UNIVERSIDAD POLITÉCNICA DE CATALUÑA

Programa de Doctorado:
AUTOMATIZACIÓN AVANZADA Y ROBÓTICA

Tesis Doctoral

ROBUST PERCEPTUAL ORGANIZATION TECHNIQUES FOR ANALYSIS OF COLOR IMAGES

Rodrigo Moreno Serrano

Directores:
Dr. Miguel Ángel García García
Dr. Domènec Puig Valls

Instituto de Organización y Control de Sistemas Industriales

2010

Abstract

This thesis focuses on the development of new robust image analysis techniques more closely related to the way the human visual system behaves. One of the pillars of the thesis is the so-called tensor voting technique. This is a robust perceptual organization technique that propagates and aggregates information encoded by means of tensors through a convolution-like process. Its robustness and adaptability have been one of the key points for using tensor voting in this thesis. These two properties are verified in the thesis by applying tensor voting to three applications where it had not been applied so far: image structure estimation, edge detection and image segmentation of images acquired through stereo vision.

The most important drawback of tensor voting is that its usual implementations are highly time consuming. In this line, this thesis proposes two new efficient implementations of tensor voting, both derived from an in-depth analysis of this technique.

Despite its adaptability, this thesis shows that the original formulation of tensor voting (hereafter, classical tensor voting) is not adequate for some applications, since the hypotheses from which it is based are not suitable for all applications. This is particularly certain for color image denoising. Thus, this thesis shows that, more than a method, tensor voting can be thought of as a methodology in which the encoding and voting process can be tailored for every specific application, while maintaining the tensor voting spirit.

By following this reasoning, this thesis proposes a unified framework for both image denoising and robust edge detection. This framework is an extension of the classical tensor voting in which both color and edginess—the likelihood of finding an edge at every pixel of the image—are encoded through tensors, and where the voting process takes into account a set of plausible perceptual criteria related to the way the human visual system processes visual information. Recent advances in the perception of color have been essential for designing such a voting process.

This new approach has been found effective, since it yields excellent results for both applications. In particular, the new method applied to image denoising has a better performance than other state-of-the-art methods for real noise. This makes it more adequate for real applications, in which an image denoiser is indeed required. In addition, the method applied to edge detection

yields more robust results than the state-of-the-art techniques and has a competitive performance in recall, discriminability, precision, and false alarm rejection.

Moreover, this thesis shows how the results of this new framework can be combined with other techniques to tackle the problem of robust color image segmentation. The tensors obtained by applying the new framework are utilized to classify pixels into likely-homogeneous and likely-inhomogeneous. Those pixels are then sequentially segmented through a variation of an efficient graph-based image segmentation algorithm. Experiments show that the proposed segmentation algorithm yields better scores in three of the five applied evaluation measures when compared to the state-of-the-art techniques with a competitive computational cost.

This thesis also proposes new evaluation techniques in the scope of image processing. First, two new measures are proposed in the field of image denoising: one to measure how an algorithm is able to preserve edges, and the second to measure how a method is able not to introduce undesirable artifacts. Second, a new methodology for assessing edge detectors that avoids possible bias introduced by post-processing is proposed. It consists of five new measures for assessing recall, discriminability, precision, false alarm rejection and robustness. Finally, two new non-parametric measures are proposed for estimating the degree of over- and undersegmentation yielded by image segmentation algorithms.

Keywords: perceptual techniques, robust methods, color image analysis, tensor-based processing techniques, tensor voting, structure estimation, color image denoising, edge detection, image segmentation, evaluation in image processing.

Resumen

Esta tesis aborda el desarrollo de nuevas técnicas de análisis robusto de imágenes estrechamente relacionadas con el comportamiento del sistema visual humano. Uno de los pilares de la tesis es la votación tensorial, una técnica robusta que propaga y agrega información codificada en tensores mediante un proceso similar a la convolución. Su robustez y adaptabilidad han sido claves para su uso en esta tesis. Ambas propiedades han sido verificadas en tres nuevas aplicaciones de la votación tensorial: estimación de estructura, detección de bordes y segmentación de imágenes adquiridas mediante estereovisión.

El mayor problema de la votación tensorial es su elevado coste computacional. En esta línea, esta tesis propone dos nuevas implementaciones eficientes de la votación tensorial derivadas de un análisis en profundidad de esta técnica.

A pesar de su capacidad de adaptación, esta tesis muestra que la formulación original de la votación tensorial (a partir de aquí, votación tensorial clásica) no es adecuada para algunas aplicaciones, dado que las hipótesis en las que se basa no se ajustan a todas ellas. Esto ocurre particularmente en el filtrado de imágenes en color. Así, esta tesis muestra que, más que un método, la votación tensorial es una metodología en la que la codificación y el proceso de votación pueden ser adaptados específicamente para cada aplicación, manteniendo el espíritu de la votación tensorial.

En esta línea, esta tesis propone un marco unificado en el que se realiza a la vez el filtrado de imágenes y la detección robusta de bordes. Este marco de trabajo es una extensión de la votación tensorial clásica en la que el color y la probabilidad de encontrar un borde en cada píxel se codifican mediante tensores, y en el que el proceso de votación se basa en un conjunto de criterios perceptuales relacionados con el modo en que el sistema visual humano procesa información. Los avances recientes en la percepción del color han sido esenciales en el diseño de dicho proceso de votación.

Este nuevo enfoque ha sido efectivo, obteniendo excelentes resultados en ambas aplicaciones. En concreto, el nuevo método aplicado al filtrado de imágenes tiene un mejor rendimiento que los métodos del estado del arte para ruido real. Esto lo hace más adecuado para aplicaciones

reales, donde los algoritmos de filtrado son imprescindibles. Además, el método aplicado a detección de bordes produce resultados más robustos que las técnicas del estado del arte y tiene un rendimiento competitivo con relación a la completitud, discriminabilidad, precisión y rechazo de falsas alarmas.

Además, esta tesis demuestra que este nuevo marco de trabajo puede combinarse con otras técnicas para resolver el problema de segmentación robusta de imágenes. Los tensores obtenidos mediante el nuevo método se utilizan para clasificar píxeles como probablemente homogéneos o no homogéneos. Ambos tipos de píxeles se segmentan a continuación por medio de una variante de un algoritmo eficiente de segmentación de imágenes basada en grafos. Los experimentos muestran que el algoritmo propuesto obtiene mejores resultados en tres de las cinco medidas de evaluación aplicadas en comparación con las técnicas del estado del arte, con un coste computacional competitivo.

La tesis también propone nuevas técnicas de evaluación en el ámbito del procesamiento de imágenes. En concreto, se proponen dos medidas de filtrado de imágenes con el fin de evaluar el grado en que un método es capaz de preservar los bordes y evitar la introducción de defectos. Asimismo, se propone una nueva metodología para la evaluación de detectores de bordes que evita posibles sesgos introducidos por el post-procesado. Esta metodología se basa en cinco medidas para estimar completitud, discriminabilidad, precisión, rechazo de falsas alarmas y robustez. Por último, se proponen dos nuevas medidas no paramétricas para estimar el grado de sobre e infrasegmentación producido por los algoritmos de segmentación de imágenes.

Palabras clave: técnicas perceptuales, métodos robustos, análisis de imágenes de color, técnicas de procesamiento basadas en tensores, votación tensorial, estimación de estructura, filtrado de imágenes de color, detección de bordes, segmentación de imágenes, evaluación en procesamiento de imágenes.

Acknowledgments

First and foremost, I would like to thank my advisors, Prof. Miguel Ángel García, and Prof. Domènec Puig for their supervision in the Intelligent Robotics and Computer Vision Group (IRCV) at Rovira i Virgili University, Tarragona, Spain. This thesis would not have been possible without our frequent and fruitful discussions.

Second, I would like to thank Prof. Joachim Weickert, who gave me the opportunity to work in the Mathematical Image Analysis Group (MIA), which he heads, during my stay at Saarland University, Saarbrücken, Germany in 2009. I would also like to thank Prof. Bernhard Burgeth for his invaluable collaboration during my stay at Saarbrücken.

Third, I would like to thank all members of the IRCV and MIA groups for their invaluable collaboration. In particular, to Jaime, Ling, Tomás, Carme, Said, Julián, Gerard, Xavier, Diana, Hatem, and, on top of them, to Marcela, from the IRCV; and Luis, Pascal, Marcus, Markus, Sven, Kai, Christian, Henning, Karteek, and Prof. Bruhn, Breuß and Welk from the MIA.

In addition, this thesis would not have reached completion without the financial support from various institutions. Primarily, from the Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR) through the scholarships FI-0513-2005 and 2009 BE1 00436. These scholarships have been partially supported by the Commissioner for Universities and Research of the Department of Innovation, Universities and Companies of the Catalonian Government and by the European Social Fund. In addition, I have received financial support from the Department of Computer Science and Mathematics at Rovira i Virgili University and from the projects DPI2004-07993-C03-03 and DPI2007-66556-C03-03 of the Spanish Ministry of Science and Technology.

Last but not least, I want to thank all those anonymous reviewers who have made worthy suggestions on my conference and journal publications.

Contents

List of Figures

List of Tables

Notation

$\mathbb{T}$	A tensor variable or a matrix
$\mathbf{v}$	A vector variable
F	A scalar variable
λ_i	Eigenvalue of a tensor
$\mathbf{e}_i$	Eigenvector of a tensor
s_i	Saliency measurement
$\hat{s}_i$	Normalized saliency measurement
ΔE	Measurement of perceptual color difference
∇u	Gradient of an image
G_σ	Decaying Gaussian function with zero mean and a standard deviation σ
$neigh(\mathbf{p})$	Set of the neighboring points of $\mathbf{p}$
$\mathscr{C}$	A segmentation of an image
$\mathbb{R}$	A region in a segmentation

INTRODUCTION

Chapter 1

Introduction

Image analysis aims at extracting information from images through computers. Image analysis includes a variety of applications such as object recognition, image segmentation, feature extraction, edge detection and image denoising. These applications have been found useful in fields such as security, medicine, robotics, photography, forensics, among many others. Although a huge amount of methods have been proposed in image analysis so far, the proposal of robust techniques is still challenging, since the performance of most of the methods is dramatically reduced under noisy conditions.

During the last years, many perception-based methods have yielded outstanding results in robust image analysis. These techniques aim at mimicking the way the human visual system processes visual stimuli. The success of these techniques relies on two facts: a) visual perception in humans is barely affected by noise, and b) many applications need methods for yielding results more appealing to humans. Despite its success, further research in perception-based methods is still required due to the inherent difficulty to devise perception-based techniques that take into account as many perceptual mechanisms as possible, while keeping efficiency under control, which is still an issue for most of these techniques. In addition, most of the visual perception processes are still not well understood and they are under active research in psychology and neural cognition.

A particular family of perception-based methods are the so-called perceptual organization techniques, which are mainly based on principles of the Gestalt psychology. They have become popular since they have yielded state-of-the-art results (cf. Chapter 2.3). However, they have mainly been restricted to the scope of the contour completion problem, which aims at inferring

object boundaries from contour fragments (Gillam, 2001). Among these techniques, tensor voting (cf. Section 2.6) appears as the most flexible one, since researchers have successfully been able to adapt it with excellent results to problems well beyond the ones which it was originally applied to. However, classical tensor voting has also shown a poor performance in important tasks such as color image segmentation (Kang and Medioni, 2001).

Furthermore, color scientists have proposed powerful tools to perceptually process color information (cf. Chapter 2.2). On the one hand, they have proposed perceptual representations of color inspired by both the physiology and visual perception of colors. On the other hand, perceptual color difference formulas have stemmed from psychophysical experiments in order to quantitatively measure the visual perception of color. Despite its limitations, these tools have been successfully used in many applications related to color management and represent the state-of-the-art in this field.

This dissertation aims at developing new robust image analysis techniques more closely related to the way the human visual system behaves. In particular, robust techniques for structure estimation, image denoising, edge detection and image segmentation are presented. These techniques are based on both tensor voting and tools for perceptual processing of color. Since classical tensor voting is only adequate for some image analysis applications (cf. Chapter 4), extended versions of tensor voting are presented, which are more suitable for the applications discussed in Part II. The major novelty of the proposed methods relies on the extension of a perceptual organization method to different contexts and on the integration of perceptual organization with tools for perceptual processing of color.

1.1 Objectives

1.1.1 General Objective

The main objective of this dissertation is to propose new perception-based techniques for robust color image analysis and processing. In particular, perceptual organization methods and perceptual tools for color processing are the foundations of the developments presented in this thesis. Two underlined principles drive the achievement of the above objectives: perceptual organization methods, in particular tensor voting, which has been used for the fulfillment of every specific objective, as well as psychological theories and psychophysical evidence about how hu-

mans perceive color and noise.

1.1.2 Specific Objectives

The specific objectives of this thesis are enumerated below:

1. Development of perception-based techniques for estimation of image structure.
2. Development of perception-based techniques for color image denoising.
3. Development of perception-based techniques for edge detection.
4. Development of perception-based techniques for the segmentation of coupled color/range images and color images.

1.2 Organization of the Thesis

This dissertation consists of eight chapters that have been arranged in four parts, namely: introduction, classical tensor voting, extensions of tensor voting and conclusions. The introductory part includes this chapter and Chapter 2, which presents a review of transversal topics related to this thesis. In order to separate transversal from specific topics, sections about previous related work have been placed just before the discussion of every proposed method.

Part I is devoted to the use of classical tensor voting in image analysis. Thus, Chapter 3 presents an in-depth analysis of the classical tensor voting method, focusing on its perceptual meaning and on implementation issues related to this technique. In addition, Chapter 4 describes two new applications of classical tensor voting to image analysis, namely structure estimation and segmentation of coupled color/range images.

Part II deals with extensions of tensor voting to applications that cannot be effectively tackled through the classical tensor voting. First, Chapter 5 describes the proposed extension of tensor voting to image denoising. The duality between the image denoising and edge detection problems has led to the proposal of a robust edge detector in Chapter 6, which stemmed from the method proposed in Chapter 5. Finally, the methods described in Chapters 5 and 6 have been used to propose a robust color image segmentation method described in Chapter 7. Experimental results show that these techniques perform significantly better than previous state-of-the-art approaches.

In the concluding part, Chapter 8 summarizes the contributions of this thesis and makes some final remarks on future lines of research.

Chapter 2

Background

This section presents a review of transversal topics related to this thesis. Section 2.1 briefly overviews the most relevant theories about how humans process visual information to analyze scenes. The importance of these theories for image processing and computer vision relies on the fact that most applications are expected to behave as humans do. The success of many so-called perception-based methods in image processing and computer vision, which intend to mimic the visual perception process, have demonstrated the usefulness of understanding such theories. Section 2.2 summarizes the most used perceptual representations of color and the issues related to quantitative measurements of perceptual color differences. This topic is especially important in this thesis since the methods proposed in Chapters 5 and 6 assume that perceptual color differences can be estimated accurately. Section 2.3 presents the state-of-the-art in perceptual organization methods. The aim of this section is to set tensor voting in the context of all perceptual organization methods and to show the advantages that led us to choose it as the theoretical platform upon which the methods proposed in this thesis are built. Section 2.4 shows the most common sources of noise in color images. This topic is particularly important for the development of robust methods intended to deal with real noise. Finally, Section 2.5 makes a brief introduction to tensors, which are used throughout the thesis.

2.1 Visual Perception

Visual perception can be defined as the process that converts retinal images into coherent and interpretable perceptual structures, such as objects and materials, as well as their spatial relationships and their relative importance in the scene.

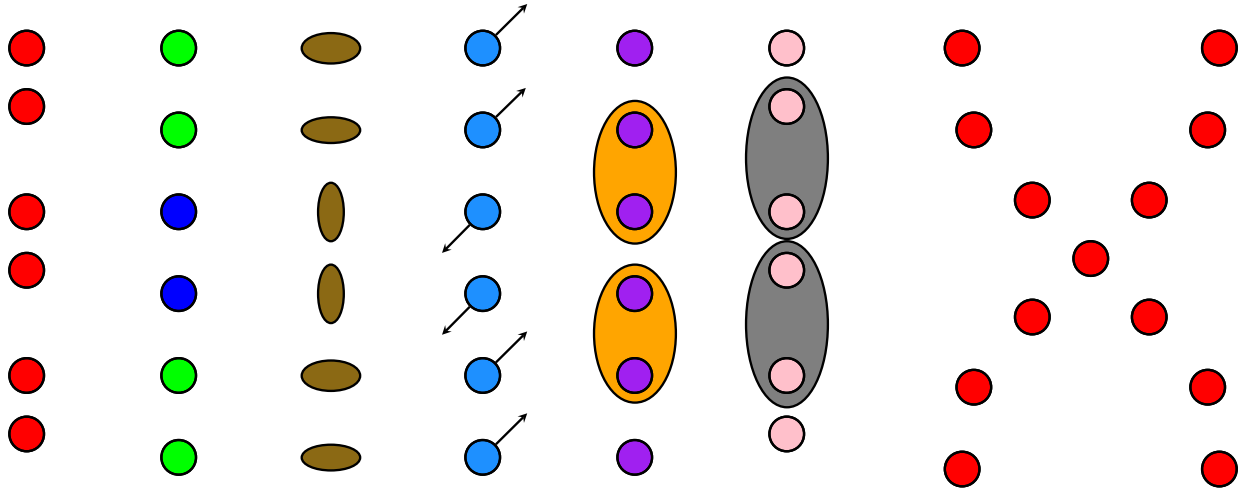


Figure 2.1: Principles of proximity (first column), similarity (second column), orientation (third and fourth columns), surroundedness (fifth and sixth columns) and good continuation (the *x*-shaped figure) of the Gestalt psychology determine whether or not every pair of points are perceptually grouped by the human visual system.

Researchers have proposed various models of human visual perception. One of the most recognized theories is *Gestalt* ("shape" in German) psychology. Gestalt psychologists argue that visual perception is driven by an organizational process that describes how certain perceptual structures are more likely than others without an a priori knowledge of the contents of the scene. They have formulated a set of principles that guide this perceptual organization, including, among many others, proximity, similarity, closure, good continuation, common fate, symmetry, relative size, surroundedness, orientation, collinearity and co-curvilinearity (Koffka, 1935; Kubovy and Pomerantz, 1981). Some of these principles are shown in Figure 2.1. Although more than one-hundred principles have already been proposed, research for finding new grouping principles is still very active. For instance, Geisler's group is investigating on statistic measurements of regularities in natural environments in order to find new grouping principles (Geisler, 2008; Geisler and Perry, 2009).

In addition, the potential power of Gestalt principles has also been investigated. Elder and Goldberg (2002) conducted psychophysical experiments to evaluate the inferential power of proximity, parallelism, co-circularity, and similarity, obtaining that proximity is the most powerful Gestalt principle to do perceptual grouping by far, and that all of them are not strongly correlated. More recent experiments support that local principles of organization can account for around 80% of the perceptual information in natural images, only leaving the remaining 20% to global principles (Fowlkes et al., 2007).

Gestalt psychologists consider that these perceptual principles are manifestations of the law of *Prägnanz* ("conciseness" or "goodness of form" in German), which states that, among various perceptual structures, the brain selects the simplest and most stable shape. Gestaltists think that ambiguity can only appear when two or more perceptual structures are equally simple and stable. The large amount of psychophysical experiments that are in accordance with this theory have led to its wide acceptance in describing the visual perception process.

The main problem faced when using Gestalt psychology to analyze scenes by means of computers is that it is descriptive rather than explanatory, that is, it only describes phenomena but not how they are produced. In this direction, methods to measure *perceptual saliency* (the importance of features to attract the viewer's attention), must be proposed in order to measure structure stability and simplicity when applying the law of Prägnanz, since Gestalt psychology does not propose a methodology to measure it. One possibility to solve this issue is to use a variety of Gestalt principles in order to reinforce the perceptual saliency. This is the approach followed by most of the methods reviewed in Section 2.3. Another possibility is to design methods not only based on Gestalt principles, but also on other psychological phenomena that help them to determine perceptual saliency. For example, Siagian and Itti (2007) classify outdoor scenes by using the phenomenon of visual attention, which is the ability of the Human Visual System (HVS) to fixate the eyes in the most relevant object in the image. More in-depth information about Gestalt principles and psychophysical studies that support this theory are presented in the classic book by Koffka (1935). More recent reviews can also be found in Desolneux et al. (2008, Chapter 2), Bruce et al. (2003, Chapter 6) or Kubovy and Pomerantz (1981).

Gestalt psychology is not sufficient to explain visual perception at higher levels of abstraction where knowledge about the scenarios becomes important. In this direction, psychological evidence has been found supporting that context information and visual memory are important cues in object recognition (Henderson and Hollingworth, 1999). According to many researchers, only at higher levels of abstraction it is possible to speak of object recognition. At these abstraction levels, previous knowledge allows objects observed in 2D and 3D scenes to be identified and classified. As a result, a previous learning step is necessary to accomplish the recognition process, as previous knowledge about the expected objects is required. Two paradigms have been proposed to understand the object recognition process followed by HVS: viewpoint-based and object-based recognition. Figure 2.2 shows both approaches. The first one assumes that a canonical and/or various views of objects are stored in the brain to subsequently match them to the

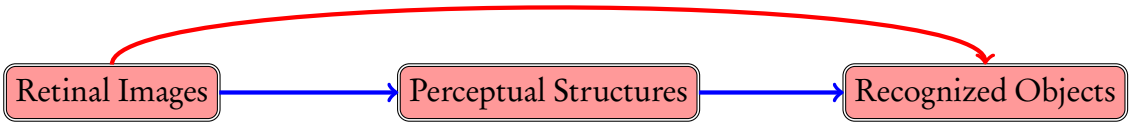


Figure 2.2: Paradigms of the object recognition process. The viewpoint-based approach follows the red path, whereas the object-based object recognition follows the blue path. Evidence has been found supporting both paradigms.

scene. The second paradigm argues that 3D object representations are stored in memory and matching is performed in 3D after obtaining a 3D representation of the scene. For example, Tarr and Bülthoff (1998) follows the first approach, while Marr (1982) follows the latter. Since psychophysical evidence has been found supporting both paradigms, the hypothesis that HVS performs object recognition by using a combination of recognition in both 2D and 3D is very likely (Hayward, 2003). A comparison of both paradigms is presented in (Bruce et al., 2003, Chapter 9).

Another point of active research in visual perception has been on how HVS processes different types of visual information. On the one hand, it has been found that HVS does not give the same relevance to different types of visual information. Thus, lightness is, by far, the most important type of visual information (Novak and Shafer, 1987). This reasoning has been used by gray-scale image analysis methods to discard other types of visual information. Not surprisingly, they have been able to yield good enough results in many applications, although they only use one type of visual information. However, more information is usually required in many other applications. Furthermore, evidence has been found that HVS is more sensitive to chromatic and texture information than to 3D shape (Pan et al., 2005). Moreover, HVS has been found more sensitive to color than to texture (Chen et al., 2005). This is likely due to the fact that HVS could process color faster than complex textures. On the other hand, evidence has been found supporting that HVS processes multiple visual cues in parallel (Foster and Gilson, 2002). Thus, parallel multi-level scene interpretation models could explain, for example, why some perceptual tasks are more sensitive to orientation changes than others (Blais et al., 2009).

Perception of noise and color are of particular interest for this thesis. Perception of noise has been associated with preattentive perception, which is the ability of HVS to discriminate patterns almost immediately without a thorough study of the image (Julesz, 1991). In addition, researchers have found that preattentive perception is mainly a local process. Furthermore, evidence has been found supporting that HVS can effectively reject noise but at a cost of a reduction

in visual acuity (Huertas et al., 2006; Chou and Liu, 2007; Lucassen et al., 2008). This is likely due to a reconstruction process carried out by HVS in order to infer the information at the contaminated pixels through an averaging mechanism. However, more research is necessary to understand the process of noise rejection carried out by HVS. Aspects related to perception of color are presented in the following Section.

2.2 Perceptual Representation of Color

Color representation has been actively investigated in the last century, leading to the proposal of many color models. A problem in color representation is that every color model has a restricted scope of use. Hence, the most appropriate color model must be selected depending on the application. For example, the RGB, CMY and CMYK color models are a very good choice to display colors on screens or paper, but they are not the best option to analyze images. Some applications need color models more closely related to the human perception of color. However, the highly non-linear behavior of HVS in front of color stimuli has made it difficult to propose perceptual color representations.

Most color models use three parameters to describe color, by considering that, physiologically, HVS has only three types of photoreceptors cells sensitive to color, namely cones L, M, and S, which are capable of recognizing long, medium and short wavelenghts of light respectively. Figure 2.3 shows the typical responsivity of these cells to different wavelenghts of light. HVS has approximately six times more M cones than S cones, and twice more L cones than M cones (Fairchild, 2005, Chapter 1). This makes the HVS more insensitive to bluish colors (with lower wavelengths) and more sensitive to greenish ones (with medium-range wavelengths), due to the overlap of responsivity of L and M cones.

HSI has been one of the most popular color models used in applications that require perceptual information because transformations from and to the RGB model are inexpensive and can separate luminance from chroma information. Nevertheless, HSI is only a simplification of the human perception of color. In the 1920s, experiments were conducted to propose the CIE XYZ color model, which intended to be a better approximation of the responsivity of eye cones. It was designed in such a way that the Y coordinate measures the relative luminance of a color, while the remaining coordinates capture its colorfulness.

However, CIE XYZ is not correlated with the perception of color. Thus, CIELAB was pro-

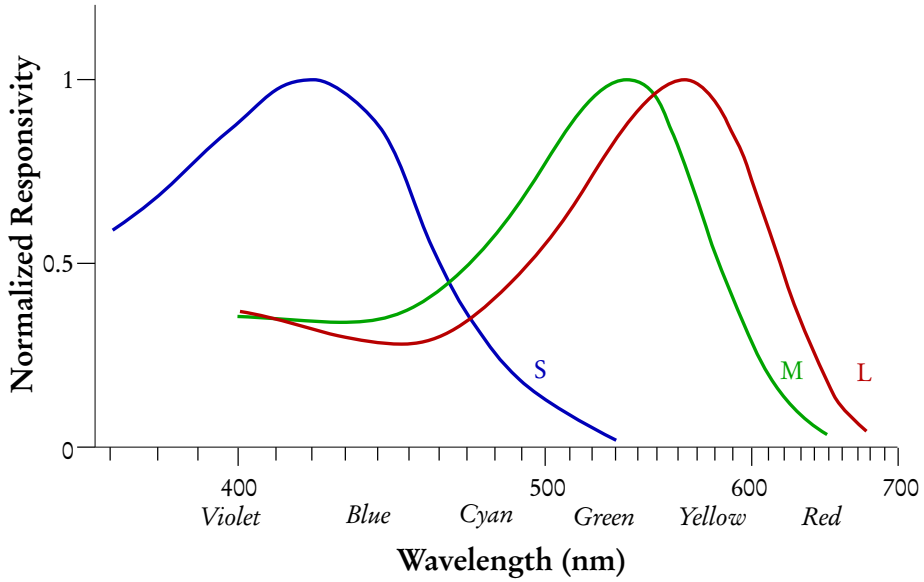


Figure 2.3: Normalized responsivity of eye cones.

posed in 1976 as an accurate model of the human perception of color (Wyszecki and Stiles, 2000, Chapter 3). This model is based on a large amount of psychophysical experiments conducted by various research groups. The most important design objective of CIELAB was to obtain a general formula to estimate perceptual color differences in order to quantitatively measure how different two colors appear to humans. It is important to note that for CIELAB, unlike HSI, color conversions from and to other color models are usually more complex, since they must take into account not only the color itself but also the global illumination and the viewing conditions (Wyszecki and Stiles, 2000, Chapter 3). The L, a and b values of the CIELAB color model are calculated from CIE XYZ through:

$$L = 116\, f(Y/Y_n) - 16, \tag{2.1}$$

$$a = 500\,[f(X/X_n) - f(Y/Y_n)], \tag{2.2}$$

$$b = 200\,[f(Y/Y_n) - f(Z/Z_n)], \tag{2.3}$$

where

$$f(t) = \begin{cases} t^{1/3} & \text{for } t > 0.008856 \\ 7.787t + 16/116 & \text{otherwise,} \end{cases} \tag{2.4}$$

with X_n, Y_n and Z_n being the CIE XYZ tristimulus values of the reference white point, which is the color that represents *white* under a specific illuminant (e.g., the sunlight or an incandescent

light). L represents lightness, which is the achromatic component of color, and a and b approximate redness-greenness and yellowness-blueness respectively. Hue, which can be thought of as the pure color, can be estimated in CIELAB as $h = arctan(b/a)$. In turn, chroma, which is the difference of a color from gray at a given hue and lightness, can be calculated as $C = \sqrt{a^2 + b^2}$.

Designers intended to make CIELAB a Euclidean color space. Thus, in theory, the Euclidean distance could directly be used to measure perceptual color differences. However, the perception of color difference by humans is a very complex, non-linear process that depends on many phenomena, and most of them were not taken into account in the experiments that led to the design of CIELAB. Instead of modifying CIELAB, some color scientists have devoted most of their efforts to proposing more accurate perceptual color difference equations based on CIELAB. Thus, the Euclidean difference has been replaced by more accurate formulas. The first one was proposed in 1994 (called CIEDE94) and a revisited version of the latter was proposed in 2000 (called CIEDE2000).

CIEDE2000 (Luo et al., 2001; Sharma et al., 2005) was designed by taking into account not only the fact that the CIELAB color space is not Euclidean (especially for bluish colors), but also the difficulty of having a single perceptual color difference formula for all possible applications. Thus, the strategy of its designers was to adjust the CIEDE2000 formula to a specific controlled environment, leaving to users' discretion the adjustment of three free parameters, k_L, k_C and k_H, in order to modify the relative importance of lightness, chroma and hue respectively. Thus, CIEDE2000 estimates the perceptual color difference between two colors c_1 and c_2 through:

$$\Delta E(c_1, c_2) = \sqrt{\left[\frac{\Delta L'}{k_L S_L}\right]^2 + \left[\frac{\Delta C'}{k_C S_C}\right]^2 + \left[\frac{\Delta H'}{k_H S_H}\right]^2 + R_T \left[\frac{\Delta C'}{k_C S_C}\frac{\Delta H'}{k_H S_H}\right]}, \qquad (2.5)$$

where $\Delta L'$, $\Delta C'$, $\Delta H'$, S_L, S_C, S_H and R_T are functions of the lightness, chroma and hue of both colors c_1 and c_2. As a rule-of-thumb, the Just Noticeable Color Difference (JNCD) usually ranges from one to five CIEDE2000 units depending on the application (Berns, 2001).

Unfortunately, the adjustment of the parameters of CIEDE2000 is very difficult, since they should take into account many phenomena. For example, Johnson and Fairchild (2003b) report at least thirteen different phenomena that directly influence the perception of color, and Xin et al. (2005) have found evidence supporting that the perception of colors is highly influenced by the neighborhood. Partial solutions to this issue have been found by modeling the influence of some phenomena on the perceptual color difference through the CIEDE2000 parameters (e.g.,

Huertas et al., 2006; Chou and Liu, 2007), in particular, the influence of the background luminance and the amount of noise. In addition, recent psychophysical experiments have yielded that perception of noise in chroma and hue are somehow correlated but, in general, k_L, k_C and k_h should take different values (Lucassen et al., 2008). Although most researchers think that CIEDE2000 is far from ideal (Oleari et al., 2009; Kuehni, 2008, 2002; Berns, 2000), they agree that CIEDE2000 is the most accurate equation to calculate perceptual differences of color so far. Other three-stimuli color perceptual models include CIELUV, LMS or CMC, but CIELAB is the most used in perceptual applications by far.

On the other hand, some color researchers have also proposed new models to take into account more variables than CIELAB. For example, CIECAM02 includes a chromatic adaptation transform that takes into account the fact that different colors can appear similar under different illumination conditions (Moroney et al., 2002). Another example is S-CIELAB, which has been proposed as a spatial extension of CIELAB that takes into account the sensitivity of HVS to spatial frequency (Zhang and Wandell, 1997; Johnson and Fairchild, 2003a). More recently, iCAM06 was proposed as a unified color model by taking into account the advantages of both S-CIELAB and CIECAM02 (Kuang et al., 2007). The major issue of all these color models is that they require even more extra information than CIELAB to convert from and to other color models. Unfortunately, that information is usually unavailable in many image processing and computer vision applications. This research area is still very active and many research groups expect to propose new color models that comply with most of the characteristics of color perception in the following years.

2.3 Perceptual Organization Techniques

Perceptual organization (also known as perceptual grouping) is the ability of HVS both to extract significant relations from input data without any previous knowledge of the content and to group these data into meaningful higher level structures, even in cases of missing or noisy data (Loss et al., 2009; Lowe, 1985). In the last decades, many researchers have tried to emulate the perceptual organization process by means of computers in order to apply it to computer vision.

A classification of perceptual organization methods was proposed by Sarkar and Boyer (1993) depending on the structure of input data:

1. *Signal level* methods look for underlying structure in pixels in 2D or clouds of points in

3D.

2. *Primitive level* methods look for underlying structure in segments, clusters of pixels or region boundaries in 2D, or surface patches, surface discontinuities or clusters of points in 3D.

3. *Structural level* methods look for underlying structure on structured boundaries in 2D or groups of surface patches in 3D.

4. *Assembly level* methods look for underlying structure in basic structures such as polygons or closed regions in 2D or basic polyhedra in 3D.

According to this classification, the algorithms for image segmentation and edge detection based on pixel level processing can be classified as methods of perceptual organization at the signal level. Only limited success has been yielded at both the structural and assembly levels, since methods that use other psychological mechanisms, such as context information or visual memory, have outperformed grouping techniques. Recently, many studies have reported good performance at the primitive level, especially in the application of contour completion, which aims at inferring object boundaries from contour fragments (Gillam, 2001). These techniques are reviewed in the following subsections. More detailed reviews of perceptual organization techniques at the primitive level are presented by Neummann and Mingolla (2001) and Williams and Thornber (1999).

2.3.1 Voting Mechanisms

These methods use local knowledge in order to infer global knowledge through consensus mechanisms. One of the most successful techniques in this line has been *tensor voting* (Guy and Medioni, 1996, 1997; Medioni et al., 2000). It is a general framework adequate for problems in which information can be encoded by means of tensors. In 3D, every point encodes a tensor that is propagated to its neighbors through a convolution-like process. Afterwards, the analysis of the results leads to the location of surfaces, edges and junctions. This approach takes advantage of the Gestalt principles of proximity, similarity and good continuation in order to estimate perceptual saliency (cf. Section 2.1 for a definition).

This technique has been used in various contexts. For instance, it has been successfully applied to visual motion analysis (Min and Medioni, 2008; Nicolescu and Medioni, 2005), image

segmentation (Park et al., 2008; Moreno et al., 2007; Lim et al., 2007; Tao et al., 2002; Kang and Medioni, 2001), image restoration (Jia and Tang, 2004), image mosaicing (Jia and Tang, 2005), surface reconstruction (Lu et al., 2005), curvature estimation (Lombardi et al., 2008; Tong and Tang, 2005), stereo estimation (Mordohai and Medioni, 2006a), gap filling in medical images (Risser et al., 2008), fiber tracking in medical images (Schultz, 2010), image super-resolution (Tai et al., 2006a), image compression (Tai et al., 2006b), epipolar geometry estimation (Tong et al., 2004; Tang et al., 2001), mesh analysis (Kim et al., 2009), registration of 3D points (Reyes et al., 2007), denoising of random dot patterns (Shao et al., 2008), and edge detection (Moreno et al., 2010d; Loss et al., 2008; Massad et al., 2003), among many other applications. Similar approaches to tensor voting have been proposed by Yen and Finkel (1998), Nafziger et al. (1999), and Page et al. (2002). A detailed study of tensor voting can be found in Section 2.6.

Although tensor voting has been outperformed by other methods in 2D, basically because they take into account more Gestalt principles than the original proposal of tensor voting, it is still one of the most relevant perceptual organization methods, since, unlike other methods, it can be applied to many applications in computer vision without any adaptation. In addition, tensor voting can be used in Euclidean spaces of any number of dimensions, only needs a single parameter, and in principle, it is a non-iterative method, although further improvements have been reported from iterating tensor voting (Loss et al., 2009; Fischer et al., 2007).

A traditional problem of tensor voting has been its high computational cost. However, many efforts have been made to improve its performance in order to use it in real-time applications (cf. Chapter 3). The main disadvantage of tensor voting is that the application of the kernels defined to propagate shape information could not be the most appropriate approach in some cases (Part II shows examples of this situation), and the method can only be applied to points, making its application to levels of perceptual organization higher than the signal level difficult.

At higher levels of perceptual organization, Foresti and Regazzoni (2000) proposed a different hierarchical voting-clustering strategy. However, its kernel definition is dependent on the objects and structures expected in the scene. Dependence of kernel definitions on the application space is preferable as this could make the voting technique more general. Other voting methods, such as Sarkar and Boyer (1994) or Parvin et al. (2007), require several parameters that are usually difficult to tune.

2.3.2 Graph-Based Approaches

These methods take advantage of graph segmentation algorithms in order to group data. Most of the efforts have been devoted to the graph creation step, since the success of these methods is highly dependent on this process. In that way, researchers have proposed different methods to measure saliencies mainly inspired by Gestalt psychology in order to encode that information in the edges of the graph. Sarkar and Boyer (1998) presented one of the pioneering works in the utilization of spectral graph theory for perceptual grouping encoding measures of seven different Gestalt principles in the edges of the graphs. Soundararajan and Sarkar (2003) extended this work in order to compare minimum, average and normalized graph cuts applied to perceptual grouping, observing that none of them are superior to each other, since their performances are scene dependent. Moreover, various graph cutting techniques applied to perceptual organization were evaluated in (Yu, 2003, Chapter 2) yielding similar results. In addition, Wang et al. (2005) proposed ratio contours, another method based on graph cuts that uses Gestalt principles of proximity, similarity, good continuation and closure in order to estimate saliencies. An extension of this approach that also takes into account the symmetry principle is based on the fact that modeling more Gestalt principles can lead to better results (Stahl and Wang, 2008). Different approaches have also been proposed, such as the use of semidefinite programming in graphs (Keuchel et al., 2003). In 3D, minimum spanning trees have been used to obtain closed surfaces in triangular meshes in (Sappa, 2006). Although graph-based methods are useful in perceptual organization, they require the support of additional techniques to estimate perceptual saliency.

2.3.3 Stochastic Approaches

These algorithms use stochastic methods to group data. Many of the proposals have been made in the context of edge grouping and linking in the context of figure completion. One of the most relevant works within this family of approaches measures saliency based on stochastic completion fields, such that saliency is proportional to the number of closed random walks that visit an edge (Williams and Thornber, 1999). This idea takes into account the Gestalt principles of proximity, similarity, good continuation and closure, and was extended to image segmentation by Mahamud et al. (2003). The strategy by Vasseur et al. (1999) uses the Dempster-Shafer theory and the Gestalt principles of parallelism, similarity and symmetry to group data. Elder et al. (2003) use Bayesian inference and an initial prior model to close contours. More recently, Des-

olneux et al. (2008) proposed a probabilistic approach (so-called *a contrario* detection scheme) based on the Helmholtz perceptual principle, which states that a structure is perceived when a large deviation from randomness occurs. This can be used in perceptual organization tasks. Although this kind of approach is usually effective, the main problem with these methods is their high computational cost and the inherent difficulty to extend them to other contexts.

2.3.4 Other Approaches

Many other approaches have been proposed to perform perceptual organization. For instance, morphological operators were used by Jiang (2000) to obtain closed surfaces in range and 3D images. Feldman (2003) proposed an approach based on logic to encode Gestalt principles independently, making it extensible. Grossberg et al. (1989); Grossberg and Kelly (1999) proposed the so-called *FACADE* (form-and-color-and-depth) framework based on feature cooperation in a neural network. More recently, Bigand et al. (2006) used fuzzy logic for grouping line segments in order to reconstruct 3D scenes. The most important disadvantage of these methods is that they were proposed for grouping edges for figure completion, making it difficult their extension to other types of data and features at higher levels of abstraction.

2.4 Noise in Color Images

Noise caused by errors in the sensing process is a common problem in image analysis. Hence, algorithms must apply robust techniques in order to make them applicable to noisy real applications.

Noise has usually been assumed to be independent from spatial coordinates and from the pixel's values (Gonzalez and Woods, 2007, Chapter 5). These assumptions have led to modeling noise by means of the probability density function that more accurately describes the distribution of noise. Accordingly, noise has been classified as Gaussian, Rayleigh, Gamma, exponential, uniform, impulse, etc., depending on such a probability density function. In addition, noise is called *white* if it has the same power in all frequencies. Moreover, noise is usually modeled as additive or multiplicative depending on how it modifies the pixel's values. Thus, for instance, noise of color images has traditionally been modeled by means of additive white Gaussian noise (AWGN). This model is based on the central limit theorem, which states that, under certain

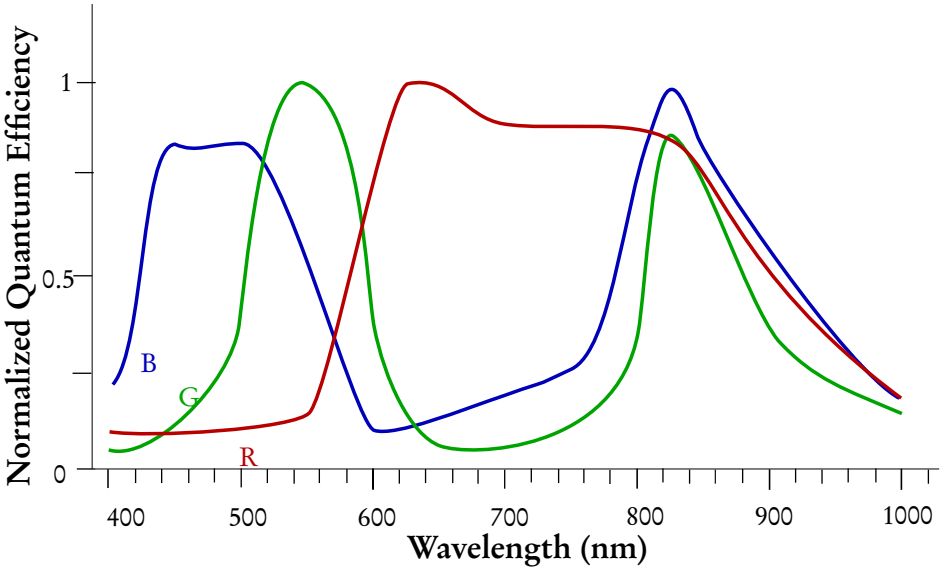


Figure 2.4: Normalized Quantum efficiency (it can be thought of as a measure of sensitivity) of a typical CCD or CMOS sensor for the three types of band pass filters, R, G, and B. An additional infrared filter is usually placed in the pipeline of the imaging sensor, since, as it is shown in the figure, R, G and B pixels are also sensitive to infrared frequencies (wavelength > 700). Compare these curves with Figure 2.3.

conditions, the sum of different independent signals with different probability density function tends to follow a Gaussian distribution. However, methods designed to reject AWGN usually have a low performance in real applications. In addition, some methods have successfully used specific sources of noise to determine whether or not an image was acquired with a camera (e.g., Chen et al., 2008). However, the discrimination of the specific source of noise used by the method from other sources of noise is only possible if noise is not Gaussian. These observations mean that the hypothesis of independence of signals made by the AWGN model is not completely supported by real data. This fact has fostered research on noise modeling in color images during the last years.

Most efforts have been made in identifying the sources and nature of noise in color images. The most important types of noise that have been associated with Charge-Couple Device (CCD) and complementary metal–oxide–semiconductor (CMOS) image sensors, which are the most used sensors for color image acquisition (Carlson, 2002; Fossum, 1997), are:

- Photon-shot noise: it is due to the quantum nature of light and especially affects bright pixels. It is usually assumed to follow a Poisson distribution.

- Photo response non-uniformity (PRNU): it is due to the fact that not all pixels have the

same response to light. It depends on the light level and on the temperature. For a given light and temperature, it can be approximated by means of a Gaussian distribution. Interestingly, the non-uniformity nature of this type of noise has been used in forensics to identify whether or not an image was taken with a specific camera, or if it was postprocessed (Chen et al., 2008).

- Cross-talk noise: it is due to energy inductions from one photoreceptor to its neighbors and depends on the physical distribution of the photoreceptors inside the sensor. This type of noise generates blurring artifacts and color mixing.

- Dark current noise: it is due to the response of the sensor during periods when it is less actively being exposed to light and it especially affects dark pixels. It can be divided into two types of noise:

  - Dark current shot noise: it is similar to the photon-shot noise and depends on temperature and time of exposure. It is usually assumed to follow a Poisson distribution.

  - Dark current non-uniformity (DCNU): it is due to the fact that different pixels produce different amounts of dark current noise and is highly dependent on temperature, making it difficult to model it through a single probability distribution.

- Thermal noise: it is related to the temperature in the sensor and readout circuitry. It is usually assumed to follow a Gaussian distribution.

- Flicker and random telegraph signal (RTS) noises: they are related to a variety of electronic phenomena in the sensors and readout circuitry. However, it has not been completely understood and modeled so far.

- Row noise: it stems from the fact that pixel's values are usually read by rows, so a noisy pixel can spread its noise to the next readings from the same row. It follows a spatial Gaussian distribution.

- Column noise: readout circuitry can also spread noise in columns. This kind of noise has been found more complicated to model.

- Quantization noise: this type of noise is unavoidable when the signal is converted from analog to digital.

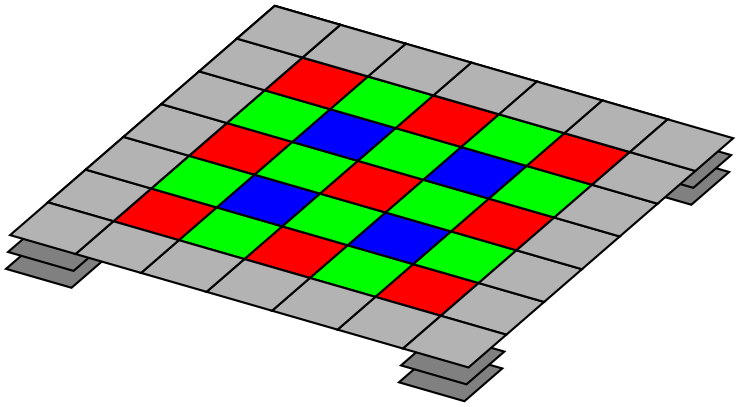


Figure 2.5: The Bayer pattern. Red, green and blue photoreceptors are intercalated in the sensor. There are twice green photoreceptors than red and blue ones, which is consistent with the fact that HVS is more sensitive to greenish colors (cf. Section 2.1).

- Fixed-pattern noise: it is mainly due to imperfections of the sensor.
- Geometric distortion: it is mainly generated by the lens.
- Atmospheric attenuation: it is mainly due to atmospheric conditions, such as dust, fog, etc.

Given the complexity of modeling all the noise sources, many researchers (e.g., Healey and Kondepudy, 1994; Tsin et al., 2001; Hwang et al., 2007b; Gow et al., 2007; Liu et al., 2008; Foi et al., 2008) have proposed simplified noise models that usually discards some of those sources. More in-depth information on these sources of noise can be found in Li and Nathan (2005) and Gow et al. (2007). In addition to these sources of noise, the camera pipeline includes unavoidable processes that are necessary to convert raw data into processable digital images. These processes can dramatically modify the original noise. There are mainly three of those processes:

1. Demosaicing: in this step, a raw image captured by a color filter array (CFA), which refers to the physical distribution of the different types of band pass filters lying on top of the photoreceptors inside the imaging sensor, is converted into an RGB image. There are three types of band pass filters, which are more prone to detect reddish, greenish and bluish colors respectively. Figure 2.4 shows the typical sensitivity of imaging sensors, including the effect of the aforementioned band pass filters. Usually, only a single type of band pass filter is used at every pixel. Thus, demosaicing algorithms aim at estimating the unknown color components at every pixel from the values acquired at their neighbors. The most usual CFA is the Bayer pattern, which is shown in Figure 2.5. As shown in the figure,

the different types of photoreceptors are intercalated in the array. The problems faced by demosaicing algorithms include spreading of noise, blurring and generation of aliasing artifacts (Li et al., 2008).

2. White balancing: the objective of this step is to globally adjust the colors in order to render the white color correctly. This process can involve complex operations on the raw image depending on the sophistication and accuracy of the algorithm.

3. Gamma correction: cameras adjust the brightness of images in order to make them more perceptually close to the real scenes. This correction involves exponential operations on every pixel. Thus, it can lead to abrupt changes in the distribution of noise.

These processes are achieved in a different manner by every brand of cameras. For example, Grossberg and Nayar (2004) have found a large variety of camera response functions, which intend to measure the effects of white balancing and gamma correction in the final image, for several camera brands.

2.5 A Brief Introduction to Tensors

Tensor-based processing techniques for color and gray-level images have become popular in the last two decades thanks to their robustness. Tensors derive from the generalization of the vector concept. According to tensor analysis, scalars and vectors are in fact tensors of zeroth and first order respectively. In turn, tensors of second order can be modeled by means of matrices. In particular, second order tensors modeled by means of symmetric positive semidefinite matrices have been the most widely used in image analysis. They are characterized by having all their eigenvalues greater or equal to zero. This kind of tensor can be graphically represented by means of ellipses in 2D or ellipsoids in 3D, as seen in Figure 2.6. As shown in the figure, the eigenvectors of the matrix that represents a tensor determine the orientation of the ellipse (or ellipsoid in 3D), whereas the eigenvalues determine its shape. Figure 2.7 shows the three extreme cases of this kind of tensor in 3D:

- *Stick* tensor: it is a tensor with only one eigenvalue greater than zero. Graphically, *stick* tensors can be seen as the extreme case in which the ellipsoid of Figure 2.6 collapses into a stick. As a convention, *stick* tensors are plotted in this thesis as double arrows, indicating that the tensor is encoding an orientation instead of a direction.

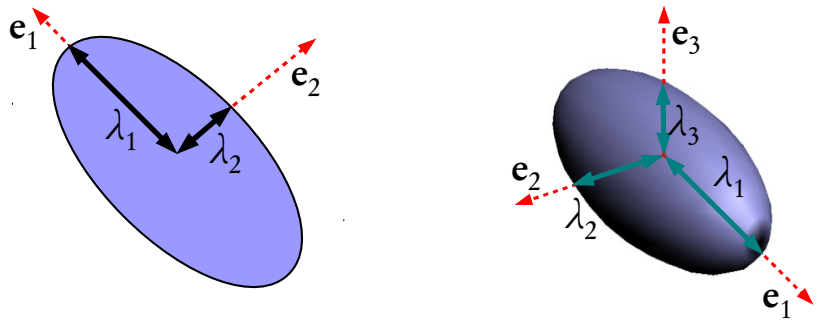


Figure 2.6: A second order tensor can be represented by an ellipse in 2D (left) or an ellipsoid in 3D (right). Their shapes depend on the eigenvalues and eigenvectors of the tensors.

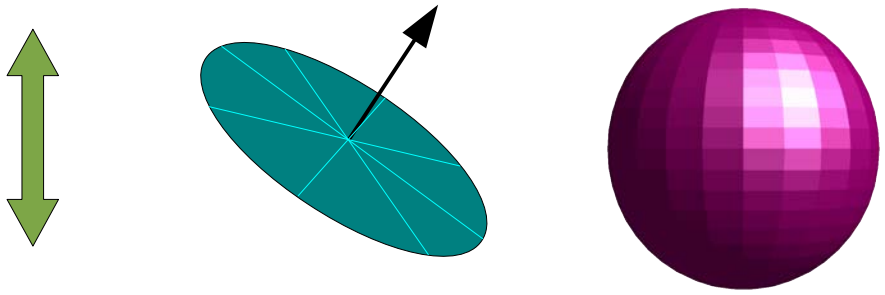


Figure 2.7: Extreme cases of second order symmetric positive semidefinite tensors in 3D. From left to right, a *stick*, a *plate* and a *ball* tensor respectively.

- *Plate* tensor: it is a tensor with two eigenvalues equal and greater than zero and the third one equal to zero. Graphically, *plate* tensors model the case in which the ellipsoid of Figure 2.6 is flattened along one dimension, acquiring the shape of a plate or disc.

- *Ball* tensor: it is a tensor whose three eigenvalues are equal and greater than zero. *Ball* tensors model the case in which the ellipsoid of Figure 2.6 becomes ball-shaped. Ball tensors are represented by means of $k\mathrm{I}$, where k is a scalar and I is the identity matrix.

The following operations on second order symmetric positive semidefinite tensors are of interest in this thesis:

- Vector tensorization: a vector, $\mathbf{v}$, can be tensorized by means of the outer product $\mathrm{T} = \mathbf{v} \otimes \mathbf{v} = \mathbf{v}\mathbf{v}^T$. The resulting matrix T has only one eigenvalue different from zero and whose principal eigenvector corresponds to $\pm\mathbf{v}$, that is, the direction of a vector is lost when it is tensorized. For this reason, it is said that tensors encode orientations instead of directions, as is the case of vectors. The result of tensorizing a vector is a *stick* tensor.

- Tensor summation: the sum of two tensors can be achieved by means of the summation of

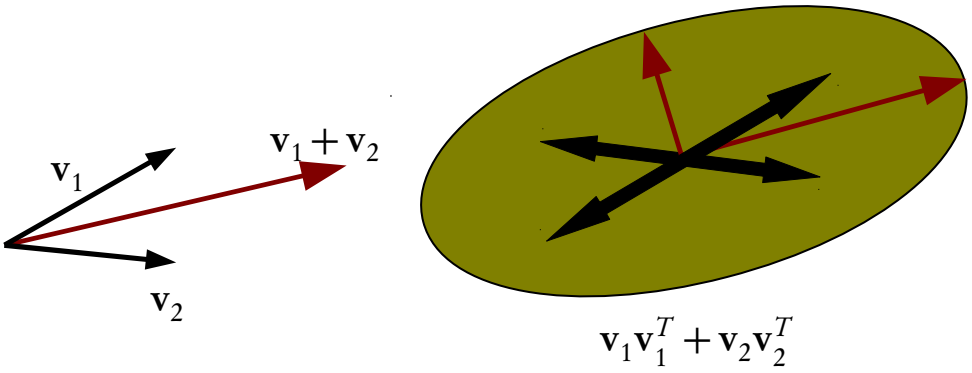


Figure 2.8: The summation of tensorized vectors (except when $\mathbf{v}_1$ and $\mathbf{v}_2$ are perpendicular) not only stores the sum of the two vectors (through the principal eigenvector and eigenvalue), but also stores how different the summands are (through the secondary eigenvector and eigenvalue).

matrices. Tensor summation has some interesting properties compared to the summation of two vectors $\mathbf{v}_1$ and $\mathbf{v}_2$:

- ⋆ If $\mathbf{v}_2 = -\mathbf{v}_1$, then $\mathbf{v}_1 + \mathbf{v}_2 = 0$. That implies that vector summation can lead to loss of information. In the same case, $\mathbf{v}_1\mathbf{v}_1^T + \mathbf{v}_2\mathbf{v}_2^T = 2\,\mathbf{v}_1\mathbf{v}_1^T$, that is, the orientation information is reinforced.
- ⋆ If two vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ are not perpendicular, the principal eigenvector of the tensorized vectors is parallel to $\mathbf{v}_1 + \mathbf{v}_2$. However, tensor summation yields more information than vector summation, since the second eigenvalue can be used to measure how different is the orientation of $\mathbf{v}_1$ from that of $\mathbf{v}_2$. This can be graphically seen in Figure 2.8.

▹ Tensor transformation: given a transformation matrix M, for example, a rotation, it can be used to transform a tensor by means of: $T' = M\ T\ M^T$, where T' is the transformed version of T.

Despite its usefulness, tensor-based techniques must deal with the inherent complexity involved in tensor processing. For instance:

▹ Difference: difference of symmetric positive semidefinite tensors, carried out by means of the difference of matrices, can lead to non-positive semidefinite matrices.

▹ Multiplication (also known as outer product) and differential operators: they increase the order of tensors, making it difficult to give them an intuitive interpretation (Welk et al., 2007).

- Norm: there is not a standard notion of norm of a tensor. The most used tensor norm is the so-called Frobenius norm that is given by: $||\mathrm{T}||_F = \sqrt{trace(\mathrm{T}\,\mathrm{T}^T)}$, but it cannot be used in all applications.

- Similarity/distance measurements: there is not a unique measurement of similarity/distance of tensors. Thus, specific measurements must be chosen for every specific application (Peeters et al., 2009).

Other types of tensors have been used in engineering applications. For example, higher-order tensors are popular in DT-MRI (Basser and Pajevic, 2007; Moakher, 2008; Schultz, 2009), non-positive semidefinite tensors are used in continuum mechanics (Holzapfel, 2000) and biomechanics (Cowin and Doty, 2007), and asymmetric tensors are becoming to be in use in computer graphics (Zhang et al., 2009). Despite that, this thesis is mainly concerned with the use of second order symmetric positive semidefinite tensors. More in-depth information about tensors can be found in textbooks on tensor analysis, such as Abraham et al. (1988) or Lebedev and Cloud (2003).

2.6 Classical Tensor Voting

Among the perceptual organization techniques reviewed in Section 2.3, tensor voting is the most flexible one, since researchers have successfully been able to adapt it with excellent results to problems well beyond the ones which it was originally applied to. This is the reason why this technique has been chosen as one of the pillars of this dissertation.

Guy and Medioni (1996, 1997), and Medioni et al. (2000) proposed tensor voting (hereafter referred to as "classical tensor voting") as a robust technique for extracting perceptual structures from a cloud of points. In 3D, the method estimates saliency measurements of how likely a point lies on a surface, an edge, 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 through the so-called *stick*, *plate* and *ball* tensor voting. As shown in Section 2.3.1, tensor voting has been proven effective in noisy environments for many applications.

The formulation of classical tensor voting presented in this section is different from, although equivalent to, the original formulation by Guy and Medioni (1996, 1997) and Medioni et al. (2000). It has been chosen since it simplifies the descriptions in the following chapters, especially

those in Chapter 3.

Classical tensor voting comprises three stages. In a first stage, a tensor is initialized at every point of the given cloud of points either with a *stick* tensor oriented with an initial estimation of its normal, or with a *ball*-shaped tensor if such a priori information is not available. That is, if $\mathbf{n}$ is the initial estimation of the normal at a point, its tensor can be initialized to $\mathbf{n}\mathbf{n}^T$. Otherwise, the tensor is initialized to the identity matrix. Afterwards, every tensor is decomposed into its three components, namely: a *stick*, a *plate* and a *ball* component (cf. Section 2.5). Every component casts votes to the neighboring points by taking into account the information encoded by the voter in that component. Every vote is a tensor that encodes the most likely direction(s) of the normal at a neighboring point. Finally, the votes are summed up and analyzed in order to estimate degrees of surfaceness, edginess and junctionness at every point. Points with low surfaceness, edginess and junctionness are assumed to be noisy observations.

More formally, the tensor voting at $\mathbf{p}$, TV($\mathbf{p}$), is given by:

$$\mathrm{TV}(\mathbf{p}) = \sum_{\mathbf{q} \in neigh(\mathbf{p})} \mathrm{SV}(\mathbf{v}, \mathrm{S_q}) + \mathrm{PV}(\mathbf{v}, \mathrm{P_q}) + \mathrm{BV}(\mathbf{v}, \mathrm{B_q}), \tag{2.6}$$

where $\mathbf{q}$ represents each of the points in the neighborhood of $\mathbf{p}$, SV, PV and 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 $\mathrm{S_q}$, $\mathrm{P_q}$ and $\mathrm{B_q}$ are the *stick*, *plate* and *ball* components of the tensor at $\mathbf{q}$ respectively:

$$\mathrm{S_q} = (\lambda_1 - \lambda_2)\left(\mathbf{e}_1\mathbf{e}_1^T\right), \tag{2.7}$$

$$\mathrm{P_q} = (\lambda_2 - \lambda_3)\left(\mathbf{e}_1\mathbf{e}_1^T + \mathbf{e}_2\mathbf{e}_2^T\right), \tag{2.8}$$

$$\mathrm{B_q} = \lambda_3\left(\mathbf{e}_1\mathbf{e}_1^T + \mathbf{e}_2\mathbf{e}_2^T + \mathbf{e}_3\mathbf{e}_3^T\right), \tag{2.9}$$

where λ_i and $\mathbf{e_i}$ are the i-th eigenvalue and its corresponding eigenvector of the tensor at $\mathbf{q}$ respectively.

Saliency measurements can be estimated from an analysis of the eigenvalues of the resulting tensors in 2.6. 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, edginess and junctionness respectively. In some applications (e.g., the application presented in Section 4.2), the normalized saliencies, given by $\hat{s_i} = s_i/\lambda_1$, are preferred to the original saliencies. Points whose three eigenvalues are small are regarded as noise. In addition, eigenvector $\pm\mathbf{e_1}$ represents the most likely normal for points lying on a surface, whereas $\pm\mathbf{e_3}$ represents the most likely tangent direction of an edge for points belonging to that edge.

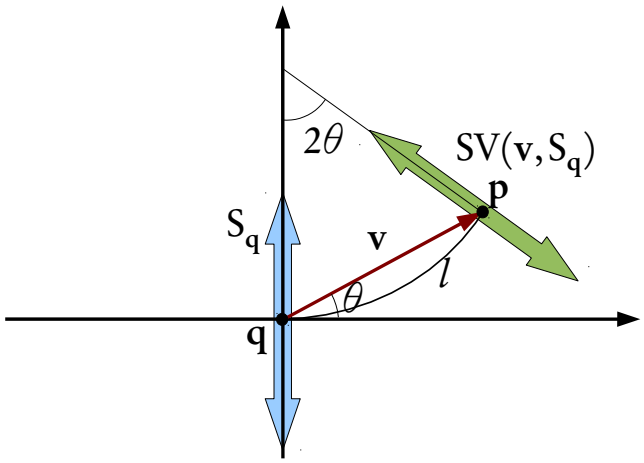


Figure 2.9: *Stick* tensor voting. A *stick* S_q casts a *stick* vote $SV(\mathbf{v}, S_q)$ to $\mathbf{p}$, which corresponds to the most likely tensorized normal at $\mathbf{p}$.

The next subsections describe the processes required to calculate *stick*, *plate* and *ball* tensor votes.

2.6.1 *Stick* Tensor Voting

Stick tensors are used by tensor voting to encode the orientation of the surface normal at a specific 3D point. Tensor voting handles *stick* tensors through the so-called *stick* tensor voting, which aims at propagating surfaceness in a neighborhood by using the perceptual principles of proximity, similarity and good continuation borrowed from the Gestalt psychology (cf. Section 2.1). The *stick* tensor voting is based on the hypothesis that surfaces are usually smooth. Thus, tensor voting assumes that normals of neighboring points lying on a surface change smoothly. This process is illustrated in Figure 2.9. Given a known orientation of the normal at a point $\mathbf{q}$, which is encoded by S_q, the orientation of the normal at a neighboring point $\mathbf{p}$ can be inferred by tracking the change of the normal on a joining smooth curve. Although any smooth curve can be used to calculate *stick* votes, a circumference is usually chosen. A decaying function, s_{1_s}, is also used to weight the vote as defined below.

For a circumference, it is not difficult to show from Figure 2.9 that:

$$SV(\mathbf{v}, S_q) = s_{1_s} \left[R_{2\theta}\, S_q\, R_{2\theta}^T \right], \tag{2.10}$$

where θ is the angle shown in Figure 2.9 and $R_{2\theta}$ is a rotation with respect to the axis $\mathbf{v} \times (S_q\, \mathbf{v})$, which is perpendicular to the plane that contains both $\mathbf{v}$ and S_q. Let λ be the eigenvalue of S_q

greater than zero. Angle θ can be calculated as:

$$\theta = \arcsin\left(\sqrt{\frac{\mathbf{v}^T\,\mathrm{S_q}\,\mathbf{v}}{\lambda\mathbf{v}^T\mathbf{v}}}\right). \tag{2.11}$$

A point $\mathbf{q}$ can only cast *stick* votes for $\theta \leq \pi/4$, since the hypothesis that both points $\mathbf{p}$ and $\mathbf{q}$ belong to the same surface becomes more unlikely for higher values of θ. On the other hand, a weighting function, s_{1s}, is used to reduce the strength of the vote with the length of the curve, l, given by:

$$l = \frac{||\mathbf{v}||\,\theta}{\sin(\theta)}, \tag{2.12}$$

and with its curvature, κ, given by:

$$\kappa = \frac{2\,\sin(\theta)}{||\mathbf{v}||}. \tag{2.13}$$

Thus, s_{1s} was defined by Medioni et al. (2000) as:

$$s_{1s}(\mathbf{v}, \mathrm{S_q}) = \begin{cases} e^{-\frac{l^2+b\kappa^2}{\sigma^2}} & \text{if } \theta \leq \pi/4 \\ 0 & \text{otherwise,} \end{cases} \tag{2.14}$$

where b and σ are parameters. In practice, l ranges from $||\mathbf{v}||$, when $\theta = 0$, to $\frac{\pi}{2\sqrt{2}}\,||\mathbf{v}|| \approx 1.11\,||\mathbf{v}||$, when $\theta = \pi/4$.

2.6.2 *Plate* Tensor Voting

Tensor voting utilizes *plate* tensors to encode edges in 3D. Ideally, if a point belongs to an edge, the third eigenvector of its tensor must be aligned with the tangent to the edge at that point, and λ_3 must be zero. Tensor voting handles *plate* tensors through the so-called *plate* tensor voting. Unlike the *stick* tensor voting, whose formulation derives from perceptual rules to propagate surfaceness, *plate* votes are computed in a constructive way. Thus, the *plate* tensor voting uses the fact that any *plate* tensor, P, can be decomposed into all possible *stick* tensors inside the *plate*. Let $\mathbf{e_i}$ be the eigenvectors of P, R_β be a rotation with respect to an axis parallel to $\mathbf{e_3}$, which is perpendicular to P, and $\mathrm{S_P}(\beta) = R_\beta \mathbf{e_1}\mathbf{e_1}^T R_\beta^T$ be a *stick* inside the *plate* P derived from $\mathbf{e_1}$. Thus, any *plate* tensor P can be written as:

$$\mathrm{P} = \frac{\lambda_1 + \lambda_2}{2\pi}\int_0^{2\pi} \mathrm{S_P}(\beta)\,d\beta. \tag{2.15}$$

Taking into account that $S_P(\beta)$ is a *stick* tensor, the *plate* vote is defined as the aggregation of *stick* votes cast by all the *stick* tensors $S_{P_q}(\beta)$ that constitute P_q. Thus, the *plate* vote is defined as:

$$PV(\mathbf{v}, P_q) = \frac{\lambda_1}{\pi} \int_0^{2\pi} SV(\mathbf{v}, S_{P_q}(\beta))\, d\beta. \tag{2.16}$$

2.6.3 *Ball* Tensor Voting

Ball tensors are utilized by tensor voting to encode junctions or noise. The *ball* tensor voting is defined similarly to the *plate* tensor voting, that is, in a constructive way. 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:

$$B = \frac{\lambda_1 + \lambda_2 + \lambda_3}{4\pi} \int_\Gamma S_B(\phi, \psi)\, d\Gamma, \tag{2.17}$$

where Γ represents the surface of the unitary sphere. Using the same argument as in the case of the *plate* tensor voting, the *ball* vote is defined as:

$$BV(\mathbf{v}, B_q) = \frac{3\lambda_1}{4\pi} \int_\Gamma SV(\mathbf{v}, S_{B_q}(\phi, \psi))\, d\Gamma. \tag{2.18}$$

Part I

CLASSICAL TENSOR VOTING

Chapter 3

Efficiency Improvement of Classical Tensor Voting

Despite the power of classical tensor voting, in practice, it cannot be used in applications where efficiency is an issue. This is mainly attributed to the high computational cost of those implementations based on the formulation of Section 2.6, especially regarding the *plate* and *ball* tensor voting.

This chapter proposes two different approaches for implementing tensor voting efficiently. The first one is based on a numerical approximation of the *plate* and *ball* tensor voting, which are mainly responsible for the complexity of the method. The second approach is based on a simplified formulation that implements the same perceptual rules followed by the classical tensor voting, although reducing its numerical complexity.

The chapter is organized as follows. Section 3.1 presents the numerical approach for implementing tensor voting efficiently. Section 3.2 proposes a simplified version of tensor voting based on the perceptual meaning of the *stick*, *plate*, and *ball* tensor voting processes. Section 3.3 shows some results to compare the classical tensor voting with the two methods proposed in the chapter. Finally, Section 3.4 discusses the obtained results and makes some final remarks.

3.1 Efficient Formulation for *Plate* and *Ball* Votes

The evaluation of the *stick* tensor voting is inexpensive, since the rotations involved in that process can be easily avoided by following the geometric constructions of Figure 2.9. Actually, the complexity of the *stick* tensor voting mainly stems from the computation of an arcsine, which

is required to calculate l, and an exponential, which is required by (2.14). In addition, these computations are not necessary for $\theta > \pi/4$.

Additional efforts have also been made to make the *stick* tensor voting even more efficient. For example, by applying steerable filters in 2D (Franken et al., 2006), and tensorial harmonics in 3D in order to compute *stick* votes in the frequency domain (Reisert and Burkhardt, 2008). Unfortunately, extensions of these methods to calculate *plate* and *ball* votes have not been proposed so far, mainly due to the difficulty to adapt the integrals in (2.16) and (2.18) to the frequency domain.

On the other hand, computing *plate* and *ball* votes is highly time consuming, since (2.16) and (2.18) cannot be analytically simplified. Thus, researchers usually interpolate precomputed tensor fields in order to reduce the complexity of the *plate* and *ball* tensor voting. Unfortunately, the amount of precomputed information can grow rapidly if several values of parameter b are used, since the voting fields strongly vary with it. This problem becomes more noticeable if (2.14) is used instead of the scale-invariant counterpart (3.2), since the shape of the voting fields will also vary with σ in that case. In practice, this fact involves the use of complex systems for data access and memory management, which are not always available in many applications.

Following a different strategy, Mordohai and Medioni (2006b) and Lu et al. (2005) discard part of the votes for the sake of efficiency. Min and Medioni (2006) have proposed an efficient implementation of tensor voting that avoids discarding such information through a parallel implementation on a graphics processing unit (GPU). However, the improvement is determined by the number of available processing units.

The following subsections present a numerical approach to implement *plate* and *ball* votes efficiently. Instead of approximating the integrals of equations (2.16) and (2.18), the proposed approach is based on the scale-invariant version of *stick* tensor voting described in the following section.

3.1.1 Scale-Invariant *Stick* Tensor Voting

Although the formulation of *stick* tensor voting given in Section 2.6 is inexpensive, it is not scale invariant. Scale invariance is a desirable property, since it can make techniques less prone to size variations that may occur in image data (Lindeberg, 1994). In addition, a scale-invariant version of the *stick* tensor voting is essential for analyzing the properties of the *plate* and *ball* tensor

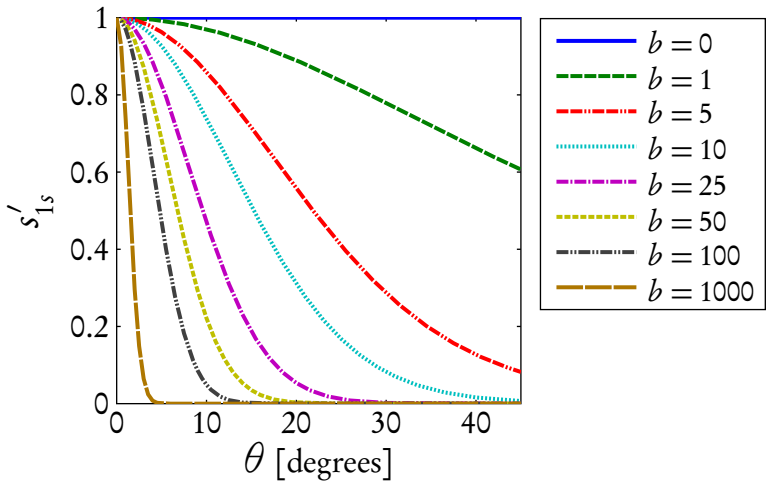


Figure 3.1: Evolution of s'_{1s} with respect to θ for some values of b.

voting, as shown in the next subsections.

The non scale-invariance of the *stick* tensor voting comes from the exponent in (2.14). From dimensional analysis, that exponent must be dimensionless in order to assure scale-invariance (Bridgman, 1922). A way to convert (2.14) to a scale-invariant equation is to use the normalized curvature, $\overline{\kappa}$, instead of the curvature κ. The normalized curvature is given by (Andrews and Gibson, 2001):

$$\overline{\kappa} = \kappa \frac{||\mathbf{v}||}{2} = \frac{2\ \sin(\theta)}{||\mathbf{v}||} \frac{||\mathbf{v}||}{2} = \sin(\theta), \tag{3.1}$$

where θ and $\mathbf{v}$ are shown in Figure 2.9. Since the normalized curvature is dimensionless, it does not require to be weighted by $1/\sigma^2$. Thus, the *stick* tensor voting becomes scale-invariant if (2.14) is replaced by:

$$s_{1s}(\mathbf{v}, \mathrm{S_q}) = \begin{cases} e^{-\frac{l^2}{\sigma^2} - b\overline{\kappa}^2} & \text{if } \theta \leq \pi/4 \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

This equation preserves the spirit of (2.14) in the sense of penalizing *stick* votes by both distance and curvature. Moreover, plate and ball votes calculated by means of (3.2) also become scale-invariant thanks to spatial symmetry. Therefore, (3.2) will be used instead of (2.14) in the remaining of the chapter due to its scale invariance.

Figure 3.1 shows the effect of parameter b on s_{1s}. In this plot, s'_{1s} models the factor of s_{1s} that does not depend on the 3D space, which is given by:

$$s'_{1s} = e^{-b\overline{\kappa}^2}. \tag{3.3}$$

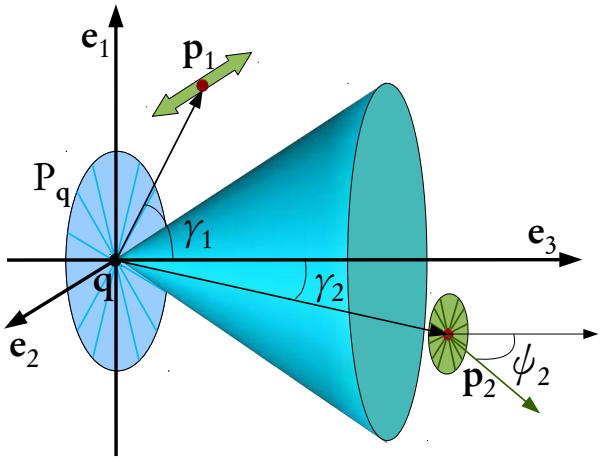


Figure 3.2: Examples of the *plate* tensor voting. A tensor P_q casts votes to its neighbors with a shape that depends on the cone shown in the figure. Votes are close to *sticks* for points outside the cone ($\mathbf{p}_1$), or close to *plates* for points inside the cone ($\mathbf{p}_2$).

The figure shows that b can be used to increase the preference for flat surfaces over curved ones. As an example, *stick* votes are negligible when $\theta > 5°$ for $b = 1000$. This means that, in this case, higher values of θ will not be considered to propagate surfaceness. This behavior could be useful to discriminate between flat surfaces and curved ones.

3.1.2 Efficient *Plate* Tensor Voting

Scale-invariance and spatial symmetry can be used to analyze the *plate* votes. Besides parameters σ and b, the *plate* vote $\mathrm{PV}(\mathbf{v}, P_q)$ depends on two variables: the distance between $\mathbf{p}$ and $\mathbf{q}$, $||\mathbf{v}||$, and the angle γ between $\mathbf{e_3}$ and $\mathbf{v}$. As it can be seen in Figure 3.2, *plate* votes are close to *sticks* for points outside the depicted cone and close to *plates* for points inside the cone ($\gamma \leq \pi/4$). From the same figure, angle γ can be calculated similarly to the angle θ of the *stick* tensor voting:

$$\gamma = \arcsin\left(\sqrt{\frac{\mathbf{v}^T P_q \mathbf{v}}{\lambda_1 \mathbf{v}^T \mathbf{v}}}\right). \tag{3.4}$$

Thanks to scale-invariance, the *plate* vote can be divided into two independent functions: a scalar decaying function f, which mainly depends on the spatial distance between the voter and the votee, and a tensorial function H, which is invariant on spatial distance. In practice, f not only depends on $||\mathbf{v}||$ and σ, but also has a slight influence from γ. Thus, (2.16) can be rewritten as:

$$\mathrm{PV}(\mathbf{v}, P_q) = \lambda_1 f(\mathbf{v}, \gamma, \sigma)\, \mathrm{H}(\gamma, b). \tag{3.5}$$

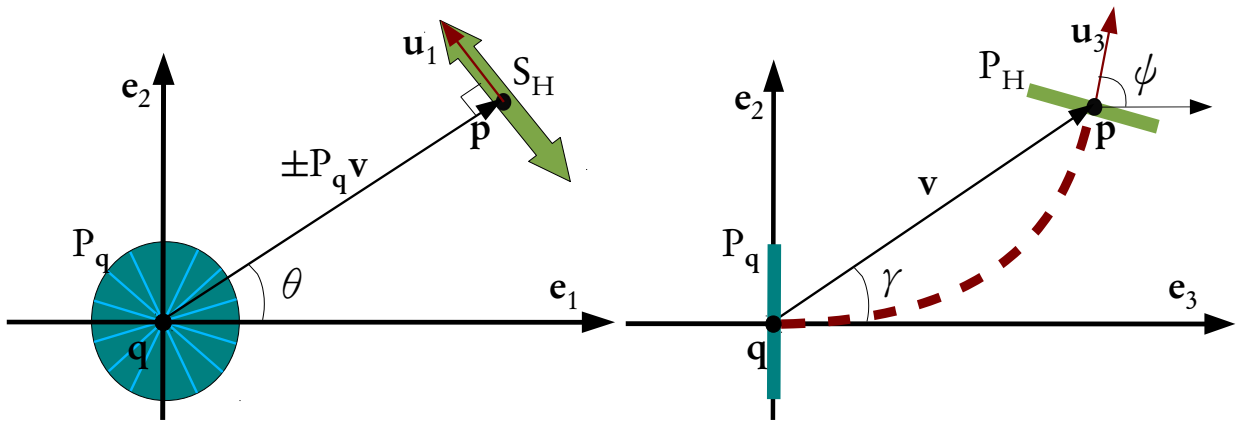


Figure 3.3: Components of function H. Left: the *stick* component, S_H, which is always perpendicular to the projection of $\mathbf{v}$ on P_q, given by $\pm P_q\mathbf{v}$. Right: the *plate* component, P_H, seen from profile.

The scalar function f is given by:

$$f(\mathbf{v},\gamma,\sigma) = e^{-\frac{t^2 \mathbf{v}^T \mathbf{v}}{\sigma^2}}, \tag{3.6}$$

where t theoretically ranges between the minimum and maximum values of $\gamma/sin(\gamma)$ for $0 < \gamma \leq \pi/4$, that is, $t \in [1, 1.11]$. However, in practice $t \in [1, 1.055]$. Factor t stems from the use of the arc length l in (3.2) instead of the distance between $\mathbf{p}$ and $\mathbf{q}$. Unfortunately it cannot be derived analytically. However, a good approximation can be obtained through $t = \frac{1+\gamma/sin(\gamma)}{2}$ for $0 < \gamma \leq \pi/4$ and $t = 1.033$ otherwise.

On the other hand, H determines the shape of the *plate* vote. H can be decomposed into its *stick* and *plate* components, whose shapes are shown in Figure 3.3:

$$\mathrm{H}(\gamma, b) = \mathrm{S_H} + \mathrm{P_H}. \tag{3.7}$$

These components can be calculated as:

$$\mathrm{S_H} = s'_{1p}\left(\mathbf{u}_1 {\mathbf{u}_1}^T\right), \tag{3.8}$$

$$\mathrm{P_H} = s'_{2p}\left(\mathbf{u}_1 {\mathbf{u}_1}^T + \mathbf{u}_2 {\mathbf{u}_2}^T\right), \tag{3.9}$$

with s'_{1p} and s'_{2p} being functions that cannot be analytically simplified, and $\mathbf{u}_i$ being the eigenvectors of function H which are calculated as follows. Symmetry makes $\mathbf{u}_1$ perpendicular to $\mathbf{e}_3$ and $\mathbf{v}$. Thus,

$$\mathbf{u}_1 = \begin{cases} \frac{\mathbf{e}_3 \times \mathbf{v}}{\|\mathbf{e}_3 \times \mathbf{v}\|} & \text{if } \mathbf{e}_3 \text{ and } \mathbf{v} \text{ are not parallel} \\ \mathbf{e}_1 & \text{otherwise} \end{cases} \tag{3.10}$$

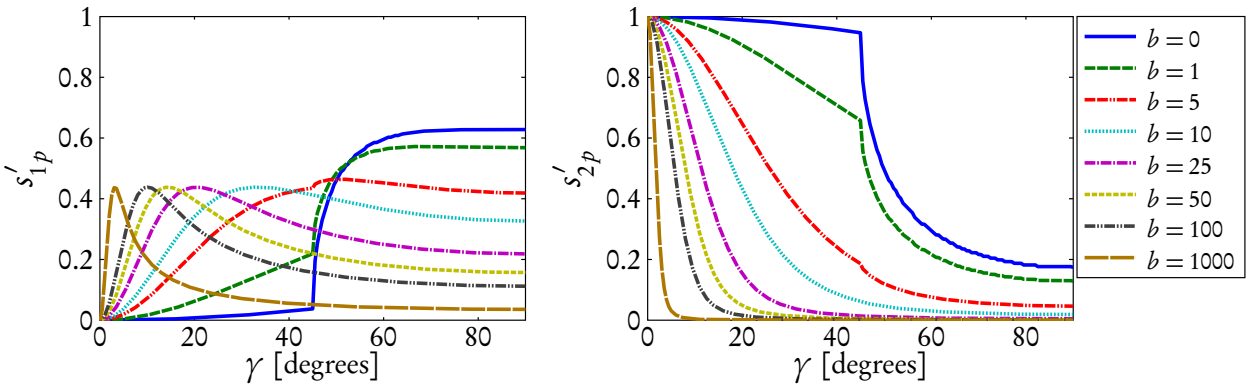


Figure 3.4: s'_{1p} and s'_{2p} as functions of γ for different values of parameter b.

Spatial symmetry also makes $\mathbf{u_3}$ lie on the plane that contains $\mathbf{e_3}$ and $\mathbf{v}$. The angle ψ between $\mathbf{u_3}$ and $\mathbf{e_3}$ is 2γ for $\gamma < \pi/4$, and $\pi - 2\gamma$ otherwise. Thus:

$$\mathbf{u_3} = R_{\psi}\mathbf{e_3}, \tag{3.11}$$

where R is a rotation with respect to axis $\mathbf{u_1}$. As in the case of the *stick* tensor voting, this rotation can be easily avoided by following the geometry of Figure 3.3. Having calculated $\mathbf{u_1}$ and $\mathbf{u_3}$, the remaining eigenvector $\mathbf{u_2}$ can be obtained as:

$$\mathbf{u_2} = \mathbf{u_3} \times \mathbf{u_1}. \tag{3.12}$$

Spatial symmetry also makes H not to have a *ball* component.

Figure 3.4 shows the curves of s'_{1p} and s'_{2p} vs. γ for different values of b. These curves have a discontinuity at $\gamma = \pi/4$. This was expected since the *stick* tensor voting also has a discontinuity at the same angle. The curves corresponding to s'_{1p} and s'_{2p} show that there are mainly two zones in the 3D space depending on which s'_{1p} or s'_{2p} is dominant, as was already shown in Figure 3.2. The curves of s'_{1p} and s'_{2p} also show that b can be used to control the spread of the voting cone of Figure 3.2, since it becomes narrower as b is increased. These curves can be approximated through any fitting method. As an example, Appendix A shows the fitting yielded by non-linear least squares.

The complexity of the proposed implementation of the *plate* tensor voting is mainly due to the computation of an arcsine (which is required to calculate γ), a logarithm required by the approximation of s'_{1p}, and two exponential functions: one for calculating $f(\mathbf{v}, \gamma, \sigma)\, s'_{1p}$ and another for calculating $f(\mathbf{v}, \gamma, \sigma)\, s'_{2p}$.

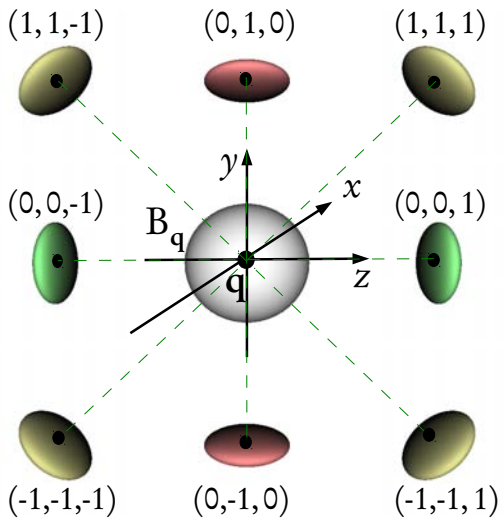


Figure 3.5: Examples of the *ball* tensor voting. Point **q** casts UFO-like shaped votes to its neighbors. $\mathrm{BV}(\mathbf{v},\mathrm{B_q})$ is shown for different positions of **p**.

3.1.3 Efficient *Ball* Tensor Voting

As in the case of *plate* tensor voting, scale-invariance and spatial symmetry can also be used to analyze the *ball* votes. Figure 3.5 shows some examples of *ball* votes. *Ball* votes are characterized by three properties. The first property is that *ball* votes have a UFO-like shape, that is, they are only flattened in one dimension. Thus, $\lambda'_1 = \lambda'_2 > \lambda'_3 > 0$, with λ'_i being the eigenvalues of $\mathrm{BV}(\mathbf{v},\mathrm{B_q})$ in (2.18). The second property is that the flattened direction of *ball* votes is always parallel to **v**. This means that the third eigenvector of a *ball* vote, $\mathbf{u_3}$, is parallel to **v**. The third property is that for some given parameters σ and b, both the size and flatness of *ball* votes only dependant on $||\mathbf{v}||$. This condition is given by the isotropic behavior of the *ball* tensor voting. Thus, the *ball* vote can be rewritten as:

$$\mathrm{BV}(\mathbf{v},\mathrm{B_q}) = \lambda_1 \left[R_\mathrm{v}\, S(s_{2b}, s_{2b}, s_{3b})\, \mathrm{B_q}\, R_\mathrm{v}^T \right], \tag{3.13}$$

where R_v is a rotation that makes $\mathbf{u_3}$ and **v** parallel, s_{2b} and s_{3b} are functions defined below, and S is a scale transformation that converts the *ball* tensor $\mathrm{B_q}$ into a UFO-like shaped tensor given by:

$$S(s_2, s_2, s_3) = \begin{pmatrix} s_{2b} & 0 & 0 \\ 0 & s_{2b} & 0 \\ 0 & 0 & s_{3b} \end{pmatrix}. \tag{3.14}$$

The main advantage of (3.13) is that the expensive integral of (2.18) is replaced by a rotation. However, this rotation term can also be avoided by constructing the tensor with **v**. Hence, (3.13)

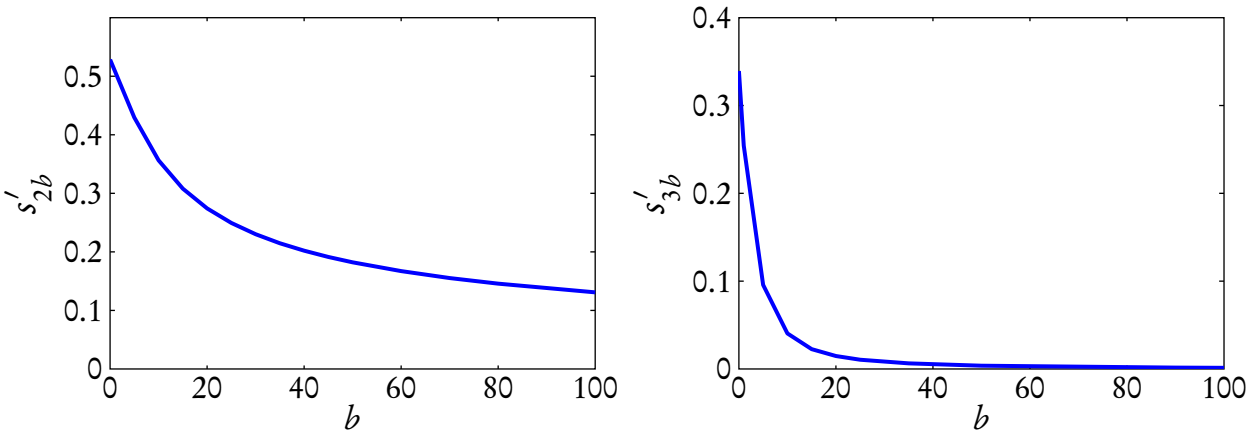


Figure 3.6: Evolution of s'_{2b} and s'_{3b} with respect to parameter b.

can be further simplified as:

$$\mathrm{BV}(\mathbf{v},\mathrm{B_q}) = \lambda_1 \left[s_{2b} \left(\mathrm{I} - \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} \right) + s_{3b}\,\mathrm{I} \right], \tag{3.15}$$

where I is the identity matrix.

The purpose of s_{2b} is to control the size of the vote, whereas the one of s_{3b} is to control how the vote is similar to a *plate* ($s_{3b} = 0$) or to a *ball* ($s_{3b} = s_{2b}$). Unlike the *plate* tensor voting, isotropy makes functions s_{2b} and s_{3b} only depend on $||\mathbf{v}||$ for specific parameters σ and b. Thus, thanks to the scale-invariance, s_{2b} and s_{3b} can be divided into a Gaussian decaying function on $||\mathbf{v}||$ with a standard deviation σ and functions s'_{2b}, s'_{3b} that only depend on b and cannot be analytically simplified. Thus, s_{ib} is defined as:

$$s_{ib}(\mathbf{v},\sigma,b) = s'_{ib}\, e^{-\frac{\mathbf{v}^T\mathbf{v}}{\sigma^2}}, \tag{3.16}$$

for $i \in [2,3]$. Figure 3.6 shows the evolution of s'_{2b} and s'_{3b} with respect to b. It can be seen that $s'_{2b} > s'_{3b}$ for all values of b. This means that *ball* votes are more similar to a *plate* than to a *ball* in general. It can also be seen that parameter b can be used to reduce the size of the *ball* vote. Appendix 3.1 shows the results of least squares fitting for these functions.

The complexity of the proposed implementation of the *ball* tensor voting is mainly due to the computation of an exponential (required in (3.16)), since the values of s'_{2b} and s'_{3b} must be computed only once.

3.2 Simplified Tensor Voting

This section explores an alternative to the numerical approach described in the previous section for calculating tensor voting efficiently. This alternative is based on a simplified formulation that reduces the numerical complexity while keeping the same perceptual rules of tensor voting.

The next subsections describe the proposed method to calculate *stick*, *plate* and *ball* tensor votes more efficiently.

3.2.1 *Stick* Tensor Voting

The classical *stick* tensor voting can be further simplified while keeping its perceptual meaning by redesigning the weighting function s_{1s} defined in (3.2). This function has two parameters: b that penalizes the curvature, and σ that penalizes both space and curvature (the latter through the $\theta/sin(\theta)$ factor included in the computation of l in (2.12)). Thus, for example, it is not possible to avoid the influence of curvature on the calculations, even selecting $b = 0$, since σ not only affects the distance but also the $\theta/sin(\theta)$ factor, which is related to curvature. For this reason, every parameter has a single task: σ becomes a scale parameter, and b a curvature parameter:

$$s_{1s}(\mathbf{v}, \mathrm{S_q}) = \begin{cases} e^{-\frac{\mathbf{v}^T\mathbf{v}}{\sigma^2} - b\ sin^2(\theta)} & \text{if } \theta \leq \pi/4 \\ 0 & \text{otherwise.} \end{cases} \tag{3.17}$$

This equation has the additional advantage that the arcsine required for calculating *stick* votes is no longer necessary.

3.2.2 *Plate* Tensor Voting

Proposing simplified equations for the *plate* tensor voting requires to understand the perceptual meaning of *plate* votes. From the analysis carried out in Subsection 3.1.2, it can be stated that, from a perceptual point of view, a *plate* vote encodes two different hypothesis, one for every component of the vote. On the one hand, the hypothesis made by the *stick* component of the *plate* vote is that a neighboring point $\mathbf{p}$ of the voter $\mathbf{q}$ should belong to a surface that abuts the edge that crosses $\mathbf{q}$. However, spatial symmetry makes such a surface be a plane, since the *stick* component is always tangent to the plate $\mathrm{P_q}$ (see Figure 3.7). Thus, the *stick* component of the

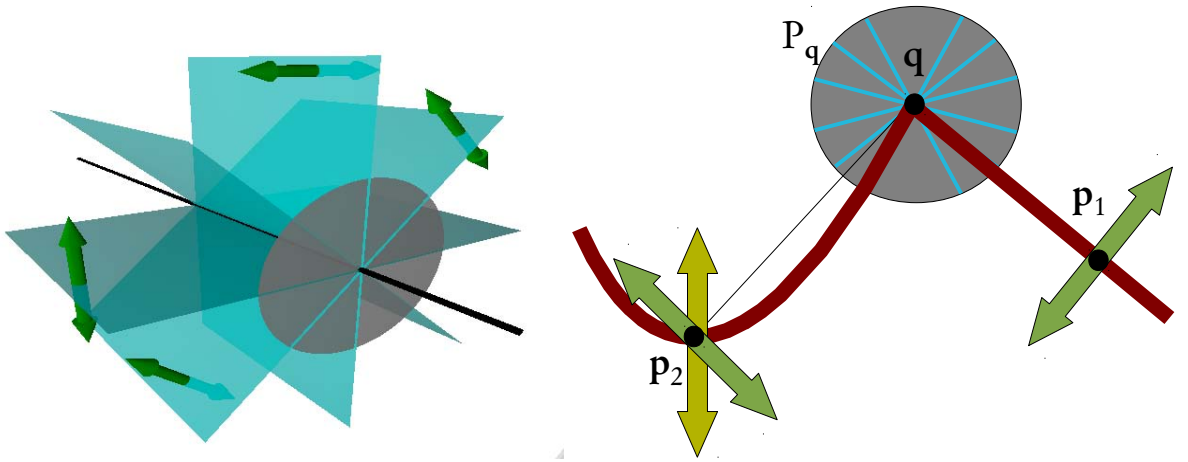


Figure 3.7: *Stick* component of the *plate* vote. Left: an edge votes for a plane that crosses the edge. Right: a *plate* tensor in the intersection between a flat and a curved surface (depicted in red). The *stick* component of the *plate* vote can reinforce surfaceness in flat surfaces (see vote at $\mathbf{p}_1$) but also can lead to errors in curved surfaces (see vote at $\mathbf{p}_2$ in green).

plate tensor voting can be thought of as a *stick* tensor voting that makes a stronger hypothesis than the *stick* tensor voting itself, since curved surfaces are only encouraged in the latter. As seen in Figure 3.7, this *stick* component is not convenient in all cases, since it can lead to errors in curved surfaces. This *stick* component mainly appears outside the cone of Figure 3.2, since points inside the cone can either belong to the edge that crosses $\mathbf{q}$ or to another surface.

On the other hand, the hypothesis made by the *plate* component of the *plate* vote is that both points, $\mathbf{p}$ and $\mathbf{q}$, should belong to the same edge. In that sense, $\mathbf{p}$ completes the path of the edge that crosses $\mathbf{q}$. This component only appears inside the cone of Figure 3.2, since points outside that cone are unlikely to belong to the same edge. Thus, the *plate* component of the *plate* vote can be thought of as the natural extension of the *stick* tensor voting in which edginess instead of surfaceness is smoothly propagated by following similar rules. Hence, the *plate* component can be considered to be based on the same Gestalt principles as the *stick* tensor voting, namely proximity, similarity and good continuation, but adapted to edginess propagation.

Taking into account these arguments, the following equation is proposed to calculate *plate* votes:

$$\mathrm{PV}(\mathbf{v},\mathrm{P}_\mathbf{q}) = s_{2p}\ \left[R_{2\gamma}\ \mathrm{P}_\mathbf{q}\ R_{2\gamma}^T\right] + \alpha_P\ \lambda_1\ s_{1p}\left(\mathbf{u}_1\mathbf{u}_1{}^T\right), \tag{3.18}$$

where $\alpha_P \in [0,1]$ is a new parameter to control the influence of the *stick* component on the *plate* vote, λ_1 is the largest eigenvalue of $\mathrm{P}_\mathbf{q}$, $\mathbf{u}_1$ is calculated through (3.10), $R_{2\gamma}$ is a rotation with

respect to $\mathbf{u}_1$, and s_{ip} are weighting functions given by:

$$s_{2p}(\mathbf{v},\mathrm{P_q}) = \begin{cases} e^{-\frac{\mathbf{v}^T\mathbf{v}}{\sigma^2}-b\ sin^2(\gamma)} & \text{if } \gamma \leq \pi/4 \\ 0 & \text{otherwise.} \end{cases} \tag{3.19}$$

$$s_{1p}(\mathbf{v},\mathrm{P_q}) = \begin{cases} e^{-\frac{\mathbf{v}^T\mathbf{v}}{\sigma^2}-b\ cos^2(\gamma)} & \text{if } \gamma > \pi/4 \\ 0 & \text{otherwise,} \end{cases} \tag{3.20}$$

As stated in the previous section, the rotation term can be avoided by following the geometry of Figure 3.3. Thus, the complexity of this alternative mainly stems from a single exponential function for s_{1p} or s_{2p} depending on the angle γ. The conflictive effect of the *stick* component in scenarios with curved surfaces can be inhibited by setting the new parameter α_P to zero.

3.2.3 *Ball* Tensor Voting

A perceptual interpretation of *ball* votes, necessary for proposing a simplified *ball* tensor voting, can be obtained from the analysis performed in Subsection 3.1.3. As stated before, a *ball* vote only consists of a *plate* and a *ball* component. On the one hand, the meaning of the *plate* component is that both points, **p** and **q**, should belong to a straight edge in the direction that joins both points. Although a *ball* tensor at **q** represents a complete uncertainty about the normal direction at that point, this uncertainty is reduced in the direction **v**, because both points could belong to a straight edge that is likely joining both points.

On the other hand, the meaning of the *ball* component is that points near a junction should have a junctionness saliency different from zero. From a different point of view, normal uncertainty at a point infers some normal uncertainty at its neighborhood. Unlike the *plate* component, it is difficult to justify from the perceptual point of view the existence of the *ball* component of the *ball* vote since junctions are not usually close to each other. However, it could be useful in iterative schemes (e.g., Fischer et al., 2007; Loss et al., 2009) in order to induce uncertainty for those cases in which the tensors are initialized with not too accurate values.

Hence, the same equation (3.15) is proposed to calculate *ball* votes, but with the following weighting functions:

$$s_{2b}(\mathbf{v}, \mathrm{B_q}) = e^{-\frac{\mathbf{v}^T\mathbf{v}}{\sigma^2}} \tag{3.21}$$

$$s_{3b}(\mathbf{v}, \mathrm{B_q}) = \alpha_B \, s_{2b}(\mathbf{v}, \mathrm{B_q}), \tag{3.22}$$

where parameter $\alpha_B \in [0, 1]$ can be used to control the influence of the *ball* component on the *ball* vote. Thus, the high complexity of the *ball* tensor voting is reduced to the computation of a single exponential function. It is important to remark that isotropy makes these functions to be independent from curvature.

3.3 Experimental Results

The formulations of the original, efficient and simplified tensor voting presented above were coded in MATLAB on an Intel Core 2 Quad Q6600 with a 4GB RAM in order to compare the new proposed schemes with the classical tensor voting.

Regarding efficiency, Table 3.1 shows that the classical formulation of tensor voting is impractical for many applications. As an example, assume that a small data set consists of 1,000 points, and that the propagation of votes is restricted to the 25 nearest points. Thus, the computation of 25,000 *stick*, *plate* and *ball* votes are required. With the classical tensor voting, they can be calculated in between 16.85 minutes and 36.04 days depending on the desired precision (controlled by the integration step). The efficient formulation of tensor voting proposed in Section 3.1 only takes 5.33×10^{-5}, 1.75×10^{-4}, and 5.36×10^{-5} seconds for every *stick*, *plate* and *ball* vote respectively. Thus, in the aforementioned example, the proposed formulation only takes 7.05 seconds. In addition, this time can be further improved by implementing the method in a non-interpreted programming language, such as C or C++. The efficiency of the simplified tensor voting is slightly better than the approach of Section 3.1. In this case, the *stick*, *plate* and *ball* votes can be processed in 5.10×10^{-5}, 1.52×10^{-4}, and 4.21×10^{-5} seconds respectively on average.

Comparisons of the results obtained through the efficient formulation of tensor voting presented in Section 3.1 and those from the classical tensor voting are not necessary, since the former was designed to yield the same results as the latter. However, the simplified tensor voting can yield different results, especially for curved surfaces or noisy scenarios. Experiments were conducted to assess such differences. Figure 3.8 shows the synthetic point-sampled surfaces used in

Table 3.1: Speed measurements of the classical tensor voting.

Integration step (degrees)	*Plate* votes		*Ball* votes	
	Time (s)	Votes per second	Time (s)	Votes per second
0.5	8.71×10^{-2}	11.48	124.48	8.03×10^{-3}
1	7.93×10^{-2}	12.61	41.10	2.43×10^{-2}
2	3.25×10^{-2}	30.77	10.90	9.17×10^{-2}
5	1.90×10^{-2}	52.63	1.75	0.57
10	1.20×10^{-2}	83.33	0.45	2.22
20	6.54×10^{-3}	152.88	0.12	8.01
30	5.61×10^{-3}	178.25	0.05	19.19
45	5.29×10^{-3}	189.04	0.03	28.49

these experiments and their noisy counterparts. These clouds of points have been designed in such a way that it is possible to obtain analytical computations of normals and edge orientations. As suggested by Medioni et al. (2000), tensors were initialized with unitary *ball* tensors, that is, with identity matrices. In addition, two rounds of tensor voting were applied: the first one only considered the *ball* tensor voting, whereas the second round only considered both the *stick* and *plate* tensor voting, as proposed by Medioni et al. (2000). Parameters α_P and α_B were set to zero, since curved surfaces are present in some data sets, σ was set to one. Independent experiments were run for b equals to 0 and 10.

The mean angular error between $\mathbf{e_1}$ and ideal normals on surfaces, and of $\mathbf{e_3}$ and ideal edge orientations at edges have been used to measure the accuracy of the algorithms for estimating normals and edge orientations respectively. Table 3.2 shows that the simplified tensor voting (STV) has a better performance than the classical tensor voting (TV) in general, except for estimating e_1 in the noiseless data sets, in which TV was marginally better. STV yielded better results for estimating edge orientations in both noiseless and noisy scenarios. As expected, both methods had a similar performance for the pyramid data sets, given the lack of curved surfaces. Regarding efficiency, TV spent around 3.5 hours to process every data set with an integration step of 20°. In contrast, STV only took around 20 seconds per sequence.

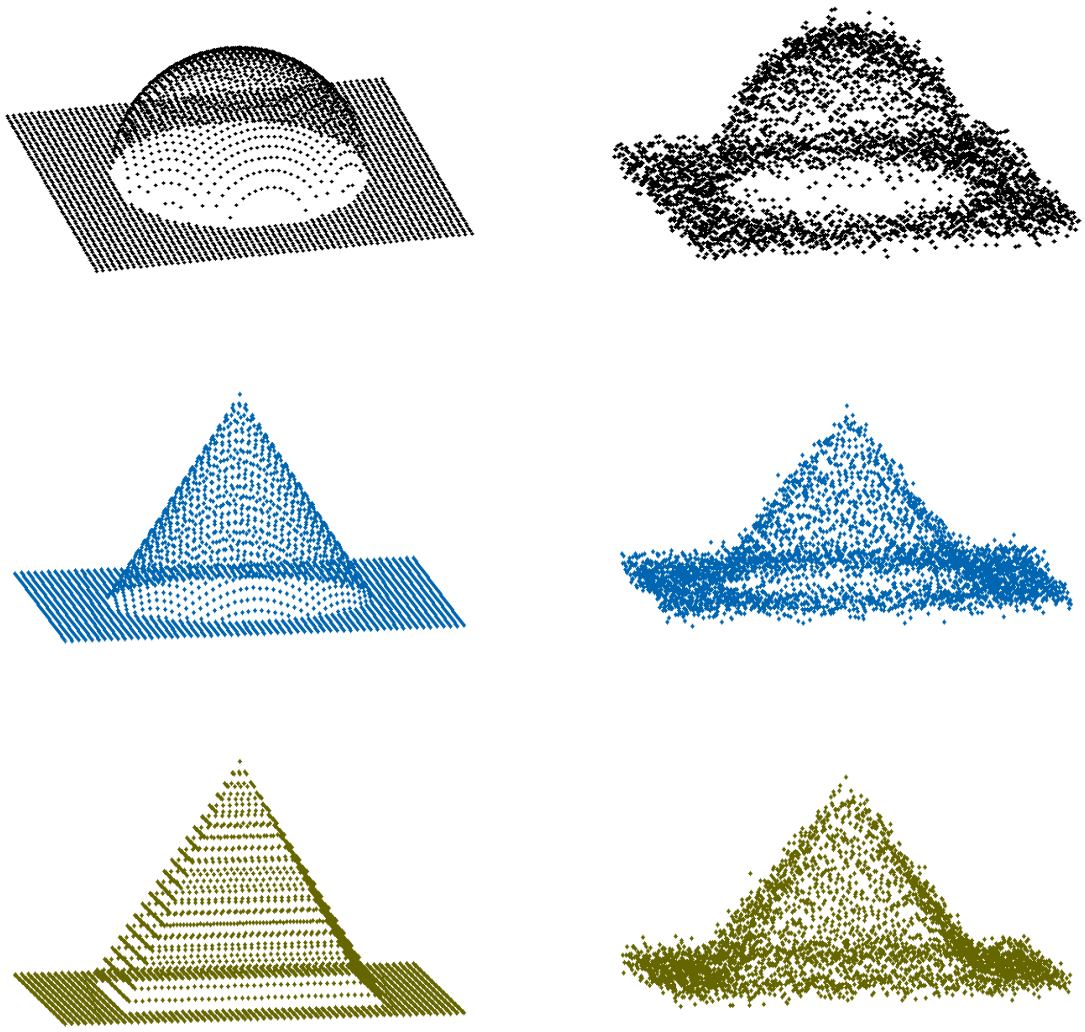


Figure 3.8: Clouds of points used in the experiments. Left: point-sampled surfaces, each constituted by 3,721 points. Right: a noisy version of the same surfaces (Gaussian noise with standard deviation of 0.2).

Table 3.2: Mean angular error of e_1 and e_3 in degrees

Sequence	b	Noiseless (TV : STV)		Noisy (TV : STV)	
		e_1	e_3	e_1	e_3
Semisphere	0	**0.51** : **0.51**	6.01 : **0.23**	6.55 : **6.07**	9.68 : **8.25**
	10	**0.50** : 0.53	5.52 : **1.25**	5.95 : **5.44**	8.49 : **5.61**
Cone	0	**1.59** : 1.69	2.61 : **0.40**	6.44 : **6.12**	8.27 : **7.52**
	10	**1.60** : 1.65	2.48 : **0.28**	6.35 : **5.88**	7.35 : **6.24**
Pyramid	0	**1.55** : 1.89	3.17 : **2.61**	6.20 : **6.18**	6.41 : **6.22**
	10	**1.67** : 1.97	**1.93** : 2.28	5.59 : **5.48**	**5.25** : **5.25**

3.4 Summary

In this chapter two alternative formulations have been proposed in order to significantly reduce the high computational complexity of the *plate* and *ball* tensor voting. The first formulation makes numerical approximations of the votes obtained from an in-depth analysis of the *plate* and *ball* voting processes. The second one proposes simplified equations to calculate votes that are based on the perceptual meaning of the classical tensor voting. Both formulations have a complexity of order $O(1)$. This can help broaden the use of tensor voting in more applications. In addition, according to the conducted experiments, the simplified tensor voting has been found more appropriate for estimating normals and edge orientations in both noisy scenarios and clouds of points with curved surfaces, by setting the new parameters α_P and α_B to zero.

In addition, the chapter has shown perceptual interpretations for the *stick*, *plate* and *ball* tensor voting. The *stick* tensor voting and the *stick* component of the *plate* tensor voting are used to reinforce surfaceness, whereas the *plate* components of both the *plate* and *ball* tensor voting are used to boost edginess. Junctionness is only intentionally strengthened by the *ball* component of the *ball* tensor voting, but it is usually small compared with the other votes.

Chapter 4

New Applications of the Classical Tensor Voting

As stated before, tensor voting has been used in a variety of applications (cf. Section 2.3.1). This chapter shows the application of tensor voting to two new applications: image structure estimation and image segmentation of coupled color/range images. Section 4.1 describes the use of tensor voting as an alternative to the structure tensor for image structure estimation. Traditionally, tensor voting has been applied to a subset of pixels of an image, for example, to those belonging to an edge. However, that section shows how tensor voting can also be applied to the whole image by taking into account some normalization factors. Section 4.2 shows how tensor voting can be combined with graph-based techniques to segment very noisy images acquired through stereo vision. The method yields good results taking into account the highly noisy nature of this kind of image.

4.1 Image Structure Estimation

Image structure estimation aims at extracting low-level features, such as edges, corners or texture. The most popular technique for this task is the so-called structure tensor (Förstner, 1986). It has been used in a multitude of applications such as edge detection (Förstner, 1994), corner detection (Kenney et al., 2005; Rohr, 1994), texture analysis (Rao and Schunck, 1991; Rousson et al., 2003), image filtering (Weickert, 1999a), image compression (Hwang et al., 2007a), and optic flow estimation (Lucas and Kanade, 1981; Bigün et al., 1991). It has gained popularity thanks to its robustness, efficiency and easiness of implementation. In addition, it depends on

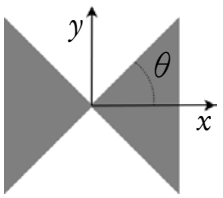


Figure 4.1: A hourglass-shaped kernel used for convolutions in 2D. For a specific value of θ, only the spatial locations whose angle in polar coordinates is in the range [-θ,θ] or [$\pi-\theta$,$\pi+\theta$] have values different from zero. It is usually combined with a radial decaying function, such as a Gaussian function.

a single parameter, which is usually easy to tune. The main hypothesis made by the structure tensor is that the gradient usually changes in a neighborhood slowly, so the structure can be estimated by means of a weighted sum of the orientations of the gradient in a neighborhood. For a gray-scale image, the structure tensor, J, is defined as the convolution of a Gaussian with the tensorized gradient of the image (Förstner, 1986):

$$\mathrm{J} = G_\sigma * \nabla u \nabla u^T, \tag{4.1}$$

where G_σ is the Gaussian with zero mean and standard deviation σ, ∇u is the gradient of the image u, and $\nabla u \nabla u^T$ represents the tensorized gradient at every pixel.

Despite its popularity, the structure tensor also has important shortcomings, such as detection of features in unstructured regions, loss of small features, appearance of false corners, and misplacement of corners. These shortcomings are mainly related to the use of a Gaussian kernel, since different orientations of the gradient can be mistakenly integrated at some pixels. This fact has encouraged researchers to propose alternatives to the structure tensor.

Most of the strategies try to avoid the integration of different orientations of the gradient by adapting the neighborhood to the data in such a way that only neighbors with similar orientations of the gradient are taken into account in the summation. For example, Nagel and Gehrke (1998), and Nath and Palaniappan (2005) use adaptive Gaussians instead of a Gaussian convolution; Köthe (2003) uses a hourglass-shaped kernel instead of the Gaussian (cf. Figure 4.1); van de Weijer and van den Boomgaard (2005) use robust statistics to choose one of the ambiguous orientations at every pixel; Brox et al. (2006), and Hahn and Lee (2009) propose non-linear diffusion processes in order to aggregate contributions of the neighbors. As the structure tensor, all of these strategies assume that the gradient usually changes in a neighborhood slowly.

In a different context, tensor voting can be thought of as a technique to estimate structure in clouds of points. Unlike the strategies described above, tensor voting argues in this context that,

in general, the structure at a point should be different from the structure at a neighboring point. Thus, it defines a methodology to estimate structure based on the hypothesis that neighboring points are connected through smooth surfaces (cf. Section 2.6). This section shows how tensor voting can be adapted in order to estimate structure in images, leading to more accurate and robust structure estimations than the structure tensor. Although tensor voting has already been applied to a variety of problems in image and video processing (cf. Section 2.3.1), as far as we know, this is the first work on the specific scope of (dense) structure estimation.

This section is organized as follows. Subsection 4.1.1 shows the relationships between the structure tensor and tensor voting. Subsection 4.1.2 describes how to apply tensor voting for estimating structure in images. Subsection 4.1.3 shows some results of tensor voting applied to image structure estimation. Finally, Subsection 4.1.4 makes some final remarks.

4.1.1 Relationships Between the Structure Tensor and Tensor Voting

Although the structure tensor and tensor voting are usually applied to two different scopes, images and clouds of points, both aim at estimating structure. This section describes the relationships between the structure tensor and tensor voting.

Similarities

Both, the structure tensor and tensor voting are structurally similar, since not only the first one can be adapted to 3D structure estimation, but also the second can also be adapted to image structure estimation.

On the one hand, the structure tensor can be adapted in order to estimate structure in 3D with the help of a norm estimator, for example, by estimating the equation of the most likely tangent plane at every point with a least squares approximation. The norms obtained with this estimator can be tensorized and convolved with a Gaussian in order to estimate structure in 3D. The resulting tensors yielded by both methods can be analyzed in the same manner. For example, $\lambda_1 - \lambda_2$ can be used as a measure of surfaceness, $\lambda_2 - \lambda_3$ as a measure of edginess, and λ_3 as a measure of junctionness (cf. Section 2.6).

Conversely, tensor voting can also be adapted to image structure estimation by designing an appropriate encoding step. Taking into account that the normal $\mathbf{n}_\mathbf{q}$ in a gray-scale image corresponds to the normalized gradient, $\nabla u_\mathbf{q}/||\nabla u_\mathbf{q}||$, the *stick* component $\mathrm{S}_\mathbf{q}$ of the tensor at $\mathbf{q}$

can be initialized as:

$$S_q = (\lambda_1 - \lambda_2) \frac{\nabla u_q \nabla u_q^T}{||\nabla u_q||^2}, \tag{4.2}$$

which can be further simplified by choosing $(\lambda_1 - \lambda_2) = ||\nabla u_q||^2$. Thus:

$$S_q = \nabla u_q \nabla u_q^T. \tag{4.3}$$

In addition, if the components P_q and B_q are initialized to zero, the input of both the structure tensor and tensor voting becomes equivalent for gray-scale images. As in the 3D case, the output of both methods can be analyzed in a similar way, since, in 2D, the shape of the tensors at edges is closer to a *stick*, while the shape tends to a *ball* at corners in both cases (in 2D, the *plate* component is always zero). However, the tensors obtained by means of tensor voting are in a different scale. Hence, it is necessary to apply a rescaling function in order to have comparable results.

Differences

Despite their similarities, both methods have two essential differences: the rotation term and the restriction of $\theta \leq \pi/4$ of the *stick* tensor voting (cf. Subsection 2.6.1). These differences are given by the different assumptions made by both methods. On the one hand, the hypothesis of tensor voting is that **p** and **q** belong to the same smooth curve and the voting processes are adjusted according to this hypothesis. On the other hand, the hypothesis made by the structure tensor is that the orientation of the gradient at neighboring points should be similar, taking into account that the orientation of the gradient usually changes in space slowly.

These differences can be seen in Figure 4.2. The structure tensor can be modeled as a voting process in which every point votes for its own orientation with a strength given by a Gaussian function. Thus, the structure tensor propagates its own orientation isotropically. In turn, tensor voting propagates a rotated version of the original orientation when $\theta \leq \pi/4$.

It is expected that tensor voting performs better than the structure tensor as it makes stronger assumptions.

4.1.2 Tensor Voting for Structure Estimation

The structural relationships shown in Sect. 4.1.1 lead to a general methodology to improve the image structure estimation by using tensor voting instead of the structure tensor. The method-

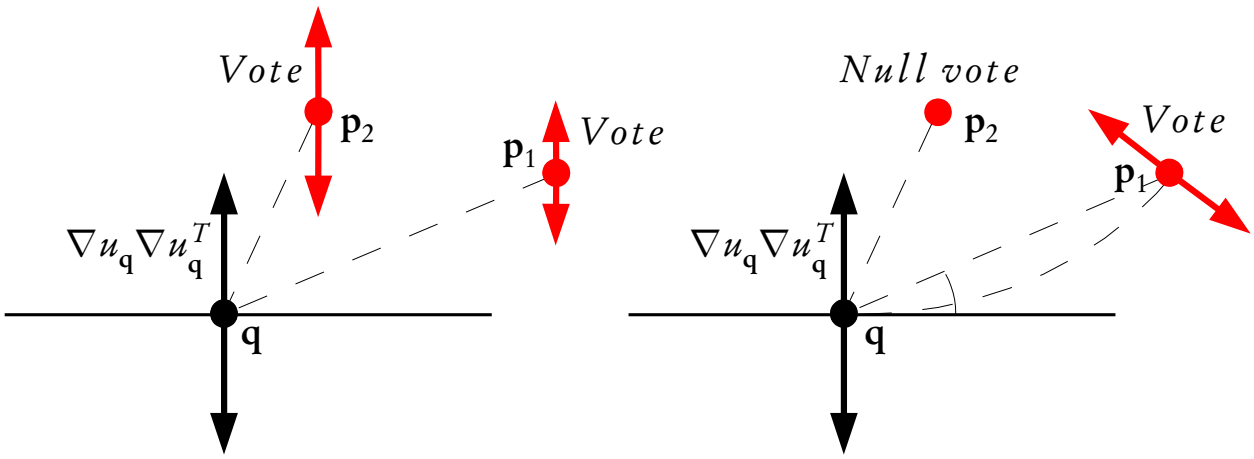


Figure 4.2: Left: the structure tensor seen as a voting process. Right: the *stick* tensor voting. The main differences between both are the rotation term (see the difference of votes at $\mathbf{p}_1$) and the anisotropic behavior of tensor voting (tensor voting does not cast votes to $\mathbf{p}_2$).

ology comprises three steps. First, initialization of tensors in the same way as for the structure tensor. Second, application of tensor voting instead of the Gaussian convolution used in the structure tensor. Finally, rescaling of results in order to renormalize the total energy stored in the tensors. The following subsections show how this methodology can be applied to different types of images.

Gray-Scale Images

Tensor voting can be adapted in order to estimate structure in gray-scale images by following the next 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. It is not necessary to apply the *plate* and *ball* voting processes since the *plate* and *ball* components are zero at every pixel in this case. Thus,

$$\mathrm{TV}(\mathbf{p}) = \sum_{\mathbf{q} \in neigh(\mathbf{p})} \mathrm{SV}(\mathbf{v}, \nabla u_\mathbf{q} \nabla u_\mathbf{q}^T). \tag{4.4}$$

In practice, 3σ can be used to determine the neighborhood of a point $\mathbf{p}$. Finally, the resulting tensors are rescaled by the factor:

$$\xi = \frac{\sum_{\mathbf{p} \in \Omega} trace(\nabla u_\mathbf{p} \nabla u_\mathbf{p}{}^T)}{\sum_{\mathbf{p} \in \Omega} trace(\mathrm{TV}(\mathbf{p}))}, \tag{4.5}$$

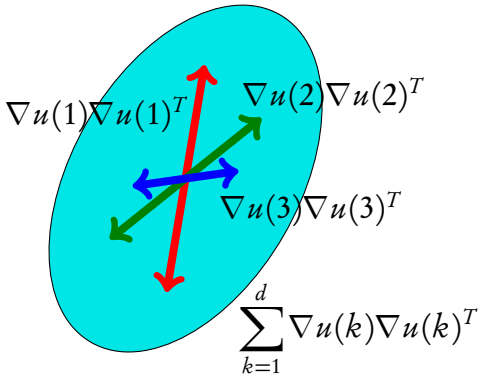


Figure 4.3: Tensor voting can be applied to different channels independently (the red, green and blue *sticks*) or to the sum of the tensorized gradients (the ellipse).

in order to renormalize the total energy of the tensorized gradient, where Ω refers to the given image.

Color and Vector-Valued Images

The structure tensor has already been extended to multivalued images in Weickert (1999b):

$$\mathrm{J} = \sum_{k=1}^{d} G_\sigma * w_k \nabla u(k) \nabla u(k)^T = G_\sigma * \sum_{k=1}^{d} w_k \nabla u(k) \nabla u(k)^T, \tag{4.6}$$

where d is the number of channels, $\nabla u(k)$ is the gradient at channel k, and w_k are weights used to give different relevance to every channel. From (4.6), the structure tensor can be equivalently estimated either by adding d structure tensors, one for every channel, or by applying a Gaussian kernel to the (weighted) summation of the tensorized gradients $\nabla u(k)\nabla u(k)^T$. Thus, there are two options to extend tensor voting for this kind of image, considering that tensor voting must replace the Gaussian convolution used in the structure tensor. The first option is to apply the *stick* tensor voting independently to every channel and then to add the individual results:

$$\mathrm{TV}(\mathbf{p}) = \sum_{k=1}^{d} \sum_{\mathbf{q} \in neigh(\mathbf{p})} w_k \, \mathrm{SV}(\mathbf{v}, \nabla u_\mathbf{q}(k) \nabla u_\mathbf{q}(k)^T). \tag{4.7}$$

The second option is to apply tensor voting to the sum of tensorized gradients with $\mathrm{S_q}$, $\mathrm{P_q}$ and $\mathrm{B_q}$ being the *stick*, *plate* and *ball* components of $\mathrm{T_q} = \sum_{k=1}^{d} w_k \nabla u_\mathbf{q}(k) \nabla u_\mathbf{q}(k)^T$. For two-dimensional images, $\mathrm{P_q} = 0$. In both options, rescaling the calculated tensors is necessary in a similar way as described for the gray-scale images. Figure 4.3 shows the options described above.

The first option has the advantage that only the application of the *stick* tensor voting is necessary, whereas for the second option, the *plate* (for 3D color images) and *ball* tensor voting are

(a) (b) (c) (d)

Figure 4.4: (a) Lenna. (b) Mandrill. (c-d) Pixels (in black) with $\lambda_2 \geq 0.1\ \lambda_1$ of T_q for both images. Processing channels independently is appropriate for most pixels of natural images.

also required, increasing the computational cost of the technique. On the other hand, the second option tends to be more robust since it is less sensitive to bad initial estimations of the gradient. However, in practice, $T_q \approx S_q$ in most pixels of natural images. As an example, in Figure 4.4 the number of pixels with λ_2 greater than the 10% of λ_1 of T_q corresponds to only 0.8% of the total for Lenna and 12.2% for the more textured Mandrill. Thus, the first option can be used in most of the pixels and the second one only in those pixels in which the approximation is not valid.

Tensor-Valued Images

A tensor-valued image is an image in which a tensor is associated with every pixel or voxel. As an example, images acquired through difussion tensor magnetic resonance imaging (DT-MRI) are tensor-valued. Figure 4.5 show an example of this kind of image.

Unlike gray-valued and color images, there are several ways to extend the structure tensor concept to tensor-valued images. One of them was proposed by Weickert and Brox (2002), in which the structure tensor is calculated through (4.6), with the channels corresponding to the entries in the tensors. Thus, the same methodology presented in the previous subsection can be used for adapting tensor voting to tensor-valued images by using the entries in the tensors as the channels of a vector-valued image. Moreover, the factors w_k can be set for tensor-valued images by using the fact that any symmetric matrix, M, can be modeled by means of a vector, m, which is given in an orthonormal tensorial basis with respect to the internal product $\langle A, B \rangle = trace(AB^T)$ (Pajevic et al., 2002; Kindlmann et al., 2007):

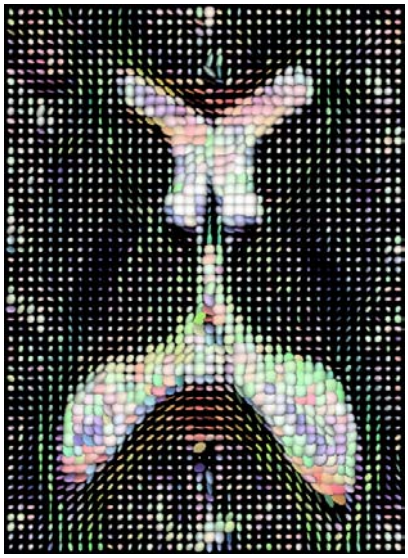


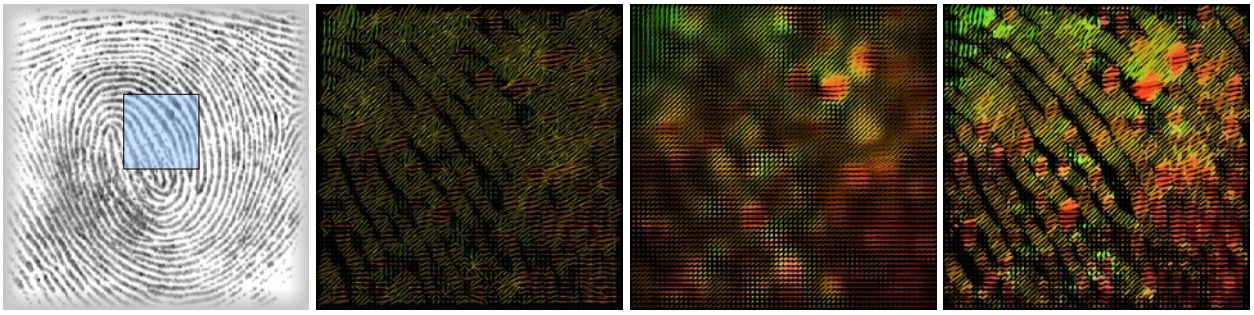
Figure 4.5: Slice of a DT-MRI data set of a human head ($40 \times 55 \times 1$ voxels). Ellipsoids are used to represent the tensors associated with every voxel.

$$\mathrm{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \iff m = \begin{bmatrix} m_{11} \\ \sqrt{2}m_{12} \\ \sqrt{2}m_{13} \\ m_{22} \\ \sqrt{2}m_{23} \\ m_{33} \end{bmatrix}. \tag{4.8}$$

This modeling makes equivalent the Frobenius norm $|\mathrm{M}|_F = \sqrt{trace(\mathrm{M}\mathrm{M}^T)}$ and the norm of m. Thus, tensor voting can be applied to m vectors instead of to M tensors by using the methodology presented in the previous subsection, with $w_k = 1$ for the diagonal entries and $w_k = \sqrt{2}$ for the other entries.

More sophisticated methods have already been proposed for extending the concept of the structure tensor to tensor-valued images. For example, Burgeth et al. (2009) use an algebraic approach to deal with the intrinsic third order nature of the gradient of tensor-valued images. Nevertheless, an adaptation of tensor voting based on these methods requires the extension of the voting processes for higher-dimensional tensor-valued images, which is out of the scope of this chapter.

On the other hand, a *stick* tensor voting can be defined for higher-order tensor-valued images of even order by using the fact that any symmetric higher-order tensor can be decomposed



(a) (b) (c) (d)

Figure 4.6: (a) A fingerprint with a region of interest (ROI).(b) $\nabla u \nabla u^T$ in the ROI. (c-d) The structure tensor and tensor voting in the ROI respectively ($\sigma = 2/\sqrt{2}$). Color indicates orientation of the first eigenvector: green= 0, yellow=$\pi/4$, red=$\pi/2$, blue=$3\pi/4$. Tensor voting preserves gaps.

by means of the generalized eigenvector decomposition (Schultz et al., 2009). The generalized eigenvector decomposition of a higher-order tensor is given by:

$$\mathrm{T} = \sum_{i=1}^{l} \lambda_i \mathbf{e}_i^{\otimes l}, \tag{4.9}$$

where l is the order of the tensor (which is even since the tensor is symmetric), and $\mathbf{e}_i^{\otimes l}$ represents l outer products of $\mathbf{e}_i$ with itself. This decomposition leads to a framework to extend tensor voting to this kind of image. Every pixel casts l *stick* votes with $\mathbf{e}_i \mathbf{e}_i^T$, which are weighted with the corresponding eigenvalues:

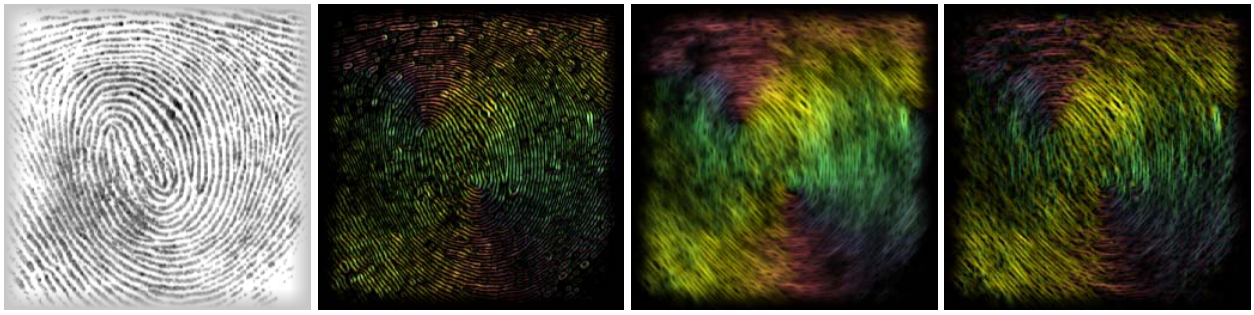
$$\mathrm{TV}(\mathbf{p}) = \sum_{\mathbf{q} \in neigh(\mathbf{p})} \sum_{i=1}^{l} \lambda_i \, \mathrm{SV}(\mathbf{v}, \mathbf{e}_i {\mathbf{e}_i}^T)^{\otimes l/2}. \tag{4.10}$$

This is the approach followed by Schultz (2010).

4.1.3 Experimental Results

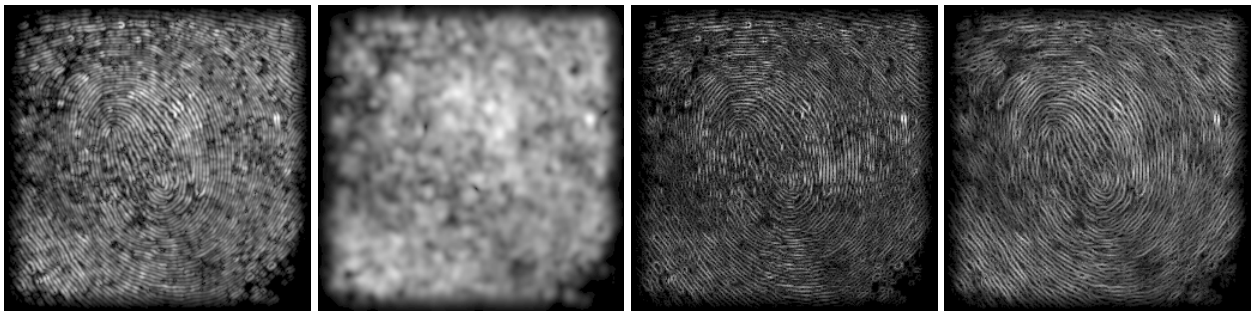
Figures 4.6 to 4.8 present the estimation of structure in a fingerprint by means of both the structure tensor and tensor voting. Figure 4.6 shows that tensor voting is able to preserve the gaps in the image, while the structure tensor is not. This means that tensor voting avoids estimating structure in unstructured regions, which is one of the known problems of the structure tensor.

Figure 4.7 shows that the orientation of the gradient is smoothed by both the structure tensor and tensor voting. This is a good property of a structure estimator, since orientation usually



(a) (b) (c) (d)

Figure 4.7: (a): Original image. (b-d) Color coded orientation (green= 0, yellow=$\pi/4$, red=$\pi/2$, blue=$3\pi/4$) of $\nabla u_q \nabla u_q{}^T$, the structure tensor and tensor voting respectively ($\sigma = 3/\sqrt{2}$). Both methods smooth the orientation of the gradient.



(a) (b) (c) (d)

Figure 4.8: (a-b) Map of $\lambda_1 - \lambda_2$ obtained with the structure tensor for $\sigma = 1/\sqrt{2}$ and $\sigma = 2/\sqrt{2}$ respectively. (c-d) Map of $\lambda_1 - \lambda_2$ obtained with tensor voting for the same values of σ. The structure tensor is more sensitive to σ.

changes slowly in an image and is noisy in $\nabla u \nabla u^T$.

Figure 4.8 shows the map of $\lambda_1 - \lambda_2$, which can be used to extract edges. It can be seen that the structure tensor is more sensitive to the selection of parameter σ, while tensor voting yields similar results for a greater range of values. Thus, it is more difficult to tune the parameter of the structure tensor than the parameter of tensor voting.

Figure 4.9 shows an example for edge detection. Since ideal edges are characterized by *stick* tensors, edges can be obtained by applying non-maximum suppression and hysteresis to the map of $\lambda_1 - \lambda_2$, which measures how far every pixel is from that condition. It can be seen that the structure tensor blurs that map. This can lead to misplacements of the binary edges extracted from these maps and to loss of small edges. For example, edges inside faces are completely lost, and the eyebrow of the totem at the left-hand side is misplaced. Tensor voting is able to keep

(a) (b) (c)

Figure 4.9: (a) Original image. (b-c) Map of $\lambda_1 - \lambda_2$ for the structure tensor and tensor voting respectively ($\sigma = 3/\sqrt{2}$). The structure tensor blurs the edges.

edges thinner, reducing in that way the problems of the structure tensor.

Most corner detectors apply a function onto the eigenvalues of the structure tensor (Kenney et al., 2005). Hence, accuracy and robustness in the estimation of eigenvalues are requirements for this application. Figures 4.10 and 4.11 show plots of λ_1 and λ_2 from tensors estimated by means of both the structure tensor and tensor voting for a noiseless and a noisy synthetic image respectively. Figure 4.10 shows that tensor voting is more robust and more accurate than the structure tensor in the estimation of λ_1. In addition, the structure tensor wrongly introduces a maximum in λ_1 in the middle of the small hole inside the star, while tensor voting does not.

Figure 4.11 shows that the structure tensor has a bad performance for both noiseless and noisy images. Actually, it blurs λ_2 in such a way that the corners are displaced. In addition, it is very sensitive to noise and generates a false maximum in the hole at the middle of the star. On the other hand, tensor voting has a more consistent performance in estimating λ_2 in both noiseless and noisy images. Although tensor voting generates a halo near edges, it can be filtered out by taking into account that it only appears near edges and has smaller values of λ_2 than in the corners.

Figure 4.12 shows that tensor voting is also a better option than the structure tensor for tensor-valued images. This figure shows the results yielded for the image of Figure 4.5 by both

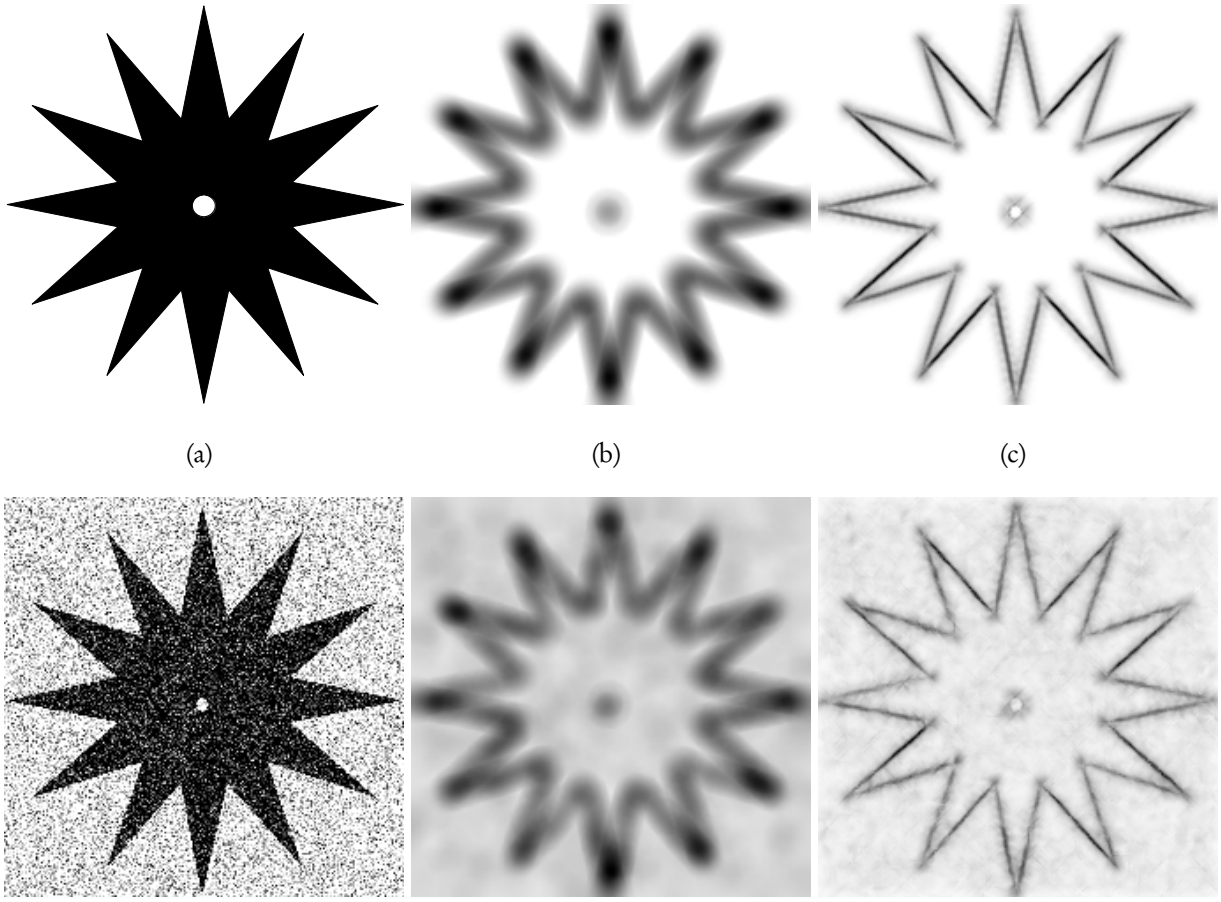


Figure 4.10: (a) Original image. (b-c) Maps of λ_1 obtained with the structure tensor and tensor voting respectively for the original image ($\sigma = 3/\sqrt{2}$). (d) Noisy image (truncated Gaussian noise with standard deviation of 100). (e-f) Maps of λ_1 obtained with the structure tensor and tensor voting respectively for the noisy image ($\sigma = 3/\sqrt{2}$). Tensor voting has a better performance in precision and robustness than the structure tensor.

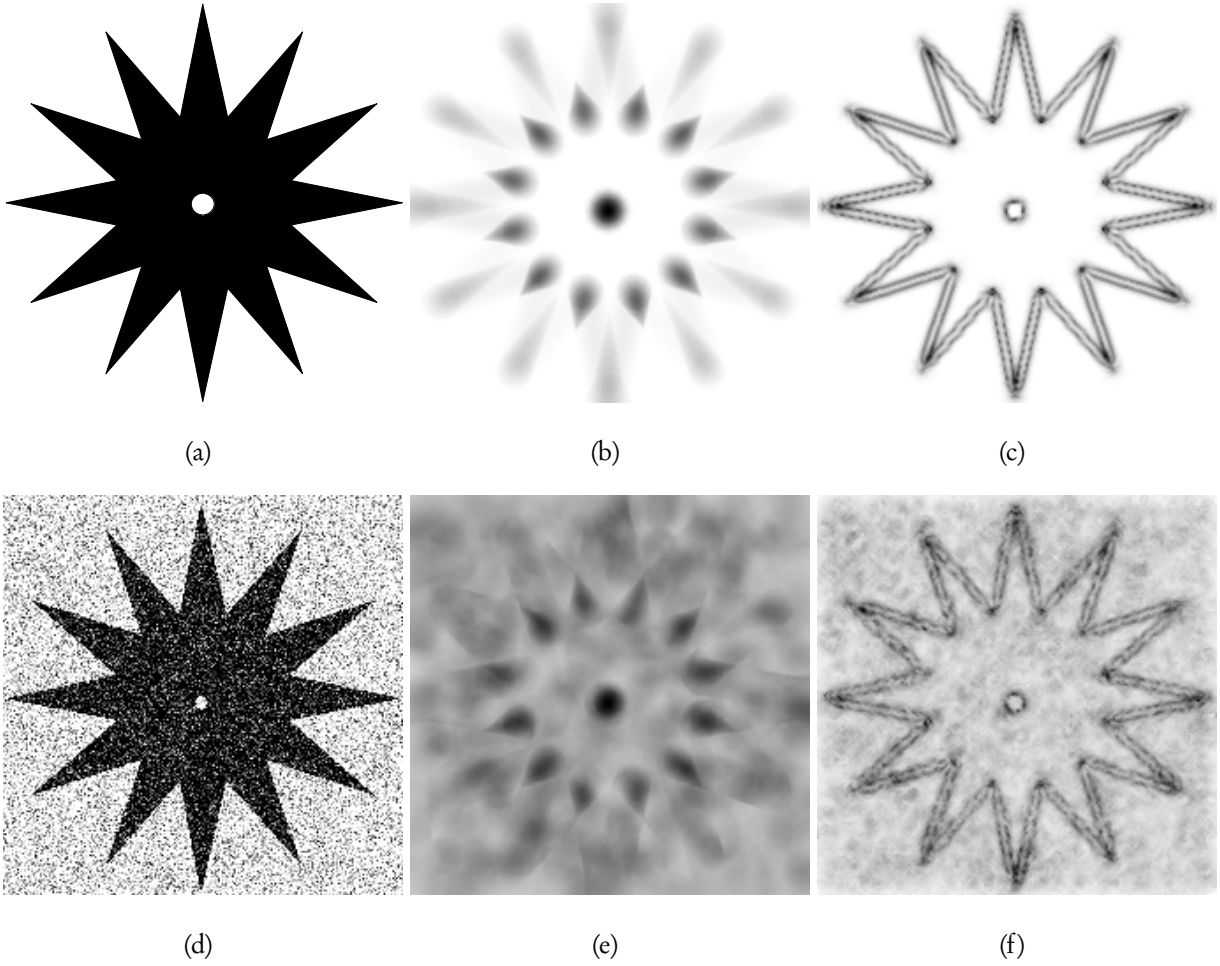
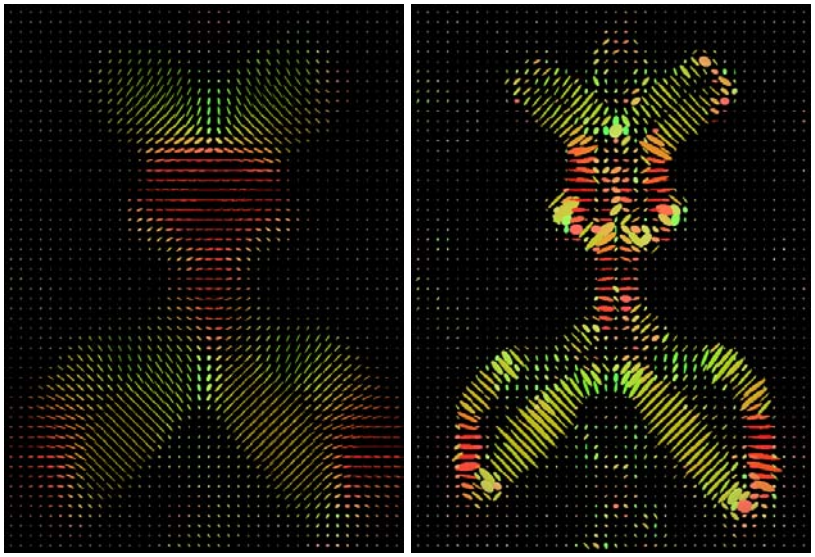


Figure 4.11: (a) Original image. (b-c) Maps of λ_2 obtained with the structure tensor and tensor voting respectively for the original image ($\sigma = 3/\sqrt{2}$). (d) Noisy image (truncated Gaussian noise with standard deviation of 100). (e-f) Maps of λ_2 obtained with the structure tensor and tensor voting respectively for the noisy image ($\sigma = 3/\sqrt{2}$).

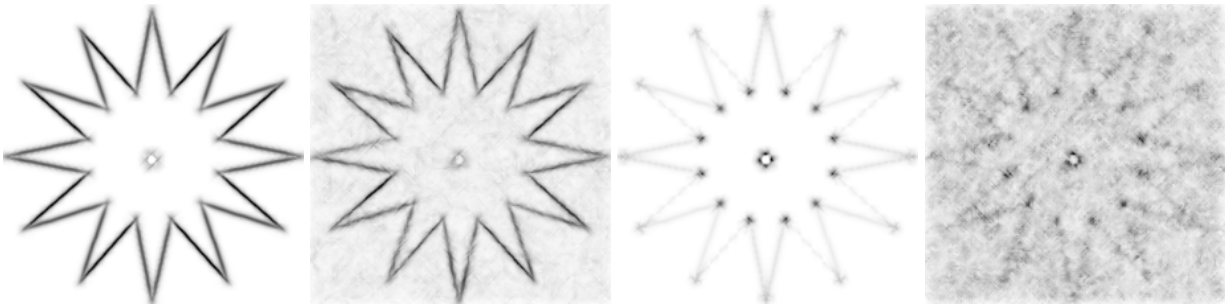


(a) (b)

Figure 4.12: (a-b) Resulting tensor fields after applying the structure tensor and tensor voting respectively ($\rho = 5/\sqrt{2}$) for the image of Figure 4.5. Ellipsoids are used to represent the tensors associated with every voxel.

the structure tensor and tensor voting calculated through (4.7) for the six channels of (4.8). It can be seen that the structure tensor blurs the resulting tensors in such a way that it is difficult to extract edges and corners from them. On the contrary, tensor voting is able to extract structure in a better way.

An additional experiment aims at evaluating the effect of the rotation term of the *stick* tensor voting. Figure 4.13 shows plots of λ_1 and λ_2 from tensors estimated by means of tensor voting without the rotation term for the star images of Figs. 4.10 and 4.11. It can be seen that the effect of the rotation term on λ_1 is almost negligible, since the results are similar for both the noiseless and noisy images (see Figure 4.10c vs. Figure 4.13a, and Figure 4.10f vs. Figure 4.13b). As for λ_2, tensor voting without the rotation term has a better performance in the noiseless image, since it does not produce halos (see Figure 4.13c). However, its performance is not robust, since it is difficult to extract maxima from its estimation for the noisy image (see Figure 4.13d). Thus, tensor voting with the rotation term is more robust in the estimation of λ_2. In conclusion, the rotation term of the *stick* tensor voting robustifies the estimation of λ_2 at a cost of introducing halos that should be filtered out a posteriori.



(a) (b) (c) (d)

Figure 4.13: (a-b) Maps of λ_1 calculated with tensor voting without the rotation term of the *stick* tensor voting for the images of Figure 4.10a and Figure 4.10d respectively ($\sigma = 3/\sqrt{2}$). (c-d) Maps of λ_2 calculated with tensor voting without the rotation term for the *stick* tensor voting for the images of Figure 4.10a and Figure 4.10d respectively ($\sigma = 3/\sqrt{2}$).

4.1.4 Summary

This section has proposed a general methodology to adapt tensor voting for estimating image structure based on the fact that the *stick* tensor voting and the structure tensor are structurally similar. This new methodology has been applied to different types of images. Experimental results have shown that tensor voting can estimate structure more appropriately than the structure tensor. In addition, tensor voting yields more robust estimations of structure than the structure tensor. Furthermore, experiments have shown that the rotation term of the *stick* tensor voting leads to more robust estimations of λ_2 but also generates halos that should be filtered out a posteriori.

4.2 Segmentation of Images Acquired Through Stereo Vision

Segmentation is one of the most important applications in computer vision as a preliminary step towards object recognition. Its goal is to partition a given image into a set of non-overlapping homogeneous regions that likely correspond to the different objects or geometric structures that may be perceived in the scene. In theory, any segmenter of range or 3D boundary images (see below) can be used for images acquired through stereo vision. However, the high presence of noise in this kind of image have made it difficult to apply traditional segmentation techniques in this scope.

The section is organized as follows. The next subsection introduces the problem and reviews

previous related work. Subsection 4.2.2 describes the proposed algorithm. Subsection 4.2.3 shows some experimental results. Finally, Subsection 4.2.4 make some concluding remarks on the proposed method.

4.2.1 Motivation

There are two types of images that convey 3D information: range images and 3D images. Range images are 2D images with depth information at every pixel, whereas real 3D images model the exact geometry of objects. In turn, a 3D boundary image is a 3D image that only stores 3D information from points lying on the boundary of the objects present in the image. Both, range and 3D boundary images can be modeled through 3D clouds of points. In this context, an image acquired through stereo vision can be seen as a coupled range/color image, where both color and depth information are available at every pixel.

The technologies to acquire range images can be divided into two groups: active and passive, depending on whether or not the sensor projects some kind of energy onto the scene. A more comprehensive taxonomy of methods was proposed by Curless (1997). One of the most popular families of active methods is based on measuring the time of flight. This method estimates distances by emitting signals from the sensor to the obstacles and then, by measuring the elapsed time until receiving the reflection of those signals. SONAR (SOund Navigation And Ranging), infrared sensors, RADAR (RAdio Detection And Ranging) and LADAR (LAser Detection And Ranging), also known as LIDAR (LIgth Detection and Ranging), belong to this category. Another popular active method is structured light. Using a camera and a source of light with known patterns, depth appears as distortions of those patterns in the image. The main disadvantage of this method is the need for a controlled environment, this being a constraint that is not always acceptable in some applications. A complete study of the latter strategy is presented by Rusinkiewicz (2001). The advantages of active techniques include their high accuracy, even for relatively far distances, but their most important drawback is their cost.

Alternatively, passive methods aim at emulating the human process of image acquisition. The most common passive method is stereo vision. In this approach, two or more pictures are taken at the same time from different positions in order to reconstruct the scene through projective geometry and estimation of disparities (Hartley and Zisserman, 2004). The main advantages of stereo vision are that it does not require any alteration of the scene, can directly obtain coupled

range/color images, and is a low-cost technology. Stereo matching is a very active research area as it is shown in the web page http://vision.middlebury.edu/stereo/ that ranks the state-of-the-art methods in stereo matching by using the methodology proposed by Scharstein and Szeliski (2002). The major drawback of stereo vision is its comparative low quality of results derived from its problems in non-textured regions and the high amount of noise present in data, especially for distant regions.

In theory, any segmenter of clouds of points, such as those proposed for range or 3D boundary images, can be used for images acquired through stereo vision, since all of them can be modeled through clouds of points. On the one hand, different approaches have been proposed for segmenting range images (e.g., Hoover et al., 1996, reviews seventeen methods). However these methods have only been applied to images acquired through active techniques (which commonly contain low amounts of noise), since they are usually unable to deal with the noise present in images acquired through stereo vision.

On the other hand, the approaches for segmenting 3D boundary images mainly follow two strategies: mesh and point segmentation. The first approach generates triangular meshes from the cloud of points before segmenting the objects (e.g., see Chen et al., 2009; Shamir, 2008, for up-to-date surveys of segmenters that follow this strategy). Unfortunately, mesh segmenters cannot be used in images acquired through stereo vision, since the mesh creation step is very sensitive to noise. The point segmenters aim at extracting regions from raw data. Although in the last few years some algorithms following this approach have been proposed (e.g., Golovinskiy and Funkhouser, 2009; Rabbania et al., 2006), they still face problems to deal with noise. A related strategy tries to segment images by extracting reconstructed surfaces. However, the main problem of these techniques is mainly related to their efficiency. The method proposed by Medioni et al. (2000) follows this approach. It applies tensor voting and a variant of the marching cubes algorithm (Lorensen and Cline, 1987) in order to recover the shape of surfaces, edges and junctions present in a given cloud of points. Finally, segmentation can be done from the extracted surfaces. Unlike other related methods, this method has been proven more robust even for images constituted by strongly noisy clouds of points.

This section presents an approach for segmenting images acquired through stereo vision. The proposed algorithm can be seen as a modification of the method by Medioni et al. (2000) in which a fast graph-based segmenter is used instead of the expensive marching cubes algorithm. Thus, regions can be extracted more efficiently with the new proposed algorithm at a cost of not

extracting the shape of the surfaces, which is not required in many applications. Unlike previous works in segmentation of stereo pairs, the proposed method explores the possibility of using the cloud of points, instead of the left and right images (as, for instance, Triantafyllidis et al., 2000, do), as the input of the algorithm without restricting the segmented objects to be planes (as, for example, Thakoor et al., 2008a,b, do). The following subsection describes the proposed method.

4.2.2 Overview of the Algorithm

The proposed algorithm consists of an iterative procedure in which a segmentation is obtained at every iteration. Each segmentation is coarser than the ones obtained in previous iterations, creating in this way a set of segmentations at different perceptual abstraction levels.

The algorithm has the following steps: first, the neighborhood size is estimated, this being a necessary parameter for the rest of the process. Afterwards, an iterative procedure is run with four stages being executed at every iteration: (a) filtering, (b) tensor voting, (c) graph creation, and (d) graph segmentation. Every iteration leads to a segmentation of the input image at a progressively higher abstraction level. This procedure is run a given number of times m.

The algorithm yields the set of all calculated segmentations. That set can be used in applications where it is interesting to obtain different levels of detail in different zones of the image. Although the number of regions has a decreasing trend as iterations increase, the segmentation set does not constitute a pyramid. One of the reasons is the merging effect: a region can be temporarily divided into two regions in the process of being joined with another. In the end, the generated segmentations can be seen as a pseudo-pyramid segmentation.

The following subsections present the filtering, graph creation and graph segmentation steps, since tensor voting has already been presented in Section 2.6.

Geometric Low-pass Filters

As discussed above, images acquired through stereo vision are very noisy. Hence, it is necessary to apply a geometric low-pass filter in order to remove that noise. Local filters are preferred to global ones since every different region in an image has its own specific features, in particular, point density and amount of noise. Local filtering algorithms are based on local processing of a neighborhood, $neigh(\mathbf{p})$, around each 3D point $\mathbf{p}$ belonging to the given cloud of points.

Local filters based on either averaging or function-based averaging (e.g., using a Gaussian

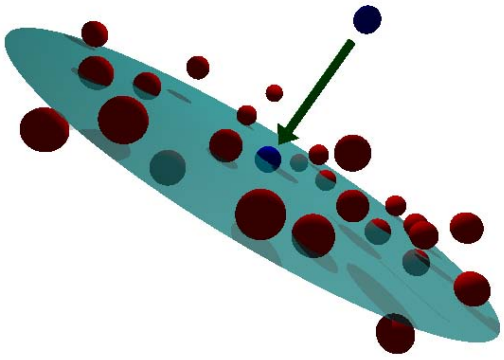


Figure 4.14: MLS-Projection. The noisy point (in blue) is filtered by projecting it onto the closest plane to its neighbors (in red).

function to give more weight to nearest neighbors) have been discarded as they do not work well with clouds of points with variable density of points, such as the ones obtained through stereo vision. The application of that kind of filter to those clouds of points would lead to regions with high densities of points becoming even denser and to low density regions becoming more scattered, thus creating holes in the image or making previous holes bigger.

An alternative way to filter points is to apply 2D linear regression by projecting those points onto the plane (or onto another desired surface) that minimizes the squared error in the neighborhood. A Gaussian can be used as a weighting function in order to give more importance to nearest neighbors in the least squares calculations. The process is illustrated in Figure 4.14. This technique is known as Moving Least Squares Projection (MLS-Projection), and it has been found useful in both computer vision and computer graphics (Levin, 1998; Fleishman et al., 2005; Kolluri, 2008; Lipman, 2009).

The MLS-Projection filter prevents the point density modification problem described above since movements towards a specific area of neighbors is avoided. The parameters of this filter are the standard deviation of the weighting Gaussian function, σ, and the number of neighbors that constitute a neighborhood, ρ. The proposed segmenter uses MLS-projection for filtering with a plane as the reference surface and a Gaussian as a weighting function.

Graph Creation and Segmentation

After applying tensor voting, every 3D point from the given cloud of points is associated with three saliency measures that denote the geometric structure (surface, edge or junction) to which

the point likely belongs to. The goal now is to group neighboring points that likely belong to the same geometric structure. For that purpose, the graph-based segmenter proposed by Felzenswalb and Huttenlocher (2004) has been applied, since it has been proven to be fast and to provide good results in image segmentation. Although that segmenter was originally conceived for intensity image segmentation, it can be easily extended to segmentation of clouds of points as its starting point is a graph that can be created in a variety of ways.

Graph Creation Given the saliencies obtained in the tensor voting step, a graph is created as follows. Let ϵ be a small constant, σ be the standard deviation used in the filtering and tensor voting stages and the following predicate:

$$\mathbf{q} \in neighbors(\mathbf{p}) \Leftrightarrow e^{-\frac{\|\mathbf{p}-\mathbf{q}\|^2}{\sigma^2}} > \epsilon. \tag{4.11}$$

A graph is built using the following two rules: (a) every point in the cloud defines a graph's vertex, (b) the vertices corresponding to any pair of 3D points $\mathbf{p}$ and $\mathbf{q}$ belonging to the cloud are connected through an edge provided that $\mathbf{q} \in neighbors(\mathbf{p})$.

Every edge connecting neighboring points $\mathbf{p}$ and $\mathbf{q}$ is associated with a weight w defined according to the following method: let $\mathbf{e}_i$ be the i-th eigenvector of the tensors calculated through tensor voting, $\hat{s_i}$ be the i-th normalized saliency of the tensors, which are calculated through $\hat{s_i} = s_i/\lambda_1$ (cf. Section 2.6), and $d_i(\mathbf{p},\mathbf{q}) = 1 - \frac{2}{\pi}\arccos(|\mathbf{e}_i(\mathbf{p})^T\,\mathbf{e}_i(\mathbf{q})|)$ be one minus the normalized angle between the ith eigenvectors of $\mathbf{p}$ and $\mathbf{q}$. Then:

$$w = 1 - (\hat{s_1}(\mathbf{p})\,\hat{s_1}(\mathbf{q})\,d_1(\mathbf{p},\mathbf{q}) + \hat{s_2}(\mathbf{p})\,\hat{s_2}(\mathbf{q})\,d_3(\mathbf{p},\mathbf{q}) + \hat{s_3}(\mathbf{p})\,\hat{s_3}(\mathbf{q})). \tag{4.12}$$

In this way, this method uses all the available saliencies and angles. The constraint $0 \leq w \leq 1$ is guaranteed since $0 \leq \hat{s_i} \leq 1$. Although any tensor dissimilarity measurement can be used instead of (4.12), for example, any of those presented by Peeters et al. (2009), this methodology has yielded good results in our experiments. Alternatively, the formula suggested by Julià et al. (2010) to be used instead of d_i, which applies an additional exponential decaying function to the deviation angles, can also be used to control the maximum allowed curvature of surfaces in the final segmentation.

Graph-Based Segmentation The graph segmentation technique by Felzenswalb and Huttenlocher (2004) is based on a region-growing approach whose success basically depends on the graph creation step and on parameter k (cf. Section 7.2 for a detailed description of this method).

Due to the noisy nature of our input data, a post-processing stage to avoid oversegmentation has been designed. The basic idea consists of iterating the same segmenter in a hierarchical way. This process is described in Algorithm 1 where "CreateUpperGraph" creates a new graph based on the previous segmentation, with its vertices being the regions found in the previous segmentation and its edges being created using the method described before, but only using a single tensor that represents all the tensors in the region.

Algorithm 1 Segmentation Algorithm

1: **function** RESEGMENT(V_0, E_0) ▷ Vertices and Edges
2: $S_0 \leftarrow Segment(V_0, E_0)$
3: **repeat**
4: $[V_{i+1}, E_{i+1}] \leftarrow CreateUpperGraph(V_i, E_i, S_i)$
5: $S_{i+1} \leftarrow Segment(V_{i+1}, E_{i+1})$
6: **until** $NumSegments(S_{i+1}) = NumSegments(S_i)$
7: **return** S_n
8: **end function**

The tensor of a region is computed by adding the tensors of the elements that constitute that region and normalizing the sum of saliencies. Edges between regions are only calculated if in the previous iteration there is an edge connecting an element from a region to another element from the other region. This prevents the resulting segmentation in an iteration from having more regions than the segmentations obtained in previous iterations.

Neighborhood Selection

In order to apply the filtering, tensor voting and graph creation stages, it is necessary to define the size of the neighborhood associated with any given point from the input cloud. Let ρ be the mean cardinality of $neigh(\cdot)$, which can be estimated from other parameters associated with the algorithm as follows. Let $\mathbf{R}$ be a set of points randomly sampled from the input cloud of points, and σ be the standard deviation of the Gaussian decaying function used in the filtering and tensor voting stages. Parameter ρ can be estimated as the mean quantity of neighbors at a distance of 3σ from all points $\mathbf{p} \in \mathbf{R}$. With this, only those points that give relevant information to their neighbors in the filtering and tensor voting stages are taken into account.

The filtering, tensor voting and graph creation stages use an ANN k-d tree (Arya et al.,

Figure 4.15: Digiclops trinocular stereo vision camera.

1998) with 3D Euclidean metric to retrieve the neighborhood of a given 3D point, using the aforementioned parameter ρ.

4.2.3 Experimental Results

A Digiclops trinocular stereo vision camera (see Figure 4.15) and the Digiclops SDK and Triclops SDK libraries developed by Point Grey Research Inc. have been used to capture the coupled color/range images upon which the proposed technique has been tested, in both indoor and outdoor conditions. The error of the estimated depth measurements, z, depends on the depth itself, so points farther away than five meters are discarded, since their error is above $8.32cm$ (Point Grey Research Inc.). Points at less than a meter have also been discarded since range estimation at those distances is not possible.

The aforementioned libraries only calculate depth when it is possible to do it with a reasonable error. For example, Figure 4.16a shows a picture taken by the right camera. Figure 4.16b depicts in black all pixels too close to or too far away from the camera or whose estimated depth could not be calculated. In this example, the libraries were able to estimate depth for 45,589 pixels from a total of 76,800, but those numbers can change dramatically (upwards or downwards) in other examples as stereo algorithms are very sensitive to scene conditions. The ground-truth segmentations were calculated by hand using 2D information.

The proposed method has been compared with a vector-based technique that uses the same filtering, graph creation and segmenter processes in order to assess the use of tensors in the proposed algorithm. Edges' weights for the vector-based technique have been calculated through a simplified version of (4.12):

$$w = \frac{2}{\pi} \arccos(|\mathbf{n}_p^T \mathbf{n}_q|), \tag{4.13}$$

(a) Right Image (b) Registered Points (c) 3D View

Figure 4.16: Indoor example of a coupled color/range image

where $\mathbf{n}_p$ and $\mathbf{n}_q$ are estimated normals at points $\mathbf{p}$ and $\mathbf{q}$, which have been calculated through MLS-Projection with a Gaussian weighting function.

The segmentation results yielded by both methods with $\sigma = 20mm$ (filtering and tensor voting parameter) and $k = 0.2$ (segmenter parameter) after five iterations are shown in Figure 4.17. The first row shows the annotated ground-truth for both scenes, and the second and third rows the segmentation results of the tensor and vector-based methods respectively. Only relevant regions (with more than 20 pixels) have been plotted in that figure. It can be seen that the tensor-based approach yields more appealing results with a segmentation closer to the ground-truth and with more relevant regions. A point to take into account is that obtaining a perfect segmentation is not possible in general, since the stereo algorithm of the libraries generates misplacing errors near big 3D discontinuities and distant zones of the scene.

The $OSEG$ and $USEG$ measures have been used to assess performance of the tested methods (cf. Subsection 7.3.3). $OSEG$ is a measure to assess oversegmentation either locally or globally. Roughly speaking, local $OSEG$ measures the degree of fragmentation of a specific region of the ground-truth caused by the segmenter. Global $OSEG$ is computed as a weighted mean of local measures of $OSEG$. In turn, $USEG$ is a measure to assess undersegmentation, also locally or globally. Local $USEG$ measures the degree of aggregation of regions of the ground-truth in a specific segmented region. Similarly, global $USEG$ can be computed as a weighted mean of local measures of $USEG$. Both $OSEG$ and $USEG$ range between zero and one, where one means a perfect performance. The reader is refered to Subsection 7.3.3 on Page 145 for implementation details of these measures.

Figure 4.18 shows in gray the oversegmentation and undersegmentation local measures calcu-

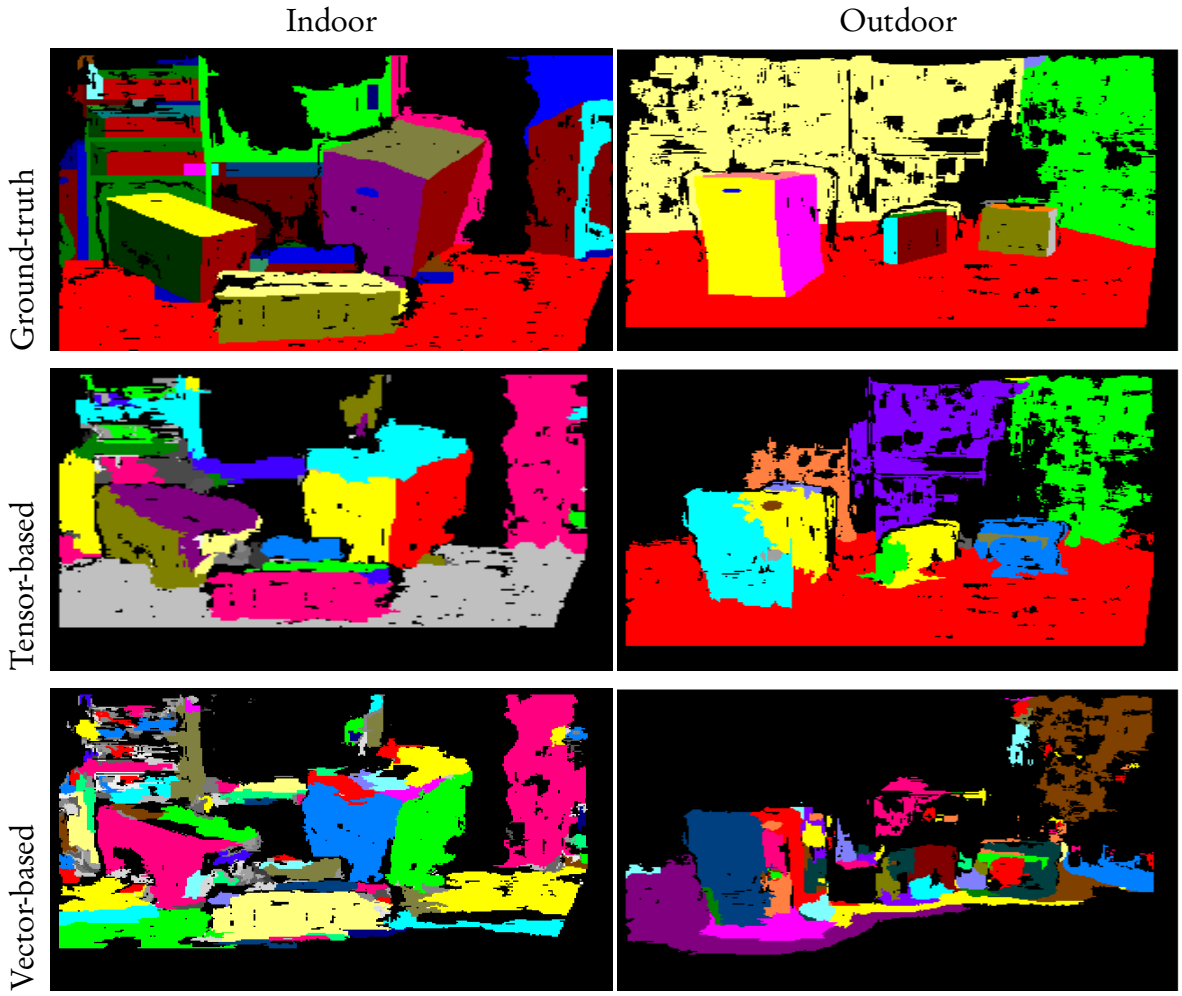


Figure 4.17: Segmentation results

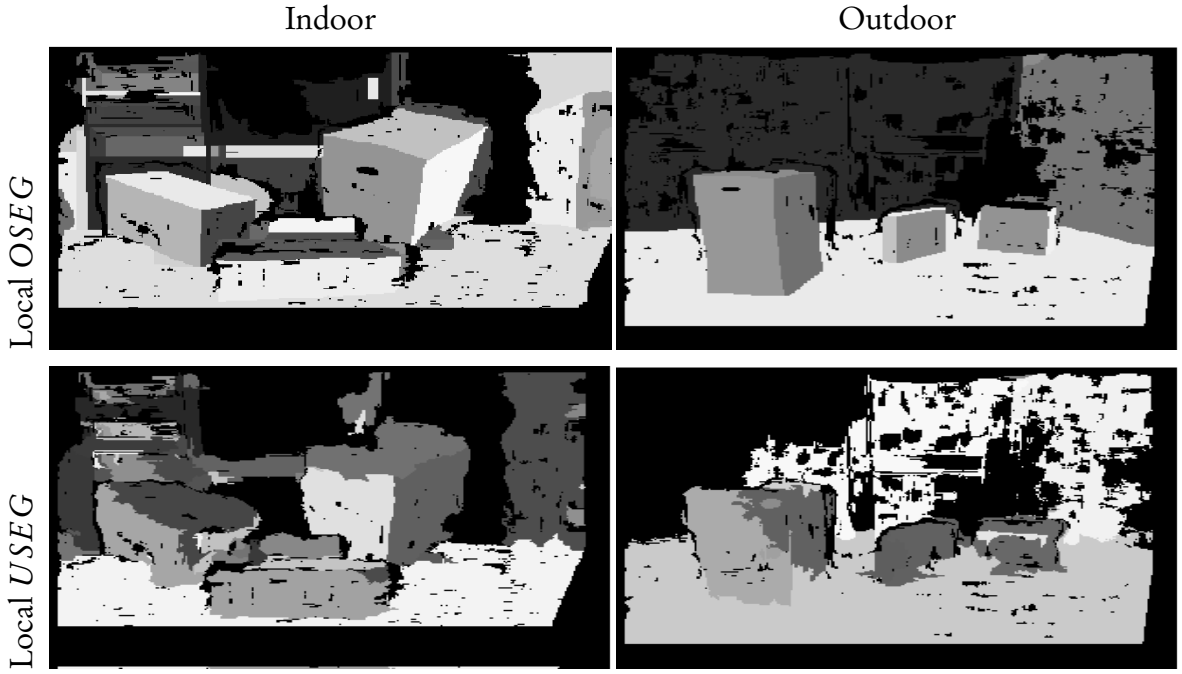


Figure 4.18: Over- and undersegmentation measurements for the tensor-based segmenter. The top panels show the ground-truth regions. The gray value of each region is proportional to its local $OSEG$. The bottom panel shows the regions produced by the segmenter. The gray level of each region is proportional to its local $USEG$. White indicates a perfect performance.

lated on the results of the tensor-based approach for each region. In addition, Table 4.1 shows the over- and undersegmentation global measures calculated on the results of both methods. These measures have been calculated for all the scene (A), for the 50% nearest points to the camera (N) and for the 50% farthest points from the camera (F).

As expected, the best results for oversegmentation were obtained by using the tensor-based approach. The global oversegmentation measure is around 0.40 (0.33 and 0.47), which means that, on average, a region in the ground-truth is segmented approximately into 1.66 regions by

Table 4.1: Oversegmentation and undersegmentation results

Method	Indoor Scene							Outdoor Scene						
	# Reg.	*OSEG*			*USEG*			# Reg.	*OSEG*			*USEG*		
		A	N	F	A	N	F		A	N	F	A	N	F
Tensor-based	151	0.33	0.11	0.56	0.39	0.10	0.62	215	0.47	0.17	0.74	0.17	0.02	0.31
Vector-based	497	0.66	0.44	0.90	0.31	0.05	0.54	681	0.79	0.62	0.92	0.10	0.00	0.21
Ground-truth	52							15						

the algorithm (cf. Subsection 7.3.3). However, it is necessary to remark that the segmentation results were suitable in zones near the camera, where accuracy is better (0.14 on average). For far points, this measure is around 0.65 (0.56 and 0.74), that is, a region in the ground-truth is divided on average into almost three regions, which is not a bad result taking into account the distribution of points and noise present in those zones. As for the undersegmentation, the vector-based method produced slightly better results driven by the high quantity of generated regions.

4.2.4 Summary

This section has presented a graph-based algorithm aimed at segmenting images acquired through stereo vision. Although the proposed algorithm obtains good results, its performance decreases in regions where stereo vision yields a wrong estimation of depth, which is common in regions with big discontinuities in z. However, these problems are expected to disappear in the near future as researchers in stereo matching are very active in proposing more and more accurate and efficient algorithms. Experimental results have also shown that the use of tensors improves the performance of the segmenter in these noisy scenarios.

Part II

EXTENSIONS OF TENSOR VOTING

Chapter 5

Tensor Voting for Edge-Preserving Color Image Denoising

Color image denoising is an important task in computer vision and image processing, as images acquired through color image sensors are usually contaminated by noise. Color image denoising algorithms can be directly used for image restoration and other higher-level tasks as a pre-processing step. The main goal of color image denoising is to suppress noise from color images while preserving their features, such as meaningful edges or texture details, as much as possible. A color image denoising algorithm is called *edge-preserving* when it is able to accomplish this goal. Liu et al. (2008) have identified the following general features that an effective, edge-preserving color image denoising algorithm must fulfill: noise must be completely removed from flat regions; edges, texture details and global contrast must be preserved; and no artifacts must appear in the result.

Designing effective, edge-preserving color image denoising algorithms is a difficult task that can be evidenced by the fact that the majority of denoising algorithms introduce undesirable blurring and/or artifacts in the filtered images. The main reason for this difficulty is that, without any other assumptions, no color image denoising algorithm can utterly comply with all the aforementioned features listed by Liu et al. (2008). This is mainly due to two reasons: the complete reconstruction of the original image from one contaminated by noise is not possible in general, and some of those features are nearly contradictory. For example, distinguishing between noise and texture is an open problem.

Two main approaches have been followed in color image denoising: spatial domain and

transform-domain filtering. The first approach filters the input image by using the color information of every pixel and its neighbors. The major problem of these filters is their tendency to blur the images. The second approach transforms the input image to a different space, typically to the wavelet domain, filters the transformed image and applies the inverse transformation to the result. Despite its good edge preservation properties, the major criticism to transform-based denoising algorithms is the introduction of undesirable artifacts. Section 5.1 presents a brief review of both approaches.

Among the techniques based on perceptual grouping, tensor voting appears to be one of the most appropriate for edge-preserving color image denoising, since it was designed as a generic framework that can be adapted to a variety of applications well beyond the ones which it was originally applied to. Although, the classical tensor voting is a good technique for image structure estimation, as it was shown in Section 4.1, at a first glance, it seems impractical for image denoising. This can be better shown with an example. Figure 5.1 shows six different initializations for the tensors of a sparse image, and the results after applying a few iterations of tensor voting. It can be seen that no matter the input, tensor voting converges to the same result after a few iterations. This result corresponds to the structure of the image: red or green *stick* tensors in flat regions or *ball* tensors in junctions. This property, of interest for the applications shown in Chapter 4, is an issue for image denoising, since the output of the (iterated) classical tensor voting does not depend on the encoding step but, on the spatial distribution of pixels (or points).

Despite this, tensor voting can still be adapted to image denoising. On the one hand, the input information of the image denoising problem must be either modeled in terms of the surface reconstruction problem, which refers to extracting surfaces, edges and junctions from a set of noisy points, or encoded into tensors through a different encoding process. This second alternative is likely to be more advantageous since the new encoding process can be specifically tailored to the problem requirements. On the other hand, the validity of the hypotheses made by the voting processes presented in Section 2.6 and Chapter 3 on the input data must be established, since the assumptions on which they are based may no longer be valid in a context not related to surface reconstruction. Actually, this is the case of image denoising, where even in ideal conditions (i.e., without noise), color can change abruptly. This suggests that the use of the canonical voting fields may not be the best option for image denoising.

This chapter proposes a new solution to the problem of edge-preserving color image denoising based on an adaptation of the classical tensor voting in order to properly handle color infor-

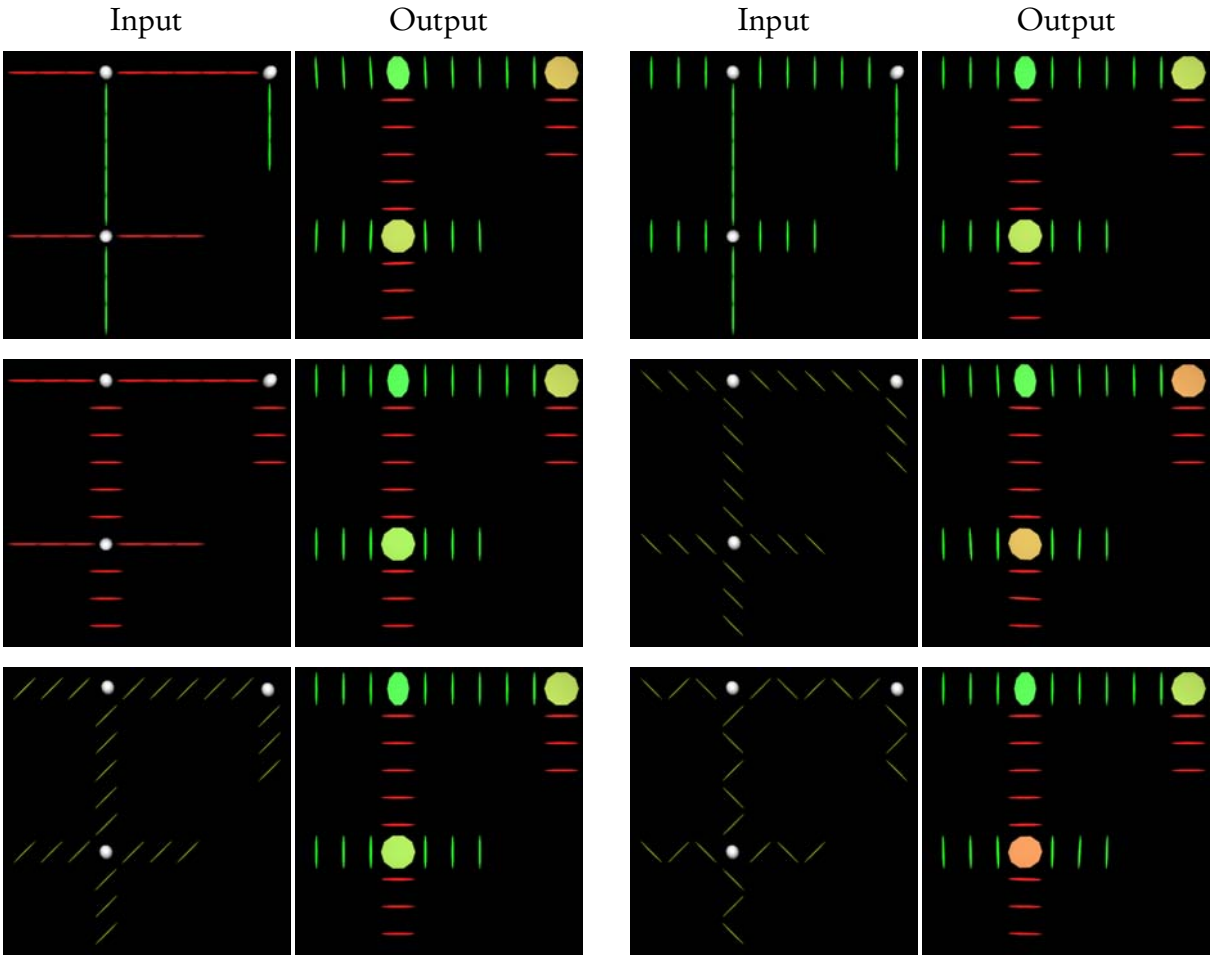


Figure 5.1: Iterated tensor voting. The first and third columns show six different initializations of the tensors before applying tensor voting to a sparse image (black regions do not contain pixels). There are only pixels where tensors are shown. Each pixel is associated with a tensor. Tensors have been initialized either with *stick* tensors (green, red or yellow *sticks*) or small *ball* tensors (in white). Columns two and four show the resulting tensors after applying a few iterations of tensor voting (fewer than five).

mation. First, an encoding process specifically designed to encode color, uniformity and *edginess* into tensors is presented. Second, a voting process specifically tailored to the edge-preserving color image denoising problem is also introduced. This voting process is based on the nature of the encoded information and on a set of criteria inspired by the perceptual process of image denoising.

This chapter is organized as follows. Section 5.1 describes previous related work. Section 5.2 presents the criteria taken into account in the design of the algorithm proposed in this chapter. Sections 5.3 and 5.4 detail the adaptation of the classical tensor voting to edge-preserving color image denoising. Section 5.6 shows a comparative analysis of the proposed method against some of the state-of-the-art, edge-preserving, color image denoising algorithms by using the quality measures described in Section 5.5. Finally, Section 5.7 discusses the obtained results and makes some final remarks.

5.1 Previous Related Work

Two main color image denoising approaches have been followed: spatial domain filtering and transform-domain filtering. Classical filters, such as mean, median or Gaussian filters (Gonzalez and Woods, 2007), bilateral filtering (Tomasi and Manduchi, 1998), non local means (Buades et al., 2005, 2008), anisotropic diffusion (Perona and Malik, 1990) and Bayesian inference (Roth and Black, 2009; Barbu, 2009; Geman and Geman, 1984), among many others, follow the spatial domain filtering approach. Classical filters are simple, efficient and easy to implement. However, they frequently blur the filtered images and/or eliminate important details. The bilateral filter extends the concept of Gaussian filtering by adding a Gaussian weighting function that depends on the difference between pixel intensities. This filter is also efficient and easy to implement. However, it is unable to filter very noisy images. Non-local means (NLM) extends bilateral filtering by taking into account differences between pixel neighborhoods instead of pixel intensities. NLM is effective for image denoising and it is considered to belong to the state-of-the-art. However, it tends to generate undesirable quantization effects in edgeless regions. Filters based on anisotropic diffusion give more weight to neighbors located in the directions where edges are not present. Anisotropic diffusion usually models the filtering problem by means of partial differential equations (PDEs) (e.g. Perona and Malik (1990); Tschumperlé (2006)), although the use of graph theory has also been proposed (Lezoray et al., 2007). Anisotropic diffusion

has been a successful approach, with many methods based on it belonging to the state-of-the-art (e.g., Lezoray et al., 2007; Tschumperlé, 2006). Techniques based on anisotropic diffusion are able to suppress noise effectively. However, they also tend to create artifacts at edges and have problems with very noisy images. Bayesian-based approaches are usually highly time consuming, or require training steps that are not allowed in many applications (e.g., Roth and Black, 2009; Barbu, 2009; Zhu and Mumford, 1997). A different successful strategy that follows the spatial domain filtering approach uses conditional random fields to detect and remove noise (Liu et al., 2008). However, its main drawback is that it is highly time consuming. A different spatial domain approach applies evolutionary computation (Lukac et al., 2006). However, its scope of use is limited, since it requires a training stage.

The most popular technique within the transform-domain filtering approach is based on wavelets (Donoho and Johnstone, 1994). Basically, small coefficients of the wavelet transform of the input image are removed before applying the inverse transformation, since they are usually due to noise. Many adaptations of this principle have been proposed in the literature. For example, Gaussian scale mixtures (Portilla et al., 2003; Miller and Kingsbury, 2008) , hidden Markov models (Ichir and Mohammad-Djafari, 2006) or optimal color space projection (Lian et al., 2006). In spite of their good edge preservation properties—some of these methods are considered to belong to the state-of-the-art—the major criticism to wavelet-based denoising algorithms is the introduction of undesirable artifacts in the images. Other approaches that filter images in a transform-domain include Wiener filters (Gonzalez and Woods, 2007), low pass filters using the Fast Fourier Transform (Gonzalez and Woods, 2007) or methods based on *blind image separation*, which tries to separate two original signals (noise and signal in image denoising) from their addition (e.g., Tonazzini et al., 2006). However, these approaches have been outperformed by other strategies. More recently, Yu et al. (2009) intended to take advantage of both transform-domain and spatial domain approaches for image denoising. However, they found that their method, which is based on wavelet-based filtering and the bilateral filter, is not satisfactory to deal with real noise.

Perceptual grouping has previously been applied to color image denoising, especially in the spatial domain. For example, Ben-Shahar and Zucker (2004) detect and remove color noise by using the perceptual grouping principle of good continuation, in taking advantage of the fact that color hue changes smoothly in most natural images. In addition, tensor voting has also been used for denoising as discussed below in this section.

Previous studies have applied tensor voting to color information mainly following two strategies. A first strategy applies tensor voting to the color components directly. For example, Kang and Medioni (2001) segment color images by encoding the position and RGB color of every pixel into tensors of five dimensions before applying tensor voting. Although this strategy uses all the color information available in the input image, it has shown limitations on noisy images. A second strategy converts color information to a simplified representation before applying tensor voting. In this direction, Massad et al. (2003) extract salient edges and junctions from gray-scale images by applying tensor voting to local edge orientation encoded through 2D tensors. Jia and Tang (2004) reconstruct damaged images by using tensor voting on $(n \times (n+1))$-dimensional tensors constructed from the gray-scale value of the $n \times n$ neighbors of every pixel plus the maximum value of them. Tai et al. (2006a) use the color gradient and local statistics in order to increase the resolution of images. More recently, Lim et al. (2007) extract text from color images by applying tensor voting on 3D tensors created from the pixel's position (row and column) and a single value calculated from its HSI color components. The results of those schemes based on the second strategy have shown that tensor voting can be successful with color information. However, these schemes may discard important information since only a part of the available color information is used.

To the best of our knowledge, only Tai et al. (2006b) have used tensor voting in the specific area of image denoising. First, they classify every pixel as an *edge* or a *region* pixel by using local statistics. Second, they apply tensor voting to the *edge* pixels in order to extract edges. Third, they define a neighborhood for each *region* pixel by using multiscale analysis and by excluding those neighbors that are separated from the *region* pixel by one of the edges extracted after the second step. Finally, the color of every pixel is calculated as the weighted mean of colors in its neighborhood. This approach has two important drawbacks: first, it depends of an initial classification of pixels that is not conducted in a robust way. Thus, the performance of the algorithm is likely to decrease in very noisy images where the number of pixels initially classified as edges increases. Second, using the weighted mean can lead to loss of texture and to quantization artifacts.

These drawbacks are mainly the consequence of using a robust technique, such as the classical tensor voting, in a single step of the whole process, with the other steps being based on non-robust techniques. Thus, the complete process could be improved by replacing those non-robust techniques by robust ones. The classical tensor voting is such a robust technique. However, the

canonical voting fields used in the classical tensor voting are not appropriate for color information as stated above. In this context, instead of using different robust techniques, this chapter explores the alternative of extending the classical tensor voting to the image denoising problem in the spatial domain so that it can be at the core of the denoising process, avoiding in such a way the drawbacks of the method by Tai et al. (2006b). This alternative has the additional advantage that the appropriateness of tensor voting for image denoising can be fully established, since the results cannot be attributed to any other supporting technique.

Previous works have proposed two different encoding of color through tensors. On the one hand, Rittner et al. (2010) represent the three channels of the HSL model through a single 2D tensor in which hue, saturation and luminance are encoded through the orientation, shape and size of the tensor respectively. Unfortunately, this representation cannot be used for tensor voting since tensors defined in such a way cannot be added, which is essential for this method. On the other hand, some works have used third order tensors to encode color patches (e.g. Wen et al., 2010; Muti and Bourennane, 2007; Tao et al., 2005; He et al., 2005). However, this encoding faces three difficulties: the encoding tensors are non-positive semidefinite in general, the perceptual interpretation of them is not clear, and the summation operation on them is not defined. Thus, since tensor voting requires positive semidefinite tensors, in addition to a summation operation, and its voting step has been devised to propagate perceptual information, they cannot be used for tensor voting either. Hence, a new encoding of color is proposed in this chapter (cf. Subsection 5.3.1).

5.2 Perceptual Criteria for Propagating Local Information in Color Image Denoising

The first step necessary to design a voting process for adapting tensor voting to a specific application is to have a reasonable description of how local information is propagated to the neighbors in that application. The canonical voting fields proposed by Guy and Medioni (1996, 1997) were designed to propagate the encoded information by using the plausible hypothesis that, in general, normal vectors tend to smoothly change over surfaces. However, those canonical voting fields are not the right option for color information, since color does not follow this property. This fact makes necessary the definition of a new voting process more appropriate for color image

denoising. This voting process should be inspired by the perceptual process carried out by the human visual system in order to perform image denoising. However, this is a difficult task, since that process depends on many factors, such as background and surrounding color, viewing distance, texture, amount and type of noise present in the image, presence of edges or local contrast, among many others (Fairchild, 2005; Hunt, 2004).

In this chapter, a set of intuitive criteria inspired by the human perceptual process of image denoising is used in order to obtain good results while keeping a reasonable complexity of the proposed algorithm. These criteria are based on the perceptual grouping laws of similarity and proximity, and constitute the foundation of the voting process proposed in this chapter (cf. Section 5.3). Three main perceptual features are involved in the definition of these criteria, namely: perceptual color difference, uniformity and edginess. Perceptual color difference aims at measuring how similar every pair of colors appears to a human. Uniformity measures how variable color appears in a specific noiseless region. Edginess aims at measuring the likelihood of finding edges or texture in a specific region. Edginess in edge-preserving image denoising is not only related to the presence of edges, but also to the presence of texture, since texture should also be preserved.

The following criteria related to perceptual color differences are taken into account. On the one hand, in the absence of other clues, a small perceptual color difference between neighboring pixels should be mainly attributed to noise. In this case, the hypothesis that both pixels should have the same color becomes plausible. In this situation, edginess is usually low, while the uniformity of the region where the pixels are located is usually high. On the other hand, in the absence of other clues, a big perceptual color difference between neighboring pixels cannot be attributed exclusively to noise, since edges or texture can also generate this type of differences. Thus, other clues are necessary to decide whether to filter these pixels or not. However, if the region is noiseless, edginess is usually high, while uniformity is usually low. A special case is due to impulse noise that appears in pixels with a high perceptual color difference with respect to all their neighbors. Pixels with impulse noise should absorb the color, uniformity and edginess of their neighbors. Impulse noise must be taken into account since it may appear in imaging sensors (cf. Section 2.4).

The following criteria related to uniformity are taken into account. On the one hand, a perceptual color difference between neighboring pixels should be mainly attributed to noise when there is evidence that both pixels belong to the same uniform region. In this case, the hypothesis

that both pixels should have the same color becomes plausible, disregarding the perceptual color difference between them. In this situation, edginess is usually low. On the other hand, other clues are necessary to decide whether to filter or not the pixels of regions with low uniformity, since this feature is also common in regions with edges or texture. However, if the region is noiseless, its edginess and perceptual color differences are usually high.

The following criteria related to edginess are taken into account. On the one hand, other clues are necessary to decide whether to filter or not the pixels of regions with high edginess, since, although this is a specific feature of regions with edges or texture, eventually, it can also be found in noisy regions. However, if the region is noiseless, its uniformity will usually be low. On the other hand, regions with low edginess should be treated similarly to uniform regions. Hence, their pixels should be filtered. In this case, uniformity will also be high.

Other general criteria are also important. First, the influence of pixels on their neighbors should depend on the distance. Thus, closer neighbors should be more affected by a pixel than farther ones. Second, if uniformity and/or edginess at a pixel in a color channel appear too different from those in the other color channels, that difference could be caused by noise and, although not always being the case, the presence of noise in this pixel becomes more likely.

5.3 Tensor Voting for Edge-Preserving Color Image Denoising

The input of the proposed method is the set of pixels of a noisy color image. Thus, positional and color information is available for every input pixel. Positional information is used to determine the neighborhood of every pixel, while color information is used to define the tensors in the encoding step. It is well known that color can be represented by a variety of color models whose selection depends on the particular application, since each model has a specific scope of use. CIELAB is the most appropriate color model for the proposed color image denoising approach, since some criteria described in the previous section are based on the estimation of perceptual color differences and CIELAB was designed to measure this kind of difference (cf. Section 2.2).

Before applying the proposed method, every CIELAB channel is normalized in the range $[0, \pi/2]$. As an example, normalization factors of $\frac{\pi}{200}$ for channel L and $\frac{\pi}{2 \times 255}$ for channels a and b are appropriate for outdoor scenarios, since the CIELAB channels are in the ranges

$0 \leq L \leq 100$, $-127 \leq a \leq 128$ and $-127 \leq b \leq 128$ respectively, when color is converted from RGB for these types of scenarios[1]. An additional shift of 127 for channels a and b is also necessary before normalization in order to avoid negative values. Obviously, this normalization process must be adjusted for other types of scenarios.

The next subsections describe the details of the proposed edge-preserving color image denoising method.

5.3.1 Encoding of Color Information

The encoding of color information must be in compliance with the perceptual criteria mentioned in Section 5.2. Thus, not only color must be encoded, but also uniformity and edginess. This objective can be carried out by means of tensors. In the first step of the method, the color 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 channel of the pixel (of every specific channel), a measure of local uniformity, and an estimation of edginess at the pixel's location. Figure 5.2 shows the graphical interpretation of a tensor for channel L. The normalized color channel is encoded as the angle α 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, while a tensor with $\alpha = \frac{\pi}{2}$ encodes white. Other values of α allow the tensors in channel L to encode other possible luminance levels in the range from black to 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 every color channel. In contrast, a pixel located at an ideal edge is represented by means of three *ball tensors*, one for every color channel.

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 α between the x axis and the principal eigenvector as described above. However, since measures of

[1]by using the standard illuminant D65 (which emulates the illuminant effect of the sun) and a two degrees observer (i.e., the subtended angle in the observer's retina by an individual color stimulus is two degrees), which are more appropriate in applications where individual color stimuli are small (Fairchild, 2005).

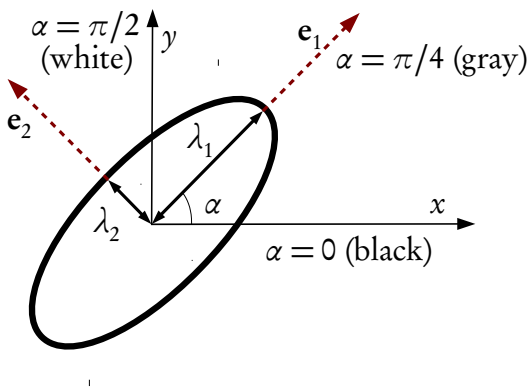


Figure 5.2: Encoding process for channel L. Color, uniformity and edginess are encoded by means of α and the normalized $\hat{s_1} = (\lambda_1 - \lambda_2)/\lambda_1$ and $\hat{s_2} = \lambda_2/\lambda_1$ saliencies respectively. Something similar is done for the a and b channels.

uniformity and edginess are usually not available at the beginning of the process, normalized saliency $\hat{s_1}$ is initialized to one and normalized saliency $\hat{s_2}$ is initialized to zero. Hence, the initial color information is encoded through *stick* tensors oriented along the directions that represent the original color of the pixel given in the normalized CIELAB channels. This initialization is carried out by:

$$\mathrm{T}_c(\mathbf{p}) = \mathbf{t}_c(\mathbf{p})\, \mathbf{t}_c(\mathbf{p})^T, \tag{5.1}$$

with:

$$\mathbf{t}_c(\mathbf{p}) = \left[\cos\left(\mathrm{C}_c(\mathbf{p})\right) \quad \sin\left(\mathrm{C}_c(\mathbf{p})\right)\right]^T, \tag{5.2}$$

where $\mathrm{T}_c(\mathbf{p})$ is the tensor of the c-th color channel (L, a and b) at pixel $\mathbf{p}$ and $\mathrm{C}_c(\mathbf{p})$ is the normalized value of the c-th color channel at $\mathbf{p}$.

Splitting color information into as many tensors as color channels is advantageous since the processing necessary to be applied to every channel may be different. For example, noise only present in some of the channels could be detected and eliminated. However, since the color channels are not independent in general, information from a channel must be used to process the others. Both facts are taken into account in the design of the voting process.

5.3.2 Voting Process

In the second step of tensor voting, 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). A difference with the surface reconstruction problem is that it is not possible to apply the canonical voting fields for the image denoising problem, since a pixel

cannot appropriately propagate its information to its neighbors without taking into account the local relations between the information at that pixel and at its neighbors. This fact is evidenced in the majority of criteria presented in Section 5.2. Hence, specially designed tensorial functions, referred to as *propagation functions*, must be used instead of the canonical voting fields proposed by Guy and Medioni (1996, 1997). These propagation functions must take into account not only the information encoded in the tensors but also the local relations between neighbors.

Two propagation functions are proposed for applying tensor voting to color information: a *stick* and a *ball* propagation function. A vote is the result of applying a propagation function from $\mathbf{q}$ (the voter) to $\mathbf{p}$ (the voted). Hence, the application of the first function leads to *stick* votes, while the application of the second function produces *ball* votes. A *stick* vote can be seen as a *stick* tensor with a strength modulated by scalar factors, whereas the *ball* tensor can be seen as a circumference-shaped tensor also weighted by appropriate scalar factors. The *stick* vote is used to propagate a specific color, while the *ball* propagation function is used to increase edginess by voting for all possible colors. The proposed voting process at every pixel is carried out by adding all the tensors propagated towards it from its neighbors by applying the proposed propagation functions. Thus, the total vote received at a pixel $\mathbf{p}$ for each color channel c, $\mathrm{TV}_c(\mathbf{p})$, is given by:

$$\mathrm{TV}_c(\mathbf{p}) = \sum_{\mathbf{q} \in neigh(\mathbf{p})} \mathrm{S}_c(\mathbf{p},\mathbf{q}) + \mathrm{B}_c(\mathbf{p},\mathbf{q}), \tag{5.3}$$

where $\mathrm{S}_c(\mathbf{p},\mathbf{q})$ and $\mathrm{B}_c(\mathbf{p},\mathbf{q})$ are the *stick* and *ball* propagation functions respectively. Section 5.4 presents these *stick* and *ball* propagation functions specifically tailored to color image denoising.

After applying the voting process, it is necessary to obtain eigenvectors and eigenvalues of $\mathrm{TV}_L(\mathbf{p})$, $\mathrm{TV}_a(\mathbf{p})$ and $\mathrm{TV}_b(\mathbf{p})$ at every pixel $\mathbf{p}$ in order to analyze its local perceptual information. Tensor voting defines a standard way to interpret the voting results: uniformity increases with the normalized $\hat{s_1}$ saliency and the likelihood that a point belongs to an edge increases as the normalized $\hat{s_2}$ saliency becomes greater than the normalized $\hat{s_1}$ saliency. Additionally, the most likely noiseless normalized color channel at a pixel is given for each color channel by the angle between the first eigenvector of the corresponding tensor and the x axis. These three angles are then used to correct the color of every pixel with the most likely noiseless one, reducing in such a way the noise of the image.

5.4 Propagation Functions for Image Denoising

This section presents the propagation functions specifically designed to color image denoising by taking into account the criteria described in Section 5.2.

5.4.1 Modeling of Variables Involved in the Voting Process

Four variables associated with local relations between neighbors are taken into account in the proposed propagation functions: the distance between pixels, the perceptual color difference, the joint uniformity measurement and the likelihood of a pixel being impulse noise. First, the Euclidean distance is used to measure the distance between pixels. Second, the perceptual color difference between pixels $\mathbf{p}$ and $\mathbf{q}$, $\Delta E(\mathbf{p},\mathbf{q})$, is calculated through CIEDE2000 (cf. Section 2.2). It is also necessary to estimate the perceptual color difference in a specific channel c, $\Delta E_c(\mathbf{p},\mathbf{q})$. This value is also calculated by means of CIEDE2000 by setting to zero the difference in all channels different from c. Third, since the uniformity of a region (cf. Section 5.2) that contains two pixels (the voter and the voted) cannot be calculated directly from their tensors (the normalized $\hat{s}_1$ saliency at $\mathbf{p}$ cannot be used directly, since it can only encode a local measurement of the uniformity at the region surrounding a pixel), a highly related measure is used instead: the joint uniformity measurement, $U_c(\mathbf{p},\mathbf{q})$, which is the product of the normalized $\hat{s}_1$ saliencies of both pixels, that is, the product of the local uniformity measurements. By definition, $U_c(\mathbf{p},\mathbf{q})$ varies in the range between zero (completely non-uniform) and one (completely uniform). Thus, let $\hat{s_1}_c(\mathbf{p})$ be the normalized s_1 saliency at $\mathbf{p}$ in channel c (L, a and b). The joint uniformity of $\mathbf{p}$ and $\mathbf{q}$ in channel c, $U_c(\mathbf{p},\mathbf{q})$, can be estimated by:

$$U_c(\mathbf{p},\mathbf{q}) = \hat{s_1}_c(\mathbf{p})\ \hat{s_1}_c(\mathbf{q}). \tag{5.4}$$

Finally, the likelihood of a pixel being impulse noise, $\eta_c(\mathbf{p})$, can be estimated as the difference of normalized $\hat{s}_2$ saliencies between the pixel and its neighbors for those pixels located at local maxima of the normalized $\hat{s}_2$ saliency. Thus, let $\hat{s_2}_c(\mathbf{p})$ be the normalized $\hat{s}_2$ saliency at $\mathbf{p}$ in channel c (L, a and b) and, $\mu_{\hat{s_2}_c}(\mathbf{p})$ be the mean of the normalized $\hat{s}_2$ saliencies in the 8-neighborhood of pixel $\mathbf{p}$ in channel c. The likelihood of $\mathbf{p}$ being impulse noise in channel c, $\eta_c(\mathbf{p})$, is given by:

$$\eta_c(\mathbf{p}) = \begin{cases} \hat{s_2}_c(\mathbf{p}) - \mu_{\hat{s_2}_c}(\mathbf{p}), & \text{if } \mathbf{p} \text{ is at a local} \\ & \text{maximum of } \hat{s_2}_c(\cdot) \\ 0, & \text{otherwise.} \end{cases} \tag{5.5}$$

5.4.2 Design of the *Stick* and *Ball* Propagation Functions

The proposed *stick* propagation function, $\mathrm{S}_c(\mathbf{p},\mathbf{q})$, which allows a pixel $\mathbf{q}$ to cast a *stick* vote to a neighboring pixel $\mathbf{p}$ for channel c is given by:

$$\mathrm{S}_c(\mathbf{p},\mathbf{q}) = GS(\mathbf{p},\mathbf{q})\ \overline{\eta_c}(\mathbf{q})\ SV'_c(\mathbf{p},\mathbf{q})\ \mathrm{ST}_c(\mathbf{q}), \tag{5.6}$$

with $\mathrm{ST}_c(\mathbf{q})$, $GS(\mathbf{p},\mathbf{q})$, $\overline{\eta_c}(\mathbf{q})$ and $SV'_c(\mathbf{p},\mathbf{q})$ being defined as follows.

First, the tensor $\mathrm{ST}_c(\mathbf{q})$ in (5.6) must encode the normalized color channel at $\mathbf{q}$, since *stick* votes cast by a pixel $\mathbf{q}$ are used to propagate its color to its neighbors. Thus, $\mathrm{ST}_c(\mathbf{q})$ is defined as the tensorized eigenvector corresponding to the largest eigenvalue of the voter pixel, that is $\mathrm{ST}_c(\mathbf{q}) = \mathbf{e}_{1c}(\mathbf{q})\ \mathbf{e}_{1c}(\mathbf{q})^T$, with $\mathbf{e}_{1c}(\mathbf{q})$ being the eigenvector with the largest eigenvalue of the tensor associated with channel c at $\mathbf{q}$.

Second, the three scalar factors in (5.6), 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. The *stick* vote strength cast by closer neighboring pixels must be greater than by farther ones. Thus, $GS(\mathbf{p},\mathbf{q})$ is defined as $GS(\mathbf{p},\mathbf{q}) = G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)$, where $G_{\sigma_s}(\cdot)$ is a decaying Gaussian function with zero mean and a user-defined standard deviation σ_s. The second factor $\overline{\eta_c}(\mathbf{q})$, defined as $\overline{\eta_c}(\mathbf{q}) = 1-\eta_c(\mathbf{q})$, is introduced in order to prevent a pixel $\mathbf{q}$ previously classified as impulse noise from propagating its information. This factor makes the vote to be zero when $\mathbf{q}$ is completely noisy and leaves the vote unaffected when $\mathbf{q}$ has not been classified as impulse noise. The third factor, SV'_c, takes into account the influence of the perceptual color difference, the uniformity and the noisiness of the voted pixel. This factor is given by:

$$SV'_c(\mathbf{p},\mathbf{q}) = \overline{\eta_c}(\mathbf{p})\ SV_c(\mathbf{p},\mathbf{q}) + \eta_c(\mathbf{p}), \tag{5.7}$$

where:

$$SV_c(\mathbf{p},\mathbf{q}) = \frac{G_{\sigma_d}(\Delta E(\mathbf{p},\mathbf{q})) + U_c(\mathbf{p},\mathbf{q})}{2}, \tag{5.8}$$

and $\overline{\eta_c}(\mathbf{p}) = 1-\eta_c(\mathbf{p})$.

$SV_c(\mathbf{p},\mathbf{q})$ models the fact that a pixel $\mathbf{q}$ must cast a stronger *stick* vote to $\mathbf{p}$ either if both pixels belong to the same uniform region or if the perceptual color difference between them is small. The joint uniformity measurement, $U_c(\mathbf{p},\mathbf{q})$, is used to determine if both pixels belong to the same region or not. A Gaussian function with zero mean and a user-defined standard deviation σ_d, $G_{\sigma_d}(\cdot)$, which decays with $\Delta E(\mathbf{p},\mathbf{q})$ is used to determine if the perceptual color difference

is small or not. A normalizing factor of two is used in order to make $SV_c(\mathbf{p},\mathbf{q})$ to vary from zero to one. The term $\eta_c(\mathbf{p})$ included in (5.7) makes noisy voted pixels, $\mathbf{p}$, to adopt the color of their voting neighbors, $\mathbf{q}$, disregarding local uniformity measurements and perceptual color differences between $\mathbf{p}$ and $\mathbf{q}$. The term $\overline{\eta_c}(\mathbf{p})$, also included in (5.7), makes SV'_c to vary from zero to one. As expected, the effect of $\eta_c(\mathbf{p})$ and $\overline{\eta_c}(\mathbf{p})$ on the strength of the *stick* vote received at a noiseless pixel $\mathbf{p}$ is null.

In turn, the *ball* propagation function, $\mathrm{B}_c(\mathbf{p},\mathbf{q})$, which allows a pixel $\mathbf{q}$ to cast a *ball* vote to a neighboring pixel $\mathbf{p}$ for channel c is given by:

$$\mathrm{B}_c(\mathbf{p},\mathbf{q}) = GS(\mathbf{p},\mathbf{q})\,\overline{\eta_c}(\mathbf{q})\,BV_c(\mathbf{p},\mathbf{q})\,\mathrm{BT}(\mathbf{q}), \tag{5.9}$$

with $\mathrm{BT}_c(\mathbf{q})$, $GS(\mathbf{p},\mathbf{q})$, $\overline{\eta_c}(\mathbf{q})$ and $BV_c(\mathbf{p},\mathbf{q})$ being defined as follows.

First, the *ball tensor*, represented by the identity matrix, $\mathbf{I}$, is the only possible tensor for $\mathrm{BT}(\mathbf{q})$, since it is the only tensor that complies with 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. The first restriction is based on the fact that the best way to increase edginess at a pixel is to cast *stick* votes for all possible colors. The second restriction avoids undesirable color biases. Thus, $\mathrm{BT}(\mathbf{q}) = \mathbf{I}$.

Second, the strength of the *ball* vote is modulated by three scalar factors, each varying between zero and one, described as follows. The first and second factors, $GS(\mathbf{p},\mathbf{q})$ and $\overline{\eta_c}(\mathbf{q})$, are the same as the ones introduced in (5.6) for the *stick* propagation function. They are included for similar reasons to those given in the definition of the *stick* propagation function. The third scalar term in (5.9), $BV_c(\mathbf{p},\mathbf{q})$, is given by:

$$BV_c(\mathbf{p},\mathbf{q}) = \frac{\overline{G_{\sigma_d}}(\Delta E(\mathbf{p},\mathbf{q})) + \overline{G_{\sigma_d}}(\Delta E_c(\mathbf{p},\mathbf{q})) + \overline{U_c}(\mathbf{p},\mathbf{q})}{3}, \tag{5.10}$$

where $\overline{G_{\sigma_d}}(\cdot) = 1 - G_{\sigma_d}(\cdot)$ and $\overline{U_c}(\mathbf{p},\mathbf{q}) = 1 - U_c(\mathbf{p},\mathbf{q})$.

$BV_c(\mathbf{p},\mathbf{q})$ models the fact that a pixel $\mathbf{q}$ must reinforce the edginess at the voted pixel $\mathbf{p}$ 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 the terms $\overline{G_{\sigma_d}}(\Delta E(\mathbf{p},\mathbf{q}))$ and $\overline{U_c}(\mathbf{p},\mathbf{q})$. The additional term $\overline{G_{\sigma_d}}(\Delta E_c(\mathbf{p},\mathbf{q}))$ is introduced in order to increase the edginess of pixels in which the only noisy channel is c. Thus, those pixels $\mathbf{p}$ with noise in a single color channel receive stronger *ball* votes in that channel. This decreases the strength of the *stick* votes cast

by those noisy pixels in next iterations. In addition, this also allows the method to eliminate noise from those pixels in the following iterations more easily, since $\eta_c(\mathbf{p})$ tends to increase. The normalizing factor of three in (5.10) allows the *ball* propagation function to cast *ball* votes with a strength between zero and one.

It is important to remark that $\Delta E_c(\mathbf{p},\mathbf{q})$ is not included in the *stick* propagation function, since a small perceptual color difference in a specific channel not always indicates a uniform region, making inconvenient the propagation of color. Similarly, the terms $\eta_c(\mathbf{p})$ and $\overline{\eta_c}(\mathbf{p})$ do not appear in the *ball* propagation function, since $\mathbf{q}$ is not propagating a single color, thus making these terms unnecessary.

It is not difficult to show that the proposed propagation functions comply with the criteria described in Section 5.2. Perceptual color differences, uniformity measures, dependency on spatial distance and impulse noise measurements explicitly appear in (5.6) and (5.9). Although the edginess of a region does not explicitly appear in the above propagation functions, it is indirectly taken into account in the *ball* propagation function, since it can be modeled as the complement of the joint uniformity, that is, $\overline{U_c}(\mathbf{p},\mathbf{q})$.

It should also be highlighted that, unlike the canonical voting fields, it is not necessary to rotate the results of the proposed propagation functions as they do not depend on the orientation of the tensors, making the process less computationally expensive than tensor voting applied to surface reconstruction.

The proposed propagation functions require the application of the voting process (Section 5.3.2) twice. The first application produces an initial estimation of the normalized $\hat{s_1}$ and $\hat{s_2}$ saliencies, as they are necessary to calculate $U_c(\mathbf{p},\mathbf{q})$ and $\eta_c(\mathbf{p})$ in (5.4) and (5.5). For this first estimation, only perceptual color differences and spatial distances are taken into account, since no more information is available. Thus, $U_c(\mathbf{p},\mathbf{q})$ and $\eta_c(\mathbf{p})$ are set to zero in this first application. At the second application, the tensors at every pixel are initialized with the tensors obtained after the first application. Therefore, $\mathrm{T}_c(\mathbf{p}) = \mathrm{TV}_c(\mathbf{p})$ instead of (5.1). In this second application, the propagation functions can be applied in their full definition, since all necessary data are available.

5.4.3 Parameters of the CIEDE2000 formula

Although there are studies proposing equations to approximate the values for the parameters of CIEDE2000, k_L, k_C and k_H (cf. Section 2.2), they are not applicable in all cases, since they

were obtained in restricted scenarios. Therefore, it is necessary to propose a set of equations to estimate the most appropriate values for k_L, k_C and k_H for the specific application of edge-preserving image denoising.

Based on the formulation given by Chou and Liu (2007), the following equations for the CIEDE2000 parameters are proposed:

$$k_L = F_{B_L} F_{\eta_L}, \quad k_C = F_{B_C} F_{\eta_C}, \quad k_H = F_{B_h} F_{\eta_h}, \tag{5.11}$$

where F_{Bm} are factors that take into account the influence of the background color on the calculation of color differences in every color component m (L, C and h), and F_{η_m} are factors that take into account the influence of noise on the calculation of color differences in every component m. These factors are defined as follows.

On the one hand, F_{B_m} takes into account the fact that, as the background luminance decreases, big color differences in chromatic channels become less perceptually visible. This effect only appears in regions with a background luminance, L_B, below fifty [2]. Chou and Liu (2007) have found that the effect of the background on the estimation of color differences is approximately a linear function of the mean background luminance Y_B, which is the second color component of the mean background color expressed in the XYZ color model. Additionally, only parameters k_C and k_H are affected by background luminance, since the latter only affects the chromatic channels. Hence, $F_{B_L} = 1$. Thus, the effect of the background on the calculation of perceptual color differences in the color components C and h, F_{B_C} and F_{B_h}, can be estimated through the following equations:

$$F_{B_C} = F_{B_h} = 1 + R_B\,(1 - Y_B), \tag{5.12}$$

where R_B is a constant parameter. A value of three for R_B has shown the best performance in our experiments. Thus, factors F_{B_C} and F_{B_h} are in the range from one to four, since Y_B varies between zero and one. A factor of one means that the background color does not have any influence on the calculation of color differences, while a factor of four means that color differences found in the corresponding color component must be divided by four as the human visual system acuity is reduced four times in those conditions of background color.

On the other hand, F_{η_m} takes into account the fact that, as noise increases, big color differences become less perceptually visible. For this reason, it is necessary to estimate the amount

[2]There is no effect of L_B on CIEDE2000 for $L_B = 50$, since the experiments that led to the definition of the CIEDE2000 formula assumed that value. Moreover, to our knowledge, there is no evidence that greater values of L_B increase the acuity of the human visual system.

of noise and texture present in the image in order to study their influence on the calculation of perceptual color differences. This objective can be achieved by assessing the variability in the image. There are three main sources of variability in an image, namely: edges, texture and noise. However, it is only necessary to determine the variability due to noise and texture since there is no evidence that k_L, k_C and k_H depend on the variability due to edges. Assuming uniformly distributed noise and texture in the noisy image, the variability due to edges can be estimated through the following steps. First, the noisy image is convolved with a Gaussian of big variance (e.g., $\sigma^2 = 25$) and zero mean in order to eliminate the variability due to noise and texture. Second, the median absolute deviation (MAD) is calculated on both the noisy and the Gaussian filtered image. Finally, the MAD of the Gaussian filtered image gives an estimation of the variability due to edges, while the difference between the MAD of the noisy image and the MAD of the Gaussian filtered one gives an estimation of the variability due to noise and texture. This process is applied to the CIELAB luminance, chroma and hue components independently. Calculations of MAD require the evaluation of perceptual differences in luminance, chroma and hue. The absolute difference can be used to calculate perceptual luminance and chroma differences. However, that function is not appropriate for calculating perceptual hue differences, since they are more perceptually visible for colors with high chroma. The equation defined in CIEDE2000 to calculate this type of differences must be used instead. Thus, the perceptual hue difference, ΔH, is calculated as:

$$\Delta H = 2\sqrt{C_1 C_2}\,\sin\left(\frac{\Delta h}{2}\right), \tag{5.13}$$

where C_1 and C_2 are the chroma of both colors and Δh is the absolute hue difference.

Three parameters, J_L, J_C and J_h, are introduced in order to control the degree of preservation of texture on each color component, given that it is important to keep not only edges but also texture while noise is suppressed. In natural images, J_C, and J_h may be set to zero, since texture in these images is mainly retained by the luminance component (Ben-Shahar and Zucker, 2004). The definition of J_L depends on the perceptual amount of perceived texture in the image without noise. Thus, big values of J_L should be used in highly textured images as this ensures the preservation of texture. In our experiments, good results were obtained for values of J_L in the range from five to ten CIEDE2000 units, depending on the amount of texture in the image.

Since a good feature of a denoising algorithm is the suppression of noise as soon as possible, the CIEDE2000 parameters have been weighted by the amount of variability due to noise with

a tolerance of a single CIEDE2000 unit. Thus, the equations proposed to model the influence of noise on the calculation of perceptual color differences are:

$$F_{\eta_m} = \begin{cases} \eta_m, & \text{if } \eta_m > 1 \\ 1, & \text{otherwise,} \end{cases} \tag{5.14}$$

with:

$$\eta_m = MAD(u)_m - MAD(g)_m - J_m, \tag{5.15}$$

where u is the noisy image, g the image filtered with the Gaussian function and $MAD(\cdot)_c$ is the MAD calculated on component m (L, C or h). The F_{η_m} factors are in the range from one to infinite. However, in practice, values larger than fifteen are uncommon. A factor of one means that the noise level is too low to influence the estimation of color differences. A factor of five, for example, means that color differences found in the corresponding color component must be divided by five, as the human visual system acuity is reduced five times due to the noise level. Additionally, these equations make F_{η_m} to converge to one in a few iterations of the algorithm, making their calculation unnecessary once they have converged, since the images will not have enough noise to influence the estimation of color differences.

Our experiments have shown that the equations presented in this subsection lead to a good performance of the algorithm when denoising outdoor natural images. However, those equations should be adjusted for other types of scenarios.

5.5 Evaluation Methodology

In accordance with the features that must fulfill an edge-preserving image denoising algorithm, measures to assess the amount of removed noise, the edge preservation feature and the presence of artifacts after filtering are necessary to evaluate the performance of image denoising algorithms. Thus, in addition to visual inspection, three measures have been utilized in our experiments: the peak signal to noise ratio ($PSNR$), the $PSNR$ weighted by the gradient ($PSNRG$), and the $PSNR$ in false edges and/or undesirable artifacts ($PSNRA$). The $PSNR$ is a well-known quality measure, while $PSNRG$ and $PSNRA$ are new measures proposed in this chapter.

The $PSNR$ measure was chosen since it is the most widely used measure to compare denoising algorithms, given that it is easy to implement and gives a good estimation of the amount of

removed noise. The $PSNR$ for a color image, u, encoded in RGB with respect to a reference image, r, is calculated as:

$$PSNR = 10\ \log_{10}\left[\frac{3\times 255^2}{MSE_R + MSE_G + MSE_B}\right], \tag{5.16}$$

with

$$MSE_c = \frac{1}{mn}\sum_{i=0}^{n}\sum_{j=0}^{m}\left(u_c(i,j) - r_c(i,j)\right)^2. \tag{5.17}$$

In addition, we propose the $PSNRG$ measure to measure the edge preservation feature. The $PSNRG$ is defined as the $PSNR$ weighted by the gradient of the reference image, $||\nabla r_c(i,j)||$. The $PSNRG$ is a measure that gives more weight to pixels located nearby edges and discards those pixels located in flat regions. The $PSNRG$ is defined as:

$$PSNRG = 10\ \log_{10}\left[\frac{3\times 255^2}{MSEG_R + MSEG_G + MSEG_B}\right], \tag{5.18}$$

with

$$MSEG_c = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m}||\nabla r_c(i,j)||\ \left(u_c(i,j) - r_c(i,j)\right)^2}{\sum_{i=0}^{n}\sum_{j=0}^{m}||\nabla r_c(i,j)||}. \tag{5.19}$$

Finally, we propose the $PSNRA$ measure to assess the introduction of false edges and/or undesirable artifacts in uniform regions. In a first step, color differences at every pixel are weighted by the difference of gradients between the reference and the filtered image. This process is only applied to those pixels where the gradient of the filtered image is greater than the gradient of the reference image. In addition, only those pixels with the greatest differences of gradients are taken into account in order to eliminate a possible influence of pixels near edges whose gradient could become slightly greater than the gradient in the reference image. Finally, the $PSNRA$ is calculated in a similar way to the $PSNR$ for the selected pixels. Let a be the set of pixels where the difference between the gradient of u and r is greater than the 90th percentile of the differences of gradients. The $PSNRA$ is defined as:

$$PSNRA = 10\ \log_{10}\left[\frac{3\times 255^2}{MSEA_R + MSEA_G + MSEA_B}\right], \tag{5.20}$$

with

$$MSEA_c = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m}\Delta Gr(i,j)\left(u_c(i,j) - r_c(i,j)\right)^2}{\sum_{i=0}^{n}\sum_{j=0}^{m}\Delta Gr(i,j)}, \tag{5.21}$$

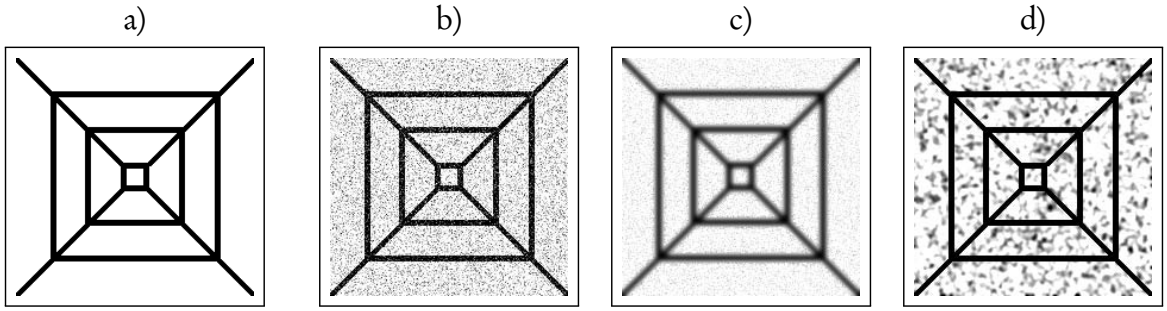


Figure 5.3: Visual assessment of *PSNRG* and *PSNRA*. a) Synthetic image. b-d) Noisy versions of the same image. The noisy images b)-d) have the same *PSNR* but different *PSNRG* and *PSNRA*.

$$\Delta Gr(i,j) = \delta(i,j)\,(||\nabla u(i,j)|| - ||\nabla r(i,j)||), \tag{5.22}$$

and

$$\delta(i,j) = \begin{cases} 1, & \text{if } ||\nabla u(i,j)|| > ||\nabla r(i,j)|| \\ & \text{and pixel } u(i,j) \in a \\ 0, & \text{otherwise.} \end{cases} \tag{5.23}$$

The utility of the new proposed measures, *PSNRG* and *PSNRA*, can be visually assessed in the synthetic example of Figure 5.3. This figure shows three different noisy versions of the same image that have the same *PSNR* but very different *PSNRG* and *PSNRA*. *PSNRG* is drastically reduced in Figure 5.3c due to egde blurring, while *PSNRA* is drastically reduced in Figure 5.3d due to the high amount of introduced artifacts.

Finally, visual inspection is also necessary in order to give a subjective assessment of the results, since the measures mentioned above were not designed to assess perceptual fidelity, that is, it is possible to find cases in which images with high *PSNR* (and therefore with low noise level) could be perceived noisier than other images with lower *PSNR*.

5.6 Experimental Results

The proposed technique, referred to as the tensor voting denoiser (TVD), has been compared to the methods proposed by Kervrann and Boulanger (2008) (an improved version of the non-local means method (Buades et al., 2005)), Tschumperlé (2006) (based on partial differential equations), and Portilla et al. (2003) (based on wavelets), since they represent the state-of-the-art in color image denoising. These methods will be referred to as NLM (Kervrann and Boulanger, 2008),

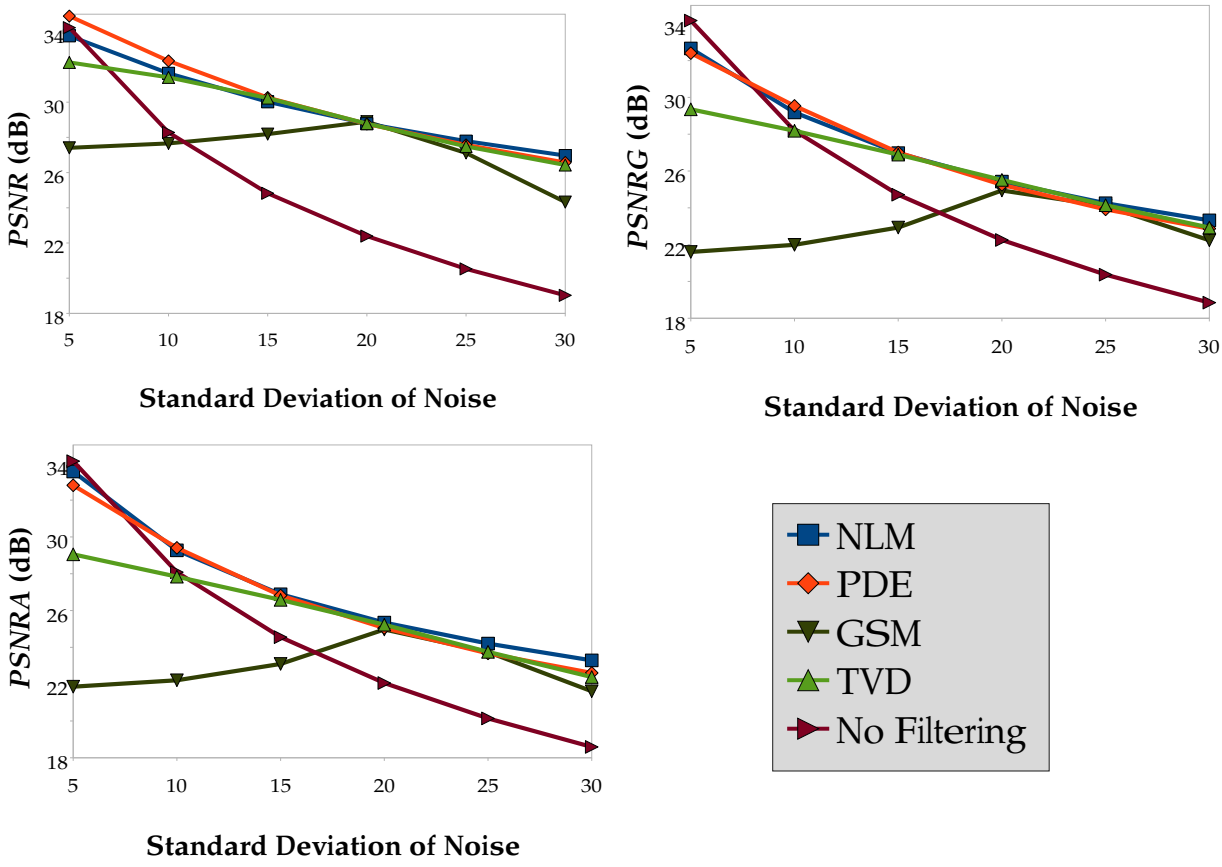


Figure 5.4: *PSNR*, *PSRNG* and *PSNRA* vs. standard deviation of AWGN for NLM, PDE, GSM and TVD.

PDE (Tschumperlé, 2006) and GSM (Portilla et al., 2003) respectively. The default parameters of NLM and PDE have been used. The GSM algorithm has been applied with $\sigma = 20$, since the best overall performance of this algorithm was attained with this standard deviation. The GSM algorithm has been applied to the three *RGB* channels independently, since this algorithm was designed for gray-level images. The TVD has been run with standard deviations $\sigma_s = 1.3$ for the G_{σ_s} Gaussian, $\sigma_d = 1.0$ for the G_{σ_d} Gaussian and $J_L = 7.0$ (cf. subsection 5.4.3). Regarding computational cost, NLM was the fastest of all tested algorithms when run on an Intel Core 2 Quad Q6600 with a 4GB RAM (about three seconds per iteration), followed by PDE (around five seconds per iteration), TVD (around twenty seconds per iteration, since every application of the voting process takes ten seconds approximately) and GSM (more than two minutes per iteration).

In the experiments, one hundred outdoor images from the Berkeley segmentation data set (Martin et al., 2001) have been contaminated with two types of noise: additive white Gaussian noise (AWGN) and CCD camera noise (modeled as in Liu et al., 2008). Although AWGN is unable to accurately model the noise generated by real cameras (cf. Section 2.2), it has been included in the comparisons since most methods of the state-of-the-art have only been tested with it. In addition, it is interesting to assess the behavior of those state-of-the-art algorithms with respect to CCD camera noise in order to validate their performance in real conditions.

Ten iterations have been run for every algorithm, input image, noise type and noise level. The output image with the highest *PSNR* obtained from those iterations was chosen. The chosen images have been used to compare the performance of all the algorithms. No pre or post-processing stages have been applied to the images in order to evaluate the ability of the algorithms to remove noise without any help. Thus, the reported measurements are only due to the performance of the algorithms and not to any additional stage. The following subsections show the comparisons with AWGN and CCD camera noise.

5.6.1 Experiments with Additive White Gaussian Noise

Figure 5.4 shows the plots of *PSNR*, *PSNRG* and *PSNRA* vs. standard deviation of AWGN for NLM, PDE, GSM and TVD. According to the *PSNR* curve, NLM, PDE and TVD have almost the same performance for standard deviations of noise greater or equal than 10. NLM has the best performance for smaller amounts of noise followed by PDE. GSM only has a good

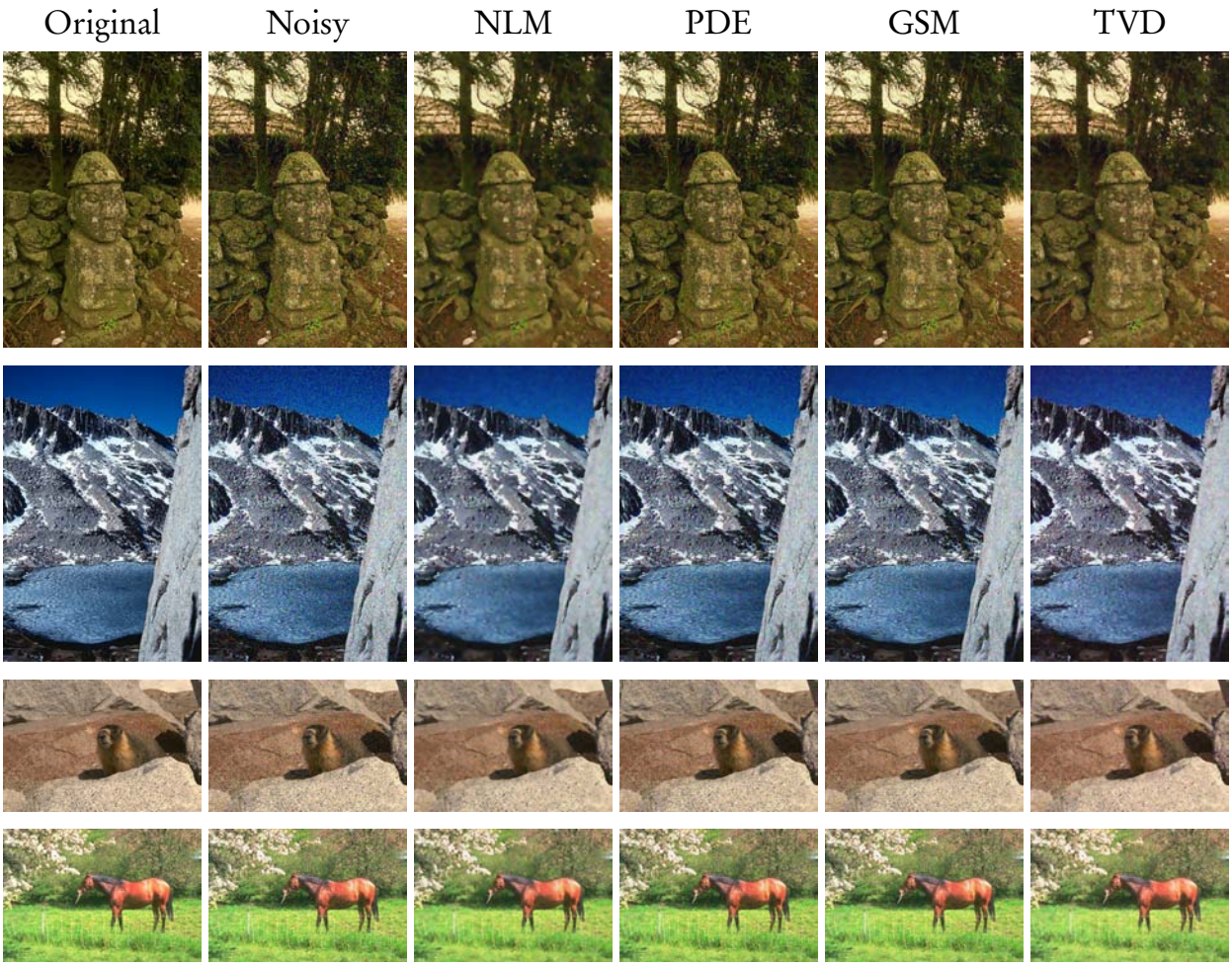


Figure 5.5: Denoising results for AWGN. The first column shows the original images. The second column shows the noisy images (standard deviation = 30). Columns three to six show the denoised images after applying NLM, PDE, GSM and TVD respectively.

performance for standard deviations of noise between 20 and 25.

According to the $PSNRG$ and $PSNRA$ curves, NLM, PDE and TVD have almost the same performance for standard deviations of noise greater than 10. That means that NLM, PDE and TVD have the same ability for preserving edges and not introducing artifacts in these scenarios. NLM and PDE have the best performance for standard deviations of noise smaller than 15. GSM is only competitive for standard deviations of noise greater than 15. In summary, TVD has a competitive performance in $PSNR$, $PSNRG$ and $PSNRA$ with respect to AWGN, especially for highly noisy scenarios. Figure 5.5 shows some denoising results obtained with the four tested algorithms[3]. It can be seen that the denoised images are similar to each other. However, NLM and TVD introduce fewer artifacts.

An additional observation from the $PSNR$, $PSNRG$ and $PSNRA$ curves is that it is difficult for all the tested methods to remove noise near edges and to avoid the introduction of undesired artifacts. For example, the improvement in $PSNR$ of the NLM for a standard deviation of noise of 30 is 7.94 dB, while it only attains 4.47 dB and 4.70 dB for $PSNRG$ and $PSNRA$ respectively.

5.6.2 Experiments with CCD Camera Noise

In order to obtain $PSNR$, $PSNRG$ and $PSNRA$ curves for real noise, it is necessary to generate synthetic noisy images from noiseless ones. Synthetic CCD camera noise was generated in the experiments as proposed by Liu et al. (2008). This methodology to generate synthetic CCD camera noise was chosen since it takes into account most of the sources of real noise for CCD cameras from a variety of brands.

Figure 5.6 shows the plots of $PSNR$, $PSNRG$ and $PSNRA$ vs. amount of CCD camera noise for NLM, PDE, GSM and TVD. Additionally, Table 5.1 shows the improvement in decibels for the tested algorithms, taking the $PSNR$ of the noisy images as the baseline. The best performances are shown in bold.

According to the $PSNR$ curve, NLM, PDE and TVD have almost the same performance for a noise of 2.5%. TVD has the best performance for larger amounts of noise followed by PDE. NLM and GSM have similar performances for amounts of noise greater than 5%. According to the $PSNRG$ curve, NLM, PDE and the TVD have almost the same performance for amounts of noise greater than 2.5%. That means that NLM, PDE and TVD have the same ability to preserve

[3]Full-resolution images are available at http://deim.urv.cat/ ~ rivi/tvd.html

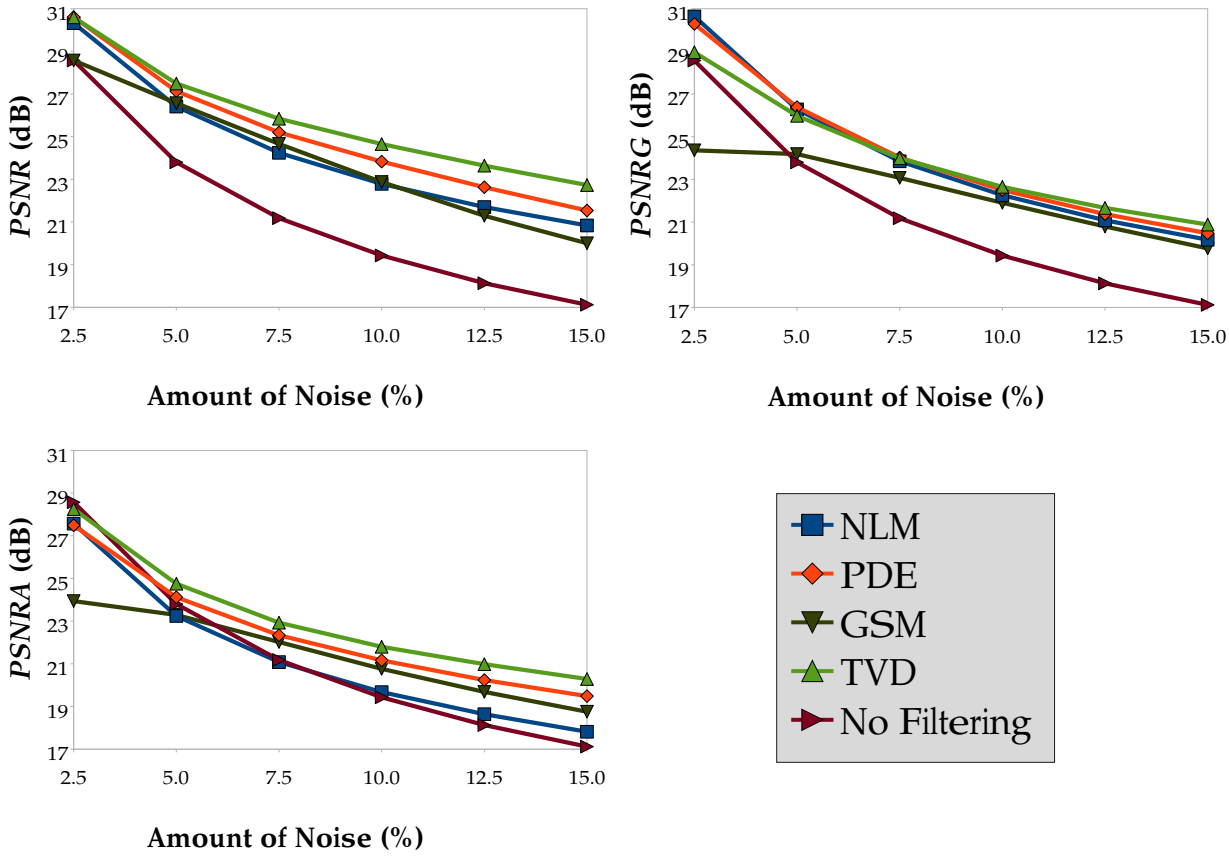


Figure 5.6: *PSNR*, *PSRNG* and *PSNRA* vs. amount of CCD camera noise generated as by Liu et al. (2008) for NLM, PDE, GSM and TVD.

edges in these scenarios. NLM and PDE have a better performance in *PSNRG* for an amount of noise of 2.5%. GSM has the worst performance for all amounts of noise. GSM has a poor performance for a noise of 2.5% since the *PSNRG* is below the baseline curve. This means that GSM is unable to preserve edges in this case. According to the *PSNRA* measure, TVD has the best performance for all amounts of noise. NLM and GSM have a poor performance in *PSNRA* as their curves appear near or even below the baseline curve for amounts of noise below 10%. This means that NLM and GSM cannot avoid introducing artifacts for amounts of noise below 10%. In general, NLM has a worse performance than GSM, since GSM improves its performance for high amounts of noise.

Table 5.1 shows that TVD outperforms the other algorithms in almost all the cases according to the *PSNR*, *PSNRG* and *PSNRA* measures. A negative value indicates that the filtered images are worse than the noisy ones. That condition appears in some cases for NLM and GSM. An interesting observation is that, in general, the algorithms for CCD camera noise yield lower values of *PSNRG* than *PSNR*, and lower values of *PSNRA* than *PSNRG* for the same amounts

Table 5.1: *PSNR*, *PSNRG* and *PSNRA* improvements for CCD camera noise

Algorithm	amount of noise = 5%			amount of noise = 10%		
	PSNR	*PSNRG*	*PSNRA*	*PSNR*	*PSNRG*	*PSNRA*
NLM	2.61	2.48	-0.56	3.36	2.83	0.24
PDE	3.33	**2.59**	0.32	4.40	3.07	1.73
GSM	2.77	0.39	-0.51	3.44	2.47	1.33
TVD	**3.69**	2.18	**0.95**	**5.22**	**3.22**	**2.37**

of noise. This means that it is more difficult for the tested algorithms to preserve edges than to reduce noise, and even much more difficult to avoid introducing artifacts than to preserve edges.

Figure 5.7 shows some close-ups of the denoising results obtained with the four tested algorithms[4]. It can be seen that NLM generates undesirable quantization artifacts and colored spots. In addition, PDE generates cross-shaped artifacts. GSM partially removes noise. TVD produces better results since it generates fewer artifacts than the other algorithms.

5.6.3 Additional Experiments

Since no source code is available from the authors of the methods proposed by Liu et al. (2008) and Tai et al. (2006b), those techniques have not been able to be tested upon the same set of images as the other techniques. However, in order to compare their performance with TVD, the latter has been applied to the same high-resolution noisy images used by Liu et al. (2008) and Tai et al. (2006b) respectively.

Figures 5.8 and 5.9 show the qualitative comparison between the resulting images processed by TVD, and the images reported in the respective papers. Figure 5.8 shows that TVD has a better performance than the Liu et al. (2008) method in real noisy images, since TVD generates fewer quantization artifacts and fewer saw-shaped artifacts in edges. Figure 5.9 shows that TVD outperforms the method presented by Tai et al. (2006b) on salt and pepper noisy images, since TVD also generates fewer artifacts in edges.

[4]Full-resolution images are available at http://deim.urv.cat/ ~ rivi/tvd.html

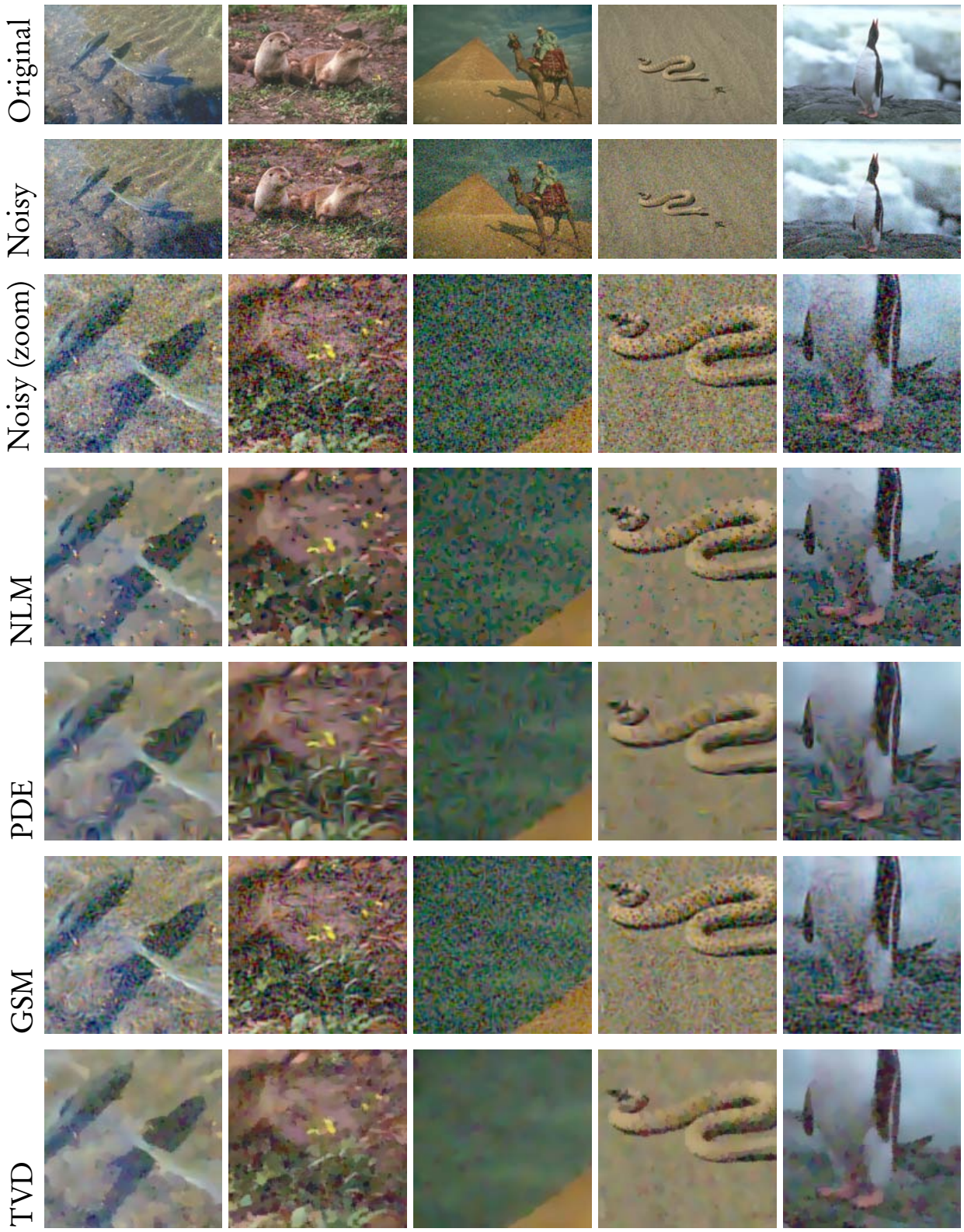


Figure 5.7: Denoising results. The first row shows the original images. The second row shows the noisy images (10% of CCD camera noise). The third row shows close-ups of the noisy images. Rows four to seven show close-ups of the denoised images after applying NLM, PDE, GSM and TVD respectively.

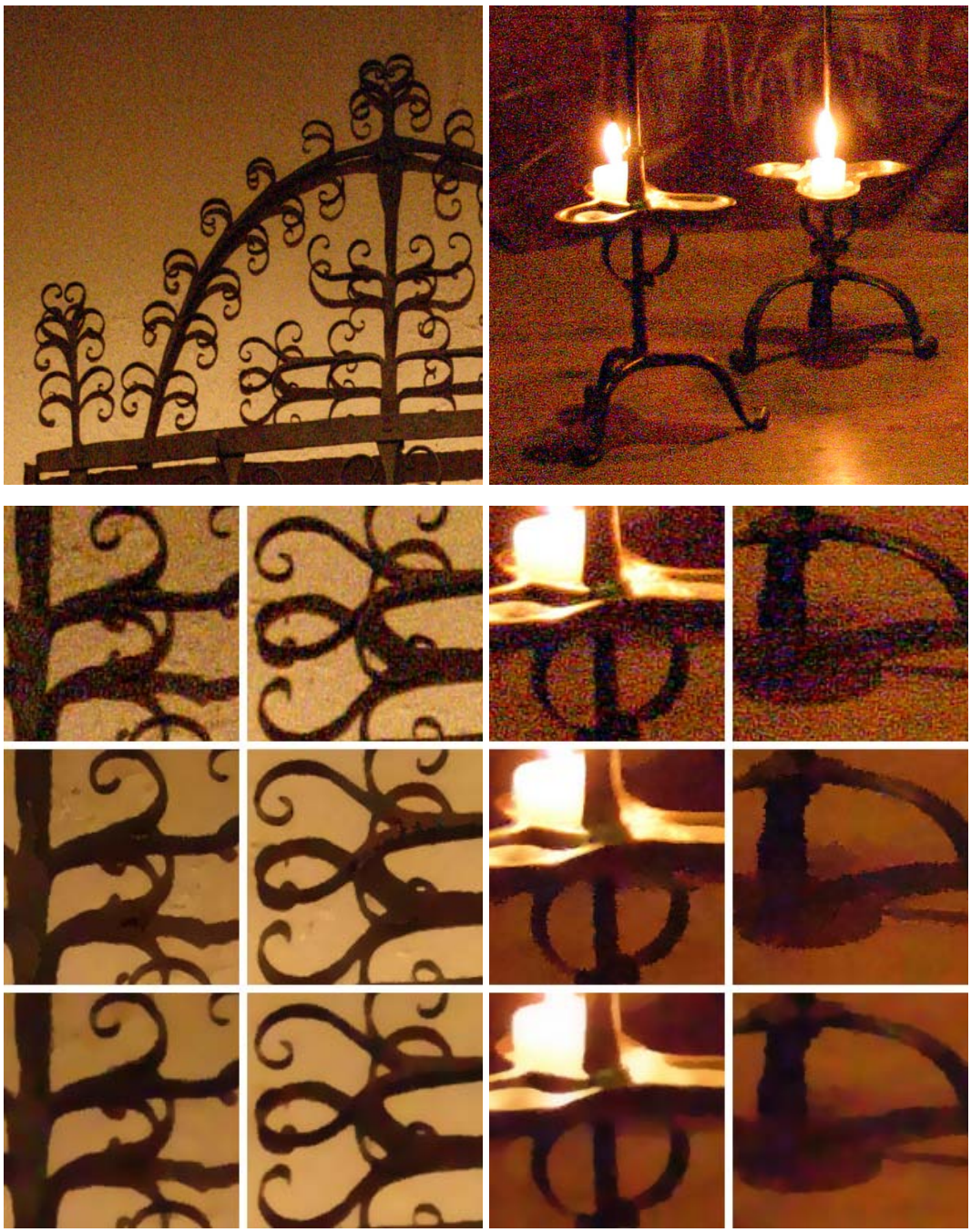


Figure 5.8: Comparison with Liu et al. (2008). The first row shows real noisy images taken from Liu et al. (2008). The second row shows some close-ups of the same images. The third and fourth rows show the denoising results of Liu et al. (2008) and the TVD respectively. High-resolution images of the first, second and third rows were directly taken from the electronic version of Liu et al. (2008).

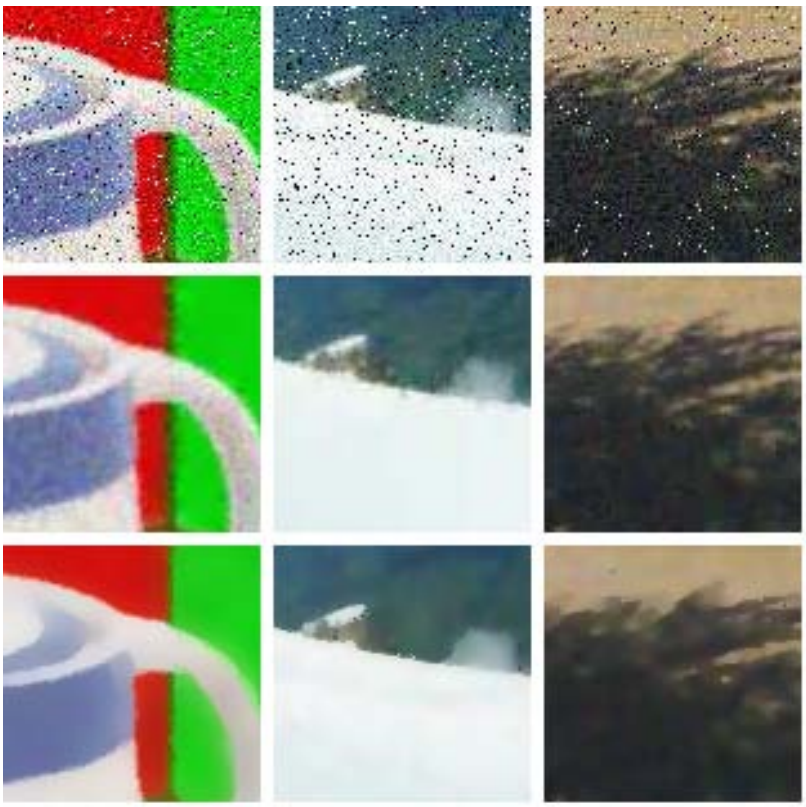


Figure 5.9: Comparison with Tai et al. (2006b). The first row shows images contaminated with salt and pepper noise. The second row shows the denoising results reported by Tai et al. (2006b). The third row shows the results obtained with TVD. High-resolution images of the first and second row were directly taken from the electronic version of Tai et al. (2006b).

5.7 Summary

A new method to denoise color images while preserving edges and texture has been proposed. This method is based on an adaptation of tensor voting. The proposed adaptation of tensor voting takes into account a set of criteria inspired by the perceptual process of image denoising. An optimized version of CIEDE2000 has been proposed to measure color differences in noisy environments by modifying its parameters k_L, k_C and k_H.

The results presented in this chapter show that the use of a specific voting process makes it possible the use of tensor voting as a powerful tool for image denoising. Measurements of removed noise, edge preservation and undesirable introduced artifacts, additionally to visual inspection have been used to compare the performance of the TVD against some of the state-of-the-art color image denoising algorithms. The two new measures, $PSNRG$ and $PSNRA$, have been proposed to measure the edge preservation and undesirable artifact rejection features of the algorithms respectively, while $PSNR$ has been used to measure the amount of removed noise. These experiments show that TVD has a similar performance to the other tested algorithms with respect to AWGN noise, and better than them regarding CCD camera noise, especially in cases with large amounts of noise. This means that TVD is more appropriate than the other tested methods for real applications, which is the final objective of any image denoiser. In addition, the new measures $PSNRG$ and $PSNRA$ have been utilized to show that it is more difficult for denoising algorithms to preserve edges and to avoid the introduction of artifacts than to reduce noise.

Chapter 6

Tensor Voting for Robust Color Edge Detection

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" means 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 then applied to the edginess maps in order to generate such binary maps (Canny, 1986).

Many color edge detectors have already outperformed gray-valued edge detectors (e.g., Ruzon and Tomasi, 2001; Evans and Liu, 2006; Maire et al., 2008; Batard et al., 2009). However, their performance usually decreases for noisy images. As an alternative, this chapter proposes adaptations of the methods described in the previous chapter and in Section 4.1 to robust color edge detection. These methods derive from an analysis of their edge detection properties.

This chapter is organized as follows. Section 6.1 describes previous related work. Section

6.2 details the adaptation of both the classical tensor voting and the adaptation of tensor voting presented in Chapter 5 to color edge detection. Section 6.4 shows a comparative analysis of the proposed methods against some of the state-of-the-art color edge detection algorithms by using the quality measures introduced in Section 6.3. Finally, Section 6.5 discusses the obtained results and makes some final remarks.

6.1 Previous Related Work

Novak and Shafer (1987) stated that almost 90% of the edges in an image can be obtained only using gray-level information. This is due to the fact that both luminance and chromatic edges tend to be highly correlated in most environments. Thus, the first efforts in this area were devoted to the development of edge detectors for gray-scale images. It is interesting to remark that, even though Canny (1986) presented his gray-valued edge detector more than twenty years ago, in almost all the studies that compare the performance of edge detectors in gray-scale images (e.g., Bowyer et al., 2001; Heath et al., 1997; Shin et al., 2001; Spreeuwers and van der Heijden, 1992; Nguyen and Ziou, 2000), Canny's detector often appears as the best. Despite this, Canny's detector has also been outperformed by many methods in the last decade.

It is important to remark that traditional evaluation methodologies require binary edge maps. Thus, post-processing must be applied before those methodologies. Canny defined a standard methodology to extract binary edge maps from edginess maps, which includes a non-maximum suppression and a hysteresis thresholding step (Canny, 1986).

However, comparing binary edge maps is problematic with Canny's algorithm, since its performance has been reported to be dependent on a non-trivial selection of parameters to post-process the edginess maps into binary edge maps (Zhang and Rockett, 2006). The main problem is that the selection of parameters directly depends on the application. Despite that, some studies have proposed methods to automatically select the parameters for post-processing edginess maps (e.g., Koren and Yitzhaky, 2006; Naik and Murthy, 2006; Medina-Carnicer et al., 2009).

Despite its usefulness, it is well known that gray-scale detectors cannot detect some kind of edges in images, since different colors may be mapped to the same gray level. Thus, recent studies have reported better performances than gray-scale edge detectors by using color information (e.g., Ruzon and Tomasi, 2001; Evans and Liu, 2006; Maire et al., 2008; Batard et al., 2009).

The following classification of approaches for color edge detection has been proposed by

Maire et al. (2008):

1. Local methods that aim at extracting edges from local measurements. In turn, these approaches have been classified by Ruzon and Tomasi (2001) into three categories:

   (a) Working with multidimensional gradient methods. For instance, Di Zenzo used tensors to estimate gradients in color images (Di Zenzo, 1986), Cumani used a variant of Di Zenzo's approach to estimate second directional derivatives in images (Cumani, 1991), and Chapron used the Dempster-Shafer theory to fuse gradients (Chapron, 2000).

   (b) Treating every color component separately and then combining the individual final results. For example, by using an OR operation (Fan et al., 2001), or by computing a weighted sum of the results of an operator (Weeks et al., 1995).

   (c) Treating color information as a color vector and solving the problem in the vector space. For example, the so-called *compass* operator (Ruzon and Tomasi, 2001) and the use of *vector order statistics* (Trahanias and Venetsanopoulos, 1996) are some of the most relevant approaches in this direction.

2. Global methods use global information in the edge detection. Most of these methods tackle both problems: image segmentation and edge detection in a unified framework, by assuming that edges are the boundaries between regions. Thus, this approach claims that any image segmentation algorithm (cf. Section 7.1 for a review of this family of algorithms) can be used to detect edges.

Local methods usually over-performed global methods since the latter are prone to discarding relevant edges that are not boundaries. However, recent methods have been proposed to use both types of information in a collaborative way (Maire et al., 2008).

In this context, the methods proposed in this chapter can be classified as local methods that follow the first and second aforementioned subcategories respectively. Complete reviews of strategies on color edge detection are presented by Koschan (1995); Zhu et al. (1999); Koschan and Abidi (2005).

Although a number of edge detectors have been proposed during the last years, only a few have been devised to deal with noise. The reason can be endorsed to the fact that edges of noisy images can be extracted from denoised versions of the input images (De Micheli et al., 1989). This

strategy is followed, for example, by Xue-Wei and Xin-Rong (2008). However, as seen in Chapter 5, denoising is not a trivial problem, especially for real noise. In addition, using image denoisers 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 algorithm.

The most popular approach in this specific area of robust edge detection has been the use of robust statistics. For instance, the methods proposed by Das and Anand (1995), Hou and Koh (2003), and Lim (2006) belong to this category. However, these methods have been designed to deal with specific types of noise by assuming simplifications that strongly restrict the scope of use of these edge detectors. Other approaches require training steps that restrict the scope of the methods. This is the case of the method by Yüksel (2007), which applies fuzzy logic and neural networks.

In this context, this chapter explores a new approach to extract edges from noisy color images by applying perceptual techniques, in particular tensor voting. Thus, both the classical tensor voting and the method proposed in Chapter 5 have been adapted to the edge extraction process. The following section details the new methods.

6.2 Tensor Voting for Robust Color Edge Detection

Since the human visual system is able to detect edges in noisy scenarios, using perceptual techniques for robust edge detection appears promising. On the one hand, the classical tensor voting seems a good candidate to be used in robust edge detection thanks to its good properties for image structure estimation (cf. Section 4.1). On the other hand, the adaptation introduced in Chapter 5 also appears to be an appropriate perceptual technique for edge detection, since it indirectly makes estimations of edginess that are used to effectively filter out noise.

The following subsections explore these two perceptual approaches for color edge detection in noisy scenarios.

6.2.1 Color Edge Detection Using the Classical Tensor Voting

As shown in Section 4.1, the classical tensor voting can be used to detect edges and corners. Thus, after having applied tensor voting and the energy normalization step described in that section, the principal eigenvalue λ_1 can be used to detect edges, since it attains high values not only at

boundaries but also at corners.

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, there are two alternatives to improve the results of the classical tensor voting: to use a previous denoising step or to iterate the method. Although both alternatives could yield good results, the second alternative is more relevant for the objectives of this thesis, since the results can be used to assess the edge detection properties of the classical tensor voting, whereas the success of the first alternative could be partially attributed to the denoising algorithm.

Iterative tensor voting can also be useful for edge detection in noiseless images. In many applications (e.g., image segmentation), only the most relevant edges are required. Thus, the most significant edges can be reinforced by iterating the classical tensor voting, at a cost of discarding small ridges. According to the experiments, a few iterations (two or three) usually give good results for both noisy and noiseless images.

In addition, based on the experiments of Section 4.1, the rotation term of the *stick* tensor voting (cf. Section 2.6.1) has a negligible influence on λ_1. Thus, this rotation can be discarded in order to speed up the edge detection process at an affordable cost in terms of quality.

6.2.2 Color Edge Detection Using the Alternative Tensor Voting

The problems of image denoising and edge detection are closely related. On the one hand, due to the derivative nature of edge detection, edge detectors are usually more accurate in noiseless than in noisy scenarios. On the other hand, using edge estimations to boost the denoising process is common in the state-of-the-art denoising algorithms, as shown in Chapter 5. This suggests that both problems can be tackled simultaneously.

Following this reasoning, the technique proposed in Chapter 5 for image denoising can be directly used to detect edges in color images. This method assumes that denoising and edge detection are dual problems that can be solved in a unified framework. Thus, after having applied the voting process described in Section 5.3.2, the edginess stored in the tensors (cf. Figure 5.2) can be used to compute an edginess map. Let E_i be the edginess at pixel i, and $\hat{s_{2c}}(i)$ be the normalized $\hat{s_2}$ saliency at channel c measured at i. E_i can be estimated as:

$$E_i = \frac{1}{3}\sum_{c=1}^{3} w_c \hat{s_{2c}}(i), \tag{6.1}$$

where w_c are weights that can be used to modulate the importance of every channel in the estimation of edginess. In order to give the same relevance to every color channel, the three parameters w_c have been set to one in the experiments.

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.

6.3 Evaluation Methodology

Many measurements have been proposed to assess the performance of edge detectors. The most used ones are the Pratt's Figure of Merit (FOM) (Pratt, 2007), the Receiver Operating Characteristic (ROC) curve (Bowyer et al., 2001; Yitzhaky and Peli, 2003) , Precision vs Recall (PR) curves (Martin et al., 2004), the F-measure (Martin et al., 2004), and indirect measurements (Shin et al., 2001; Baker and Nayar, 1999). All of them are based on measuring performance on the features that an edge detector must comply with (Canny, 1986): a) good detection (edges should be detected even in high levels of noise) , b) good localization (detected edges must be as close to the real edges as possible) and c) only one response to a single edge (the number of false maxima must be minimized).

The main drawback of these measurements is that they assume that the output of the edge detector is a binary edge map, making necessary the use of application-dependent post-processing steps before the evaluation. For example, edge-based segmentation and image enhancement methods that operate on edginess maps, henceforth referred to as image edge enhancement methods, usually require a different post-processing. Thus, these measurements could be biased by the scope in which the edge detector is being applied, making the results only valid for such a scope. For example, it has been shown that the performance of the Canny's edge detector (Canny, 1986) highly depends on the post-processing step (Zhang and Rockett, 2006).

Assessing edge detection before the post-processing steps can avoid those biases, giving a more accurate measure of how good the edge detector is, disregarding the context. Taking this fact into account, this section proposes a set of desirable features that a general purpose edge detector should comply with and some tools to measure them. These features can be measured directly on the edginess maps, thus avoiding possible bias generated by post-processing steps.

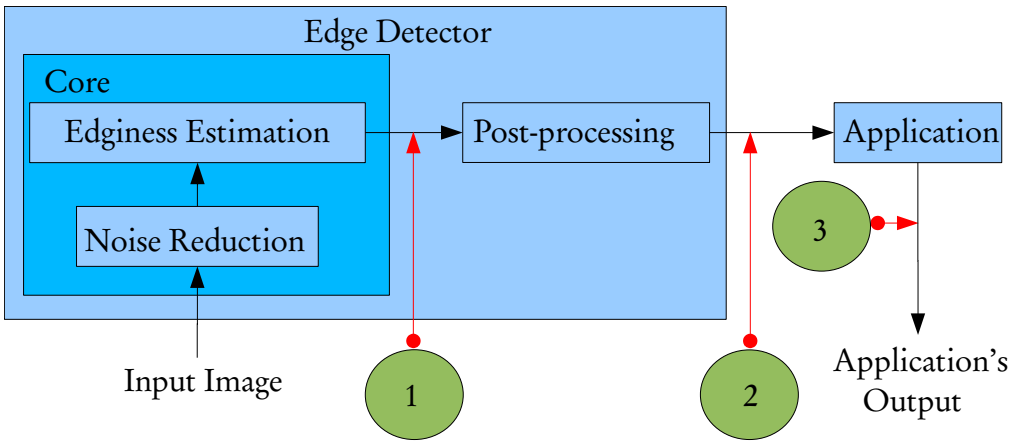


Figure 6.1: The edge detection process. The performance of edge detectors can be assessed at the points marked in green.

6.3.1 Quality Measurements

In general, the edge detection process comprises three steps (see Figure 6.1). 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 the post-processing step outside, since this final step 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 Figure 6.1. Measurements on edginess maps can be applied at the first point, to binary edge maps at the second point, and as application-dependent measurements at the third point of the figure. Direct measurements at the output of the algorithms are used at the first and second points, while performance at the third point is indirectly assessed by means of measurements of performance of 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 such an application. Many evaluation methodologies have been proposed to evaluate performance at the second (e.g., Pratt, 2007; Heath et al., 1997; Bowyer et al., 2001; Martin et al., 2004) and third points (Shin et al., 2001; Baker and Nayar, 1999) but, to the best of our knowledge, measuring

performance at the first point has not been proposed so far. Despite this, assessing 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. However, it is important to remark that assessing performance at the second or third points can still be useful to evaluate edge detectors in specific contexts.

This section proposes the measure of performance at the first point by means of four features: 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 next subsections.

Completeness

Completeness is referred to as 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 or not an edge is relevant only depends on the application. For instance, applications such as image edge enhancing, edge-based segmentation or texture feature extraction usually give a different relevance to every detected edge. Thus, an edge detector reduces its scope when it decides to discard edges. Despite that, most edge detectors usually opt for decreasing their scope for the sake of improving their performance in other features, such as discriminability or robustness.

Complete ground-truths with all the 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_i be the distance between the i-th pixel in the ground-truth and its corresponding matching pixel 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. The R-measurement given by:

$$R = \frac{1}{M}\sum_{i=1}^{M}\phi(D_i), \tag{6.2}$$

can be used to estimate recall. An issue related to measuring recall when $N > M$ is the fact that every edge detector generates a different number of edges and this can give advantage to detectors

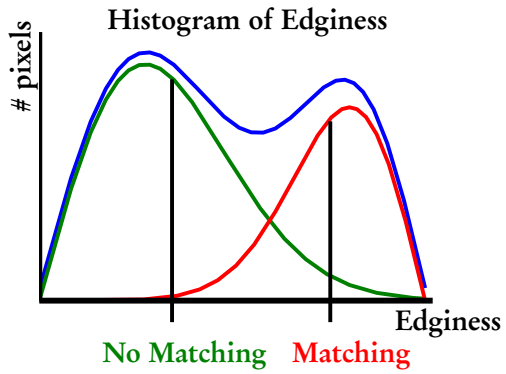


Figure 6.2: Computation of the DS-measurement. The histogram of edginess of the NMSE map can be seen as the summation of the histogram of "Matching" (in red) and "No matching" (in green) pixels to the ground-truth. The DS-measurement calculates how separated both histograms are from each other by substracting their means.

that generate higher number of edges, since D_i tends to be reduced when N increases. This bias can be eliminated 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 taken from the NMSE maps. R vs. N' plots can also be used to analyze the evolution of R.

Discriminability

Discriminability is referred to as the ability of an edge detector to discriminate between important and not important edges. This feature is application-dependent since importance 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 (the simplest post-processing algorithm) 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. The process can be seen in Figure 6.2. Function $\phi(\cdot)$ is used to measure the degree of membership of every pixel to the "Matching" and "No matching" sets plotted in the figure.

Let E_i be the edginess at pixel i of the NMSE map, usually in the 0 to 255 range. The DS-

measurement is given by:

$$DS = \frac{\sum_{i=1}^{N} E_i \phi(D_i)}{\sum_{i=1}^{N} \phi(D_i)} - \frac{\sum_{i=1}^{N} E_i (1 - \phi(D_i))}{\sum_{i=1}^{N} 1 - \phi(D_i)}. \tag{6.3}$$

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 these errors. Let $\overline{D}$ be the mean distance from pixels of the ground-truth that match the NMSE map. The P-measurement can be used to estimate precision:

$$P = \phi(\overline{D}). \tag{6.4}$$

A feature related to precision is the false alarm rejection feature, which represents the ability of edge detectors not to detect edges in flat regions. The FAR-measurement is given by:

$$FAR = \frac{1}{N} \sum_{i=1}^{N} \phi(D_i). \tag{6.5}$$

It 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 the calculations of 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.

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 derivative 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 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 (Prieto and Allen, 2003), 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. PSNR can be calculated through (5.16) with the noiseless image as the image of reference.

6.3.2 Relationships with Other Assessment Measures

The R, DS, P and FAR-measurements can be seen as generalizations of the Pratt's FOM (Pratt, 2007) by choosing:

$$\phi(x) = \frac{1}{1+\alpha x^2}, \tag{6.6}$$

with α being a constant.

Related measures such as the Pratt's FOM (Pratt, 2007) and the pixel correspondence metric (Prieto and Allen, 2003) are unable to measure recall, precision and false alarm rejection separately. PR curves (Martin et al., 2004), ROC curves (Bowyer et al., 2001; Yitzhaky and Peli, 2003), the average risk (Spreeuwers and van der Heijden, 1992) and the F-measure (Martin et al., 2004) are based on the measurement of three features of edge detectors: precision, recall, and false alarm rejection on binary edge maps.

All of them are based on the classification of the detected edges into four groups: true positives, true negatives, false positives and false negatives. Unfortunately, a correct classification of

edges is only possible for complete ground-truths, and only incomplete, application-dependent ground-truths are usually available for natural images. Thus, for example, a detected edge which does not match a specific ground-truth could be misclassified as a false positive, since it still could match an edge if the complete ground-truth is used instead.

Similarly, a pixel that is not marked both by the detector and by the specific ground-truth could be misclassified as a true negative, since an additional true edge still could appear at the pixel's location in the complete ground-truth. The proposed measurements can evaluate these three features more effectively since they do not require the classification of edges.

6.3.3 Ground-Truths for Edge Detection Assessment

As mentioned above, ground-truths are very important for assessing edge detectors. However, they must be used carefully in order to obtain reliable and fair assessments. This subsection lists the limitations of every type of ground-truth. This information can be used to choose the most appropriate ground-truth for every specific application.

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 (e.g., the Berkeley database presented by Martin et al., 2001, for image segmentation belongs to this category) and, ground-truths generated by consensus are obtained from the output of a bank of edge detectors (e.g., Fernández-García et al., 2008).

Artificial ground-truths are not generally used in comparisons since the results can barely be extrapolated to real scenes (Bowyer et al., 2001). 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 directions given by the experimenters. For example, humans do not usually mark the edges inside a textured region when the instruction is to annotate the edges necessary to separate regions. Hence, a manual ground-truth obtained for image segmentation cannot 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 in marking edges. An additional problem is that gray-scale manual ground-truths are almost impossible to obtain for natural scenes (Heath et al., 1998).

Ground-truths generated by consensus rely on the hypothesis that the bank of edge detectors have a good performance in all the 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.

6.4 Experimental Results

A total of fifteen images from the Berkeley segmentation data set (Martin et al., 2001) have been used in the experiments. In addition to the Laplacian of Gaussians (LoG), Sobel and Canny detectors, the methods proposed by Maire et al. (2008), referred to as the LGC method, and by Ruzon and Tomasi (2001), 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. Parameters $\sigma = 1.3$ and $b = 0$ (since curved edges may appear) have been used for the method based on the classical tensor voting, referred to as CTV, while parameters $\sigma_s = 1.3$, $\sigma_d = 2.5$ and $J_L = 0$ have been chosen for the edge detector based on the denoiser described in Chapter 5, referred to as TVED.

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 (Fernández-García et al., 2008), which could be used in the image edge enhancement application, and the hand-made ground-truth of the Berkeley segmentation data set (Martin et al., 2001), whose validity has only been proven in segmentation related applications. We will refer to those ground-truths as GT1, GT2 and GT3 respectively.

The function defined in (6.6) was used with $\alpha = 1/9$ as suggested by Pratt (2007). We matched every detected edge to its closest pixel in the ground-truth, allowing for up to one match for every ground-truth pixel. However, more sophisticated matching algorithms could be used (e.g., Martin et al., 2004; Prieto and Allen, 2003). Gaussian noise with different standard deviations has been added to the images for the robustness analysis.

Table 6.1 shows the proposed performance measurements for GT1 and GT2. Evolution plots for the proposed performance measurements are not necessary for GT1 and GT2 since $M \geq N$

Table 6.1: Performance measurements for ground-truths GT1 and GT2.

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	**0.84**	**0.75**	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	**44.33**	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	**0.98**	**0.92**	**0.95**	**0.87**
TVED	0.20	0.53	**21.24**	34.39	**0.98**	0.75	**0.95**	0.69

for them.

As for 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 Sobel has a good performance in recall. In general, these results were expected since $M \geq N$.

As for 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 to better recall figures for the sake of discriminability, and precision and false alarm rejection respectively. Sobel is the best in recall and has a competitive performance in the other measures.

Figure 6.3 shows the evolution of the proposed performance measurements for GT3 with N'. It can be seen in the figure 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 in FAR. The performance of the Sobel and LoG detectors is the worst, but it increases with N' even surpassing in R and P the LGC for high values of N'. The new proposed methods have competitive results, especially TVED. For example, unlike LGC, TVED has an increasing trend with N', and outperforms Compass in DS. Although Canny has been outperformed in all measurements, it is still competitive in DS.

As for the robustness analysis (see Figure 6.4), original images have been contaminated with AWGN with different standard deviations. TVED appears to be the most robust algorithm with

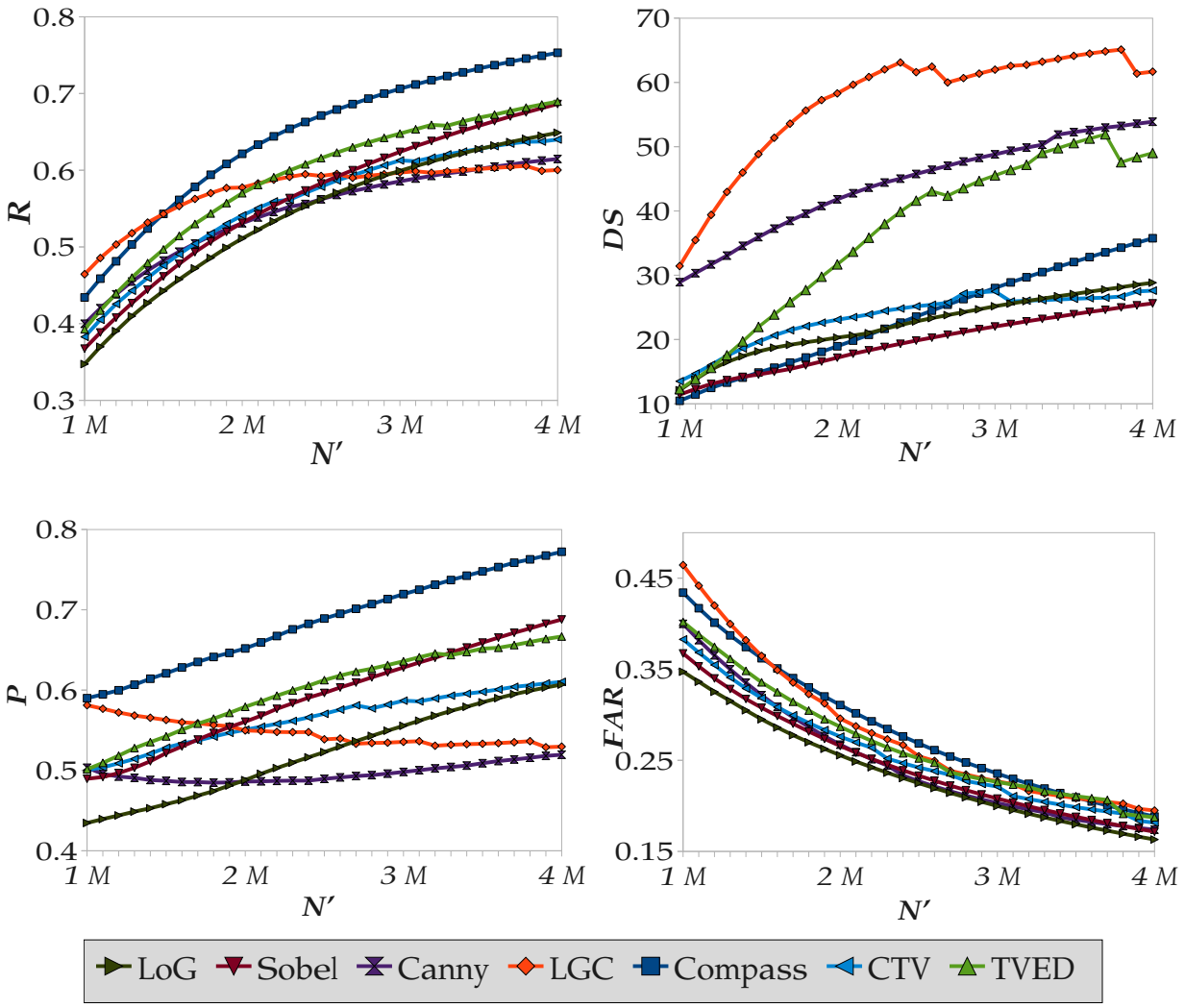


Figure 6.3: Performance measurements for GT3: top left: R vs. N' (recall); top right: DS vs. N' (discriminability); bottom left: P vs. N' (precision); bottom right: FAR vs. N' (false alarm rejection).

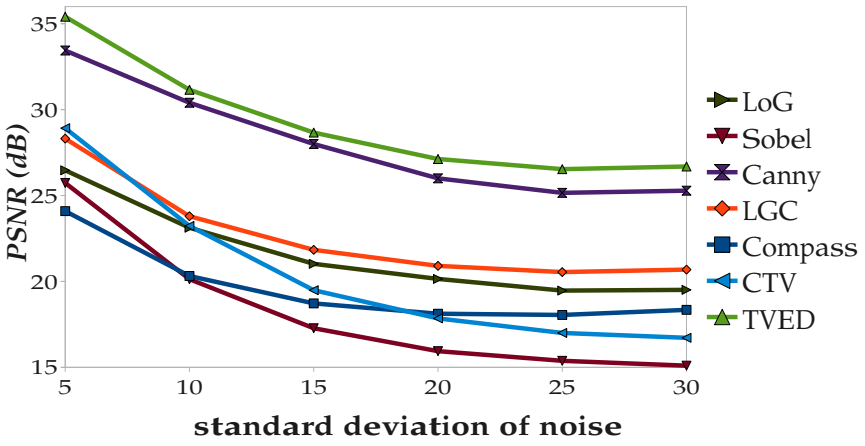


Figure 6.4: Robustness analysis.

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 (see Figures 6.5 to 6.7). 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. Figures 6.5, 6.6, and 6.7 show the edginess maps detected for nine of the tested images and their noisy counterparts[1]. 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 seen 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 stood out. 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, since they only detect the most important edges and are less influenced by noise. However, TVED generates sharper edges than Canny.

Regarding computational cost, Sobel was the fastest of all tested algorithms when run on an Intel Core 2 Quad Q6600 with a 4GB RAM (0.06 seconds), followed by Canny (0.15 seconds),

[1]Full-resolution images are available at http://deim.urv.cat/ ~ rivi/tved.html

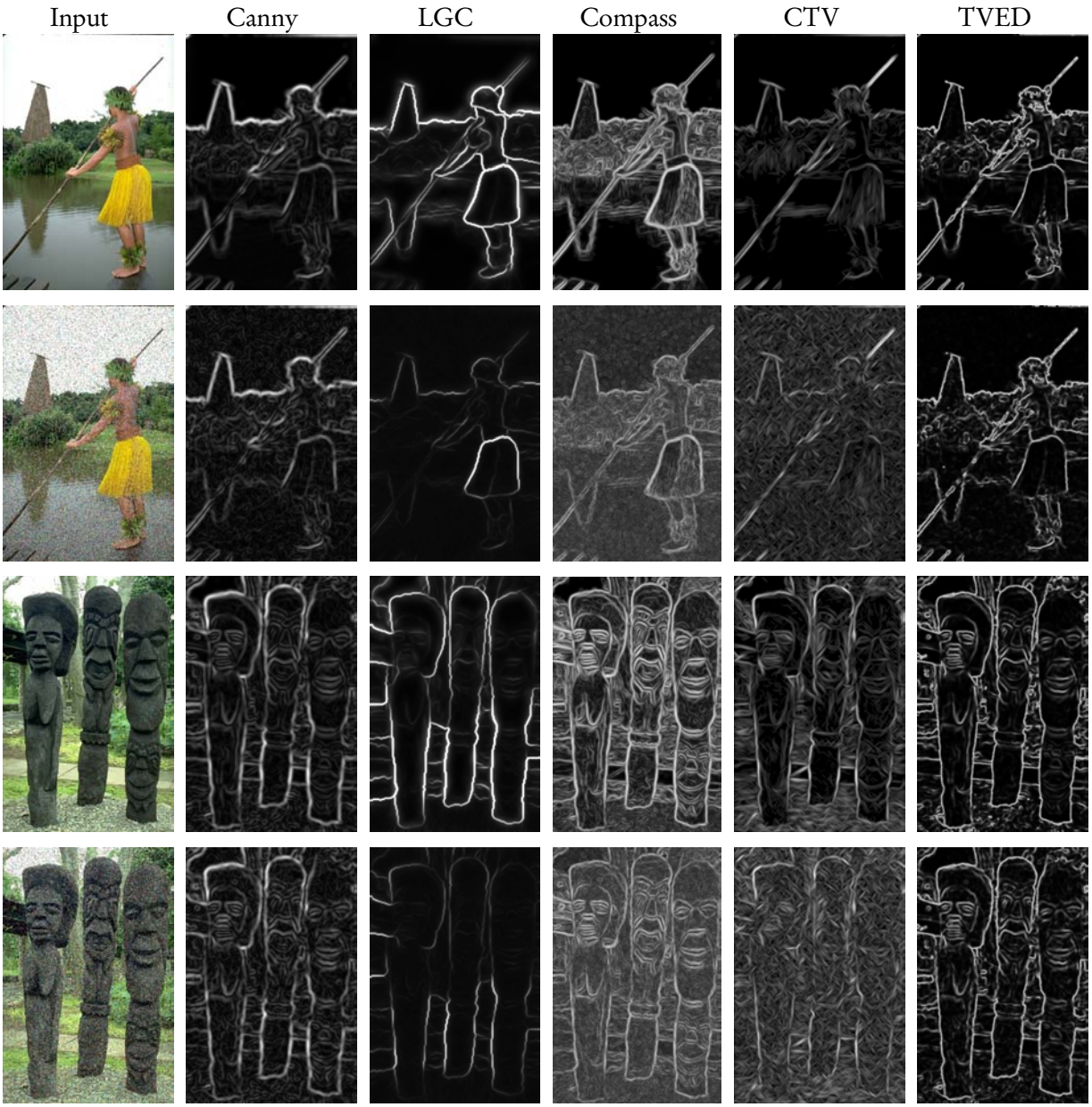


Figure 6.5: Visual comparison of results. First column: original images and their noisy counterparts. Columns three to six: edginess maps generated by the Canny, LGC, Compass, CTV and TVED methods respectively for the corresponding images.

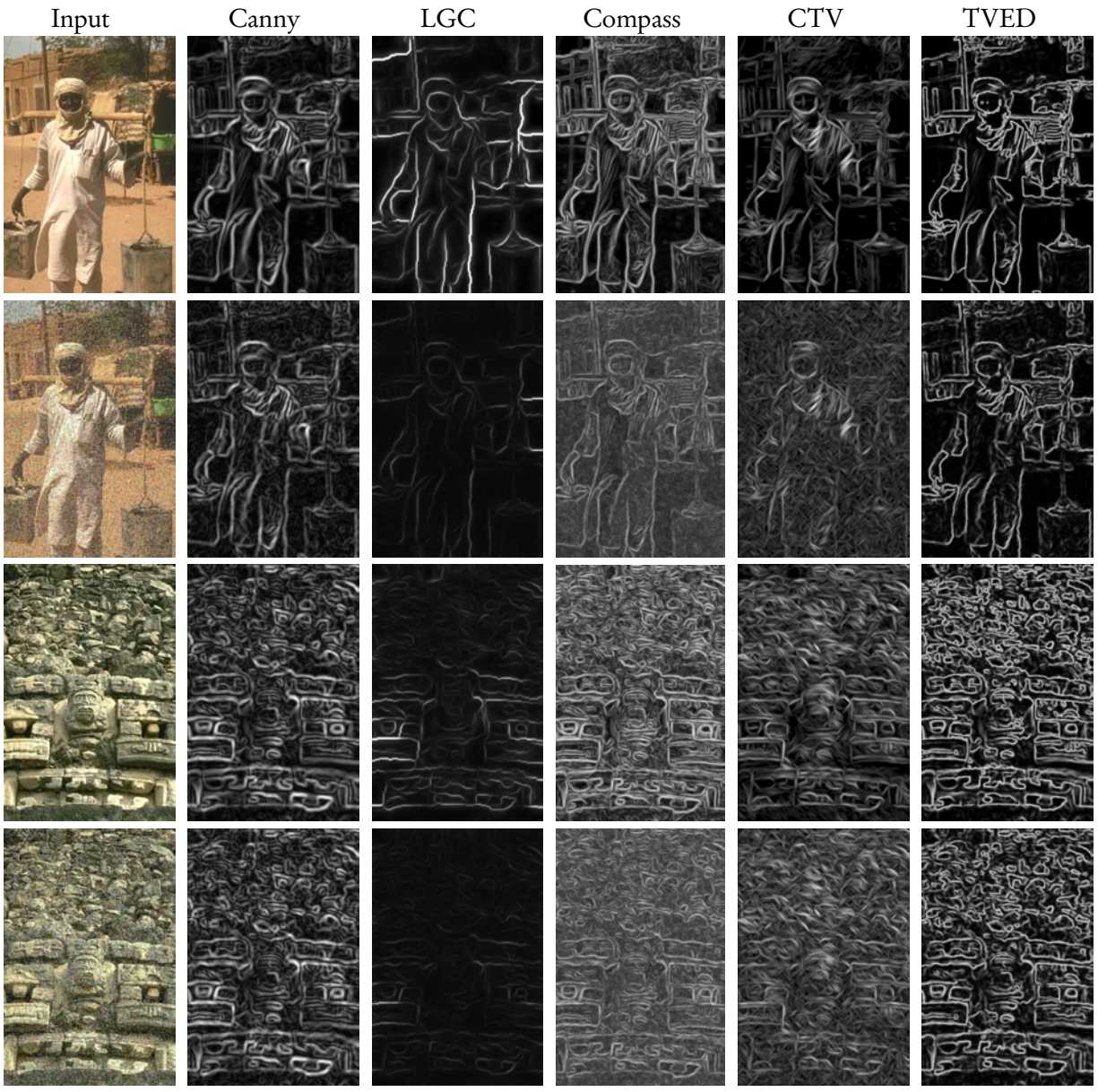


Figure 6.6: Visual comparison of results. First column: original images and their noisy counterparts. Columns three to six: edginess maps generated by the Canny, LGC, Compass, CTV and TVED methods respectively for the corresponding images.

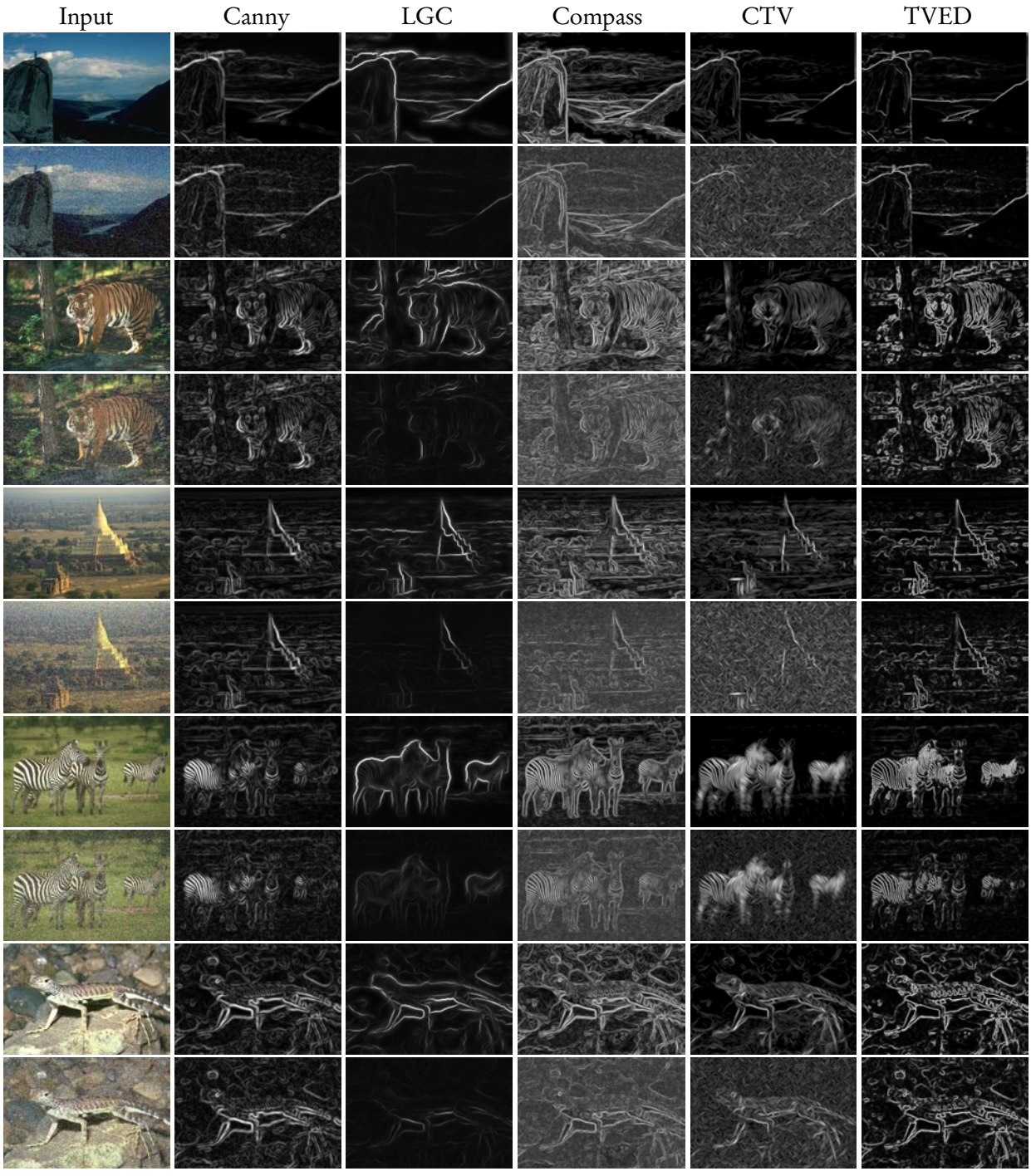


Figure 6.7: Visual comparison of results. First column: original images and their noisy counterparts. Columns three to six: edginess maps generated by the Canny, LGC, Compass, CTV and TVED methods respectively for the corresponding images.

Table 6.2: Selection of edge detection method.

Application	Feature	Best tested method
Image Segmentation	Discriminability	LGC
Image Segmentation	Precision	Compass
Image Edge Enhancement	Recall	Sobel
Image Edge Enhancement	Precision	CTV
Any	Robustness	TVED and Canny
Any	Speed	Sobel, Canny and LoG

LoG (0.35 seconds), Compass (around 20 seconds), CTV (around 25 seconds), TVED (around 40 seconds). The slowest by far was LGC (2 minutes and 36 seconds). The efficiency of the proposed methods can be further improved, since the reported execution times correspond to non-optimized implementations.

6.5 Summary

Two new methods for edge detection based on tensor voting have been proposed: the first method based on the classical tensor voting proposed by Guy and Medioni (1996, 1997), and the latter based on the method presented in Chapter 5. In addition, a set of measures for edge detection evaluation has been proposed. These measures aim at measuring the features of completeness, discriminability, precision and robustness on edginess maps. The main advantage of these new measures with respect to previous proposals is that they are directly applied to edginess maps, avoiding possible bias generated by post-processing steps.

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. 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 strenghts 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 6.2 shows some examples of which method should be chosen for some scenarios among the tested methods.

In addition, although Canny's detector still shows a competitive performance in discriminability, robustness and speed, the conducted experiments support the claims made in the last decade by many methods, in the sense that it has been outperformed, at least under some features.

Chapter 7

Tensor Voting for Robust Color Image Segmentation

Image segmentation is one of the most important stages in computer vision as a preliminary step towards further analysis and recognition stages. Its goal is to partition a given image into a set of non-overlapping homogeneous regions that likely correspond to the different objects or geometric structures that may be perceived in the scene. Researchers have used shape, gray level, color and texture information in order to estimate the homogeneity of regions. In addition, automatic and human-assisted strategies have been investigated.

Despite the huge amount of image segmentation algorithms proposed in the literature, this research area is still very active since the problem is not completely solved in general. Thus, as an alternative, this chapter proposes a new method for robust color image segmentation based on the methods presented in Chapters 5 and 6, and an efficient graph-based image segmentation technique. The strategy followed by the proposed method processes likely-homogeneous and likely-inhomogeneous pixels in a different way. Using tensor voting makes the method appropriate for noisy environments at a competitive computational cost.

The chapter is organized as follows. The next subsection reviews the most important approaches in the image segmentation field. Subsection 7.2 presents the proposed segmentation method. Subsection 7.4 shows the experimental results yielded by applying the measures introduced in Subsection 7.3. Finally, Subsection 7.5 summarizes the chapter and makes some concluding remarks.

7.1 Previous Related Work

Several reviews of image segmentation methods have been made so far and many classifications have been proposed (e.g., Pal and Pal, 1993; Skarbek and Koschan, 1994; Lakare, 2000; Pham et al., 2000; Cheng et al., 2001; Lucchese and Mitra, 2001; Freixenet et al., 2002; Vergés-Llahí, 2005; Noble and Boukerroui, 2006; Cremers et al., 2007). Based on them, the following subsections describe the most important approaches in image segmentation.

7.1.1 Thresholding

A first family groups thresholding-based segmenters. This approach classifies the pixels of gray level images into two groups: pixels whose intensity is either lower or higher than a certain threshold. This technique can be easily extended to multiple thresholds, where every cluster of pixels is located between lower and upper thresholds. The grouping of non-connected regions may be avoided by using proximity information of pixels.

Frequently, the automatic selection of thresholds is a complicated task that is performed by analyzing the intensity histogram. In color images, this technique is extended to identifying clusters in three-dimensional color histograms. The main advantages of this approach are its simplicity and effectiveness in highly contrasted scenarios. Nevertheless, the technique is not robust to noise, results are very sensitive to the selection of thresholds, automatically estimating thresholds is complicated for most applications and illumination directly affects performance. Despite that, it is often used in combination with other techniques. A review and comparison of the most relevant thresholding techniques is presented by Sezgin and Sankur (2004).

7.1.2 Clustering Algorithms

A second family of segmenters is based on clustering. The goal of this approach is to group pixels by affinity on the feature space (space formed by the relevant features extracted from the pixels and their neighborhoods) by using classical clustering techniques borrowed from machine learning, such as K-means, fuzzy c-means, artificial neural networks, Markov random fields and belief propagation (Jain et al., 1999). These methods tend to be more robust than other strategies since many of those clustering algorithms can effectively reject outliers. Nevertheless, their performance depends on how the feature space is modeled. In addition, some clustering algorithms

are highly time consuming and others need expensive training procedures.

Comaniciu and Meer (2002) proposed an efficient clustering technique in the feature space based on the mean shift algorithm. Recently, Markov random fields, multi agent theory and genetic algorithms have been put together in a clustering strategy (Melkemi et al., 2006). In addition, Randall et al. (2008) used a neural network to segment images based on a set of estimations for the Gestalt principles of similarity, closure, good continuation and co-circularity (cf. Section 2.1). More recently, Yang et al. (2008) applied a lossy data compression approach in order to cluster regions.

7.1.3 Edge-Based Approaches

A third family of segmenters groups edge-based algorithms. This approach applies edge detection techniques for subsequently grouping edges into closed contours that ultimately become borders of regions. This approach is rarely applied alone because edge detection algorithms usually generate spurious edges that must be eliminated before segmentation. Thus, as shown in Section 6.1, some edge detectors have been devised specifically to be used in image segmentation applications. Related to this approach, many perceptual organization techniques have been proposed to solve the contour completion problem, which aims at inferring object boundaries from contour fragments (cf. Section 2.3 and references therein). More recently, Arbeláez et al. (2009) have applied a hierarchical segmentation approach whose input is an edginess map.

7.1.4 Region-Based Approaches

A fourth family of segmenters groups region-based algorithms. There are two classical approaches based on regions: the first one is *region growing*, which progressively appends similar neighbors to "seed" points. The second approach is the so-called *split and merge* method, which subdivides the image into regions that are recursively merged or split according to a similarity function. A common strategy to split data consists of using *quadtrees* in 2D and *octrees* in 3D. The main advantage of these strategies is that similarity can be modeled in a high variety of ways. Nevertheless, they are not robust to noise and it is necessary to choose appropriate "seeds" in the region growing approach. These methods and their variants are presented in any textbook (e.g., Gonzalez and Woods, 2007).

The *watershed transform* has been one of the most relevant region-based techniques over the

last years. The watershed transform is a mathematical morphology technique that, based on the gray-level gradient of the image, emulates a flooding generated from a set of water sources located at some "seed" points on a terrain surrounded by mountains represented by high changes in gradient. A pixel belongs to an edge if it is touched by two or more sources of water. The most important disadvantages of the original approach are its sensitivity to noise and the need for defining "seed" points. However, many variants have been proposed in order to overcome these problems. Recently, Jung (2007) has applied the watershed transform for segmenting images through a multiscale approach. In 3D, the watershed transform has also been applied in order to segment surface meshes (e.g., Mangan and Whitaker, 1999). A review of the watershed transform and its variants is presented by Roerdink and Meijster (2000).

7.1.5 Graph-Based Approaches

A fifth family of segmenters groups graph-based methods. This approach models images with graphs in order to segment them by applying graph cut algorithms. For instance, Hagen and Kahng (1992); Shi and Malik (2000); Bühler and Hein (2009) have approximated the *NP*-complete problem of graph weighted cutting by means of spectral graph theory (Chung, 1997). Dhillon et al. (2007) have found that most of these graph cutting strategies can be modeled through a generalized version of k-means, leading to more efficient implementations.

More recently, Cousty et al. (2009) have extended the ideas of the watershed transform to graph cutting. Grady (2006) used random walks, graphs and potential theory to segment images in supervised environments. Using other approaches, Felzenswalb and Huttenlocher (2004) proposed a greedy region growing strategy by means of graphs and Tao et al. (2007) have post-processed segmentations obtained through mean shift with graph-based strategies.

The main advantage of these methods is that they can be applied without modifications to any type of feature space since their input is a graph. Nevertheless, their performance depends on the graph creation process and some algorithms can be highly time consuming.

7.1.6 Deformable Models

A sixth family of segmenters is based on deformable models. These models are based on variational methods that aim at segmenting images by deforming initial closed contours until the boundaries of the regions are reached. Contour deformation is guided by energy minimization

functions or probabilistic models. Active contours (Kass et al., 1988), the Mumford-Shah variational model (Mumford and Shah, 1989), and level set (Osher and Sethian, 1988) methods belong to this category of segmenters.

Unlike other strategies that use discrete representations of space, these algorithms are based on spatially continuous representations. The following two problems are faced when using these methods: sensitivity to initial surrounding contour selection (usually difficult to calculate) and difficulty to extend them to color or texture segmentation. Detailed reviews of deformable strategies are presented in Bresson (2005, Chapter 2) and Cremers et al. (2007).

7.1.7 Physics-Based Approaches

A seventh family of segmenters groups physics-based algorithms. In a first stage, this approach models the interaction of light with the materials present in the scene and with the camera at multiple viewing conditions in order to characterize every material by means of its expected appearance on images at various light stimuli and viewing conditions. In a second stage, regions are classified by comparing observed appearances of the materials to expected ones. Thus, this approach models segmentation as the inverse problem of scene rendering.

An advantage of this approach is that shadows and highlights can be eliminated, leading to better segmentations. The main problem with this strategy is that knowledge of properties of materials is required, limiting its application to controlled environments. This approach is generally combined with other methods for segmenting images, especially with edge-based and clustering algorithms. The most representative methods are the dichromatic reflection model described by Ong and Matsuyama (1998) and the approximate color-reflectance model proposed by Healey (1992).

7.1.8 Soft Segmenters

An eighth family of segmenters groups the so-called soft segmenters. These methods estimate for every pixel a degree of membership for every detected region. Similarly to the edge detection case, hard segmentations can be obtained by post-processing membership maps through thresholding or a more sophisticated algorithm. In this field, for example, Prewer and Kitchen (2001) have used graph theory, and Shen (2006) has applied the Mumford-Shah variational model. More recently, Tai et al. (2007) have proposed a probabilistic framework based on a Gaussian mixture

model that follows this approach. It is interesting to remark that, in practice, the membership degrees computed by soft segmenters only make sense for pixels close to borders.

7.1.9 Model Guided Segmentation

Finally, a new approach tackles both problems, image segmentation and object recognition, in a unified framework instead of processing them sequentially. This approach is based on psychophysical evidence supporting that object recognition and image segmentation processing could be carried out concurrently by the HVS (Peterson and Gibson, 1994). This has led to differentiating segmentation approaches based on object recognition techniques, called top-down techniques, from others, called bottom-up techniques.

Instead of only applying top-down or bottom-up segmenters, researchers have found more adequate to integrate both strategies. In this line, Borenstein and Ullman (2008) have used a bank of fragments of images to make an initial segmentation of the image that is then refined through the graph-based segmentation technique by Sharon et al. (2006). Using a similar approach, Levin and Weiss (2009) use conditional random fields for querying the bank of fragments. The most important drawback of this approach is the need for a training step and/or the use of ground-truths, which are not available in some applications.

7.1.10 Hybrid Approaches

In addition to the reviewed families, hybrid strategies have also been proposed. These strategies aim at obtaining a better performance by taking advantage of the individual strengths of a set of segmenters, for example, methods that integrate both region and edge-based approaches to segment images are reviewed by Muñoz et al. (2003). These algorithms tend to behave better than their base algorithms on average and worse in specific situations due to the overhead caused by the required processes of selection and coordination of their base algorithms. A recent study merged five different strategies, only obtaining better results than the base strategies on images that contain a few regions (Ge et al., 2007). In the same line, Shah (2008) proposed to select the most appropriate segmenter from a bank of algorithms, based on the particular properties of every image. A different strategy has used an unsupervised evaluation of sub-optimal segmentation results yielded by a segmenter algorithm in order to automatically tune its parameters (Pichel et al., 2006).

Another approach followed by researchers is to apply image pre-processing before the segmentation step (Weickert, 2001; Park et al., 2008), or to apply post-processing onto sub-optimal segmentation (Bagon et al., 2008) in order to improve the performance of the algorithms.

7.1.11 Segmentation Using Color and Multiple Cues

Most of the color image segmentation methods can be seen as extensions of their gray-level counterparts. However, two main considerations have been taken into account when using color information for segmenting images: the selection of an appropriate color model for each specific application, and how to work on the color channels: independently or jointly. The first strategy applies segmentation algorithms to each channel and then uses a strategy to merge the results. The second approach directly works on the feature space to segment images. Comprehensive reviews of color segmentation techniques that follow these strategies are given by Skarbek and Koschan (1994); Cheng et al. (2001).

On the other hand, many methods that use more than a single cue for segmenting images have been proposed. Since evidence indicates that color and texture are not strongly correlated (Mäenpää and Pietikäinen, 2004), many researchers have proposed methods that use these two cues to segment images. Mirmehdi and Petrou (2000) filtered images by using the kernels proposed in the S-CIELAB color model (Zhang and Wandell, 1997; Johnson and Fairchild, 2003a) at multiple viewing distances. An initial pixel clustering estimated at the coarsest image is refined with finer resolutions through probabilistic relaxation. The approach by Chen et al. (2005) initially processes color and texture independently in order to merge the results by prioritizing color segmentation in non-textured regions and, in a second step, texture segmentation in textured ones.

Deng and Manjunath (2001) proposed the so-called *JSEG* method based on color quantization followed by multiscale spatial segmentation. The *Edgeflow* method described by Ma and Manjunath (2000) estimates at each pixel the flow direction that points to the closest boundary in order to detect edges where two opposite directions of edge flow appear. Flow direction is calculated by using a linear combination of energy estimations on every color channel and texture features. A robust version of this method has been recently proposed (Ghosh et al., 2010). Freixenet et al. (2004) have applied independent texture-based and color-based segmentation algorithms which are then merged using kernel density estimation. Gevers (2002) used a supervised strategy to

extract color textured regions in images. Kang and Medioni (2001) jointly encoded color and position into tensors and performed 2D image segmentation by applying tensor voting in 5D.

7.1.12 Robust Image Segmentation

Although most image segmentation methods have not been devised to deal with noise, in practice, some of them yield good results for noisy images. This can be attributed to the use of texture features that make them less prone to erroneously segmenting noisy scenarios. For instance, although the methods by Dhillon et al. (2007); Rao et al. (2009) were not proposed for noisy scenarios, they yield robust results (cf. Section 7.4). Similarly, Weickert (2001) has found that results can be improved by applying prefiltering steps before segmenting the images.

A different strategy is based on robust variants of existing methods. For instance, variants of the so-called fuzzy c-means, which can be interpreted as the fuzzy variant of the well-known k-means clustering algorithm, has shown state-of-the-art properties (e.g., Wang and Bu, 2010). In addition, Ghosh et al. (2010) have proposed a robust version of edge flow, and Ali (2009) has used genetic algorithms for robust medical image segmentation.

7.1.13 Tensor Voting for Color Image Segmentation

Tensor voting has already been used for image segmentation. For example Kang and Medioni (2001) segment color images by encoding the position and RGB color of every pixel into tensors of five dimensions before applying tensor voting. Unfortunately, this approach has limitations on noisy images, mainly due to the fact that the classical tensor voting has not been designed for dealing with noisy color images, as shown in Chapter 5.

Following a similar approach, Park et al. (2006, 2007b,a, 2008) and Lim et al. (2007), modeled color through a single value before applying tensor voting in 3D in order to detect outliers that are then filtered out through non-tensor voting techniques. The filtered image is then segmented through standard image segmenters, such as mean-shift, k-means or statistical methods in order to extract text from the image.

Unlike these methods, the technique proposed in the following section is robust to noise, and it uses tensor voting not only in a preprocessing step, but also throughout the main segmentation process.

7.2 Tensor Voting for Robust Color Image Segmentation

The proposed method can be seen as a robust variant of an efficient region growing strategy. The main advantage of region growing approaches is their efficiency. Although region growing approaches usually have a good performance in homogeneous regions, they are also prone to errors mainly attributed to erroneous processing of inhomogeneous regions. For example, using pixels at borders as growing seeds is inconvenient, since they can lead to dummy regions generated from similar pixels close to borders (Muñoz et al., 2003).

In addition, these methods are usually unable to segment noisy images effectively. This is due to the difficulty to distinguish between edges and noise. However, using robust edge detectors, such as those presented in Chapter 6, could partially avoid this problem.

Moreover, results from soft segmenters suggest that hard segmentation, such as region growing, is suitable for most parts of the image, while pixels close to borders should be processed in a different way. Hence, the proposed method processes pixels within homogeneous regions and pixels at borders in a different way, by taking into account their different nature.

The proposed robust color image segmentation method is based on the methods presented in Chapters 5 and 6 and on the efficient segmentation algorithm presented by Felzenswalb and Huttenlocher (2004). On the one hand, TVD (cf. Chapter 5) is used to filter out noise in order to improve the segmentation results, following the same strategy as Weickert (2001). On the other hand, edginess maps generated through TVED (cf. Chapter 6) are used to distinguish between likely-homogeneous and likely-inhomogeneous pixels. TVED is preferred to CTV in this task, since it does not require extra calculations thanks to the fact that both TVD and TVED belong to the same unified framework, and it has also yielded more robust results than CTV (cf. Section 6.4). The efficient method described in Felzenswalb and Huttenlocher (2004) is used to cluster pixels through a region growing strategy.

The proposed method comprises four stages. At a first stage, a graph is created in a similar way as in Felzenswalb and Huttenlocher (2004) from the filtered image processed through TVD. Afterwards, pixels are classified into likely-homogeneous and likely-inhomogeneous with the help of the edginess map calculated through TVED. Likely-homogeneous pixels are clustered with the graph-based segmenter of Felzenswalb and Huttenlocher (2004) in a third stage. In the final stage, likely-inhomogeneous pixels are aggregated to the presegmented image through a modified version of the same segmenter used in the third stage. These stages are explained in

more detail below.

Every pixel corresponds to a vertex in the graph created at the first stage, while every pair of neighboring pixels leads to an edge whose weight is given by the color difference between those pixels. In addition, local edginess is also stored at every edge of the graph.

Likely-inhomogeneous pixels can be extracted by thresholding the edginess map. This procedure has the advantage that not only edges are classified as likely-inhomogeneous, but also a strip of pixels at both sides of edges. Previous region growing methods are particularly prone to errors in those strips. Edginess maps obtained through robust techniques, such as tensor voting, are required for this stage, since they are the only way to correctly classify noisy pixels as likely-homogeneous in noisy scenarios.

The third and fourth stages of the proposed method are based on the graph-based segmenter described in Felzenswalb and Huttenlocher (2004), which can be summarized as follows. Let $H(\mathbb{R})$ be the internal difference of region $\mathbb{R}$, defined as the largest weight of the edges in its minimum spanning tree, and $\tau(\mathbb{R}) = k/|\mathbb{R}|$, where k is a parameter. The segmenter follows the next stages:

1. the graph edges are sorted in ascending order of weight.

2. a different region is created for every vertex in the graph.

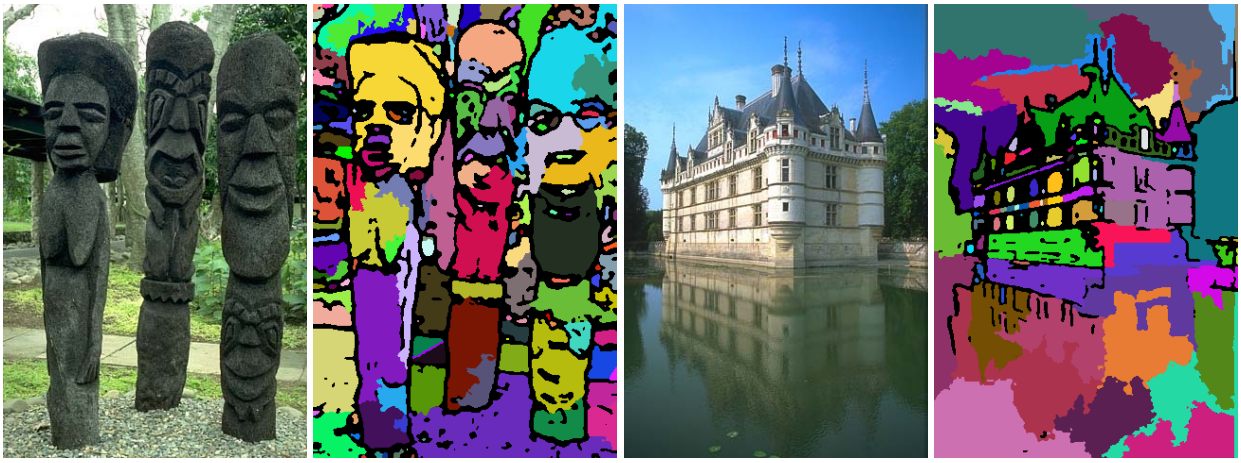
3. for each sorted edge E, the following region growing strategy is applied: let $\mathbb{R}_i$ and $\mathbb{R}_j$ be the neighboring regions connected in the graph by means of E and w its weight. If $w \leq D(\mathbb{R}_i, \mathbb{R}_j)$, with:

$$D(\mathbb{R}_i, \mathbb{R}_j) = \min(H(\mathbb{R}_i) + \tau(\mathbb{R}_i), H(\mathbb{R}_j) + \tau(\mathbb{R}_j)), \tag{7.1}$$

   both regions are merged and $H(\mathbb{R}_i \cup \mathbb{R}_j)$ is updated with w in such a case.

In the third stage of the method, this graph-based segmenter is applied to the edges of the graph that do not connect likely-inhomogeneous pixels. After that, post-processing can be optionally applied in order to remove small islands surrounded by big regions. Figure 7.1 shows an example of the results yielded after this third stage.

A soft segmentation can be obtained by calculating a membership degree from the distance between every likely-inhomogeneous pixel to its nearest regions obtained after the third stage. However, a hard segmentation can also be obtained by applying the fourth stage of the method.



(a) (b) (c) (d)

Figure 7.1: (a-c) Original images. (b-d) Results of the method after the third stage of the proposed method. Black pixels represent likely-inhomogeneous pixels, which are not processed until the fourth stage of the method.

Finally, the fourth stage joins likely-inhomogeneous pixels to the most likely region obtained in the third stage through the same segmenter, which is applied with the following four variations. First, the remaining edges of the graph are sorted by edginess instead of weight, since the former is more relevant than the latter for inhomogeneous pixels. However, weight is still used to merge regions, since it encodes similarity among pixels. Second, a pixel must be joined to a preexistent region in order to avoid the creation of dummy regions. Third, function $H(\mathbb{R})$ is not updated when a pixel is joined to region $\mathbb{R}$ in order to give more relevance to the similarity of neighboring pixels than that of neighboring regions, which is essential in the third stage. Finally, the process is iterated until both all the graph's edges and all the likely-inhomogeneous pixels have been processed.

7.3 Evaluation Methodology

Evaluating image segmentation algorithms is a difficult task that has fostered research during the last decades. One of the problems faced by assessment methodologies is that the criteria of a good segmentation are application-dependent (Zhang et al., 2008). In addition, similarly to edge detectors, it is difficult to select a single measure to evaluate image segmentation techniques, since most of these measures look for different properties of the segmentations yielded by the tested methods.

Roughly speaking, there are two approaches for evaluating image segmentation algorithms: supervised and unsupervised evaluation depending on whether or not the results are compared to ground-truths (Zhang et al., 2008). The following two subsections review these strategies. The third subsection proposes two new measures to assess under and oversegmentation.

7.3.1 Unsupervised Approaches

Unsupervised assessment approaches are based on measuring how a segmentation fits a set of desirable properties that an ideal segmentation must have (Zhang, 1996). Unlike supervised approaches, these measures do not require ground-truths. This seems especially helpful as having a single valid ground-truth is impossible, since every human can make a different ground-truth.

In this line, Haralick and Shapiro (1985) proposed the following four criteria to assess the performance of an image segmentation algorithm:

1. Extracted regions must be uniform under a set of features.
2. Adjacent regions must have significant differences compared to each other.
3. The shape of regions should be simple (maybe not containing too many holes).
4. Boundaries should be simple.

Most of the unsupervised measurement measures are based on one or more of these criteria. Unfortunately, the majority of these measures have only been proven effective to compare different parametrizations of the same segmentation algorithm (Zhang et al., 2008).

7.3.2 Supervised Approaches

Most of the supervised methods are based on the so-called confusion matrix, also known as the contingency matrix. It can be defined as follows. Let a segmentation $\mathscr{C}$ be a set of M disjoint regions, and $|\mathbb{R}|$ be the cardinality of a region $\mathbb{R}$. A confusion matrix T between two segmentations $\mathscr{C}_1$ and $\mathscr{C}_2$ is an $M_1 \times M_2$ matrix in which every T_{ij} entry corresponds to the number of pixels that are in both the i-th region from segmentation $\mathscr{C}_1$, $\mathbb{R}_{1i}$, and the j-th region from segmentation $\mathscr{C}_2$, $\mathbb{R}_{2j}$. For evaluation purposes, $\mathscr{C}_1$ stands for the segmentation yielded by the algorithm, whereas $\mathscr{C}_2$ stands for the ground-truth (usually human-made) segmentation. The next four values can be extracted from the confusion matrix:

1. A_{11}: the number of pairs of pixels that were segmented into the same region in both $\mathscr{C}_1$ and $\mathscr{C}_2$. It is given by:

$$A_{11} = \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \frac{\mathrm{T}_{ij}(\mathrm{T}_{ij} - 1)}{2}, \tag{7.2}$$

2. A_{10}: the number of pairs of pixels that were segmented into the same region in $\mathscr{C}_1$ and into a different region in $\mathscr{C}_2$. It is given by:

$$A_{10} = \sum_{i=1}^{M_1} \frac{|\mathbb{R}_{1i}|(|\mathbb{R}_{1i}| - 1)}{2} - A_{11}, \tag{7.3}$$

3. A_{01}: the number of pairs of pixels that were segmented into the same region in $\mathscr{C}_2$ and into a different region in $\mathscr{C}_1$. It is given by:

$$A_{01} = \sum_{j=1}^{M_2} \frac{|\mathbb{R}_{2j}|(|\mathbb{R}_{2j}| - 1)}{2} - A_{11}, \tag{7.4}$$

4. A_{00}: the number of pairs of pixels that were segmented into a different region in both $\mathscr{C}_1$ and $\mathscr{C}_2$. It is given by:

$$A_{00} = \frac{N(N-1)}{2} - A_{11} - A_{10} - A_{01}, \tag{7.5}$$

with N being the number of pixels in the image.

Many measures have been proposed based on these four values. Two of the most popular are the Rand index (Rand, 1971), given by:

$$RI(\mathscr{C}_1, \mathscr{C}_2) = \frac{A_{11} + A_{00}}{A_{11} + A_{01} + A_{10} + A_{00}}, \tag{7.6}$$

and the Jaccard index (Jaccard, 1901), given by:

$$JI(\mathscr{C}_1, \mathscr{C}_2) = \frac{A_{11}}{A_{11} + A_{01} + A_{10}}. \tag{7.7}$$

There are some known problems with these classical measures. For example, the Rand Index does not yield a constant value for two random segmentations. Thus, some improved measures have been proposed, such as the adjusted Rand index (Hubert and Arabie, 1985; Santos and Embrechts, 2009), the probabilistic Rand index (Unnikrishnan and Hebert, 2005), and the normalized probabilistic Rand index (Unnikrishnan et al., 2007). As an example, the probabilistic

Rand index (PRI) of a segmentation $\mathscr{C}$ with respect to a set of ground-truth segmentations, $\mathscr{H}$, is given by:

$$PRI(\mathscr{C},\mathscr{H}) = \frac{1}{K}\sum_{k=1}^{K} RI(\mathscr{C},\mathscr{H}_k). \tag{7.8}$$

is more appropriate when multiple equally reliable ground-truths are available.

More recently, Brouwer (2009) has proposed fuzzy extensions of these measures for evaluating soft segmentation algorithms. Furthermore, Ortiz and Oliver (2006) have proposed to compute the percentage of correctly grouped pixels (CG) and under (US) and oversegmentation (OS) measurements directly from the confusion matrix, without computing the aforementioned four values. These measures are given by:

$$CG(\mathscr{C}_1,\mathscr{C}_2,p) = \frac{1}{N}\sum_{i=1}^{M_1}\sum_{j=1}^{M_2} V_1(\mathbb{R}_{2j},p)\,\mathrm{T}_{ij} \tag{7.9}$$

$$OS(\mathscr{C}_1,\mathscr{C}_2 p) = \frac{1}{N}\sum_{i=1}^{M_1}(1-V_2(\mathbb{R}_{2j},p))\,|\mathbb{R}_{2j}| \tag{7.10}$$

$$US(\mathscr{C}_1,\mathscr{C}_2 p) = \frac{1}{N}\sum_{j=1}^{M_2}(1-V_3(\mathbb{R}_{1i},p))\,|\mathbb{R}_{1i}| \tag{7.11}$$

with

$$V_1(\mathbb{R}_{2j},p) = \begin{cases} 1 & \text{if } \dfrac{\mathrm{T}_{ij}}{|\mathbb{R}_{2j}|} \geq p \\ 0 & \text{otherwise} \end{cases} \tag{7.12}$$

$$V_2(\mathbb{R}_{2j},p) = \begin{cases} 1 & \text{if } \dfrac{\max\limits_{j=1..M_2}\{\mathrm{T}_{ij}\}}{|\mathbb{R}_{2j}|} \geq p \\ 0 & \text{otherwise} \end{cases} \tag{7.13}$$

$$V_3(\mathbb{R}_{1i},p) = \begin{cases} 1 & \text{if } \dfrac{\max\limits_{i=1..M_1}\{\mathrm{T}_{ij}\}}{|\mathbb{R}_{1i}|} \geq p \\ 0 & \text{otherwise} \end{cases} \tag{7.14}$$

The main drawback of these measures is that they are computed with respect to parameter p, whose selection seems unclear.

Another popular performance measurement is the so-called Global Consistency Error (GCE), proposed by Martin et al. (2001), which intends to measure performance by comput-

ing the degree of overlap between every pair of regions. GCE is given by:

$$GCE(\mathscr{C}_1, \mathscr{C}_2) = \frac{1}{N} \min \left\{ \sum_{k=1}^{N} E(\mathscr{C}_1, \mathscr{C}_2, x_k) \right\}, \tag{7.15}$$

where x_k is the k-th pixel of the image, and E is computed as:

$$E(\mathscr{C}_1, \mathscr{C}_2, x_k) = \frac{|\mathbb{R}_{1k} \setminus \mathbb{R}_{2k}|}{|\mathbb{R}_{1k}|}, \tag{7.16}$$

with $\mathbb{R}_{ck}$ being the region where x_k is located in segmentation c. More recently, methods by Cardoso and Corte-Real (2005, 2006) have been proposed as an alternative to GCE, but with similar results.

Using a different approach, Meilă (2007) has proposed the so-called Variation of Information measure, VI, which is based on the computation of the conditional entropies between both segmentations:

$$VI(\mathscr{C}_1, \mathscr{C}_2) = -\frac{1}{N} \sum_{i=1}^{M_1} |\mathbb{R}_{1i}| \log \frac{|\mathbb{R}_{1i}|}{N} - \frac{1}{N} \sum_{j=1}^{M_2} |\mathbb{R}_{2j}| \log \frac{|\mathbb{R}_{2j}|}{N} - 2 \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} P_{ij} \log \frac{N^2 P_{ij}}{|\mathbb{R}_{1i}||\mathbb{R}_{2j}|} , \tag{7.17}$$

with $P_{ij} = |\mathbb{R}_{1i} \cap \mathbb{R}_{2j}|/N$. A disadvantage of this measure is that it is not in the range between zero and one.

7.3.3 New Measures for Assessing Under and Oversegmentation

This dissertation proposes two new measures, $OSEG$ and $USEG$, which aim at measuring over- and undersegmentation respectively in a non-parametric way. These measures are calculated from the confusion matrix T. The oversegmentation measure with respect to a specific region of the ground-truth is calculated as the mean of the squared relative areas of the intersecting regions in the segmented image. In turn, the global oversegmentation measure, $OSEG$, weights every regional oversegmentation estimation by the size of every region in the ground-truth. More formally:

$$OSEG(\mathscr{C}_1, \mathscr{C}_2) = 1 - \frac{1}{N} \sum_{j=1}^{M_2} |\mathbb{R}_{2j}| \sum_{i=1}^{M_1} \left(\frac{\mathrm{T}_{ij}}{|\mathbb{R}_{2j}|} \right)^2 . \tag{7.18}$$

The global undersegmentation measure, $USEG$, is calculated in the same way as $OSEG$ by exchanging the ground-truth and the segmented image:

$$USEG(\mathscr{C}_1, \mathscr{C}_2) = 1 - \frac{1}{N} \sum_{i=1}^{M_1} |\mathbb{R}_{1i}| \sum_{j=1}^{M_2} \left(\frac{\mathrm{T}_{ij}}{|\mathbb{R}_{1i}|} \right)^2 . \tag{7.19}$$

Both $OSEG$ and $USEG$ range from zero to one. $OSEG$ is a measure of how often a region in the ground-truth is split into two or more regions in the segmented image. Thus, a segmentation with $OSEG$ equals to zero means that it is equivalent to the ground-truth, so it has a perfect performance. In addition $1/(1-OSEG)$ can be interpreted as an estimate of the average number of regions of the segmented image that constitute a region in the ground-truth. In turn, $USEG$ measures how often a region in the segmented image has mistakenly been merged to two or more regions of the ground-truth. Similarly to $OSEG$, a segmentation with $USEG$ equals to zero means that it is equivalent to the ground-truth, so it has a perfect performance. Likewise, $1/(1-USEG)$ can be used to estimate the average number of regions of the ground-truth that are mistakenly merged by the segmentation method.

The main advantage of these measure with respect to the related measures proposed by Ortiz and Oliver (2006) (cf. Subsection 7.3.2) is that the latter require a parameter, while the proposed ones do not.

7.4 Experimental Results

A total of fifteen images from the Berkeley segmentation data set (Martin et al., 2001) and their corresponding ground-truths have been used in the experiments. Mean shift (MS) (Comaniciu and Meer, 2002), the method by Felzenswalb and Huttenlocher (2004), referred to as FH, Graclus (GC) (Dhillon et al., 2007), and the method by Rao et al. (2009), referred to as TBES, have been used in the comparisons, since they are representative of the state-of-the-art in color image segmentation. The default parameters of MS (undersegmentation), FH, and TBES have been used. TVD and TVED have been run with the parameters of the experiments in Chapter 6. GC has been run with 20 clusters, as recommended by Martin et al. (2001). Parameter k has been experimentally set to 800, and edginess maps have been thresholded for values greater than 20% of the maximum for the proposed method, referred to as RIS. The methods have been applied to noiseless and noisy images contaminated with AWGN with a standard deviation σ of 30.

In addition to visual inspection, performance has been evaluated by means of the measure PRI by Unnikrishnan and Hebert (2005), GCE by Martin et al. (2001), VI by Meilă (2007) and the new proposed $OSEG$ and $USEG$ measures (cf. Section 7.3). Higher values are better for PRI, whereas lower values are better for GCE, VI, $OSEG$ and $USEG$. Table 7.1 shows that GC, TBES and RIS are robust since they obtain similar scores for both noiseless ($\sigma = 0$)

Table 7.1: Average performance measurements for the tested images

Amount of AWGN (σ)	Measure	MS	FH	GC	TBES	RIS
0	PRI	0.73	0.75	0.73	0.77	**0.78**
	GCE	**0.12**	0.20	0.22	0.31	0.16
	VI	5.06	1.96	3.43	**1.94**	2.84
	$OSEG$	0.82	**0.31**	0.79	0.38	0.61
	$USEG$	**0.09**	0.27	0.22	0.29	0.16
30	PRI	0.54	0.57	0.74	**0.79**	**0.79**
	GCE	0.31	**0.11**	0.22	0.32	0.17
	VI	2.73	2.04	3.36	**1.84**	2.95
	$OSEG$	0.34	**0.12**	0.78	0.36	0.61
	$USEG$	0.38	0.45	0.22	0.30	**0.17**

and noisy ($\sigma = 30$) images for all the applied measures. On the contrary, MS and FH face high variations when applied to noisy scenarios. A more detailed view of these results shows that RIS has the best performance in PRI in both scenarios when compared to the other methods. In addition, RIS is also the best among robust algorithms under GCE. Moreover, TBES was the best algorithm under VI, closely followed by FH. The worst algorithm was GC. Furthermore, FH was the algorithm with the best performance in oversegmentation. Among the robust methods, TBES was the algorithm that generates less oversegmentation. Finally, RIS is the most consistent algorithm to avoid undersegmentation.

Figures 7.2 and 7.3 show the segmentation results for noiseless images and their noisy counterparts respectively[1]. Visual comparison confirms that GC, TBES and RIS are robust since they obtain similar segmentations for both noiseless ($\sigma = 0$) and noisy ($\sigma = 30$) images. MS and FH are more affected by noise. MS and FH have a poor performance, since they mistakenly oversegment some areas (e.g., the grass) and undersegment others (e.g., MS is unable to segment the noisy snake). GC yields better results but its performance depends on the number of regions in the image, which is usually unknown a priori. TBES tends to mistakenly undersegment the

[1]Full-resolution images are available at http://deim.urv.cat/ ~ rivi/ris.html

images (e.g., it integrates the wolf with the background and is unable to segment the snake and statue in the wall). Hence, RIS succeeds in attaining a good balance between under and oversegmentation.

Regarding computational cost, MS and FH were the fastest among all tested algorithms when run on an Intel Core 2 Quad Q6600 with 4GB RAM (less than 100ms), GC took around 80 seconds, and TBES was the slowest with 22 minutes and 35 seconds for every image. The efficiency of RIS is mainly determined by TVD and TVED (around 40 seconds), since the segmentation step of RIS took less than 100ms. As stated in Chapters 5 and 6, the efficiency of the proposed method can be further improved, since the reported execution time correspond to non-optimized implementations.

7.5 Summary

A new method for robust color image segmentation has been presented. The tensors obtained by applying TVD and TVED (cf. Chapters 5 and 6) are used to classify pixels into likely-homogeneous and likely-inhomogeneous. Those pixels are then segmented through a variation of the method proposed by Felzenswalb and Huttenlocher (2004).

Furthermore, two new measures for assessing the degree of over- and undersegmentation of a given segmentation with respect to a ground-truth have been proposed. The main advantage of these new measures is that they do not require any parameters, unlike previous proposals.

Experiments show that the proposed algorithm has better scores in three of the five applied evaluation measures when compared to robust segmentation methods from the state-of-the-art. In addition, the new method also has a competitive computational cost.

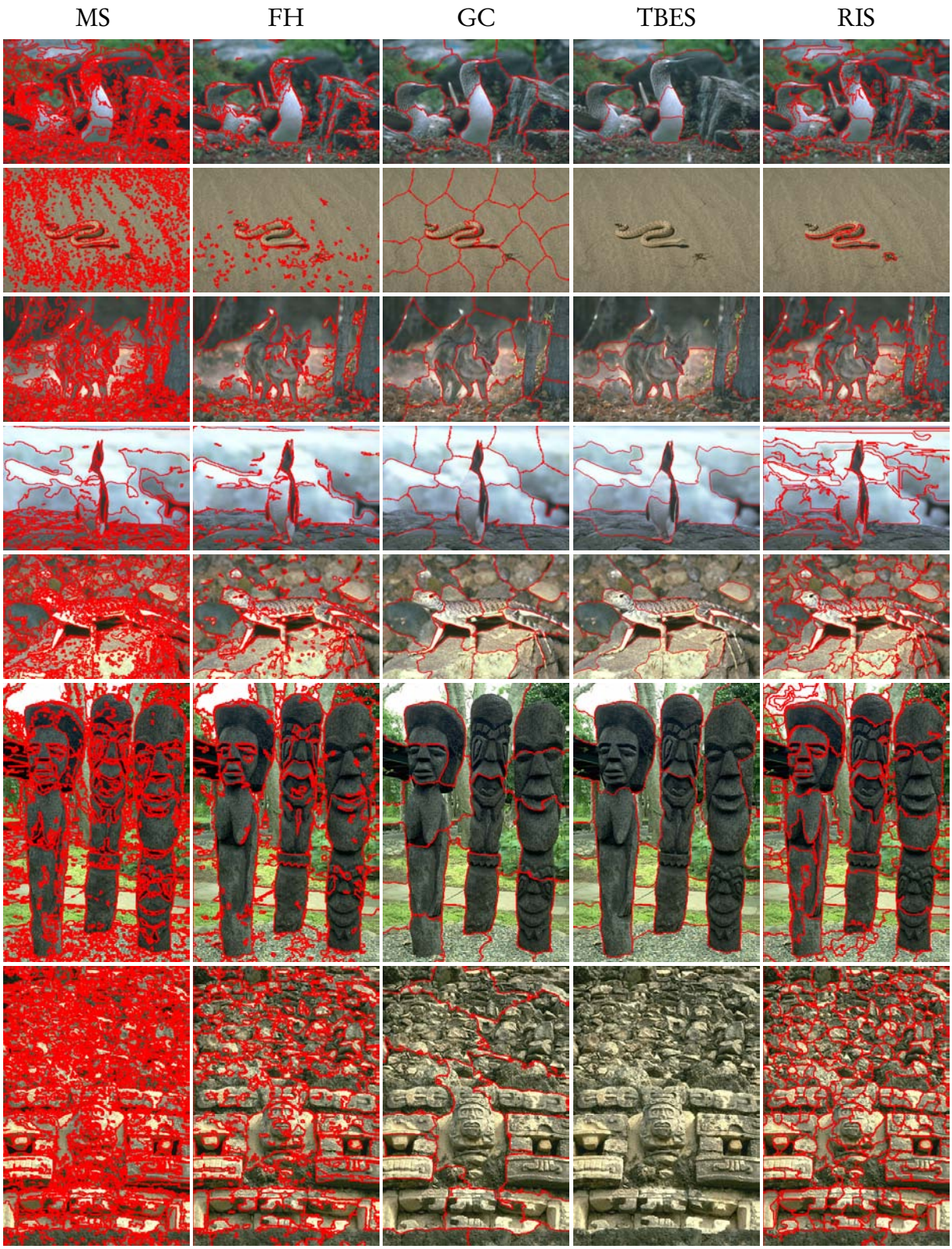


Figure 7.2: Results for images without noise. Borders are marked in red.

Figure 7.3: Results for images with AWGN ($\sigma = 30$). Borders are marked in red.

Part III

CONCLUSIONS

Chapter 8

Concluding Remarks

This chapter presents the final remarks of this dissertation. It is organized as follows. Section 8.1 summarizes the contributions made throughout the development of this thesis. Section 8.2 proposes future lines of research that arise from this work. Finally, Section 8.3 lists the publications that have been derived from this thesis.

8.1 Contributions

The following contributions to the fields of image processing and computer vision have been made in this dissertation:

1. **Efficiency of Tensor Voting**: Although tensor voting is a powerful tool to extract perceptual information from noisy data and has been effectively utilized in many applications of computer vision, its most important drawback so far has been its high computational cost. For this reason, it has only been used in applications where the processing time is not relevant, somehow limiting its scope of use. This problem has been tackled in Chapter 3 by proposing two different efficient implementations of tensor voting in 2D and 3D. Unlike previous works, this is the first attempt to propose efficient implementations in which not only the *stick* but also the *plate* and *ball* tensor voting are required. The results of that chapter have shown that it is possible to avoid this limitation by using the new proposed schemes. This is especially helpful in applications where efficiency is an issue. Hence, this thesis has contributed to extending the scope of use of tensor voting through the in-depth analysis of tensor voting developed in Chapter 3.

2. **Perceptual Meaning of the *Plate* and *Ball* Tensor Voting**: Depite the high amount of work on tensor voting, to the best of our knowledge, Chapter 3 constitutes the first effort to understand the perceptual meaning of the *plate* and *ball* tensor voting, which have always been defined in a constructive way. Therefore, many applications have only applied the *stick* tensor voting, since the perceptual meaning of the *plate* and *ball* tensor voting remained unclear somehow. The analysis presented in Chapter 3 will encourage the use of both the *plate* and *ball* tensor voting in order to improve the perceptual estimation of features.

3. **Framework for Utilizing the Classical Tensor Voting in Images**: So far, tensor voting has mainly been applied to clouds of points. Section 4.1 presents a new framework to extend the classical tensor voting to any kind of image, by taking into account its structural relationship with the well-known structure tensor, as well as some appropriate rescaling steps. Although the rotation term of the stick tensor voting generates halos in the second eigenvalue, it increases the robustness of the calculations in very noisy images, as shown in that section of the thesis.

4. **Classical Tensor Voting for Segmentation of 3D Images Acquired Through Stereo Vision**: Section 4.2 presents a new application of the classical tensor voting to image segmentation of images acquired through stereo vision. Previous range image segmentation algorithms in the literature have mainly dealt with images acquired through LADAR, structured light, or images where noise is not a big issue. The results of that section show that, to a certain extent, the classical tensor voting can also deal with the amount of noise of 3D images obtained from stereo rigs, especially for points located close to the camera.

5. **Unsuitableness of the Classical Tensor Voting for Image Denoising**: Chapter 5 shows that the classical tensor voting can only be applied to image denoising with the help of other methods. This mainly stems from the fact that the classical tensor voting has been devised to estimate structure. In other words, the classical tensor voting has a derivative nature, while image denoising algorithms usually require an integral approach.

6. **Tensor Voting as a General Methodology**: So far, tensor voting has only been applied as proposed by Guy and Medioni (1996, 1997). However, this thesis shows that, beyond a method, tensor voting can be thought of as a methodology in which information encoded

through tensors is propagated and aggregated in a local neighborhood. Thus, Chapters 5 and 6 show that tensor voting can yield state-of-the-art results by tailoring the way in which tensors are encoded, propagated and aggregated, while maintaining the tensor voting spirit.

7. **Encoding of Color Through Tensors**: Chapters 5 and 6 propose a new representation of color through tensors. The main advantage of the new representation over previous proposals (cf. Chapter 5) is that the resulting tensors can be added without any restriction. This point is especially important, since the power of tensors in computer vision mainly relies on the robust performance of the summation operation. Both the classical tensor voting and its extensions proposed in this thesis are based on this property. In addition, the proposed encoding always makes the tensors positive semidefinite.

8. **Adaptation of CIEDE2000 to Image Processing**: Chapters 5 and 6 show that the CIEDE2000 formula can yield more accurate estimations of perceptual color difference in noisy scenarios by adapting its parameters. Although previous works have proposed adaptations of these parameters, this thesis first applies this methodology to the field of image processing, in particular to edge preserving image denoising and color edge detection in noisy scenarios.

9. **Tensor Voting for Color Image Denoising**: Chapter 5 presents a new image denoising technique which has a better preformance than the state-of-the-art methods, not only for AWGN but also for real noise. The keypoints of this method are: the new proposal for encoding color through tensors, the specifically tailored voting process based on a set of perceptual criteria, and the use of an optimized version of CIEDE2000 for noisy scenarios.

10. **New Measures for Assessing Image Denoising Methods**: Chapter 5 proposes two new measures, $PSNRG$ and $PSNRA$, for comparing the performance of image denoising algorithms. Unlike previous measures, the new ones have been devised to measure the edge preservation feature and the rejection of undesirable artifacts.

11. **New Methodology for Assessing Edge Detectors**: This thesis proposes a new methodology to compare the performance of edge detection techniques. Unlike previous approaches, the new five proposed measurements get rid of possible bias introduced by post-processing steps by measuring performance before such steps (cf. Figure 6.1). Thus, the new proposed R, DS, P, FAR and $PSNR$ measures assess without the aforementioned

bias the features of recall, discriminability, precision, false alarm rejection and robustness respectively.

12. **Tensor Voting for Color Edge Detection**: Chapter 6 presents two new edge detectors based on tensor voting: one based on the classical tensor voting and the second closely related to the method presented in Chapter 5. On the one hand, the former has shown the best results under the precision and false alarm rejection measures when compared to the state-of-the-art schemes for image edge enhancement. On the other hand, the latter has shown very good results in robustness. Both proposals are also competitive under other measures, such as recall and discriminability, for both image segmentation and image edge enhancement.

13. **Relationships between Image Denoising and Edge Detection**: One of the most important contributions of this thesis has been to show that image denoising and edge detection are closely related and, thanks to that fact, they can be tackled in a unified framework. Interestingly enough, this unified framework (the TVD and TVED methods presented in Chapters 5 and 6) has yielded state-of-the-art results for both problems. This suggests that the human visual system likely undertakes image denoising and edge detection at the same time, in a similar way as researchers have found evidence that object recognition and image segmentation are performed at the same time by humans (cf. Chapter 7).

14. **Tensor Voting for Color Image Segmentation**: Chapter 7 describes a color image segmentation method that combines the techniques proposed in Chapters 5 and 6 with a modified version of an efficient graph-based image segmentation algorithm. This approach yields better image segmentation results than state-of-the-art methods with a competitive computational cost.

15. **New Measures for Assessing Image Segmentation Methods**: Chapter 7 proposes two new segmentation evaluation measures, $OSEG$ and $USEG$, which have been devised for respectively measuring the degree of over- and undersegmentation of a machine-computed segmentation respectively. Unlike previous measures for over- and undersegmentation, the new proposals do not require parameters and have an intuitive meaning (cf. Subsection 7.3.3).

16. **Integration of Tensor Voting with Other Techniques**: Section 4.2 and Chapter 7 show

two examples of how tensor voting can be combined with other techniques. In the first case, the results yielded by the classical tensor voting are encoded in a graph that is segmented a posteriori. In the second case, the edginess maps yielded by the method presented in Chapter 6 are used to steer the segmentation process.

In addition to the previous list, the following general statements can be derived from the results obtained throughout this dissertation.

1. **Efficient and Simplified Tensor Voting vs. Classical Tensor Voting**: The simplified tensor voting presented in Chapter 3 has been found to be not only significantly more efficient, but also more appropriate than the classical tensor voting for estimating saliencies in both noisy 3D scenarios and clouds of points with curved surfaces, by appropriately adjusting its parameters. In addition, the efficient tensor voting has also been found to be considerably more efficient than the classical tensor voting.

2. **Tensor Voting vs. Structure Tensor**: The structure tensor has been broadly utilized in many applications in both computer vision and image processing. This dissertation has shown that tensor voting is more effective for image structure estimation than the structure tensor, at a slightly higher computational cost. This opens the door to improved algorithms by replacing the structure tensor with tensor voting, as proposed in Section 4.1.

3. **Distribution of Noise in 3D Images Acquired Through Stereo Vision**: Experiments from Section 4.2 show that the amount of noise present in 3D images acquired through stereo vision for distant points makes any segmentation algorithm more prone to failures. Thus, image segmentation algorithms for this kind of image based on both 3D and 2D information can arguably have a better performance.

4. **Performance of Image Denoising Methods for AWGN and Real Noise**: Most of the previous work on image denoising has been devoted to filtering additive white Gaussian noise (AWGN). However, real camera noise has little to do with AWGN, as shown in Chapter 2. The results of Chapter 5 are in accordance with this fact, since the performance of filtering AWGN has been shown not always to be correlated to the one of real noise. Recent advances on modeling real noise have been essential to make objective comparisons among image denoising methods for this type of noise.

5. **Ability of Image Denoising Methods for Preserving Edges and Avoiding Artifacts**: The new $PSNRG$ measure proposed in Chapter 5 has been used to show that the performance of image denoising methods is better in flat regions than in edges. This difference in performance can be used to assess how good a method is for preserving edges. Even more interestingly, the new $PSNRA$ measure has been essential to show that, without exception, image denoising methods have more problems for avoiding artifacts than for preserving edges. Since the human visual system is especially prone to focusing on artifacts in images, future research in image denoising should take this fact into account.

6. **Choice of Edge Detector**: Usually, it has been considered that new proposals for edge detection should improve previous approaches for every application under any performance measure. On the contrary, this thesis shows that it is very difficult for an edge detector to have a perfect performance for all applications and measures. Even classical edge detectors, such as Sobel, have been found to be more appropriate for some applications than more sophisticated methods. This difficulty is mainly due to the fact that the performance of edge detectors constitutes a trade-off among several, almost contradictory features, such as completeness and robustness (cf. Chapter 6.3.1).

7. **Robustness of Image Segmentation Algorithms**: The results presented in Chapter 7 suggest that robust image segmentation can be attained by using texture features or by using robust techniques, such as tensor voting. However, the second approach has the advantage of being computationally more efficient than the first one.

8. **Tensor Voting as an Image Processing Tool**: The experimental results presented throughout this dissertation show that tensor voting is a powerful tool for image processing, especially for estimating structure, denoising color images, detecting edges and segmenting images. The robust performance of tensor voting makes it particularly appropriate for noisy scenarios, where it has been found to belong to the state-of-the-art.

8.2 Future Lines of Research

This thesis opens several lines of research. Once shown that tensor voting can be extended to a variety of applications by tailoring its encoding and voting processes, the most important future

research line consists of proposing new extensions of tensor voting for other applications and/or types of images. In addition, the following research lines can be derived from this work:

1. Propose extensions of the methods presented in Chapter 3 to N-D.
2. Propose efficient implementations of the classical *plate* and *ball* votes in the frequency domain.
3. Compare different ways to perform classical tensor voting on tensor-valued images.
4. Extend the method presented in Section 4.2 to integrate 2D and 3D information for segmenting images acquired through stereo vision.
5. Extend classical tensor voting to higher-order tensors of odd order.
6. Extend classical tensor voting to non-positive semidefinite tensors.
7. Extend the methods presented in Section 4.2 and Chapter 7 to multiscale image segmentation.
8. Extend the methods presented in Chapters 5, 6 and 7 to other types of images, such as tensor-valued images.
9. Find correlations between the human visual system and the new proposed $PSNRG$, $PSNRA$, R, DS, P, FAR, $OSEG$ and $USEG$ measures.
10. Propose evaluation measures for color image denoising, edge detection and image segmentation that do not require ground-truths.
11. Propose tensor encoding processes for other types of information, such as texture or features extracted from tensor-valued images.
12. Extend tensor voting to other applications, such as perceptual object recognition or multiple cue integration.

8.3 Publications

The following publications have been derived from this thesis:

1. A paper based on the efficient implementations of tensor voting descibed in Chapter 3 is currently under revision in the IEEE Transactions on Pattern Analysis and Machine Intelligence (Moreno et al., 2010c).

2. The method introduced in Section 4.1 was presented in the Workshop "New Developments in the Visualization and Processing of Tensor Fields" in July 2009 (Burgeth and Laidlaw, 2009).

3. A book chapter with an extended version of the previous article has already been accepted for publication (Moreno et al., 2010d).

4. The method presented in Section 4.2 was presented in the 6th Workshop on Graph-based Representations in Pattern Recognition in June 2007 (Moreno et al., 2007).

5. A variant of the graph creation step presented in Section 4.2 has been proposed in a paper that has recently been published in the Electronic Letters journal (Julià et al., 2010).

6. The TVD method presented in Chapter 5 was presented in the 13th International Conference in Analysis of Images and Patterns in September 2009 (Moreno et al., 2009a).

7. An extended version of the previous paper is currently under revision in the Computer Vision and Image Understanding journal (Moreno et al., 2010b).

8. The TVED edge detector described in Chapter 6 was presented in the International Conference on Image Processing in November 2009 (Moreno et al., 2009b).

9. The evaluation methodology of edge detectors introduced in Chapter 6 was presented in the International Conference on Image Processing in November 2009 (Moreno et al., 2009c).

10. The segmentation algorithm presented in Chapter 7 has been accepted for publication in the 20th International Conference on Pattern Recognition to be held in August 2010 (Moreno et al., 2010a).

APPENDICES

Appendix A

Non-Linear Least Squares Fitting for the *Plate* and *Ball* Tensor Voting

The functions s'_{1_p} and s'_{2_p}, required for computing *plate* votes in Section 3.1 can be approximated through non-linear least squares fitting on γ as:

$$s'_{1_p} \approx \begin{cases} 0 & \text{if } \gamma = 0 \\ \frac{c_{11}}{\gamma}\, e^{-\left(\frac{\ln(\gamma)-c_{12}}{c_{13}}\right)^2} & \text{if } 0 < \gamma \leq \pi/4 \\ c_{14} + c_{15}\, e^{-c_{16}\,(\gamma-\pi/4)} & \text{otherwise,} \end{cases} \tag{A.1}$$

$$s'_{2_p} \approx \begin{cases} e^{-\left(\frac{\gamma}{c_{21}}\right)^2} & \text{if } \gamma \leq \pi/4 \\ c_{22} + c_{23}\, e^{-\left(\frac{\gamma-\pi/4}{c_{24}}\right)} & \text{otherwise,} \end{cases} \tag{A.2}$$

where c_{ij} are factors that must be computed only once, since they only depend on b. In turn, these factors can also be approximated through non-linear least squares fitting on b by:

$$c_{11} \approx \frac{6.5360\ b - 3.7920}{b^2 + 4.8680\ b - 5.4220} - 0.3301\ e^{-0.1501\ b} + \frac{0.0600}{1 + e^{-0.0401\ (b-25.0)}} \tag{A.3}$$

$$c_{12} \approx 1.9501\ e^{-0.3010\ b} + 1.8001\ e^{-0.0101\ b} - 1.5750 \tag{A.4}$$

$$c_{13} \approx 1.2010 - 0.2511\ e^{-0.3001\ b} \tag{A.5}$$

$$c_{14} \approx 0.4901\ e^{-0.0801\ b} + 0.1301\ e^{-0.0010\ b} \tag{A.6}$$

$$c_{15} \approx 0.0785\ e^{-0.0221\ b} - 0.5701\ e^{-0.3501\ b} + 0.0466 \tag{A.7}$$

$$c_{16} \approx 0.1720\ e^{-0.2501\ (b-1.005)^2} + 0.0550 \tag{A.8}$$

$$c_{21} \approx 1.1197\ e^{-0.3114\ b} + 0.3552\ e^{-0.0125\ b} + 1.7894\ e^{-10.0\ b} + \frac{0.0450}{1 + e^{-(b-100.0)}} \tag{A.9}$$

$$c_{22} \approx 0.1977\ e^{-0.3101\ b} \tag{A.10}$$

$$c_{23} \approx 0.4278\ e^{-0.4051\ b} + 0.0836\ e^{-0.0881\ b} \tag{A.11}$$

$$c_{24} \approx -0.1294\ e^{-0.0878\ b} - 0.0243\ e^{0.0346\ b} \tag{A.12}$$

Following the same methodology, the functions s'_{2b} and s'_{3b}, required for computing *ball* votes, can be approximated by:

$$s'_{2b} \approx 0.2838\ e^{-0.0795b} + 0.2461\ e^{-0.0065b} \tag{A.13}$$

$$s'_{3b} \approx 0.3380\ e^{-0.3197b}. \tag{A.14}$$

In addition, s'_{3b} can be approximated by s'_{2b}/b for high values of b (e.g., $b > 30$).

Bibliography

Abraham, R., Marsden, J. E., Ratiu, T., 1988. Manifolds, tensor analysis, and applications, 2nd Edition. Springer.

Ali, Y. M. B., 2009. Edge-based segmentation using robust evolutionary algorithm applied to medical images. Journal of Signal Processing Systems 54 (1–3), 231–238.

Andrews, E. W., Gibson, L. J., 2001. The role of cellular structure in creep of two-dimensional cellular solids. Materials Science and Engineering A 303 (1-2), 120–126.

Arbeláez, P., Maire, M., Fowlkes, C., Malik, J., 2009. From contours to regions: An empirical evaluation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. pp. 2294–2301.

Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R., Wu, A. Y., 1998. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. Journal of the ACM 45 (6), 891–923.

Bagon, S., Boiman, O., Irani, M., 2008. What is a good image segment? a unified approach to segment extraction. In: Proc. European Conf. on Computer Vision, Lecture Notes in Computer Science 5305. pp. IV:30–44.

Baker, S., Nayar, S. K., 1999. Global measures of coherence for edge detector evaluation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. pp. II:373–379.

Barbu, A., 2009. Training an active random field for real-time image denoising. IEEE Transactions on Image Processing 18 (11), 2451–2462.

Basser, P. J., Pajevic, S., 2007. Spectral decomposition of a 4th-order covariance tensor: Applications to diffusion tensor MRI. Signal Processing 87 (2), 220–236.

Batard, T., Saint-Jean, C., Berthier, M., 2009. A metric approach to nD images edge detection with Clifford algebras. Journal of Mathematical Imaging and Vision 33 (3), 296–312.

Ben-Shahar, O., Zucker, S. W., 2004. Hue geometry and horizontal connections. Neural Networks 17 (5-6), 753–771.

Berns, R., 2001. The science of digitizing paintings for color-accurate image archives: A review. Journal of Imaging Science and Technology 45 (4), 305–325.

Berns, R. S., 2000. Billmeyer and Saltzman's Principles of Color Technology, 3rd Edition. John Wiley and Sons Inc.

Bigand, A., Evrard, L., Dubus, J., 2006. A new perceptual organization approach to 3D measuring system based on the fuzzy integral. Image and Vision Computing 24 (4), 381–393.

Bigün, J., Granlund, G., Wiklund, J., 1991. Multidimensional orientation estimation with applications to texture analysis and optical flow. IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (8), 775–790.

Blais, C., Arguin, M., Marleau, I., 2009. Orientation invariance in visual shape perception. Journal of Vision 9 (2), 1–23.

Borenstein, E., Ullman, S., 2008. Combined top-down/bottom-up segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (12), 2109–2125.

Bowyer, K., Kranenburg, C., Dougherty, S., 2001. Edge detector evaluation using empirical ROC curves. Computer Vision and Image Understanding 84 (1), 77–103.

Bresson, X., 2005. Image segmentation with variational active contours. Ph.D. thesis, École Polytechnique Fédéral de Lausanne.

Bridgman, P. W., 1922. Dimensional analysis. Yale University Press.

Brouwer, R. K., 2009. Extending the Rand, adjusted Rand and Jaccard indices to fuzzy partitions. Journal of Intelligent Information Systems 32 (3), 213–235.

Brox, T., Weickert, J., Burgeth, B., Mrázek, P., 2006. Nonlinear structure tensors. Image and Vision Computing 24 (1), 41–55.

Bruce, V., Green, P. R., Georgeson, M. A., 2003. Visual Perception: physiology, psychology and ecology, 4th Edition. Psychology Press.

Buades, A., Coll, B., Morel, J. M., 2005. A non-local algorithm for image denoising. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. pp. II:60–65.

Buades, A., Coll, B., Morel, J.-M., 2008. Nonlocal image and movie denoising. International Journal of Computer Vision 76 (2), 123–139.

Bühler, T., Hein, M., 2009. Spectral clustering based on the graph p-Laplacian. In: Proc. Int. Conf. on Machine Learning. pp. 81–88.

Burgeth, B., Didas, S., Weickert, J., 2009. A general structure tensor concept and coherence-enhancing diffusion filtering for matrix fields. In: Laidlaw, D., Weickert, J. (Eds.), Visualization and Processing of Tensor Fields: Advances and Perspectives. Springer, pp. 305–323.

Burgeth, B., Laidlaw, D. H., 2009. Abstracts collection. In: Burgeth, B., Laidlaw, D. H. (Eds.), New Developments in the Visualization and Processing of Tensor Fields. No. 09302 in Dagstuhl Seminar Proceedings. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, Dagstuhl, Germany.
URL http://drops.dagstuhl.de/opus/volltexte/2009/2238

Canny, J. F., 1986. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (6), 679–698.

Cardoso, J., Corte-Real, L., 2005. Toward a generic evaluation of image segmentation. IEEE Transactions on Image Processing 14 (11), 1773–1782.

Cardoso, J., Corte-Real, L., 2006. A measure for mutual refinements of image segmentations. IEEE Transactions on Image Processing 15 (8), 2358–2363.

Carlson, B., 2002. Comparison of modern CCD and CMOS image sensor technologies and systems for low resolution imaging. In: Proc. IEEE Conf. on Sensors. pp. I: 171–176.

Chapron, M., 2000. A color edge detector based on Dempster-Shafer theory. In: Proc. IEEE Int. Conf. on Image Processing. Vol. 2. pp. 812–815.

Chen, J., Pappas, T. N., Mojsilovic, A., Rogowitz, B. E., 2005. Adaptive perceptual color-texture image segmentation. IEEE Transactions on Image Processing 14 (10), 1524–1536.

Chen, M., Fridrich, J., Goljan, M., Lukas, J., 2008. Determining image origin and integrity using sensor noise. IEEE Transactions on Information Forensics and Security 3 (1), 74–90.

Chen, X., Golovinskiy, A., Funkhouser, T., 2009. A benchmark for 3D mesh segmentation. In: Proc. ACM SIGGRAPH. pp. 1–12.

Cheng, H. D., Jiang, X. H., Sun, Y., J., W., 2001. Color image segmentation: advances and prospects. Pattern Recognition 34 (12), 2259–2281.

Chou, C.-H., Liu, K.-C., 2007. A fidelity metric for assessing visual quality of color images. In: Proc. Int. Conf. Computer Communications and Networks. pp. 1154–1159.

Chung, F. R. K., 1997. Spectral graph theory. AMS Providence.

Comaniciu, D., Meer, P., 2002. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5), 603–619.

Cousty, J., Bertrand, G., Najman, L., Couprie, M., 2009. Watershed cuts: Minimum spanning forests and the drop of water principle. IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (8), 1362–1374.

Cowin, S. C., Doty, S. B., 2007. Tissue Mechanics. Springer.

Cremers, D., Rousson, M., Deriche, R., 2007. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. International Journal of Computer Vision 72 (2), 195–215.

Cumani, A., 1991. Edge detection in multispectral images. Graphical Models and Image Processing 53 (1), 40–51.

Curless, B., 1997. New methods for surface reconstruction from range images. Ph.D. thesis, Stanford University.

Das, M., Anand, J., 1995. Robust edge detection in noisy images using an adaptive stochastic gradient technique. In: Proc. IEEE Int. Conf. on Image Processing. Vol. 2. pp. 149–152 vol.2.

De Micheli, E., Caprile, B., Ottonello, P., Torre, V., 1989. Localization and noise in edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (10), 1106–1117.

Deng, Y., Manjunath, B. S., 2001. Unsupervised segmentation of color-texture regions in images and video. IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (8), 800–810.

Desolneux, A., Moisan, L., Morel, J.-M., 2008. From Gestalt Theory to Image Analysis: A Probabilistic Approach. Springer.

Dhillon, I., Guan, Y., Kulis, B., 2007. Weighted graph cuts without eigenvectors a multilevel approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (11), 1944–1957.

Di Zenzo, S., 1986. A note on the gradient of a multi-image. Computer Vision, Graphics, and Image Processing 33 (1), 116–125.

Donoho, D. L., Johnstone, I. M., 1994. Ideal spatial adaptation by wavelet shrinkage. Biometrika 81 (3), 425–455.

Elder, J., Krupnik, A., Johnston, L., 2003. Contour grouping with prior models. IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (6), 661–674.

Elder, J. H., Goldberg, R. M., 2002. Ecological statistics of Gestalt laws for the perceptual organization of contours. Journal of Vision 2 (4), 324–353.

Evans, A. E., Liu, X. U., 2006. A morphological gradient approach to color edge detection. IEEE Transactions on Image Processing 15 (6), 1454–1463.

Fairchild, M. D., 2005. Color Appearance Models, 2nd Edition. Wiley-IS&T.

Fan, J., Yau, D. K. Y., Elmagarmid, A. K., Aref, W. G., 2001. Automatic image segmentation by integrationg color-edge extraction and seeded region growing. IEEE Transactions on Image Processing 10 (10), 1454–1466.

Feldman, J., 2003. Perceptual grouping by selection of a logically minimal model. International Journal of Computer Vision 55 (1), 5–25.

Felzenswalb, P. F., Huttenlocher, D. P., 2004. Efficient graph-based image segmentation. International Journal of Computer Vision 59 (2), 167–181.

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 and Visual Computing 26 (4), 496–511.

Fischer, S., Bayerl, P., Neumann, H., Cristobal, G., Redondo, R., 2007. Iterated tensor voting and curvature improvement. Signal Processing 87 (11), 2503–2515.

Fleishman, S., Cohen-Or, D., Silva, C. T., 2005. Robust moving least-squares fitting with sharp features. In: Proc. ACM SIGGRAPH. pp. 544–552.

Foi, A., Trimeche, M., Katkovnik, V., Egiazarian, K., 2008. Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. IEEE Transactions on Image Processing 17 (10), 1737–1754.

Foresti, G. L., Regazzoni, C., 2000. A hierarchical approach to feature extraction and grouping. IEEE Transactions on Image Processing 9 (6), 1056–1074.

Förstner, W., 1986. A feature based correspondence algorithm for image matching. In: Int. Archives of Photogrammetry and Remote Sensing. Vol. 26. pp. 150–166.

Förstner, W., 1994. : A framework for low-level feature extraction. In: Proc. European Conf. on Computer Vision, Lecture Notes in Computer Science 801. pp. 383–394.

Fossum, E., 1997. CMOS image sensors: electronic camera-on-a-chip. IEEE Transactions on Electron Devices 44 (10), 1689–1698.

Foster, D. H., Gilson, S. J., 2002. Recognizing novel three-dimensional objects by summing signals from parts and views. Proceedings of the Royal Society B: Biological Sciences 269, 1939–1947.

Fowlkes, C. C., Martin, D. R., Malik, J., 2007. Local figure-ground cues are valid for natural images. Journal of Vision 7 (8), 1–9.

Franken, E., van Almsick, M., Rongen, P., Florack, L., ter Haar Romeny, B., 2006. An efficient method for tensor voting using steerable filters. In: Proc. European Conf. on Computer Vision. pp. 228–240.

Freixenet, J., Muñoz, X., Martí, J., Lladó, X., 2004. Colour texture segmentation by region-boundary cooperation. In: Proc. European Conf. on Computer Vision. pp. 250–261.

Freixenet, J., Muñoz, X., Raba, D., Martí, J., Cufí, X., 2002. Yet another survey on image segmentation: Region and boundary information integration. In: Proc. European Conf. on Computer Vision, Lecture Notes In Computer Science 2352. pp. III:408–422.

Ge, F., Wang, S., Liu, T., 2007. A new benchmark for image segmentation evaluation. Journal of Electronic Imaging 16 (3).

Geisler, W. S., 2008. Visual perception and the statistical properties of natural scenes. Annual Review of Psychology 59, 167–192.

Geisler, W. S., Perry, J. S., 2009. Contour statistics in natural images: Grouping across occlusions. Visual Neuroscience 26, 109–121.

Geman, S., Geman, D., 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence (6), 721–741.

Gevers, T., 2002. Image segmentation and similarity of color-texture objects. IEEE Transactions on Multimedia 4 (4), 509–516.

Ghosh, P., Bertelli, L., Sumengen, B., Manjunath, B., 2010. A non-conservative flow field for robust variational image segmentation. IEEE Transactions on Image Processing, in press.

Gillam, B., 2001. Varieties of grouping and its role in determining surface layout. In: Shipley, T. F., Kellman, P. J. (Eds.), From fragments to objects: segmentation and grouping in vision. North Holland, Ch. 8, pp. 247–352.

Golovinskiy, A., Funkhouser, T., 2009. Min-cut based segmentation of point clouds. In: IEEE Workshop on Search in 3D and Video at ICCV. pp. 1–8.

Gonzalez, R. C., Woods, R. E., 2007. Digital image processing, 3rd Edition. Prentice Hall, Inc.

Gow, R. D., Renshaw, D., Findlater, K., Grant, L., McLeod, S. J., Hart, J., Nicol, R. L., 2007. A comprehensive tool for modeling CMOS image-sensor-noise performance. IEEE Transactions on Electron Devices 54 (6), 1321–1329.

Grady, L., 2006. Random walks for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (11), 1768–1783.

Grossberg, M., Nayar, S., 2004. Modeling the space of camera response functions. IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (10), 1272–1282.

Grossberg, S., Kelly, F., 1999. Neural dynamics of binocular brightness perception. Vision Research 39 (22), 3796–3816.

Grossberg, S., Mingolla, E., Todorovic, D., 1989. A neural network architecture for preattentive vision. IEEE Transactions on Biomedical Engineering 36 (1), 65–84.

Guy, G., Medioni, G., 1996. Inferring global pereeptual contours from local features. International Journal of Computer Vision 20 (1–2), 113–133.

Guy, G., Medioni, G., 1997. Inference of surfaces, 3D curves and junctions from sparse, noisy 3D data. IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (11), 1265–1277.

Hagen, L., Kahng, A., 1992. New spectral methods for ratio cut partitioning and clustering. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 11 (9), 1074–1085.

Hahn, J., Lee, C.-O., 2009. A nonlinear structure tensor with the diffusivity matrix composed of the image gradient. Journal of Mathematical Imaging and Vision 34, 137–151.

Haralick, R. M., Shapiro, L., 1985. Survey: image segmentation techniques. Compuer Vision, Graphics and Image Processing 29 (1), 100–132.

Hartley, R., Zisserman, A., 2004. Multiple View Geometry in Computer Vision, 2nd Edition. Cambridge University Press.

Hayward, W. G., 2003. After the viewpoint debate: Where next in object recognition? Trends in Cognitive Sciences 7, 425–427.

He, X., Cai, D., Liu, H., Han, J., 2005. Image clustering with tensor representation. In: Proc. ACM Int. Conf. on Multimedia. pp. 132–140.

Healey, G., 1992. Segmenting images using normalized color. IEEE Transactions on Systems, Man, and Cybernetics 22 (1), 64–73.

Healey, G. E., Kondepudy, R., 1994. Radiometric CCD camera calibration and noise estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (3), 267–276.

Heath, M., Sarkar, S., Sanocki, T., Bowyer, K., 1998. Comparison of edge detectors: A methodology and initial study. Computer Vision and Image Understanding 69 (1), 38–54.

Heath, M., Sarkar, S., Sanocki, T., Bowyer, K. W., 1997. A robust visual method for assessing the relative performance of edge-detection algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (12), 1338–1359.

Henderson, J. M., Hollingworth, A., 1999. High-level scene perception. Annual Review of Psychology 50, 243–271.

Holzapfel, G. A., 2000. Nonlinear Solid Mechanics : A Continuum Approach for Engineering. Wiley.

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P., Bunke, H., Goldgof, D., Bowyer, K., Eggert, D., Fitzgibbon, A., Fisher, R., 1996. An experimental comparison of range image segmentation algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (7), 673–689.

Hou, Z., Koh, T., 2003. Robust edge detection. Pattern Recognition 36, 2083–2091.

Hubert, L., Arabie, P., 1985. Comparing partitions. Journal of Classification 2 (1), 193–218.

Huertas, R., Melgosa, M., Hita, E., 2006. Influence of random-dot textures on perception of suprathreshold color differences. Journal of the Optical Society of America A 23 (9), 2067–2076.

Hunt, R. W. G., 2004. The Reproduction of Colour, sixth Edition. Wiley-IS&T.

Hwang, C., Zhuang, S., Lai, S.-H., 2007a. Efficient intra mode selection using image structure tensor for H.264/AVC. In: Proc. IEEE Int. Conf. on Image Processing. pp. V:289–292.

Hwang, Y., Kim, J.-S., Kweon, I.-S., 2007b. Sensor noise modeling using the skellam distribution: Application to the color edge detection. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. pp. 1–8.

Ichir, M. M., Mohammad-Djafari, A., 2006. Hidden markov models for wavelet-based blind source separation. IEEE Transactions on Image Processing 15 (7), 1887–1899.

Jaccard, P., 1901. Étude comparative de la distribution florale dans une portion des Alpes et du Jura. Bulletin del la Société Vaudoise des Sciences Naturelles 37, 547–579.

Jain, A. K., Murty, M. N., Flynn, P. J., 1999. Data clustering: a review. ACM Computing Surveys 31 (3), 264–323.

Jia, J., Tang, C. K., 2004. Inference of segmented color and texture description by tensor voting. IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (6), 771–786.

Jia, J., Tang, C. K., 2005. Tensor voting for image correction by global and local intensity alignment. IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (1), 36–50.

Jiang, X., 2000. An adaptive contour closure algorithm and its experimental evaluation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (11), 1252–1265.

Johnson, G. M., Fairchild, M. D., 2003a. A top down description of S-CIELAB and CIEDE2000. Color Research and Application 28 (6), 425–435.

Johnson, G. M., Fairchild, M. D., 2003b. Visual psychophysics and color appearance. In: Sharma, G. (Ed.), Digital Color Imaging Handbook. CRC Press, Ch. 2, pp. 115–171.

Julesz, B., 1991. Early vision and focal attention. Reviews of Modern Physics 63 (3), 735–772.

Julià, C., Moreno, R., Puig, D., Garcia, M. A., 2010. Image segmentation through graph-based clustering from surface normals estimated by photometric stereo. Electronics Letters 46 (2), 134–135.

Jung, C. R., 2007. Unsupervised multiscale segmentation of color images. Pattern Recognition Letters 28 (4), 523–533.

Kang, E., Medioni, G., 2001. Color image segmentation based on tensor voting. In: Third Workshop on Perceptual Organization in Computer Vision.

Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: Active contour models. International Journal of Computer Vision 1 (4), 321–331.

Kenney, C., Zuliani, M., Manjunath, B., 2005. An axiomatic approach to corner detection. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. pp. I:191–197.

Kervrann, C., Boulanger, J., 2008. Local adaptivity to variable smoothness for exemplar-based image regularization and representation. International Journal of Computer Vision 79 (1), 45–69.

Keuchel, J., Schnorr, C., Schellewald, C., Cremers, D., 2003. Binary partitioning, perceptual grouping, and restoration with semidefinite programming. IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (11), 1364–1379.

Kim, H. S., Choi, H. K., Lee, K. H., 2009. Feature detection of triangular meshes based on tensor voting theory. Computer-Aided Design 41 (1), 47–58.

Kindlmann, G., Ennis, D. B., Whitaker, R., Westin, C.-F., 2007. Diffusion tensor analysis with invariant gradients and rotation tangents. IEEE Transactions on Medical Imaging 26 (11), 1483–1499.

Koffka, K., 1935. Principles of Gestalt Psychology. Harcourt, Brace & Jovanovich.

Kolluri, R., 2008. Provably good moving least squares. ACM Transactions on Algorithms 4 (2), 1–25.

Koren, R., Yitzhaky, Y., 2006. Automatic selection of edge detector parameters based on spatial and statistical measures. Computer Vision and Image Understanding 102 (2), 204–213.

Koschan, A., 1995. A comparative study on color edge detection. In: Proc. Asian Conf. on Computer Vision. pp. 574–578.

Koschan, A., Abidi, M., 2005. Detection and classification of edges in color images. IEEE Signal Processing Magazine 22 (1), 64–73.

Köthe, U., 2003. Edge and junction detection with an improved structure tensor. In: Proc. DAGM Symp. on Pattern Recognition, Lecture Notes in Computer Science 2781. pp. 25–32.

Kuang, J., Johnson, G. M., D., F. M., 2007. A refined image appearance model for HDR image rendering. Journal of Visual Communication and Image Representation 18, 406–114.

Kubovy, M., Pomerantz, J. R., 1981. Perceptual Organization. Lawrence Erlbaum Associates, Inc.

Kuehni, R., 2002. CIEDE2000, milestone or final answer? Color Research and Application 27 (2), 126–128.

Kuehni, R., 2008. Color difference formulas: An unsatisfactory state of affairs. Color Research and Application 33 (4), 324–326.

Lakare, S., 2000. 3D segmentation techniques for medical volumes. Tech. rep., State University of New York at Stony Brook.

Lebedev, L. P., Cloud, M. J., 2003. Tensor Analysis. World Scientific.

Levin, A., Weiss, Y., 2009. Learning to combine bottom-up and top-down segmentation. International Journal of Computer Vision 81 (1), 105–118.

Levin, D., 1998. The approximation power of moving least-squares. Mathematics of Computation 67 (224), 1517–1531.

Lezoray, O., Elmoataz, A., Bougleux, S., 2007. Graph regularization for color image processing. Computer Vision and Image Understanding 107 (1-2), 38–55.

Li, F., Nathan, A., 2005. CCD image sensors in deep ultraviolet degradation behavior and damage mechanisms. Springer, Ch. Overview of CCD, pp. 7–21.

Li, X., Gunturk, B., Zhang, L., 2008. Image demosaicing: A systematic survey. In: Proc. SPIE. Vol. 6822.

Lian, N.-X., Zagorodnov, V., Tan, Y.-P., 2006. Edge-preserving image denoising via optimal color space projection. IEEE Transactions on Image Processing 15 (9), 2575–2587.

Lim, D. H., 2006. Robust edge detection in noisy images. Computational Statistics & Data Analysis 50, 803–812.

Lim, J., Park, J., Medioni, G., 2007. Text segmentation in color images using tensor voting. Image and Vision Computing 25 (5), 671–685.

Lindeberg, T., 1994. Scale-Space Theory in Computer Vision. Kluwer Academic Publishers.

Lipman, Y., 2009. Stable moving least-squares. Journal of Approximation Theory 161, 371–384.

Liu, C., Szeliski, R., Kang, S. B., Zitnick, C., Freeman, W., 2008. Automatic estimation and removal of noise from a single image. IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2), 299–314.

Lombardi, G., Casiraghi, E., Campadelli, P., 2008. Curvature estimation and curve inference with tensor voting: A new approach. In: Proc. Advanced Concepts for Intelligent Vision Systems, Lecture Notes in Computer Science 5259. pp. 613–624.

Lorensen, W. E., Cline, H. E., 1987. Marching cubes: A high resolution 3D surface construction algorithm. In: Proc. ACM SIGGRAPH. pp. 163–169.

Loss, L., Bebis, G., Nicolescu, M., Skurikhin, A., 2008. Investigating how and when perceptual organization cues improve boundary detection in natural images. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Workshops. pp. 1–8.

Loss, L., Bebis, G., Nicolescu, M., Skurikhin, A., 2009. An iterative multi-scale tensor voting scheme for perceptual grouping of natural shapes in cluttered backgrounds. Computer Vision and Image Understanding 113 (1), 126–149.

Lowe, D. G., 1985. Perceptual Organization and Visual Recognition. Kluwer Academic Publishers.

Lu, D., Zhao, H., Jiang, M., Zhou, S., Zhou, T., 2005. A surface reconstruction method for highly noisy point clouds. In: Variational, Geometric and Level Set Methods in Computer Vision, Lecture Notes in Computer Science 3752. pp. 283–294.

Lucas, B. D., Kanade, T., 1981. An iterative image registration technique with an application to stereo vision. In: Proc. Imaging Understanding Workshop. pp. 121–130.

Lucassen, M. P., Bijl, P., Roelofsen, J., 2008. The perception of statical colored noise: detection and masking described by CIE94. Color Research and Application 33 (3), 178–191.

Lucchese, L., Mitra, S. K., 2001. Color image segmentation: A state-of-the-art survey. In: Proc. Indian National Science Academy. Vol. 67. pp. 207–221.

Lukac, R., Plataniotis, K. N., Venetsanopoulos, A. N., 2006. Color image denoising using evolutionary computation. International Journal of Imaging Systems and Technology 15 (5), 236–251.

Luo, M. R., Cui, G., Rigg, B., 2001. The development of the CIE 2000 colour-difference formula: CIEDE2000. Color Research and Application 26 (5), 340–350.

Ma, W. Y., Manjunath, B. S., 2000. Edgeflow: a technique for boundary detection and image segmentation. IEEE Transactions on Image Processing 9 (8), 1375–1388.

Mäenpää, T., Pietikäinen, M., 2004. Classification with color and texture: jointly or separately? Pattern Recognition 37 (8), 1629–1640.

Mahamud, S., Williams, L. R., Thornber, K. K., Xu, K., 2003. Segmentation of multiple salient closed contours from real images. IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (4), 433–444.

Maire, M., Arbeláez, P., Fowlkes, C., Malik, J., 2008. Using contours to detect and localize junctions in natural images. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. pp. 1–8.

Mangan, A., Whitaker, R., 1999. Partitioning 3d surface meshes using watershed segmentation. IEEE Transactions on Visualization and Computer Graphics 5 (4), 308–321.

Marr, D., 1982. Vision. W. H. Freeman and Co.

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: Proc. IEEE Int. Conf. Computer Vision. pp. II:416–423.

Martin, D. R., Fowlkes, C. C., Malik, J., 2004. Learning to detect natural image boundaries using local brightness, color and texture cues. IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (1), 530–549.

Massad, A., Babós, M., Mertsching, B., 2003. Application of the tensor voting technique for perceptual grouping to grey-level images: quantitative evaluation. In: Proc. Int. Symp. Image and Signal Processing and Analysis. pp. I: 504–509.

Medina-Carnicer, R., Madrid-Cuevas, F., Carmona-Poyato, A., Muñoz-Salinas, R., 2009. On candidates selection for hysteresis thresholds in edge detection. Pattern Recognition 42 (7), 1284–1296.

Medioni, G., Lee, M. S., Tang, C. K., 2000. A Computational Framework for Feature Extraction and Segmentation. Elsevier Science.

Meilă, M., 2007. Comparing clusterings—an information based distance. Journal of Multivariate Analysis 98 (5), 873–895.

Melkemi, K. E., Batouche, M., Foufou, S., 2006. A multiagent system approach for image segmentation using genetic algorithms and extremal optimization heuristics. Pattern Recognition Letters 27 (11), 1230–1238.

Miller, M., Kingsbury, N., 2008. Image denoising using derotated complex wavelet coefficients. IEEE Transactions on Image Processing 17 (9), 1500–1511.

Min, C., Medioni, G., 2006. Tensor voting accelerated by graphics processing units (GPU). In: Proc. Int. Conf. on Pattern Recognition. pp. III:1103–1106.

Min, C., Medioni, G., 2008. Inferring segmented dense motion layers using 5D tensor voting. IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (9), 1589–1602.

Mirmehdi, M., Petrou, M., 2000. Segmentation of color textures. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2), 142–159.

Moakher, M., 2008. Fourth-order cartesian tensors: old and new facts, notions and applications. The Quarterly Journal of Mechanics and Applied Mathematics 61 (2), 181–203.

Mordohai, P., Medioni, G., 2006a. Stereo using monocular cues within the tensor voting framework. IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (6), 968–982.

Mordohai, P., Medioni, G., 2006b. Tensor Voting: A perceptual Organization Approach to Computer Vision and Machine Learning. Morgan and Claypool Publishers.

Moreno, R., Garcia, M. A., Puig, D., 2007. Graph-based perceptual segmentation of stereo vision 3D images at multiple abstraction levels. In: Proc. Workshop on Graph-based Representations in Pattern Recognition, Lecture Notes in Computer Science 4538. pp. 148–157.

Moreno, R., Garcia, M. A., Puig, D., 2010a. Robust color image segmentation through tensor voting. In: Proc. Int. Conf. on Pattern Recognition. To appear.

Moreno, R., Garcia, M. A., Puig, D., Julià, C., 2009a. On adapting the tensor voting framework to robust color image denoising. In: Proc. Int. Conf. on Computer Analysis of Images and Patterns, Lecture Notes in Computer Science 5702. pp. 492–500.

Moreno, R., Garcia, M. A., Puig, D., Julià, C., 2009b. Robust color edge detection through tensor voting. In: Proc. IEEE Int. Conf. on Image Processing. pp. 2153–2156.

Moreno, R., Garcia, M. A., Puig, D., Julià, C., 2010b. Edge-preserving color image denoising through tensor voting. Compuer Vision and Image Understanding, under revision.

Moreno, R., Garcia, M. A., Puig, D., Pizarro, L., Burgeth, B., Weickert, J., 2010c. On improving the efficiency of tensor voting. IEEE Transactions on Pattern Analysis and Machine Intelligence, under revision.

Moreno, R., Pizarro, L., Burgeth, B., Weickert, J., Garcia, M. A., Puig, D., 2010d. New developments in the visualization and processing of tensor fields. In: Burgeth, B., Laidlaw, D. (Eds.), Adaptation of Tensor Voting to Image Structure Estimation. Springer, to appear.

Moreno, R., Puig, D., Julià, C., Garcia, M. A., 2009c. A new methodology for evaluation of edge detectors. In: Proc. IEEE Int. Conf. on Image Processing. pp. 2157–2160.

Moroney, N., Fairchild, M. D., Hunt, R. W., Li, C., Luo, M. R., Newman, T., 2002. The CIECAM02 color appearance model. In: Proc. IS&T/SID Color Imaging Conf. pp. 23–27.

Muñoz, X., Freixenet, J., Cufí, X., Martí, J., 2003. Strategies for image segmentation combining region and boundary information. Pattern Recognition Letters 24 (1-3), 375–392.

Mumford, D., Shah, J., 1989. Optimal approximations by piecewise smooth functions and associated variational problems. Communications on Pure and Applied Mathematics 42 (5), 577–685.

Muti, D., Bourennane, S., 2007. Survey on tensor signal algebraic filtering. Signal Processing 87 (2), 237–249.

Nafziger, J. S., Yen, S. C., Finkel, L. H., 1999. Psychophysical determination of the spatial connectivity function in a model of contour salience. Neurocomputing 26–27, 823–830.

Nagel, H.-H., Gehrke, A., 1998. Spatiotemporally adaptive estimation and segmentation of OF-fields. In: Proc. European Conf. on Computer Vision, Lecture Notes in Computer Science 1407. pp. 86–102.

Naik, S. K., Murthy, C. A., 2006. Standarization of edge magnitude in color images. IEEE Transactions on Image Processing 15 (9), 2588–2595.

Nath, S., Palaniappan, K., 2005. Adaptive robust structure tensors for orientation estimation and image segmentation. In: Proc. Int. Symp. Visual Computing, Lecture Notes in Computer Science 3804. pp. 445–453.

Neummann, H., Mingolla, E., 2001. Computational neural models of spatial integration in perceptual grouping. In: Shipley, T. F., Kellman, P. J. (Eds.), From Fragments to Objects: Grouping and Segmentation in Vision. Elsevier Science, Ch. 12, pp. 353–400.

Nguyen, T. B., Ziou, D., 2000. Contextual and non-contextual performance evaluation of edge detectors. Pattern Recognition Letters 21 (9), 805–816.

Nicolescu, M., Medioni, G., 2005. A voting-based computational framework for visual motion analysis and interpretation. IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (5), 739–752.

Noble, J., Boukerroui, D., 2006. Ultrasound image segmentation: A survey. IEEE Transactions on Medical Imaging 25 (8), 987–1010.

Novak, C. L., Shafer, S. A., 1987. Color edge detection. In: Proceedings of DARPA Image Understanding Workshop. Vol. 1. pp. 35–37.

Oleari, C., Melgosa, M., Huertas, R., 2009. Euclidean color-difference formula for small-medium color differences in log-compressed OSA-UCS space. Journal of the Optical Society of America A 26 (1), 121–134.

Ong, C. K., Matsuyama, T., 1998. Robust color segmentation using the dichromatic reflection model. In: Proc. Int. Conf. on Pattern Recognition. pp. I:780–784.

Ortiz, A., Oliver, G., 2006. On the use of the overlapping area matrix for image segmentation evaluation: A survey and new performance measures. Pattern Recognition Letters 27 (16), 1916–1926.

Osher, S., Sethian, J. A., 1988. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics 79 (1), 12–49.

Page, D. L., Sun, Y., Koschan, A. F., Paik, J., Abidi, M. A., 2002. Normal vector voting: Crease detection and curvature estimation on large, noisy meshes. Graphical Models 64 (3–4), 199–229.

Pajevic, S., Aldroubi, A., Basser, P. J., 2002. A continuous tensor field approximation of discrete DT-MRI data for extracting microstructural and architectural features of tissue. Journal of Magnetic Resonance 154, 85–100.

Pal, N. R., Pal, S. K., 1993. A review on image segmentation techniques. Pattern Recognition 26 (9), 1277–1294.

Pan, Y., Cheng, I., Basu, A., 2005. Quality metric for approximating subjective evaluation of 3D objects. IEEE Transactions on Multimedia 7 (2), 269–279.

Park, J., Kien, N. T., Kim, H., Lee, G., 2007a. Corrupted region restoration using second order tensors and segmentation. In: Proc. Int. Conf. on Fuzzy Systems and Knowledge Discovery. pp. IV:405–409.

Park, J., Kien, N. T., Lee, G., 2007b. Noise removal and restoration using voting-based analysis and image segmentation based on statistical models. In: Proc. Conf. on Energy Minimization Methods in Computer Vision and Pattern Recognition, Lecture Notes in Computer Science 4679. pp. 242–252.

Park, J., Kim, G., Dinh, T. N., Lee, G., 2008. Second order tensor voting in 3D and mean shift method for image segmentation. In: Proc. Int. Conf. on Advanced Language Processing and Web Information Technology. pp. 226–229.

Park, J., Yoo, J., Lee, G., 2006. A tensor voting for corrupted region inference and text image segmentation. In: Proc. Conf. on Advances in Multimedia Modeling, Lecture Notes in Computer Science 4351. pp. 751–761.

Parvin, B., Yang, Q., Han, J., Rydberg, B., Rydberg, B., Barcellos-Hoff, M., 2007. Iterative voting for inference of structural saliency and characterization of subcellular events. IEEE Transactions on Image Processing 16 (3), 615–623.

Peeters, T. H. J. M., Rodrigues, P. R., Vilanova, A., ter Haar Romeny, B. M., 2009. Analysis of distance / similarity measures for diffusion tensor imaging. In: Laidlaw, D., Weickert, J. (Eds.), Visualization and Processing of Tensor Fields: Advances and Perspectives. Springer, pp. 113–133.

Perona, P., Malik, J., 1990. Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (7), 629–639.

Peterson, M. A., Gibson, B. S., 1994. Must figured-ground organization precede object recognition? an assumption in peril. Psychological Science 5 (5), 253–259.

Pham, D. L., Xu, C., Prince, J. L., 2000. Current methods in medical image segmentation. Annual Review of Biomedial Engineering 2, 315–337.

Pichel, J. C., Singh, D. E., Rivera, F. F., 2006. Image segmentation based on merging of sub-optimal segmentations. Pattern Recognition Letters 27 (10), 1105–1116.

Point Grey Research Inc., USA. http://www.ptgrey.com.

Portilla, J., Strela, V., Wainwright, M., Simoncelli, E., 2003. Image denoising using scale mixtures of Gaussians in the wavelet domain. IEEE Transactions on Image Processing 12 (11), 1338–1351.

Pratt, W. K., 2007. Digital Image Processing: PIKS Scientific Inside, 4th Edition. Wiley-Interscience.

Prewer, D., Kitchen, L., 2001. Weighted linked pyramids and soft segmentation of colour images. Pattern Recognition Letters 22 (2), 123–132.

Prieto, M., Allen, A., 2003. A similarity metric for edge images. IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (10), 1265–1273.

Rabbania, T., van den Heuvelb, F. A., Vosselmanc, G., 2006. Segmentation of point clouds using smoothness constraint. In: ISPRS Commission V Symposium 'Image Engineering and Vision Metrology'. pp. 248–253.

Rand, W. M., 1971. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66 (336), 846–850.

Randall, J., Guan, L., Li, W., Zhang, X., 2008. The HCM for perceptual image segmentation. Neurocomputing 71 (10–12), 1966–1979.

Rao, A. R., Schunck, B. G., 1991. Computing oriented texture fields. CVGIP: Graphical Models and Image Processing 53, 157–185.

Rao, S., Mobahi, H., Yang, A., Sastry, S., Ma, Y., 2009. Natural image segmentation with adaptive texture and boundary encoding. In: Asian Conf. Comput. Vis.

Reisert, M., Burkhardt, H., 2008. Efficient tensor voting with 3D tensorial harmonics. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Workshops. pp. 1–7.

Reyes, L., Medioni, G., Bayro, E., 2007. Registration of 3D points using geometric algebra and tensor voting. International Journal of Computer Vision 75 (3), 351–369.

Risser, L., Plouraboue, F., Descombes, X., 2008. Gap filling of 3-D microvascular networks by tensor voting. IEEE Transactions on Medical Imaging 27 (5), 674–687.

Rittner, L., Flores, F. C., Lotufo, R. A., 2010. A tensorial framework for color images. Pattern Recognition Letters 31 (4), 277–296.

Roerdink, J. B. T. M., Meijster, A., 2000. The watershed transform: Definitions, algorithms and parallelization techniques. Fundamenta Informaticae 41, 187–228.

Rohr, K., 1994. Localization properties of direct corner detectors. Journal of Mathematical Imaging and Vision 4, 139–150.

Roth, S., Black, M. J., 2009. Fields of experts: A framework for learning image priors. International Journal of Computer Vision 82 (2), 205–229.

Rousson, M., Brox, T., Deriche, R., 2003. Active unsupervised texture segmentation on a diffusion based feature space. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. pp. II–699–704.

Rusinkiewicz, S., 2001. Real-time acquisition and rendering of large 3D models. Ph.D. thesis, Stanford University.

Ruzon, M., Tomasi, C., 2001. Edge, junction, and corner detection using color distributions. IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (11), 1281–1295.

Santos, J. M., Embrechts, M., 2009. On the use of the adjusted rand index as a metric for evaluating supervised classification. In: Proc. Int. Conf. on Artificial Neural Networks, Lecture Notes in Compute Science 5769. pp. II:175––184.

Sappa, A. D., 2006. Unsupervised contour closure algorithm for range image edge-based segmentation. IEEE Transactions on Image Processing 15 (2), 377–384.

Sarkar, S., Boyer, K. L., 1993. Perceptual organization in computer vision: A review and a proposal for a classificatory structure. IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2), 382–399.

Sarkar, S., Boyer, K. L., 1994. A computational structure for preattentive perceptual organization: graphical enumeration and voting methods. IEEE Transactions on Systems, Man, and Cybernetics 24 (2), 246–267.

Sarkar, S., Boyer, K. L., 1998. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. Computer Vision and Image Understanding 71 (1), 110–136.

Scharstein, D., Szeliski, R., 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Journal International Journal of Computer Vision 47 (1–3), 7–42.

Schultz, T., 2009. Feature extraction for visual analysis of DW-MRI data. Ph.D. thesis, Saarland University.

Schultz, T., 2010. New developments in the visualization and processing of tensor fields. In: Laidlaw, D., Burgeth, B. (Eds.), Towards resolving fiber crossings with higher order tensor inpainting. Springer, to appear.

Schultz, T., Weickert, J., Seidel, H.-P., 2009. Visualization and processing of tensor fields: Advances and perspectives. In: Laidlaw, D., Weickert, J. (Eds.), A Higher-order structure tensor. Springer, pp. 263–280.

Sezgin, M., Sankur, B., 2004. Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic Imaging 13 (1), 146–165.

Shah, S. K., 2008. Performance modeling and algorithm characterization for robust image segmentation. International Journal of Computer Vision 80 (1), 92–103.

Shamir, A., 2008. A survey on mesh segmentation techniques. Computer Graphics Forum 27 (6), 1539–1556.

Shao, X.-F., Ye, L.-W., Cai, M.-J., Wang, Y., 2008. Simultaneous image denoising and curve extraction by tensor voting. In: Proc. Int. Symp. on Computer Science and Computational Technology. pp. I: 536–538.

Sharma, G., Wu, W., Dalal, E. N., 2005. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. Color Research and Application 30 (1).

Sharon, E., Galun, M., Sharon, D., Basri, R., Brandt, A., 2006. Hierarchy and adaptivity in segmenting visual scenes. Nature 442, 810–813.

Shen, J., 2006. A stochastic-variational model for soft Mumford-Shah segmentation. International Journal of Biomedical Imaging 2006, 1–14.

Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8), 888–905.

Shin, M. C., Goldgof, D. B., Bowyer, K. W., Nikiforou, S., 2001. Comparison of edge detection algorithms using a structure from motion task. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 31 (4), 589–601.

Siagian, C., Itti, L., 2007. Rapid biologically-inspired scene classification using features shared with visual attention. IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2), 300–312.

Skarbek, W., Koschan, A., 1994. Colour image segmentation: A survey. Tech. rep., Institute for Technical Informatics, Technical University of Berlin.

Soundararajan, P., Sarkar, S., 2003. An in-depth study of graph partitioning measures for perceptual organization. IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (6), 642–660.

Spreeuwers, L. J., van der Heijden, F., 1992. Evaluation of edge detectors using average risk. In: Proc. Int. Conf. on Pattern Recognition. Vol. 3. pp. 771–774.

Stahl, J. S., Wang, S., 2008. Globally optimal grouping for symmetric closed boundaries by combining boundary and region information. IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (3), 395–411.

Tai, Y.-W., Jia, J., Tang, C.-K., 2007. Soft color segmentation and its applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (9), 1520–1537.

Tai, Y.-W., Tong, W.-S., Tang, C.-K., 2006a. Perceptually-inspired and edge-directed color image super-resolution. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. pp. II:1948–1955.

Tai, Y.-W., Tong, W.-S., Tang, C.-K., 2006b. Simultaneous image denoising and compression by multiscale 2D tensor voting. In: Proc. Int. Conf. on Pattern Recognition. pp. III:818–821.

Tang, C.-K., Medioni, G., Lee, M. S., 2001. N-Dimensional tensor voting and application to epipolar geometry estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (8), 829–844.

Tao, D., Maybank, S., Hu, W., Li, X., 2005. Stable third-order tensor representation for colour image classification. In: Proc. IEEE/WIC/ACM Int. Conf. on Web Intelligence. pp. 641–644.

Tao, L., Murino, V., Medioni, G., 2002. A tensor voting approach for the hierarchical segmentation of 3-D acoustic images. In: Proc. Int. Symp. on 3D Data Processing Visualization and Transmission. pp. 126–135.

Tao, W., Jin, H., Zhang, Y., 2007. Color image segmentation based on mean shift and normalized cuts. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 37 (5), 1382–1389.

Tarr, M. J., Bülthoff, H. H., 1998. Image-based object recognition in man, monkey and machine. Cognition 67 (1–2), 1–20.

Thakoor, N., Gao, J., Devarajan, V., 2008a. Multihypothesis prior for segmentation of stereo disparity. IEEE Signal Processing Letters 15, 613–616.

Thakoor, N., Gao, J., Devarajan, V., 2008b. Multistage branch-and-bound merging for planar surface segmentation in disparity space. IEEE Transactions on Image Processing 17 (11), 2217–2226.

Tomasi, C., Manduchi, R., 1998. Bilateral filtering for gray and color images. In: Proc. IEEE Int. Conf. on Computer Vision. pp. 839–846.

Tonazzini, A., Bedini, L., Salerno, E., 2006. A markov model for blind image separation by a mean-field EM algorithm. IEEE Transactions on Image Processing 15 (2), 473–482.

Tong, W.-S., Tang, C.-K., 2005. Robust estimation of adaptive tensors of curvature by tensor voting. IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (3), 434–449.

Tong, W.-S., Tang, C.-K., Medioni, G., 2004. Simultaneous two-view epipolar geometry estimation and motion segmentation by 4D tensor voting. IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (9), 1167–1184.

Trahanias, P. E., Venetsanopoulos, A. N., 1996. Vector order-statistics operators as color edge detectors. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 26 (1), 135–143.

Triantafyllidis, G., Tzovaras, D., Strintzis, M., 2000. A Bayesian segmentation of stereo pairs. In: Proc. IEEE Int. Conf. on Multimedia and Expo. pp. III:1353–1356.

Tschumperlé, D., 2006. Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE's. International Journal of Computer Vision 68 (1), 65–82.

Tsin, Y., Ramesh, V., Kanade, T., 2001. Statistical calibration of CCD imaging process. In: Proc. IEEE Int. Conf. on Computer Vision. pp. I:480–487.

Unnikrishnan, R., Hebert, M., 2005. Measures of similarity. In: IEEE Workshop on Applications of Computer Vision. pp. 394–400.

Unnikrishnan, R., Pantofaru, C., Hebert, M., 2007. Toward objective evaluation of image segmentation algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (6), 929–944.

van de Weijer, J., van den Boomgaard, R., 2005. Least squares and robust estimation of local image structure. International Journal of Computer Vision 64 (2/3), 143–155.

Vasseur, P., Pegard, C., Mouaddib, E. M., Delahoche, L., 1999. Perceptual organization approach based on dempster-shafer theory. Pattern Recognition 32 (8), 1449–1462.

Vergés-Llahí, J., 2005. Color constancy and image segmentation techniques for applications to mobile robotics. Ph.D. thesis, Polytechnic University of Catalonia.

Wang, S., Kubota, T., Siskind, J. M., Wang, J., 2005. Salient closed boundary extraction with ratio contour. IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (4), 546–561.

Wang, X.-Y., Bu, J., 2010. A fast and robust image segmentation using fcm with spatial information. Digital Signal Processing, in press.

Weeks, A. R., Felix, C. E., Myler, H., 1995. Edge detection of color images using the HSL color space. In: Proc. SPIE. Vol. 2424. pp. 291–301.

Weickert, J., 1999a. Coherence-enhancing diffusion filtering. International Journal of Computer Vision 31 (2-3), 111–127.

Weickert, J., 1999b. Coherence-enhancing diffusion of colour images. Image and Visual Computing 17, 199–212.

Weickert, J., 2001. Efficient image segmentation using partial differential equations and morphology. Pattern Recognition 34 (9), 1813–1824.

Weickert, J., Brox, T., 2002. Inverse problems, image analysis, and medical imaging. In: Nashed, M. Z., Scherzer, O. (Eds.), Diffusion and regularization of vector- and matrix-valued images. AMS, Providence, pp. 251–268.

Welk, M., Weickert, J., F., B., Schnörr, C., Feddern, C., Burgeth, B., 2007. Median and related local filters for tensor-valued images. Signal Processing 87 (2), 291–308.

Wen, J., Gao, X., Yuan, Y., Tao, D., Li, J., 2010. Incremental tensor biased discriminant analysis: A new color-based visual tracking method. Neurocomputing, to appear.

Williams, L. R., Thornber, K. K., 1999. A comparison of measures for detecting natural shapes in cluttered backgrounds. International Journal of Computer Vision 34 (2–3), 81–96.

Wyszecki, G., Stiles, W., 2000. Color Science: Concepts and Methods, Quantitative Data and Formulae, 2nd Edition. John Wiley and Sons.

Xin, J. H., Shen, H., Lam, C. C., 2005. Investigation of texture effect on visual colour difference evaluation. Color Research and Application 30 (5), 341–347.

Xue-Wei, L., Xin-Rong, Z., 2008. A perceptual color edge detection algorithm. In: Proc. International Conference on Computer Science and Software Engineering. Vol. 1. pp. 297–300.

Yang, A. Y., Wright, J., Ma, Y., Sastry, S. S., 2008. Unsupervised segmentation of natural images via lossy data compression. Computer Vision and Image Understanding 110 (2), 212–225.

Yen, S. C., Finkel, L. H., 1998. Extraction of perceptually salient contours by striate cortical networks. Vision Research 38 (5), 719–741.

Yitzhaky, Y., Peli, E., 2003. A method for objective edge detection evaluation and detector parameter selection. IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (8), 1027–1033.

Yu, H., Zhao, L., Wang, H., 2009. Image denoising using trivariate shrinkage filter in the wavelet domain and joint bilateral filter in the spatial domain. IEEE Transactions on Image Processing 18 (10), 2364–2369.

Yu, S. X., 2003. Computational models of perceptual organization. Ph.D. thesis, Carnegie Mellon University.

Yüksel, M. E., 2007. Edge detection in noisy images by neuro-fuzzy processing. AEU - International Journal of Electronics and Communications 61 (2), 82–89.

Zhang, E., Yeh, H., Lin, Z., Laramee, R., 2009. Asymmetric tensor analysis for flow visualization. IEEE Transactions on Visualization and Computer Graphics 15 (1), 106–122.

Zhang, H., Fritts, J. E., Goldman, S. A., 2008. Image segmentation evaluation: A survey of unsupervised methods. Computer Vision and Image Understanding 110 (2), 260–280.

Zhang, X., Wandell, B., 1997. A spatial extension of CIELAB for digital color image reproduction. Society for Information Display Symposium Technical Digest 27, 731–734.

Zhang, Y., Rockett, P. I., 2006. The Bayesian operating point of the Canny detector. IEEE Transactions on Image Processing 15 (11), 3409–3416.

Zhang, Y. J., 1996. A survey on evaluation methods for image segmentation. Pattern Recognition 29, 1335–1346.

Zhu, S. C., Mumford, D., 1997. Prior learning and Gibbs reaction-diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (11), 1236–1250.

Zhu, S. Y., Plataniotis, K. N., Venetsanopoulos, A. N., 1999. Comprehensive analysis of edge detection in color image processing. Optical Engineering 38 (4), 612–625.