overline{\eta}_k(\mathbf{p}) BV_k(\mathbf{p}, \mathbf{q}) \mathbf{BT}(\mathbf{p}), \quad (18)$$

with  $\mathbf{BT}(\mathbf{p})$ ,  $GS(\mathbf{p}, \mathbf{q})$ ,  $\overline{\eta}_k(\mathbf{p})$  and  $BV_k(\mathbf{p}, \mathbf{q})$  being defined as follows. First, the *ball tensor*, represented by the identity matrix,  $\mathbf{I}$ , is the only possible tensor for  $\mathbf{BT}(\mathbf{p})$ , since it is the only tensor that complies with the two main design restrictions: a *ball* vote must be equivalent to casting *stick* votes for all possible colors using the hypothesis that all of them are equally likely, and the normalized  $\hat{s}_1$  saliency must be zero when only *ball* votes are received at a pixel. Second,  $GS(\mathbf{p}, \mathbf{q})$  and  $\overline{\eta}_k(\mathbf{p})$  are the same as the factors introduced in (15) for the *stick* propagation function. They are included for similar reasons to those given in the definition of the *stick* propagation function. Finally, the scalar factor  $BV_k(\mathbf{p}, \mathbf{q})$  is given by:

$$BV_k(\mathbf{p}, \mathbf{q}) = \frac{\overline{G_{\sigma_d}}(\Delta E_{\mathbf{p}\mathbf{q}}) + \overline{U}_k(\mathbf{p}, \mathbf{q}) + \overline{G_{\sigma_d}}(\Delta E_{\mathbf{p}\mathbf{q}}^k)}{3}, \quad (19)$$

where  $\overline{G_{\sigma_d}}(\cdot) = 1 - G_{\sigma_d}(\cdot)$  and  $\overline{U}_k(\mathbf{p}, \mathbf{q}) = 1 - U_k(\mathbf{p}, \mathbf{q})$ .  $BV_k(\mathbf{p}, \mathbf{q})$  models the fact that a pixel  $\mathbf{p}$  must reinforce the edginess at the voted pixel  $\mathbf{q}$  either if there is a big perceptual color difference between  $\mathbf{p}$  and  $\mathbf{q}$ , or if  $\mathbf{p}$  and  $\mathbf{q}$  are not in a uniform region. This behavior is modeled by means of  $\overline{G_{\sigma_d}}(\Delta E_{\mathbf{p}\mathbf{q}})$  and  $\overline{U}_k(\mathbf{p}, \mathbf{q})$ . The additional term  $\overline{G_{\sigma_d}}(\Delta E_{\mathbf{p}\mathbf{q}}^k)$  is introduced in order to increase the edginess of pixels in which the only noisy channel is  $k$ , where  $\Delta E_{\mathbf{p}\mathbf{q}}^k$  denotes the perceptual color difference only measured in the specific color channel  $k$ . The normalizing factor of three in (19) allows the *ball* propagation function to cast *ball* votes with a strength between zero and one.

The proposed voting process at every pixel is carried out by adding all the tensors propagated towards it from its neighbors by applying the above propagation functions. Thus, the total vote received at a pixel  $\mathbf{q}$  for each color channel  $k$ ,  $\mathbf{TV}_k(\mathbf{q})$ , is given by:

$$\mathbf{TV}_k(\mathbf{q}) = \sum_{\mathbf{p} \in \mathcal{N}(\mathbf{q})} S_k(\mathbf{p}, \mathbf{q}) + \mathbf{B}_k(\mathbf{p}, \mathbf{q}). \quad (20)$$

The voting process is applied twice. The first application is used to obtain an initial estimation of the normalized  $\hat{s}_1$  and  $\hat{s}_2$  saliencies, as they are necessary to calculate  $U_k(\mathbf{p}, \mathbf{q})$  and  $\eta_k(\mathbf{p})$ . For this first estimation, only perceptual color differences and spatial distances are taken into account. At the second application, the tensors at every

pixel are initialized with the tensors obtained after the first application. After this initialization, (15) and (18) can be applied in their full definition, since all necessary data are available.

After applying the voting process described above, it is necessary to obtain eigenvectors and eigenvalues of  $TV_L(\mathbf{p})$ ,  $TV_a(\mathbf{p})$  and  $TV_b(\mathbf{p})$  at every pixel  $\mathbf{p}$  in order to analyze its local perceptual information. The voting results can be interpreted as follows: uniformity increases with the normalized  $\hat{s}_1$  saliency and edginess increases as the normalized  $\hat{s}_2$  saliency becomes greater than the normalized  $\hat{s}_1$  saliency. Hence, the map of normalized  $\hat{s}_2$  saliencies can be directly used as an edginess map:

$$E(\mathbf{p}) = \sum_{k=1}^3 \alpha_k \hat{s}_{2k}(\mathbf{p}), \quad (21)$$

where  $E(\mathbf{p})$  is the edginess at  $\mathbf{p}$  and  $\alpha_k$  are weights that can be used to modulate the importance of every channel in the estimation of edginess.

The results can be improved by reducing the noise in the image. This denoising step can be achieved by replacing the pixel's color by the most likely normalized noiseless color encoded in its tensors. Similarly to the method based on the classical tensor voting, this edge detector is expected to yield better results by iterating the process. Experimentally, it has been found that a few iterations (less than five in any case) can yield good results for both noisy and noiseless images.

### 3.3 Parameters of the CIEDE2000 Formula

The CIEDE2000 formula [13], which estimates the perceptual color difference between two pixels  $\mathbf{p}$  and  $\mathbf{q}$ ,  $\Delta E_{\mathbf{p}\mathbf{q}}$ , has three parameters,  $k_L$ ,  $k_C$  and  $k_H$ , to weight the differences in CIELAB luminance, chroma and hue respectively. They can be adjusted to make the CIEDE2000 formula more suitable for every specific application by taking into account factors such as noise or background luminance, since those factors were not explicitly taken into account in the definition of the formula. These parameters must be greater than or equal to one. The following equations can be used to compute these parameters:

$$k_L = B_L \eta_L, \quad k_C = B_C \eta_C, \quad k_H = B_h \eta_h, \quad (22)$$

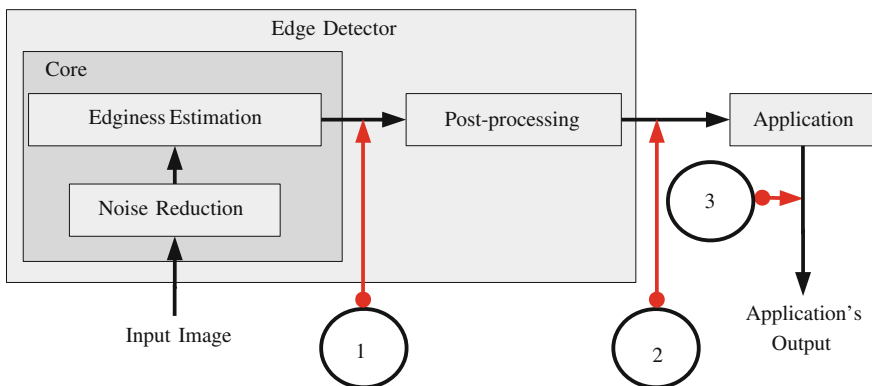
where  $B_m$  are factors that take into account the influence of the background color on the calculation of color differences for the color component  $m$  ( $L$ ,  $C$  and  $h$ ) and  $F_{\eta_m}$  are factors that take into account the influence of noise on the calculation of color differences in component  $m$ . On the one hand, big color differences in chromatic channels become less perceptually visible as background luminance decreases. Thus, the influence of the background on the CIEDE2000 formula can be modeled by  $B_L = 1$  and  $B_C = B_h = 1 + 3(1 - Y_B)$ , where  $Y_B$  is the mean background

luminance. On the other hand, big color differences become less perceptually visible as noise increases. The influence of noise on CIEDE2000 can be modeled by means of  $\eta_m = MAD(I)_m - MAD(G)_m$ , where  $I$  is the image,  $G$  is a Gaussian blurred version of  $I$  and  $MAD(\cdot)_m$  is the median absolute difference (MAD) calculated on component  $m$ . In turn,  $\eta_m$  is set to 1 in noiseless regions.

### 4 Evaluation Methodology

In general, edge detectors apply three steps (cf. Fig. 4). First, a filtering step is applied to the input image, since edge detectors are very sensitive to noise. Second, once the input image is noiseless, edge detectors estimate the likelihood of finding an edge for every pixel. The output of this step is an edginess map. Finally, post-processing is applied to the edginess map in order to obtain a binary edge map. The core of the edge detection algorithms embodies only the first two steps, leaving aside the post-processing step, since the latter is usually application-dependent. In addition, it is not possible to separate the denoising and edginess estimation steps in general, since many algorithms carry out both processes in a unified framework.

The performance of edge detection algorithms can be assessed at three different points of the process, as shown in Fig. 4. Measurements on edginess maps can be obtained at the first point, on binary edge maps at the second point, and application-dependent measurements can be made at the third point of the figure. Direct measurements at the output of the algorithms are made at the first and second points, while the performance at the third point is indirectly assessed by taking into account the application in which the edge detector is used. Indirect assessment is based on the assumption that the performance of an edge detector used in the context of a specific application is correlated with the general performance of that application. Assess-



**Fig. 4** The edge detection process. The performance of edge detectors can be assessed at the points 1, 2 and 3



ing performance at the first point appears to be advantageous since the core of the edge detectors can be evaluated no matter the context or the applied post-processing. Many evaluation methodologies have been proposed to evaluate performance at the second e.g., [4, 10, 15, 26] and third points [2, 29]. On the other hand, to the best of our knowledge, the methodology introduced in [22] has been the first attempt to measure performance at the first point.

There are mainly four features that can be measured from edge detectors: completeness, discriminability, precision and robustness. Without loss of generality, completeness, discriminability and precision can be measured on non-maximum suppressed edginess maps, here referred to as NMSE maps, since the location of edges must be the same, disregarding the strength given to them by the detector. On the other hand, robustness can be directly assessed on the edginess maps. These features are described in the following paragraphs.

#### 4.1 Completeness

Completeness is the ability of an edge detector to mark all possible edges of noiseless images. Completeness is a desirable feature of general purpose edge detectors since the decision of whether an edge is relevant or not only depends on the application. For instance, applications such as image edge enhancement based on edge detection, edge-based segmentation or texture feature extraction usually give a different relevance to every detected edge. Consequently, an edge detector will reduce its scope when it discards edges. Despite that, most edge detectors usually opt for decreasing their scope of use for the sake of improving their performance in other features, such as discriminability or robustness.

Complete ground-truths with all possible edges must be used to measure completeness. Unfortunately, that kind of ground-truth is not usually available. Thus, recall, the ground-truth dependent counterpart of completeness, can be used to give partial estimations of completeness. Let  $D(\mathbf{p}_i)$  be the distance between the  $i$ th pixel in the ground-truth  $\mathbf{p}_i$ , and its corresponding matching pixel  $\mathbf{q}_i$  in the NMSE map or infinity if such a matching pixel does not exist,  $M$  and  $N$  be the number of marked pixels in the ground-truth and in the NMSE map respectively, and let  $\phi(\cdot)$  be a radial decaying function in the range from zero to one. Function  $\phi(x) = 1/(1 + (1/9)x^2)$  has been used in the experiments of Sect. 5. Recall can be estimated through the  $R$ -measurement defined as [22]:

$$R = \frac{1}{M} \sum_{i=1}^M \phi(D(\mathbf{p}_i)). \quad (23)$$

A problem associated with the measure of recall when  $N > M$  is the fact that every edge detector generates a different number of edges. This can give advantage to detectors that generate a larger number of edges, since  $D(\mathbf{p}_i)$  tends to be reduced

when  $N$  increases. This bias can be suppressed by taking the same number of detected edges for all the edge detectors to be compared. This can be done by taking the  $N'$  strongest detected edges from the NMSE maps.  $R$  vs.  $N'$  plots can also be used to analyze the evolution of  $R$ .

## 4.2 Discriminability

Discriminability is the ability of an edge detector to discriminate between relevant and irrelevant edges. This feature is application-dependent since relevance can only be assessed in a specific scope. For example, the discriminability of an edge detector could be high when applied to image edge enhancing or low when applied to edge-based segmentation. Discriminability is one of the most desirable features of edge detectors since low levels of discriminability make it necessary to use more sophisticated post-processing algorithms that can partially fix the drawbacks of the edge detector. Thus, global thresholding (which is the simplest post-processing) could be used for edge detectors with maximum discriminability.

Discriminability can be measured related to a specific ground-truth through the  $DS$ -measurement: the difference between the weighted mean edginess of the pixels that match the ground-truth and the weighted mean edginess of the pixels that do not match it. Let  $E(\mathbf{q}_i)$  be the edginess at pixel  $\mathbf{q}_i$  of the NMSE map, and  $D(\mathbf{q}_i)$  be the distance between  $\mathbf{q}_i$  and its matching pixel in the ground-truth or infinity if such a pixel does not exist. The  $DS$ -measurement is given by [22]:

$$DS = \frac{\sum_{i=1}^N E(\mathbf{q}_i) \phi(D(\mathbf{q}_i))}{\sum_{i=1}^N \phi(D(\mathbf{q}_i))} - \frac{\sum_{i=1}^N E(\mathbf{q}_i) (1 - \phi(D(\mathbf{q}_i)))}{\sum_{i=1}^N 1 - \phi(D(\mathbf{q}_i))}. \quad (24)$$

## 4.3 Precision

Precision measures the ability of an edge detector to mark edges as close as possible to real edges. Precision is mandatory for edge detection, since the performance of applications in which the detectors are used depends on this feature. Unlike discriminability, precision is, in essence, an application-independent feature. However, in practice, application-independent measures of precision are difficult to obtain since complete ground-truths are required. Thus, only precision measurements related to specific ground-truths can be obtained. Ideally, all edges of the ground-truth should be found at distance zero in the NMSE map. However, if hand-made ground-truths are used, it is necessary to take into account that those ground-truths are not precise,

since some pixels can appear misplaced due to human errors. Despite that, those ground-truths can still be used to compare edge detectors, since all edge detectors are equally affected by those errors. Let  $\bar{D}$  be the mean distance between pixels of the ground-truth and their corresponding matching pixels in the NMSE map. The  $P$ -measurement can be used to estimate precision:

$$P = \phi(\bar{D}). \quad (25)$$

Observe that pixels without a matching pixel in the NMSE are not considered for the  $P$ -measurement, since they are irrelevant for measuring precision. Notice that based on the definition of function  $\phi$  described in Sect. 4.1, values of  $P$  below 0.69 and above 0.90 correspond to a mean distance between matching points in the ground-truth and the NMSE above 2 pixels and below 1 pixel, respectively. Indeed, this behavior can be modified by varying function  $\phi$ .

A feature related to precision is the false alarm rejection (FAR) feature, which represents the ability of edge detectors not to detect edges in flat regions. The  $FAR$ -measurement is given by:

$$FAR = \frac{1}{N} \sum_{i=1}^N \phi(D(\mathbf{q}_i)). \quad (26)$$

This measurement can be used as a numeric, ground-truth dependent estimation of false alarm rejection. Similarly to the  $R$ -measurement, the  $N'$  strongest detected edges from the NMSE map must be selected before computing the  $P$  and  $FAR$ -measurements in order to avoid biases related to  $N$  when  $N > M$ . Thus, plots of  $P$  vs.  $N'$  and  $FAR$  vs.  $N'$  can also be used to evaluate the evolution of the  $P$  and  $FAR$ -measurements.

#### 4.4 Robustness

Robustness measures the ability of an edge detector to reject noise. Thus, an ideal robust edge detector should produce the same output for both noisy and noiseless images. Robustness is one of the most difficult features to comply with since edge detection is essentially a differential operation which is usually very sensitive to noise. In fact, most edge detectors include filtering steps in order to improve their robustness. However, most of those filters mistakenly eliminate important details by treating them as noise, thus reducing the completeness and recall features of the detector. Despite that, robustness is usually preferred to completeness.

Since edginess maps can be modeled by means of gray-scale images, measures of image fidelity can be used to measure robustness. The peak signal to noise ratio (PSNR) is the most widely used measure of image fidelity. Although PSNR is not suitable to measure precision [27], it is appropriate to measure robustness. The edge

detector is applied to both the noiseless and the noisy version of the same image. The PSNR between both outputs is calculated in order to measure the difference between them. Unlike the aforementioned measurements, it is not necessary to use ground-truths or to apply non-maximum suppression to the edginess maps before computing PSNR.

#### ***4.5 Ground-Truths for Edge Detection Assessment***

Ground-truths are very important for assessing edge detectors. However, they must be used carefully in order to obtain reliable and fair assessments [24]. Ground-truths can be classified into artificial, manual or generated by consensus. Artificial ground-truths are trivially obtained from synthetic images, manual ground-truths are obtained by manually annotating edges on the images, such as the Berkeley database presented in [14] for image segmentation and ground-truths generated by consensus are obtained from the output of a bank of edge detectors (e.g., [7]).

Artificial ground-truths are not generally used in comparisons since the results can barely be extrapolated to real scenes [4]. In turn, manual ground-truths are useful since edge detector outputs must correlate with the opinion of humans. However, manual ground-truths must be treated as partial ground-truths, since humans annotate edges depending on the instructions given by the experimenters. For example, humans do not usually mark the edges inside a textured region when the instruction is to annotate the necessary edges to separate regions. Thus, a manual ground-truth obtained for image segmentation should not be used for image edge enhancement, for example. Moreover, precision measurements using this kind of ground-truth can only be seen as estimates, since humans are prone to committing precision errors when marking edges. An additional problem is that gray-scale manual ground-truths are almost impossible to obtain for natural scenes [9].

Ground-truths generated by consensus rely on the hypothesis that the bank of edge detectors have a good performance in all contexts. This kind of ground-truth is easy to construct, including gray-scale ground-truths. However, the validity of these ground-truths directly depends on the choice of the bank of edge detectors.

### **5 Experimental Results**

Fifteen images from the Berkeley segmentation data set [14] have been used in the experiments. In addition to the Laplacian of Gaussians (LoG), Sobel and Canny detectors, the methods proposed in [1], referred to as the LGC method, and [28], referred to as the Compass method, have been used in the comparisons, since they are considered to represent the state-of-the-art in color edge detection, and on top of that, implementations are available from their authors. The Compass, LoG and Canny algorithms have been applied with  $\sigma = 2$ , since the best overall performance

of these algorithms has been attained with this standard deviation. Three iterations of the proposed methods have been run. The parameters of the method based on the classical tensor voting, referred to as CTV, have been set to  $\sigma = 1.3$  and  $b = 1$ , while parameters  $\sigma_s = 1.3$  and  $\sigma_d = 2.5$  have been chosen for the edge detector based on the denoiser described in Sect. 3, referred to as TVED. In addition,  $\alpha_k = 1$  for all  $k$ . The efficient implementation proposed in [20] has been used for CTV.

Three ground-truths have been considered in the experiments: the NMSE map generated by the Prewitt's edge detector, a computer generated consensus ground-truth [7] and the hand-made ground-truth of the Berkeley segmentation data set [14]. We will refer to those ground-truths as GT1, GT2 and GT3, respectively. It is important to remark that the validity of GT3 has only been proven in segmentation related applications. We matched every detected edge to its closest pixel in the ground-truth, allowing for up to one match for every ground-truth pixel. Gaussian noise with different standard deviations has been added to the input images for the robustness analysis.

Table 1 shows the performance of the different methods for GT1 and GT2. Evolution plots for the proposed performance measurements are not necessary for GT1 and GT2 since  $M \geq N$  for them.

Regarding GT1, all the tested algorithms have a good precision and false alarm rejection but a poor discriminability. Although TVED has a better performance in discriminability than the others, some applications could require even better results. Only the Sobel detector has a good performance in recall. This result was expected since Prewitt detects significantly more edges than the other tested edge detectors, with the exception of the Sobel detector. TVED is the best method with respect to the other three measurements.

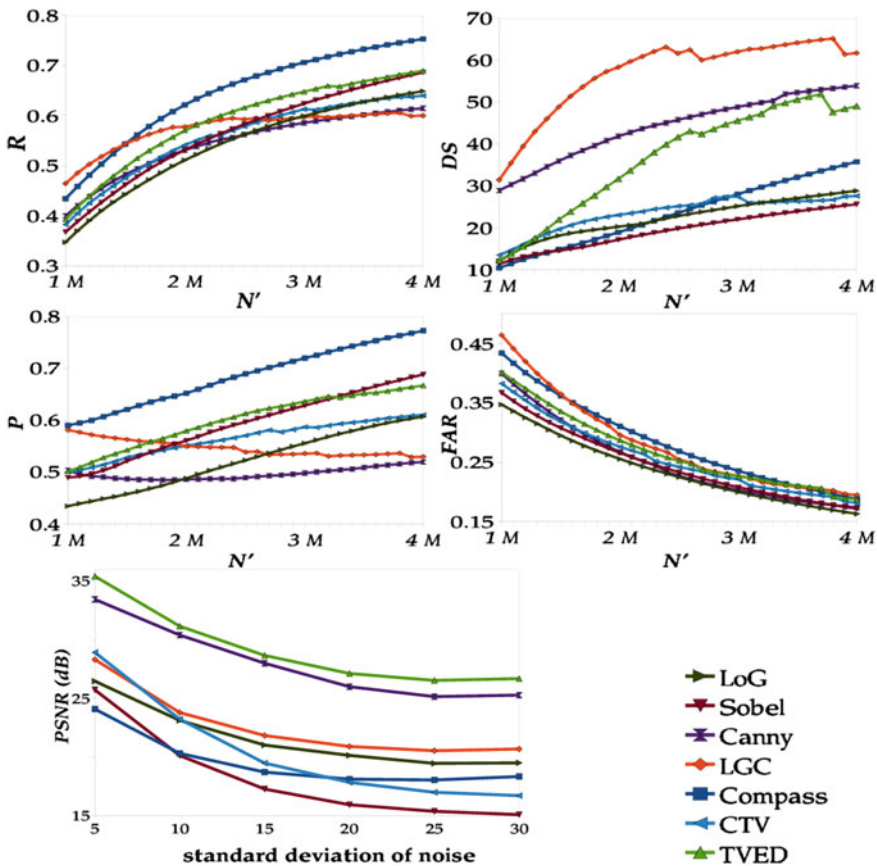
Regarding GT2, LGC is the best algorithm according to discriminability. However, LGC shows a poor performance in recall. On the other hand, CTV is the best in precision and false alarm rejection. However, CTV shows a poor performance in recall. Thus, LGC and CTV relinquish better recall figures for the sake of discriminability, and precision and false alarm rejection respectively. The Sobel detector is the best in recall and has a competitive performance in the other measures. Only LoG

**Table 1** Performance measurements for ground-truths GT1 and GT2. The best performance per column is marked in *bold*

Method	$R$		$DS$		$P$		$FAR$	
	GT1	GT2	GT1	GT2	GT1	GT2	GT1	GT2
LoG	0.44	0.68	9.45	38.81	0.93	0.63	0.90	0.51
Sobel	<b>0.84</b>	<b>0.75</b>	9.89	34.07	0.94	0.88	0.84	0.74
Canny	0.23	0.29	7.98	39.65	0.96	0.79	0.93	0.74
LGC	0.15	0.40	4.46	<b>44.33</b>	0.96	0.85	0.93	0.78
Compass	0.57	0.61	8.62	41.10	0.93	0.71	0.89	0.57
CTV	0.23	0.34	7.17	21.66	<b>0.98</b>	<b>0.92</b>	<b>0.95</b>	<b>0.87</b>
TVED	0.20	0.53	<b>21.24</b>	34.39	<b>0.98</b>	0.75	<b>0.95</b>	0.69

has a  $P$  value below 0.69, which means that it is the only method where the mean distance between the matched points in the ground-truth and the NMSE is above 2 pixels.

Figure 5 shows the evolution of the proposed performance measurements for GT3 with  $N'$  and the robustness analysis. It can be observed that the Compass detector has the best evolution for the  $R$  and  $P$ -measurements, the LCG is the best for the  $DS$ -measurement and both have a similar performance with respect to  $FAR$ . The performance of the Sobel and LoG detectors is the worst, but it increases with  $N'$  even surpassing LGC in  $R$  and  $P$  for large values of  $N'$ . The proposed methods have competitive results, especially TVED. For example, unlike LGC, TVED has an increasing trend with  $N'$ , and outperforms Compass in  $DS$ . TVED is the most consistent method for GT3 as its performance is usually in the top three of the

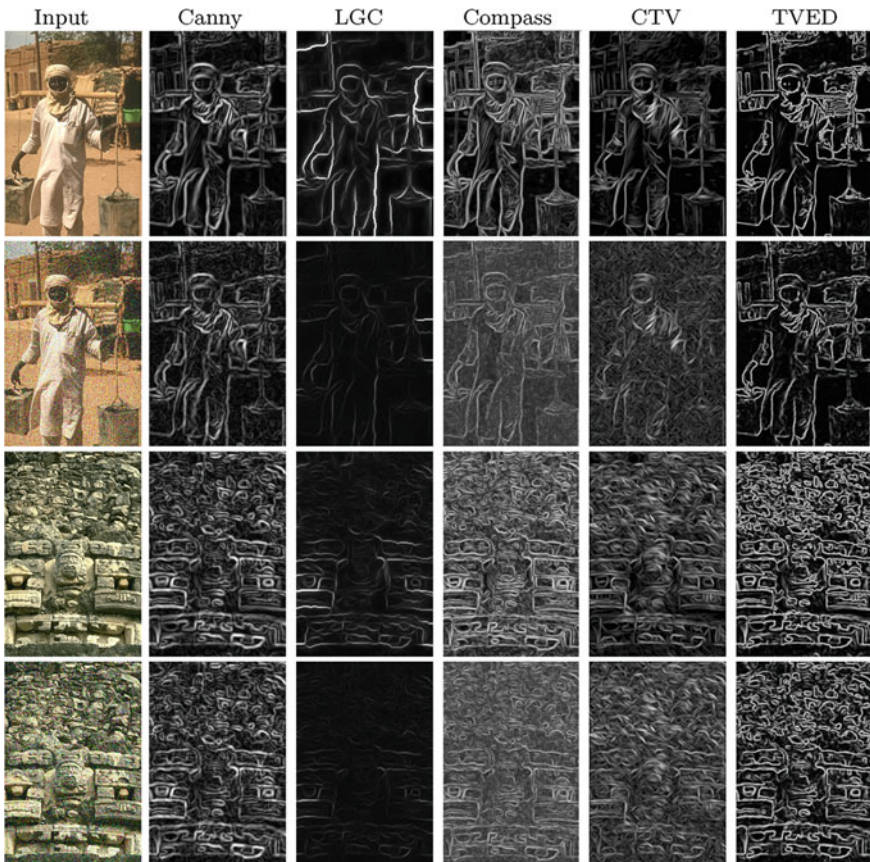


**Fig. 5** Performance measurements for GT3. *Top left:*  $R$  vs.  $N'$  (recall); *top right:*  $DS$  vs.  $N'$  (discriminability); *middle left:*  $P$  vs.  $N'$  (precision); *middle right:*  $FAR$  vs.  $N'$  (false alarm rejection); *bottom left:*  $PSNR$  vs. standard deviation of noise (robustness); *bottom right:* conventions

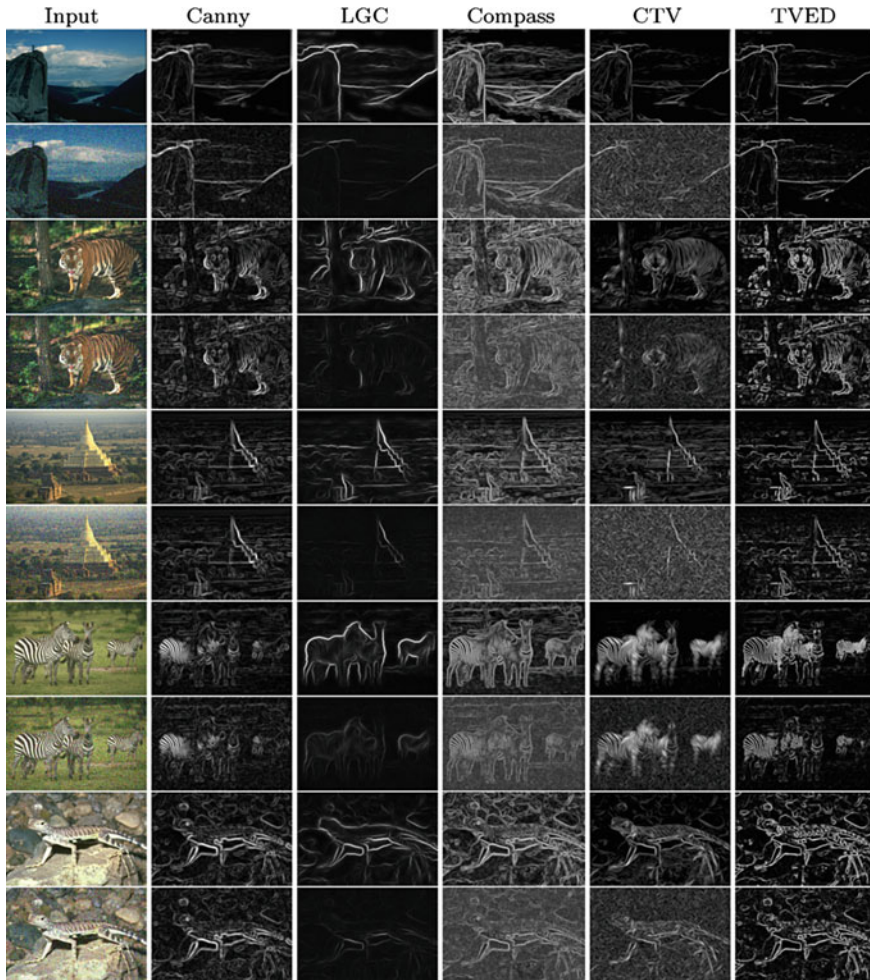
tested methods with respect to these four measurements. Although Canny has been outperformed in all measurements, it is still competitive in  $DS$ .

As for the robustness analysis, original images have been contaminated with additive white Gaussian noise (AWGN) with different standard deviations. TVED appears to be the most robust algorithm with around 1 dB above Canny, and 7dB above LGC. CTV has a good performance with low amounts of noise, but it rapidly decreases due to the appearance of artifacts (cf. Figs. 6 and 7). This could mean that denoising and detecting edges at the same time seems a better alternative than iterating tensor voting in noisy scenarios. The LoG, Sobel and Compass detectors are more sensitive to noise.

A visual comparison can also give noteworthy information of the properties of the tested edge detectors. Figs. 6 and 7 show the edginess maps detected for some of the



**Fig. 6** Visual comparison of results. *First column*: original images and their noisy counterparts. *Columns three to six*: edginess maps generated by the Canny, LGC, Compass, CTV and TVED methods respectively for the corresponding images



**Fig. 7** Visual comparison of results. *First column*: original images and their noisy counterparts. Columns three to six: edginess maps generated by the Canny, LGC, Compass, CTV and TVED methods respectively for the corresponding images

tested images and their noisy counterparts. The noisy images have been generated by adding AWGN with a standard deviation of thirty. This standard deviation of noise aims at simulating very noisy scenarios.

It can be appreciated that LGC generates fewer edges than the others, but also misses some important edges and their strength is reduced for the noisy images. The Compass operator generates too many edges and the number of edges increases with noise. CTV yields good results for noiseless images. However, its performance is largely degraded for noisy images, where undesirable cross-shaped artifacts are generated. This is mainly due to the fact that CTV is more prone to detecting straight lines by mistakenly joining noisy pixels. TVED and Canny have a better behavior,



**Table 2** Examples of selection of edge detector

Application	Feature	Best tested method
Image segmentation	Discriminability	LGC
Image segmentation	Precision	Compass
Image segmentation	All	TVED
Image edge enhancement	Recall	Sobel
Image edge enhancement	Precision	CTV
Any	Robustness	TVED and Canny
Any	Speed	Sobel, Canny and LoG

since they only detect the most important edges and are less influenced by noise. However, TVED generates sharper edges than Canny.

Regarding computational cost, the Sobel detector was the fastest of all tested algorithms when run on an Intel Core 2 Quad Q6600 with a 4GB RAM (0.06 s), followed by Canny (0.15 s), LoG (0.35 s), Compass (around 20 s), CTV (around 25 s), TVED (around 40 s). The slowest detector by far was LGC (2 min and 36 s).

## 6 Concluding Remarks

Two new methods for edge detection based on tensor voting have been presented: the first method based on the classical tensor voting, and the latter based on an adaptation of the tensor voting procedure. The evaluation has been performed by measuring the features of completeness, discriminability, precision and robustness of edge detectors.

Experimental results show that tensor voting is a powerful tool for edge detection. On the one hand, TVED has been found to be more robust than the state-of-the-art methods while having a competitive performance in recall, discriminability, precision, and false alarm rejection with respect to three different ground-truths. TVED was the most consistent of the tested methods for image segmentation since, unlike other methods, it was usually in the top three of the tested methods under all measurements. CTV has demonstrated good properties of edge detection in both noiseless and images with a small amount of noise.

The results also show that it is difficult for an edge detector to have a good performance for all the features and applications. This means that every edge detector has strengths and weaknesses that makes it more suitable for some applications than for others under a specific measure. This fact should be taken into account in order to choose the most appropriate edge detector for every context. For instance, Table 2 shows some examples of which method among the tested methods should be chosen for some particular scenarios.

**Acknowledgments** This research has been supported by the Swedish Research Council under the project VR 2012-3512.

## References

1. Arbelaez P, Maire M, Fowlkes C, Malik J (2011) Contour detection and hierarchical image segmentation. *IEEE Trans Pattern Anal Mach Intell* 33(5):898–916
2. Baker S, Nayar SK (1999) Global measures of coherence for edge detector evaluation. In: *Proceedings of IEEE conference on computer vision and pattern recognition*, pp II:373–379
3. Batard T, Saint-Jean C, Berthier M (2009) A metric approach to nD images edge detection with Clifford algebras. *J Math Imaging Vision* 33(3):296–312
4. Bowyer K, Kranenburg C, Dougherty S (2001) Edge detector evaluation using empirical ROC curves. *Comput Vis Image Underst* 84(1):77–103
5. Canny JF (1986) A computational approach to edge detection. *IEEE Trans Pattern Anal Mach Intell* 8(6):679–698
6. De Micheli E, Caprile B, Ottonello P, Torre V (1989) Localization and noise in edge detection. *IEEE Trans Pattern Anal Mach Intelligence* 11(10):1106–1117
7. Fernández-García N, Carmona-Poyato A, Medina-Carnicer R, Madrid-Cuevas F (2008) Automatic generation of consensus ground truth for the comparison of edge detection techniques. *Image Visual Comput* 26(4):496–511
8. Förstner W (1986) A feature based correspondence algorithm for image matching. *Int Arch Photogrammetry Remote Sens* 26:150–166
9. Heath M, Sarkar S, Sanocki T, Bowyer K (1998) Comparison of edge detectors: a methodology and initial study. *Comput Vis Image Underst* 69(1):38–54
10. Heath M, Sarkar S, Sanocki T, Bowyer KW (1997) A robust visual method for assessing the relative performance of edge-detection algorithms. *IEEE Trans Pattern Anal Mach Intell* 19(12):1338–1359
11. Koschan A (1995) A comparative study on color edge detection. In: *Proceedings of Asian conference on computer vision*, pp 574–578
12. Koschan A, Abidi M (2005) Detection and classification of edges in color images. *IEEE Signal Process Mag* 22(1):64–73
13. Luo MR, Cui G, Rigg B (2001) The development of the CIE 2000 colour-difference formula: CIEDE2000. *Color Res Appl* 26(5):340–350
14. Martin D, Fowlkes C, Tal D, Malik J (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proceedings of IEEE international conference on computer vision*, pp II:416–423
15. Martin DR, Fowlkes CC, Malik J (2004) Learning to detect natural image boundaries using local brightness, color and texture cues. *IEEE Trans Pattern Anal Mach Intell* 26(1):530–549
16. Medioni G, Lee MS Tang CK (2000) *A Computational framework for feature extraction and segmentation*. Elsevier Science, Amsterdam
17. Moreno R, Garcia MA, Puig D, Julià C (2009) On adapting the tensor voting framework to robust color image denoising. In: *Proceedings of international conference on computer analysis of images and patterns*. *Lecture Notes in Computer Science* vol 5702, pp 492–500
18. Moreno R, Garcia MA, Puig D, Julià C (2009) Robust color edge detection through tensor voting. In: *Proceedings of IEEE international conference on image processing*, pp 2153–2156
19. Moreno R, Garcia MA, Puig D, Julià C (2011) Edge-preserving color image denoising through tensor voting. *Comput Vis Image Underst* 115(11):1536–1551
20. Moreno R, Garcia MA, Puig D, Pizarro L, Burgeth B, Weickert J (2011) On improving the efficiency of tensor voting. *IEEE Trans Pattern Anal Mach Intell* 33(11):2215–2228
21. Moreno R, Pizarro L, Burgeth B, Weickert J, Garcia MA, Puig D (2012) Adaptation of tensor voting to image structure estimation. In: *Laidlaw D. and Vilanova, A. (eds) New developments in the visualization and processing of tensor fields*, Springer, pp 29–50
22. Moreno R, Puig D, Julià C, Garcia MA (2009) A new methodology for evaluation of edge detectors. In: *Proceedings of IEEE international conference on image processing*, pp 2157–2160
23. Nguyen TB, Ziou D (2000) Contextual and non-contextual performance evaluation of edge detectors. *Pattern Recogn Lett* 21(9):805–816

24. Papari G, Petkov N (2011) Edge and line oriented contour detection: state of the art. *Image Vision Comput* 29(2–3):79–103
25. Plataniotis K, Venetsanopoulos A (2000) *Color image processing and applications*. Springer, Berlin
26. Pratt WK (2007) *Digital Image Processing: PIKS Scientific Inside*, 4th edn. Wiley-Interscience, California
27. Prieto M, Allen A (2003) A similarity metric for edge images. *IEEE Trans Pattern Anal Mach Intell* 25(10):1265–1273
28. Ruzon M, Tomasi C (2001) Edge, junction, and corner detection using color distributions. *IEEE Trans Pattern Anal Mach Intell* 23(11):1281–1295
29. Shin MC, Goldgof DB, Bowyer KW, Nikiforou S (2001) Comparison of edge detection algorithms using a structure from motion task. *IEEE Trans Syst Man Cybern Part B Cybern* 31(4):589–601
30. Smolka B, Venetsanopoulos A (2006) Noise reduction and edge detection in color images. In: Lukac R, Plataniotis KN (eds) *Color image processing: methods and applications*, CRC Press, pp 88–120
31. Spreeuwens LJ, van der Heijden F (1992) Evaluation of edge detectors using average risk. In: *Proceedings of international conference on pattern recognition*, vol 3, pp 771–774
32. Xue-Wei L, Xin-Rong Z (2008) A perceptual color edge detection algorithm. In: *Proceedings of international conference on computer science and software engineering*, vol 1, pp 297–300
33. Zhu SY, Plataniotis KN, Venetsanopoulos AN (1999) Comprehensive analysis of edge detection in color image processing. *Opt Eng* 38(4):612–625

# Color Categorization Models for Color Image Segmentation

Teresa Alarcon and Oscar Dalmau

**Abstract** In 1969, Brent Berlin and Paul Kay presented a classic study of color naming where experimentally demonstrated that all languages share a universal color assignment system of 11 basic color categories. Based on this work, new color categorization models have appeared in order to confirm this theory. Some of these models assign one category to each color in a certain color space, while other models assign a degree of membership to each category. The degree of membership can be interpreted as the probability of a color to belong to a color category. In the first part of this work we review some color categorization models: discrete and fuzzy based models. Then, we pay special attention to a recent color categorization model that provides a probabilistic partition of a color space, which was proposed by Alarcon and Marroquin in 2009. The proposal combines the color categorization model with a probabilistic segmentation algorithm and also generalizes the probabilistic segmentation algorithm so that one can include interaction between categories. We present some experiments of color image segmentation and applications of color image segmentation to image and video recolourization and tracking.

**Keywords** Soft segmentation, Probabilistic segmentation, Color image segmentation, Color categorization model, Universal color categories, Color interaction modeling, Bayesian technique, Video segmentation, Tracking, Colorization

---

T. Alarcon (✉)

Centro Universitario de los Valles, Universidad de Guadalajara, Carretera  
Guadalajara–Ameca Km. 45.5, 46600 Ameca, JAL, México  
e-mail: teresa.alarcon@profesores.valles.udg.mx

O. Dalmau

Centro de Investigación en Matemáticas, A.C., Jalisco S/N, Col. Valenciana,  
36240 Guanajuato, México

# 1 Introduction

In this work we investigate and discuss some methods of color image segmentation derived from color categorization models. The categorization models obtained from linguistic labels are a consequence of human perception. It is known that humans express the interpretation of the color stimulus by means of names or categories. In 1969, Berlin and Kay experimentally demonstrated that all languages share a universal color assignment system of 11 basic universal color categories. Inspired by this research, some color categorization models have been proposed. These kind of models can be used in many computer vision tasks, such as color image segmentation, among other applications. Additionally, we reflect on color categorization models obtained from two kinds of color naming experiments, namely: in real world images and on chip-based color naming. Based on these experiments, a color categorization model is generated in a color space. Furthermore, we study some color modifiers that allow us to obtain a more detailed description of the color name. We have also extended the categorization model so that it can use any number of color categories, this could be useful in some applications. Once the model is built, it can be used for segmentation purposes. We analyze different segmentation approaches based on color categorization models, including those with and without spatial coherence. A principled and effective way to take into account the spatial structure of segmentation is based on the theory of Markov Random Fields (MRFs) together with Bayesian estimation. Its success stems from the fact that the solutions obtained are similar to the ones produced by the perception of humans. A description of this kind of segmentation method combined with a color categorization model is given. In this study we also reflect on the use of perceptual color interactions described by Boynton and Olson in 1987 in the segmentation process. An interesting result of the combination between color categorization models with segmentation is the obtained edge map with perceptually salient borders in the segmented image. Applications of the color categorization model for image editing are presented, in particular for image and video recolourization.

The structure of this work is the following. Section 2 presents a review of research about *color naming* and the *basic color categories*. In Sect. 3 we present some *Color Categorization Models*. In the last part of this section we present the Alarcon and Marroquin model and in Sect. 4 its application to color image segmentation. Here we present two cases. When the image is composed by homogeneous regions we can directly apply the color categorization model to segment the image. However, in more complex scenes the segmented image could be very granular, for this case we present a probabilistic Markov Random Field segmentation algorithm. Finally in Sect. 5, applications of the Alarcon and Marroquin Color Categorization Model to *Detection of perceptually salient edges*, *Video Tracking* and *Image and Video Recolourization* are presented.

## 2 Color Names and Basic Color Categories

The interpretation of color stimulus expresses itself by means of names or categories. In 1969, Berlin and Kay [4], through an experimental study of 20 languages and the research of color names in 78 other languages, presented evidence indicating that all languages share a universal color assignment system. The participants of the experiment first defined the colors considered by them as basics. After that, they observed 329 color patches using the Munsell color system [26]. Each observer selected the focal point (the best prototype of each basic color or category) and points in the border of each found color group, i.e., the observer selected other color samples similar to the focal point for each color group. The result of this experiment is illustrated in [4].

Based on this research, they arrived at the conclusion that there are eleven basic universal color categories: *black, white, gray, red, green, yellow, blue, brown, pink, orange* and *purple*. The first three categories are achromatic and the remainder are chromatic. The most important characteristics of these color categories are the following:

- They are monolexemic.
- They can be used for describing not only one unique object (for example: objects in lemon) but also multiple objects (the word ‘green’ allows us to describe several objects in green, including those in lemon).
- They represent psychological information salients, and they are reference points for describing any color.

The results obtained by Berlin and Kay are the starting point for the research in the field of color categorization.

McDaniel in 1972 [15] and Kay and McDaniel in 1978 [12] investigated the relation between neurofunctional mechanisms during color assignment and linguistic terms used by humans. In this research they concluded that the universal color categories are inherent to human perception since color vision is a result of a neurophysiological process which is common to everyone.

In 1987, Boynton and Olson [5] established the difference between the eleven basic color terms [4] and non-basic color terms (for example: salmon, cyan, violet). For this, they designed an experiment in which 424 color samples were randomly presented to seven subjects. Six of them could use both basic and non-basic color terms, with the restriction of naming each color using only monolexemic color terms, i.e., they could use neither compound terms (blue-green) nor intensity modifiers (light, dark). Boynton and Olson also defined the localization of the eleven basic colors in the OSA space [5]. The most important conclusions are the following:

- Color categories represent regions in the color space.
- There are consensus zones defined as regions in which all subjects assigned the same color term.
- Consensus zones coincide with those associated with the eleven basic color categories.

- Consensus zones associated with green and blue are in almost all intensity levels of the space. Blue and green categories occupy the largest area in the chromatic plane.
- Some colors can be described for more than one category. These colors belong to transition zones between color categories. This was revealed in the experiment when some colors were assigned to different color categories for the majority of subjects. Such color categories are considered to be perceptually linked, otherwise they are described as unlinked.

Sturges and Whitfield in 1995 [21] extended Boynton and Olson's work. They carried out the same experiments provided in [5] but in the Munsell space. In order to avoid the drawback of assigning a color category among zones of yellow, brown, pink, white and orange, Sturges and Whitfield proposed a new category: beige. In the experiment they obtained the same Boynton and Olson's results and additionally arrived at the following conclusions:

- There is no significant difference between men and women with respect to the number of basic and non-basic color terms used.
- They state that the use of the non basic color terms beige and cream could solve the drawback of naming zones among white, yellow, pink and brown.
- Munsell color space provides a better color representation compared with the OSA space.

Although, in the Munsell system the results are better than in the OSA space, from the authors' point of view all color systems have limitations to model the complexity of human color perception. In the experiments of previous works, the authors also noted that the time response of subjects when they assign a color name to a point in a color space is different. For example blue, green, red and yellow have a faster response than purple, orange and brown, although the difference is small.

In summary, in [5, 21] the research is devoted to locating color basic categories in a color space. The color name assignment of each point in the space is discrete, i.e., one point can belong to only one category, which is a disadvantage. Another limitation is that the authors do not provide an implementation of their proposal for applications.

### 3 Color Categorization Models

Color categorization is intrinsically related to color naming. A color categorization model is a mathematical representation of the color naming process. According to the previous section, color naming refers to the labeling, in a given space, of a set of stimuli in certain conditions.

In 1978, McDaniel and Kay [12] proposed a model to explain the color naming process. In that model they established a relation between neurofunctional mechanisms presented in the color naming process and the used linguistic color terms,

through a fuzzy set theory. Two of the most important ideas discussed in that work are the following:

- Neurophysiological processes that explain the color categorization are based on continuous functions [25], so the discrete formalism of color categorization is not appropriate. Expressions as *light blue* or *dark red* are typical for any language and demonstrate that humans assign color categories taking into consideration a degree of membership.
- The regions located by Berlin and Kay can be interpreted as those where the membership function reached the maximum value.

The fuzzy representation of color categories, given by the authors in [12], is formed from the 4 fundamental spectral responses of the human vision system: red, green, yellow and blue. The spectral responses are obtained by Wooten [25] through experiments with human observers.

The points of maximum response, in the spectral curves, for the blue, red, yellow and green in [25] have the same wavelength as the focal points for blue, red, yellow and green categories found by Berlin and Kay, see details in [25]. The novelty of the model proposed by McDaniel and Kay is the use of fuzzy theory, however, the authors only considered 4 fuzzy sets and it is not clear how to define the membership functions for the remaining categories and how to train the model to include new categories.

The work done by Cairo [7] is another example of color categorization model based on neurofunctional mechanisms of the color naming process. Cairo proposes a model with four physical quantities: wavelength, intensity, purity and the adaptation state of the retina. The model represents each of the eleven color categories as a combination of the four mentioned quantities. Besides, this author proposes four new categories: *blue sky*, *turquoise*, *lemon* and *khaki*. Although the model is interesting because of the included physical quantities, the use of physical parameters leads to a hard implementation as in the McDaniel and Kay [12] model.

In 1994, Lammens [14] proposed the first parametrical model to describe the color naming process in different color spaces. Lammens model represents each of the eleven basic color categories given by Berlin and Kay [4] through a function, whose parameters are computed using data provided in [4]. Below we comment on the most important details of the research:

- A new color space is created and is named neuropsychophysical space (NPP), due to the use of neurophysiological experiments published by De Valois et al. [23]. The research in [23] allows us to begin to understand human color perception and was done through a study of monkey brain behavior during color perception.
- The data obtained by Berlin and Kay [4] (focal points and regions for each color category) in Munsell space, are located in the NPP space.
- For modeling each basic color category the Gaussian function is selected. The choice is justified by the investigation done by Shepard in 1987 [20] in Psychology. Human perception leads to the categorization through linguistic labels. Shepard argues that the probability of generalization of an existing category (known) to a



new stimulus (unknown) is a monotonic function of the normalized distance in psychological space of the unknown stimulus to known stimuli belonging to the category [14]. Because of the previous statement, Shepard [20] specifies that the described function can be approximated by a simple exponential decay or, under certain circumstances, by a Gaussian function. The Euclidean distance can be used as a distance metric [14, 20]. Then, the model proposed, by Lammens [14], for each category is given by Eq. (1):

$$G_k(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}_k\|^2}{2\sigma_k^2}\right), \quad k = 1, 2, \dots, 11. \quad (1)$$

In the Eq. (1)  $\boldsymbol{\mu}_k$  indicates the localization of the center or focal point for each  $k$  category;  $\sigma_k$  controls the volume of color space that is included in each  $k$  category and the function value  $G_k(\mathbf{x})$  determines the membership of the color stimuli  $\mathbf{x}$  to a  $k$  category. The  $\boldsymbol{\mu}_k$  and  $\sigma_k$  values are computed through a minimization of a functional described in [14] for each category, using the color points of the boundaries and the foci of the regions obtained in [4]. Once the parameters are known, one can assign a color category for any arbitrary color stimuli  $\mathbf{x}$  as follows:

1. Compute the model function  $G_k(\mathbf{x})$  for  $k = 1, 2, \dots, 11$ .
2. Assign to the color stimuli  $\mathbf{x}$  the color category or index that maximizes  $G_k(\mathbf{x})$ .

The Lammens color categorization model is constructed not only in NPP color space, but also in the  $XYZ$  and  $Lab$  spaces. This proposal does not consider intensity and saturation modifiers and despite of its novelty, it has not been extensively applied yet.

Another interesting approach about color naming and description of color composition of the scene is given by [17]. The computational model for color naming done by research in [17] adopted the ISCC-NBS dictionary [13] because this dictionary allows to carry out controlled perceptual experiments and includes basic color terms defined by Berlin and Kay [4]. In the color naming experiment 10 subjects participated and four experiments were done:

1. Color Listing Experiment: it is aimed at testing 11 basic color categories from Berlin and Kay's investigation. Each subject was asked to name at least twelve colors in order to test the relevance of color terms not included in basic ones.
2. Color composition experiment: the aim of this experiment is to determine the vocabulary used in describing complex color scenes. The participants observed 40 photographic images in a sequence and they described the color composition of each image under the restriction to use common color terms, common modifiers of brightness and saturation and to avoid rare color names (for example, names derived from objects or materials).
3. Two Color-Naming Experiments: Each subject observed 267 centroid colors from the ISCC-NBS color dictionary and assigned a color name. In the first experiment  $64 \times 64$  pixel color patches were arranged into a  $9 \times 6$  matrix and were displayed to

the subjects. The light gray was the display background. In the second experiment only one  $200 \times 200$  pixel color patch was observed.

From these experiments, in [17] arrived at the following conclusions:

- None of the subjects listed more than 14 color names during Color Listing Experiment. The eleven basic colors specified by Berlin and Kay were found in the color list. Beige, violet and cyan were also included by some of the subjects with less frequency. Modifiers for hue, saturation and luminance were not used for this color listing experiment. This stage in color naming was described as *fundamental level*: the color names are expressed using only generic hue or generic achromatic terms.
- The subjects used almost similar vocabulary to describe image color composition in the Color Composition Experiment. Modifiers for hue, saturation and luminance were used to distinguish different types of the same hue. Although the images had rich color histograms, no more than 10 names were included in the color list. In this case, color naming can be classified in the *coarse level*, in which the color names are expressed using luminance and generic hue or luminance and generic achromatic term. Another level presented in the Color Composition Experiment was the *medium level*: color names are described adding the saturation modifiers to the coarse description.
- The results obtained in the two color naming experiments were almost identical. The difference is explained by the use of different luminance modifiers. The same color is described by a different luminance modifier when it is displayed in a small and in a large window, i.e., the size of the color patches influences in the color naming decision by a subject. For this experiment the author classified the color vocabulary as *minute level*, in which the complete color information is given.
- In general, well-defined color regions are easier to describe than dark regions, which, in general, exist due to shadows or due to illumination problems.
- All experimentation confirmed that not all color terms included in the ISCC-NBS dictionary are well understood by the general public. For that reason Mojsilovic decided to use all prototype colors specified in ISCC-NBS dictionary [17], except those that were not perceived by the participants of the experiments. On the other hand, in order to reflect the participant decisions, some of the color names were changed. With all these considerations, a new color vocabulary was created in [17] for describing the color naming process.

A very interesting issue of the research is that they designed a metric in order to know the similarity of any arbitrary color point  $\mathbf{c}_x$  with respect to a color prototype  $\mathbf{c}_p$ . The new metric, which is based on the findings from the experiment, is given by the following equation:

$$D(\mathbf{c}_p, \mathbf{c}_x) = D_{Lab}(\mathbf{c}_p, \mathbf{c}_x)[1 + k(D_{Lab}(\mathbf{c}_p, \mathbf{c}_x))\Delta d(\mathbf{c}_p, \mathbf{c}_x)], \quad (2)$$

where  $\Delta d(\mathbf{c}_p, \mathbf{c}_x)$  is a distance function in the *HSL* color space proposed in [17] and  $D_{Lab}$  is a distance in the *Lab* color space

$$D_{Lab}(\mathbf{c}_p, \mathbf{c}_x) = \sqrt{(l_{c_p} - l_{c_x})^2 + (a_{c_p} - a_{c_x})^2 + (b_{c_p} - b_{c_x})^2}, \quad (3)$$

where  $l_c, a_c, b_c$  represent the components  $l, a, b$  of the color  $\mathbf{c}$  in the  $Lab$  color space.

In Eq. (2),  $k(\cdot)$  is a function which is introduced with the aim to avoid modifying distances between very close points in  $Lab$  space and to control the amount of increase for large distance  $D_{Lab}(\cdot, \cdot)$ , the factor  $k(\cdot)$  is defined as:  $k(t) = 0$ , if  $t < 7$ ,  $k(t) = const$ , if  $t > 30$  [17]. Observe that the Eq. (2) is a combination of distances of two spaces:  $Lab$  and  $HSL$ .

The author remarks that any other distance function that satisfies the requirements for the color-naming metric can be used, for example, CMC or dE2000 color difference metric, that explains the non-uniformity of the  $Lab$  space. For validation purposes, the metric was compared with human observations arriving at 91 % agreement.

In [17] Mojsilovic proposed the application of the model to describe the color composition of the image after the color segmentation process through the *Mean Shift algorithm* [10].

The experimentation confirms that the human visual system performs an spatial average, which depends on color frequencies, interactions between colors, observed object size and global context. For instance: we perceived one color for uniform color regions; pixels labeled as color edge and texture edge are not averaged; edge density determines the amount of averaging performed in the textured areas: fine texture (more edge density) has more color averaging than coarse ones (less edge density). Therefore, the human color perception can be interpreted as an adaptive low-pass filter, which must be considered for any application related to human color perception [17].

Robert Benavente et al. [2] proposed a computational model based on the idea proposed by Kay and McDaniel [12]. They pointed out that color naming is a fuzzy decision. Taking into consideration this postulate, the best way to model the color naming is through the fuzzy set theory. Therefore, each color category is defined as a fuzzy set. The color naming experiment in [2] was carried out as follows:

- 10 subjects observed 422 color samples and they were asked to distribute 10 points among the 11 basic color categories taking in consideration the grade of belonging of the observed color sample to each of the basic color terms. When the subject was absolutely sure about the sample color name, 10 points were assigned to the corresponding category. The restriction for using only 11 color terms is justified by research in [5, 21].
- For each sample, the scores were averaged and normalized to the  $[0, 1]$  interval. The experiment was performed twice for each subject. As a result of the color naming process, an experimental color descriptor,  $CD(\mathbf{x})$ , is obtained for each sample:

$$CD(\mathbf{x}) = [m_1(\mathbf{x}), m_2(\mathbf{x}), \dots, m_{11}(\mathbf{x})], \quad (4)$$

where  $m_k(\mathbf{x}) \in [0, 1]$  and  $\sum_k m_k(\mathbf{x}) = 1$ .  $m_k(\mathbf{x})$  can be interpreted as the degree of membership of  $\mathbf{x}$  to the color category  $k = 1, 2, \dots, 11$ .

The results of the color naming experiment define the training set to find membership functions  $f_k(\mathbf{x}^j, \theta_k)$  of the color fuzzy sets. In order to know the parameters,  $\theta_k$ , of each membership function (model for each color category), the minimization of the following functional

$$\min_{\theta_k} \frac{1}{2} \sum_{j=1}^J \left( f_k(\mathbf{x}^j, \theta_k) - m_k(\mathbf{x}^j) \right)^2, \quad \forall k = 1, 2, \dots, 11; \quad (5)$$

is carried out, see [2]. The previous functional, Eq. (5), represents the mean squared error between the membership values of the model  $f_k(\mathbf{x}^j, \theta_k)$  and the  $k$ -th component  $m_k(\mathbf{x}^j)$  of the color descriptor obtained in the color naming experiment, Eq. (4);  $\theta_k$  is the set of parameters of the model,  $J$  is the number of samples in the training set and  $\mathbf{x}^j$  is the  $j$ -th color sample of the training set. According to [2] the model  $f_k(\mathbf{x}^j, \theta_k)$  depends on the type of the color category (chromatic or achromatic).

In [2], the authors created a color space named *uvI*, whose first two components represent the color components and the third one the intensity. They carried out the experimental work in the *uvI* space and for comparison purposes they also employed the *Lab* space. From the color naming data and data fitting process through an optimization algorithm they concluded that the Gaussian function is a good model, as a membership function, for achromatic categories. However, they found that the combination of sigmoid and Gaussian functions is an appropriate model for chromatic categories. They also concluded that the results obtained in the *Lab* space were no better than those obtained in *uvI* space.

Seaborn et al. [19] also proposed a computational model based on fuzzy sets and considered the eleven color basic terms. The research was done in the Munsell space and the data were taken from Sturges et al. [21] investigation. As in [2] they made a distinction between membership function for achromatic and membership function for chromatic classes.

They proposed a similarity measurement based on Tversky research [22]. This measurement is compared with Euclidean distance and they conclude that the new proposal achieved a higher agreement with human judge. They compared the new similarity measurement in the *RGB*, *Lab*, *Luv* and *HSV* color spaces and concluded that the *HSV* space had the best performance to detect similarity and difference between colors.

Joost van de Weijer and et al. [24] elaborated a model to learn color names from real world images, through a Probabilistic Latent Semantic Analysis (PLSA), a generative model introduced by Hofmann [11] for document analysis. In the research, only the eleven color basic categories defined by Berlin and Kay are considered. The data color samples are extracted from <http://lear.inrialpes.fr/data> (data here were collected by the authors), from the auction website Ebay, from Google and from those published by Benavente et al. [3]. The latter data were taken in order to compare the color assignment in real world images with a chip-based approach described in [2]. Three color spaces were used: *RGB*, *HSL* and *Lab*. Two ways to assign color names to individual pixel are considered [24]:

1. *PLSA – bg* model: It is based only on the pixel value. The model is expressed as follows:

$$P(z|w) \propto P(z)P(w|z), \quad (6)$$

where  $P(z|w)$  represents the probability of a color name  $z$  (color category) given a pixel  $w$ . The prior probability over the color names is taken to be uniform [24].

2. *PLSA – bg\** model: It takes into account the pixel  $w$  and some region  $d$ , around it. The Equation for the model is the following:

$$P(z|w, d) \propto P(w|z)P(z|d), \quad (7)$$

where  $P(z|w, d)$  represents the probability of a color name  $z$  (color category) given a pixel  $w$  in a region  $d$ . The  $P(z|d)$  is estimated using an EM algorithm [11].

The experimental work considers data in three color spaces: *RGB*, *HLS* and *Lab*. According to the investigation, the *Lab* space slightly outperformed the others. Besides the PLSA method, the authors employed the Support Vector Machine algorithm [6].

Menegaz et al. [16] presented a computational model obtained by a linear interpolation of a three dimensional Delaunay triangulation of the *Lab* color space.

Alarcon and Marroquin's color categorization model [1], assigns to each voxel  $\mathbf{v}$ , with components  $(x, y, z)$ , in a given color space,  $\Omega$ , a discrete probability distribution  $\mathbf{l}(\mathbf{v}) = [l_1(\mathbf{v}), \dots, l_K(\mathbf{v})]^T$  where the component  $l_k(\mathbf{v})$  is interpreted as the probability of  $\mathbf{v} \in \Omega$  to belong to the color category  $k$ ,  $\mathcal{C}_k$ , given the color  $\mathbf{c}$  at  $\mathbf{v}$ , denoted here as  $\mathbf{c}(\mathbf{v})$ . Note that, the components of  $\mathbf{v}$  correspond to the color  $\mathbf{c}(\mathbf{v})$ , i.e.,  $\mathbf{c}(\mathbf{v}) = \mathbf{v}$ . Then,  $l_k(\mathbf{v})$  is defined as follows:

$$l_k(\mathbf{v}) \stackrel{def}{=} P(\mathbf{v} \in \mathcal{C}_k | \mathbf{c}(\mathbf{v})), \quad (8)$$

and represents the *posterior probability* of a voxel to belong to a color category. In [1] each category  $k$  is modeled as a linear combination of 3D quadratic splines

$$l_k(\mathbf{v}) = \sum_{j=1}^N \alpha_{kj} \beta_j(\mathbf{v}), \quad \forall k \in \mathcal{K} \stackrel{def}{=} \{1, 2, \dots, K\}, \quad \mathbf{v} \in \Omega, \quad (9)$$

where  $\beta_j(\mathbf{v}) = \beta(\frac{x-x_j}{\Delta_x})\beta(\frac{y-y_j}{\Delta_y})\beta(\frac{z-z_j}{\Delta_z})$  are located in a node lattice in  $\Omega$ ,  $N$  is the number of nodes in the lattice;  $\Delta_x$ ,  $\Delta_y$  and  $\Delta_z$  define the resolution of the node lattice;  $x_j$ ,  $y_j$ , and  $z_j$  denote the coordinates of the  $j$ -th node,  $\alpha_{kj}$  is the contribution of each spline function  $\beta_j(\cdot)$  to each category  $k$  and  $\beta(\cdot)$  is the quadratic basis function defined in the following equation:

$$\beta(x) = \begin{cases} \frac{1}{2}(-2x^2 + 1.5), & |x| \in \left[0, \frac{1}{2}\right]; \\ \frac{1}{2}(x^2 - 3|x| + 2.25), & |x| \in \left[\frac{1}{2}, 1.5\right]; \\ 0, & |x| \geq 1.5. \end{cases} \quad (10)$$

In order to determine  $l_k(\mathbf{v})$  in (9) one needs to compute the parameters  $\alpha_{kj}$ . Hence, the authors propose to minimize the following functional:

$$\min_{\alpha_k} \sum_{\mathbf{v} \in \mathcal{D}} \left[ l_k(\mathbf{v}) - \sum_{j=1}^N \alpha_{kj} \beta_j(\mathbf{v}) \right]^2 + \tau \sum_{\langle m, n \rangle} (\alpha_{km} - \alpha_{kn})^2, \quad \forall k \in \mathcal{K}, \quad (11)$$

where  $\alpha_k = [\alpha_{ki}]_{i=1, \dots, N}^T$ ,  $\mathcal{D} \subset \Omega$  represents a voxel set for which  $l_k(\cdot)$  is known and  $\langle \cdot, \cdot \rangle$  denotes a pair of neighboring splines. In this case,  $l_k(\cdot)$  is obtained, for all voxels in  $\mathcal{D}$ , through a color naming experiment based on isolated color patches. The second term in (11) controls the smoothness between spline coefficients,  $\tau > 0$  is a parameter that controls the smoothness level.

The solution of the optimization problem (11) is computed by calculating the partial derivatives with respect to  $\alpha_{ki}$  and setting the derivatives equal to zero, i.e.,

$$- \sum_{\mathbf{v} \in \mathcal{D}} \left[ l_k(\mathbf{v}) - \sum_{j=1}^N \alpha_{kj} \beta_j(\mathbf{v}) \right] \beta_i(\mathbf{v}) + \tau \sum_{m \in \mathcal{N}_i} (\alpha_{ki} - \alpha_{km}) = 0, \quad (12)$$

$|\mathcal{N}_i|$  is the cardinality of the neighboring nodes for the  $i$ -th spline. Equation (12) yields the following system of linear equations:

$$\mathbf{A} \alpha_k = \mathbf{b}_k, \quad (13)$$

where  $\mathbf{A} = [a_{ij}]_{i, j=1, \dots, N}$ ,  $\mathbf{b}_k = [b_{ki}]_{i=1, \dots, N}^T$  and

$$a_{ii} = \sum_{\mathbf{v} \in \mathcal{D}} \beta_i^2(\mathbf{v}) + \tau |\mathcal{N}_i|, \quad (14)$$

$$a_{ij} = \sum_{\mathbf{v} \in \mathcal{D}} \beta_i(\mathbf{v}) \beta_j(\mathbf{v}) - \chi_{\mathcal{N}_i}(j) \tau, \quad i \neq j, \quad (15)$$

$$b_{ki} = \sum_{\mathbf{v} \in \mathcal{D}} l_k(\mathbf{v}) \beta_i(\mathbf{v}). \quad (16)$$

In Eq. (15),  $\chi_{\mathcal{N}_i}(j)$  is the indicator function of the subset  $\mathcal{N}_i$ , then  $\chi_{\mathcal{N}_i}(j) = 1$  if  $j \in \mathcal{N}_i$  and  $\chi_{\mathcal{N}_i}(j) = 0$  if  $j \notin \mathcal{N}_i$ .

After solving the linear system (13) for  $k \in \{1, 2, \dots, K\}$  we can compute  $\mathbf{I}(\mathbf{v})$  for all  $\mathbf{v} \in \Omega$ . This provides a ‘soft’ segmentation of the color space. In particular, one

can obtain a partition  $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_K\}$  (a ‘hard’ segmentation) of the color space by finding the component that maximizes  $l_k(\mathbf{v})$ , i.e., by solving the optimization problem:

$$k^* = \arg \max_{k \in \mathcal{K}} l_k(\mathbf{v}), \text{ for all } \mathbf{v} \in \Omega, \quad (17)$$

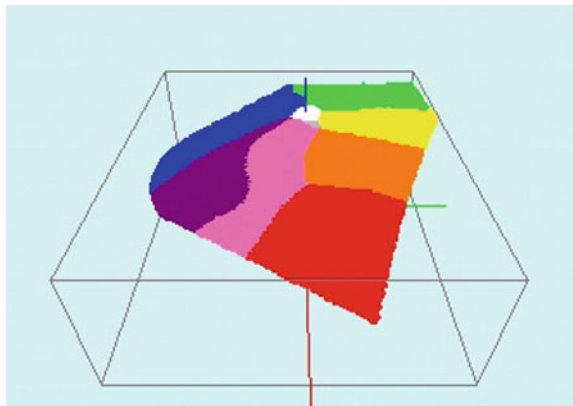
and assigning the corresponding voxel  $\mathbf{v}$  to the region  $\mathcal{R}_{k^*}$ . We note that this model is very general and accepts any number of color categories. The inclusion of more or less categories depend on the selected color vocabulary in the color naming experiment. On the other hand, it is very easy to implement because one only needs to solve a linear system of equations for each category. Also note that the matrix  $\mathbf{A}$  does not depend on the category  $k$ , then one computes the inverse of  $\mathbf{A}$  only once. Alarcon and Marroquin elaborated the described color categorization model in *RGB*, *Lab* and *Luv* color spaces. Figure 1 shows the obtained *Luv* partition (‘hard’ segmentation of the space).

### 3.1 Perceptual Color Experiments

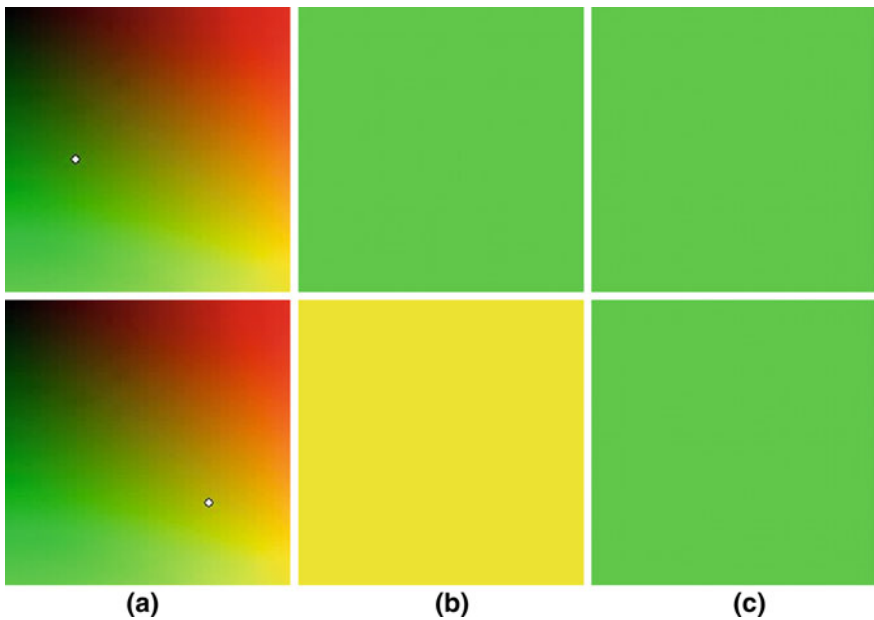
An important step to construct a color categorization model is the design of the color naming experiment. From Sect. 3 it is understood that there are two kinds of color naming experiments:

1. Color naming experiments based on real world images (with contextual color information).
2. Color naming experiment based on isolated color chips (without contextual color information).

**Fig. 1** Partition of the *Luv* space, using the color categorization model by Alarcon and Marroquin



In [24] the authors pointed out that color naming models based on isolated color chips are time consuming due to the necessity of collecting data. In addition, it is unclear the extension of this kind of models to color naming in real color images when the setup conditions are uncontrolled. While these arguments are certainly reasonable, there is an important aspect that must be considered and it was examined in [1]. Alarcon and Marroquin corroborated the influence of the contextual information during the color naming process, especially in those color regions that can be described by more than one category. In terms of information theory, it means that the entropy in these regions is high and the color assignment decision for different subjects is different. This issue, along with the fact of doing an independent color naming experiment using the selected color images, explains why authors in [1] decided to use the color naming experiment based on isolated color chips. Figure 2 illustrates the research in [1] with respect to both kinds of color naming experiments. Observe that for a color point located outside transition zones the subject decision is the same. However, for color points located at transition zones the answer by the same subject is different in both kinds of color naming experiments.



**Fig. 2** **a** A color sample in the *RGB* color plane, **b** a color name decision by a subject when the color naming experiment considered the contextual information, **c** a color name decision by a subject in the color naming experiment without contextual information



## 4 Segmentation Based on Color Categorization Model

Color categorization models reflect the color description used by humans. Therefore they can be used for extracting perceptual color information in the image. In this section we describe the segmentation algorithm proposed in [1]. This algorithm allows to obtain a compact and meaningful color description of the image.

First, we introduce the notation used in this section. Let  $g : \mathcal{L} \rightarrow \mathbb{R}^3$  be the color image to be segmented, where  $\mathcal{L}$  is the corresponding lattice. Let  $\mathcal{N}_r \subset \mathcal{L}$  be a neighborhood of pixels centered at pixel  $r$ , here we use the following neighborhood  $\mathcal{N}_r = \{s \in \mathcal{L} : \|s - r\| \leq 1, s \neq r\}$  where  $\|\cdot\|$  is the Euclidean norm.  $|\mathcal{N}_r|$  represents the cardinality of the neighborhood at pixel  $r$ . The symbol  $\langle \cdot, \cdot \rangle$  denotes a pair of neighboring pixels.

The label set, for the image segmentation, is denoted similar to previous sections as  $\mathcal{K} = \{1, 2, \dots, K\}$ . In this section, we also assume that one has certain information about the confidence of pixels of the image to belong to a class, or color category. The confidence of pixel  $r$  to belong to  $k$  class is denoted by  $v_{kr}$  and satisfies that  $\sum_k v_{kr} = 1$  and  $v_{kr} > 0$ .

### 4.1 Direct Label Assignment

If the image  $g$  is composed by homogeneous regions then the application of the color categorization model proposed by Alarcon and Marroquin is direct. In this case, the grade of membership of pixels in  $g$  to belong to a color category is computed with

$$v_{kr} = l_k(g(r)), \quad (18)$$

where  $l_k(\cdot)$  is the Alarcon and Marroquin model provided in Eq. (9). In order to obtain the ‘hard’ segmentation, i.e., to assign one category per pixel, one needs to solve the following optimization problem:

$$k^* = \arg \max_k v_{kr}, \quad \forall r \in \mathcal{L}. \quad (19)$$

That is, one assigns for all pixels in the image the color category that maximizes the degree of membership  $v_{kr}$ . Figure 3 depicts some color image segmentations using the direct label assignment based on Alarcon and Marroquin’s model.

There are images where the direct label assignment does not produce good segmentation results, and the segmentation could be very granular or too noisy. In order to solve this drawback one can use a more sophisticated segmentation model that takes into account the local information and promotes smooth solutions. This can be done using a probabilistic segmentation approach that assumes the segmentation as a Markov Random Field, see Sect. 4.2.



**Fig. 3** Direct Label Assignment using Alarcon and Marroquin's model. First row: Original image, second row: segmented image

## 4.2 Label Assignment Using Markov Random Field

Alarcon and Marroquin in [1] proposed a generalization of the Entropy controlled Gauss-Markov Random Measure Field model presented by Rivera et al. [18]:

$$\begin{aligned} \min_{\mathbf{p}, \theta} U(\mathbf{p}, \theta) = & - \sum_{r \in \mathcal{L}} \sum_{k \in \mathcal{K}} p_{kr}^2 \log v_{kr}(\theta_k) + \lambda \sum_{s \in \mathcal{N}_r} \sum_{k \in \mathcal{K}} (p_{kr} - p_{ks})^2 \\ & - \mu \sum_{r \in \mathcal{L}} \sum_{k \in \mathcal{K}} p_{kr}^2, \end{aligned} \quad (20)$$

subject to:

$$\sum_{k \in \mathcal{K}} p_{kr} = 1, \quad \forall r \in \mathcal{L}; \quad p_{kr} \geq 0, \quad \forall k \in \mathcal{K}, r \in \mathcal{L}. \quad (21)$$

In particular, Alarcon and Marroquin [1] proposed a modification of the regularization term in Eq.(20), so that, it can take into account the interaction between classes:

$$\sum_{s \in \mathcal{N}_r} \sum_{k \in \mathcal{K}} (p_{kr} - p_{ks})^2 = \sum_{s \in \mathcal{N}_r} \sum_{k \in \mathcal{K}} p_{kr}^2 + p_{ks}^2 - 2p_{kr} p_{ks}. \quad (22)$$

In the previous expression there is no interaction between classes. Note that, the term  $\sum_k p_{kr} p_{ks}$  can be rewritten as follows:

$$\sum_{k \in \mathcal{K}} p_{kr} p_{ks} = \mathbf{p}_r^T \mathbf{I} \mathbf{p}_s, \quad (23)$$

where  $\mathbf{I}$  is the identity matrix and  $\mathbf{p}_r = [p_{kr}]_{k \in \{1, 2, \dots, K\}}$ . Now, the identity matrix could be changed for a more general matrix whose elements express the relation between classes of neighboring pixels, or in other words, between the components of vectors  $\mathbf{p}_r$  and  $\mathbf{p}_s$ . Let  $\mathbf{B}$  be such an interaction matrix

$$\mathbf{B} = [b_{kl}]; \quad k, l \in \{1, 2, \dots, K\}; \quad (24)$$

then the new interaction term is the following:

$$\mathbf{p}_r^T \mathbf{B} \mathbf{p}_s = \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{K}} b_{kl} p_{kr} p_{ls}, \quad (25)$$

and the new probabilistic model, the Generalized Entropy-Controlled Quadratic Markov Measure Field model, takes the form

$$\begin{aligned} U_{ECG}(\mathbf{p}) = & \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{L}} p_{kr}^2 (-\log v_{kr} - \mu) - \lambda \sum_{(r,s)} \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{K}} p_{kr} p_{ls} b_{kl} \\ & + \frac{\lambda}{2} \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{L}} |\mathcal{N}_r| p_{kr}^2, \end{aligned} \quad (26)$$

where  $\lambda > 0$  is a regularization parameter that controls the smoothness of the solution,  $\mu$  controls the entropy of the discrete probability  $\mathbf{p}_r$  at pixel  $r$  and  $|\mathcal{N}_r|$  represents the cardinality of the neighborhood at pixel  $r$ . This yields the following optimization problem:

$$\min_{\mathbf{p}} U_{ECG}(\mathbf{p}), \quad (27)$$

subject to the following constraints:

$$\sum_{k \in \mathcal{K}} p_{kr} = 1, \quad \forall r \in \mathcal{L}; \quad (28)$$

$$p_{kr} \geq 0, \quad \forall k \in \mathcal{K}, r \in \mathcal{L}. \quad (29)$$

The solution of the previous problem is obtained by using the Lagrange multipliers method for constrained optimization.

The Lagrangian, neglecting the non-negativity constraint, is

$$L(\mathbf{p}) = U_{ECG}(\mathbf{p}) - \sum_{r \in \mathcal{L}} \gamma_r (1 - \sum_{k \in \mathcal{K}} p_{kr}), \quad (30)$$

where  $\gamma_r$  represents the Lagrange multipliers. Then, the derivative of (30) with respect to  $p_{kr}$  is set to zero and using the equality constraints (28) one obtains the following equation:

$$p_{kr} = \frac{\gamma_r - \lambda \sum_{s \in \mathcal{N}_r} \sum_{l \in \mathcal{K}} p_{ls} b_{kl}}{2 \log v_{kr} + 2\mu - \lambda |\mathcal{N}_r|}, \tag{31}$$

where

$$\gamma_r = \frac{1 + \lambda S_1}{S_2}, \tag{32}$$

$$S_1 = \sum_{k \in \mathcal{K}} \frac{1}{2 \log v_{kr} + 2\mu - \lambda |\mathcal{N}_r|}, \tag{33}$$












$$S_2 = \sum_{k \in \mathcal{K}} \frac{\sum_{s \in \mathcal{N}_r} \sum_{l \in \mathcal{K}} p_{ls} b_{kl}}{2 \log v_{kr} + 2\mu - \lambda |\mathcal{N}_r|}. \tag{34}$$

The Eq.(31) provides an iterative Gauss-Seidel scheme and as an initial point we can take  $p_{kr} = v_{kr}$ . In order to satisfy the inequality conditions (29), we can use a projected Gauss-Seidel version, in which after updating  $p_{kr}$  with Eq.(31) one computes  $p_{kr} = \max\{0, p_{kr}\}$  and renormalizes the previous results for each pixel so that  $\sum_{k \in \mathcal{K}} p_{kr} = 1$ .

The values of the interaction matrix **B** are taken based on Boynton and Olson research [5], they are represented in Table 1 [1].

The numbers in bold denote the 11 eleven color categories. The values of the matrix are defined in Eq. (35).

**Table 1** Interaction matrix **B** between 11 color basic categories. **1** red, **2** green, **3** blue, **4** yellow, **5** purple, **6** orange, **7** gray, **8** black, **9** white, **10** pink, **11** brown

											
<b>1</b>	<b>1</b>	0	0	0	-1	-1	0	0	0	-1	-1
<b>2</b>	0	<b>1</b>	-1	-1	0	0	-1	0	0	0	-1
<b>3</b>	0	-1	<b>1</b>	0	-1	0	-1	0	0	0	0
<b>4</b>	0	-1	0	<b>1</b>	0	-1	0	0	0	0	-1
<b>5</b>	-1	0	-1	0	<b>1</b>	0	0	0	0	-1	-1
<b>6</b>	-1	0	0	-1	0	<b>1</b>	0	0	0	-1	-1
<b>7</b>	0	-1	-1	0	0	0	<b>1</b>	0	0	0	-1
<b>8</b>	0	0	0	0	0	0	0	<b>1</b>	0	0	0
<b>9</b>	0	0	0	0	0	0	0	0	<b>1</b>	0	0
<b>10</b>	-1	0	0	0	-1	-1	0	0	0	<b>1</b>	-1
<b>11</b>	-1	-1	0	-1	-1	-1	-1	0	0	-1	<b>1</b>

$$b_{kl} = \begin{cases} 1, & k = l; \\ -1, & k \leftrightarrow l; \\ 0, & \text{otherwise.} \end{cases} \quad (35)$$

where  $k \leftrightarrow l$  indicates the existence of interactions between  $k$  and  $l$  color categories according to experimental results in [5].

The inclusion of the interaction matrix, together with neighborhood information around the pixel, allows to consider the complexity of real color images and also justifies the type of color naming experiment selected by Alarcon and Marroquin. In the proposed color segmentation method, each extracted color region in the image has a descriptor denoted as  $\langle C, S \rangle$ , where  $C$  and  $S$  are the color and intensity attributes, respectively. The possible values for  $C$  and  $S$  are the following:

$$C \in [\textit{red}, \textit{green}, \textit{blue}, \textit{yellow}, \textit{purple}, \textit{orange}, \textit{gray}, \textit{black}, \textit{white}, \textit{pink}, \textit{brown}] \quad (36)$$

$$S \in [\textit{light}, \textit{dark}] \quad (37)$$

The estimation of the attribute  $C$  is based on the solution of the optimization problem (27–29), i.e., the iterative process provided by Eq. (31). Hence, the attribute  $C_r$  for each pixel  $r$  is computed with:

$$C_r = \arg \max_{k \in \mathcal{K}} p_{kr}. \quad (38)$$

On the other hand, the attribute  $S$  is a refinement of the attribute  $C$ . The estimation of  $S$  allows us to subdivide each found color category in two subcategories according to the intensity information, see (37). For computing the attribute  $S$  we solve the optimization problem (20–21) for each found category  $C$ . In this case, the value of  $v_{kr}$  is calculated as follows:

$$v_{kr}(\mu_k, \sigma_k) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{(L_r - \mu_k)^2}{2\sigma_k^2}\right), \quad (39)$$

where  $k \in \mathcal{K}$ , with  $K = 2$ , indicates the models for light and dark subcategories,  $L_r$  is the intensity value at pixel  $r$  in the  $Luv$  space,  $\mu_k$  is the expected value for the intensity model  $k$  and  $\sigma_k$  is the corresponding standard deviation.

In order to compute the parameters in Eq. (39) we solve the optimization problem (20–21) because in the refinement step the interaction between classes is not considered. The algorithm is an iterative process that alternates between the computation of  $\mathbf{p}$  (Probabilistic segmentation step) and the model parameter estimation  $\theta = \mu$  (Model estimation step). In the first step,  $U(\mathbf{p}, \mu)$  is minimized with respect to  $\mathbf{p}$  for a given  $\mu$  and in the second step  $U(\mathbf{p}, \mu)$  is minimized with respect to  $\mu$  keeping  $\mathbf{p}$  fixed. The parameter  $\sigma_k$  is assumed to be a constant. The initial guess for  $\mu_k$  is a value between the minimum and maximum levels in the regions associated to the

attribute  $C$ . The value of the  $\sigma_k$  is estimated as the mode of the sample variance computed over a  $3 \times 3$  sliding window throughout regions in  $C$ .

In the first step, the values of  $\mathbf{p}_r$  are computed deriving  $U(\mathbf{p}, \mu)$  with respect to  $p_{kr}$ , equating to zero and solving for  $p_{kr}$

$$p_{kr} = \frac{n_{kr}}{m_{kr}} + \frac{1 - \sum_{l=1}^K \frac{n_{lr}}{m_{lr}}}{\sum_{l=1}^K \frac{m_{kr}}{m_{lr}}}, \quad (40)$$

where  $n_{kr} = \lambda \sum_{s \in \mathcal{N}_r} p_{ks}$ ,  $m_{kr} = (-\log v_{kr} - \mu) + \lambda |\mathcal{N}_r|$ ,  $|\mathcal{N}_r|$  is the cardinality of the neighborhood of the pixel  $r$  and  $K = 2$ .

In the second step, one sets the derivative of  $U(\mathbf{p}, \mu)$  with respect to  $\mu_k$  equal to zero and solves for  $\mu_k$ . Then, one obtains the following closed formula:

$$\mu_k = \frac{\sum_{r \in \mathcal{L}} p_{kr}^2 L_r}{\sum_{r \in \mathcal{L}} p_{kr}^2}. \quad (41)$$

The iterative process of computing  $p_{kr}$  and  $\mu_k$  continues until convergence. Finally the  $S_r$  attribute for each  $r$  is obtained with:

$$S_r = \arg \max_{k \in \mathcal{K}} p_{kr}. \quad (42)$$

Segmentation results obtained through a described proposal are illustrated in Fig. 4.

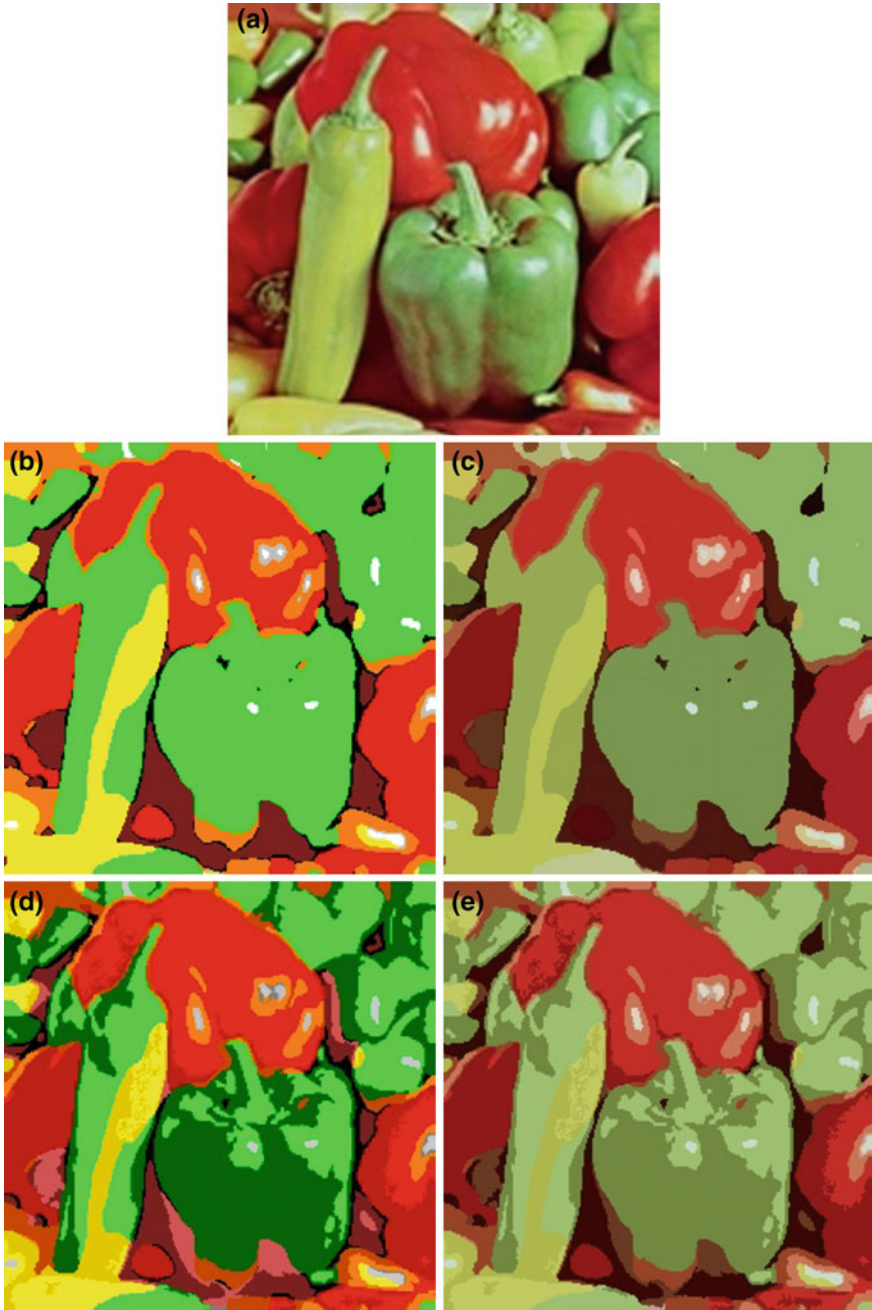
## 5 Other Applications

In this section we present some examples of applications of the color categorization model proposed in [1].

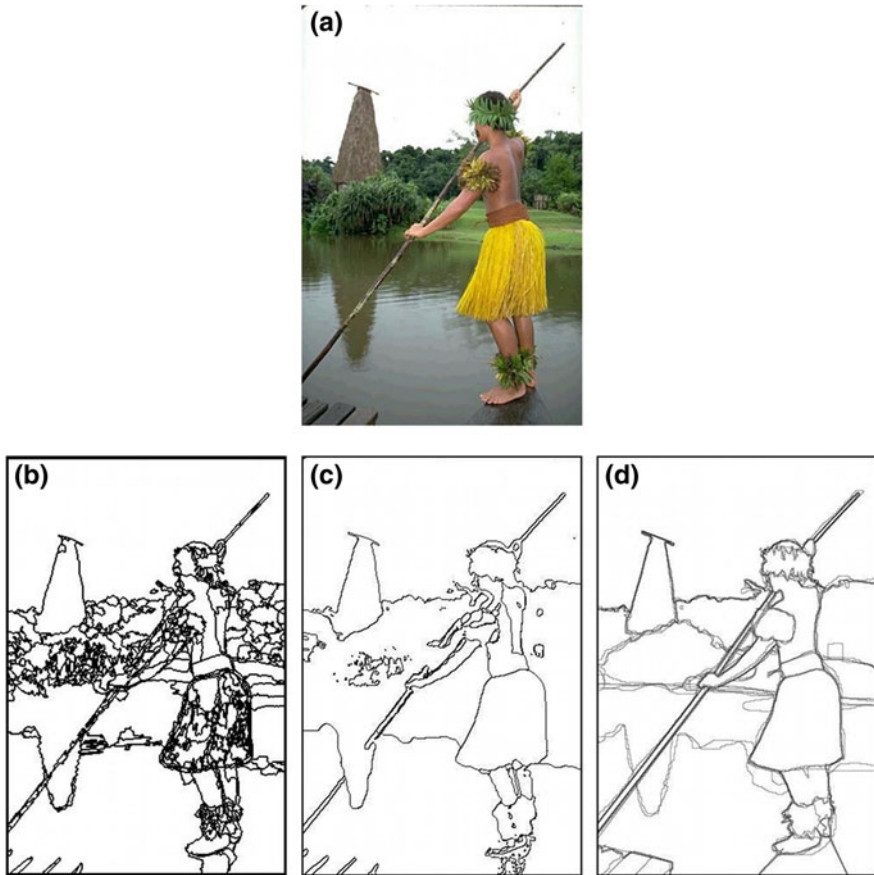
### 5.1 Detection of Perceptually Salient Edges

An interesting aspect of using color categorization model for image segmentation is the obtained edge map. The extracted edges are the most significant and they enclose the most significant regions in the image from the perceptual point of view.

The results depicted in Fig. 5 demonstrate that the use of 11 basic color categories allows one to extract only the contours that are perceptually significant which leads to a more concise description of color boundaries. On the other hand, observe that the results of the edge detection based on color categorization model present a better approximation of a human judge. Figure 5 **a**, **d** are available online at <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>.



**Fig. 4** **a** Original image, **b** estimation of  $C$  attribute, **c** representation of the result using the average  $RGB$  color information in each segment of the image in **b**, **d** estimation of  $S$  given  $C$ , **e** representation of the result in **d** using the average  $RGB$  color information in each segment of the image in **d**



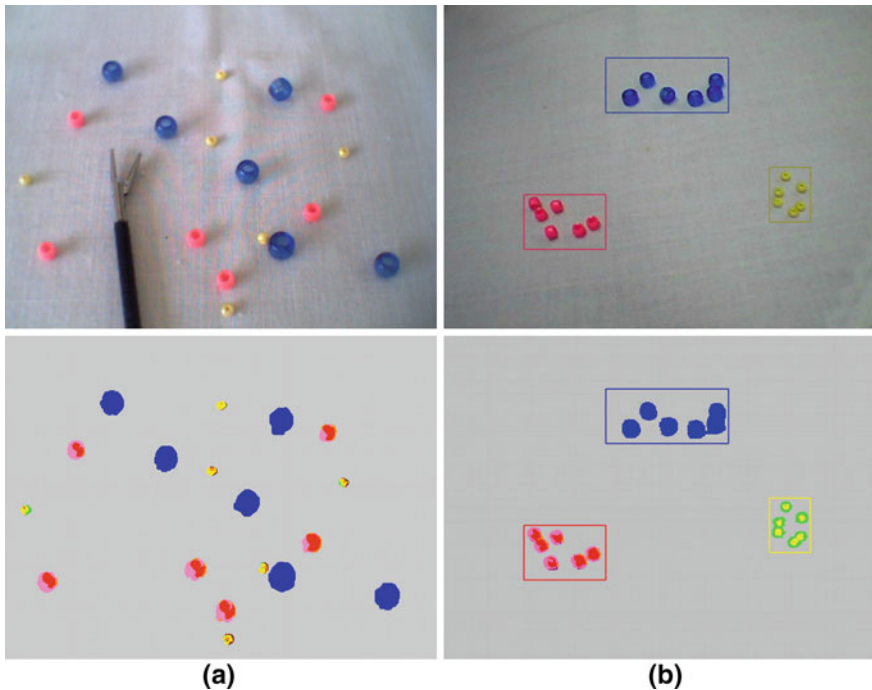
**Fig. 5** **a** Original image, **b** borders detected by *Mean Shift* algorithm, **c** borders detected by the Alarcon and Marroquin proposal, **d** borders detected by several human observers during the research published in <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>

### 5.2 Video Tracking

Laparoscopic surgery is defined as a minimal invasive surgery. Laparoscopic procedure includes a video camera and several thin instruments. During the surgery, small incisions are made and plastic tubes, called ports, are placed through these incisions. The camera and the instruments are then introduced through the ports which allows access inside the patient. The images of organs inside the abdomen and sensed by the camera are transmitted onto a television monitor. Unlike the conventional surgery, in a laparoscopic procedure, the video camera becomes the surgeon’s eyes since the surgeon uses the image from the video camera located inside the patient’s body to perform the procedure. Training for this kind of surgery is a challenge for physicians: how can a surgeon acquire enough skills to do a laparoscopic procedure without a



real patient? The answer to this question could be found in Digital Image Processing, which has been widely used in Medicine and nowadays has become a powerful tool in the medical field. An interesting experiment was done using the results in [1] with the purpose to elaborate an algorithm for a laparoscopic trainer through Digital Image Processing<sup>1</sup>. In a simulated surgery laboratory a physician does similar exercises as in a real laparoscopic procedure, i.e., he makes small incisions inside a simulated body and does the necessary operations to reach an objective. In the experiment, Fig. 6, the surgeon must move balls of different colors to a specified location of the simulated body in a limited time. If the surgeon places a ball in a wrong location or if the time is over, the algorithm emits a warning signal. Using the color categorization model in [1] with a direct label assignment (see Sect. 4.1), we segmented the video sequence of the simulated laparoscopic surgery in order to track the surgeon movements during the training. Figure 6 shows the results of the experiments for one frame of the video sequence of 36 seconds. The Overall 366 frames were segmented in 1 minute and 17 seconds.



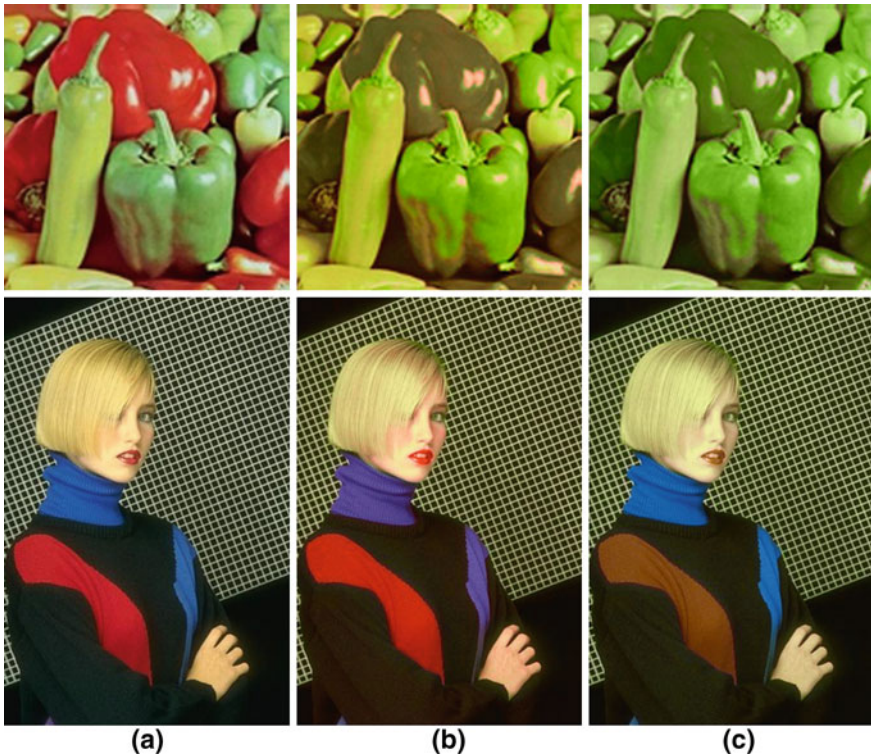
**Fig. 6** Two frames of the video sequence. **a** A frame (top image) and its segmentation (bottom image) using Alarcon and Marroquín direct label assignment, **b** the final frame (top image) and its segmentation, squares indicate the expected right locations for the balls

<sup>1</sup> Contract grant sponsor: *PROMEP/103.5/10/596*, acknowledge for the Universidad Anahuac

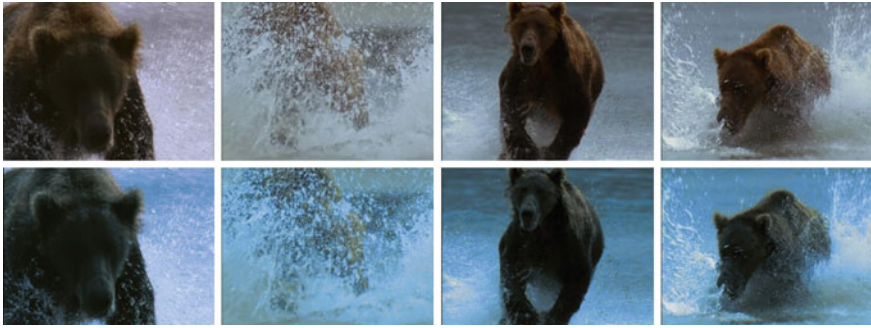
In Fig. 6 a we can observe that the algorithm only detects gray, red, blue, yellow, green and brown categories. As a prior information we know that the pin, Fig. 6 a in the top image, does not need to be tracked. As the pin is in black, it can be easily detected and removed from the segmentation. In this case the segmentation algorithm assigns the same color as in the background part of the image. From the Fig. 6 b we can see that the target areas are also detected.

### 5.3 Image and Video Recolourization

Image recolourization is a technique that consists of changing colors in a color image [8]. The recolourization begins with a probabilistic segmentation of the image, see Sect. 4.2, or a sequence of images in case of a video, in regions with an assumed same color. Then, the user assigns a color for each segmented class. These colors are used to recolourize the corresponding region. The computed probabilities or memberships, Eq. (31), allow us to define the weights of a linear combination of colors provided by the user, see details in [9, 8].



**Fig. 7** Image Recolourization Algorithm [8] based on the color categorization model proposed by Alarcon and Marroquin. **a** Original Image, **b** Recolourization, **c** Recolourization



**Fig. 8** Video Recolorization Algorithm [8] based on the color categorization model proposed by Alarcon and Marroquin. First row: some frames of the original video, second row: corresponding recolourized frames

Figure 7 depicts two recolourized images. The probabilities were computed using the Color Categorization Model proposed by Alarcon and Marroquin [1]. Except for assignment of colors to each color category, the procedure is completely automatic.

In case of video recolourization, the process is applied frame-by-frame. First row of Fig. 8 shows 4 selected frames of a sequence of 900 frames and the second row shows the corresponding recolourized frame.

## 6 Conclusions

Color categorization models based on the discrete formalism are not appropriate, because humans assign color categories taking into consideration a degree of membership. All color categorization models allow us to segment a color space in color categories defined by a selected color vocabulary. The segmentation classes are the specified color terms. The use of color categorization model for segmenting color images allows us to select a nonarbitrary number of color classes, because it is based on relevant findings about human color naming.

The basic color categories defined by Berlin and Kay in 1969 are universal, but we can use intensity, saturation and hue modifiers in order to have a detailed description of the color composition of the scene.

Alarcon and Marroquin's color categorization model takes into account the research done by Berlin and Kay. Although the model is based on the 11 basic color categories proposed by Berlin and Kay, an extension to other vocabularies is possible, being only necessary to change the experimental likelihood estimation. The segmentation algorithm considers the perceptual interactions established by Boynton and Olson what makes it possible to model the complexity of real color scenes. The segmentation method also allows us to obtain an edge map which may be used in a filtering process that preserves the perceptually salient borders in an image.

## References

1. Alarcon TE, Marroquin JL (2009) Linguistic color image segmentation using a hierarchical Bayesian approach. *Color Res Appl* 34:299–309
2. Benavente R, Vanrell M, Baldrich R (2004) Estimation of fuzzy Sets for computacional colour categorization. *Color Res Appl* 29:5342–5353
3. Benavente R, Vanrell M, Baldrich R (2006) A data set for colour fuzzy naming. *Color Res Appl* 31(1):48–56
4. Berlin B, Kay P (1969) Basic color terms: Their universality and evolution. University of California Press, Berkeley
5. Boynton RM, Olson CX (1987) Locating basic colors in the OSA space. *Color Res Appl* 12:94–105
6. Burges CJC (1998) A tutorial on support vector machines for pattern recognition. *Data Min Knowl Disc* 2:121–167
7. Cairo JE (1977) The neurophysiological basis of basic color terms. Ph.D. thesis, State University of New York at Binghamton, New York
8. Cedeño OD, Rivera M, Alarcon T (2010) Bayesian scheme for interactive colourization, recolourization and image/video editing. *Comput Graph Forum* 29(8):2372–2386
9. Cedeño OD, Rivera M, Mayorga PP (2007) Computing the alpha-channel with probabilistic segmentation for image colorization. In: IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, 14–20 October 2007, pp. 1–7
10. Comaniciu D, Meer P (2002) Mean shift analisis and applications. *IEEE Trans Pattern Anal Mach Intell* 24(5):603–619
11. Hofmann T (1999) Probabilistic latent semantic indexing. In: SIGIR'99, ACM. Conference on Research and Development. Inf Retrieval, pp. 50–57
12. Kay P, McDaniel C (1978) The linguistic significance of meanings of basic color terms. *Color Res Appl* 54:610–646
13. Kelly K, Judd D (1955) The ISCC-NBS color names dictionary and the universal color language (the iscc-nbs method of designating colors and a dictionary of color names). NBS circular 553
14. Lammens JM (1994) A computacional model of color perception and color naming. Ph.D. thesis, Buffalo. University of New York, US
15. McDaniel C (1972) Hue perception and hue naming. Ph.D. thesis, Harvard college, Cambridge
16. Menegaz G, Le Troter A, Sequeira J, Boi JM (2007) A discrete model for color naming. *EURASIP J Appl Sig Process* 2007(1):113–129
17. Mojsilovic A (2005) A method for color naming and description of color composition in images. *IEEE Trans Image Process* 14(5):690–699
18. Rivera M, Ocegueda O, Marroquin JL (2005) Entropy controlled Gauss-Markov random measure field models for early vision. *VLSM, LNCS* 3752:137–148
19. Seaborn M, Hoplewhite L, Stonham J (2005) Fuzzy colour category map for the measurement of colour similarity and dissimilarity. *Pattern Recognit* 38(2):165–177
20. Shepard RN (1987) Toward a universal law of generalization for psychological science. *Science* 237(4820):1317–1323
21. Sturges J, Whitfield T (1995) Locating basic colors in the Munsell space. *Color Res Appl* 20:364–376
22. Tversky A (1977) Features of similarity. *Psychol Rev* 84(4):327–352
23. Valois RLD, Abramov I, Jacobs GH (1966) Analysis of response patterns of LGN cells. *J Opt Soc Am* 56(7):966–977
24. Joost van de Weijer Cordelia Schmid JV (2007) Learning Ccolor names from real-world images. In: IEEE Conference on Computer Vision. Pattern Recognit, pp. 1–8
25. Wooten BR (1970) The effects of simultaneous and successive chromatic constraint on spectral hue. Ph.D. thesis, Brown University, Providence
26. Wyszecki G, Stiles W (1982) Color science: Concepts and methods, quantitative data and formulae, 2nd edn. Wiley, New York

# Skin Detection and Segmentation in Color Images

Michal Kawulok, Jakub Nalepa and Jolanta Kawulok

**Abstract** This chapter presents an overview of existing methods for human skin detection and segmentation. First of all, the skin color modeling schemes are outlined, and their limitations are discussed based on the presented experimental study. Then, we explain the techniques which were reported helpful in improving the efficacy of color-based classification, namely (1) textural features extraction, (2) model adaptation schemes, and (3) spatial analysis of the skin blobs. The chapter presents meaningful qualitative and quantitative results obtained during our study, which demonstrate the benefits of exploiting particular techniques for improving the skin detection outcome.

**Keywords** Skin detection · Skin segmentation · Skin color models · Adaptive skin modeling · Face detection and tracking · Hand detection and tracking

## 1 Introduction

Skin detection and segmentation is a challenging problem in color image processing that has been extensively studied over the years. In general, the existing techniques are based on the premise that the skin color can be effectively modeled in various color spaces, which in turn allows for segmenting the skin regions in a given color image. Applications of skin detection are of a wide range and significance, including:

---

An erratum to this chapter is available at [10.1007/978-94-007-7584-8\\_14](https://doi.org/10.1007/978-94-007-7584-8_14)

---

M. Kawulok (✉) · J. Nalepa · J. Kawulok  
Institute of Informatics, Silesian University of Technology, Gliwice, Poland  
e-mail: Michal.Kawulok@polsl.pl

J. Nalepa  
e-mail: Jakub.Nalepa@polsl.pl

J. Kawulok  
e-mail: Jolanta.Kawulok@polsl.pl

1. Hand and face detection and tracking for gesture recognition and human-computer interaction [10, 33, 43].
2. Objectable content filtering for the sake of blocking nude images and videos [59, 101].
3. Feature extraction for content-based image retrieval [56].
4. Image coding using regions of interest [1, 14, 17, 26], and many more.

First of all, we would like to clearly define the terms of *skin detection* and *skin region segmentation*, as they are often used interchangeably which leads to certain confusion. *Skin detection* is a process whose aim is to determine whether a given video sequence, an image, a region or a pixel presents human skin. In most cases, it is performed at a pixel level first—every pixel is classified independently based on its chrominance and other low-level features extracted from its neighborhood. Afterwards, the decision may be verified at a higher level, and for example if it occurs that the pixels classified as skin are sparse in a given image, the detection outcome would be negative for that very image. Skin detection is the preliminary step to *skin region segmentation*, whose aim is to determine the boundaries of skin regions. In the simplest approach, this may not involve any additional processing over the pixel-wise skin detection, based on the assumption that the skin regions are formed by the adjacent pixels classified as skin. However, it has been reported in many works that using more advanced methods the segmentation precision may be definitely improved. Segmenting skin regions is helpful for extracting shapes of hands, faces and other body parts, but it may also be used as a verification step which improves the efficacy of skin detection measured at the pixel level.

This chapter contains a detailed overview of the skin color modeling algorithms, as well as it presents the methods which reduce the skin detection errors. General overview of existing skin detectors is given in Sect. 2, and structure of the chapter is presented there as well. The discussed methods are compared both on the theoretical and experimental basis. Their performance is evaluated quantitatively and qualitatively using images from the benchmark data sets. The evaluation procedure is discussed in Sect. 3.

## 2 General Categories of Skin Detection Methods

A conventional approach towards solving the skin detection problem consists in defining a skin color model, using which every individual pixel can later be classified based on its position in the color space, independently from its neighbors. This research direction has been widely explored and plenty of different pixel-wise methods were proposed that operate in most of existing color spaces. These techniques are discussed in Sect. 4. Skin color can be modeled either as a set of rules and thresholds defined in color spaces based on the experimental study (Sect. 4.2), or it can be trained using machine learning (Sect. 4.3). There were several good reviews published on skin color modeling. In 2003, V. Vezhnevets et al. presented the first thorough survey on skin detection [87]. Later, in 2005, several rule-based methods

were compared by F. Gasparini et al. [29]. In the same year, S.L. Phung et al. presented a comparison, mainly focused on statistical color modeling [64]. The most recent interesting survey on skin detection was published in 2007 by P. Kakumanu et al. [41].

It is worth noting that while color-based methods present different characteristics and therefore they can be found suitable in specific conditions, their effectiveness is limited due to high variance of skin color and its low specificity. Skin color depends on such individual factors like race, age or complexion. Intra-personal differences may also be substantial because of variations in lighting conditions or individual's physical state. Moreover, background objects often have skin-like color, which results in observing false positive errors in the segmentation outcome.

Despite the aforementioned shortcomings, it must be stated that the color is the basic feature for detecting skin regions. However, its discriminating power is highly limited, and by taking advantage of additional data sources, the errors can be significantly reduced. There are some effective approaches towards improving the performance of the pixel-wise color-based detectors, which we have grouped into three main categories, namely:

- A. Texture-based models (Sect. 5). The textural features may improve the stability of the color-based skin models by rejecting the regions which are not smooth. According to our observation, simple textural features are most useful here.
- B. Model adaptation techniques (Sect. 6). Skin color models may be adapted to a particular scene based on a whole-image analysis (Sect. 6.1), tracking (Sect. 6.2), or from a given skin sample (Sect. 6.3). A skin sample, i.e. a region which contains a representative set of skin pixels, can be either acquired automatically using face and hand detectors or be marked manually by an operator. This makes the skin color model more specific, which boosts the detection rate and decreases the false positives.
- C. Spatial analysis (Sect. 7). Spatial analysis consists in segmenting the skin blobs which may help decrease the detection errors by rejecting isolated false positive pixels.

It is worth noting that all of the methods discussed in this chapter require color-based skin modeling, which constitutes the primary source of information for skin detection. While skin color modeling has been widely addressed in the literature, these directions were paid much less attention. This may be somehow surprising, given how much they may contribute to increasing the robustness of skin detectors.

## 3 Evaluation Procedure

### 3.1 Evaluation Metrics

Skin detection performance is measured based on the number of correctly (i.e.  $TP$ —true positives and  $TN$ —true negatives) and incorrectly classified pixels (i.e.

$FN$ —false negatives and  $FP$ —false positives). Based on these values, the following ratios can be computed:

- A. *False positive rate*:  $\delta_{fp} = FP/(FP + TN)$ , i.e. the percentage of background pixels misclassified as skin [40].
- B. *False negative rate*:  $\delta_{fn} = FN/(FN + TP)$ , i.e. the percentage of skin pixels misclassified as background [40].
- C. *Recall*, also referred to as *correct detection rate* or *true positive rate*:  $\eta_{tp} = TP/(FN + TP) = 1 - \delta_{fn}$ , i.e. the percentage of skin pixels correctly classified as skin [29].
- D. *Precision*:  $\eta_{prec} = TP/(TP + FP)$ , i.e. the percentage of correctly classified pixels out of all the pixels classified as skin [29].
- E. *F-measure*: the harmonic mean of precision and recall [52].

If the classification is non-binary, i.e. for every pixel the skin probability is computed, then the false positive and false negative rates depend on the acceptance threshold. The higher the threshold is, the less false positives are reported, but also the false negatives increase. Mutual relation of these two errors is often presented using *receiver operating characteristics (ROC)* [64], and the *area under curve* can also be used as the effectiveness determinant [69].

In our works we also investigated the false negative rate ( $\delta_{fn}^{(\eta)}$ ) obtained for a fixed false positive error  $\delta_{fp} = \eta$  [46]. Furthermore, we used the *minimal detection error* ( $\delta_{min} = (\delta_{fp} + \delta_{fn})/2$ ), where the threshold is set to a value, for which this sum is the smallest. In this chapter we rely on false positive and false negative rate, and we present their dependence using *ROC* curves, when it is applicable.

## 3.2 Data Sets

In their overview on skin color modeling, S.L. Phung et al. introduced a new benchmark data set, namely *ECU face and skin detection database* [64]. This data set consists of 4000 color images and ground-truth data, in which skin areas are annotated. The images were gathered from the Internet to provide appropriate diversity, and results obtained for this database were often reported in many works on skin detection. Therefore, this set was also used for all the comparisons presented in this chapter. Some examples of images from the data set are shown in Fig. 1. In all experiments reported here, the database was split into two equinumerous parts. The first 2000 images are used for training (*ECU-T*), and the remaining 2000 images are used for validation (*ECU-V*). If an algorithm does not require training, then only the *ECU-V* set is used. The data set contains some images acquired in the same conditions, but as they appear close to each other in the data set, there is no risk that two similar images will repeat in *ECU-T* and *ECU-V*.

Among other data sets which can be used for evaluating skin detection, the following may be mentioned: (1) M.J. Jones and J.M. Rehg introduced the Compaq database [39], (2) S.J. Schmutge et al. composed a data set based on images derived



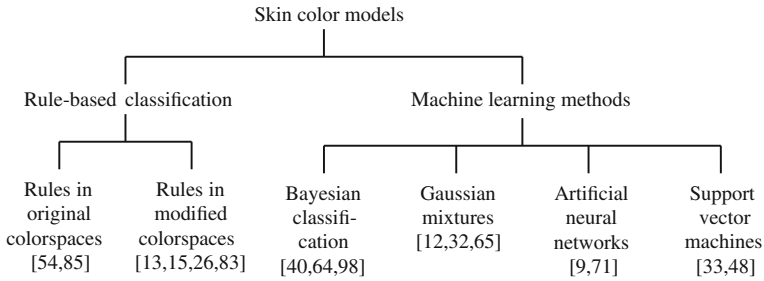


Fig. 1 Examples of images from ECU face and skin detection database [64]

from existing databases, in which they annotated skin regions [69], (3) we have created our hand image database for gesture recognition purposes (available at <http://sun.aei.polsl.pl/~mkawulok/gestures>) [46].

### 4 Skin Color Modeling

Classification of skin color modeling methods, which are given more attention in this section, is presented in Fig. 2, and several most important references are given for every category. In general, the decision rules can be defined explicitly in commonly used color spaces or in a modified color space, in which the skin could be easier separated from the background. Machine learning methods require a training set, from which the decision rules are learned. A number of learning schemes have been used for this purpose, and the most important categories are presented in the figure and outlined in the section.



**Fig. 2** General categories of skin color models

### 4.1 Color Spaces and Color Normalization

Skin color has been modeled in many color spaces using various techniques, which is summarized in Table 1. In many works it is proposed to reject the luminance component to achieve invariance to illuminance conditions. Hence, those color spaces are then preferred, in which the luminance is separated from the chrominance components, e.g.  $YC_bC_r$ ,  $YUV$ ,  $HSV$ . However, it was reported by many researchers that the illumination plays an important role in modeling the skin color and should not be excluded from the model [65, 72, 90]. The problem of determining an optimal color space for skin detection was addressed by A. Albiol et al., who provided a theoretical

**Table 1** Color spaces used for skin color modeling

Color space	Skin color models
$RGB$	Shin [72], Kovac [54, 76], Brand [11], Jones [40], Choi [17], Bhoyar [9], Seow [71], Taqa [82], Han [33], Ng [62], Jiang [38], Conci [19]
$YC_bC_r$	Hsu [36], Phung [65], Hossain [35], Kawulok [48]
$rg$	Stoerring [79], Greenspan [32], Caetano [12]
$HSI$	Schmugge [69], Jagadesh [37, 67]
$HSV$	Sobotka [74], Tsekeridou [85]
$CIELUV$	Yang [92]
$YIQ$	Duan [23]
$YUV$	Zafarifar [97]
Multiple color spaces	Kukharev [57], Wang [89], Abin [3], Fotouhi [27]

proof [5] that in every color space optimal skin detection rules can be defined. Following their argumentation, small differences in performance are attributed exclusively to the quantization of a color space. This conclusion is certainly correct, however the simplicity of the skin model may depend on the selection of color space. Based on the scatter analysis, as well as 2D and 3D skin-tone histograms, M.C. Shin et al. reported that it is the  $RGB$  color space which provides the best separability between skin and non-skin color [72]. Furthermore, their study confirmed that the illumination is crucial for increasing the separability between the skin and non-skin pixels. In their later works, they argued that the  $HSI$  color space should be chosen when the skin color is modeled based on the histogram analysis [69].

Color normalization plays an important role in skin color modeling [8, 53, 86]. M. Stoerring et al. investigated skin color appearance under different lighting conditions [79]. They have observed that location of the skin locus in the normalized  $rg$  color space, where  $r = R/(R + G + B)$  and  $g = G/(R + G + B)$ , depends on the color temperature of the light source. Negative influence of the changing lighting can be mitigated by appropriate color normalization. There exist many general techniques for color normalization [24, 25, 30, 58, 68], which can be applied prior to skin detection. Among them, the gray world transform is often reported as quite effective, while simple technique. Given an image with sufficient amount of color variations, the mean value of the  $R$ ,  $G$ , and  $B$  channels should average to a common gray value, which equals 128 in the case of the  $RGB$  color space. In order to achieve this goal, each channel is scaled linearly:

$$c_N = c \cdot 128/\mu_c, \quad (1)$$

where  $c$  indicates the channel (i.e.  $R$ ,  $G$  or  $B$ ),  $\mu_c$  is the mean value in the channel for a given image prior to normalization, and  $c_N$  is the value after normalization. In the modified gray world transform [24], the scale factors are determined in such a way that each color is counted only once for a given image, even if there are multiple pixels having the same position in the color space.

There are also a number of normalization techniques designed for the sake of skin detection. R.-L. Hsu et al. proposed a lighting compensation technique, which operates in  $RGB$  color space prior to applying an elliptical skin model [36]. The top 5% of the gamma-corrected luminance values in the image define the *reference white* ( $L' = L^\gamma$ , where  $L$  and  $L'$  are input and gamma-corrected luminance values, respectively). After that, the  $R$ ,  $G$  and  $B$  components are linearly scaled, so that the average gray value of the reference-white pixels equals 255. This operation normalizes the white balance and makes the skin model applicable to various lighting conditions. P. Kakumanu [42] proposed to use a neural network to determine the normalization coefficients dynamically for every image. J. Yang et al. introduced a modified Gamma correction [91] which was reported to improve the results obtained using statistical methods. U. Yang et al. took into account the physics of the image acquisition process and proposed a learning scheme [93] which constructs an illumination-invariant color space based on a presented training set with annotated skin regions. The method operates in a modified  $rg$  color space. Skin pixels are

subject to principal components analysis to determine the direction of the smallest variance. Thus, a single-dimensional space is learned, which minimizes the variance of the skin tone for a given training set.

## 4.2 Rule-Based Classification

Rule-based methods operate using a set of thresholds and conditions defined either in one of existing color spaces (e.g.  $RGB$ ) or the image is first transformed into a new color space, in which the skin color can be easier separated from the non-skin color. There have been a number of methods proposed which adopt such an approach and new algorithms are still being proposed. Unfortunately, the recent methods do not contribute much to the state of the art, which is confirmed by the presented experimental results.

One of the first skin color modeling techniques was proposed by K. Sobottka and I. Pitas in 1996. They observed that the skin tone can be defined using two ranges of  $S \in [0.23, 0.68]$  and  $H \in [0, 50]$  values in the  $HSV$  color model [74]. A modification of this simple technique was later proposed by S. Tsekeridou and I. Pitas [85] and it was used for face region segmentation in the image watermarking system [63]. The rule takes the following form in the  $HSV$  color space:

$$\left\{ \begin{array}{l} (0 \leq H \leq 25) \vee (335 \leq H \leq 360) \\ (0.2 \leq S \leq 0.6) \wedge (0.4 \leq V) . \end{array} \right. \quad (2)$$

Projection of these rules onto the  $RGB$  color space is shown in Fig. 3a using  $RG$ ,  $RB$ ,  $GB$  and normalized  $rg$  planes. The darker shade indicates the higher density of skin pixels.

J. Kovac et al. defined fixed rules [54, 76] that determine skin color in two color spaces, namely  $RGB$  and  $C_bC_r$  (ignoring the luminance channel). The rules in  $RGB$  are as follows:

$$\left\{ \begin{array}{l} (R > 95) \wedge (G > 40) \wedge (B > 20) \\ \max(R, G, B) - \min(R, G, B) > 15 \\ |R - G| > 15 \wedge (R > G) \wedge (R > B) \end{array} \right. \quad (3)$$

for uniform daylight illumination or

$$\left\{ \begin{array}{l} (R > 220) \wedge (G > 210) \wedge (B > 170) \\ |R - G| \leq 15 \wedge (R > G) \wedge (R > B) \end{array} \right. \quad (4)$$

for flashlight lateral illumination. If the lighting conditions are unknown, then a pixel is considered as skin, if its color meets one of these two conditions. These rules are illustrated in  $RG$ ,  $RB$ ,  $GB$  and  $rg$  planes in Fig. 3b.

R.-L. Hsu et al. defined the skin model [36] in the  $YC_bC_r$  color space, which is applied after the normalization procedure outlined earlier in Sect. 4.1. The authors

observed that the skin tone forms an elliptical cluster in the  $C_b C_r$  subspace. However, as the cluster's location depends on the luminance, they proposed to nonlinearly modify the  $C_b$  and  $C_r$  values depending on the luminance  $Y$ , if it is outside the range  $Y \in [125, 188]$ . Afterwards, the skin cluster is modeled with an ellipse in the modified  $C'_b C'_r$  subspace. Skin distribution modeled by these rules is presented in Fig. 3c.

Elliptical model of the skin cluster was also presented by J.-C. Terrillon et al., who argued that the skin color can be effectively modeled using the Mahalanobis distances computed in a modified  $STV$  color space [83]. This model was further improved by F. Tomaz et al. [84].

G. Kukharev and A. Nowosielski defined the skin detection rules [57] using two color spaces, i.e.  $RGB$  and  $YC_b C_r$ . Here, a pixel value is regarded as skin:

$$\begin{cases} R > G \wedge R > B \\ (G \geq B \wedge 5R - 12G + 7B \geq 0) \vee (G < B \wedge 5R + 7G - 12B \geq 0) \\ C_r \in (135, 180) \wedge C_b \in (85, 135) \wedge Y > 80. \end{cases} \quad (5)$$

This model is presented in  $RGB$  and  $(r, g)$  coordinates in Fig. 3d.

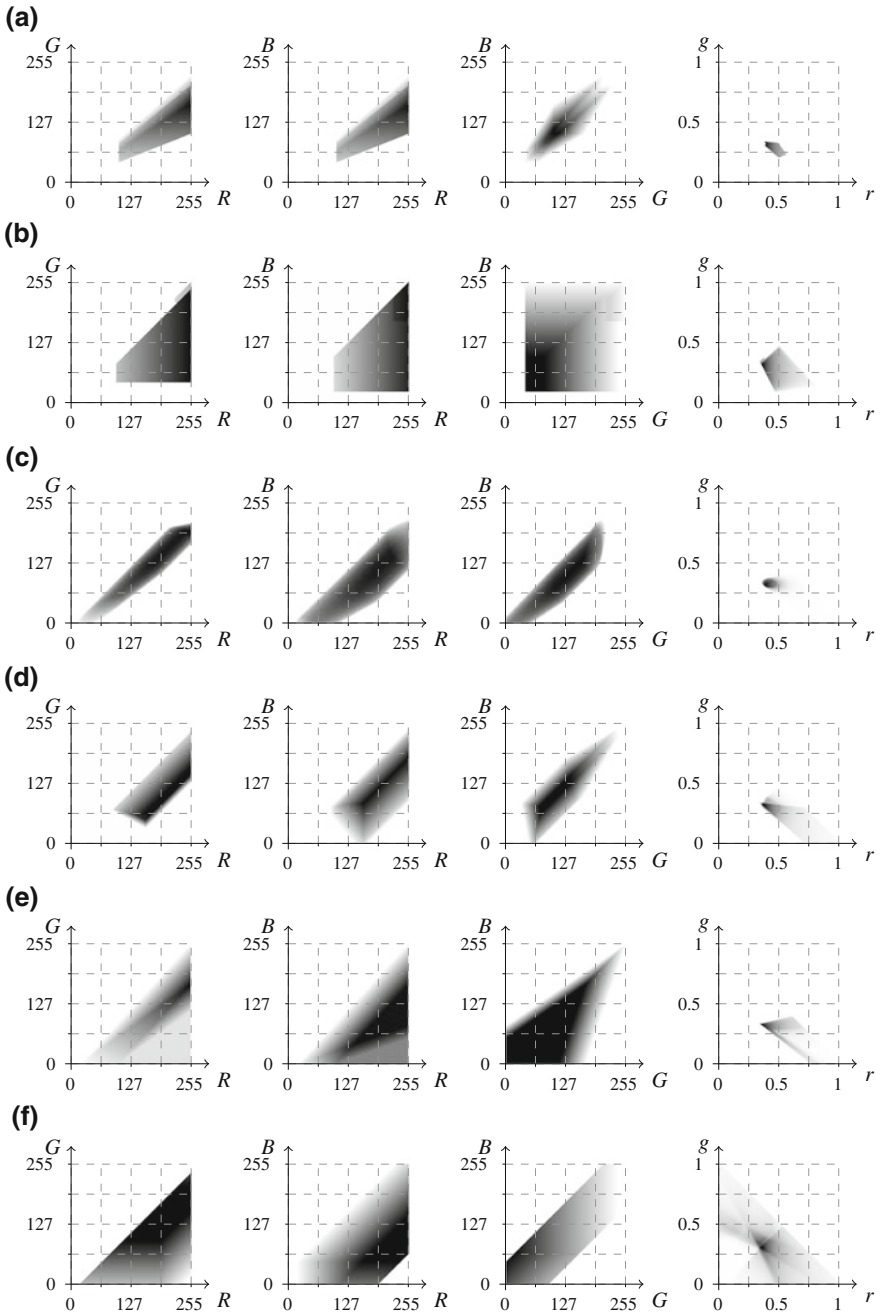
In 2009, A. Cheddad et al. proposed to transform the normalized  $RGB$  color space into a single-dimensional error signal, in which the skin color can be modeled using a Gaussian curve [13]. After the transformation, a pixel is regarded as skin if it fits between two fixed thresholds determined based on the standard deviation of the curve. The model is also illustrated in the  $RGB$  color space in Fig. 3e.

Recently, Y.-H. Chen et al. analyzed the distribution of skin color in a color space derived from the  $RGB$  model [15]. They observed that the skin color is clustered in a three-channel color space obtained by subtracting the  $RGB$  values:  $sR = R - G$ ,  $sG = G - B$ ,  $sB = R - B$ . After that they set the thresholds in the new color space to classify every pixel. The rules are visualized in Fig. 3f.

Some skin detection results obtained using six different methods are presented in Fig. 4. False positives are marked with a red shade, while false negatives—with the blue, and true positives are rendered normally. It can be seen from the figure that the detection error is generally quite high, however some methods present better performance in some specific cases. The overall performance scores are compared in Fig. 5. Rule-based models deliver worse results than the Bayesian skin model, however their main advantage lies in their simplicity. If the lighting conditions are controlled and fixed, then using a rule-based model may be a reasonable choice. Nevertheless, in a general case, the machine learning approaches outperform the models based on fixed rules defined in color spaces.

### 4.3 Machine Learning Methods

In contrast to the rule-based methods, the skin model can also be learned from a classified training set of skin and non-skin pixels using machine learning techniques. In most cases, such an approach delivers much better results and it does not require



**Fig. 3** Skin color models presented in  $RG$ ,  $RB$ ,  $GB$  and  $r_g - g$  planes. S. Tsekeridou and I Pitas [85] (a). J. Kovac et al. [54] (b). R.-L. Hsu et al. [36] (c). G. Kukharev and A. Nowosielski [57] (d). A. Cheddad et al. [13] (e). Y.-H. Chen et al. [15] (f)

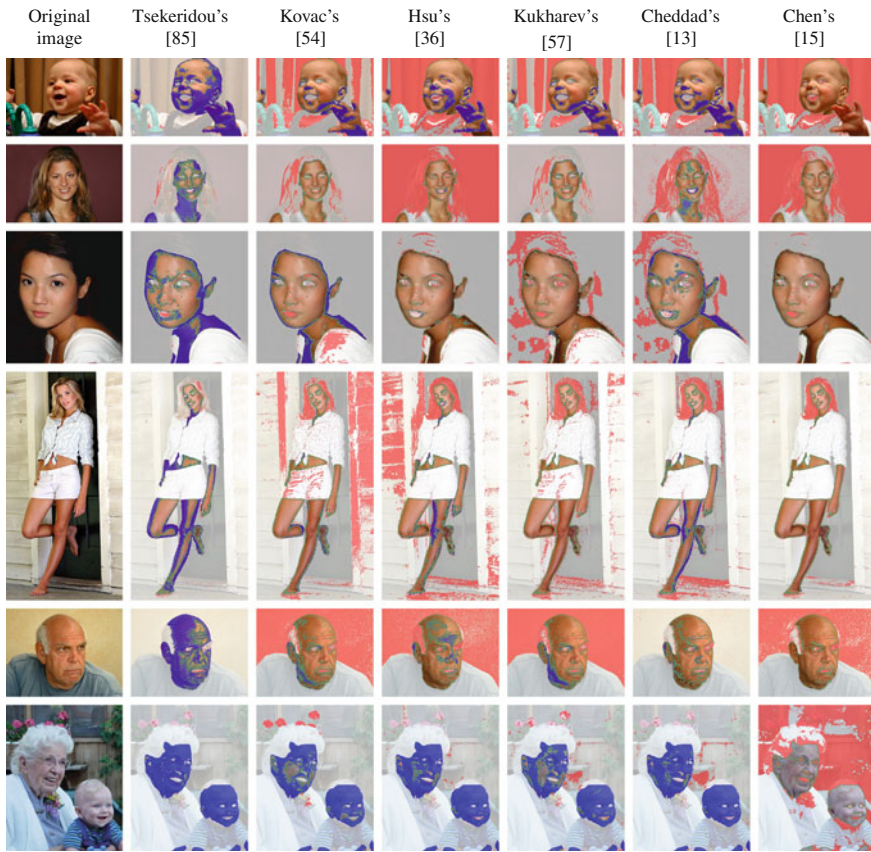


Fig. 4 Skin detection results obtained using different rule-based methods

any prior knowledge concerning the camera characteristics or lighting conditions. It is often argued that the main advantage of the rule-based methods is that they do not require a training set. However, the rules are designed based on observed skin-tone distribution, which means that a form of a training set is required as well. J. Brand and J.S. Mason confirmed in their comparative study that the histogram-based approach to skin modeling outperforms the methods which operate using fixed thresholds in the *RGB* color space [11]. Some machine learning techniques require large amount of training data (e.g. Bayesian classifier), while others are capable of learning effectively from small, but representative training sets. In this section the most important machine learning techniques used for skin detection are presented and discussed.

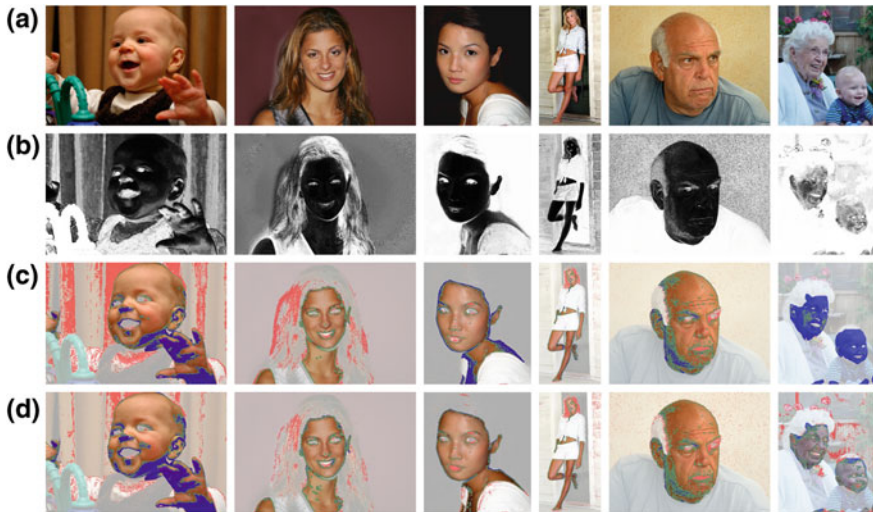
## Bayesian Classifier

Analysis of skin and non-skin color distribution is the basis for many skin detection methods. They may consist in a simple analysis of 2D or 3D histograms of skin color acquired from a training set or may involve the Bayesian theory to determine the probability of observing the skin given a particular color value. Such an approach was adapted by B.D. Zarit et al., whose work [98] was focused on the analysis of the skin and non-skin histograms in two-dimensional color spaces. At the same time, M.J. Jones and J.M. Rehg proposed to train the Bayesian classifier in the *RGB* space using all three components [39, 40]. The main principles of these techniques are as follows.

At first, based on a training set, histograms for the skin ( $C_s$ ) and non-skin ( $C_{ns}$ ) classes are built. The probability of observing a given color value ( $v$ ) in the  $C_x$  class can be computed from the histogram:

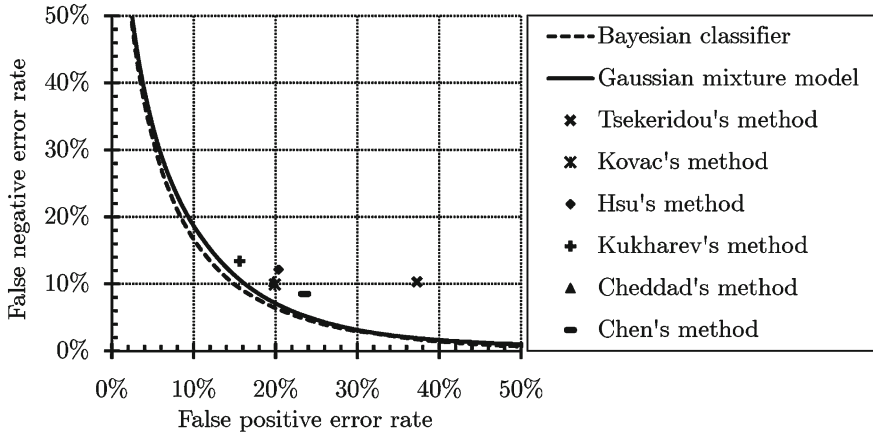
$$P(v|C_x) = C_x(v)/N_x, \quad (6)$$

where  $C_x(v)$  is the number of  $v$ -colored pixels in the class  $x$  and  $N_x$  is the total number of pixels in that class. Maximal number of histogram bins depends on the pixel bit-depth and for most color spaces it equals  $256 \times 256 \times 256$ . However, it is often reported beneficial to reduce the number of bins per channel. Our experiments, reported later in this section, indicated that the optimal histogram bin number depends on the training set size. Basically, the smaller the training set, the smaller number of bins should be used to achieve higher generalization.



**Fig. 5** Skin detection results obtained using Bayesian classifier: original image (a), skin probability map (b), segmentation using a threshold optimized for the whole *ECU-V* data set (c) and using the best threshold determined for each particular image (d)





**Fig. 6** ROC curves obtained for the Bayesian classifier and the Gaussian mixture model, and errors obtained for the rule-based skin detectors

It may be expected that a pixel presents the skin, if its color value has a high density in the skin histogram. Moreover, the chances for that are larger, if the pixel's color is not very frequent among the non-skin pixels. Taking this into account, the probability that a given pixel value belongs to the skin class is computed using the Bayes rule:

$$P(C_s|v) = \frac{P(v|C_s)P(C_s)}{P(v|C_s)P(C_s) + P(v|C_{ns})P(C_{ns})}, \tag{7}$$

where *a priori* probabilities  $P(C_s)$  and  $P(C_{ns})$  may be estimated based on the number of pixels in both classes, but very often it is assumed that they both equal  $P(C_s) = P(C_{ns}) = 0.5$ . If the training set is large enough, then the probabilities  $P(C_s|v)$  for all possible color values can be determined, and the whole color domain is densely covered. For smaller training sets, the number of histogram bins should be decreased to provide proper representation for every bin. The learning phase consists in creating the *skin color probability look-up table* ( $\mathbf{P}_s$ ), which maps every color value in the color space domain into the skin probability (which is also termed as *skinness*). After training, using the look-up table, an input image is converted into a skin probability map, in which skin regions may be segmented based on an acceptance threshold ( $P_{th}$ ). The threshold value should be set to provide the best balance between the false positives and false negatives, which may depend on a specific application. This problem is further discussed in Sect. 7.

Examples of the skin segmentation outcome obtained using the Bayesian classifier are presented in Fig. 6. Original images (a) are transformed into skin probability maps (b) which are segmented using two threshold values, namely globally (c) and locally (d) optimized. The former is set, so as to minimize the detection error for the whole data set, while the latter minimizes the error independently for each image. It can be noticed that the detection errors are smaller than in case of using the rule-based

methods, whose results were shown in Fig. 4. The advantage is also illustrated in Fig. 5 in a form of *ROC* curves. Here, the results obtained for the rule-based methods are presented as points, because their performance does not depend on the acceptance threshold. In the case of the Bayesian classifier, as well as the Gaussian mixture model, which is discussed later in this section, skin probability maps are generated, hence the *ROC* curves may be rendered. It can be seen that the Bayesian classifier outperforms the rule-based methods and also it is slightly better than the Gaussian mixture model.

## Gaussian Mixture Models

Using non-parametric techniques, such as those based on the histogram distributions, the skin probability can be effectively estimated from the training data, providing that the representation of skin and non-skin pixels is sufficiently dense in the skin color space. This condition is not necessarily fulfilled in all situations. A technique which may be applied to address this shortcoming, consists in modeling the skin color using a *Gaussian mixture model* (GMM). Basically, if the histogram is approximated with a mixture of Gaussians, then it is smoothed at the same time, which is particularly important in case of sparse representation. GMM has been used for skin detection in various color spaces, in which a single pixel is represented by a vector  $\mathbf{x}$  of the dimensionality  $d$ , whose value depends on a particular color space. Usually  $d = 2$  or  $d = 3$ , but also skin color was modeled using Gaussians in the one-dimensional spaces [13, 94].

In general, using the adaptive Gaussian mixture model, the data are modeled with a mixture of  $\mathcal{K}$  Gaussian distributions. In the majority of approaches, only the skin-colored pixels are modeled with the Gaussian mixtures, nevertheless non-skin color could also be modeled separately. Thus, in such situations, these two models (i.e. skin and non-skin) are created. Each Gaussian distribution function is characterized with a weight  $\alpha_i > 0$  in the model, where  $\sum_{i=1}^{\mathcal{K}} \alpha_i = 1$ . The probability density function of an observation pixel  $\mathbf{x}$  in the mixture model is given as:

$$p(\mathbf{x}|\Theta) = \sum_{i=1}^{\mathcal{K}} \alpha_i p(\mathbf{x}|i; \theta_i), \quad (8)$$

where  $p(\mathbf{x}|i; \theta_i)$  is the probability for a single Gaussian:

$$p(\mathbf{x}|i; \theta_i) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_i|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_i)^T \Sigma_i^{-1} (\mathbf{x} - \mu_i)\right). \quad (9)$$

Here,  $\alpha_i$  is the  $i$ th Gaussian weight estimation and  $\Theta = (\theta_1, \dots, \theta_{\mathcal{K}})$  is the parameter vector of a mixture composition.  $\theta_i = \{\mu_i, \Sigma_i\}$  consists of  $i$ th Gaussian distribution parameters, that is, the mean value  $\mu_i \in \mathbb{R}^d$  and covariance  $\Sigma_i$ , which is a

$d \times d$  positive definite matrix. The parameters of GMM are estimated based on the expectation-maximization (EM) algorithm.

The EM algorithm is an iterative method for finding the maximum likelihood (ML) function:

$$\mathcal{L}(\mathbf{X}; \Theta) = \prod_{n=1}^N p(\mathbf{x}_n | \Theta). \quad (10)$$

This function estimates the values of the model parameters, so as they best describe the sample data.

The EM algorithm includes two steps, namely:

*Expectation:* Calculate the expected value of the log likelihood function:

$$Q(\Theta | \Theta^{(t)}) = E \left[ \log \mathcal{L}(\mathbf{X}; \Theta) | \mathbf{X}, \Theta^{(t)} \right], \quad (11)$$

where  $\Theta^{(t)}$  is the current set of the parameters.

*Maximization:* Find the parameter that maximizes this quality:

$$\Theta^{(t+1)} = \arg \max_{\Theta} Q(\Theta | \Theta^{(t)}). \quad (12)$$

In this algorithm, the GMM parameters are determined as follows:

$$\hat{\mu}_i^{(t+1)} = \frac{\sum_{n=1}^N p^{(t)}(i | \mathbf{x}_n) \mathbf{x}_n}{\sum_{n=1}^N p^{(t)}(i | \mathbf{x}_n)}, \quad (13)$$

$$\hat{\Sigma}_i^{(t+1)} = \frac{\sum_{n=1}^N p^{(t)}(i | \mathbf{x}_n) (\mathbf{x}_n - \hat{\mu}_i) (\mathbf{x}_n - \hat{\mu}_i)^T}{\sum_{n=1}^N p^{(t)}(i | \mathbf{x}_n)}, \quad (14)$$

$$\hat{\alpha}_i^{(t+1)} = \frac{1}{N} \sum_{n=1}^N p^{(t)}(i | \mathbf{x}_n), \quad (15)$$

$$p^{(t)}(i | \mathbf{x}_n) = \frac{\alpha_i^{(t)} p(\mathbf{x}_n | \theta_i^{(t)})}{p(\mathbf{x}_n | \Theta^{(t)})}. \quad (16)$$

The EM algorithm is initiated with a given number of Gaussian mixtures ( $\mathcal{K}$ ) and the model parameters are obtained using the  $k$ -means algorithm.

As mentioned earlier, skin color has been modeled using GMM in various color spaces. Normalized  $rg$  chromaticity space (i.e.  $\mathbf{x} = [r, g]^T$ ) was used by H. Greenspan et al. [32], who divided the color space into  $\mathcal{K}$  face color regions and a complementary non-face region (which gives  $\mathcal{K} + 1$  decision regions). For each pixel  $\mathbf{x}_n$ , they compute the probability from each Gaussian (8) and the one with the strongest response is selected. When the probability is less than a defined threshold, then the pixel is classified as non-skin, otherwise the pixel is labeled as a member of the selected Gaussian and classified as skin. T.S. Caetano et al. [12] proposed to

compute the probability using the whole mixture model, where the probability for each component is calculated separately. Moreover, they presented and discussed the results if the skin was modeled using different number of Gaussians.

B. Choi et al. presented an adult image detection system which is based on skin color modeling followed by a support vector machines classifier [17]. Here, the Gaussian mixtures operate in the  $RGB$  color space (i.e.  $\mathbf{x} = [R, G, B]^T$ ). Not only is the probability from the skin model ( $P_s$ ) used here, but also the probability of the non-skin model ( $P_{ns}$ ) is considered for classification. If the ratio of  $P_s$  to  $P_{ns}$  is below a defined threshold, then the pixel is classified as non-skin.

S.L. Phung et al. modeled the skin color using three single Gaussians in a 3D color space ( $\mathbf{x} = [Y, C_b, C_r]^T$ ), instead of using the mixture model [65]. These Gaussians correspond approximately to three levels of luminance: low, medium and high. The pixel is classified as skin if it satisfies the following two tests:

1.  $75 \leq C_b \leq 135$  and  $130 \leq C_r \leq 180$ ;
2. The minimum Mahalanobis distance from  $\mathbf{x}$  to the skin clusters (i.e. to the three Gaussians) is below a certain threshold.

M.F. Hossain et al. also used the  $YC_bC_r$  color space for skin regions extraction [35]. They addressed the problem of varying illumination using two skin models defined for normal and bright lighting conditions. The Gaussian mixture model is used only for chrominance ( $\mathbf{x} = [C_b, C_r]^T$ ), which automatically segments the skin portion. Appropriate Gaussian is selected based on the mean value of the luminance  $Y$ .

M.-H. Yang and N. Ahuja transformed the  $RGB$  color space to  $CIELUV$  and discarded the  $L$  component to make the model independent from lighting conditions [92] ( $\mathbf{x} = [u, v]^T$ ). An image region is regarded as a skin area, if more than 70% of the pixels in the field are classified to the skin class.

According to the experiments reported by B.N. Jagadesh et al. [37, 67], the skin distribution is not symmetric and mesokurtic, which means that it cannot be modeled properly using a mixture of Gaussians. They argue that the Pearson distribution of type-IIIb and IVa is most suitable to model skin color distribution in the  $HSI$  color space.

## Artificial Neural Networks

*Artificial neural networks* (ANN) were considered for skin detection by many researchers, and they were used to classify every individual pixel as skin or not based on its value in the color space [9, 23, 71, 82]. Recently, a thorough survey on these methods was published by H.K. Al-Mohair [4]. Not only were ANN used for modeling the skin color itself in different color spaces, but they were also applied to color normalization [42] and model adaptation [59, 90], which is discussed later in Sect. 6.1.

## Support Vector Machines

*Support vector machines* (SVM) [20] were also applied to skin detection. SVM is a robust and widely adopted binary classifier which has been found highly effective for a variety of pattern recognition problems. Based on a labeled training set, it determines a hyperplane that linearly separates two classes in a higher-dimensional kernel space. The hyperplane is defined by a small subset of the vectors from the entire training set, termed *support vectors* (SV). Afterwards, the hyperplane is used to classify the data of the same dimensionality as the training set data. In the case of skin detection, SVM can be used to classify the pixels, represented as vectors in the input color space. Here, the main problem lies in high, i.e.  $O(n^3)$  time and  $O(n^2)$  memory complexity of SVM training, where  $n$  is the number of samples in the training set. Usually, the number of available skin and non-skin pixels is too large to train SVM, and a sort of training set selection must be proceeded. J. Han et al. proposed to use active learning for this purpose [33] which is one of widely adopted approaches towards dealing with huge training sets for SVM [61, 70]. During our works we have found genetic algorithms quite effective for reducing large training sets for the sake of skin detection using SVM [48].

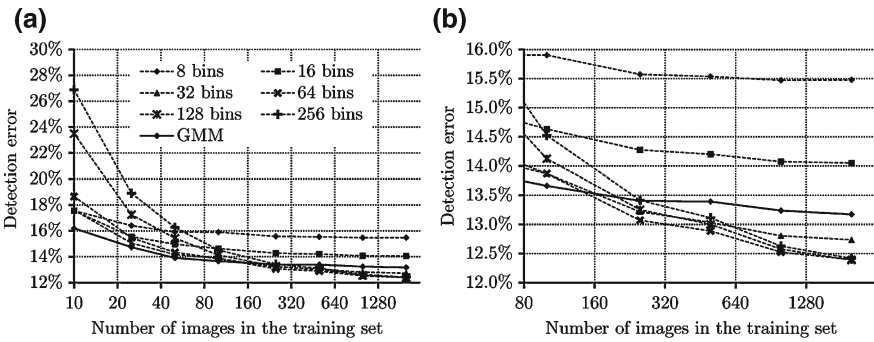
## Conclusions on Using Machine Learning for Skin Detection

The learning schemes discussed earlier in this section are commonly used for skin detection. Apart from them, there were also some attempts to use other popular learning machines. G. Gomez proposed a rule induction method [31] which optimizes the decision rules to minimize the detection error. R. Khan et al. used a random forest [51], trained in the *IHLS* color space [34]. The random forest classifier was claimed to outperform other learning machines, but this may have been attributed to their non-optimal parameters and settings, which are not discussed in that work. Furthermore, size of the training set has not been quoted neither, and it may have fundamental influence on the obtained results. Poor performance of the Bayes classifier quoted in that work may suggest that the training set was small and not representative. Similar objections may be raised in the case of a recent review on using machine learning schemes for skin classification, published by the same authors [52].

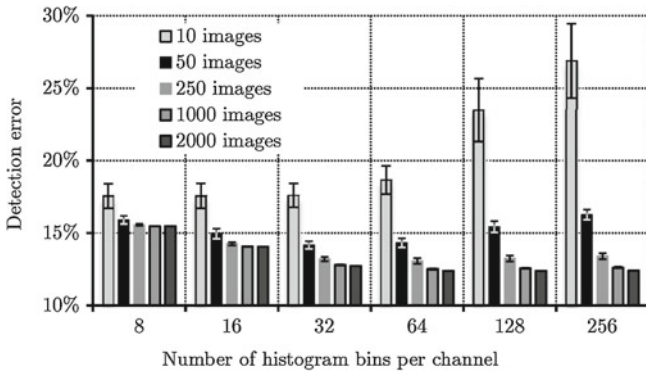
A serious drawback of using advanced machine learning methods (e.g. GMM, ANN, SVM) for skin detection is a long classification time, which usually excludes real-time applications. However, as the dimensionality of the input data is low (i.e. two or three dimensional color space) and the resolution of each dimension is usually limited to 256 values, it is feasible to build a look-up table once the classifier has been trained. This provides an effective optimization of the execution time, however it requires additional computations during the training phase which are usually acceptable.

It was observed that for sufficiently large training sets the effectiveness of skin detection is similar for different classifiers [64]. However, in case of small training sets, some classifiers may better generalize than others, and this was also studied

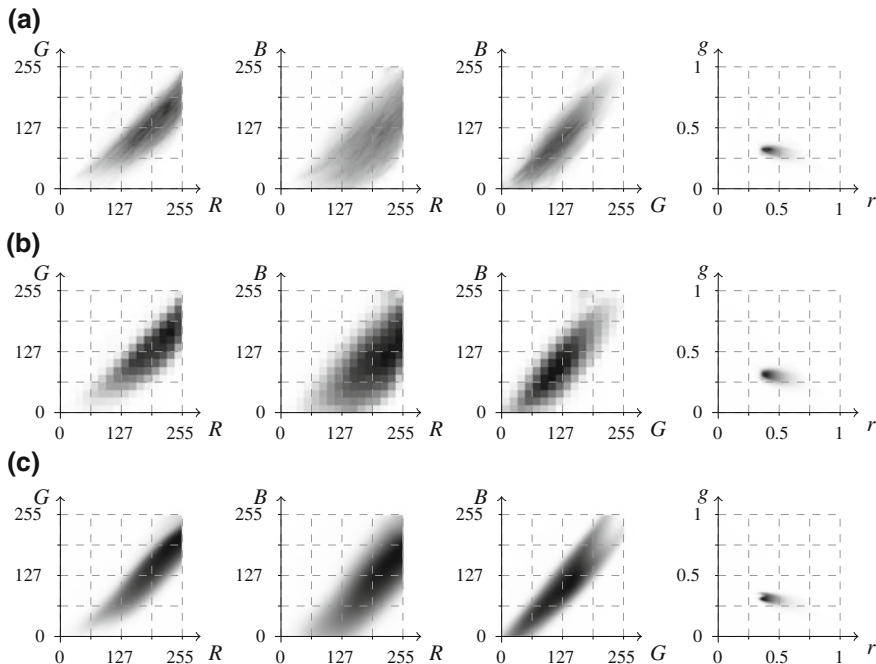
during our research. In our study we investigated the performance of the Bayesian classifier and Gaussian mixture model depending on the number of images in the training set and color space quantization for the Bayesian classification. In the case of GMM, the skin distribution was modeled using two Gaussians, whereas three Gaussians were used to approximate the non-skin distribution. The dependence between the skin detection error ( $\delta_{min}$ ) and the train set size is presented in Fig. 7, where the Bayesian classifier was trained in the *RGB* color space quantized to six different numbers of bins per channel. For the training sets smaller than 2000 images, each presented result was averaged based on the scores obtained using six randomly selected subsets of the *ECU-T* data set. The errors and their standard deviation obtained for five different sizes of the training set are also presented in Fig. 8. Skin distribution learned from the entire available training data is also presented in Fig. 9. It can be observed from Fig. 7 that for small sizes of training sets, it is beneficial to decrease the number of histogram bins per channel, but the improvement is limited using these sets when more training data are available. GMM performs definitely better than the



**Fig. 7** Dependence between the skin detection error ( $\delta_{min}$ ) and the training set size for GMM and Bayesian classifier with a different number of histogram bins. The whole investigated range is presented in (a), and the results for training sets larger than 100 images are shown in (b)



**Fig. 8** Skin detection error ( $\delta_{min}$ ) obtained using the Bayesian classifier depending on the color space quantization and training set size



**Fig. 9** Skin probability distribution presented in  $RG$ ,  $RB_gGB$  and  $r - g$  planes modeled using Bayesian learning with 256 (a) and 32 (b) bins per channel, and GMM with 2 skin and 3 non-skin Gaussians (c). Bayesian classifier (256 bins per channel) (a). Bayesian classifier (32 bins per channel) (b). Gaussian mixture model (2 skin and 3 non-skin Gaussians) (c)

Bayesian classifier for training sets containing less than 100 images, but for large training sets it gets slightly outperformed by the latter. Here, we used two Gaussians to model skin pixels and three Gaussians to model non-skin pixels. Generally, the more Gaussians are used, the better results can be obtained for larger training sets, however we have not observed any advantage over the Bayesian classifier in these cases.

## 5 Textural Features

Analysis of textural features has been investigated in order to improve the performance and to enhance the color-based methods. It is usually applied as the second step in skin segmentation in color images. X. Wang et al. combined the segmentation in the  $RGB$  and  $YC_gC_b$  color spaces with the analysis of the textural information extracted from an image [89]. Firstly, the white balance in the  $YC_bC_r$  color space is performed before the skin modeling to minimize the influence of the external conditions on the color information. The color model in the  $RGB$  color space is defined

by a set of fixed rules. The authors investigated the skin distribution in various color spaces and noticed that the distribution in the  $YC_gC_b$  space is regular and of a circular shape. Thus, the  $YC_gC_b$  space was chosen to define the second skin model. In the proposed method, the detection outcome obtained by anding two resulting binary images for each color space is further improved by incorporating the texture analysis. The grey-level co-occurrence matrix (GLCM) is used to extract the textural features. Given a grey-scale image  $I$  of size  $n \times m$ , the GLCM  $P$  is defined as:

$$P(i, j) = \sum_{x=1}^n \sum_{y=1}^m \begin{cases} 1, & \text{if } I(x, y) = i \wedge I(x + \Delta_x, y + \Delta_y) = j, \\ 0, & \text{otherwise,} \end{cases} \quad (17)$$

where  $(\Delta_x, \Delta_y)$  is the offset between the pixels  $I(x, y)$  and  $I(x + \Delta_x, y + \Delta_y)$ . It is worth noting that the time complexity of determining the GLCM is dependent on the number of grey levels  $g$  and is proportional to  $O(g^2)$  [18], thus  $g$  must be reduced for the real-time processing. The textural features extracted from the GLCM, including contrast, entropy, angular second moment, correlation and homogeneity are finally used for further skin detection. According to the experiments performed for a set of 500 images with the complex background, the method resulted in significantly worse skin detection rate (80.1%) comparing to the detection in the  $RGB$  (90.3%) and  $YC_gC_b$  (83.7%) color spaces only. However, it turned out to be effective in detection of the background and outperformed (88.4%) the background detection rates obtained for  $RGB$  (71.1%) and  $YC_gC_b$  (82.4%).

P. Ng and C.M. Pun proposed a method combining a color-based segmentation with the texture analysis using 2-D Daubechies wavelets [62]. Here, the Gaussian mixture model (GMM) classifier is applied as the initial skin segmentation approach for the original  $RGB$  images. Then, the 2-D Daubechies wavelets are calculated for the sub-images around each pixel classified as a skin pixel. The authors introduced the mask-leakage permit concept allowing for the efficient selection of sub-images. The texture feature of each skin pixel is represented by the wavelet energy vector  $\mathbf{v}_e$  obtained by applying the Shannon entropy on the wavelet coefficients vector  $\mathbf{v}_c$ . The  $\mathbf{v}_e$  vectors are the observation data of the  $k$ -means clustering. Given a set of  $n$  observations  $(\mathbf{v}_e^1, \mathbf{v}_e^2, \dots, \mathbf{v}_e^n)$ , the aim of the  $k$ -means clustering is to divide the observations into  $k$ ,  $k < n$ , clusters minimizing the total intra-cluster variance. Finally, some clusters that are claimed to be the non-skin ones are eliminated based on the properties of the Shannon entropy. According to the experimental results showed for the set of 100 images selected randomly from the Compaq database, the proposed method did not improve the segmentation significantly. The false positives (21.6%) were decreased by approximately 3% comparing to the GMM classifier. However, the true positives (86.8%) dropped by approximately 2%.

The approach integrating color, texture and space analysis was described by Z. Jiang et al. [38]. Here, the skin probability map (SPM) color filter is used in the  $RGB$  color space. The method does not rely only on the color information, thus the authors lower the threshold  $\Theta$  used as the acceptance threshold for the color filter in order to determine all probable skin pixels, which results in increasing the false positives. A filter utilizing the textural features extracted using the Gabor wavelets is proposed



to reduce the false acceptance rate (FAR). A 2-D Gabor function  $g(x, y)$  is given as:

$$g(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi j W x\right), \quad (18)$$

where  $\sigma_x$  and  $\sigma_y$  are the scaling parameters and  $W$  is the central frequency of the filter. Let  $g(x, y)$  be the generating function of the Gabor filter family. The Gabor functions (wavelets)  $g_{m,n}(x, y)$ ,  $m = 0, 1, \dots, M - 1$ ,  $n = 0, 1, \dots, N - 1$ , are created by rotating and scaling of  $g(x, y)$ :

$$g_{m,n}(x, y) = a^{-m} g(x', y'), \quad (19)$$

where  $x' = a^{-m}(x \cos \theta + y \sin \theta)$ ,  $y' = a^{-m}(-x \sin \theta + y \cos \theta)$ ,  $\theta = n\pi/N$ ,  $a > 1$ ,  $M$  is the total number of scales and  $N$  is the total number of orientations. Given an image  $I$ , the texture feature of each pixel is calculated as:

$$T(x, y) = \frac{\sqrt{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} J_{m,n}^2(x, y)}}{MN}. \quad (20)$$

Here,  $J_{m,n}$  is the image Gabor wavelets transform defined as:

$$J_{m,n}(x, y) = \sum_{x_1} \sum_{x_2} I(x_1, x_2) g_{m,n}^*(x - x_1, y - y_2), \quad (21)$$

where  $g_{m,n}^*(x, y)$  is the complex conjugate of  $g_{m,n}(x, y)$ . The grey-scale image  $I$  is obtained by transforming the input  $RGB$  image:

$$I(x, y) = 0.3R(x, y) + 0.59G(x, y) + 0.11B(x, y). \quad (22)$$

Finally, the texture mask image is defined as:

$$M(x, y) = \begin{cases} 1, & \text{if } T(x, y) \leq \Theta_T, \\ 0, & \text{otherwise,} \end{cases} \quad (23)$$

where  $\Theta_T$  is the texture threshold. On the one hand, applying the texture filter results in decreasing the FAR. However, on the other hand, it can also filter out properly classified skin pixels, which leads to reducing the true acceptance rate (TAR). It is worth noting that the performance of the texture filter is limited in case of textured skin regions (e.g. wrinkles). Additionally, if the image background is as smooth as skin then its pixels are not filtered out, thus the false positives are not reduced (see examples in Fig. 10). The authors proposed to exploit the space information after the color and texture analysis to grow the skin regions by using the watershed segmentation with the region markers. Thus, the markers are selected according to the following rules: (1) if the pixels either do not pass the color filtering or have a



**Fig. 10** Skin detection results obtained using the SPM color filter and refined using the Gabor wavelets: original image (a), skin probability map (b), texture image (c), segmentation using the threshold  $\Theta = 180$  (d), image after applying the texture mask for various  $\Theta_T$  ( $20 \leq \Theta_T \leq 120$ ) (e)

large texture value, then they are set as the non-skin local minima markers, (2) if the pixels are classified by both color and texture filters as the skin pixels, then they are set as the skin local minima markers. The mean and the standard deviation are calculated for the resulting closed regions. Finally, the pixels are classified as the skin pixels if they either pass the initial color filtering or the mean and the standard deviation of the region that encloses the pixel are smaller than the given thresholds. The experimental results obtained for the database containing 600 images collected by the authors proved that the method reduces the FAR significantly (from 20.1 to 4.2 %) with the simultaneous increase in the TAR (from 92.7 to 94.8 %) comparing to the SPM approach. The authors did not provide any sensitivity analysis, thus it is unclear how to set the parameters of the algorithm which turns out to be its significant drawback.

A. Taqa and H.A. Jabab increased performance of the ANN-based skin classifier by including simple textural feature [82]. It is reported in the paper that although the textural features have very low discriminating power, they are capable of improving stability of the color-based models.

A. Conci et al. [19] used the spectral variation coefficient (SVC) to extract textural features relevant to differentiate skin pixels from non-skin background regions for

the  $RGB$  images. In this approach, the relation between the pixel positions on the texture element (i.e. a *texel*) of size  $M \times M$ ,  $M = 3, 5, \dots, 21$ , is considered. Firstly, the color intensities for each  $R$ ,  $G$  and  $B$  channels are blended together according to the defined function and form the new channel values. Next, the SVC values are computed to determine the average and the standard deviation of the distances within a texel according to the given metric. Here, the  $D_4$  (Manhattan) metric is used:

$$D_4(p_1, p_2) = |x_1 - x_2| + |y_1 - y_2|, \quad (24)$$

where  $p_1$  and  $p_2$  are the pixels at the positions  $(x_1, y_1)$  and  $(x_2, y_2)$  respectively. It is worth noting that the number of  $D_4$  distance classes to consider is dependent on the texel size  $M \times M$ . The SVC values are finally calculated for each blended channel and for each class of distances:

$$SVC = \arctan\left(\frac{\mu}{s+1}\right) \sqrt{\mu^2 + (s+1)^2}, \quad (25)$$

where  $\mu$  is the average and  $s$  is the standard deviation of the blended channel. They are considered as the coordinates in the Euclidean space and the  $k$ -means clustering is applied for the sample classification. The proposed method was tested using four color images. The authors claim that the algorithm reduces the false negatives, however to obtain meaningful results it would be necessary to perform the validation on a larger data set.

A technique combining a Bayesian skin color classifier based on the non-parametric density estimation of skin and non-skin classes with a multi-scale texture analysis was proposed by B. Zafarifar et al. [97]. Here, the authors point out that the reflections and shadows can cause sudden changes in the luminance channel of a color image. According to this, a new textural feature is defined:

$$T(i, j) = [|Y(i, j-1) - 2Y(i, j) + Y(i, j+1)| + |Y(i-1, j) - 2Y(i, j) + Y(i+1, j)| - \tau]_0^l, \quad (26)$$

where  $Y(i, j)$  is the value of the luminance channel,  $\tau$  is the noise threshold and  $f(x) = [x]_0^l$  clips the value of  $x$  between 0 and  $l$ . Finally, the extracted features are exploited to determine the final segmentation output. The experimental results obtained for 173 annotated natural images showed the decrease in the false positive rates. The disadvantage of this method is its limited performance for heavily textured skin areas (e.g. skin of the elders).

Simple textural features were used by D.A. Forsyth and M.M. Fleck [26] in a system for human nudes detection. Here, the difference between the original and median-filtered intensity image indicates the texture amplitude. Small amplitude increases the skin probability, computed in the log-opponent color space derived from  $RGB$ . A very similar approach was adopted by A.A. Abin et al. [3], who improved the median-based filtering by measuring the variance in  $R$ ,  $G$  and  $B$  channels. Results of the median-based and variance-based filters are combined with each other and used

to refine the color-based skin probability map. The refined probability map serves as an input to a cellular learning automata which is further discussed in Sect. 7. M. Fotouhi et al. proposed to extract textural features using a contourlet transform [27]. The input image is split into small patches, in which the contourlet features are computed in small patches that surround the skin-color pixels. The contourlet feature vectors are later subject to principal components analysis to reduce their dimensionality and classified using a multilayer perceptron with two hidden layers. The main disadvantage of this method is that it is extremely slow compared with alternative approaches (ca. 30 min per  $256 \times 256$  image), while the effectiveness is not competitive, following the results quoted in [27].

In general, simple textural features can be applied to skin detection, taking advantage from the observation that skin regions are usually smooth. As it is the absence of any textural features rather than the presence of their characteristic indicators, it is sufficient to extract relatively simple features. This makes it possible to reject those regions, in which skin colored pixels appear, but their roughness indicates that the particular region does not present human skin. Obviously, such regions would be misclassified using pixel-wise skin color models. The roughness may be manifested in the luminance and color channels, but it can also be most significant only in the skin probability map as observed in our earlier research [45, 47].

## 6 Adaptive Models

As it was explained earlier in Sect. 4, the performance of skin color models is limited because of an overlap between the skin and non-skin pixels in the color space. Basically, the more general the model is supposed to be, the larger the overlap is, and the more it does degrade the detection score. The skin model may be made more specific to a given scene, video sequence, or an image, if they were acquired in fixed lighting conditions and present the same individuals. In such cases, the overlap is usually much smaller and the discriminative power of a skin color model may be definitely higher. However, it is not a trivial task to optimize a color model, and there is no universally effective adaptation procedure. There have been a number of methods proposed which address the problem of adapting a color model, and they can be categorized taking into account two criteria, namely the information source for the adaptation, and the model type which is being adapted. A general classification

**Table 2** Classification of model adaptation methods

		Information source for the adaptation		
		Tracking	Whole-image analysis	Face and hand detection
Model used for the adaptation	Threshold-based method	[21]	[59, 66]	[10, 60, 95]
	Histogram analysis	[73, 77]	[6, 99]	[43]
	Gaussian mixture	–	[80, 90, 100, 101]	[28, 78]

of these methods is presented in Table 2. Later in this section three general groups are described—self-adaptation techniques, which adapt the model to an image using the information obtained by the universal model, and content-based techniques, which rely on other data sources delivered by object detectors.

## 6.1 Global Adaptation Techniques

S.L. Phung et al. observed that an optimal value for the acceptance threshold applied in the probability map depends on the particular image [66]. Also, they argue that a coherent skin region should have homogenous textural features, extracted based on the  $3 \times 3$  Sobel operator. Hence, the acceptance threshold is iteratively adapted, so as to maximize the skin blobs homogeneity.

M.-J. Zhang and W. Gao used ANN for determining an optimal acceptance threshold in a given skin probability map obtained with the Bayes classifier [99]. It is assumed that the threshold should be located in a local minimum of the probability image histogram. Hence, for every minimum, 13 features are extracted, which are regarded as a feature vector that is classified by the neural network. In this way, for every image an optimal threshold is determined which decreases the detection error.

A.A. Argyros and M.I.A. Lourakis [6] proposed to apply a globally-trained Bayesian classifier to extract skin region seeds using a high acceptance threshold. Such classification results in low false positive rate at a possible cost of high false negative error. After that, they apply a threshold hysteresis to expand the skin regions, which are later used to learn the Bayesian classifier. This creates a local skin model, which is combined with the global model, and used to classify the entire image. Finally, the method is applied to track the skin-colored objects, but it is the spatial analysis rather than tracking which delivers the data for the adaptation.

Q. Zhu et al. proposed a double-level approach towards adapting the skin model [100, 101]. First, the skin pixels are detected using a generic model characterized with a very low false negative rate, achieved at a cost of high false positives. Afterwards, the potential skin pixels are modeled using two Gaussians—one is supposed to represent the skin pixels, while the other is expected to be formed by false positives. Color, spatial and shape features are extracted for each Gaussian and the feature vectors are classified using SVM to select the Gaussian which represents the real skin pixels.

J.-S. Lee et al. proposed a method based on a multilayer perceptron which accommodates to specific lighting conditions observed in the input image [59]. At first, for each image in the training set, a chroma histogram of skin pixels is extracted. After that, the chroma histograms for all of the images were merged and grouped, which finally delivered five major chroma clusters. Hence, every cluster represents a number of images from the training set, which, according to the authors, were acquired under similar lighting conditions. Then, a multilayer perceptron was trained with image patches to decide which chroma cluster should be applied to a particular image. Hence, for every input image the patches around the image center are classified by the neural network, and based on their response, the most appropriate color

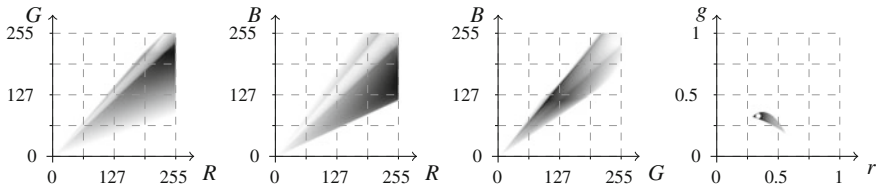
model is applied. In this way, the method adapts to a closed set of predefined lighting conditions.

ANN were also used for the adaptation by G. Yang et al. [90]. They observed that skin color distribution in the  $C_r C_b$  plane depends on the luminance ( $Y$ ), and it is more compact if the luminance value range is smaller. Hence, they used a neural network to determine the parameters of a Gaussian (i.e. the mean and standard deviation) based on the image histogram in  $Y$ . The Gaussian is subsequently used for skin detection. Unfortunately, the authors did not present any comparison with non-adaptive modeling in the three-dimensional  $Y C_b C_r$  space.

An interesting adaptation technique was proposed by H.-M. Sun [80]. First of all, a histogram-based global skin model is trained, and it is applied to every input image to detect skin pixels. Then, the pixels classified as skin are subject to clustering in the color space and their distribution is modeled using a Gaussian mixture. The number of clusters and GMM's parameters are determined based on the distribution of the pixels classified as skin using the global model. Finally, the entire image is classified again based on a linear combination of the dynamically learned GMM and the global model. This combination stabilizes the result and is more effective than relying exclusively on the dynamic local model. The local adaptation makes it possible to decrease the skin probability of the pixels which do not form clusters in the color space. Usually, the groups of skin pixels in the input image are characterized with similar chrominance, which results in observing clusters in the color space. Contrary to that, the false positives are often isolated and will not be detected as a cluster during the adaptation phase.

## 6.2 Tracking-Based Adaptation

M. Soriano et al. proposed to adapt the model dynamically based on observed changes in a histogram extracted from the tracked skin region [77]. Initially, the skin is modeled in the  $rg$  color space based on the image histograms. The *skin histogram* values are divided by the corresponding values from the *whole-image histogram*. This produces a *ratio histogram* which is further used for classification. The ratio histogram can be defined only for non-zero whole-image histogram values, and the remaining values are assigned with zero skin probability. The detected skin regions are tracked in video sequences and the ratio histogram is adjusted to compensate the changes in lighting conditions that influence the skin appearance. The tracking procedure consists in determining the largest connected component of the pixels classified as skin, which lies inside an expanded bounding box of a facial region detected in the previous frame. From this region, the skin histogram is generated, and the ratio histogram is computed using that histogram as well as the histogram of the entire image. Using the updated ratio histogram, the final position of the facial region is determined. However, there is a risk that if the tracking works incorrectly, then the model gets destabilized at some point. In order to prevent such a situation, it is assumed that a pixel can be classified as skin only if it falls inside a certain skin



**Fig. 11** Skin locus which determines the adaptation boundary in the Soriano’s method [77]

locus, limited by two quadratic functions defined in  $rg$  color space, as presented in Fig. 11. Most of the skin pixels fall inside the locus indeed, however it cannot be used directly as a decision surface for the classification. Otherwise, the false positives are very high.

L. Sigal et al. addressed the problem of varying lighting conditions which may severely affect the skin detection efficacy [73]. They used the expectation maximization algorithm to adapt the histogram-based classifier based on tracked skin regions.

F. Dadgostar proposed to take advantage of the motion detectors to adapt the skin color model [21]. Here, the skin color is defined by the lower and upper Hue thresholds in the  $HSV$  color model. If the pixel’s Hue value is positioned between these thresholds, then it is considered as skin. Hue histogram of the in-motion skin pixels is extracted and analyzed to determine the optimal values of the thresholds. In this way the skin model is adapted to a scene (i.e. the lighting conditions and an individual, whose skin is to be detected) and skin regions can be segmented with higher effectiveness.

### 6.3 Face or Hand Region-Based Adaptation

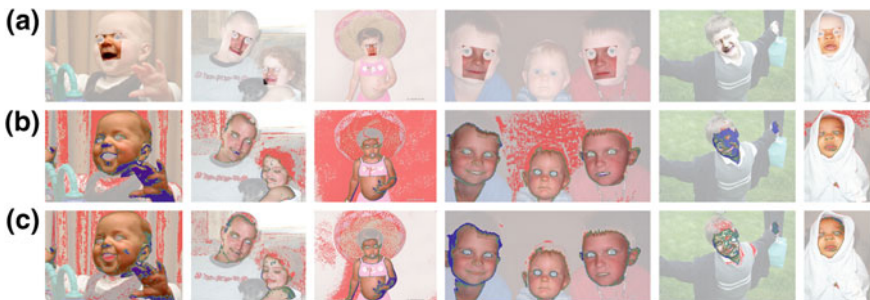
There are methods which effectively adapt the skin model to local conditions based on the detected faces. The first such an approach was proposed in 2002 by J. Fritsch et al. [28]. The adaptation scheme is similar to the Soriano’s method [77], discussed earlier in the chapter. Here, the skin pixels are modeled using a unimodal Gaussian defined in the  $rg$  color space. The Gaussian’s parameters are updated based on a facial region obtained using a face detector, which operates in the luminance channel. Similarly to [77], the adaptation is stabilized using a skin locus. The pixels from the facial region which do not lie inside the locus are ignored for the adaptation. Also, the adaptation works with a certain inertia, and the updated parameters are determined as a weighted mean of the current values and those extracted from the facial region.

H. Stern and B. Efron proposed to dynamically select the best color space, in which the skin model can be learned based on a detected facial region [78]. First, the color space is selected ( $RG$ ,  $rg$ ,  $HS$  or  $C_bC_r$ ), in which the skin pixels from the facial region can be best separated from the non-skin pixels around the region. After choosing the color space, the skin color is modeled using a single Gaussian.

R. Khan et al. combined the adaptive modeling based on face detection with spatial analysis using graph cuts [50]. Here, the faces are used as foreground seeds during the segmentation procedure. The main drawback, however, is the processing time of 1.5 seconds for small  $100 \times 100$  images. S. Bilal et al. used both faces and hands detected using Haar features [88] to define the skin color ranges in  $C_b$  and  $C_r$  channels of the  $YC_bC_r$  color space [10]. Such a local model was later applied to all of the subsequent images registered in a video sequence, which improved the hand tracking algorithm. J.F. Lichtenauer et al. used the positive skin samples acquired from detected faces to adapt the skin model in an introduced adaptive chrominance space [60].

In our earlier works we also proposed to adapt the Bayesian skin classifier based on detected facial region [43]. The faces were detected using our face detector [49], which is capable of detecting the eyes with higher precision than the Viola-Jones method [88]. Based on the positions of the eyes, the face region is created as presented in Fig. 12a. The pixels which fall inside this region are regarded as skin, hence the skin histogram is generated. We reported that while the global model keeps 64 bins per channel, the skin is best modeled locally using only 16 bins, and then it is extrapolated to the resolution of 64 bins per channel. The probability look-up tables for the local and global model are afterwards combined with each other and applied to the whole image. Examples of segmentation results are presented in Fig. 12. Although the detection errors have not been eliminated, and in some cases they are still high, the improvement is significant, compared with the results obtained using the global model (Fig. 12b).

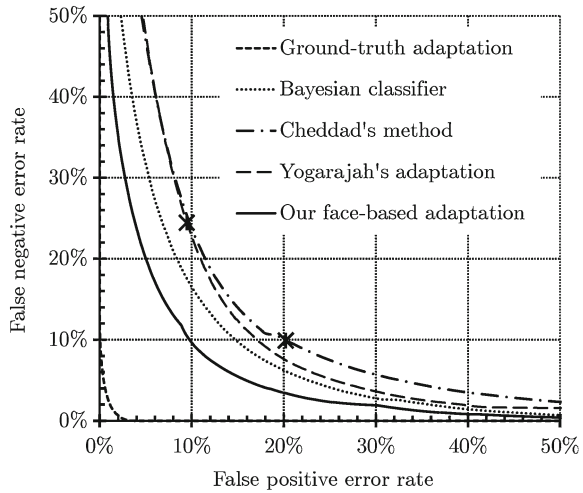
P. Yogarajah et al. presented a method for dynamic adaptation of the thresholds in the method proposed by A. Cheddad [13]. The thresholds in the single-dimensional error space are set dynamically based on the detected facial region [94, 95]. First, the error signal value distribution in the facial region is modeled using a single Gaussian. Then, the thresholds are set in the same way as proposed in [13]. Performance of this method is presented and compared with our earlier approach in Fig. 13 in the form of *ROC* curves. Here, the results are presented for 1375 images from *ECU-V* set, in which the faces were detected. Although Cheddad's and Yogarajah's methods offer binary classification, the probability maps can be easily generated by scaling



**Fig. 12** Detected facial regions (a) and skin segmentation result obtained using the Bayesian classifier (b) and adaptive facial region-based method [43] (c)



**Fig. 13** ROC curves obtained for two global models and two face-based adaptation techniques. Ground-truth adaptation is presented as well. Binary decision result for Cheddad's and Yogarajah's methods are marked with asterisks on the curves



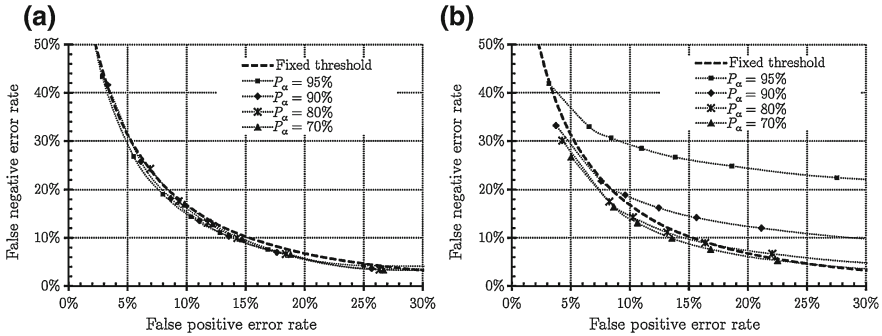
the distance from the approximated Gaussian mean. In this way, the ROC curves can be rendered, and the binary-decision results (which are obviously positioned on the curve) are marked in the figure with asterisks. It may be seen that both adaptation techniques offer some improvement, however it is much more significant using our approach [43]. Yogarajah's method was recently extended by W.R. Tan et al. [81], who combined the threshold-based adaptation with GMM modeling.

## 7 Spatial Analysis Methods

The limitations of the color-based skin models were investigated and reported by Q. Zhu et al. [100, 101]. They have shown that even if the color model is perfectly adapted to every individual image, the skin pixels usually cannot be separated from the non-skin pixels. This boundary is illustrated in Fig. 13 in a form of the ground-truth adaptation. Here, the Bayesian classifier was trained based on every individual image using the ground-truth information, and after that the learned rules were applied to the very same image. The detection errors are very small, but nevertheless they illustrate the limitations of the pixel-wise detectors. Naturally, the errors are much higher when the used skin color model is not perfectly adapted to the image.

The limits of pixel-wise detectors can be quite effectively addressed by analyzing the spatial alignment of the pixels classified as skin. It may be observed that skin pixels usually form consistent blobs in the images, while many false positives are scattered in the spatial domain. This general observation underpins several skin segmentation methods outlined in this section.

H. Kruppa et al. assumed that in most cases the skin regions are of an elliptical shape. Hence, they proposed a skin model [56] which maximizes the mutual infor-



**Fig. 14** ROC curves obtained using a threshold hysteresis without (a) and with (b) size-based verification

mation between color-based and shape-based probability. The model is iteratively adapted for every image using a gradient descent optimization.

As it was mentioned in Sect. 6.1, A.A. Argyros and M.I.A. Lourakis [6] used a threshold hysteresis to perform skin regions segmentation. A similar technique has also been applied by H. Baltzakis et al. for tracking skin regions in video sequences [7]. This operation offers a certain reduction of the detection errors. First, the pixels which exceed a high probability threshold ( $P_\alpha$ ) are considered as seeds for the region growth procedure. Next, the adjacent pixels, whose skin probability is over the second, lower threshold ( $P_\beta$ ), are iteratively adjoined to the region. ROC curves obtained for different  $P_\alpha$  are presented in Fig. 14a. Each curve was rendered on the errors obtained using different values of the lower acceptance threshold  $P_\beta$ . It can be observed that the higher  $P_\alpha$  is, the lower false positive error ( $\delta_{fp}$ ) is obtained in the seeds (i.e. the initial point of each curve with the smallest false positive value). The curves obtained using the threshold hysteresis are positioned slightly below the fixed-threshold curve, and for  $\delta_{fp} < 15\%$ ,  $P_\alpha = 95\%$  offers the largest improvement. During our research [44] we proposed to verify the skin seeds based on their relative size. The verification consists in rejecting those seeds, whose area is smaller than 10% of the largest skin seed. ROC curves obtained following this procedure are presented in Fig. 14b. Here, the initial points are located below the fixed-threshold curve, hence it may be concluded that the small rejected seeds contain a significant amount of false positives. However, it can be noticed that if  $P_\alpha$  is high, then the false negatives decrease very slowly compared with the false positives increase, which positions the results over the fixed-threshold curve. The reason is that some skin regions contain only a small number of pixels with probability exceeding  $P_\alpha$ , and they are rejected during the verification. This points out that if  $P_\alpha$  is too large, then many skin regions will be excluded, and if it is too small, then the false positive error may be large already in the seeds, which by definition cannot be decreased using any region-growth algorithm. From the presented curves it may be concluded that the verification step is beneficial, because the error reduction is larger, but  $P_\alpha$  should not exceed 80% in that case.

J. Ruiz-del-Solar and R. Verschae proposed to perform a controlled diffusion in color or skin probability domain [75]. This procedure consists of two general steps: (1) diffusion seeds extraction, and (2) the proper diffusion process. The seeds are extracted using pixel-wise skin probability maps, and they are formed by those pixels, whose skin probability exceeds a certain high threshold ( $P_\alpha$ ). During the second step, the skin regions are built from the seeds by adjoining the neighboring pixels which meet the diffusion criteria, defined either in the probability map or color space domain. These criteria are as follows: (1) distance between a source pixel  $x$  and a pixel  $y$  (that is to be adjoined) in the diffusion domain  $D_d$  is below a given threshold:  $|D_d(x) - D_d(y)| < \Delta_{max}$ , and (2) skin probability for the pixel which is to be adjoined must be over a certain threshold:  $P(y) > P_\beta$ . It is worth to note that this is the threshold hysteresis with an additional constraint imposed on a maximal difference between the neighboring pixels (either in terms of their probability or color). Hence, this works well if the region boundaries are sharp (diffusion stops due to high local differences), but the method is identical to the threshold hysteresis when there exists a smooth transition between the pixel values that leads from one region to another (the diffusion will then “leak” outside the region).

Spatial analysis of skin pixels was also the subject of our research. At first, we proposed an energy-based scheme for skin blob analysis [44]. Skin seeds are formed by high-valued pixels in the skin probability maps, similarly as in the diffusion method. In addition, we perform the size-based verification as explained earlier in this section. It is assumed that the seed pixels receive a maximal energy amount equal to 1, which is spread over the image. Amount of the energy that is passed depends on the probability value of the target pixel. If there is no energy to be passed, then the pixel is not adjoined to the skin region. Although this method implements a cumu-

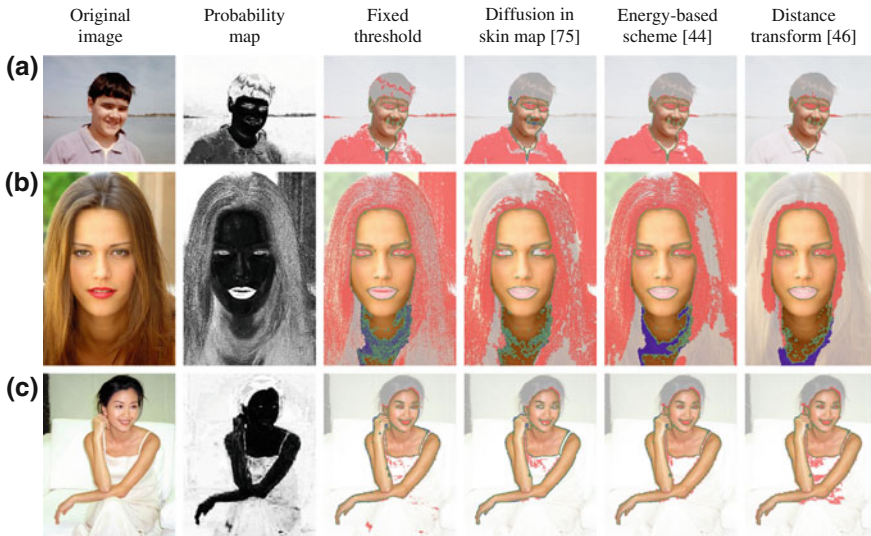
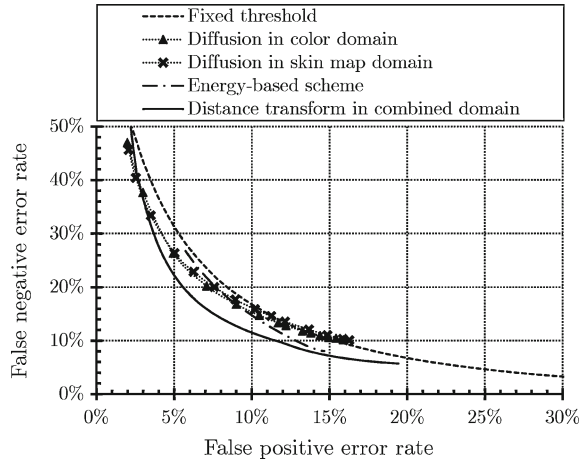


Fig. 15 Examples of skin detection results obtained using different spatial analysis methods

**Fig. 16** *ROC* curves obtained using spatial analysis methods



lative propagation (which helps reduce the “leakages” identified in the controlled diffusion), only skin probability is taken into account and local differences between the pixels are ignored. Recently, we have proposed to rely on the distance transform proceeded in a combined domain of hue, luminance and skin probability [46]. This approach is more effective than the energy-based scheme as the latter uses only skin probability during the propagation. Moreover, it has also a serious advantage over the Solar’s method, because the cumulative character of the distance transform perfectly addresses the main shortcoming of the diffusion, i.e. the vulnerability to smooth transitions between skin and non-skin regions. *ROC* curves obtained for different spatial analysis methods are presented in Fig. 15 and some qualitative results are shown in Fig. 16. For images (a) and (b) the distance transform in a combined domain reduces the errors significantly, while the alternative spatial analysis techniques do not offer much improvement. In the case of image (c), the energy-based scheme delivers the best result, however the errors are small for all of the tested methods.

There were also many other approaches towards using spatial analysis for improving skin detection. A.A. Abin et al. used the cellular automata to determine skin regions [3], but this process requires many iterations to achieve satisfactory results and cannot be applied for real-time processing. Also, K. Chenaoua and A. Bouridane used conditional random fields [55] to take advantage from spatial properties of skin regions [16]. However, this may be a time-consuming procedure as it involves simulated annealing for every analyzed image. M. Abdullah-Al-Wadud [2] proposed to transform an image into a single-dimensional color distance map, in which a water-flow procedure is carried out to segment skin regions. A.Y. Dawod used basic edge detectors to determine the boundaries of skin regions [22]. Z. Yong-jia et al. investigated the possibilities of using level sets for determining the boundaries of skin regions in a given skin probability map [96]. Unfortunately, it is difficult to conclude from the paper whether the method offers any advantage over other techniques.

## 8 Summary

There are plenty of approaches towards skin detection and segmentation, and the most relevant out of the existing techniques were outlined in this chapter. For virtually all of the methods, skin color modeling is the primary source of information here, because it is the color which constitutes the principal discriminative feature of human skin. Although skin detection is still an open problem and no satisfactory solution has been developed so far, in our opinion there is not much to be gained in the skin color modeling itself. This is caused by the overlap between the skin and non-skin pixels that can be observed in all of the popularly used color spaces. This overlap determines an effectiveness limit which cannot be eliminated globally relying on the pixel-wise classification.

A great potential is still hidden in the adaptive skin color modeling, which was well demonstrated by the ground-truth adaptation *ROC* curve in Fig. 13. Another group of powerful techniques for improving skin detection is underpinned by the spatial analysis and segmentation-based verification of the detection outcome. It may be beneficial to combine these two approaches by iteratively adapting the model, taking advantage of the spatial analysis. A similar direction was explored by A.A. Argyros and M.I.A. Lourakis in 2004 [6], but many advances have been made since then, both concerning the model adaptation as well as the spatial analysis. In our opinion, this is the most promising, while still little explored possibility for increasing the efficacy of skin detection that is worth being investigated in the future.

**Acknowledgments** This work has been supported by the Polish Ministry of Science and Higher Education under research grant no. IP2011 023071 from the Science Budget 2012–2013 and the European Union from the European Social Fund (grant agreement number: UDA-POKL.04.01.01-00-106/09).

## References

1. Abdullah-Al-Wadud M, Chae O (2007) Region-of-interest selection for skin detection based applications. In: International conference on convergence information technology, pp 1999–2004
2. Abdullah-Al-Wadud M, Chae O (2008) Skin segmentation using color distance map and water-flow property. In: Proceedings of the information assurance and security (ISIAS '08), pp 83–88
3. Abin AA, Fotouhi M, Kasaei S (2009) A new dynamic cellular learning automata-based skin detector. *Multimedia Syst* 15(5):309–323
4. Al-Mohair HK, Mohamad-Saleh J, Suandi SA (2012) Human skin color detection: a review on neural network perspective. *Int J Innovative Comput Inf Control (IJICIC)* 8(12):8115–8131
5. Albiol A, Torres L, Delp E (2001) Optimum color spaces for skin detection. In: Proceedings of the IEEE international conference on image processing, pp 122–124
6. Argyros AA, Lourakis MIA (2004) Real-time tracking of multiple skin-colored objects with a possibly moving camera. In: Proceedings of the ECCV, LNCS, vol 3023. Springer, pp 368–379

7. Baltzakis H, Pateraki M, Trahanias P (2012) Visual tracking of hands, faces and facial features of multiple persons. *Mach Vis Appl* 23:1141–1157
8. Berbar MA (2011) Novel colors correction approaches for natural scenes and skin detection techniques. *Int J Video Image Process Netw Secur* 11(2):1–10
9. Bhoyar KK, Kakde OG (2010) Skin color detection model using neural networks and its performance evaluation. *J Comput Sci* 6(9):963–968
10. Bilal S, Akmeliawati R, Salami MJE, Shafie AA (2012) Dynamic approach for real-time skin detection. *J Real-Time Image Process*
11. Brand J, Mason J (2000) A comparative assessment of three approaches to pixel-level human skin-detection. In: *Proceedings of the 15th international conference on pattern recognition vol 1*, pp 1056–1059
12. Caetano TS, Olabbarriaga SD, Barone DAC (2003) Do mixture models in chromaticity space improve skin detection? *Pattern Recogn* 36:3019–3021
13. Cheddad A, Condell J, Curran K, Mc Kevitt P (2009) A skin tone detection algorithm for an adaptive approach to steganography. *Signal Process* 89(12):2465–2478
14. Chen MJ, Chi MC, Hsu CT, Chen JW (2003) ROI video coding based on H.263+ with robust skin-color detection technique. In: *IEEE international conference on consumer electronics*, pp 44–45
15. Chen YH, Hu KT, Ruan SJ (2012) Statistical skin color detection method without color transformation for real-time surveillance systems. *Eng Appl Artif Intell* 25(7):1331–1337
16. Chenaoua K, Bouridane A (2006) Skin detection using a Markov random field and a new color space. In: *Proceedings of the IEEE international conference on image processing*, pp 2673–2676
17. Choi B, Chung B, Ryou J (2009) Adult image detection using Bayesian decision rule weighted by SVM probability. In: *Proceedings of the 4th international conference on computer sciences and convergence information technology (ICCIT '09)*, pp 659–662
18. Clausi D, Jernigan M (1998) A fast method to determine co-occurrence texture features. *IEEE Trans Geosci Remote Sensing* 36(1):298–300
19. Conci A, Nunes E, Pantrigo JJ, Sánchez Á (2008) Comparing color and texture-based algorithms for human skin detection. In: *Proceedings of the ICEIS*, pp 166–173
20. Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20(3):273–297
21. Dadgostar F, Sarrafzadeh A (2006) An adaptive real-time skin detector based on hue thresholding: a comparison on two motion tracking methods. *Pattern Recogn Lett* 27(12):1342–1352
22. Dawod A, Abdullah J, Alam M (2010) Adaptive skin color model for hand segmentation. In: *Proceedings of the international conference on computer applications and industrial electronics (ICCAIE)*, pp 486–489
23. Duan L, Lin Z, Miao J, Qiao Y (2009) A method of human skin region detection based on PCNN. In: *Proceedings of the international symposium on neural networks: advances in neural networks, ISNN, Part III, LNCS, vol 5553*. Springer, Berlin, pp 486–493
24. Finlayson G, Hordley S, Hübner P (2001) Color by correlation: a simple, unifying framework for color constancy. *IEEE Trans Pattern Anal Mach Intell* 23(11):1209–1221
25. Finlayson GD, Schiele B, Crowley JL (1998) Comprehensive colour image normalization. In: *Proceedings of the european conference on computer vision (ECCV)*, vol 1, Freiburg, Germany, pp 475–490
26. Forsyth DA, Fleck MM (1999) Automatic detection of human nudes. *Int J Comput Vis* 32:63–77
27. Fotouhi M, Rohban M, Kasaei S (2009) Skin detection using contourlet-based texture analysis. In: *Proceedings of the 4th international conference on digital telecomm (ICDT'09)*, pp 59–64
28. Fritsch J, Lang S, Kleinehagenbrock M, Fink G, Sagerer G (2002) Improving adaptive skin color segmentation by incorporating results from face detection. In: *Proceedings of the IEEE international workshop on robot and human interactive, communication*, pp 337–343
29. Gasparini F, Corchs S, Schettini R (2005) Pixel based skin colour classification exploiting explicit skin cluster definition methods. In: *Proceedings of the 10th congress of the international colour association*, vol 1, pp 543–546

30. Gatta C, Rizzi A, Marini D (2000) Ace: an automatic color equalization algorithm. In: Proceedings of the first european conference on color in graphics image and vision (CGIV02)
31. Gomez G, Morales EF (2002) Automatic feature construction and a simple rule induction algorithm for skin detection. In: Proceedings of the ICML workshop on machine learning in computer vision, pp 31–38
32. Greenspan H, Goldberger J, Eshet I (2001) Mixture model for face-color modeling and segmentation. *Pattern Recogn Lett* 22:1525–1536
33. Han J, Awad G, Sutherland A, Wu H (2006) Automatic skin segmentation for gesture recognition combining region and support vector machine active learning. In: Proceedings of the IEEE international conference on automatic face and gesture recognition. IEEE Computer Society, Washington DC, USA, pp 237–242
34. Hanbury A (2003) A 3D-polar coordinate colour representation well adapted to image analysis. In: Proceedings of the Scandinavian conf on image analysis (SCIA). Springer, Berlin, pp 804–811
35. Hossain MF, Shamsi M, Alsharif MR, Zoroofi RA, Yamashit K (2012) Automatic facial skin detection using Gaussian mixture model under varying illumination. *Int J Innovative Comput Inf Control* 8(2):1135–1144
36. Hsu RL, Abdel-Mottaleb M, Jain A (2002) Face detection in color images. *IEEE Trans Pattern Anal Mach Intell* 24(5):696–706
37. Jagadesh BN, Rao K, Satyanarayana C, RajKumar GVS (2012) Skin colour segmentation using finite bivariate pearsonian type-IIb mixture model and k-means. *Signal Image Process Int J* 3(4):37–49
38. Jiang Z, Yao M, Jiang W (2007) Skin detection using color, texture and space information. *Proc Int Conf Fuzzy Syst Knowl Discov* 3:366–370
39. Jones M, Rehg J (1999) Statistical color models with application to skin detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), vol 1, pp 637–663
40. Jones M, Rehg J (2002) Statistical color models with application to skin detection. *Int J Comput Vis* 46:81–96
41. Kakumanu P, Makrogiannis S, Bourbakis NG (2007) A survey of skin-color modeling and detection methods. *Pattern Recogn* 40(3):1106–1122
42. Kakumanu P, Makrogiannis S, Bryll R, Panchanathan S, Bourbakis N (2004) Image chromatic adaptation using ANNs for skin color adaptation. In: Proceedings of the IEEE international conference on tools with artificial intelligence (ICTAI), pp 478–485
43. Kawulok M (2008) Dynamic skin detection in color images for sign language recognition. In: Proceedings of the ICISP, LNCS, vol 5099. Springer, pp 112–119
44. Kawulok M (2010) Energy-based blob analysis for improving precision of skin segmentation. *Multimedia Tools Appl* 49(3):463–481
45. Kawulok M (2012) Texture analysis for skin probability maps refinement. In: Proceedings of the MCPR, LNCS, vol 7329. Springer, pp 75–84
46. Kawulok M (2013) Fast propagation-based skin regions segmentation in color images. In: Proceedings of the IEEE international conference on automatic face and gesture recognition, FG, pp 1–7
47. Kawulok M, Kawulok J, Smolka B (2012) Discriminative textural features for image and video colorization. *IEICE Trans Inf Syst* 95–D(7):1722–1730
48. Kawulok M, Nalepa J (2012) Support vector machines training data selection using a genetic algorithm. In: Statistical techniques in pattern recognition, S+SSPR 2012, LNCS, vol 7626. Springer, pp 557–565
49. Kawulok M, Szymanek J (2012) Precise multi-level face detector for advanced analysis of facial images. *IET Image Process* 6(2):95–103
50. Khan R, Hanbury A, Sablatnig R, Stottinger J, Khan F, Khan F (2012) Systematic skin segmentation: merging spatial and non-spatial data. In: Multimedia tools and applications, pp 1–25

51. Khan R, Hanbury A, Stöttinger J (2010) Skin detection: a random forest approach. In: Proceedings of the 17th IEEE international image processing (ICIP) conference, pp 4613–4616
52. Khan R, Hanbury A, Stöttinger J, Bais A (2012) Color based skin classification. *Pattern Recogn Lett* 33(2):157–163
53. Kovac J, Peer P, Solina F (2002) Eliminating the influence of non-standard illumination from images. Technical report
54. Kovac J, Peer P, Solina F (2003) Human skin color clustering for face detection. In: EUROCON 2003 computer as a tool, vol 2, pp 144–148
55. Krahenbuhl P, Koltun V (2011) Efficient inference in fully connected CRFs with Gaussian edge potentials. In: Proceedings of the neural information processing systems (NIPS)
56. Kruppa H, Bauer MA, Schiele B (2002) Skin patch detection in real-world images. In: Proceedings of the DAGM symposium on pattern recognition, LNCS, vol 2449. Springer, pp 109–117
57. Kukharev G, Nowosielski A (2004) Fast and efficient algorithm for face detection in colour images. *Mach Graph Vis* 13:377–399
58. Lam HK, Au O, Wong CW (2004) Automatic white balancing using standard deviation of RGB components. In: Proceedings of the international symposium on circuits and systems (ISCAS) vol 3, pp 921–924
59. Lee JS, Kuo YM, Chung PC, Chen EL (2007) Naked image detection based on adaptive and extensible skin color model. *Pattern Recogn* 40:2261–2270
60. Lichtenauer J, Reinders MJT, Hendriks EA (2007) A self-calibrating chrominance model applied to skin color detection. In: Proceedings of the VISAPP, vol 1, pp 115–120
61. Musicant DR, Feinberg A (2004) Active set support vector regression. *IEEE Trans Neural Netw* 15(2):268–275
62. Ng P, Pun CM (2011) Skin color segmentation by texture feature extraction and k-mean clustering. In: Proceedings of the 2011 3rd international conference on computational intelligence, communication systems and networks (CICSyN), pp 213–218
63. Nikolaidis A, Pitas I (2000) Robust watermarking of facial images based on salient geometric pattern matching. *IEEE Trans Multimedia* 2(3):172–184
64. Phung S, Bouzerdoum A, Chai D (2005) Skin segmentation using color pixel classification: analysis and comparison. *IEEE Trans Pattern Anal Mach Intell* 27(1):148–154
65. Phung SL, Bouzerdoum A, Chai D (2002) A novel skin color model in YCbCr color space and its application to human face detection. In: Proceedings of the international conference on image processing, vol 1, pp I-289–I-292
66. Phung SL, Chai D, Bouzerdoum A (2003) Adaptive skin segmentation in color images. In: Proceedings of the IEEE international conference on acoustics, speech and signal proceedings, pp 353–356
67. Rao K, Jagadeesh BN, Satyanarayana C (2012) Skin colour segmentation using finite bivariate pearsonian type-IVa mixture model. *Comput Eng Intell Syst* 3(5):45–56
68. Ratnasingham S, McGinnity T (2012) Chromaticity space for illuminant invariant recognition. *IEEE Trans Image Process* 21(8):3612–3623
69. Schmutz SJ, Jayaram S, Shin MC, Tsap LV (2007) Objective evaluation of approaches of skin detection using roc analysis. *Comput Vis Image Underst* 108(1–2):41–51
70. Schohn G, Cohn D (2000) Less is more: active learning with support vector machines. In: Proceedings of the 17th international conference on machine learning, pp 839–846. Morgan Kaufmann Publishers Inc, USA
71. Seow MJ, Valaparla D, Asari V (2003) Neural network based skin color model for face detection. In: Proceedings of the applied imagery pattern recognition workshop, pp 141–145
72. Shin M, Chang K, Tsap L (2002) Does colorspace transformation make any difference on skin detection? In: Proceedings of the IEEE workshop on applications of computer vision (WACV), pp 275–279
73. Sigal L, Sclaroff S, Athitsos V (2003) Skin color-based video segmentation under time-varying illumination. *IEEE Trans Pattern Anal Machine Intell* 26:862–877



74. Sobottka K, Pitas I (1996) Face localization and facial feature extraction based on shape and color information. In: Proceedings of the IEEE international conference on image processing (ICIP), vol 3, pp 483–486
75. del Solar JR, Verschae R (2004) Skin detection using neighborhood information. In: Proceedings of the IEEE international conference on automatic face and gesture recognition, pp 463–468
76. Solina F, Peer P, Batagelj B, Juvan S (2002) 15 seconds of fame: an interactive, computer-vision based art installation. In: Proceedings of the international conference on control, automation, robotics and vision (ICARCV), vol 1, pp 198–204
77. Soriano M, Martinkauppi B, Huovinen S, Laaksonen M (2000) Skin detection in video under changing illumination conditions. In: Proceedings of the international conference on pattern recognition (ICPR), vol 1, pp 839–842
78. Stern H, Efros B (2002) Adaptive color space switching for face tracking in multi-colored lighting environments. In: Proceedings of the IEEE international conference on automatic face and gesture recognition (FG 2002). IEEE Computer Society, Washington DC, USA, pp 249–254
79. Stoerring M, Andersen HJ, Granum E, Granum E (1999) Skin colour detection under changing lighting conditions. In: Proceedings of the 7th symposium on intelligent robotics systems, pp 187–195
80. Sun HM (2010) Skin detection for single images using dynamic skin color modeling. *Pattern Recogn* 43(4):1413–1420
81. Tan WR, Chan CS, Yogarajah P, Condell J (2012) A fusion approach for efficient human skin detection. *IEEE Trans Ind Inf* 8(1):138–147
82. Taqa A, Jalab H (2010) Increasing the reliability of skin detectors. *Sci Res Essays* 5(17):2480–2490
83. Terrillon J-C, David M, Akamatsu S (1998) Automatic detection of human faces in natural scene images by use of a skin color model and of invariant moments. In: Proceedings of the 3rd international conference on automatic face and gesture recognition, pp 112–117, Nara, Japan
84. Tomaz F, Candeias T, Shahbazkia H (2003) Improved automatic skin detection in color images. In: Proceedings of the 7th digital computing: techniques and applications, pp 419–427
85. Tsekeridou S, Pitas I (1998) Facial feature extraction in frontal views using biometric analogies. In: Proceedings of the EUSIPCO '98, pp 315–318
86. Tu Y, Yi F, Chen G, Jiang S, Huang Z (2010) Skin color detection by illumination estimation and normalization in shadow regions. In: Proceedings of the IEEE international conference on information and automation (ICIA), pp 1082–1085
87. Vezhnevets V, Sazonov V, Andreeva A (2003) A survey on pixel-based skin color detection techniques. In: IN Proceedings of the GRAPHICON-2003, pp 85–92
88. Viola P, Jones M (2004) Robust real-time face detection. *Int J Comput Vis* 57(2):137–154
89. Wang X, Zhang X, Yao J (2011) Skin color detection under complex background. In: Proceedings of the international conference on mechatronic science, electric engineering and computer, pp 1985–1988
90. Yang G, Li H, Zhang L, Cao Y (2010) Research on a skin color detection algorithm based on self-adaptive skin color model. In: Proceedings of the international conference on communications and intelligence information security (ICCIIS), pp 266–270
91. Yang J, Fu Z, Tan T, Hu W (2004) Skin color detection using multiple cues. In: Proceedings of the international conference on image processing (ICPR), vol 1, pp 632–635
92. Yang MH, Ahuja N (1999) Gaussian mixture model for human skin color and its applications in image and video databases. In: *ProcSPIE* 99, CA, San Jose, pp 458–466
93. Yang U, Kang M, Toh KA, Sohn K (2010) An illumination invariant skin-color model for face detection. In: Proceedings of the IEEE international conference on biometrics: theory applications and systems (BTAS), pp 1–6
94. Yogarajah P, Condell J, Curran K, Cheddad A, McKeivitt P (2010) A dynamic threshold approach for skin segmentation in color images. In: Proceedings of the IEEE international conference on image processing (ICIP), pp 2225–2228

95. Yogarajah P, Condell J, Curran K, McKeivitt P, Cheddad A (2012) A dynamic threshold approach for skin segmentation in color images. *Int J Biometrics* 4(1):38–55
96. Yong-jia Z, Shu-ling D, Xiao X (2008) A Mumford-Shah level-set approach for skin segmentation using a new color space. In: *Proceedings of the international conference on system simulation and scientific computing (ICSC)*, pp 307–310
97. Zafarifar B, Martiniere A, de With P (2010) Improved skin segmentation for TV image enhancement, using color and texture features. In: *Proceedings of the international conference on consumer electronics (ICCE)*, pp 373–374
98. Zarit BD, Super BJ, Quek FKH (1999) Comparison of five color models in skin pixel classification. In: *Proceedings of the international workshop on recognition, analysis, and tracking of faces and gestures in, real-time systems*, pp 58–63
99. Zhang MJ, Gao W (2005) An adaptive skin color detection algorithm with confusing backgrounds elimination. In: *Proceedings of the international conference on image processing (ICIP)*, vol 2, pp 390–393
100. Zhu Q, Cheng KT, Wu CT, Wu YL (2004) Adaptive learning of an accurate skin-color model. In: *Proceedings of the IEEE international conference on automatic face and gesture recognition (FG 2004)*. IEEE Computer Society, Washington DC, USA, pp 37–42
101. Zhu Q, Wu CT, Cheng KT, Wu YL (2004) An adaptive skin model and its application to objectionable image filtering. In: *Proceedings of the ACM international conference on multimedia (MULTIMEDIA '04)*. ACM, New York, USA, pp 56–63

# Contribution of Skin Color Cue in Face Detection Applications

Dohyoung Lee, Jeaff Wang and Konstantinos N. Plataniotis

**Abstract** Face detection has been considered as one of the most active areas of research due to its wide range of applications in computer vision and digital image processing technology. In order to build a robust face detection system, several cues, such as motion, shape, color, and texture have been considered. Among available cues, color is one of the most effective ones due to its computational efficiency, high discriminative power, as well as robustness against geometrical transform. This chapter investigates the role of skin color cue in automatic face detection systems. General overview of existing face detection techniques and skin pixel classification solutions are provided. Further, illumination adaptation strategies for skin color detection are discussed to overcome the sensitivity of skin color analysis against illumination variation. Finally, two case studies are presented to provide more realistic view of contribution of skin color cue in face detection frameworks.

## 1 Introduction

Face detection refers to a task of processing image/video data in order to determine the position/scale of any faces in them [31]. With recent advancement in digital imaging devices and multimedia technologies, face detection has received a significant deal of attention from the research communities due to its wide range of applications from video surveillance system, human-computer interface (HCI), to face image database

---

D. Lee (✉) · J. Wang · K. N. Plataniotis  
Multimedia Lab, The Edward S. Rogers Department of Electrical and Computer Engineering,  
University of Toronto, 10 King's College Road, Toronto, Canada  
e-mail: dohyoung.lee@utoronto.ca

J. Wang  
e-mail: jeaff.wang@utoronto.ca

K. N. Plataniotis  
e-mail: kostas@comm.utoronto.ca

management. For instance, a successful solution to the face detection problem is potentially useful for user scenarios such as:

1. Surveillance cameras have been widely deployed in many strategic places to ensure public safety by monitoring criminal and terrorist activities. As a result, efforts have been made on the implementation of an intelligent system that performs monitoring task automatically without human intervention. Such task requires a preliminary operation that locates faces within the video scene, followed by other surveillance operations, such as face tracking and gait recognition.
2. As mobile phones become increasingly powerful in terms of processing resource and data capacity, security of the data stored in them such as contact information and confidential documents becomes very important. The conventional way of protecting such sensitive data is to have them password protected. However, latest mobile phones started to offer face recognition technology as a more secure and convenient security means since it provides distinctive print to get access while the user does not need to memorize passwords. To this end, an user can take a still image of himself/herself for identification through the image sensor. The detection of face region in captured image data is an essential initialization stage to complete user verification.

The importance of face detection cannot be overemphasized in the aforementioned applications since they operate under an assumption that faces are already accurately located in the given image or in the image sequence prior to main processing operation. In the literature, numerous strategies have been proposed; however, face detection is still considered to be challenging since faces are non-rigid objects and vary substantially in shape, color, and texture [75]. Key technical challenges involved in face detection problem include:

- **Pose Variation:** face object varies significantly depending on its relative position between camera-face (from frontal views to different angles of side and tilt views)
- **Imaging Condition:** the variation of illumination condition (e.g. color temperature of light source) and camera characteristic (e.g. sensor spectral sensitivity) affects the appearance of color or texture of face region.
- **Occlusions:** complete face region might not be fully visible to camera, and some facial feature might be hidden by other object

Progress has been made to solve these problems by using different aspects of facial characteristics including geometric relationships between facial features (e.g. eyes, nose, and mouth), skin color, facial texture patterns, and so forth. Although many different cues have been utilized in the face detection system, skin color cue is one of the most effective ones due to its robustness towards geometric changes (e.g. scaling and rotation) and computational efficiency.

In this chapter, we investigate the role of skin color cue in automatic face detection systems. First we provide general overview of face detection techniques with emphasis on use of color information (Sect. 2). In Sect. 3, we provide review of various color representation and skin pixel classification methodologies based on color information. In addition, illumination adaptation strategies for skin color detection is

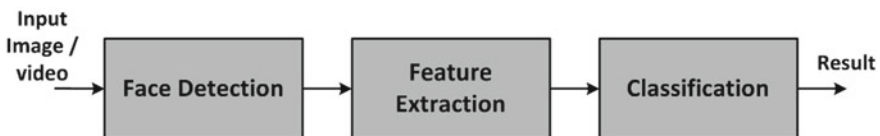
discussed. In Sects. 4 and 5, we introduce two case studies of skin color cue usage in face detection framework. By presenting two distinct use cases of color information in facial analysis, the effectiveness of color cue in terms of detection performance and computational efficiency is emphasized. Additionally, the influence of stable color representation on face detection performance is addressed by applying color constancy algorithms prior to skin color detection process. Finally, conclusions are drawn in Sect. 6.

## 2 Face Detection and Color Cue

Face detection problem has received tremendous research interests since the 1970s due to its importance in computer vision applications such as human identification and tracking, human-computer interaction, and content-based image retrieval. For instance, the recognition of a face in visual data typically involves three main stages (Fig. 1), where the detection of face region is an essential initialization step for face alignment (i.e. registration) that any subsequent operations are directly influenced by its accuracy [81].

The purpose of face detection is to process still images or image sequences to find the location(s) and the size(s) of any faces in them [31]. Although face detection is a trivial task for humans, it is very challenging to build a stable automatic solution since face patterns significantly vary under different facial poses/expressions, occlusion conditions, and imaging condition (e.g. illumination condition, sensor characteristics). Various face detection algorithms have been proposed [31, 75, 79], but achieving highly accurate detection performance while maintaining reasonable computational costs still remains to be a challenging issue.

In general, existing face detection methods are grouped into the two main categories [31]: (i) *feature-based approach*, (ii) *image-based approach*. In feature-based approach [33, 48, 54, 58, 72], explicit face features (e.g. eyes, mouth, nose and face contour) are extracted, then relationships between them (such as geometric and morphologic relationships) are used to determine the existence of the face. For instance, Sobottka and Pitas [72] proposed a two-stage face detection framework to locate face regions in color image. The first stage is dedicated to segment face-like regions by skin color analysis using hue and saturation information in HSV colorspace, followed by shape analysis using ellipse fitting. Afterwards, grayscale information of detected face-like regions are examined to verify them by locating eye and mouth features.



**Fig. 1** Block diagram of a generic face recognition system

Extending the feature-based approach, Hsu et al. [33] localized face candidates from color image using skin color cue in YCbCr colorspace and constructed eye, mouth, face boundary maps to verify each face candidate. The methods in this category are advantageous due to their relatively simple implementation and high detection accuracy in uncluttered backgrounds. In particular, skin color cue is exceptionally popular and successful in feature-based approach due to its simplicity and high discrimination power. Some of these methods remain very popular nowadays in certain applications such as mobile phone applications. However, they tend to have difficulties in dealing with challenging imaging conditions such as varying illumination and complex background, as well as low resolution images containing multiple faces.

Alternatively, image-based approach [9, 13, 67, 80] uses machine learning techniques to capture unique and implicit face features, treating the face detection problem as a binary classification problem to discriminate between face and non-face. Often, methods in this category require tremendous amount of time and training data to construct a stable face detection system. However, in recent years, the rapid advancement in digital data storage and digital computing resources has made the image-based approaches feasible to many real-life applications and they become extremely popular due to their enhanced robustness and superior performance against challenging conditions compared to feature-based approaches.

One of the most representative works in image-based approach is the Viola-Jones's face detection framework [67], a Haar-like feature based frontal face detection system for grayscale images. The Haar-like feature represents the differences in grayscale between two or more adjacent rectangular regions in the image, characterizing local texture information. In Viola-Jones framework, AdaBoost learning algorithm is used to handle following three fundamental problems: (i) learning effective features from a large Haar-like feature set, (ii) constructing weak classifiers, each of which is based on one of the selected features, (iii) boosting the weak classifiers to construct a strong classifier. The authors applied the integral image technique for fast computation of Haar-like features under varying scale and location, achieving real-time operation<sup>1</sup>. However, the simplicity of Haar-like features in Viola-Jones face detector causes limited performance under many complications, such as illumination variation. More recent approaches address this problem by using an alternative texture feature called Local Binary Pattern (LBP), which is introduced by Ojala et al. [50] to offer enhanced discriminative power and tolerance towards illumination variation. The LBP descriptor encodes the relative gray value differences from a  $3 \times 3$  neighborhood of an image patch as demonstrated in Fig. 2. By taking the center pixel as a threshold value, every neighboring pixel is compared against the threshold to produce a 8-bit string, and then binary string is converted to decimal label according to assigned weights. Many variants of LBP features are considered thereafter, such as LBP Histogram [26], Improved Local Binary Pattern (ILBP) [35], Multi-Block LBP (MB-LBP) [80], and Co-occurrence of LBP (CoLBP) [47].

---

<sup>1</sup> The term real-time implies the capability to process image frames with a rate close to the examined sequence frame rate. In [67], real-time requirement is defined to be approximately 15 frames per second for 384x288 image.

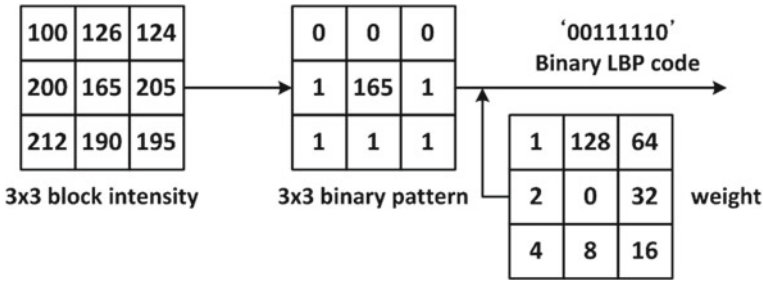


Fig. 2 Example of basic LBP feature computation

Although considerable progress has been made in aforementioned image-based face detection methodologies, the main emphasis has been placed on exploiting various grayscale texture patterns. A few recent researches [12, 25] have shown that color cue could provide complementary information to further enhance the performance of image-based approaches. Specifically, color information could potentially enhance the performance in following two aspects: (i) using skin color information, one may effectively reduce the search regions for potential face candidates by identifying skin color regions and performing subsequent texture analysis on detected skin region only, hence avoiding heavy computations caused by exhaustive scan of the entire image, (ii) color is a pixel-based cue which can be processed regardless of spatial arrangement, hence, it offers computational efficiency as well as robustness against geometric transformation such as scaling, rotation, and partial occlusion.

Overall, color can be applied in face detection systems, either as a primary or a complementary feature to locate faces along with shape, texture, and motion. In feature-based approach, which is suitable for resource constrained systems, color provides visual cue to focus attention in the scene by identifying a set of skin-colored regions that may contain face objects. It is followed by subsequent feature analysis where each skin-colored region is analyzed using facial geometry information. In image-based approach, faster and more accurate exhaustive face search can be achieved by using skin color modality with the purpose of guiding the search. Typical examples of each face detection approach and use of color cue are described in Sects. 4 and 5. Since accurate classification between skin and non-skin pixel is a key element for reliable implementation of face detection systems, in Sect. 3, we provide extensive review of existing skin color classification methodologies.

### 3 Skin Color Detection

Color is an effective cue for identifying regions of interest/objects in the image data. For visual contents of natural scene, a widely accepted assumption is that the color corresponds to a few categories have the most perceptual impact on the

human visual system [76]. Researches indicate that skin tones, blue sky, and green foliage constitute such basic classes and belong to a group of color termed *memory colors*. Among memory colors, skin color has been regarded as the most important ones due to its importance in human social interaction and its dominant usage in image/video analysis. The application of skin color analysis is not only limited to face detection system, which is the main focus of this chapter, but also includes content-based image retrieval system, human-computer interaction domain, and memory color enhancement system.

Skin color analysis in face detection framework involves a pixel-wise classification to discriminate skin and non-skin pixels in color images. Therefore, skin color detection process can be seen as a binary classification problem that a certain color pixel  $\mathbf{c} = [c_1, c_2, c_3]^T$  is mapped to an output label  $y \in \{w_s, w_n\}$ , where  $w_s$  and  $w_n$  represent skin and non-skin classes respectively.

Detection of skin color is considerably challenging not only because it is sensitive to varying illumination conditions and camera characteristics, but also it should be able to handle individual differences caused by ethnicity, age, and gender. Skin color detection involves two important sub-problems: (i) selection of a suitable color representation to perform classification (discussed in Sect. 3.1), (ii) selection of modeling scheme to represent skin color distribution (discussed in Sect. 3.2).

### 3.1 Color Representation

An appropriate representation of color signal is crucial in detection of skin pixels. An ideal colorspace for skin color analysis is assumed to: (i) minimize the overlap between skin and non-skin color distributions in the given colorspace, (ii) provide robustness against varying illumination condition, (iii) provide separability between luminance and chrominance information. Several colorspace, such as RGB [36, 61], normalized RGB [4, 6, 23, 59], YCbCr [7, 29, 33, 62], HSV [30, 49, 55], CIELAB [78] and CIELUV [74], have been used in skin color detection. In this section, we will focus on most widely used colorspace in the image processing research, including RGB, normalized RGB, YCbCr, and HSV.

#### 3.1.1 RGB and Normalized RGB

RGB is a fundamental way of representing color signal which is originated from cathode ray tube (CRT) display. The RGB model (Fig. 3) is represented by a 3-dimensional cube with R, G, and B at the corners on each axis. RGB is most dominantly used representation for processing and storing of digital image data. Therefore, it has been used in many skin color detection researches [36, 51, 61]. However, its poor separability between luminance and chromaticity information, and its highly sensitive nature against illumination variation are main limitations for skin color analysis purpose. Rather than original RGB format, its normalized variant is consid-



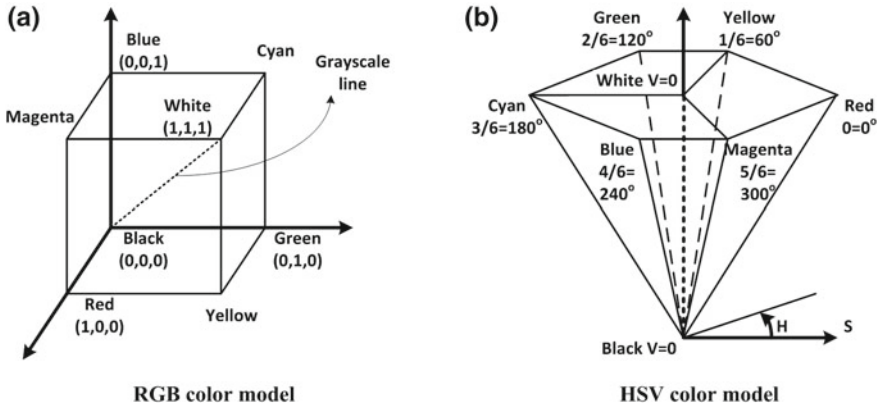


Fig. 3 RGB cube and HSV hexcone representations

red to be more robust in skin color detection since it reduces the dependency of each component to illumination changes<sup>2</sup> [38]. Normalized RGB representation can be obtained by normalizing RGB values by their intensity ( $I = R + G + B$ ):

$$r = \frac{R}{R + G + B}, \quad g = \frac{G}{R + G + B}, \quad b = \frac{B}{R + G + B} \tag{1}$$

where each RGB primaries are given in linear RGB<sup>3</sup>. Because  $r + g + b = 1$ , no information is lost if only two elements are considered.

### 3.1.2 YCbCr

YCbCr is the Rec. 601 international standard (see 2) for studio quality component digital video. In YCbCr, color is represented by luma(Y), computed as a weighted sum of nonlinear (gamma-corrected) RGB, and two chroma components Cr and Cb that are formed by subtracting luma value from R and B components. The Rec. 601 specifies 8 bit (i.e. 0–255) coding of YCbCr. All three Y, Cb, and Cr components have reserved ranges to provide footroom and headroom for signal processing as follows:

<sup>2</sup> Under the white illumination condition with Lambertian reflection assumption, normalized RGB is invariant to illumination direction and illumination intensity [21]

<sup>3</sup> Linear RGB implies that it is linear to the physical intensity, whereas nonlinear RGB is non-linear to intensity. Such nonlinearity is introduced to RGB signal by gamma correction process in order to compensate a nonlinear response of CRT display devices.

$$\begin{cases} Y' &= 16 + (65.738R' + 129.057G' + 25.064B') \\ Cb &= 128 + (-37.945R' - 74.494G' + 112.439B') \\ Cr &= 128 + (112.439R' - 94.154G' - 18.285B') \end{cases} \quad (2)$$

where  $R', G', B' \in [0, 1]$  are gamma-corrected RGB primaries. YCbCr is dominantly used in compression applications since it reduces the redundancy in RGB color signals and represents the color with statistically independent components. The explicit separation of luminance and chrominance components and its wide adoption in image/video compression standards makes YCbCr popular for skin color analysis [7, 29, 33, 62].

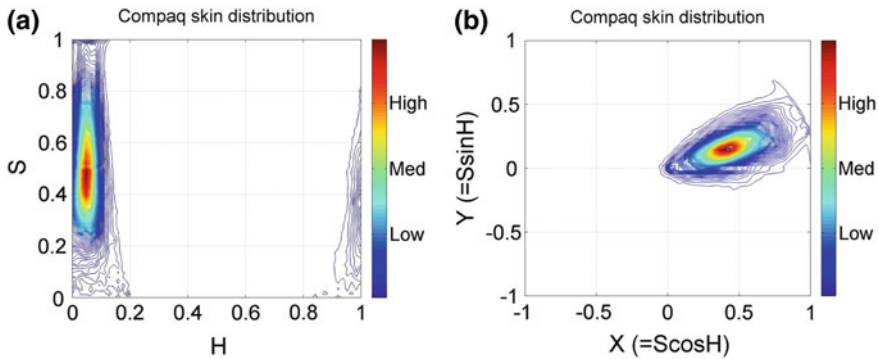
### 3.1.3 HSV

The HSV coordinate system, originally proposed by Smith [57], defines color by: (i) Hue(H)—the property of a color related to the dominant wavelength in a mixed light wave, (ii) Saturation(S)—the amount of white light mixed with color that varies from gray through pastel to saturated colors, (iii) Value(V)—the property according to which an area appears to exhibit more or less light that varies from black to white. The HSV representation corresponds more closely to the human perception of color than aforementioned ones since it is derived from the intuitive appeal of the artist’s tint, shade, and tone. The set of equations used to transform a point in the RGB to the HSV coordinate system ( $[H, S, V] \in [0, 1], [R', G', B'] \in [0, 1]$ ) is given as follows<sup>4</sup>:

$$\begin{cases} H &= \frac{1}{6} \times \begin{cases} 5 + (M - B')/C, \text{ if } M = R', m = G' \\ 1 - (M - G')/C, \text{ if } M = R', m \neq G' \\ 1 + (M - R')/C, \text{ if } M = G', m = B' \\ 3 - (M - B')/C, \text{ if } M = G', m \neq B' \\ 3 + (M - G')/C, \text{ if } M = B', m = R' \\ 5 - (M - R')/C, \text{ if } M = B', m \neq R' \end{cases} \\ S &= C/M \quad (\text{if } C=0, \text{ then } H = \text{undefined}) \\ V &= M \end{cases}, \begin{cases} M = \max(R', G', B') \\ m = \min(R', G', B') \\ C = M - m \end{cases} \quad (3)$$

The HSV colorspace is traditionally shown as a hexcone model Fig. 3b and, in fact, HSV hexcone is a projection of the RGB cube along the gray-scale line. In hexcone model, H is represented as the angle and S corresponds to the horizontal distance from the vertical axis. V varies along the vertical axis with  $V = 0$  being

<sup>4</sup> It is noteworthy to mention that the original literature [57] does not clearly indicate whether linear or nonlinear RGB is used in conversion. Although there is such an ambiguity, we use nonlinear RGB in this paper, which is implicit in image processing applications [52].



**Fig. 4** Distribution of skin color pixel from Compaq dataset [36] in **a** H-S plane of original HSV colorspace, **b** Cartesian representation of H-S plane

black, and  $V = 1$  being white. When  $S = 0$ , color is a gray value. When  $S = 1$ , color is on the boundary of hexcone. The greater the  $S$ , the farther the color is from white/gray/black. Adjusting the hue varies the color from red at  $H = 0$ , through green at  $H = 1/3$ , blue at  $H = 2/3$ , and back to red at  $H = 0$ . When  $S = 0$ , or  $V = 0$ , (along the achromatic axis) the color is grayscale and  $H$  is undefined.

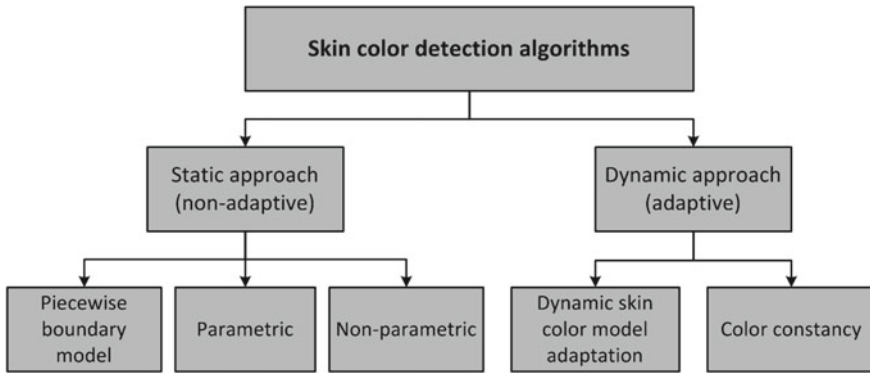
Two chromatic components,  $H$  and  $S$ , are known to be less variant to changes of illumination direction and illumination intensity [20], and thus HSV is a reliable colorspace for skin color detection. However, one should be careful in exploiting HSV space for skin color analysis since manipulation of  $H$  involves circular statistics as  $H$  is an angular measure [27]. As can be seen in Fig. 4a, the cyclic nature of  $H$  component disallows use of a color distribution model which requires a compact cluster, e.g. a single Gaussian model, since it generates two separate clusters on both sides of  $H$  axis. To address this issue, polar coordinate system of H-S space can be represented in Cartesian coordinates X-Y as follows [4]:

$$X = S \cos(2\pi H), \quad Y = S \sin(2\pi H) \tag{4}$$

where  $H, S \in [0, 1]$  and  $X, Y \in [-1, 1]$ . Here,  $X$  component can be regarded as a horizontal projection of color vector representing a pixel in H-S space, while  $Y$  component can be regarded as a vertical projection of color vector representing a pixel in H-S space. In the Cartesian representation of H-S space (Fig. 4b), skin color distribution forms a tightly distributed cluster.

### 3.2 Skin Color Distribution Model

Skin color distribution model defines a decision rule to discriminate between skin and non-skin pixels in given colorspace. To detect skin color pixel, a multitude



**Fig. 5** Classification of skin color detection approaches

of solutions merge from two distinct categories (Fig. 5). The first type of methodologies use a static color classification rule and can be further divided into three sub-classes depending on the classifier type: (i) piecewise boundary model [7, 23, 30, 42], (ii) non-parametric skin distribution model [4, 36, 77], (iii) parametric skin distribution model [45, 73, 74]. However, static skin color classification methods are typically very sensitive to imaging conditions, e.g. skin color of same individual varies depending on the color temperature of light source (e.g. incandescent, fluorescent, and sunlight) as well as the characteristics of image acquisition devices (e.g. sensor spectral sensitivity and embedded white balancing algorithm). Therefore, it requires appropriate adaptation schemes to maintain stable performance in real-world environments. The dynamic approaches address such problematic cases either by pre-processing given input image to alleviate the influence of imaging condition on color description or by dynamically updating color classification model according to imaging condition.

Since skin color typically forms a small cluster in the colorspace, one of the easiest methods to build a skin color classifier is to explicitly define fixed decision boundaries of skin regions. Single or multiple ranges of threshold values for each color component are defined and the image pixel values that fall within these pre-defined ranges are defined as skin pixels. Piecewise boundary model has been exploited in various colorspace and Table 1 presents some representative examples.

Non-parametric skin color modeling methods estimate the probability of a color value to be a skin by defining a model that has no dependency on a parameter. The most representative methods in this class is Jones and Rehg's method [36] which uses a 2D or 3D color histogram to represent the distribution of skin color in colorspace. Under this approach, the given colorspace is quantized into a number of histogram bins and each histogram bin stores the likelihood that a given color belongs to the skin. Jones and Rehg built two 3D RGB histogram models for skin and non-skin from Compaq database [36] which contains around 12K web images. Given skin and non-skin histograms, the probability that a given color belongs to skin and non-skin class is defined as:

**Table 1** Commonly used piecewise boundary models for skin color detection

Authors (Colorspace)	Skin color classification rule
<b>Kovac et al. [42] (RGB)</b>	Uniform daylight illumination $R > 95, G > 40, B > 20, \text{Max}(R, G, B) - \text{Min}(R, G, B) < 15,  R - G  > 15, R > G, R > B$ Flashlight or daylight lateral illumination $R > 220, G > 210, B > 170,  R - G  \leq 15, B < R, B < G^a$
<b>Gomez&amp;Morales [23] (nRGB)</b>	$\frac{r}{g} > 1.185, \frac{rb}{(r+g+b)^2} > 0.107, \frac{rg}{(r+g+b)^2} > 0.112^b$
<b>Chai&amp;Ngan [7] (CbCr)</b>	$77 \leq Cb \leq 127, 133 \leq Cr \leq 173^c$
<b>Herodotou et al. [30] (HSV)</b>	$0.94 \leq H \leq 1$ or $0 \leq H \leq 0.14, 0.2 \leq S, 0.35 \leq V^d$

$$p(\mathbf{c}|w_s) = \frac{s(\mathbf{c})}{\text{Total skin pixel count}}, \quad p(\mathbf{c}|w_n) = \frac{n(\mathbf{c})}{\text{Total non-skin pixel count}} \quad (5)$$

where  $s(\mathbf{c})$  is the pixel count in the color bin  $\mathbf{c}$  of the skin histogram,  $n(\mathbf{c})$  is the pixel count in the color bin  $\mathbf{c}$  of the non-skin histogram. For skin pixel detection, we need to estimate  $p(w_s|\mathbf{c})$ —a probability of observing skin pixel given a  $\mathbf{c}$  color vector. To compute this probability, the Bayesian rule is applied using the given conditional probabilities of skin and non-skin:

$$p(w_s|\mathbf{c}) = \frac{p(\mathbf{c}|w_s)p(w_s)}{p(\mathbf{c}|w_s)p(w_s) + p(\mathbf{c}|w_n)p(w_n)} \quad (6)$$

Instead of calculating the exact value of  $p(w_s|\mathbf{c})$ , the ratio between  $p(w_s|\mathbf{c})$  and  $p(w_n|\mathbf{c})$  can be compared (i.e. likelihood ratio test) for classification as follows:

$$\begin{aligned} \mathbf{c} \in w_s, & \text{ if } \frac{p(w_s|\mathbf{c})}{p(w_n|\mathbf{c})} > K \\ \Rightarrow & \text{ if } \frac{p(w_s|\mathbf{c})}{p(w_n|\mathbf{c})} = \frac{p(\mathbf{c}|w_s)p(w_s)}{p(\mathbf{c}|w_n)p(w_n)} > K \\ \Rightarrow & \text{ if } \frac{p(\mathbf{c}|w_s)}{p(\mathbf{c}|w_n)} > \theta, \text{ where } \theta = K \times \frac{p(w_n)}{p(w_s)} \end{aligned} \quad (7)$$

where  $\theta$  is an adjustable threshold that controls the trade-off between true positive (TP) and false positive (FP) rates (See 13 for definitions of TP and FP).

The third categories in static approaches are parametric skin color modeling methods where color classification rule is derived from parameterized distributions. The parametric models have the advantage over the non-parametric ones that they require smaller amount of training data and storage space. Key problems for parametric skin color modeling are to find the best model and to estimate its parameters. The most popular solutions include single Gaussian model (SGM) [73], Gaussian mixture model (GMM) [24, 32, 74], and elliptical model [45].

Under controlled environment, skin colors of different subject cluster in a small region in the colorspace and hence, the distribution can be represented by SGM [73]. A multivariate Gaussian distribution of a  $d$ -dimensional color vector  $\mathbf{c}$  is defined as:

$$G(\mathbf{c}; \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp \left[ -\frac{(\mathbf{c} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{c} - \boldsymbol{\mu})}{2} \right] \quad (8)$$

where  $\boldsymbol{\mu}$  is the mean vector and  $\Sigma$  is the covariance matrix of the normally distributed color vector  $\mathbf{c}$ . The model parameters are estimated from the training data using the following equations:

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{c}_i, \Sigma = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{c}_i - \boldsymbol{\mu})(\mathbf{c}_i - \boldsymbol{\mu})^T \quad (9)$$

Either the  $G(\mathbf{c}; \boldsymbol{\mu}, \Sigma)$  probability or the Mahalanobis distance from the  $\mathbf{c}$  color vector to the mean vector  $\boldsymbol{\mu}$ , given the covariance matrix  $\Sigma$ , can be used to measure the similarity of the pixel with the skin color.

Although SGM has been a successful model to represent skin color distribution, the assumption of SGM requires a single cluster which smoothly varies around the mean. However, such an assumption often causes intolerable error in skin/non-skin discrimination since different modes (due to skin color types and varying illumination conditions) can co-exist within the skin cluster. Therefore, Yang and Ahuja introduced the GMM model [74] to represent more complex shaped distribution. The GMM probability density function (pdf) can be defined as a weighted sum of Gaussians:

$$p(\mathbf{c}; \alpha_i, \boldsymbol{\mu}_i, \Sigma_i) = \sum_{i=1}^N \alpha_i G_i(\mathbf{c}; \boldsymbol{\mu}_i, \Sigma_i) \quad (10)$$

where  $N$  is the number of mixture components,  $\alpha_i$  is the weight of  $i$ -th component ( $\alpha_i > 0$ ,  $\sum_{i=1}^N \alpha_i = 1$ ),  $G_i$  is a Gaussian pdf with parameters  $\boldsymbol{\mu}_i$  and  $\Sigma_i$ . The parameters of a GMM are approximated from the training data via the iterative expectation-maximization (EM) technique [11].

Lee and Yoo [45] claimed that SGM is not accurate enough to approximate the skin color distribution because of the asymmetry of the skin cluster with respect to its density peak. They proposed an elliptical boundary model based on their observations that the skin cluster is approximately elliptic in shape. The elliptical boundary model is defined as:

$$\Phi(\mathbf{c}) = (\mathbf{c} - \boldsymbol{\phi})^T \Lambda^{-1} (\mathbf{c} - \boldsymbol{\phi}) \quad (11)$$

$\boldsymbol{\phi}$  and  $\Lambda$  are two model parameters to be estimated from training data:

$$\phi = \frac{1}{n} \sum_{i=1}^n \mathbf{c}_i \quad , \quad \Lambda = \frac{1}{N} \sum_{i=1}^n f_i (\mathbf{c}_i - \boldsymbol{\mu})(\mathbf{c}_i - \boldsymbol{\mu})^T \quad , \quad \boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^n n f_i \mathbf{c}_i \quad (12)$$

where  $n$  is the number of distinctive training color vectors  $\mathbf{c}_i$  of the training skin pixels,  $f_i$  is the number of skin samples of color vector  $\mathbf{c}_i$ , and  $N$  is the total number of samples ( $N = \sum_{i=1}^n n f_i$ ). An input pixel  $\mathbf{c}$  is classified as skin if  $\Phi(\mathbf{c}) < \theta$  where  $\theta$  is a threshold value.

### 3.3 Comparison and Discussion of Skin Color Distribution Models and Color Representations

Comparative assessment of skin color detection methods in different color representations has been discussed in many literatures [8, 41, 51, 53, 60, 63, 64], but they report different results mainly due to their different experimental conditions (e.g. selection of training/testing datasets). In this section, we will highlight some general conclusions derived from existing researches. Typically, the performance of skin color detection is measured using true positive rate (TPR) and false positive rate (FPR), defined as follows:

$$TP = \frac{\text{number of correctly identified skin pixels}}{\text{total number of skin pixels}}$$

$$FP = \frac{\text{number of falsely identified non-skin pixels as skin pixels}}{\text{total number of non-skin pixels}} \quad (13)$$

Most classification methods have an adjustable threshold parameter that controls the classifier decision boundary. As a result, each threshold value produces a pair of FP and TP values, generating a receiver operating characteristics (ROC) curve which demonstrates the relationship between TP and FP in different threshold values. For comparative evaluation of different classifiers, ROC performance is often summarized by a single scalar value, the area under ROC curve (AUC). AUC is known to be a fairly reliable performance measure of the classifier [14]. Since the AUC is a subregion of the unit square, its value lies between 0 and 1, and larger AUC value implies better classification performance.

A piecewise boundary model has fixed decision boundary parameters and hence, the corresponding ROC plots have only one point. Its boundary parameter values differ from one colorspace to another and one illumination to another. Although methods in this category are computationally fast, in general, they suffer from high FP rates. For example, Phung et al. [51] indicated that piecewise boundary classifier in CbCr space [7] achieves 92 % TP at 28 FP on Edith Cowan University (ECU) dataset [51] (consists of 4K color images from web or taken with digital camera, containing skin pixels), while both 3D SGM classifier of skin/non-skin (YCbCr) and Bayesian classifier with 3D histogram (RGB) achieves higher than 95 % TP at the same FP.

The Gaussian distribution based classifiers, e.g. SGM and GMM, and an elliptical model classifier have been widely used for skin color analysis since they generalize well with small amount of training data. In order to compare the performance of SGM and GMM, Caetano et al. [6] conducted comparative evaluation in normalized-rg colorspace using a dataset of 800 images from various ethnic groups (publicly not available). The authors noted that: i) GMM generally outperforms SGM for FP rates higher than 10 %, while both models yield similar performance for low FP rates, ii) detection performance remains unchanged except minor fluctuations when increasing mixture components for GMM from 2 to 8. Fu et al. [18] performed similar comparative assessment of both Gaussian classifiers using Compaq dataset [36] and confirmed that increasing mixture components doesn't provide significant performance improvement for  $n > 5$  in four representative colorspace (RGB, YCbCr, HSV, and normalized RGB). This is due to an overfitting issue, implying that a classifier describes training sample well but is not flexible enough to describe general samples. Moreover, using GMM is slower during classification since multiple Gaussian components must be computed to obtain the probability of a single color value. Therefore, one should be careful in selecting appropriate number of mixture components.

The performance of the representative non-parametric method, Bayesian classifier with 3D histogram [36], has been compared with other parametric approaches in [36, 51]. The histogram technique in 3D RGB color space achieved 90 % TP (14.2 FP) on Compaq database, slightly outperforming GMM or SGM in terms of detection accuracy. But it requires a very large training dataset to get a good classification rate, as well as higher storage space. For example, a 3D RGB histogram with 256 bins per channel requires more than 16 millions entries. To address this issue, some literature presented color bin quantization method to reduce color cube size. Jones and Rehg [36] compared the use of different numbers of histogram bins ( $256^3$ ,  $32^3$ ,  $16^3$ ) and found that  $32^3$  histogram performed best, particularly when small amount of training data was used.

Often, skin detection methods solely based on color cue in Sect. 3.2 (summarized in Table 2) are not sufficient for distinguishing between skin regions and skin-colored background regions. In order to minimize false acceptance of skin-colored background objects as skin regions, textural and spatial properties of skin pixels can be exploited [10, 40, 69]. Such methods generally rely on the facts that skin texture is smoother than other skin similar areas. For example, Wang et al. [69] initially generated a skin map via pixel-wise color analysis, then carried out texture analysis using Gray-Level Co-occurrence Matrices (GLCM) features to refine the original skin map (i.e. remove false positives). Khan et al. [40] proposed a systematic approach employing spatial context in conjunction with color cue for robust skin pixel detection. At first, a foreground histogram of probable skin colors and a background histogram of the non-skin colors are generated using skin pixel samples in the input image extracted via a face detector. These histograms are used to compute foreground/background weights per pixel, representing the probability of each pixel being skin or non-skin. Subsequently, spatial context is taken into account by



**Table 2** Summary of various skin color detection methods (In characteristic column, + and – represents pros and cons respectively)

Category	Method	Characteristic
<b>Piecewise boundary</b>	Chai (YCbCr) [7], Gomez (normalized RGB) [23], Herodotou (HSV) [30], Kovac (RGB) [42]	+ Simple implementation – Limited flexibility due to fixed threshold and high false positive rate
<b>Non-parametric</b>	Brown’s self organizing map (SOM) [4], Jones’s Bayesian approach with 3D histogram [36], Zarit’s lookup table (LUT) [77]	+ Higher detection accuracy and less dependency on choice of colorspace – Require larger amount of training data and storage compared to parametric solutions
<b>Parametric</b>	Single Gaussian Model [73], Gaussian Mixture Model [24, 32, 74], Lee’s elliptical model [45]	+ Better generalization with less training data – Potential long training delay (for mixture model) and high dependency on choice of colorspace

applying the graph-cut based segmentation on basis of computed weights, producing segmented skin regions of reduced false positives.

Selection of the best colorspace for skin classification is a very challenging task. This problem has been analyzed in numerous literatures with various combinations of skin color distribution models and training/testing datasets. In general, effectiveness of specific color representation in skin color detection can be measured based on their separability between skin and non-skin pixels, and robustness towards illumination variation. However, there is no single best colorspace that is clearly superior to others in all images and often only marginal improvement can be achieved by choice of colorspace.

The effectiveness of colorspace is also dependent on selection of skin color distribution model. For example, non-parametric models, such as histogram-based Bayes classifier are less sensitive to colorspace selection compared to parametric modeling schemes, such as SGM and GMM [1, 38, 65]. Some literatures indicate that transforming 3D colorspace to 2D by discarding the luminance component may enhance skin detection performance since chrominance components are more important cue for determination of skin color. However, elimination of luminance component should be avoided since it decreases classification performance [1, 38, 53], and therefore, is not recommended unless one wants to have faster solution (due to dimensionality reduction) at the cost of classification accuracy.

### 3.4 Illumination Adaptation for Skin Color Detection

Most of the skin color detection methodologies presented in Sect. 3.2 remain stable only to slight variation in illumination since the appearance of color is heavily

dependent on the illumination condition under which the object is viewed. In order to maintain reliable performance over a wide range of illumination conditions, several illumination adaptation schemes have been proposed, which can be subdivided into two main approaches [38]: (i) Dynamic model adaptation: by updating trained skin color models dynamically according to the illumination and imaging conditions, (ii) Color constancy: by pre-processing an input image to produce a transformed version of the input as if the scene is rendered under standard illumination condition.

### 3.4.1 Dynamic Adaptation of Skin Color Model

Dynamic model adaptation approaches usually depend on the results of high-level vision tasks such as face detection and tracking to improve skin color detection performance. Sigal et al. [55] introduced a video-based dynamic adaptation scheme using a second-order Markov model to predict the transition of skin color model over time. Initially, they trained a non-parametric skin color classifier with RGB histogram using Compaq database in offline. This pre-trained model is used to segment the skin color region of the first frame of the input image sequence for tracking purpose. Then skin mask from the first frame is used to re-estimate histogram for skin and non-skin for subsequent frames. In particular, the time varying pattern of skin color distribution in colorspace is parameterized by affine transformations: translation, rotation, and scaling. From the comparative analysis against Jones and Rehg's static histogram approach [36], the authors reported enhanced skin segmentation accuracy in 17 out of 21 testing video sequences (containing illumination conditions ranging from white to non-white light sources, and shadows).

It is possible to adapt the skin color model even for single images, given that input image contains human face regions and they can be accurately located [3, 17, 39, 40, 78]. Zeng and Luo [78] presented an image-dependent skin color detection framework, where an elliptical skin color model trained offline in CIELAB colorspace, is adapted for the input image using a face detector. Skin pixels of given input image is extracted from the face region located by a Viola-Jones face detector [67] and a pre-trained elliptical skin color model is shifted towards the mean value of collected skin samples. During the model shift, the detection boundary threshold is tightened in order to effectively reduce false positives. Kawulok [39] also proposed a systematic solution, which dynamically updates a non-parametric skin color model by exploiting skin samples obtained from Support Vector Machine (SVM) based face detection module. The author noted that by applying the face detector in conjunction with Jones and Rehg's non-parametric method [36], the detection error rate (i.e. sum of false positive rate and false negative rate) is decreased from 26 % to 15 on ECU dataset [51] compared to the absence of the face detection operation. Aforementioned adaptation schemes taking advantage of skin samples extracted from face detectors are not only effective on varying illumination conditions but also beneficial on dealing with the skin color difference between individuals under constant illumination. Pre-trained color model tends to be more general than one estimated on basis of a

present face, thereby fine-tuning the pre-trained model matching to an individual presented in the given image leads to reduction of false skin detections.

Sun [61] proposed an adaptive scheme without deploying additional high-level analysis. Initially, a global non-parametric skin color model [36] is trained in RGB colorspace and potential skin pixels are extracted from the input image via the global model. Then, pixels which are very likely to be skin are identified from initially extracted skin pixels using an accumulated histogram of skin likelihood ratio. For identified skin samples, a local GMM color model is constructed using K-mean clustering. Finally the globally trained skin model and the local skin model are linearly combined to produce a final adaptive model. The performance of this adaptive scheme depends on the number of selected skin pixel samples and the weighting factors for combining two models. The author indicated that it outperforms Jones and Rehg's method [36] on Compaq dataset in terms of detection accuracy, especially in the range of low false positive rate. This approach can be considered as a cost-effective alternative to aforementioned face detection based solutions, but it may not as accurate as them since it only makes use of color feature during adaptation.

Overall, the effectiveness of the dynamic skin color model adaptation depends on the validity of assumptions behind the adaptation criteria. The adaptation schemes generally use a general skin model obtained from a representative image set and then fine-tune it into an image specific model.

### 3.4.2 Color Constancy

Color constancy method attempts to minimize the effect of illumination and imaging condition by preprocessing the input image, instead of adjusting skin color classification model. Color constancy is the ability of human visual system HVS to recognize object color regardless of the illumination conditions. The aim of computational color constancy algorithm is to compensate the effect of the scene illuminant (i.e. light source) on the recorded image in order to recover underlying color of object [22]. Typically, color constancy algorithms can be viewed as a two-stage operation where the first step estimates the characteristics of scene illuminant from the image data, followed by the second step that applies correction on the image to generate a new image of the scene as if it were taken under a canonical (or reference) illuminant<sup>5</sup> (Fig. 6).

#### General formulation of color constancy

Let's consider an image acquisition device equipped with a lens that focuses light from a scene onto an array of sensors. If we assume the spectral power distribution (SPD) of the scene illuminant is constant, the illuminant can be specified by its SPD,  $E(\lambda)$ , which describes the energy per second at each wavelength  $\lambda \in \mathbb{R}$ . The light is reflected from surfaces of objects to be imaged and focused onto the sensor array. The

---

<sup>5</sup> For image reproduction applications, the canonical illuminant is often defined as an illuminant for which the camera sensor is balanced [2].

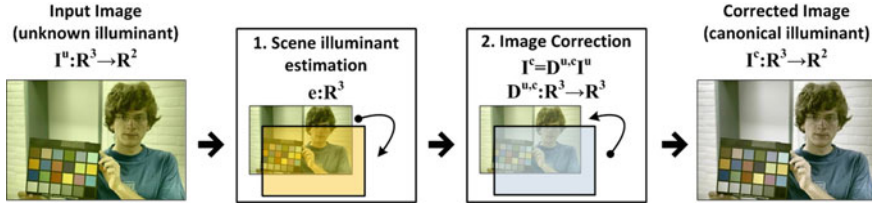


Fig. 6 Overview of color constancy procedure (Sample image is taken from [19])

property of light reflected from an object toward location  $\mathbf{x}$  of the sensor is determined by the surface spectral reflectance,  $S(\mathbf{x}, \lambda)$ . Here,  $\mathbf{x} \in \mathbb{Z}^2$  denotes the spatial position on the 2D sensor array at which the object is imaged. The light arriving at each location  $\mathbf{x}$  on the sensor array is described by the function  $E(\lambda)S(\mathbf{x}, \lambda)$ .

Now, we assume that there are  $p$ -distinct classes of sensors at each location  $\mathbf{x}$ . Typical digital camera devices sample the light by Red, Green, and Blue sensors (i.e.  $p = 3$ ). We denote the sensor spectral sensitivity of the  $k$ -th color channel as  $\rho_k(\lambda)$  ( $k \in \{R, G, B\}$ ). According to Lambertian reflection model<sup>6</sup>, these spectral functions are translated into the RGB values  $\mathbf{I}(\mathbf{x}) = [I_R(\mathbf{x}), I_G(\mathbf{x}), I_B(\mathbf{x})]^T$  within the sensor as follows:

$$I_k(\mathbf{x}) = \int_{\omega} \rho_k(\lambda) E(\lambda) S(\mathbf{x}, \lambda) d\lambda \tag{14}$$

where the integral is taken over the entire visible spectrum  $\omega$  (wavelength of approximately from 400 nm to 700). Eq. (14) shows that the responses induced in a camera sensor depend on the spectral characteristics of the illuminant and the surface. Essentially, color constancy problem can be posed as recovering an estimate of illuminant spectra  $E(\lambda)$  or its projection on the RGB space

$$\mathbf{e} = [e_R, e_G, e_B]^T = \int_{\omega} \boldsymbol{\rho}(\lambda) E(\lambda) d\lambda \tag{15}$$

from given sensor responses  $\mathbf{I}(\mathbf{x})$ .

Without prior knowledge, the estimation of  $\mathbf{e}$  is an under-constrained problem. In practice, color constancy algorithms rely on various assumptions on statistical properties of the illuminants, and surface reflectance properties to estimate  $\mathbf{e}$ . Once the illuminant is estimated, then all colors in the input image, taken under an unknown illuminant, are transformed to colors as they appear under the canonical illuminant. Each pixel of the image under an unknown illuminant  $\mathbf{I}^u = [I_R^u, I_G^u, I_B^u]^T$  can be mapped to the corresponding pixel of the image under a canonical illuminant  $\mathbf{I}^c =$

<sup>6</sup> Lambertian reflection model explains the relationship between the surface reflectance and color image formation for flat, matte surfaces. Although this model does not hold true for all materials, it provides a good approximation in general, and thus widely used in design of tractable color constancy solutions

$[I_R^c, I_G^c, I_B^c]^T$  by a transformation matrix,  $\mathcal{D}^{u,c} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$

$$\mathbf{I}^c = \mathcal{D}^{u,c} \mathbf{I}^u \tag{16}$$

Most existing algorithms make use of a diagonal matrix for this transformation. The diagonal model maps the image taken under an unknown illuminant to another simply by treating each channel independently:

$$\mathbf{I}^c = \mathcal{D}^{u,c} \mathbf{I}^u \Rightarrow \begin{pmatrix} I_R^c \\ I_G^c \\ I_B^c \end{pmatrix} = \begin{pmatrix} d_R & 0 & 0 \\ 0 & d_B & 0 \\ 0 & 0 & d_G \end{pmatrix} \begin{pmatrix} I_R^u \\ I_G^u \\ I_B^u \end{pmatrix} \tag{17}$$

where diagonal entries of  $\mathcal{D}^{u,c}$  can be computed as:

$$d_k = e_k / \left\{ \sqrt{3(e_R^2 + e_G^2 + e_B^2)} \right\} \tag{18}$$

This model is derived from the Von Kries hypothesis [43] that human color constancy is an independent gain regulation of the three cone signals, through three different gain coefficients.

### Representative color constancy algorithms

Various color constancy algorithms have been proposed in the literatures [22], and they can be categorized into two main groups: (i) static approach : estimates the illuminant solely based on the content of a single image with certain assumptions on general nature of color images, (ii) learning approach : requires training data in order to build a statistical model prior to estimation of the scene illuminant. Among available solutions, static approaches, such as Grayworld [5], White Patch [44], Shades of Gray [15], and Gray Edge [71] have been widely used in practical applications due to their simple implementation and fast execution speed. The most widely used Grayworld (GW) algorithm [5] assumes that the average reflectance in a scene is gray (i.e. achromatic) under a neutral illuminant, and thus any deviation from gray is caused by the effects of the illuminant. Hence, the RGB value of the illuminant in the image  $\mathbf{I}$ ,  $\mathbf{e} = [e_R, e_G, e_B]^T$ , can be estimated by computing the average pixel value:

$$\int_{\mathbf{x}} \mathbf{I}(\mathbf{x}) d\mathbf{x} = k \mathbf{e} \tag{19}$$

where  $\mathbf{I}(\mathbf{x})$  is RGB value of the two-dimensional spatial coordinates  $\mathbf{x} \in \mathbb{Z}^2$ , and  $k$  is a multiplicative constant. Another popular algorithm, the White Patch (WP) [44] assumes that a surface with perfect reflectance property<sup>7</sup> exists in the scene, and the color of the perfect reflectance is the color of the scene illuminant:

---

<sup>7</sup> A surface with perfect reflectance property reflects the incoming light in the entire visible spectral range (between wavelengths of about 400 and 700 nm of the electromagnetic spectrum)

$$\max_{\mathbf{x}} \mathbf{I}(\mathbf{x}) = \left[ \max_{\mathbf{x}} I_R(\mathbf{x}), \max_{\mathbf{x}} I_G(\mathbf{x}), \max_{\mathbf{x}} I_B(\mathbf{x}) \right]^T = k\mathbf{e} \quad (20)$$

Finlayson and Trezzi [15] demonstrated that GW and WP are two different instantiations of a more general color constancy algorithm based on the Minkowski norm. This method is called Shades of Gray (SoG) and is computed by:

$$\left[ \int (\mathbf{I}(\mathbf{x}))^p d\mathbf{x} \right]^{(1/p)} = k\mathbf{e} \quad (21)$$

where  $p$  is the Minkowski norm. For  $p = 1$ , the equation is equivalent to the GW assumption, while for  $p = \infty$ , it is equivalent to color constancy by WP. The authors investigated the performance of the illuminant estimation with various  $p$  values and reported that the best results are obtained with a Minkowski norm of  $p = 6$  across many dataset.

Aforementioned methods (GW, WP, and SoG) use only RGB pixel values to estimate the illuminant of an image, completely ignoring other information. More recently, Weijer et al. [71] extended pixel based color constancy methods to incorporate derivative information, which resulted in the Gray Edge (GE) algorithm. Gray edge is based on the hypothesis that the average of the reflectance differences in a scene is achromatic. Under Gray Edge assumption, the color of light source can be computed from the average color derivative in the image given by:

$$\left[ \int |\mathbf{I}_{\mathbf{x}}^{\sigma}(\mathbf{x})|^p d\mathbf{x} \right]^{(1/p)} = k\mathbf{e} \quad (22)$$

where subscript  $\mathbf{x}$  indicates the spatial derivative, and  $\mathbf{I}^{\sigma}$  is a convolution of the image  $\mathbf{I}$  with a Gaussian smoothing filter  $\mathbf{G}$  with standard deviation  $\sigma$ .

It is worthwhile to note that due to under-constrained nature of illumination estimation problem, no single color constancy method is superior to others in all images and it may yield suboptimal result when its underlying assumption doesn't hold true. Aforementioned methods assume that the scene is lit by a single illuminant; although in reality this assumption is often violated due to the presence of multiple illuminants. Furthermore, assumptions on scene statistics may not be correct for given input image. For example, the underlying assumption of GW algorithm fails when the image contains dominant colors in the scene (e.g. image of forest, or ocean), since the average color won't be gray. In such cases, the application of GW algorithm will result in under or over-compensated scene. Such problems can be addressed by exploiting more advanced solution, such as the one based on extensive training data or complex assumptions. However, in general, aforementioned four color constancy algorithms are known to yield acceptable performance at low computational cost, and thus are suitable for practical face detection systems [12].

### 4 Case Study 1 : Use of Skin Color Cue in Image-Based Face Detection System

In this section, we demonstrate a case study of usage of color cue in the practical face detection system. A face detection framework exploiting a representative image-based approach is designed and a skin color classification module is integrated into the system to provide complementary information. In order to reduce the effect of scene illumination on detection performance, illumination compensation is performed on the input color image prior to facial analysis. The overview of the proposed framework is illustrated in Fig. 7.

For this study, we exploit the Boosting based face detection framework with LBP feature due to its superior performance in real-life face detection applications. Among LBP variants, MBLBP [80] is selected since: (i) MBLBP feature is an advanced version of the original LBP feature with high discriminative power, capable of extracting not only local texture as the original LBP, but also larger-scale structure information, (ii) Computation of MBLBP feature can be very fast by using integral image. Consequently, this particular case study allows us to demonstrate the contribution of skin color cue within state-of-the-art texture-based face detection framework. To build MBLBP based detector, we adopted the training procedure and design of Zhang et al.'s framework in [80] (More detail is provided in Sect. 4.1.3).

Following two experiments have been conducted for the performance evaluation.

1. The effectiveness of skin color cue in terms of improving detection accuracy and computational efficiency of the texture based facial analysis is evaluated by comparing two scenarios:

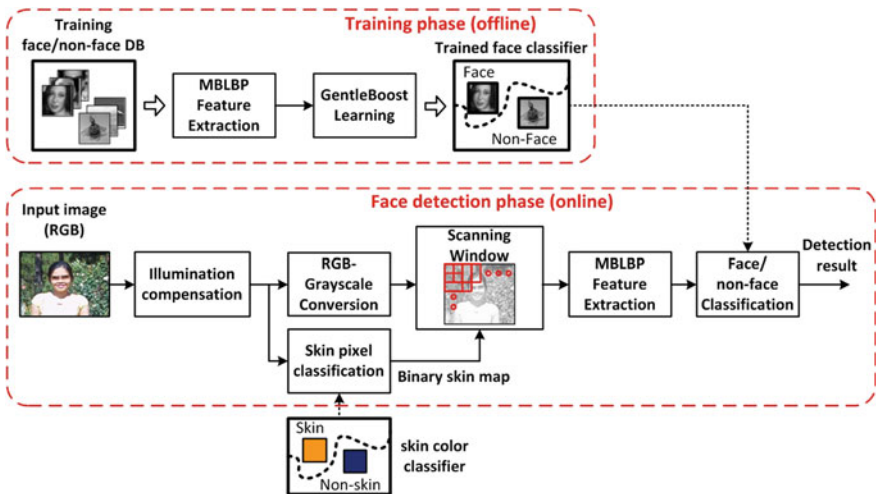


Fig. 7 Overview of the proposed image-based face detection pipeline exploiting MBLBP feature and skin color cue

- Texture feature only pipeline : Face detection is carried out without exploiting color information to measure the performance of the detector solely based on grayscale texture. To facilitate this scenario, illumination compensation and skin color classification blocks are disabled and all remaining operations are performed in grayscale domain.
  - Hybrid pipeline (color feature in conjunction with texture) : Skin color classification block is enabled, and the binary skin map is generated from input color image to identify skin pixels.
2. The importance of stable color representation in face detection analysis is demonstrated. Several representative color constancy methods are applied to color input image to eliminate the color bias caused by non-standard illumination condition. Then, comparative assessment is done by comparing detection accuracy with illumination compensation enabled and disabled.

## 4.1 Proposed Face Detection Framework

The proposed face detection framework consists of three main components: (i) illumination compensation, (ii) skin color classification, (iii) MBLBP feature based face detection module. In this section, brief overview of each component is provided.

### 4.1.1 Illumination Compensation

Illumination compensation module allows the face detection framework to maintain stable performance over wide range of illumination conditions. In this module, the input RGB color images  $X : \mathbb{R}^2 \rightarrow \mathbb{R}^3$  is processed by color constancy algorithm to produce the corrected RGB image  $I : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ . In this experiment, we make use of four representation color constancy algorithms: Grayworld, White Patch, Shades of Gray, and Gray Edge. They are cost-effective algorithms with simple implementation and thus, suitable for practical image processing applications.

### 4.1.2 Skin Color Classification

For the corrected color image  $I : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ , skin color classification module performs a pixel-wise binary classification and generates a binary skin map,  $sMap : \mathbb{R}^2 \rightarrow \mathbb{R}$ , where  $sMap(\mathbf{x}) = 1$  if  $I(\mathbf{x}) \in w_s$ , while  $sMap(\mathbf{x}) = 0$  if  $I(\mathbf{x}) \in w_n$  ( $\mathbf{x} \in \mathbb{Z}^2$  is two-dimensional spatial coordinates in the image). In order to perform classification (online), statistical color models are constructed using training skin/non-skin samples (offline). In our experiment, GMM models are derived to represent both skin and non-skin color distributions and the likelihood ratio test is applied to classify each pixel into skin/non-skin class. In other words,  $p(\mathbf{c}|w_s)$



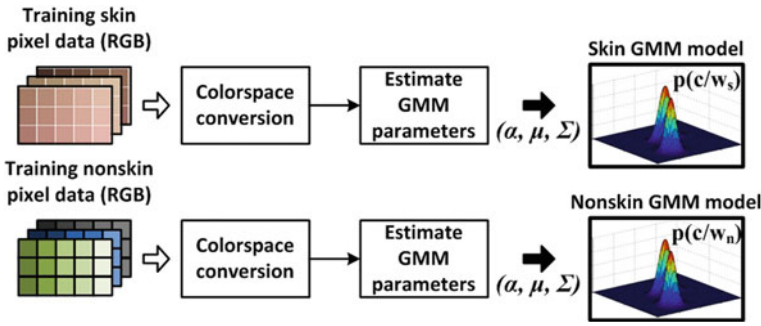


Fig. 8 Overview of skin color classifier training process

and  $p(c|w_n)$  are directly computed from given skin and non-skin Gaussian models, and  $\frac{p(c|w_s)}{p(c|w_n)}$  is compared with a threshold value for classification as described in 7. GMM is selected due to following reasons: (i) it generalizes well with relatively small number of training samples, (ii) compared to other parametric models, it provides more reliable means to represent multi-modal distribution of human skin color under varying illumination.

### Skin Classifier Training/Testing

We examined a GMM based Bayesian skin classifier with various combinations of mixture components and colorspace to identify its optimal configuration. Skin color classifiers are trained in five commonly used colorspace in skin color analysis, including RGB, normalized RGB, YCbCr, HSV, and Cartesian-HSV (denoted as cHSV). For each colorspace, we examined up to four Gaussian mixtures for skin/non-skin pair, allowing us to test 20 combinations in total<sup>8</sup>. In order to train GMM models of skin and non-skin color distributions, a dataset of 1923 images (1014 containing human skin pixels, 909 without human skin pixels) are collected from world wide web. The skin subset contains images of Asian (311 images), Caucasian (319), and Dark skin group (384), while the non-skin subset contains images of art/illustrations (233), Natural Scene (558), and Product (118). All collected images are in JPEG format with sRGB 8-bit representation, and uniform in spatial resolution as  $1024 \times 768$  (both landscape/portrait format) to ensure each image has almost equivalent contribution in total training pool. For training, skin and non-skin pixels in training data are converted to each colorspace and GMM parameters are estimated by EM algorithm along with k-mean clustering initialization, adopting a recommendation from [74] (Fig. 8).

The skin pixel classification performance was tested on 2000 color images from Compaq database [36], containing skin color pixels from various ethnic backgrounds

<sup>8</sup> Assigning more mixture components for non-skin class than skin class is beneficial due to less compact shape of non-skin sample distribution. However, we found that performance gain from having more components for non-skin class is marginal and thus we maintain the same number of components for both classes in this experiment.

**Table 3** The AUC index of skin color classification performance of 20 combinations

Number of mixtures	Colorspace				
	RGB	HSV	YCbCr	cHSV	nRGB
1	0.8901	0.8723	0.8901	0.8868	0.8513
2	0.8967	0.8991	0.8928	0.8979	0.8772
3	0.8956	0.8932	0.8955	0.8970	0.8788
4	0.8961	0.8923	0.8967	0.8996	0.8713

and illumination conditions. This database is used for validation since it is one of the most frequently used benchmark databases from research community with pre-defined ground truth information. Images in Compaq database are given in JPG or GIF file format with sRGB 8-bit representation. Detection performance of various experimental configurations are compared in AUC measure (Table 3).

Following observations are made from this experiment:

1. The normalized RGB provides the worst classification power among five colorspaces since luminance component is lost during its transformation from 3D to 2D, implying that discarding luminance component should be avoided to achieve highly accurate skin color classification.
2. GMM outperforms SGM in all five colorspaces, and particularly in HSV colorspaces where skin color distribution doesn't form a compact cluster. In general, increasing mixture components results in high classification performance upto certain number of components before overfitting occurs (e.g. for HSV colorspace, having more than two mixture components degrades detection performance). We observed that the best skin detection performance can be achieved by using Cartesian HSV colorspace with four mixture components for both skin and non-skin GMM models. Therefore, it is selected as our main trained classifier model and used throughout subsequent sections.

### 4.1.3 MBLBP Feature Based Face Detection System

This section briefly describes the MBLBP feature based detection system which contains all remaining modules in Fig. 7 except aforementioned two modules. Similarly to Viola-Jones's framework [67], the proposed framework performs exhaustive search on the input image to determine the existence of the face. This requires the grayscale version of input, which can be obtained by converting input image  $I$  from RGB to YCbCr and retaining only Y channel. Instead of scanning whole image for face search, the proposed system uses the generated binary skin mask  $sMap$  to narrow down search region. Adopting the strategy from [12], particular sub-window is examined during face search, only if it contains sufficient number of skin color pixels. Let  $W_k$  is the  $k$ th sub-window examined during iterative window scan to determine whether this sub-window is face image or not. MBLBP based face/non-face classification is carried out only if:

$$\frac{\sum_{\mathbf{x} \in W_k} sMap(\mathbf{x})}{w_k \times h_k} \geq L \quad (23)$$

where  $w_k$  and  $h_k$  are the width and height of window  $W_k$ , respectively, and  $L$  is the threshold for minimum skin pixel count. For our testing dataset, we found  $L = 0.4$  yields satisfactory results.

#### MBLBP based face detector training

To train MBLBP based face detector, 9916 gray scale face images in  $24 \times 24$  pixels are collected by combining Viola and Jones dataset and Ole Jensen dataset [34]. These face images contain large variations including but not limited to pose, facial expression, and illumination conditions. The baseline size ( $24 \times 24$  pixels) is applied directly to training as in [67]. Furthermore, more than 100K of negative training image patches are extracted from 2,000 high resolution non-face images collected from the web. For consistency, the negative image patches are resized to the same size as face image patches. From the entire set, 7,916 face images and 10,000 non-face images are randomly extracted to train classifier, and another independent 2,000 face images and 10,000 non-face images are randomly selected for validation.

The face detector is trained aiming for high detection accuracy of approximately 97 % detection rate. To achieve this, the stage classifiers are trained to have higher than 99 % true positive rate (TPR) on validation dataset. In addition, a pre-assigned false positive rate (FPR) is achieved on validation dataset by adjusting the number of features and threshold for each stage. For fast processing, only a small number of features are used in the initial stage allowing a relatively high FPR (around 50 %). Each succeeding stage is designed to use increased number of features to reduce FPR by around 10 % from its prior stage. The number of stages is increased until the desired overall performance is achieved. Overall, the face detector achieves 96.9 % TPR and  $7.94 \times 10^{-6}$  FPR on validation dataset. This particular TPR and FPR configuration is chosen adopting general practice from the literatures<sup>9</sup>. GentleBoost machine learning algorithm is used to train the classifier due to its exceptional performance among the variants of Boosting algorithm [46].

## 4.2 Experimental Results

In order to evaluate the proposed face detection method, we have chosen the Bao color face database [16]. The Bao database is chosen due to following two reasons: (i) it represents real-life scenarios by containing wide variety of images, including faces of various ethnic groups (Asian, Caucasian, and Dark skin group), poses (frontal and non-frontal), illumination conditions (indoor and outdoor), and image

---

<sup>9</sup> Viola and Jones [67] indicate that around  $1 \times 10^{-6}$  of FPR is a common value for practical uses. However, it is extremely difficult to achieve the precise value and generally it is acceptable if FPR is within the same magnitude. For instance, Jun and Kim [37] achieves 96 % TPR at  $2.56 \times 10^{-6}$  FPR, and Louis and Plataniotis [47] achieves 92.27 % TPR at  $6.2 \times 10^{-6}$  FPR

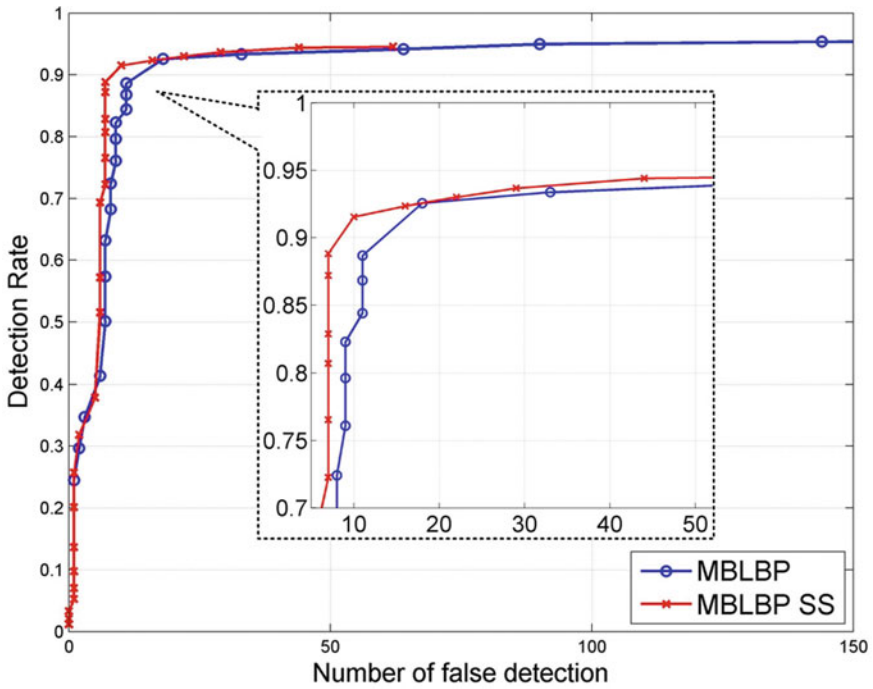
**Fig. 9** Example of groundtruth face box generation (Sample image is taken from [16])



resolutions (from  $57 \times 85$  to  $1836 \times 1190$ ), (ii) it is publicly available database which is widely used for evaluation of face detector [12, 68]. All images in this dataset are provided in 8-bit JPEG format and we used 117 images containing a single face and 220 images containing multiple faces, examining a total of 1360 faces. Since the original Bao database does not contain the groundtruth information for the face locations, we manually generated groundtruth by locating a rectangle using the eye locations (Fig. 9). The correctness of face detection hypothesis is evaluated based on the groundtruth and the following two criterions [70]: (i) the Euclidean distance between the hypothesis face box center and groundtruth face box center must be within 30 % of the groundtruth width, (ii) the detection hypothesis width must be within 50 % of the groundtruth width.

In Fig. 10, the Free Receiver Operator Characteristic (FROC) is illustrated to compare two detection pipelines: MBLBP texture only pipeline (denoted as MBLBP) and hybrid of texture and color pipeline (denoted as MBLBP SS). FROC is similar to ROC except it plots the detection rate (DR) versus number of false positive detections instead of false positive rates. As can be seen, by incorporating color information, the proposed MBLBP feature based face detection yields enhanced detection accuracy over almost entire range of number of false positive (nFP). Since the texture only pipeline discards the chrominance information, it will generate false positive if the scene contains a background object of face-like texture pattern. By using skin color information, such false positives can be successfully filtered out during face search. The degree of improvement is approximately 1 % increase in TP from 93 % to 94 at nFP of 40. In Fig. 11, the detection results of both pipelines are compared on three samples images from Bao database, containing various skin types in indoor and outdoor lighting conditions. The figures demonstrate that all false positives have been successfully eliminated by exploiting skin color cue.

To compare computational complexity of both pipelines, we measure the total number of scanned sub-windows during face search for all images in Bao dataset. The number of scanned sub-windows is an important cue for computational speed, since only a subset of sub-windows which are sufficiently populated with skin color pixels are scanned for hybrid pipeline while all sub-windows have to be scanned for texture only pipeline. In addition, we measure the total execution time to process all 337 images in Bao dataset to evaluate computational speed in real MATLAB



**Fig. 10** FROC of face detection result on Bao dataset using MBLBP texture only pipeline and hybrid pipeline

**Table 4** Computational complexity comparison between two pipelines

	MBLBP-texture	Hybrid
Total number of scanned sub-window during exhaustive search	43387629	9150270
Total processing delay (in seconds)	62930	14286

implementation<sup>10</sup>. Experimental results are obtained on Core 2 Duo 3.0 GHz CPU with 4GB RAM running Windows 7 operating system. Although there is software overhead in MATLAB implementation, the obtained results are proportional to the number of scanned windows, as shown in Table 4. In hybrid pipeline, only 2.75 % of total execution delay (i.e. 393s out of 14286s) accounts for skin color detection, while more than 95 % of delay accounts for texture analysis, demonstrating computational efficiency of color analysis. Overall, hybrid pipeline not only allows us to enhance detection accuracy by removing false positives with non-skin color, but also significantly reduces computational complexity.

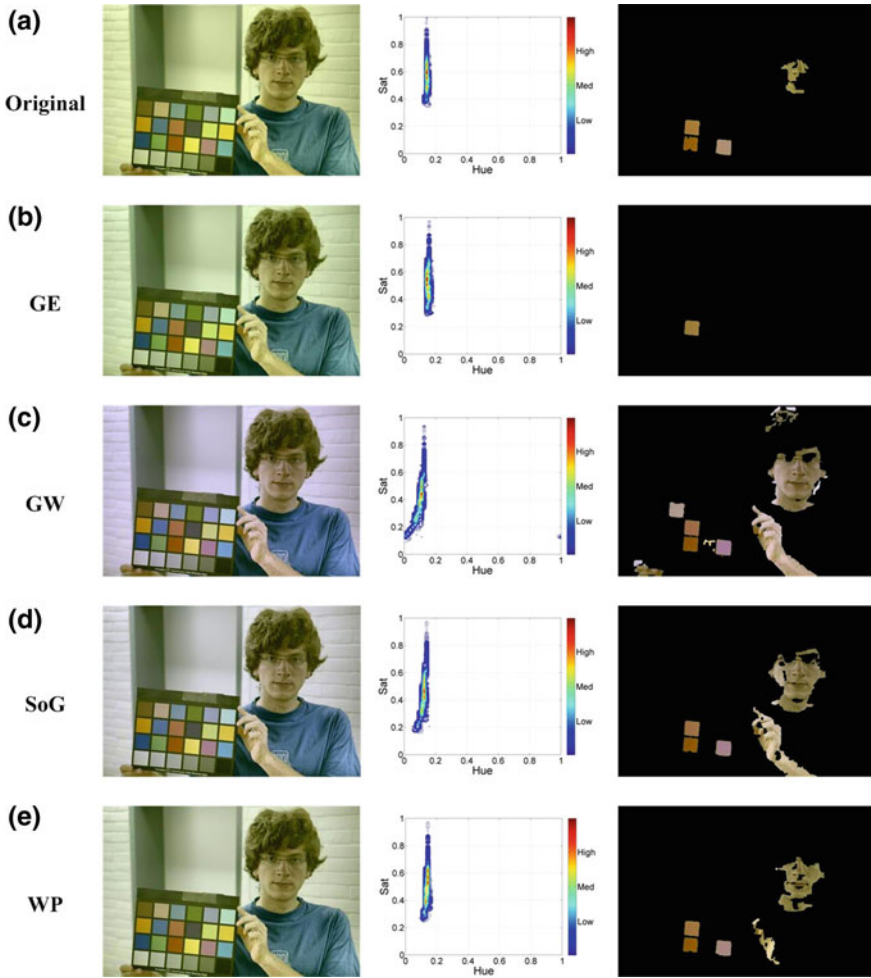
<sup>10</sup> Instead of measuring average number of scanned window and execution time per frame, we measured the sum of them, since test images in Bao database vary in spatial resolutions



**Fig. 11** Detection results of the proposed face detector for sample images from the Bao database [16]: upper images are obtained with MBLBP texture only pipeline whereas lower images are obtained with hybrid pipeline

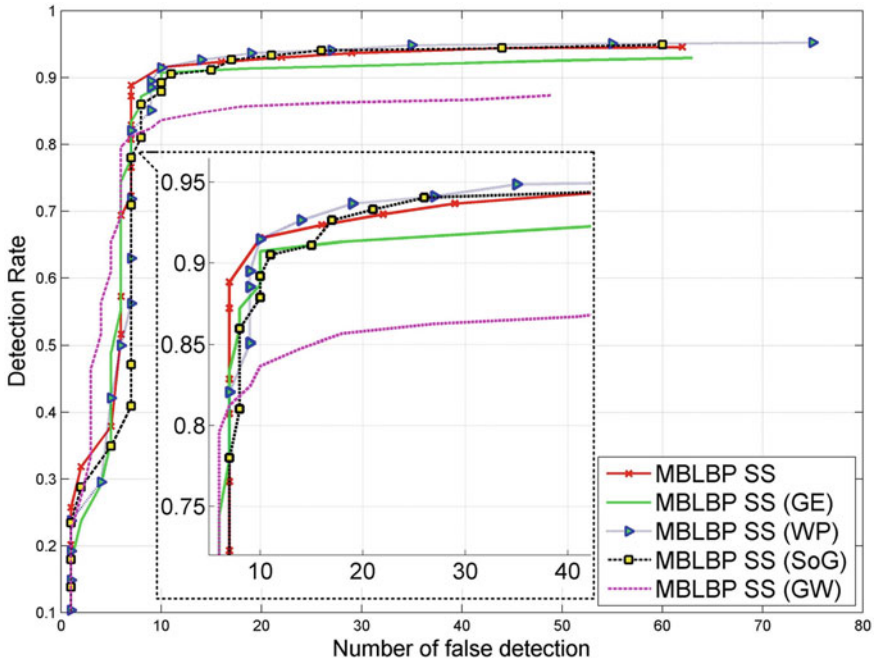
Effectiveness of color constancy solution is evaluated in two different aspects that: (i) color constancy algorithm is applied to images under abnormal lighting condition and its impact on face detection performance is analyzed, (ii) color constancy algorithm is applied to images under generic lighting condition (e.g. images from standard image database) and its impact on face detection performance is analyzed. Such analysis allows us to identify color constancy solutions that improves detection performance on images under challenging illumination condition while providing comparable performance on general images.

Figure 12 presents a sample image rendered under indoor lighting with yellow cast, and adjusted images using four color constancy algorithms. The skin pixels of each image are manually extracted, and their RGB values are converted to HSV and plotted in the H-S plane. As can be seen, the skin color distribution of the original image is slightly biased towards yellow hue than the one under normal lighting condition (Refer to Table 1 for commonly accepted hue values for skin color). Consequently, skin color classifier fails to detect most of skin pixels in the scene, which eventually results in face localization error. By applying color constancy algorithm, the reliability of skin color classification is improved (e.g. with GW, SoG, and WP algorithms), allowing subsequent texture analysis to be performed in detected skin region. However, Fig. 12b shows that failure of illumination compensation may lead to even severe color distortion on skin pixels. It demonstrates that the performance of color constancy is image dependent and thus, to obtain meaningful measure of algorithm accuracy, average performance over a set of images should be assessed. In Fig. 13, the effectiveness of color constancy algorithms are evaluated on Bao database. Experimental results indicate that applying color constancy algorithms on large-scale image dataset generally yields comparable or slightly higher detection



**Fig. 12** Skin color pixel distribution in H-S plane and skin pixel detection results on a sample image taken from Color Checker Dataset [19] : **a** Original image, **b–e** images processed by GE, GW, SoG, and WP color constancy algorithms, respectively

performance, except for the GW algorithm, where DR drops significantly by approximately 5%. It implies that redundant illumination compensation on images under generic illumination condition may deteriorate face detection performance and therefore algorithm should be carefully validated over a wide variety of images to build robust real-world solution.



**Fig. 13** FROC of face detection result on Bao dataset using hybrid pipeline with various illumination compensation methods

## 5 Case Study 2 : Use of Skin Color Cue in Feature-Based Face Detection System

In Sect. 4, we have demonstrated that color provides complementary information to reduce falsely detected faces, while minimizing face search delay in image-based face detection system. However, the aforementioned detection system has two major limitations: (i) even with complementary color information, it is still found to be computationally intensive due to exhaustive search process, (ii) its performance drops significantly when face is not frontal upright since training of system is done with frontal upright faces. Several advanced face detection methodologies have been proposed to achieve rotation invariant detection by cascading  $N$  pose specific detectors in parallel [66], but often such system exhibits increased complexity, fails to meet real-time requirements.

In this section, we demonstrate another example of face detection system, where aforementioned limitations of image-based approaches are addressed by using feature-based face detection approach in conjunction with color analysis. As mentioned earlier, color is an attribute that is invariant to rotation and scaling, and therefore, such property can be exploited to improve the robustness of face detection



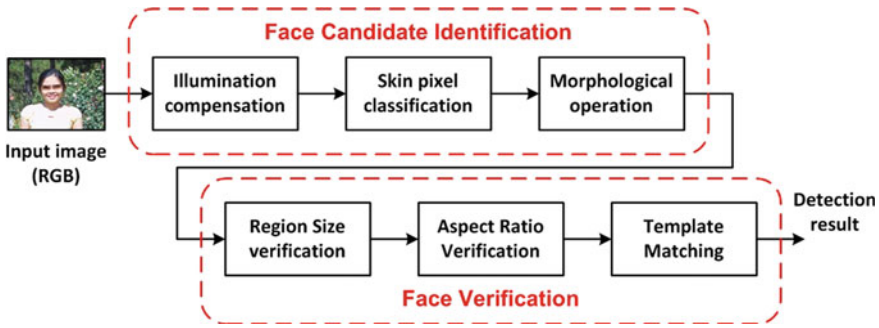


Fig. 14 Overview of feature-based face detection pipeline exploiting skin color cue

system towards rotated face. The workflow of the proposed face-detection algorithm presented in Fig. 14 can be summarized as follows:

- Illumination compensation is performed on the input color image to reduce the effect of prevailing illumination, followed by skin pixel classification to extract skin-like pixels
- By utilizing a series of binary morphological operations, face candidate regions are formed from detected skin pixels
- Each face candidate regions are verified through several analysis to determine whether if corresponding region is a face or not

## 5.1 Proposed Face Detection Framework

### 5.1.1 Face Candidate Identification

We start by extracting skin pixels using the GMM based classifier in Cartesian HSV colorspace outlined earlier in Sect. 4.1.2. Once all of the pixels have been classified by generating binary skin map,  $sMap : \mathbb{R}^2 \rightarrow \mathbb{R}$ , a series of binary morphological operations are subsequently applied to  $sMap$  to refine the extracted skin region. Morphological operation simplifies image data by eliminating detail smaller than the structuring element while preserving their global structural information [28].

We utilize two morphological operations in series: morphological closing followed by morphological opening. Both operations are carried out with a disk-shaped structuring element of radius 3, which provide a good balance between noise reduction and structural detail preservation. Essentially, closing an image with a disk shape smoothes contour; eliminates small holes; and fills gaps within the objects, while opening an image with a disk shape eliminates small islands and sharp peaks. Consequently, object boundaries are smoothen out and small artifacts are effectively removed, achieving semantically meaningful segmentation. Subsequently, cluster of connected pixels are obtained from the refined binary image through connected component labelling to generate set of face candidates to be verified individually.

### 5.1.2 Face Verification

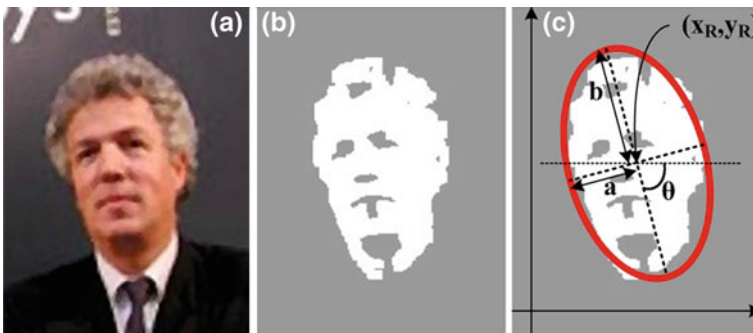
The input to face verification mechanism may contain objects other than the facial areas, such as other part of human body or skin-colored background objects. The verification module exploits general characteristics of human face region to distinguish actual face from a set of candidates. More specifically, we consider following criteria in sequential manner to verify face candidates:

1. Region Size : Any candidate regions of insignificant size are discarded from further verification by imposing minimum size constraint. Clusters of area less than 0.5 % of the image dimension are eliminated since such small regions generally don't correspond to actual face region if image was captured from reasonable distance.
2. Aspect Ratio : Given the geometry of the human face, it is reasonable to expect that the ratio of height to width falls within a specific range. If the dimensions of a candidate object satisfy the commonly accepted dimensions of the human face, then it can be classified as a facial area.

To examine aspect ratio, each face candidate region  $F$  is approximated by the best-fit ellipse using statistical moments [58]. The ellipse is represented by a state parameter  $\mathbf{s} = (x_F, y_F, \theta_F, a_F, b_F)$ , where  $(x_F, y_F)$  is the center of ellipse,  $\theta_F$  is the angle of major axis of the ellipse with the horizontal axis,  $a_F$  and  $b_F$  are the length of minor and major axis of the ellipse (Fig. 15). The parameters are obtained by finding an ellipse that has same normalized central moment as the candidate region. Initially,  $(x_F, y_F)$  is defined as the centroid of the region  $F$ . The normalized second central moments of the region  $F$  can be computed ( $p + q = 2, p \geq 0, q \geq 0$ ):

$$\bar{\mu}_{p,q}(F) = \mu_{p,q}(F) \cdot (1/\mu_{0,0}(F))^{(p+q+2)/2} \quad (24)$$

where  $\mu_{p,q}(F)$  is the central moment of the region  $F$ , defined as:



**Fig. 15** Face candidate region represented by ellipse: **a** Original image, **b** Binary skin map of image **a**, **c** Ellipse matched to face candidate region

$$\mu_{p,q}(F) = \sum_{(i,j) \in F} (i - x_F)^p \cdot (j - y_F)^q \quad (25)$$

The orientation  $\theta_F$  of the major axis ( $\theta_F \in [-\pi/2, \pi/2]$ ) can be found from the central moments:

$$\theta_F = \frac{1}{2} \arctan \left( \frac{2 \cdot \mu_{1,1}(F)}{\mu_{2,0}(F) - \mu_{0,2}(F)} \right) \quad (26)$$

The length of minor and major axis,  $a_F$  and  $b_F$  can be computed as:

$$a_F = 2\sqrt{(\lambda_1/|F|)} \quad , \quad b_F = 2\sqrt{(\lambda_2/|F|)} \quad (27)$$

where  $|F|$  is number of pixels in region  $F$ , and

$$\lambda_1 = \sqrt{(\mu_{2,0}(F) + \mu_{0,2}(F))/2 - \sqrt{4\mu_{1,1}(F)^2 + (\mu_{2,0}(F) - \mu_{0,2}(F))^2}} \quad (28)$$

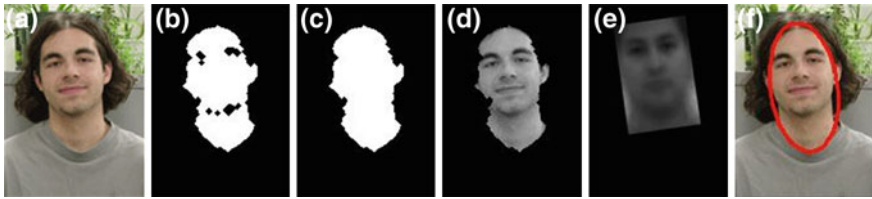
$$\lambda_2 = \sqrt{(\mu_{2,0}(F) + \mu_{0,2}(F))/2 + \sqrt{4\mu_{1,1}(F)^2 + (\mu_{2,0}(F) - \mu_{0,2}(F))^2}}$$

On the basis of the computed ellipse parameter, the candidate regions with the aspect ratio  $b_R/a_R$  in the interval of [1.0, 3.0] are retained as face regions.

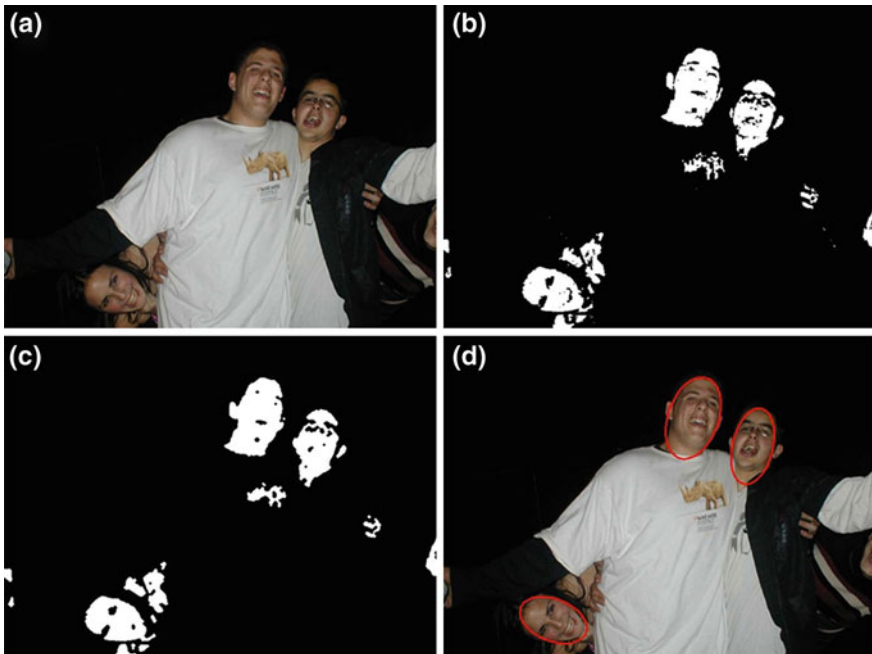
3. **Template Matching** : The final stage of verification involves template matching where pixel intensity comparison between the pre-defined template face and the grayscale face candidate region is performed. The cross correlation is used as a similarity measure between the template and the face candidate. The template face is obtained by averaging frontal face image from CMU-PIE (Carnegie Mellon University Pose, Illumination, and Expression) dataset [56]. Initially, each remaining face candidate region is extracted from grayscale image using the binary mask from previous stage (Any holes within the binary mask are filled to retain eye and mouth features—Fig. 16c). The resultant image is transformed to grayscale for matching (Fig. 16d). Then grayscale template image  $T$  is aligned with considered candidate region (Fig. 16e) by: (i) resizing  $T$  according to the length of minor and major axis ( $a_R, b_R$ ) estimated from previous stage, (ii) rotating by  $\theta_R$  prior to place it on the centroid of the candidate region. Finally cross correlation is evaluated between aligned template and considered face candidate region. In our experiment, the cross correlation threshold is set to 0.6.

## 5.2 Experimental Result

We applied the scheme outlined in Sect. 5.1 to locate face region in images taken from Bao database. Figure 17 shows the procedure used to detect faces in a scene where



**Fig. 16** Overview of template matching process



**Fig. 17** Face detection sequence: **a** Original image, **b** Binary skin map obtained from skin color classification, **c** potential face candidates identified after morphological operations on binary skin map, **d** Final detection result after verification

three faces are successfully detected while other regions classified as skin are rejected from the verification stage. In particular, this example demonstrates the effectiveness of proposed feature-based detection solution in dealing with wider range of face poses than the detector outlined in Sect. 4, where the former successfully located face of almost  $90^\circ$  rotated.

Another strength of this feature-based detection system is its computational speed. Compared to the image-based solution outlined in Sect. 4 which requires exhaustive scanning of the image, the runtime of this solution is much faster since verification only takes place for identified face candidate regions. For example, the MATLAB implementation of the feature-based detection system took only 2146 seconds to

process all 337 images from Bao dataset in the same system environment used in Sect. 4 (Table 5). It is worthwhile to point out that direct comparison of processing speed may not be meaningful as MATLAB implementations are not optimized. However, given the computational complexity of underlying workflow, it is reasonable to assume that feature-based face detection system is much faster even in real-world implementation. Figure 18 illustrates the robustness of the proposed scheme for faces of different ethnicity under various lighting conditions, including indoor and outdoor.

Although the proposed detector yields encouraging results, there are certain types of challenging condition that limit the performance of the system.

- The detector becomes unstable if other part of body (e.g. neck or hand) is placed in contact with a face (Fig. 19a), or if face overlaps with other skin-colored region (e.g. background objects or other faces) in the line of camera’s sight (Fig. 19b). In this case, it’s unable to separate each object into distinct clusters during face

**Table 5** Performance comparison between two face detection systems outlined in Sects. 4 and 5 on 337 images of Bao dataset

Face detection method	No. FP	Detection rate (%)	Processing delay (s)
Hybrid method in Sect. 4	64	94.6	14286
Feature-based method in Sect. 5	318	78.3	2146



**Fig. 18** Examples of the successful detection of faces with different skin colors and poses



**Fig. 19** Examples of the failed detection of faces

candidate identification and verify them individually. This particular scenario often results in generation of false positives or false negatives.

- Template matching tends to be highly tolerable to false positives (e.g. skin-colored background objects, and other part of skin-colored body) and thus, often fails to properly filters out other part of bodies with skin color. (Fig 19c)

Due to aforementioned limitations, face detection system introduced in this section yields suboptimal detection accuracy compared to image-based approach from the last section (See Table 5). Therefore, there exists a tradeoff between computational efficiency and detection accuracy. Following modification can be considered in order to enhance the detection performance of the feature-based detector:

- The true positive rate can be improved by properly segmenting face regions when face is presented in contact with other skin area, or presented in front of skin-colored background during face candidate identification stage. This requires use of other modalities, such as shape or texture, since color cue itself is not sufficient to distinguish between them.
- The number of false positives can be decreased by introducing more constraints into verification stage. For example, [30, 58] impose pose angle constraints that if the vertical orientation of face region is beyond commonly accepted interval, the corresponding region is rejected as non-face.

## 6 Conclusion

The face plays a crucial role in our human social interaction; face conveys people's identity and facial expression can be used as an important means of communication. Therefore, it is not surprising that human facial analysis has been considered as one of the most active areas of research in digital image processing and computer vision applications. While detection of faces is one of the basic tasks for human, it is not trivial task in computer vision, since faces have a high degree of variation in shape, color, and texture depending on imaging condition. In order to build a reliable and robust face detection system, several cues, such as motion, shape, color, and texture have been taken into account. Among available cues, color cue is advantageous due to

its low computational complexity, high discriminative power, and robustness against geometrical transformation under stable illumination condition.

This chapter addressed several frequently encountered issues when using skin color as a feature for face detection, such as: (i) selection of the suitable color representation (i.e. colorspace) to perform color classification, (ii) selection of modeling scheme to represent the skin color distribution, (iii) dealing with its dependence to the illumination condition, (iv) how to apply skin color classification results in high-level analysis system.

Following conclusions can be drawn from this chapter:

1. Skin color analysis for face detection application involves a pixel-wise classification to discriminate skin and non-skin pixels. In order to select an optimal combination of color representation and skin color modeling scheme, the desired performance requirements (in terms of accuracy and processing speed), the available computational resources, as well as the amount of data available for training should be considered.
2. Color constancy algorithms can improve skin pixel classification performance by compensating the effect of scene illuminant in the recorded image and revealing the underlying color of object. In this chapter, we mainly reviewed low-complexity solutions, suitable to be deployed as a pre-processor for face detection framework. Experimentation performed on standard color face database demonstrates that face detection accuracy can be enhanced by applying them prior to skin color analysis.
3. Skin color cue can be utilized in face detection system in following two ways: (i) In image-based face detection approach, the faster and more accurate exhaustive face search can be achieved by using skin color with the purpose of guiding the search. Pixel-level classification of skin and non-skin is sufficient for this purpose, (ii) In feature-based face detection approach, color provides visual cue to focus attention in the scene by identifying a set of skin-colored regions that may contain face objects. Typically, local spatial context of skin pixel distribution is considered after skin color classification by exploiting morphological operations and connected component analysis.

## References

1. Albiol A, Torres L, Delp E (2001) Optimum color spaces for skin detection. In: Proceedings of international conference on image processing, Thessaloniki, vol 1, pp 122–124
2. Barnard K, Cardei V, Funt B (2002) A comparison of computational color constancy algorithms. I: methodology and experiments with synthesized data. *IEEE Trans Image Process* 11(9):972–984
3. Bilal S, Akmeliawati R, Salami M, Shafie A (2012) Dynamic approach for real-time skin detection. *J Real-Time Image Process* pp 1–15
4. Brown D, Craw I, Lewthwaite J (2001) A SOM based approach to skin detection with application in real time systems. In: Proceedings of the British machine vision conference, University of Manchester, UK, pp 491-500

5. Buchsbaum G (1980) A spatial processor model for object colour perception. *J Franklin Inst* 310(1):1–26
6. Caetano T, Olabarriaga S, Barone D (2002) Performance evaluation of single and multiple-Gaussian models for skin color modeling. In: *Proceedings XV Brazilian symposium on computer graphics and image processing, Brazil*, 275–282
7. Chai D, Ngan KN (1998) Locating facial region of a head-and-shoulders color image. In: *Proceedings of the international Conference on face and gesture recognition*, pp 124–129
8. Chaves-Gonzalez JM, Vega-Rodriguez MA, Gomez-Pulido JA, Sanchez-Perez JM, (2010) Detecting skin in face recognition systems: a colour spaces study. *Digital Sig Proc* 20(3):806–823
9. Chen HY, Huang CL, Fu CM (2008) Hybrid-boost learning for multi-pose face detection and facial expression recognition. *Pattern Recogn* 41(3):1173–1185
10. Conci A, Nunes E, Pantrigo JJ, Sánchez A (2008) Comparing color and texture-based algorithms for human skin detection. In: *Computer interaction* 5:166–173
11. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. *J R Stat Soc, Series B* 39(1):1–38
12. Erdem C, Ulukaya S, Karaali A, Erdem A (2011) Combining Haar feature and skin color based classifiers for face detection. In: *IEEE international conference on acoustics, speech and signal processing*, pp 1497–1500
13. Fasel I, Fortenberry B, Movellan J (2005) A generative framework for real time object detection and classification. *Comput Vis Image Underst* 98(1):182–210
14. Fawcett T (2006) An introduction to ROC analysis. *Pattern Recogn Lett* 27(8):861–874
15. Finlayson GD, Trezzi E (2004) Shades of Gray and Colour Constancy. In: *Twelfth color imaging conference: color science and engineering systems, technologies, and applications*, pp 37–41
16. Frischholz R (2008) Bao face database at the face detection homepage. <http://www.facedetection.com>. Accessed 07 Dec 2012
17. Fritsch J, Lang S, Kleinhagenbrock M, Fink G, Sagerer G (2002) Improving adaptive skin color segmentation by incorporating results from face detection. In: *Proceedings of the IEEE international workshop on robot and human interactive communication*, pp 337–343
18. Fu Z, Yang J, Hu W, Tan T (2004) Mixture clustering using multidimensional histograms for skin detection. In: *Proceedings international conference on pattern recognition*, vol 4. Brighton, 549–552
19. Gehler P, Rother C, Blake A, Minka T, Sharp T (2008) Bayesian color constancy revisited. <http://www.kyb.tuebingen.mpg.de/bs/people/pgehler/colour/>. In: *IEEE conference on computer vision and pattern recognition*, pp 1–8
20. Gevers T, Smeulders AW (1999) Color-based object recognition. *Pattern Recognit* 32(3):453–464
21. Gevers T, Gijssenij A, van de Weijer J, Geusebroek JM (2012) *Pixel-based photometric invariance*, Wiley Inc., pp 47–68
22. Gijssenij A, Gevers T, van de Weijer J (2011) Computational color constancy: survey and experiments. *IEEE Trans Image Process* 20(9):2475–2489
23. Gomez G, Morales EF (2002) Automatic feature construction and a simple rule induction algorithm for skin detection. In: *Proceedings of the ICML workshop on machine learning in computer vision*, pp 31–38
24. Greenspan H, Goldberger J, Eshet I (2001) Mixture model for face-color modeling and segmentation. *Pattern Recogn Lett* 22(14):1525–1536
25. Hadid A, Pietikäinen M (2006) A hybrid approach to face detection under unconstrained environments. In: *International conference on pattern recognition*, vol 1:227–230
26. Hadid A, Pietikäinen M, Ahonen T (2004) A discriminative feature space for detecting and recognizing faces. In: *Proceedings IEEE Conference on computer vision and pattern recognition*, vol 2, pp II-797–II-804
27. Hanbury A (2003) Circular statistics applied to colour images. In: *Proceedings of the computer vision winter workshop, Valtice*, pp 55–60



28. Haralick R, Shapiro L (1992) Computer and robot vision, Addison-Wesley Longman Publishing Co Inc, 1st edn. vol 1, Boston
29. Hassanpour R, Shahbahrani A, Wong S (2008) Adaptive Gaussian mixture model for skin color segmentation. *Eng Technol* 31(July):1–6
30. Herodotou N, Plataniotis KN, Venetsanopoulos AN (2000) Image Processing Techniques for Multimedia Processing. In: Guan L, Kung SY, Larsen J (eds) *Multimedia image and video processing*. CRC Press, chap 5:97–130
31. Hjelmås E, Low BK (2001) Face detection: a survey. *Comput Vis Image Underst* 83(3):236–274
32. Hossain MF, Shamsi M, Alsharif MR, Zoroofi RA, Yamashita K (2012) Automatic facial skin detection using gaussian mixture model under varying illumination. *Int J Innov Comput I* 8(2):1135–1144
33. Hsu RL, Abdel-Mottaleb M, Jain A (2002) Face detection in color images. *IEEE Trans Pattern Anal Mach Intell* 24(5):696–706
34. Jensen OH (2008) Implementing the Viola-Jones face detection algorithm. Technical University of Denmark, department of informatics and mathematical modeling, master's thesis, Denmark
35. Jin H, Liu Q, Lu H, Tong X (2004) Face detection using improved LBP under Bayesian framework. In: *Proceeding of the international conference on image and graphics*, pp 306–309
36. Jones MJ, Rehg JM (2002) Statistical color models with application to skin detection. *Int J Comput Vision* 46(1):81–96
37. Jun B, Kim D (2012) Robust face detection using local gradient patterns and evidence accumulation. *Pattern Recognit* 45(9):3304–3316
38. Kakumanu P, Makrogiannis S, Bourbakis N (2007) A survey of skin-color modeling and detection methods. *Pattern Recognit* 40(3):1106–1122
39. Kawulok M (2008) Dynamic skin detection in color images for sign language recognition. In: Elmoataz A, Lezoray O, Nouboud F, Mamassani D (eds) *ICISP Lecture notes in computer science*, vol 5099. Springer, France, 112–119
40. Khan R, Hanbury A, Sablatnig R, Stöttinger J, Khan F, Khan F (2012 a) Systematic skin segmentation: merging spatial and non-spatial data. *Multimed tools appl* pp 1–25
41. Khan R, Hanbury A, Stöttinger J, Bais A (2012 b) Color based skin classification. *Pattern Recognit Lett* 33(2):157–163
42. Kovac J, Peer P, Solina F (2003) Human skin color clustering for face detection. In: *The IEEE region 8 EUROCON 2003 computer tool*, vol 2:144–148
43. von Kries J (1970) Influence of adaptation on the effects produced by luminous stimuli. In: MacAdam D (ed) *Sources of color science*, MIT Press, pp 109–119
44. Land EH (1977) The retinex theory of color vision. *Sci Am* 237(6):108–128
45. Lee JY, Yoo SI (2002) An elliptical boundary model for skin color detection. In: *Proceedings international conference on imaging science, systems, and technology*. pp 81–96
46. Lienhart R, Maydt J (2002) An extended set of Haar-like features for rapid object detection. In: *Proceedings international conference on image processing*, vol 1, pp 900–903
47. Louis W, Plataniotis KN (2011) Co-occurrence of local binary patterns features for frontal face detection in surveillance applications. *EURASIP J Image Video Process* 2011
48. Moon H, Chellappa R, Rosenfeld A (2002) Optimal edge-based shape detection. *IEEE Trans Image Process* 11(11):1209–1227
49. Naji SA, Zainuddin R, Jalab HA (2012) Skin segmentation based on multi pixel color clustering models. *Digital Sig Process* 22(6):933–940
50. Ojala T, Pietikainen M, Harwood D. (1996) A comparative study of texture measures with classification based on featured distributions. *Pattern Recognit* 29(1):51–59
51. Phung S, Bouzerdoum SA, Chai SD (2005) Skin segmentation using color pixel classification: analysis and comparison. *IEEE Trans Pattern Anal Mach Intell* 27(1):148–154
52. Plataniotis KN, Venetsanopoulos AN (2000) *Color image processing and applications*. Springer-Verlag, New York
53. Schugge SJ, Jayaram S, Shin MC, Tsap LV (2007) Objective evaluation of approaches of skin detection using ROC analysis. *Comput Vis Image Underst* 108:41–51

54. Schwartz WR, Gopalan R, Chellappa R, Davis LS (2009) Robust human detection under occlusion by integrating face and person detectors. In: Proceedings international conference on advances in biometrics, pp 970–979
55. Sigal L, Sclaroff S, Athitsos V (2004) Skin color-based video segmentation under time-varying illumination. *IEEE Trans Pattern Anal Mach Intell* 26(7):862–877
56. Sim T, Baker S, Bsat M (2003) The CMU pose, illumination, and expression database. *IEEE Trans Pattern Anal Mach Intell* 25(12):1615–1618
57. Smith AR (1978) Color gamut transform pairs. *SIGGRAPH Comput Graph* 12(3):12–19
58. Sobottka K, Pitas I (1996) Extraction of facial regions and features using color and shape information. In: Proceedings international conference of pattern recognition, pp 421–425
59. Soriano M, Martinkauppi B, Huovinen S, Laaksonen M (2000) Skin detection in video under changing illumination conditions. In: Proceedings international conference on pattern recognition, vol 1:839–842
60. Störring M (2004) Computer vision and human skin colour: a Ph.D. Computer vision and media technology laboratory. Dissertation, Aalborg University
61. Sun HM (2010) Skin detection for single images using dynamic skin color modeling. *Pattern Recognit* 43(4):1413–1420
62. Terrillon JC, David M, Akamatsu S (1998) Automatic detection of human faces in natural scene images by use of a skin color model and of invariant moments. In: Proceedings IEEE international conference on automatic face and gesture recognition, pp 112–117
63. Terrillon JC, Shirazi M, Fukamachi H, Akamatsu S (2000) Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In: Proceedings of the IEEE international conference on automatic face and gesture recognition, pp 54–61
64. Terrillon JC, Pilpre A, Niwa Y, Yamamoto K (2003) Analysis of a large set of color spaces for skin pixel detection in color images. In: International Conference on quality control by artificial vision, vol 5132, pp 433–446
65. Vezhnevets V, Sazonov V, Andreeva A (2003) A survey on pixel-based skin color detection techniques. In: Proceedings of the GRAPHICON-2003, pp 85–92
66. Viola M, Jones MJ, Viola P (2003) Fast multi-view face detection. In: Proceedings of the computer vision and pattern recognition
67. Viola P, Jones MJ (2004) Robust real-time face detection. *Int J Comput Vision* 57(2):137–154
68. Wang X, Xu H, Wang H, Li H (2008) Robust real-time face detection with skin color detection and the modified census transform. In: International conference on information and automation, pp 590–595
69. Wang X, Zhang X, Yao J (2011) Skin color detection under complex background. In: International conference on mechatronic science, electric engineering and computer (MEC), pp 1985–1988
70. Wei Z, Dong Y, Zhao F, Bai H (2012) Face detection based on multi-scale enhanced local texture feature sets. In: IEEE International conference on acoustics, speech and signal processing, pp 953–956
71. van de Weijer J, Gevers T, Gijssenij A (2007) Edge-based color constancy. *IEEE Trans Image Process* 16(9):2207–2214
72. Yang G, Huang TS (1994) Human face detection in a complex background. *Pattern Recognit* 27(1):53–63
73. Yang J, Lu W, Waibel A (1997) Skin-color modeling and adaptation. In: Proceedings of the Asian conference on computer vision-volume II, Springer-Verlag, pp 687–694
74. Yang MH, Ahuja N (1999) Gaussian mixture model for human skin color and its applications in image and video databases. In: Proceedings of the SPIE, pp 458–466
75. Yang MH, Kriegman DJ, Ahuja N (2002) Detecting faces in images: a survey. *IEEE Trans Pattern Anal Mach Intell* 24(1):34–58
76. Yendrikhovskij SN, Blommaert FJJ, de Ridder H (1999) Color reproduction and the naturalness constraint. *Color Res Appl* 24(1):52–67

77. Zarit BD, Super BJ, Quek FKH (1999) Comparison of five color models in skin pixel classification. In: Proceedings of the International workshop on recognition, analysis, and tracking of faces and gestures in real-time systems, pp 58–63
78. Zeng H, Luo R (2012) A new method for skin color enhancement. In: Proceedings of the SPIE, vol 8292. pp 82,920K.1–9
79. Zhang C, Zhang Z (2010) A survey of recent advances in face detection. Tech Rep MSR-TR-2010-66, Microsoft Research
80. Zhang L, Chu R, Xiang S, Liao S, Li S (2007) Face detection based on Multi-Block LBP Representation. In: Advances in biometrics, lecture notes in computer science, vol 4642. Springer-Verlag, pp 11–18
81. Zhao W, Chellappa R, Phillips PJ, Rosenfeld A (2003) Face recognition: a literature survey. *ACM Comput Surv* 35(4):399–458

# Color Saliency Evaluation for Video Game Design

Richard M. Jiang, Ahmed Bouridane and Abbes Amira

**Abstract** This chapter presents the saliency evaluation approach for visual design of video games, where visual saliency is an important factor to evaluate the impact of visual design on user experience of video games. To introduce visual saliency into game design, we carried out an investigation on several state-of-art saliency estimation methods, and studied on three approaches for saliency estimation: color-based, histogram-based, and information theory based methods. In experiments, these approaches were evaluated on a public saliency dataset and compared with the state-of-art technologies, and it was shown that the proposed information theoretic saliency model can attain a better performance in comparison with several state-of-art methods. Then we applied the information theoretic saliency model to visual game design with image and video examples and demonstrated on how to help game designers to evaluate their visual design with respect to the salience awareness of human visual perception systems.

**Keywords** Color saliency · Video game · Visual design · User interaction · Information theory · Computer vision · Human perception

---

R. M. Jiang (✉)  
Department of Automatic Control and System Engineering,  
The University of Sheffield, Sheffield, UK  
e-mail: m.jiang@acm.org

A. Bouridane  
Department of Computing, Engineering and Information Science,  
Northumbria University, Newcastle, UK

A. Amira  
School of Computing, The University of West Scotland, Paisley, UK

# 1 Introduction

## 1.1 *Visual Design of Interactive Video Games*

In the early days of game design, game designers were leading programmers and often the only programmers for a game, and hence spent more time on technology development and paid less attention to user experience. In those old days, game engineering [1–4] was primarily about low-level optimization—writing codes that would run swiftly on the target computer or embedding system, leveraging smart little tricks whenever possible. But in the recent years, game design has ballooned in its design complexity. Nowadays the primary technical task is simply getting the code to work to produce a final product that meets to the desired functionality. On the other side, a game designer is more concerned with user-centered high-level performance evaluation than ever. Hence game design, as a discipline, requires a focus on games in and of themselves, and crosses multi-disciplines involving architect design, 3D graphics, and interactive visualization with a special interest to meet the demands of users as well as the market.

Modern game design often starts with a narrative story [2, 3], fits in with the state-of-art game programming techniques, and matches with user-interactive visualization design. As shown in Fig. 1, on the top level, the game design usually starts from an initial game proposal that documents the concept, game play, feature list, setting and story, target audience, requirements and schedule. It then goes down into five different sections. First, the game designer needs to choose suitable hardware that meets the requirements and avails in the development. Second, the game designer has to figure out what is the best choice of user interface. Some may prefer the old-fashion joystick, and many new developers favor new fancy technologies such as Microsoft's *Kinect* that can track players' pose. Third, there is considerable work on graphic and visual design that reflects the narrative games. Fourth, coding and programming provides the technology platform to accommodate all three aspects together as a system. Finally, there will usually be a user-centered evaluation to appraise on the usability of the game with respect to user experience and predict market response. While game technology is becoming more and more mature, user-centered visual design, on the other end, is yet far from being thoroughly researched.

Visual design in game development usually needs to be considered from two aspects. In the one side, most modern games have narrative elements which define a stylish time-space context to host the imagination of graphic designers, and make games less abstractive and richer in entertainment value. Hence, visual designer needs to highlight these aspects in their designs with appropriate visual impact. On the other side, nowadays game design is user-centered, and visual design has to be adapted to capture the attention of users in an entertaining way. Combining both together, we then have an open question for visual game design: how could a visual design highlight what the game designer wants to emphasize and on the other side, capture user's attention? Naturally, this leads to a well-known topic, visual saliency estimation.

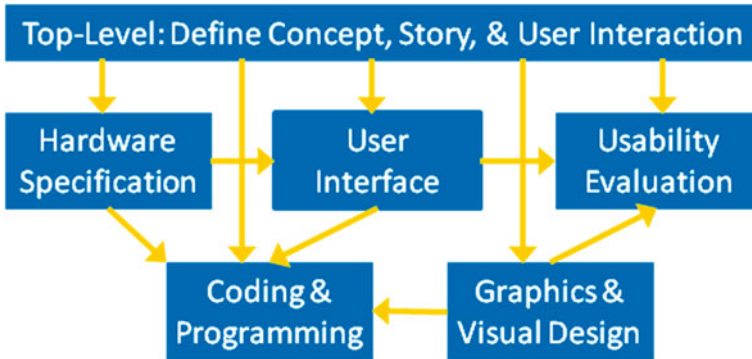


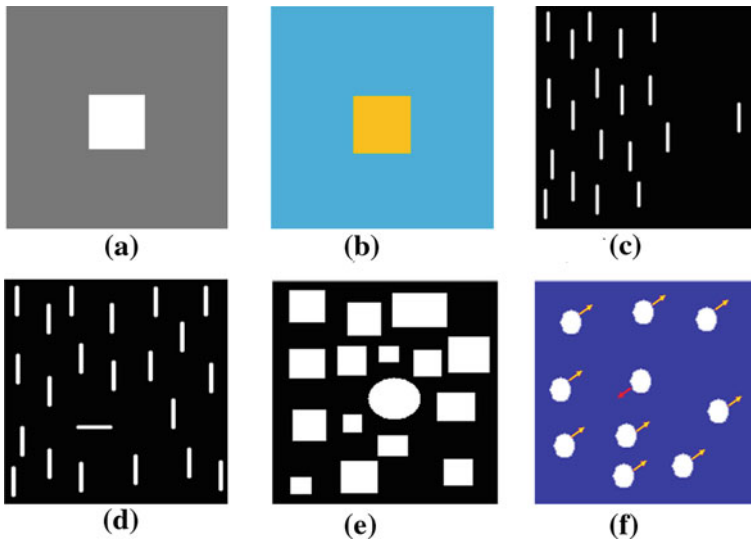
Fig. 1 Modern game design has many aspects to take care of

## 1.2 Visual Saliency

Visual saliency is a multi-disciplinary scientific terminology across biology, neurology and computer vision. It refers to the mechanism about how salient visual stimuli attract human attention. Complex biological systems need to rapidly detect potential prey, predators, or mates in a cluttered visual world. However, it is computationally unaffordable to process all interesting targets in one's visual field simultaneously, making it a formidable task even for the most evolved biological brains [5], let alone for any existing computer or robot. Primates and many other animals have adopted a useful strategy to limit complex vision process to a small area or a few objects at any one time. These small regions are referred as salient regions in this biological visual process [6].

Visual salience is the consequence of an interaction of a visual stimulus with other visual stimuli, arising from fairly low-level and stereotypical computations in the early stages of visual processing. The factors contributing to salience are generally quite comparable from one observer to the next, leading to similar experiences across a range of observers as well as of behavioral conditions. Figure 2 gives several examples about how visual salience is produced from various contrasts among visual stimuli (or pixels) in an image or a pattern. Figure 2a demonstrates intensity-based salience, where the center rectangle has different intensity in contrast with its surrounding. Figure 2b is the case of color-based salience. Figure 2c shows the location of an item may produce salience as well. Figure 2d–e demonstrates other types of salience caused by orientation, shape and motion. In summary, we can see that salience usually refers to how different a pixel or region is in contrast to its surrounding.

Visual salience is a bottom-up, stimulus-driven process that declares a location being so different from its surroundings as to be worthy of your attention [7, 8]. For example, a blue object in a yellow field will be salient and will attract attention in a bottom-up manner. A simple bottom-up framework to emulate how salience may be



**Fig. 2** Visual saliency may arouse from intensity, color, location, orientation, shape, and motion

computed in biological brains has been developed over the past three decades [6–9]. In this framework, incoming visual information is first processed by early visual neurons that are sensitive to the various elementary visual features of the stimulus. The preprocessing procedure is often operated in parallel over the entire visual field and at multiple spatiotemporal scales, producing a number of initial cortical feature maps that embody the amount of a given visual feature at any location in the visual field and highlight locations significantly different from their neighbors. Finally, all feature maps are fused into a single saliency map which represents a pure saliency signal through weighted neural network [8]. Figure 3 illustrates this framework to produce a saliency map from an input image.

The combination of feature maps can be carried out by a winner-take-all scheme [8], where feature maps containing one or a few locations of much stronger response than anywhere else contribute strongly to the final perceptual saliency. The formulations of this basic principle can have slightly different terms, including defining salient locations as those which contain spatial outliers [10], which may be more informative in Shannon’s sense [11].

## 2 Color-Based Visual Saliency Modeling

As it has been discussed in the previous section, visual saliency can be related to motion, orientation, shape, intensity and color. When visual saliency is concerned in video game design, it is mostly related to color images and graphics, where

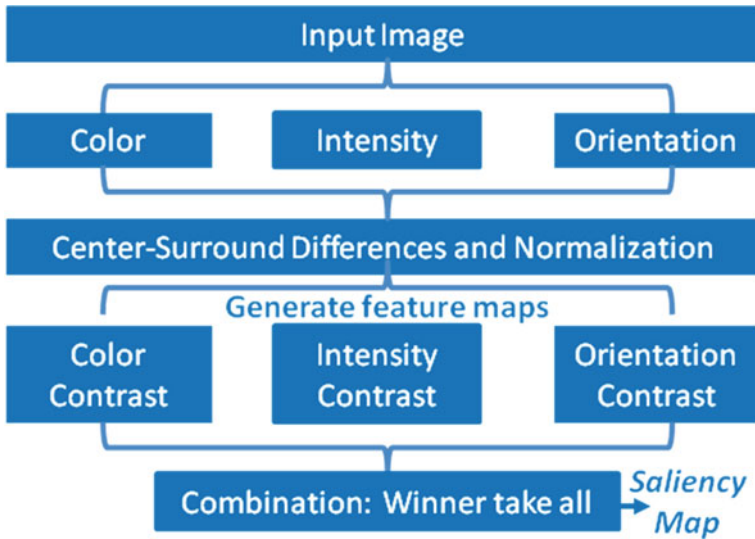


Fig. 3 An early saliency model to emulate human perception system

color-aroused saliency becomes the dominant factor. In this section, we introduce the definition of color saliency and its various models.

### 2.1 Previous Work

From the viewpoint of a graphics designer, it is desired to have a quantifiable model to measure visual saliency, other than to understand its biologic process fully. Actually an accurate modeling of visual saliency has been sought after by computer scientists. Nearly three decades ago, Koch and Ullman [8] proposed a theory to describe the underlying neural mechanisms of vision and bottom-up saliency. They posited that human eyes selects several features that pertain to a stimulus in the visual field and combines these features into a single topographical ‘saliency map’. In the retina, photoreceptors, horizontal, and bipolar cells are the processing elements for edge extraction. Visual input is passed through a series of these cells, and edge information is then delivered to the visual cortex. These systems combine with further processing in the lateral geniculate nucleus that plays a role in detecting shape and pattern information such as symmetry, as a preprocessor for the visual cortex to find a saliency region [12]. Therefore, saliency is a biological response to various stimuli.

Visual saliency has been a multidisciplinary topic for cognitive psychology [6], neurobiology [12], and computer vision [10]. As described in Sect. 5, most early work [13–23] pays more efforts to build their saliency models on low-level image features based on local contrast. These methods investigate the rarity of image regions



with respect to local neighbours. Koch and Ullman [8] presented the highly influential biologically inspired early representation model, and Itti *et al* [9] defined image saliency using central surrounded differences across multi-scale image features. Harel *et al* [14] combined the feature maps of Itti *et al* with other importance maps and highlighted conspicuous parts. Ma and Zhang [15] used an alternative local contrast analysis for saliency estimation. Liu *et al* [16] found multi-scale contrast in a Difference-of-Gaussian (DoG) image pyramid.

Recent efforts have been made toward using global visual contrast, most likely in hierarchical ways. Zhai and Shah [17] defined pixel-level saliency based on a pixel's contrast to all other pixels. Achanta *et al* [18] proposed a frequency tuned method that directly defines pixel saliency using DoG features, and used mean-shift to average the pixel saliency stimuli to the whole regions. More recently, Goferman *et al* [20] considered block-based global contrast while global image context is concerned. Instead of using fixed-size block, Cheng *et al* [21] proposed to use the regions obtained from image segmentation methods and compute the saliency map from the region-based contrast.

## 2.2 Color Saliency

Color saliency [21] refers to the salient stimuli created by color contrast. Principally, it can be modeled by comparing a pixel or a region to its surrounding pixels in color space. In mathematics, the pixel-level color saliency can be formulated by the contrast between a pixel and all other pixels in the global range of an image,

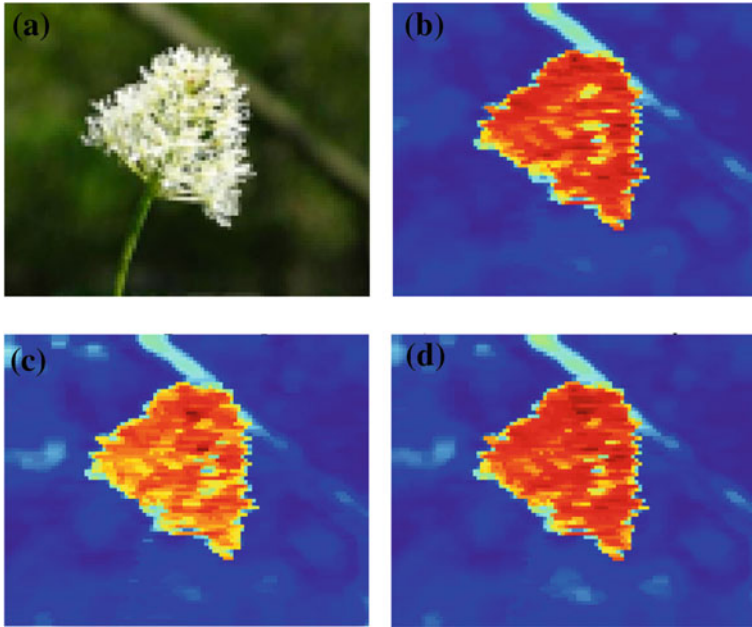
$$S_k = \frac{1}{N} \sum_{i \in \text{img}} d_{k,i} \quad (1)$$

Where typically  $d_{k,i}$  stands for the color distance between the  $i$ -th and  $k$ -th pixels,

$$d_{k,i} = \|c_k - c_i\|, \quad (2)$$

$N$  is the number of pixels in an image, and  $c_k$  and  $c_i$  are the feature vectors of the  $k$ -th and  $i$ -th pixels.

There are several typical color spaces that can be applied to measure the color saliency. *RGB* is the conventional format. Besides, we may have *HSV*, *YUV* and *Lab* color systems. Sometimes they can be combined together to form a multiple channel image data. Figure 4 gives such an example to show how these different color systems produce different color saliency. Figure 4a is the sample image, Fig. 4b shows the estimated saliency from Eq. (1) in *RGB* space, Fig. 4c gives the saliency map in *HSV* space, and Fig. 4d shows the saliency in *RGB+HSV* space. In this example, we can see that different color systems do bring out some differences in their saliency maps.



**Fig. 4** Color saliency estimation. Here, ‘jet’ color map is applied to visualize grey-scale saliency values on pixels (similarly for the rest figures in this chapter)

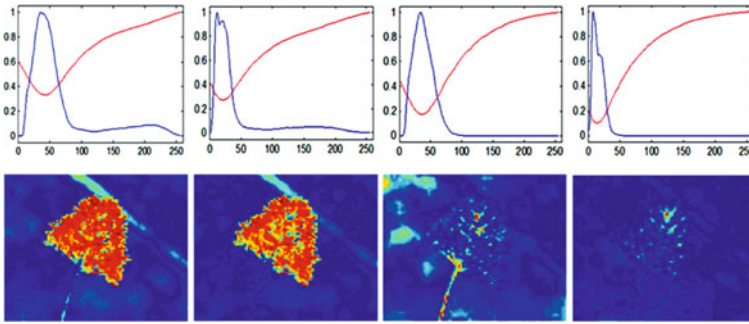
### 2.3 Histogram Based Color Saliency

A major drawback of the above computational model of color saliency is its computing time. While each pixel needs to be compared with all other pixels, its computational complexity is increased to  $O(N^2)$ . It may take long latency time even for a small-size image.

To speed up the computation, a trick using histogram can then be applied. Other than directly computing the color saliency of every pixel, we can compute the saliency of all possible colors first, and use the color of pixel to index its saliency from the pre-computed table. Here, all possible colors in true color space may have  $256 \times 256 \times 256$  candidates. Obviously, that’s even more than the number of pixels. Alternatively, we can compute the color saliency in each channel separately, and then combine them together. Therefore, we have only  $3 \times 256$  bins to compute.

Given that we have the histogram  $h_k$  for a color channel of the image (Here  $h_k$  stands for the number of pixels in the  $k$ -th bin), the color contrast among bins can then be estimated by,

$$S_k = \frac{1}{N - h_k} \sum_{j \in H} \|c_j - c_k\| h_j \tag{3}$$



**Fig. 5** The computation of histogram saliency  $S_k$ . From left to right: red, blue, H and V channels. Upper row: single-channel histogram (blue curves) and its bin-based saliency (red curves); Lower row: single-channel saliency maps

Here,  $c_j$  is the feature value of the  $j$ -th bin,  $h_j$  stands for the pixel number of this bin, and  $N$  is the total number of all pixels in the image.

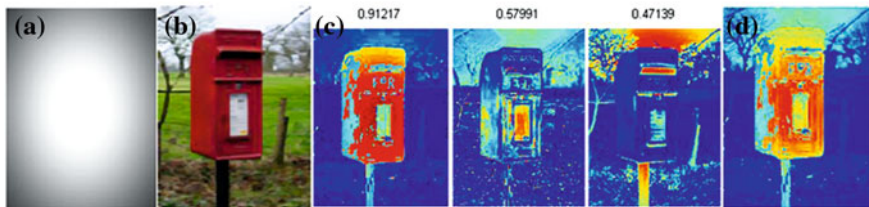
It is noted that the above equation is mathematically the same as Eq. (1). They principally produce the same saliency map of an input image. The only difference is their computing time. While the naive color saliency model in Eq. (1) takes  $O(N^2)$  time, its equivalent representation in Eq. (3) only takes  $O(n^2)$  time. Here  $n$  is the number of histogram bins. While  $n$  is mostly far smaller than  $N$ , the computing complexity could be thousands times reduced by this histogram trick. Our test shows that to process the image in Fig. 4a, using Eq. (1) (on Matlab platform) took about 1.6 seconds, and using the above histogram-based method only took about 0.0085 seconds.

With the above simple scheme, we can easily obtain the initial saliency map for each channel of a color image. Figure 5 shows an example, where the saliency maps were computed for RGB and HSV channels. However, it is obvious these initial estimations are far from accurate. Rather than combining them together linearly, we then proposed a mutual information scheme to refine these initial results by an adaptively weighted fusion procedure, as presented in the following section.

### 2.3.1 Color Saliency Modeling with Information Theory

To attain an accurate and coherent fusion of multiple maps from all color channels, we can here apply mutual information (MI) to weight the initially estimated saliency maps. Basically, we can assume a priori knowledge that human perception always pays attention to the objects around the center of a scene. We can then model this using a centered anisotropic Gaussian distribution,

$$P_r(x) = N(x, r) \quad (4)$$



**Fig. 6** Saliency estimation per channel and the fusion of all results using mutual information. From left to right: **a** Prior saliency map; **b** Original image; **c** Single channel saliency (their MI scores is shown on the top); **d** Final fusion result

Where  $N$  stands for Gaussian distribution,  $x$  is the location of a pixel, and  $r$  is the parameters of the Gaussian model. Figure 6a demonstrates such a priori saliency map. With this expectation, we can then evaluate the initial single-channel saliency maps against this priori map, and then combine them together through their computed weights.

Mutual information (MI) can be considered a statistic method for assessing independence between a pair of variables, and has a well-specified asymptotic distribution. To calculate the MI score between saliency maps  $S_k$  and the priori map  $P_r$ , the following information theoretic formula can be applied:

$$H(S_k, P_r) = - \sum_b p(h_{S_k}, h_{P_r}) \log \{h_{S_k}, h_{P_r}\} \quad (5)$$

Here,  $h_X$  stands for the histogram of the variable  $X$  (for  $S_k$  or  $P_r$ ). Details on MI can be found in the survey by Verdu *et al* [24].

Taking the image in Fig. 6b as an example, we computed its initial saliency maps per color channel using Eq. (3), as shown in Fig. 6c, and compared their single channel maps against the priori map to obtain their MI scores. Their MI scores were labeled on the top of their maps.

Once we have the computed MI score, it becomes simple to fuse the multi-channel saliency maps together, which can be implemented as a weighted total,

$$S_{Total}(x) = \sum_k H(S_k, P_r) S_k(x) \quad (6)$$

where,  $x$  stands for the coordinates of a pixel in its saliency map.

Figure 6d shows the final saliency map. We can see that in comparison to single-channel saliency maps, the fusion result computed by the proposed information theoretic scheme can better capture the salient object in the foreground, with a much higher contrast between the foreground object and its background.

### 3 Evaluation on Color Saliency Estimation

After we introduced our information theoretical saliency estimation method, in this section, we carried out an experimental evaluation on our method.

#### 3.1 Experimental Conditions

To carry out a vigorous evaluation, we use the dataset provided by [15] for experiment. As shown in Fig. 7, this dataset has 1000 images selected from MSRA dataset and is provided with the ground truth in the form of well-segmented salient objects, which is more suitable for our purpose about developing a saliency approach for vision and graphic applications.

Here, a number of saliency methods are chosen for our comparison, including: (1) SR [22]; (2) IT [9]; (3) GB [14]; (4) MZ [15]; (5) LC [17]; (6) FT [18]; (7) CA [20]; (8) HC [21]; and the last, our information theoretic method. The abbreviations were taken from Ref.[21] and Ref.[18] and named after the author's names (such as IT=Itti) or methods (such as HC=Histogram Contrast). It is noted that we mainly care about color saliency for the purpose to evaluate visual game design. Hence, methods toward other saliency factors (such as orientation, location, shape and motion) were not included in our comparison.

We ran our Matlab codes on this large dataset of 1000 images, and compared the results of different methods. When running our Matlab codes on a laptop with 3.0GHz Intel CPU, it is shown that the average computing time per image in the database [18] is around 0.11 seconds. Usually, Matlab implementation can be ten times slower than C/C++ implementation.



Fig. 7 Image dataset used in experiment

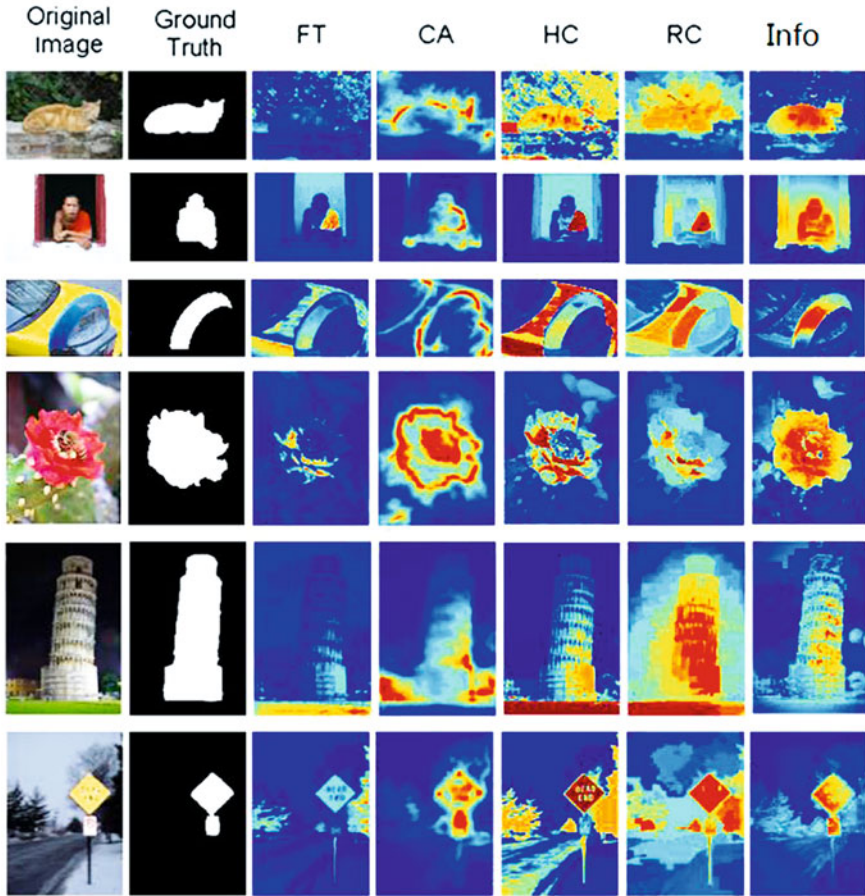


Fig. 8 Visual comparison of various saliency models

### 3.2 Evaluation Results

Figure 8 demonstrates the visual comparison with a number of examples that have been challenging the state-of-art methods. In comparison, we can see that our saliency map can correctly found the salient object in the image and robustly provided a higher saliency contrast ratio between the salient regions and the background.

Figure 9a shows the typical statistic results of precision-recall curves. The curves were computed in the same way as reported by [18, 21]. Here, precision refers to the percentage of salient pixels correctly assigned, while recall corresponds to the fraction of detected salient pixels in relation to the ground truth number of salient pixels. The parameters for all images were set the same in our evaluation.

As shown in Fig. 9a, our information theoretic method has steadily attained the best performance among all compared methods, in term of either precision or recall

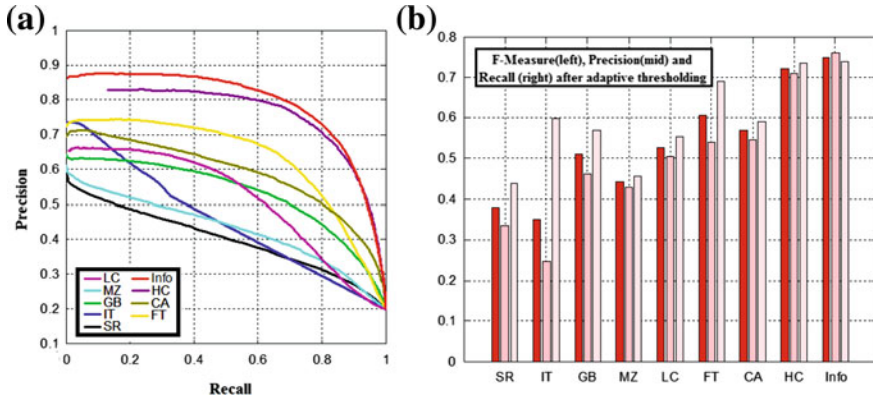


Fig. 9 Statistic evaluation and comparison

rates. With this validation, it can be seen that our method using information theory produced better saliency maps consistent with human visual system.

Other than precision and recall, we can further use  $F$ -measure to evaluate the balanced performance combining recall and precision together. Here, we use the adaptive threshold similarly as proposed by [18], defined as twice the mean saliency of the image:

$$T_{th} = 2 \times mean (s (i, j)) \tag{7}$$

Applied the threshold  $T_{th}$  to saliency map, we can then have the precision and recall of its binary cut, and  $F$ -measure can be subsequently computed by,

$$F = \frac{(1 + \beta^2) \times Precision \times Recall}{(Recall + \beta^2 Precision)} \tag{8}$$

Here,  $\beta^2$  is usually set to 1 for  $F$  measure. Figure 9b shows the  $F$ -measure results. It is clearly shown that our approach attained the best results in term of  $F$ -measure as well as the highest recall and precision. With this initial evaluation, we can then confidently move forward to leverage our method for visual game design.

### 4 Using Color Saliency in Game Design

Today’s game design has become a user-centered task. The designers usually make more efforts to adapt their game for user experience with less difficulty in implementing their design with sophisticated programming skills. While visual design is a primary element to guarantee the users to understand and enjoy the game, game designers usually want to know if their visual design can capture user attention in a proper way. In this section, we proposed the idea of using our saliency model to evaluate visual designs in video games, and demonstrated with image and video examples.

### 4.1 Saliency-Aware Color Design

In a visual game design, a primary question is how to choose colors for the design. As we can see from previous sections, different colors usually create different visual impact to users. A bright color can make most games easier, and a lower contrast can camouflage the targets in the game and make it harder for game players. Hence, a proper choice of colors can justify the sense of a visual design.

Figure 10 demonstrates the usage of saliency estimation for color evaluation. Figure 10a shows a squirrel running in a cluttered scene, and game players needs to shoot the target to get their scores. In this context, visual saliency is a sensitive factor to affect the performance of players. As it is shown in Fig. 10a, from its estimated saliency map we can see the design has properly highlighted the squirrel with a higher saliency value than its background, making it easier for a game player to see their prey in the game.

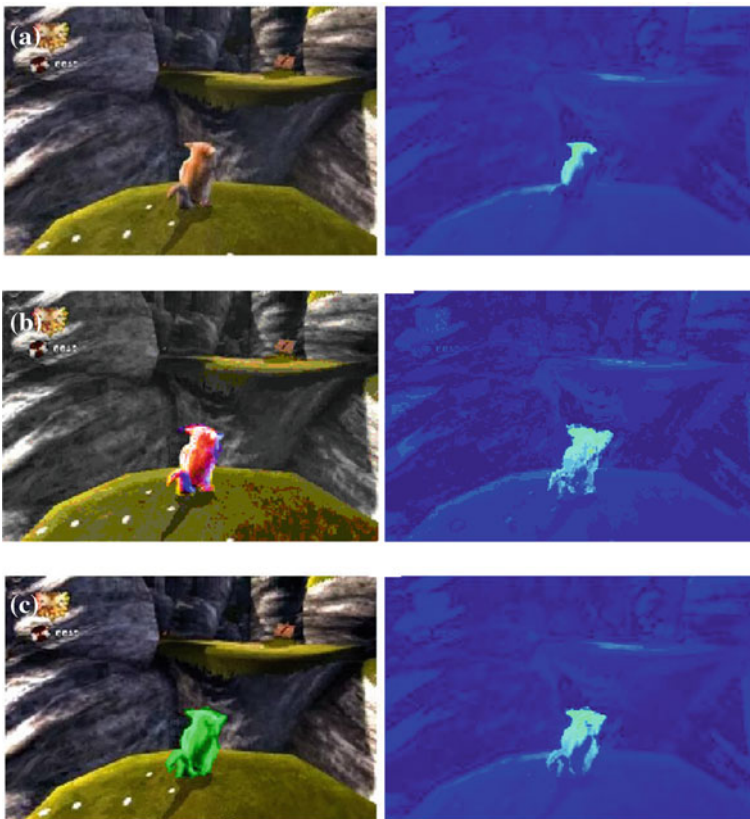


Fig. 10 Saliency-aware color selection in game design



Figure 10b and 10c show that if we change the color, the saliency map could be altered as well. With a bright red or green, the target becomes much easier to be traced, making the game too easier for players to pass.

From this example, we can see that saliency evaluation is a useful procedure for game design. It can easily allow visual designers to immediately know if their color design fits well with human perception system, rather than carrying out a time-consuming user study after every details of visual design are fixed.

### 4.2 Saliency-Aware Illumination

For 3D game design, usually illumination can be an important factor to produce different visual impact. It may drastically change the colors in 3D graphics rendering process and produce a very different contrast between the foreground objects and their background.

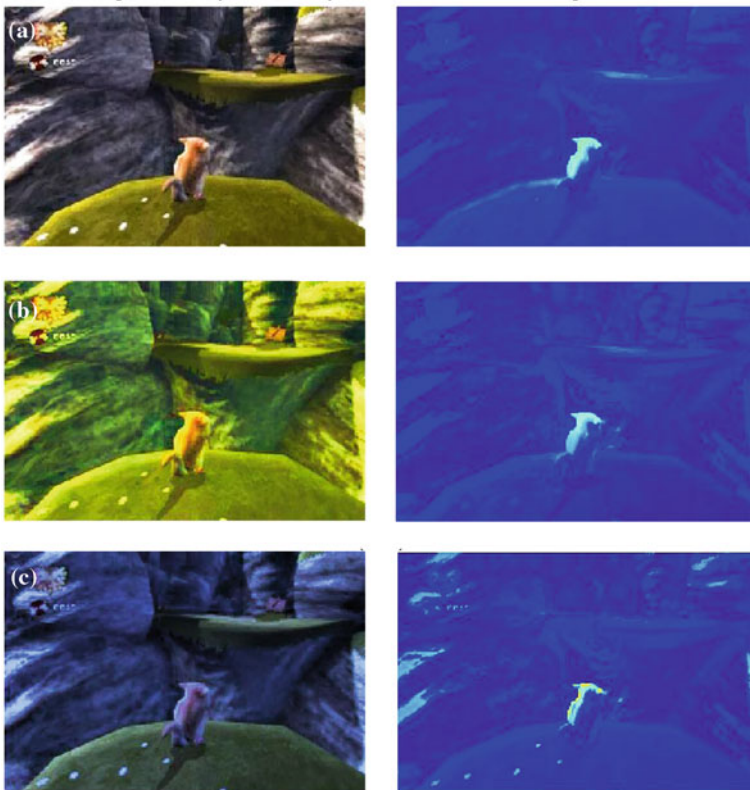


Fig. 11 Saliency evaluation on different illumination

Figure 11 shows such an example. Figure 11a has the original illumination (the same as Fig. 10a). In Fig. 11b, a brighter illumination was applied, and we can see it enlightened the background and hence reduced the contrast slightly. Consequently, the foreground target (the squirrel) becomes less salient in comparison with its background.

In Figure 11c, the illumination was darkened and hence both the foreground and the background became dark as well. Consequently, the squirrel became less outstanding to its background. It also made other brighter regions more salient, distracting the player’s attention from the target.

### 4.3 Visual Saliency in Video Sequence

Usually the evaluation of a video game needs to be put in its spatiotemporal context, and saliency evaluation needs to be carried out on the whole video sequence of a game.

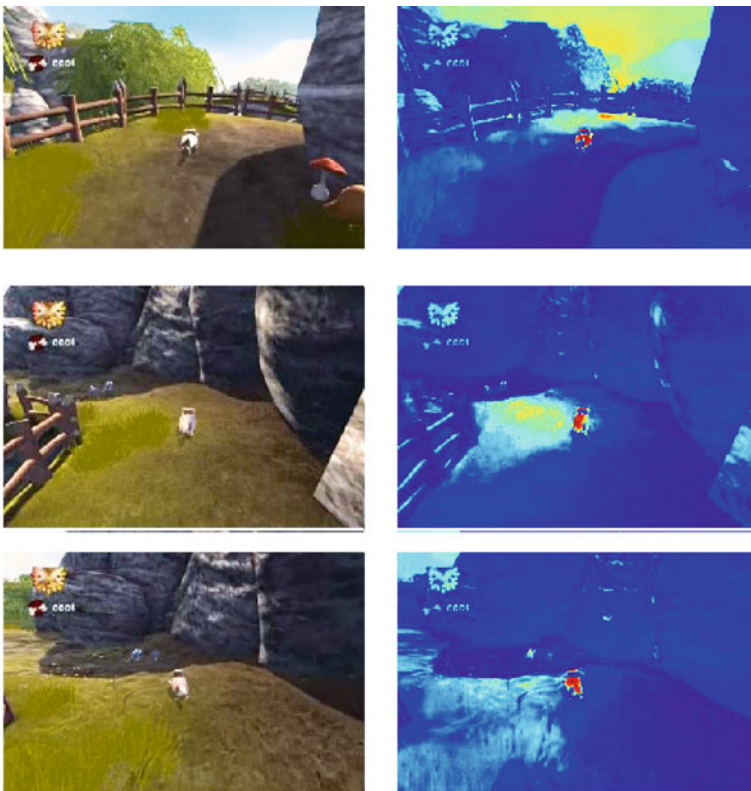


Fig. 12 Saliency evaluation on video sequence

Figure 12 demonstrates the evaluation of a video game design. In the game, the squirrel runs along a path along the mountain and the hunter tries to shoot it down from behind. From this evaluation, we can see the designed visual context always highlighted the running squirrels, which is good for a player to hunt easily. With this result, it is expected that game designer can build a confidence on what they are going to deliver to their customers.

## 5 Conclusion

In conclusion, we proposed to use saliency models to evaluate visual design quality of video games. A robust color saliency model has been developed and vigorously tested on a publically available salient object dataset. The results validated that our information theoretic method consistently outperformed several state-of-art color saliency models. Based on this validation, we further applied our saliency estimation method to visual game evaluation.

There are many ways to use visual saliency to guide the visual game design. In this chapter, we demonstrated several examples, including color selection, illumination evaluation, and video sequence evaluation. From these demos, we can see how a saliency model can help game designers in their modern user-centered game design. A benefit here is, with a handy saliency estimation model, the visual designers can immediately know the impact of their visual design on users during the design procedure, rather than waiting until the last user study stage after the visual design of all details has been finished and it is too late to go back to fix problems in visual design.

## References

1. Blow J (2004) Game development: harder than you think. *Queue—Game Development* 1(10):28
2. Adams E (1999) Three problems for interactive Storytellers. Designer's notebook column, Gamasutra. Springer, Berlin
3. Costikyan G (2000) Where stories end and games begin. *Game developer*, pp. 44–53 September 2000
4. Juul J (1998) A clash between games and narrative. In: *Proceedings of the digital arts and culture conference*, Bergen
5. Tsotsos JK (1991) Is complexity theory appropriate for analysing biological systems? *Behav Brain Sci* 14(4):770–773
6. Treisman A, Gelade G (1980) A feature integration theory of attention. *Cogn Psychol* 12:97–136
7. Desimone R, Duncan J (1995) Neural mechanisms of selective visual attention. *Annu Rev Neurosci* 18(1):193–222
8. Koch C, Ullman S (1985) Shifts in selective visual attention: towards the underlying neural circuitry. *Hum Neurobiol* 4:219–227
9. Itti L, Koch C, Niebur E (1998) A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans Pattern Anal Mach Intell* 20(11):1254–1259

10. Rosenholtz R (1999) A simple saliency model predicts a number of motion popout phenomena. *Vision Res* 39:3157–3163
11. Bruce N, Tsotsos JK (December 2005) Saliency based on information maximization. *Adv Neural Inf Process Syst* 18:155–162
12. Sillito AM, Grieve KL, Jones HE, Cudeiro J, Davis J (1995) Visual cortical mechanisms detecting focal orientation discontinuities. *Nature* 378:492–496
13. Rutishauser A et al (2004) Is bottom-up attention useful for object recognition? In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 37–44 June 2004
14. Harel J, Koch C, Perona P (December 2006) Graph-based visual saliency. *Adv Neural Inf Process Syst* 19:545–552
15. Ma Y.F, Zhang H.-J (2003) Contrast-based image attention analysis by using fuzzy growing. In: *Proceedings of the ACM multimedia 2003 conference*, pp. 374–381 October 2003
16. Liu T, Yuan Z, Sun J, Wang J, Zheng N (2011) Learning to detect a salient object. *IEEE Trans Pattern Anal Mach Intell* 33(2):353–367
17. Zhai Y, Shah M (2006) Visual attention detection in video sequences using spatiotemporal cues. In: *Proceedings of the ACM multimedia 2006 conference*, pp 815–824 October 2006
18. Achanta A et al (2009) Frequency-tuned salient region detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1597–1604 June 2009
19. Judd K et al (2009) Learning to predict where humans look. In: *Proceedings of the international conference on computer vision*, pp. 2106–2113 September 2009
20. Goferman S, Zelnik-Manor L, Tal A (2010) Context-aware saliency detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1915–1926 June 2010
21. Cheng MM et al (2011) Global contrast based salient region detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 409–416 June 2011
22. Hou X, Zhang L (2007) Saliency detection: a spectral residual approach. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–8 June 2007
23. Perazzi F et al (2012) Saliency filters: contrast based filtering for salient region detection. In: *Proceedings of the conference on computer vision and pattern recognition*, pp. 733–740 June 2012
24. Verdu S, McLaughlin SW (1999) *Information theory: 50 years of discovery*. IEEE, New York (in Press)

# Erratum to: On the von Kries Model: Estimation, Dependence on Light and Device, and Applications

Michela Lecca

**Erratum to:**  
**Chapter “On the von Kries Model: Estimation, Dependence on Light and Device, and Applications”**  
**in: M. E. Celebi and B. Smolka (eds.), *Advances in Low-Level Color Image Processing*,**  
**DOI [10.1007/978-94-007-7584-8\\_4](https://doi.org/10.1007/978-94-007-7584-8_4)**

The ‘old Table 9’ should be replaced with ‘new Table 9’:

**Table 9** UEA Dataset—AMP across a change of device and illuminant when the von Kries model is used. The images acquired by a camera  $c$  under a fixed illuminant  $\sigma$  (see the first column) are used as references, and matched with the images acquired by the other cameras and illuminants (see the second, third and fourth columns). “Cam.” stands for “Camera”

Camera and Illuminant	Camera and Illuminant		
(a) <i>Cam. &amp; Ill.</i>	( <i>Cam. 1, Ill A</i> )	( <i>Cam. 1, Ill D65</i> )	( <i>Cam. 1, Ill TL84</i> )
(Cam. 1, Ill A)	1.0000	0.9868	0.9960
(Cam. 1, Ill D65)	0.9723	1.0000	0.9974
(Cam. 1, Ill TL84)	0.9947	0.9947	1.0000
(Cam. 2, Ill A)	0.9947	0.9735	0.9868
(Cam. 2, Ill D65)	0.9458	1.0000	0.9960
(Cam. 2, Ill TL84)	0.9723	0.9788	0.9987

(continued)

The online version of the original chapter can be found under DOI [10.1007/978-94-007-7584-8\\_4](https://doi.org/10.1007/978-94-007-7584-8_4)

M. Lecca (✉)

Fondazione Bruno Kessler, via Sommarive 18, 38123 Trento, Italy  
e-mail: lecca@fbk.eu

**Table 9** (continued)

Camera and Illuminant	Camera and Illuminant		
(Cam. 3, Ill A)	0.9206	0.9497	0.9286
(Cam. 3, Ill D65)	0.9101	0.9577	0.9445
(Cam. 3, Ill TL84)	0.9008	0.9445	0.9312
(Cam. 4, Ill A)	0.9788	0.9431	0.9484
(Cam. 4, Ill D65)	0.9894	0.9934	0.9828
(Cam. 4, Ill TL84)	0.9775	0.9683	0.9709
<i>(b) Cam. &amp; Ill.</i>	<i>(Cam. 2, Ill A)</i>	<i>(Cam. 2, Ill D65)</i>	<i>(Cam. 2, Ill TL84)</i>
(Cam. 1, Ill A)	0.9818	0.9431	0.9616
(Cam. 1, Ill D65)	0.9577	0.9987	0.9854
(Cam. 1, Ill TL84)	0.9828	0.9934	1.0000
(Cam. 2, Ill A)	1.0000	0.9286	0.9524
(Cam. 2, Ill D65)	0.9101	1.0000	1.0000
(Cam. 2, Ill TL84)	0.9193	1.0000	1.0000
(Cam. 3, Ill A)	0.8297	0.8796	0.8664
(Cam. 3, Ill D65)	0.9325	0.9537	0.9471
(Cam. 3, Ill TL84)	0.9352	0.9550	0.9616
(Cam. 4, Ill A)	0.9140	0.8929	0.8915
(Cam. 4, Ill D65)	0.9828	0.9921	0.9696
(Cam. 4, Ill TL84)	0.9974	0.9921	0.9894
<i>(c) Cam. &amp; Ill.</i>	<i>(Cam. 3, Ill A)</i>	<i>(Cam. 3, Ill D65)</i>	<i>(Cam. 3, Ill TL84)</i>
(Cam. 1, Ill A)	0.8320	0.8651	0.8505
(Cam. 1, Ill D65)	0.8214	0.8942	0.8651
(Cam. 1, Ill TL84)	0.7500	0.8334	0.8399
(Cam. 2, Ill A)	0.8003	0.8638	0.8783
(Cam. 2, Ill D65)	0.8095	0.8612	0.8704
(Cam. 2, Ill TL84)	0.7593	0.8241	0.8598
(Cam. 3, Ill A)	1.0000	0.9960	0.9974
(Cam. 3, Ill D65)	0.9987	1.0000	0.9987
(Cam. 3, Ill TL84)	1.0000	1.0000	1.0000
(Cam. 4, Ill A)	0.8796	0.8704	0.8730
(Cam. 4, Ill D65)	0.8928	0.8769	0.8796
(Cam. 4, Ill TL84)	0.8505	0.8558	0.8743
<i>(d) Cam. &amp; Ill.</i>	<i>(Cam. 4, Ill A)</i>	<i>(Cam. 4, Ill D65)</i>	<i>(Cam. 4, Ill TL84)</i>
(Cam. 1, Ill A)	0.9696	0.9828	0.9934
(Cam. 1, Ill D65)	0.9259	0.9788	0.9894
(Cam. 1, Ill TL84)	0.9272	0.9643	0.9881
(Cam. 2, Ill A)	0.8690	0.9921	0.9974
(Cam. 2, Ill D65)	0.8638	0.9921	0.9947
(Cam. 2, Ill TL84)	0.8466	0.9828	0.9960
(Cam. 3, Ill A)	0.9259	0.9683	0.9445
(Cam. 3, Ill D65)	0.9537	0.9775	0.9749
(Cam. 3, Ill TL84)	0.9577	0.9709	0.9723
(Cam. 4, Ill A)	1.0000	0.9894	0.9934
(Cam. 4, Ill D65)	0.9868	1.0000	0.9974
(Cam. 4, Ill TL84)	0.9974	0.9987	1.0000

# Erratum to: Skin Detection and Segmentation in Color Images

Michał Kawulok, Jakub Nalepa and Jolanta Kawulok

**Erratum to:**  
**Chapter “Skin Detection and Segmentation in Color Images” in: M. E. Celebi and B. Smolka (eds.), *Advances in Low-Level Color Image Processing*, DOI [10.1007/978-94-007-7584-8\\_11](https://doi.org/10.1007/978-94-007-7584-8_11)**

Caption of Figs. 5 and 6 should read as below:

---

The online version of the original chapter can be found under DOI [10.1007/978-94-007-7584-8\\_11](https://doi.org/10.1007/978-94-007-7584-8_11)

---

M. Kawulok (✉) · J. Nalepa · J. Kawulok  
Institute of Informatics, Silesian University of Technology, Gliwice, Poland  
e-mail: [Michal.Kawulok@polsl.pl](mailto:Michal.Kawulok@polsl.pl)

J. Nalepa  
e-mail: [Jakub.Nalepa@polsl.pl](mailto:Jakub.Nalepa@polsl.pl)

J. Kawulok  
e-mail: [Jolanta.Kawulok@polsl.pl](mailto:Jolanta.Kawulok@polsl.pl)

M. E. Celebi and B. Smolka (eds.), *Advances in Low-Level Color Image Processing*, Lecture Notes in Computational Vision and Biomechanics 11,  
DOI: [10.1007/978-94-007-7584-8\\_14](https://doi.org/10.1007/978-94-007-7584-8_14), © Springer Science+Business Media Dordrecht 2014

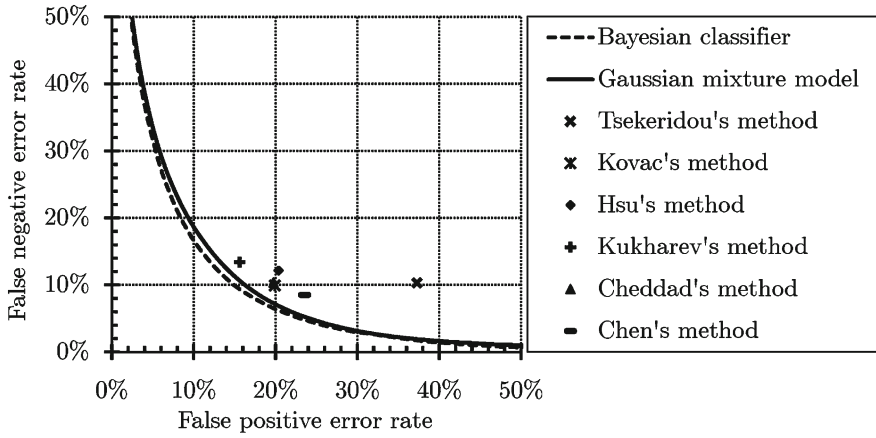


Fig. 5 ROC curves obtained for the Bayesian classifier and the Gaussian mixture model, and errors obtained for the rule-based skin detectors

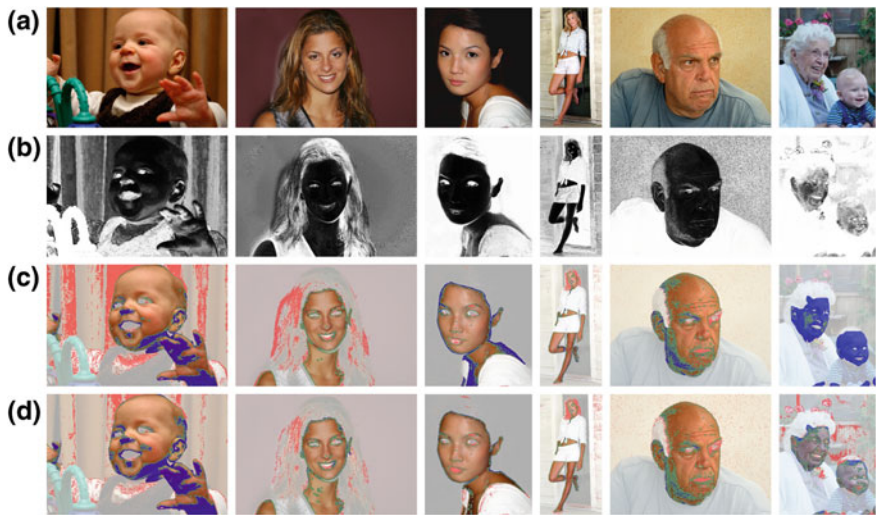


Fig. 6 Skin detection results obtained using Bayesian classifier: original image (a), skin probability map (b), segmentation using a threshold optimized for the whole ECU-V data set (c) and using the best threshold determined for each particular image (d)



Caption of Figs. 15 and 16 should read as below:

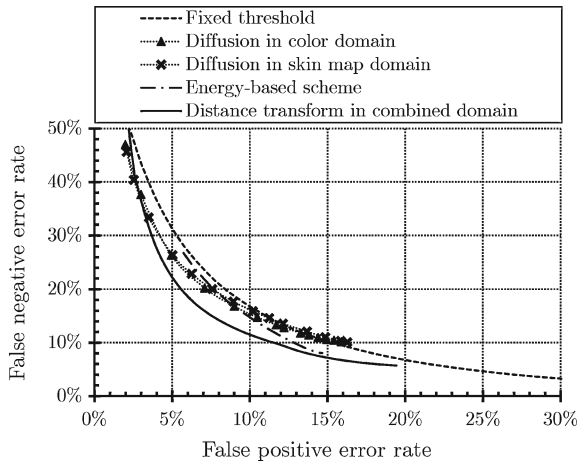


Fig. 15 ROC curves obtained using spatial analysis methods

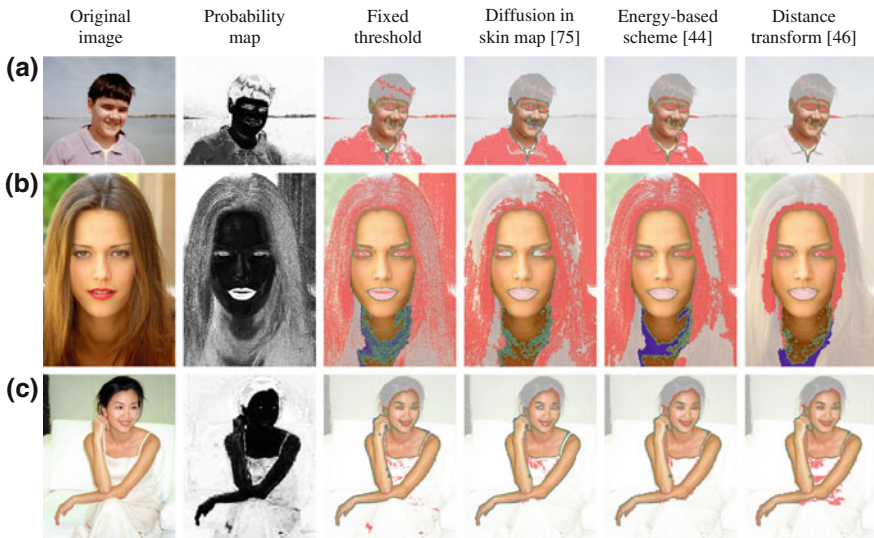


Fig. 16 Examples of skin detection results obtained using different spatial analysis methods