AN ADAPTIVE SPECTRALLY WEIGHTED STRUCTURE TENSOR APPLIED TO TENSOR ANISOTROPIC NONLINEAR DIFFUSION FOR HYPERSPECTRAL IMAGES

By

Maider J. Marin Quintero

A dissertation thesis submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY
in
COMPUTING AND INFORMATION SCIENCE AND ENGINEERING

University of Puerto Rico
MAYAGUEZ CAMPUS
June 2012

Approved by:

_____	_____
Shawn Hunt, PhD. Member, Graduate Committee	Date
_____	_____
Vidya Manian, PhD. Member, Graduate Committee	Date
_____	_____
Wilson Rivera, PhD. Member, Graduate Committee	Date
_____	_____
Miguel Velez-Reyes, PhD. President, Graduate Committee	Date
_____	_____
Nestor Rodriguez, PhD. Program Coordinator	Date
_____	_____
Francisco Maldonado, PhD. Graduate School Representative	Date

To my two Daniel with all my love.....

ACKNOWLEDGMENTS

It is with immense gratitude that I acknowledge the support and help of Dr. Miguel Veléz-Reyes. This dissertation would not have been possible without his dedication and guidance through all my doctorate years working with him. I would like to thank him for all the opportunities he put in front of me, for his friendship and help in other aspects of my life. I also thank to Isnardo Arenas and professor Paul Castillo for useful comments during the implementation process of this thesis. To Julio Duarte-Carvajalino from the University of Minnesota for his comments in the papers. Special thanks to my husband Daniel McGee for all his love and support. For listening when I was confused and for proofreading this thesis and all papers I had submitted during the course of this work.

Furthermore, I would like to thank to the Laboratory for Applied Remote Sensing and Image Processing (LARSIP) staff in particular Maribel and Evelyn for all their support, friendliness and for making my life easier. To all my friends that at some point life put us together in LARSIP. In particular, I want to express my gratitude to Santiago Velasco for his friendship and advice that guided me towards Image Processing. I am also grateful to Maria C. Torres, Leidy Dorado, Samuel Rosario, Victor Asencio, Fanny Nina, Andrea Santos, Miguel Goenaga and their families for their friendship and support during all this years.

Thanks to the Mathematical Science Department for providing the funds for my first two years of my doctorate and the Electrical and Computer Engineering Department for providing the academic environment that has allowed me to grow as both a professional and person. To the staff and director of the Computing and In-

formation Sciences and Engineering Doctorate program for their friendliness and understanding.

I wish to thank my parents Faride y Rafael for all their love and for the many times they left everything in Colombia to come and support us during high stress times. To my twin sister Deiry for the spiritual and moral support that you provide me during all this time in the distance and my siblings Gelmy , Rafael and Mileidy for always being there for me.

This work was supported by U.S. Department of Homeland Security under Award Number 2008-ST-061-ML0002 and partial support also was provided under DHS Award Number 2008-ST-061-ED0001. This work used facilities of the Bernard M. Gordon Center of Subsurface Sensing and Imaging sponsored by the NSF ERC program under award EEC-9986821. My last year was supported by NASA under Award Number NNX10AM80H. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of the U.S. Department of Homeland Security or the National Science Foundation or NASA.

Abstract of Dissertation Presented to the Graduate School of the
University of Puerto Rico in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy.

AN ADAPTIVE SPECTRALLY WEIGHTED STRUCTURE TENSOR APPLIED TO TENSOR ANISOTROPIC NONLINEAR DIFFUSION FOR HYPERSPECTRAL IMAGES

By
Maider Judith Marin Quintero

2012

Chair: Miguel Veléz-Reyes,
Ph.D Major Program: Computing and Information Science and Engineering.

ABSTRACT

The structure tensor for vector valued images is most often defined as the average of the scalar structure tensors in each band. The problem with this definition is the assumption that all bands provide the same amount of edge information giving them the same weights. As a result non-edge pixels can be reinforced and edges can be weakened resulting in a poor performance by processes that depend on the structure tensor. Iterative processes, in particular, are vulnerable to this phenomenon. In this work, a structure tensor for Hyperspectral Images (HSI) is proposed. The initial matrix field is calculated using a weighted smoothed gradient. The weights are based on the Heat Operator. This definition is motivated by the fact that in HSI, neighboring spectral bands are highly correlated, as are the bands of its gradient. To use the heat operator, the smoothed gradient is modeled as the initial heat distribution on a compact manifold **M**. A Tensor Anisotropic Nonlinear Diffusion (TAND) method using the spectrally weighted structure tensor is proposed to do two kind of processing: Image regularization known as Edge Enhancing Diffusion (EED) and structure enhancement known as Coherence Enhancing Diffusion (CED). Diffusion tensor and a stopping criteria were also developed in this work. Comparisons between methods show that the structure tensor with weights based on the

heat operator better discriminates edges that need to be persistent during the iterative process with EED and produces more complete edges with CED. Remotely sensed and biological HSI are used in the experiments.

Resumen de la Disertación Presentada a la Escuela Graduada de la
Universidad de Puerto Rico en cumplimiento parcial of de los requisitos
para el grado de Doctor en Filosofía.

UN TENSOR DE ESTRUCTURA ADAPTIVO ESPECTRALMENTE Y SU USO EN DIFUSION ANISOTROPICA NO LINEAR USANDO UN TENSOR PARA IMAGENES HIPERESPECTRALES

Por

Maider Judith Marin Quintero

2012

Concejero: Miguel Veléz-Reyes,
Programa doctoral: Ciencias e Ingeniería de la Información y la Computación

RESUMEN

El tensor de estructura para imágenes vectoriales es comúnmente definido como el promedio de los tensores de estructura que ha sido previamente calculado para cada banda de la imagen. El problema con esta definición es que ella asume que todas las bandas proveen la misma cantidad de información de los bordes. Por lo tanto le da el mismo peso a cada una de las bandas. Como resultado, pixeles que no son bordes son reforzados y los bordes pueden ser debilitados. Esto hace que otros procesos que dependan del tensor de estructura den resultados mediocres. Los procesos iterativos son los más vulnerables a este fenómeno. En este trabajo se propone un tensor de estructura para imágenes HiperEspectrales (IHE). El campo matricial inicial es calculado usando un gradiente suavizado ponderado. Los pesos son basados en el operador de calor. Esta definición es motivada por una propiedad de las IHE y de su gradiente. Esta es: las bandas espectrales que están cercanas son altamente correlacionadas. Para poder hacer uso del operador del calor, el gradiente suavizado es modelado como la distribucion inicial de calor en una variedad compacta denotada por $\mathbf{M}$. Este tensor de estructura será aplicado a la Difusión Anisotrópica No Lineal basada en Tensores (DANT)

para hacer Difusión que Preserva Bordes (DPB) y Difusión que Realza Coherencia (DRC). Comparación entre los métodos muestran que el tensor de estructura ponderado con pesos basados en el operador de calor discrimina mejor los bordes con DPB y produce bordes mas completos con DRC. Estos métodos han sido aplicados a IHE sensadas remotamente como tambien imágenes biológicas adquiridas con microscopios hiperespectrales.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

1

BACKGROUND

1.1 INTRODUCTION

During the last three decades, hyperspectral remote sensing has been studied and developed as a powerful and versatile field. Hyperspectral remote sensors, collect image data simultaneously in hundreds of narrow, adjacent spectral bands. Hyperspectral images (HSI) are used to monitor different types of ecosystems, detect and identify objects such as minerals, terrestrial vegetation, man-made materials and backgrounds. The main advantage of this technology is that it provides a continuous and complete record of spectral responses of materials over a wavelength-interval of the electromagnetic spectrum. However, this same advantage leads to complexity in terms of processing and analysis. In addition, the capability of detection and identification of objects is reduced by physical and/or chemical variability of the material spectra, and noise and degradation produced by the sensing system. Hyperspectral images (HSI) contain a wealth of information. The high dimension of this kind of data makes it difficult to apply methods used in pattern recognition or computer vision without model adaptation or extension. Even methods used in the study of Multispectral images (MSI) may not produce good results when applied to HSI directly.

Spectral methods used in vector-valued images such as MSI can be extended to HSI. Usually the extension is done by seeing HSI as a set of spectral signatures of the material in the scene. Then vector methods to process spectra are used. For

example, Spectral Angle Mapper (SAM) and statistical methods in which each pixel (a vector) is seen as a random variable. Usually those methods do not care about the spatial position of the pixels and do not have into account its neighbors. As consequence of ignoring the spatial information those methods perform poorly [1]. One can think in another way of processing HSI by seeing them as a set of gray value images (or bands), each of them showing different features of the same scene. Then, extend computer vision methods used in gray images by processing them in a band by band mode. There are several problems with this kind of processing: (*i*) it ignores the physics involved in capturing the spectrum of each pixel and its local correlation. (*ii*) Given an object in the scene, some bands will show the whole object, some others will show part of its features and in other bands, the object will not appear at all. So, processing the bands independently will process the object differently in each band producing unwanted discontinuities and artifacts [2, 3]. To obtain good results processing HSI it is necessary to include the spectral and spatial information on the processing [4, 5].

The most stable and reliable descriptor of the spatial local structure of an image is the Structure Tensor (ST) [6]. The structure tensor is based on the outer product of the spatial gradient. This tensor is a symmetric positive semi-definite matrix, that at each pixel determine the orientation of minimum and maximum fluctuation of gray values in a neighborhood of the pixel. In the case of two dimensional (2-D) images, the eigen-decomposition of the structure tensor can be written as the sum of two expression that describe two basic local structures: linear and isotropic structures. This is possible since it provides the main directions of the gradient in a specified spatial neighborhood of a point. In addition, information on how strongly the gradient is biased towards a particular direction, known as *coherence*, can be extracted. Therefore, it can be used for both orientation estimation and analysis of image structure [7]. It has proven its usefulness in many applications for gray

value and color images such as corner detection [8], texture analysis [9, 10, 2, 11], diffusion filtering [2, 12] and optic flow estimation [10, 13, 14]. It is also used to define edge detectors [15, 16, 17], and to find the local structure [18, 19, 20, 21] and the structure inside patches [22] in several processes used to spatially regularize such images.

The structure tensor is used in a divergence-driven Partial Differential Equation (PDE)-based anisotropic diffusion method using a tensor, known as Tensor Anisotropic Nonlinear Diffusion (TAND). The eigen-decomposition of the structure tensor for 2D-images produce linear and isotropic structures that can be used to design filters for image regularization and structure enhancement in images. *Image regularization processes* consist in smoothing the image preserving its edges. *Structure Enhancement processes* consist in the enhancement of only some features in the images leaving the rest of the image almost intact. In the case of 2D images, flow-like structures can be distinguished from the linear structures found using the structure tensor. They are present in clouds, wakes, plumes, grass fields, fluids inside cells and so on. These structures can be found by looking to the orientation of lowest fluctuation of gray value in an image, which is known as *coherence orientation*; and it is determined by the eigenvector of the structure tensor with the smallest eigenvalue. The structure enhancement process in the coherence orientation is known as *coherence enhancement*; informally speaking, it is also known as enhancing flow-like structures or completing interrupted lines. To the best of our knowledge, this process have never been used to enhance HSI. Coherence enhancement have been studied in the context of gray and color images [23] by pattern recognition and computer vision fields and used for automatic grading of fabrics or wood surfaces [20], segmenting two-photon laser scanning microscopy images [24], enhancing gray level fingerprint images [6], enhancing corners [25] and three

dimensional (3-D) medical imaging [20, 26]. But little has been done with vector-valued data such as MSI or HSI.

On the other hand, image regularization using the structure tensor in a divergence-driven PDE setting have not been applied to HSI. The closest method [5] which consists in smoothing along the edges and inhibit the diffusing across it using a scalar diffusivity function works very well for high contrast edges otherwise fail (see Chapter IV). Since this method does not find the directions of change in the gradient it only can be used it for image regularization and not for structure enhancement. TAND is a method designed to process color images. In this work it will be presented an extension for HSI.

1.2 HYPERSPECTRAL REMOTE SENSING

Remote sensing is the field of study associated with extracting information about an object without coming into physical contact with it [27]. This is a broad definition that includes vision, astronomy, space probes, most medical devices, sonar and many other areas. For the interest of this work, this definition is restricted to the objects observed on the Earth's surface. The data consist in sensing and recording reflected or emitted electromagnetic (EM) radiation from the objects and it is acquired by aircraft and satellite [28, 27]. There are two main types of remote sensing: passive remote sensing and active remote sensing. Passive remote sensing uses sensors that collect energy that is either emitted directly by the objects such as thermal self emission or reflected from natural sources, such as the sun. While active ones emits energy in order to scan objects and areas whereupon a sensor then detects and measures the radiation that is reflected or back scattered from the target [27].

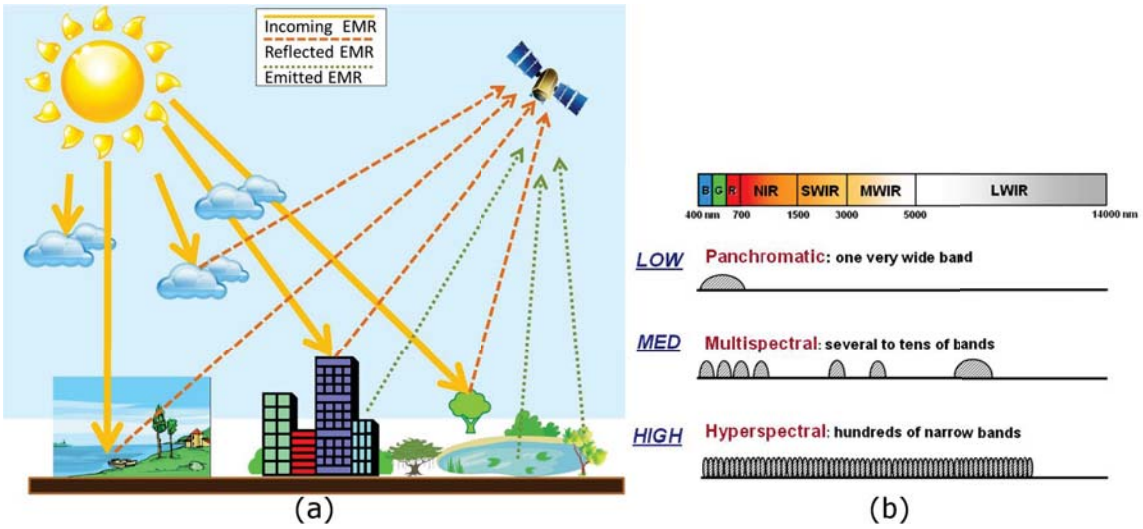

Figure 1.2.1: Remote sensing acquisition and spectral sampling. (a) Remote Sensing and EMR. (b) Types of spectral sampling in spectral imaging, taken from (Resmini [30])

Hyperspectral Remote Sensing belong to the passive aerospace remote sensing of the earth with emphasis on the $0.4 - 12$ μm interval in the EM spectrum. Hyperspectral Remote sensing is based in the fact that materials reflect, absorb, and emit ElectroMagnetic Radiation (EMR) at specific wavelengths, see Figure 1.2.1(a). A hyperspectral image (HSI) is one in which the spectral signature from each pixel is measured at many narrow, contiguous wavelength intervals. Such an image provides for every pixel high resolution spectral signatures, which gives information about the energy-matter interaction ([29]). Multispectral sensors, on the other hand, acquire images simultaneously but at separate non-contiguous and broad wavelength intervals or bands in the electromagnetic spectrum. They typically record tens of bands. The most simple sensor is the one that produces a panchromatic image, this is one very broad band in the visual wavelength range rendered in black and white, see Figure 1.2.1(b).

The quality of remote sensing data consists of its spatial, spectral, radiometric and temporal resolutions. *Spatial resolution* refers to the area that the size of a pixel

may correspond to. Usually square areas ranging in side length from 1 to 1,000 meters (3.3 to 3,300 ft). *Spectral resolution* refers to the wavelength width of the different frequency bands recorded and the number of frequency bands recorded by the platform. For example, the Hyperion sensor on Earth Observing-1 has higher spectral resolution than Landsat TM. Hyperion resolves 220 bands from 0.4 to 2.5 μm, with a spectral resolution of ∽ 10 to 11 *nm* per band, while Landsat TM resolves seven bands, including several in the infra-red spectrum, ranging from a spectral resolution of 0.07 to 2.1 μm. *Radiometric resolution* refers to the number of different intensities of radiation the sensor is able to distinguish. Typically, this ranges from 8 to 14 bits, corresponding to 256 levels of the gray scale and up to 16,384 intensities or "shades" of color, in each band. It also depends on the instrument noise. *Temporal resolution* refers to the frequency of flyovers by the satellite or plane. In general, this is relevant in time-series studies or studies that requires an averaged or mosaic image for monitoring conditions on the ground [28].

Figure 1.2.2 illustrates how spatial and spectral information is represented by a cube whose base is the spatial coordinates row and column, and the depth is spectral information (bands or channels).With the spectral signatures provided in HSIs, it is possible to discriminate between materials or identify different objects based on spectroscopic techniques. Few materials can be distinguished using spectral features with multispectral imagery and none with panchromatic imagery. The discriminatory capability of HSIs, advances in analysis and development of fast methods for processing make this technology suitable to be used in fields such as environmental monitoring [31], precision farming [32], insurance and car navigation at global and local scales. More recently, Hyperspectral imaging technology has found applications beyond earth remote sensing in agriculture[33], medical diagnosis[34, 35], biology [36], pharmaceutical industry [37], forensics medicine

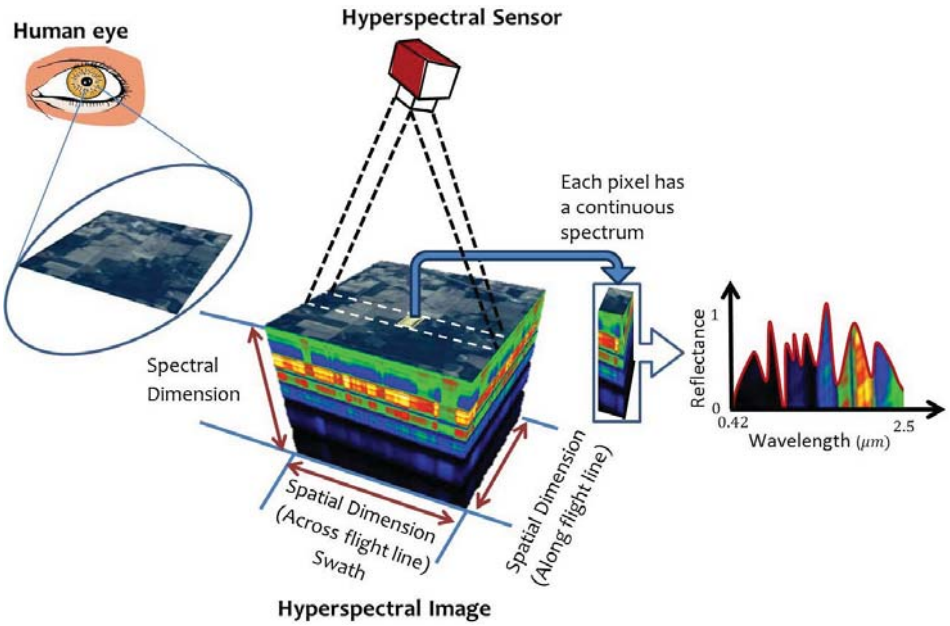

Figure 1.2.2: Hyperspectral remote sensing Acquisition.

[38], food quality and control [39], image segmentation [40], archeology[41], just to name a few.

Digital Image Processing (DIP) refers to the usage of a digital computer to process digital images. Such process can be characterized by its input-output, DIP includes processes whose inputs and outputs are images, in addition, it includes processes that extract attributes from images, up to and including the recognition of individual objects ([42]). According to these authors there is a paradigm that considers three types of computerized processes: low-, mid-, and high-level processes. A low-level process is characterized by the fact that both its inputs and outputs are images. This process involves primitive operations such as image pre-processing to reduce noise, contrast enhancement, and image sharpening. A mid-level process is characterized by the fact that its inputs generally are images, but its outputs are attributes extracted from those images (e.g., edges, contours, or, the identity of individual objects). Some tasks in this processing are segmentation and classification. And finally, higher-level processing involves interpretation of the recognized objects, as in image analysis, and, at the far end, computer vision, performing

the cognitive functions normally associated with vision. DIP overlaps with image analysis in the area of recognition of individual regions or objects in an image [42]. Therefore, in this paradigm, this thesis belong to a low-level image processing that will aid mid-level image processes as part of a system that its main goal is information extraction with the aid of computer algorithms.

In this work, low-level image processing such as image enhancement and regularization methods will be proposed so higher level processes such as classification, anomaly detection, and target detection can be improved. *Classification* of pixels in a scene is the process of assigning a class to each pixel. On the other hand, given a target material of known spectral composition *target detection* attempts to locate pixels in the scene that are similar in spectrum to the target. *Anomaly detection* in a scene tries to locate pixels that are different from all other pixels around it, usually is used when the target model is unknown. Some practical difficulties in target detection are: (*i*) target spectra is mixed with its surrounding spectra and (*ii*) classical algorithms based on Principal Components Analysis (PCA) may not work. Due to first problem, the brute force algorithm of comparing each pixel in the image with our spectral library signature can be fruitless, producing high rate of false alarms [1] or it will miss the target. The second problem arises because in many cases the target compresses few pixels in the image compared to its background, then a method such as Principal Components Analysis (PCA) will put the target signal in the smallest variance bands which frequently are not taken into account for having unwanted signals. So naive and classical approaches do not help in this problem. Given all the aforementioned applications, it is a necessity to perform target detection in an accurate and timely manner. There are many different types of target detection algorithms used in HSI (for a review see [29, 43]). Many of them can be classified either as geometrical or statistical models. Both models try

1 This is, when a pixel is classified as target when it is not

to suppress the background or clutter and enhance the contrast of potential targets. The geometrical models use structured backgrounds and the statistical models do not make any assumptions on the background and use statistical distributions to characterize it.

Image regularization or restoration refers to a set of methods in which the noisy image I is seen as a surface. Then, *regularizing* the image I is equivalent to to find a smooth surface similar enough to the original noisy one [3]. Figure 1.2.3(a) shows band 2 of the HSI Indian Pines (see Section 5.1.1), Figure 1.2.3(b) shows its surface. The surface is plotted by making the $z-$ axis equal to the intensity of each spatial position of the band. Figure 1.2.3(c) shows a regularized version of (a) using TAND for Edge Enhancing Diffusion (EED) after two iterations. Note that the majority of edges are preserved and denoised while the homogeneous regions have been smoothed. This is also observable in Figure 1.2.3(d).

Structure Enhancement is a set of methods that look for a surface close to I but only some local features are smoothed [44][2]. Figure 1.2.4 shows the band 2 of a thyroid tissue HSI in (a). (b) shows its respective noisy surface. Figure 1.2.4(b) show the same band after 30 iterations of the proposed TAND for Coherence Enhancement Diffusion (CED) and (c) its surface. Comparing the numbered regions from 1 to 4 in the images it can be seen that there is almost no smoothing on those regions. While in the edges of the cells (showed in blue) have been smoothed. In this case the image have been enhanced and not regularized since there are big regions that have not been smoothed.

This thesis is focused on developing image regularization and structure enhancement methods to preprocess HSI, such that higher level processes can be improved.

2 This definition is suggested in §3.4

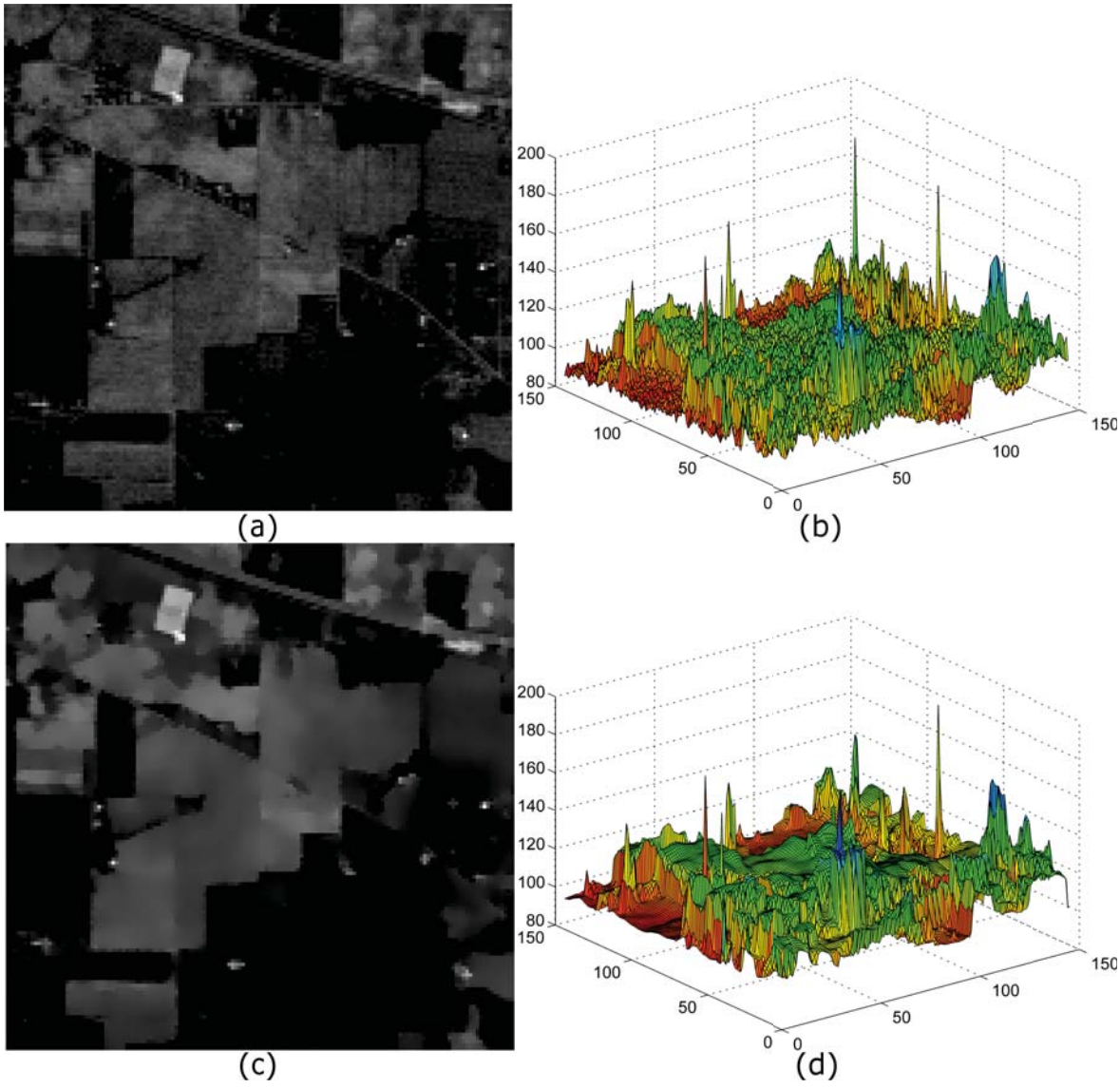

Figure 1.2.3: Image regularization, treated as the evolution of a surface.(a) Original Image. (b) Noisy surface of (a). (c) Smoothed image (d) regularized surface of (c). (b) and (c) where processed using the proposed TAND-EED in Chapter IV.

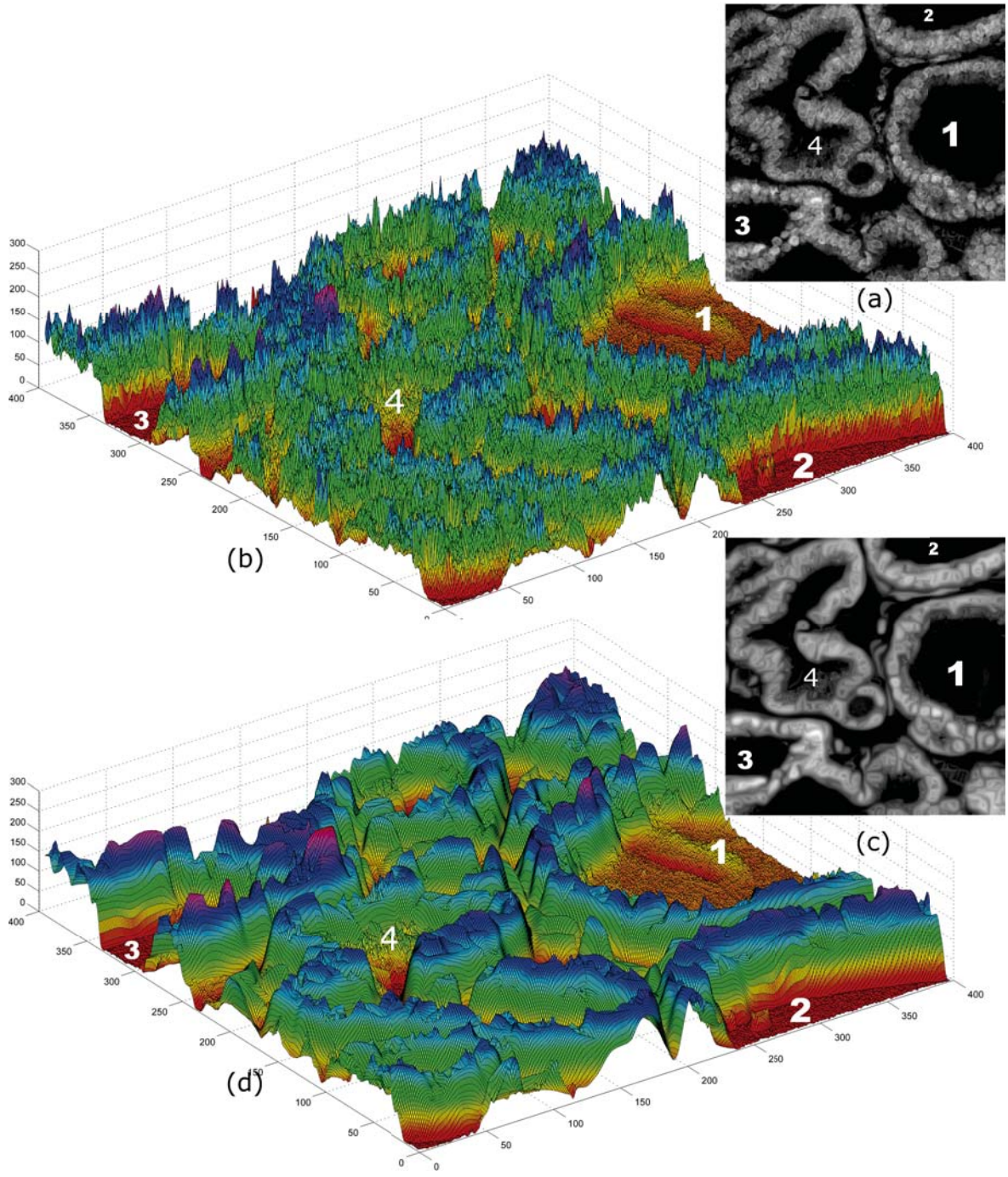

Figure 1.2.4: Structure Enhancement. (a) Original image. (b) Surface of the original image. (c) Original image after 30 iterations of the proposed TAND-CED as structure enhancement method. (d) Surface of the image in (d).

1.3 PROBLEM STATEMENT

The structure tensor for vector valued images is most often defined as the average of the scalar structure tensors in each band. The problem with this definition is the assumption that all bands provide the same amount of edge information giving them the same weights. As a result non-edge pixels can be reinforced and edges can be weakened resulting in poor performance by processes that depend on the structure tensor. Iterative processes, in particular, are vulnerable to this phenomenon. As mentioned in the last section the Structure Tensor (ST) is a tool used to aid in the solution of a variety of image processing problems, applied in the majority of cases to gray value and color images. So it is necessary to find a way to adapt this tool to vector valued images. In particular, the class of vector valued images in which their values are highly correlated in a local neighborhood of its spectral dimension. The classical definition of the structure tensor is based on spatial information.

One research question that this work is focus on is:

- How the spectral information of vector valued images can be included in the definition of a structure tensor for such images? So the structure tensor can distinguish between the interesting features that need to be preserved, while removing the unimportant ones.

Several Geometric PDE-based local diffusion methods depend on the structure tensor. One of them is known as Tensor Anisotropic Nonlinear Diffusion (TAND) for Edge Enhancing diffusion (EED) and Coherence Enhancing Diffusion (CED). Therefore, another research question presented in this thesis is:

- How TAND will be affected with a structure tensor that includes the spectral information? Does TAND need modifications to take advantage of the information produced by this structure tensor?

Geometric PDE-based local diffusion methods for image processing are highly effective [12, 2]. After the discretization of the PDE, the best way to solve them, in terms of time to compute the desired diffusion, accuracy and quality of the solution, is by using semi-implicit methods [45, 46]. The price paid is that those methods produce linear systems that need to be solved at each iteration. Depending on the size of the neighborhood used to discretize the derivatives, these linear systems have special structures. For vector valued images, $\mathbf{A}$ can be five-diagonal if 4-neighbors are used and nine-diagonal if eight neighbors are used. Nine-diagonal linear systems also result if the discretization includes mixed derivatives as is the case of TAND. When $\mathbf{A}$ is five diagonal, there are methods, such as the *Thomas algorithm* also known as *forward and backward substitution,* that solve those problems in $O(n)$ time [45]. There are no known efficient methods to solve a linear system when $\mathbf{A}$ is nine-diagonal. So, the idea is to find a method that helps the linear system to converge quicker, to find the desired diffusion. Then another question that this work will try to answer is:

- How to accelerate the convergence of a nine diagonal linear system coming from the discretization of a Geometrical PDE applied to vector valued images.

1.4 TECHNICAL APPROACH

This thesis presents a method to incorporate the spectral information inherent on a HSI in the ST. The initial matrix field (see Section 2.3.5) is calculated using a weighted smoothed gradient. The spectral weights to fuse the data from each band of the structure tensor are proposed. The weights will be defined using the heat

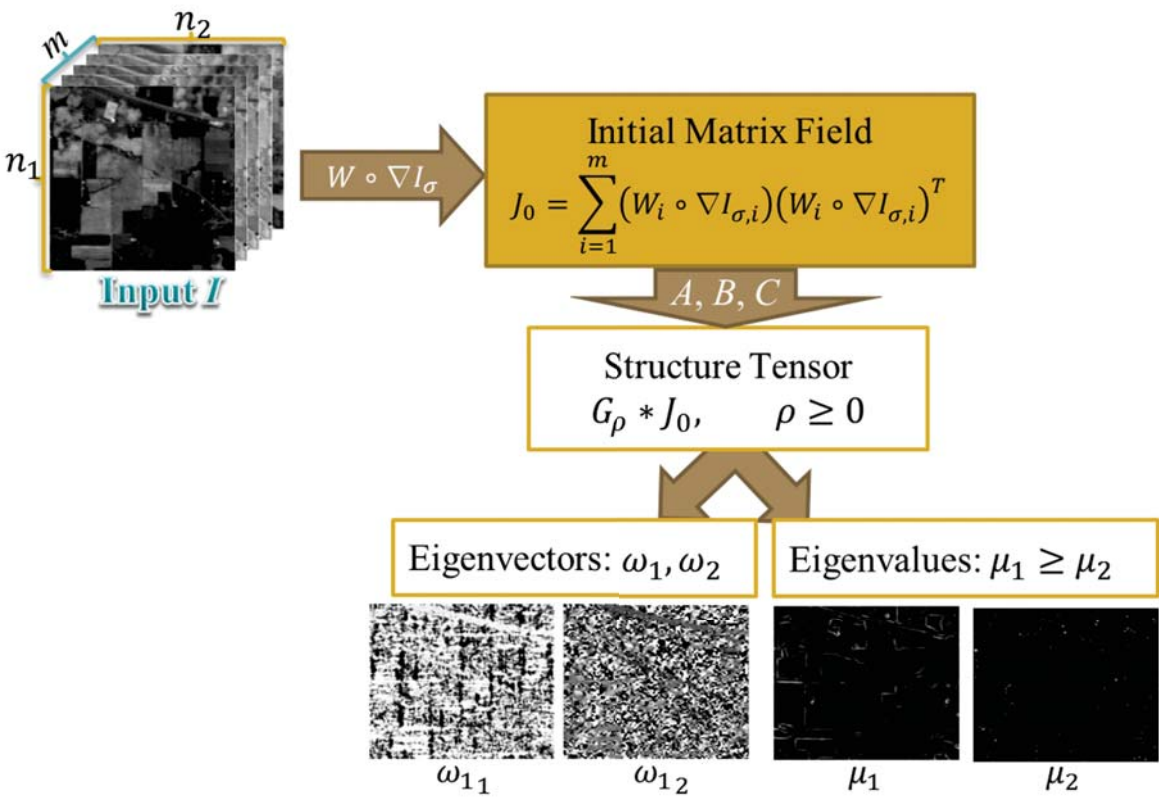

Figure 1.4.1: Main steps used to calculate the proposed Structure Tensor.

operator acting on the spectrum of each pixel of the smoothed gradient. To use the heat operator, the smoothed gradient is modeled as the initial heat distribution on a compact manifold $\mathbf{M}$. This model is motivated by the fact that in HSIs, neighboring spectral bands are highly correlated, as are the bands of its gradient. Hence, instead of weighting each smoothed gradient pixel using a uniform distribution, as in the classic definition, the heat operator acting on each pixel is used. Figure 1.4.1 shows a flow chart of the main step used to calculate the proposed structure tensor.

Using the spectrally adapted structure tensor proposed in this thesis a Tensor Anisotropic Nonlinear Diffusion (TAND) method is proposed and studied. The diffusion tensors are modified so TAND take full advantage of the information produced by the structure tensor. Diffusion tensors were proposed for TAND for Edge Enhancement Diffusion (EED) and for Coherence Enhancement Diffu-

sion (CED). Those proposed diffusion tensors used the orientation and eigenvalues of heat weighted structure tensor developed in Chapter 4. This structure tensor make TAND adaptive to the spectral characteristics of HSI. This structure tensor is presented in the linear framework since it is linear in the first iteration. However, in succeeding iterations, TAND finds the structure tensor of the smoothed image of the former iteration and the iterative process results in a non-linear structure tensor. The diffusion tensor proposed for TAND-EED produced less blurred edges than using Weickert's diffusion tensor. This was achieved by adapting the smallest eigenvalue to the features in the images. The proposed diffusion tensor for TAND-CED is more sensitive to the values of the contrast parameter used to define the edges, while Weickert's one is sensitive to its square. The experiments in this thesis show that using the heat weighted structure tensor help the diffusion tensor TAND-EED to better discriminate which edges to keep longer as TAND-EED iterate. It also help TAND-CED to produce less broken edges and to obtain a better structure enhancement than using the classical structure tensor. Figure 1.4.2 shows a flow chart of the steps need for TAND.

All aspects of TAND implementation have been studied. After implementing three methods to discretize the derivatives, the standard central difference scheme used to discretize the mixed derivatives ∂_{xy}, ∂_{yx}, and the standard central difference scheme applied to half distances to discretize ∂_x^2 and ∂_y^2 obtained good results in term of interpolations, less computational time and good results. The performance of two methods to solve non-symmetric linear systems was studied to solve the linear system, BiCGStab and GMRES. BiCGStab was chosen since it needed less iterations and less time to find a solution. A preconditioner is proposed. To study which is the best preconditioning method to accelerate the solution of TAND's linear system comparison with standard preconditioning methods, ILU(0) and the

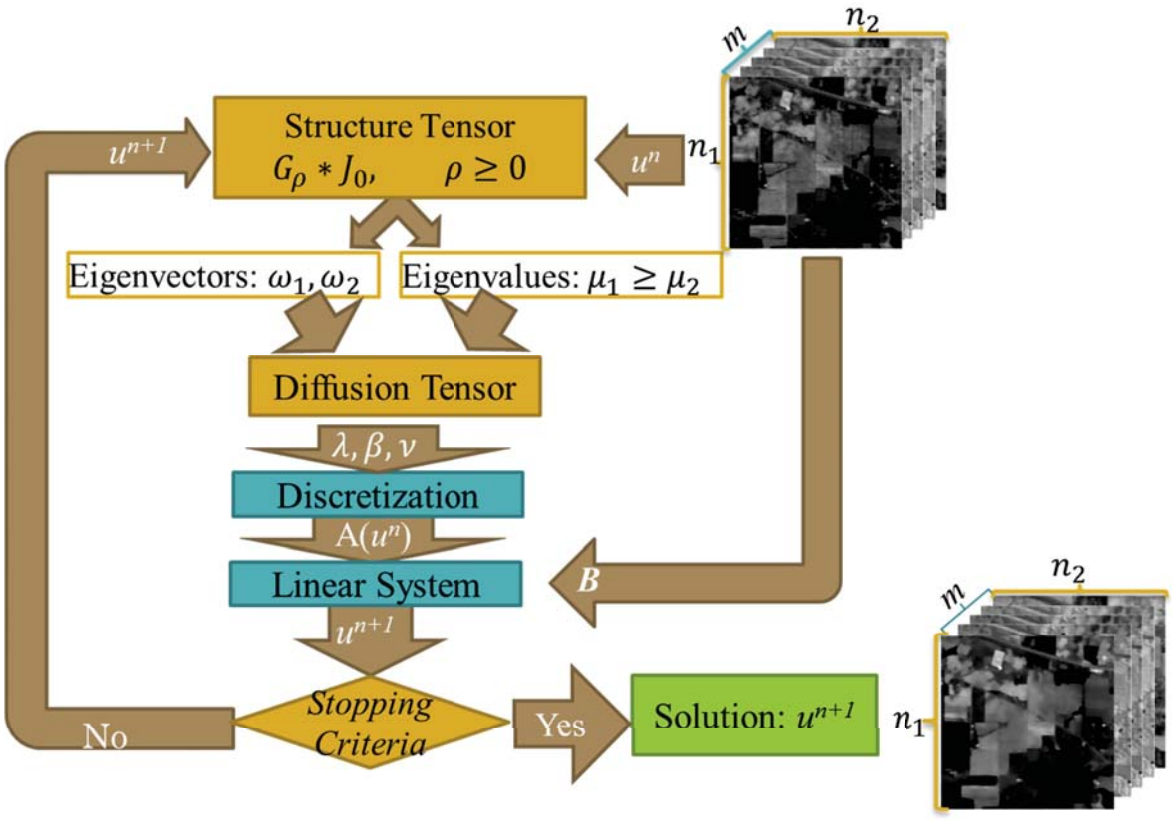

Figure 1.4.2: TAND Flow chart .

Jacobi explicit diagonalization is carry out. The ILU(0) need less iterations and less time to find the solution.

1.5 THESIS CONTRIBUTIONS

The Main contributions of this thesis is:

- A framework for the spectrally adapted structure tensor for vector-valued images.

Since the structure tensor is a tool that can be used for so many image processing tasks, it is of great importance to have a framework that adapt this kind of tool to vector-valued data by taking into account their spectral information. Having that in mind, this thesis also presents the following important contributions that deal

with the practical aspects and applications of proposing a spectrally adapted ST in particular to hyperspectral images:

- An spectrally adapted structure tensor is proposed for vector-valued images that locally are highly correlated in its spectral dimension.
- Diffusion tensors for Tensor Anisotropic Nonlinear Diffusion Method are proposed and studied.

This thesis belong to the sub-field of Computational Signal and Image Processing in the Computer Science and Engineering specialty of the Computing and Information Sciences and Engineering (CISE) Ph.D program. The engineering component of this work is represented by the models used to develop the framework and the design of the filters used. The computational component is represented by the design and implementation of the algorithms used. The representation of the edge information of HSI encoded by the proposed structure tensor and its part in the transformation of the images using TAND summarize the information science component of this work.

1.6 THESIS OVERVIEW

This thesis is organized as follows: Chapter 1 presents mathematical definitions and notation. Chapter 2 presents an overview of the classical structure tensor and several variants. It also give an overview of the state of the art of the Divergence-based PDE diffusion and of Tensor Anisotropic Nonlinear Diffusion (TAND). Chapter III presents the proposed Structure Tensor. Chapter IV introduces the proposed diffusion tensors for TAND. Chapter V presents the experimental results showing the effectiveness of the structure tensor and comparing the proposed spectrally adapted TAND with the state of the art. Chapter VI will discuss some ethical issues

on the public access to remote sensing imagery. Chapter VII presents conclusions and future work.

2

LITERATURE REVIEW AND DEFINITIONS

2.1 MATHEMATICAL NOTATION

2.1.1 *Definition of Images*

In this digital era, digital images are stored in computers using discrete representations of the data, such as vectors, matrices, etc. Recently a Discrete Exterior Calculus theory [47] have been developed due to the necessity of such a theory and also due to the fact that historically the discrete setting have always existed. In Discrete Calculus the discrete domain is treated as entirely its own domain and not as a sampling of a continuous counterpart. This theory found equivalences to the main results of Continuous Calculus using topological properties of many of those results. However, in this work this setting is not used. Instead of a discrete theory, here it is used discretization methods to approximate continuous solutions. It is assumed that the images as discrete signals can be approximated by continuous mathematical functions, or at least piecewise continuous. This hypothesis implicitly assumes that the spatial discretization is fine enough, that is, the sampling step between values is small enough [12]. This assumption is not accepted in all circles. Some discussion can be found in [48] but still the application of classical mathematical tools and continuous models in image processing has proved to be really useful to solve many problems and in this work it is used when needed.

Let $\Omega \subset \mathbb{R}^d$ be a closed spatial domain of dimension $d \in \mathbb{N}^+$. For 2D images, $d = 2$, for 3D images, that is volumes, $d = 3$ and for functions defined on a subset of $\mathbb{R}$, $d = 1$. A functional definition of scalar and vector valued images will be given.

A *scalar image* is defined as:

$$I : \left| \begin{array}{l} \Omega \subset \mathbb{R}^d \to \mathbb{R} \\ \qquad \mathrm{x} \to I(\mathrm{x}) \end{array} \right.$$

If $d = 1$ then $\mathrm{x} = x$, if $d = 2$ then $\mathrm{x} = (x, y)$ and $\mathrm{x} = (x, y, z)$ when $d = 3$. Note that it is assumed that $I(\mathrm{x})$ takes values in the continuous space $\mathbb{R}$, even if pixel values of digital images are discrete and bounded. Scalar images produce one single intensity/radiance value per pixel representing what it is known as gray-level images and some volumes. On the other hand, a *vector valued image* at each position x produces a vector of dimension $m \in \mathbb{N}^+$, that is:

$$\mathbf{I} : \left| \begin{array}{l} \Omega \subset \mathbb{R}^d \to \mathbb{R}^m \\ \qquad \mathrm{x} \to I(\mathrm{x}) \end{array} \right.$$

Therefore, in color images, that correspond to $m = 3$, a pixel at position x will be a 3-entry vector . Vector-valued images can be represented using scalar images as follows:

$$\forall \mathrm{x} \in \Omega, \mathbf{I}(\mathrm{x}) = (I_1(\mathrm{x}), ..., I_m(\mathrm{x}))^T$$

where $I_i : \Omega \to \mathbb{R}$ is a scalar image, $1 \leq i \leq m$ and the superscript T indicates the matrix transpose operation. Each scalar image is known as a *band* or *channel.* Generally, multi-valued variables will be denoted by bold letters. This includes vector-valued as well as matrix-valued images (i.e when $\mathbf{I} : \Omega \to \mathbb{R}^{d \times e}$, $d,\ e \in \mathbb{N}$).

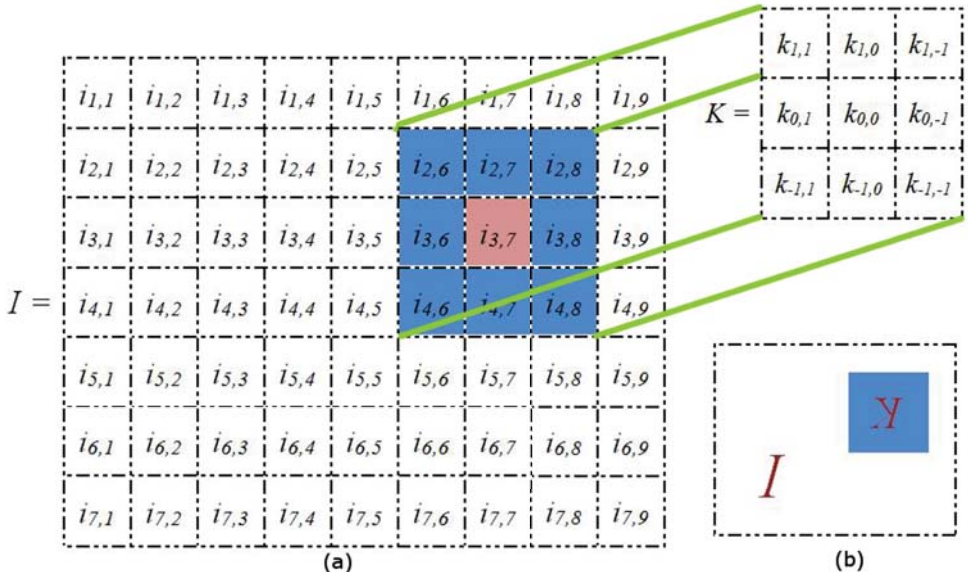

Figure 2.1.1: Convolution. (a) Convolution of a small image I and a "flipped" kernel, K. The labels within each grid square are used to identify the position of the pixels. (b) The kernel is "flipped" to calculate the convolution.

2.1.2 *Convolution*

Convolution denoted by $*$ is a simple mathematical operation used to combine two signals, f and g to form a third one. The combination is done by calculating the amount of overlap of f as it is shifted over g. Convolution of two functions over a finite range $[0, t]$, for gray level images $t = 255$. For an $J \times L$ image the convolution is given by:

$$[f * g](v, w) = \frac{1}{JL} \sum_{j,l=0}^{J,L} f[v - j, w - l] g[j, l]$$

In Image Processing convolution can be used to implement operators whose output pixel values are simple linear combinations of certain input pixel values. It provides a way to combine two arrays of numbers, generally of different sizes, but of the same dimensionality, to produce a third array of numbers of the same dimensionality. As shown in Figure 2.1.1, one of the input arrays can be a 2D image I, the second array is usually much smaller and two-dimensional although

it may be just a single pixel thick, and is known as the *kernel*. Note that this kernel has been rotated 180^0counterclockwise. The convolution is performed by sliding the kernel over the image, generally starting at the top left corner. In the strict definition of convolution, the kernel is moved through all the positions where its fits entirely within the boundaries of the image. In practice, this restriction is relaxed by extending the domain Ω, so the borders of the image I can be included in the convolution, this process is known as *boundary conditions*. There are different ways to do the extension of Ω:

- Neumann: symmetrically mirroring the pixels at the border of I.
- Dirichlet: everything outside of I is set to zero.
- Periodic: the plane is tiled with copies of I
- Reflective: the plane is tiled with copies of I, which are mirrored at each boundary. In this thesis the Neumann boundary condition will be used.

2.1.3 *Image Derivatives*

The derivative of an image I with respect to a variable w, denoted by, $I_w = \frac{\partial I}{\partial w}$, produces an image of the same size as I but with the changes of intensity in the direction w.

For a vector valued image $\mathbf{I}$, its derivative with respect to a variable w at position $p = (x_i, y_j)$, $\mathbf{I}_w(\mathrm{p}) \in \mathbb{R}^m$ is defined as:

$$\mathbf{I}_w = \left(\frac{\partial I_1}{\partial w}, ..., \frac{\partial I_m}{\partial w} \right)^T.$$

The *image gradient* is the derivative of a scalar image with respect to its spatial coordinates x, it is denoted by: $\nabla I : \Omega \to \mathbb{R}^d$and defined as:

$$\nabla I = \left[I_x, I_y\right]^T \qquad for\, d = 2$$

∇I forms a vector valued field representing the direction and magnitude of maximum variations in the image. *The norm of the gradient*, $\|\nabla I\|$ is defined as :

$$\|\nabla I\| = \sqrt{I_x^2 + I_y^2} \qquad for\, d = 2.$$

$\|\nabla I\|$ is an image that gives a scalar and point-wise measure of the image variations.

2.1.4 *Some basic kernels*

2.1.4.1 *The mean Kernel*

The mean kernel is a simple, intuitive and easy to implement method of *smoothing images*. It is often used to reduce noise in images. A 3×3 mean filter is given by:

$$K = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

2.1.4.2 *The Median Filter*

The median kernel is another method of smoothing images and reduce noise. It is particularly very effective with salt and pepper noise. The median filter does not create a weighted average as the mean kernel. Instead, it checks the $n \times n$ neighborhood around each pixel in the image and decides whether or not it is representative of its surroundings. In the median filter, the pixel value is replaced by the median of the values of its $n \times n$ neighborhood .

2.1.4.3 *The Gaussian Kernel.*

This kernel is based on the zero mean Gaussian distribution with standard deviation σ:

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} = G_\sigma(x) G_\sigma(y). \tag{2.1.1}$$

This is a 2D convolution operator that in image processing it is used to "blur" images, and remove detail and noise. The operator in (2.1.1) has a special property that it is separable, that is, it can be expressed as the convolution of two 1-D kernels. Thus, the 2-D convolution can be performed by first convolving the image with a 1-D Gaussian in the x direction, and then convolving with the trasposed 1-D Gaussian in the y direction. In theory, the Gaussian distribution is non-zero everywhere, which would require an infinitely large convolution kernel, but in practice the kernel is effectively zero more than about six standard deviations from the mean, and so it is truncated at this point [42]. An alternative is to truncate it using a threshold, a small value as 0.0001 or less is used, see Figure 2.1.2. Those kernels are normalized by the sum of the absolute vale of its entries so the kernel sum to one.

2.1.4.4 *Derivative of a Gaussian kernel*

The derivatives of the Gaussian kernel in the x and $y-$direction are calculated based on the derivative of the continuous Gaussian distribution. The derivative of the Gaussian kernel in the x and $y-$direction will be denoted for simplicity by G_x and G_y instead of $(G_\sigma)_x$ and $(G_\sigma)_y$. They are defined as follows:

$$G_x(x,y) = -\frac{x}{2\pi\sigma^3} e^{-\frac{x^2+y^2}{2\sigma^2}} = -\frac{x}{\sigma^2} G(x,y)$$

and

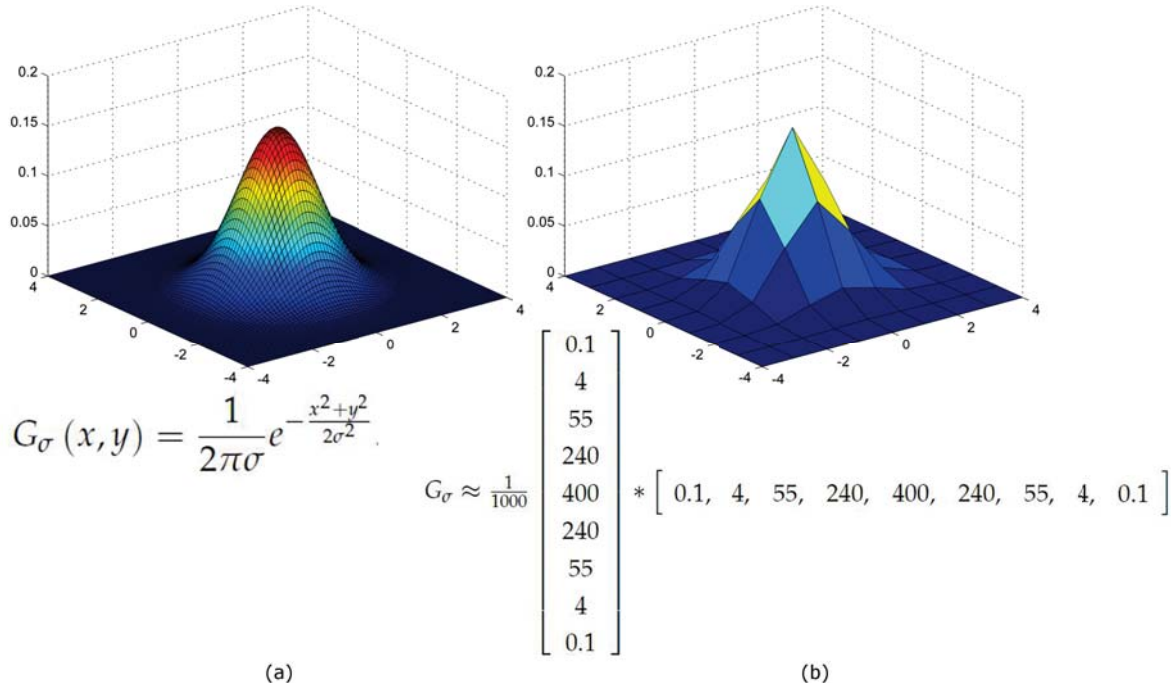

Figure 2.1.2: Gaussian distribution G_σ and its discrete approximation. (a) Gaussian distribution $G_\sigma$with standard variation $\sigma = 1$. Approximation obtained after truncating G_σ at 1×10^{-4}, producing a 9×9 Gaussian kernel K_σ for $\sigma = 1$.

$$G_y(x,y) = -\frac{y}{2\pi\sigma^3} e^{-\frac{x^2+y^2}{2\sigma^2}} = -\frac{y}{\sigma^2} G(x,y)$$

Its second derivatives are given by:

$$G_{xx}(x,y) = \frac{\left(x^2-\sigma^2\right)}{\sigma^4} G(x,y), \; G_{yy}(x,y) = \frac{\left(y^2-\sigma^2\right)}{\sigma^4} G(x,y)$$

and

$$G_{xy}(x,y) = \frac{xy}{\sigma^4} G(x,y)$$

As with the Gaussian convolution, these filters are also truncated. They are separable and symmetric. So, to calculate the derivatives of the Gaussian filter in direction x and y, it is sufficient to calculate one. The other derivative is calculated as the transpose of the one calculated

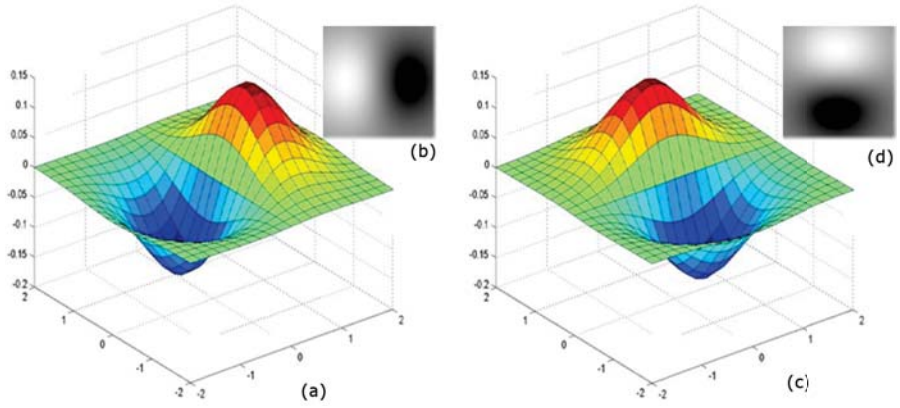

Figure 2.1.3: Derivatives of the Gaussian kernel in x and $y-$direction. (a) Plot of the $x-$derivative of the Gaussian distribution; (b) 2D G_x, the $x-$derivative of the Gaussian kernel. (c)Plot of the $y-$derivative of the Gaussian distribution; (b) 2D G_y, the $y-$derivative of the Gaussian kernel.

2.1.4.5 *The smoothed gradient.*

The smoothed gradient of a scalar image I is defined as :

$$\nabla I_\sigma = \nabla \left(G_\sigma * I \right) = \nabla G_\sigma * I \tag{2.1.2}$$

This definition is possible since the gradient is linear and translation invariant so it can be represented by a convolution. Then the associativity property of convolution can be used.

2.2 DEFINITION OF TENSOR

In this thesis, the term *tensor* will be used to designate a *symmetric and positive semi-definite matrix*. In image processing, these particular matrices take this name for their association to the *diffusion tensor* [12]. It is important not to confuse them with the multidimensional arrays from multi-linear algebra [49]. They can also be classified as a symmetric second order tensor in Tensor Analysis. For more details see [50]. The definition of a tensor has some important properties. Symmetry guar-

antees invariance to rotations and at the same time assures that all the eigenvalues of the matrix are real valued. The positive semi definiteness guarantees that all of them are non-negative. Mathematically they are summarized in the following definition:

Definition 1. Let $\mathbb{S}^d_+$ the space of all $d \times d$ tensors, this is, symmetric, positive semi-definite matrices. $\mathbf{T} = (t_{ij}) \in \mathbb{S}^d_+$ then,

$\mathbf{T}$ is *symmetric* if and only if for all $i, j \in [1, d]$, $t_{ij} = t_{ji}$

$\mathbf{T}$ is *positive semi-definite* if and only if for all $a \in \mathbb{R}^d$, $a^T \mathbf{T} a \geq 0$.

The definition of tensor provides special properties for the eigenvalues μ_k and eigenvectors v_k of $\mathbf{T}$, such as:

$\mathbf{T}$ is *real and symmetric* if and only if v_k form an *orthonormal vector basis* in $\mathbb{R}^d$. This means that for all $k, l \in [1, d]$,

$$v_k \cdot v_l = \delta_{kl} = \begin{cases} 1 & if\ k = l \\ 0 & if\ k \neq l \end{cases}$$

$\mathbf{T}$ is *positive semi-definite* if and only if for all $k \in [1, d]$, $v_k \geq 0$.

Therefore $\mathbf{T}$ may be written as:

$$\mathbf{T} = \mathbf{R}\mathbf{D}\mathbf{R}^T \tag{2.2.1}$$

where $\mathbf{D} \in \mathbb{R}^{d \times d}$ is a diagonal matrix of the eigenvalues μ_k,

$$D = diag\,(\mu_1, ..., \mu_d) = \begin{bmatrix} \mu_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mu_{d-1} & 0 \\ 0 & \cdots & 0 & \mu_d \end{bmatrix}$$

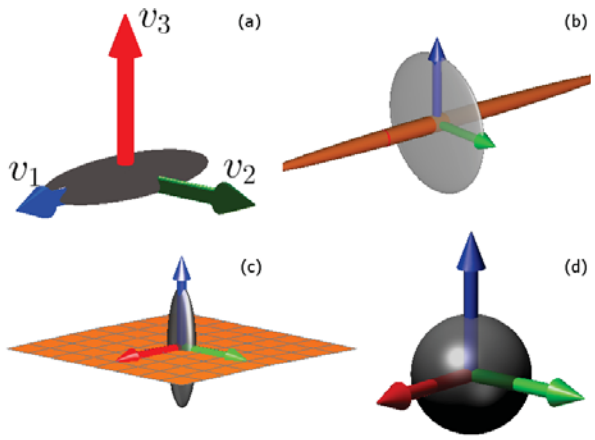

Figure 2.2.1: Ellipsoidal representation of a 3D tensor. Ellipsoid with axis (a) $\mu_1 \gg \mu_2 > \mu_3$ (b) $\mu_1 \approx \mu_2 \gg \mu_3$ (c) $\mu_1 \gg \mu_2 \approx \mu_3$ (d) $\mu_1 \approx \mu_2 \approx \mu_3$. Taken from [52]

and it has determinant, $det\,(\mathbf{R}) = 1$. $\mathbf{R}$ provides the *orientation* and $\mathbf{D}$ the *diffusivities* of the tensor $\mathbf{T}$. Note that this decomposition is known as principal component analysis

2.2.1 *Some Geometrical interpretations*

From (2.2.1) $\mathbf{T}$ can be expressed using its eigen-decomposition as:

$$\mathbf{T} = \sum_{k=1}^{d} \mu_k v_k v_k^T \tag{2.2.2}$$

μ_i are the eigenvalues providing the average contrast along the eigenvectors, v_i for $i = 1, ..., m$. Note that the eigenvalues μ_k of $\mathbf{T}$, and the corresponding eigenvectors v_k, $k = 1, ..., m$ summarize the distribution of gradient directions within the neighborhood of a pixel p [51].

2.2.1.1 *Tensor as ellipsoids*

A simple geometrical interpretation of the eigen-decomposition (2.2.2) of tensor $\mathbf{T}$ is as an ellipsoid where the semi-axes are equal to the eigenvalues and directed along their corresponding eigenvectors [53].

2.2.1.2 *Tensor as a sum of weighted elementary orthogonal tensors*

An interpretation, see [12], of the eigen-decomposition (2.2.2) is that $\mathbf{T}$ can be considered as *the sum of weighted elementary orthogonal tensors* $\left(v_k v_k^T\right)$[12]. Since the v_k are an orthonormal basis, then the d eigenvalues of $\left(v_k v_k^T\right)$ are : 1 for some eigenvector v_k and 0 for the other $d-1$ eigenvectors perpendicular to v_k. The elementary tensors $\left(v_k v_k^T\right)$ can be viewed as thin ellipsoids with one axis of length 1 and the others of length 0. In that form they represent the orientation of the tensor. A whole tensor $\mathbf{T}$ is simply a combination of these (weighted) orthogonal orientations [12]. If for all $k \in [1, d]$ the eigenvalues μ_k of $\mathbf{T}$ are equal to a constant μ then:

$$\mathbf{T} = \sum_{k=1}^{d} \mu v_k v_k^T = \mu Id_d.$$

where Id_d is the $d \times d$ identity matrix. in this case, $\mathbf{T}$ does not have a preferred diffusion direction and it is independent of the orientation. So, $\mathbf{T}$ describes *isotropic structures* and it is visualized as a sphere with a radius μ. The corresponding diffusion process is done with the same weight in all the directions of the space.

2.2.1.3 *Tensor as encoder of the local structure-Case 2-D and 3-D*

Another way to visualize the geometrical representation of the eigen-decomposition (2.2.2) of $\mathbf{T}$ is known as *glyphs* developed by Wiklund in [54]. Glyphs depict tensor variables by mapping the tensor eigenvectors and eigenvalues to the orientation and shape of a geometric primitive, such as a cuboid or ellipsoid [55]. They are

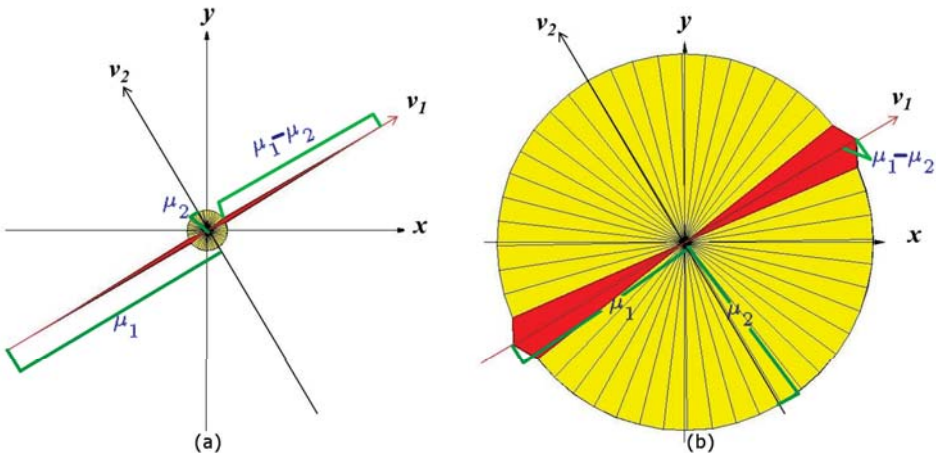

Figure 2.2.2: Representation of basic local structures of a tensor $\mathbf{T} \in \mathbb{R}^{2\times 2}$, using tensor glyphs. (a) Linear structure when $\mathbf{T}_1$ is dominant, vectors v_1, v_2 are the eigenvectors to the eigenvalues $\mu_1 = 1, \mu_2 = 0.1$ respectively. Right: Isotropic structure when $\mathbf{T}_2$ is dominant, for this case the eigenvalues are $\mu_1 = 0.67, \mu_2 = 0.63$ and have the same eigenvalues as in (a). This figure was rendered using the tensor visualization tool T-FLASH [54].

commonly used to represent the state of a tensor field point-wise and their collective behavior, when e.g. arranged in a grid, help to gain an intuition on the change of shape and orientation of the tensors [56]. Ellipses are used for $d = 2$ or d-dimensional ellipsoids for $d > 2$. The eigenvectors becomes the axis of the ellipsoids and the eigenvalues its magnitude, see Figure2.2.2. The difference with the other ellipsoid representation is that the structures for all eigenvectors are color coded. So in one representation it is depicted the behavior of all eigenvalues and eigenvectors independently and in the same ellipsoid.

When the dimension of the tensor is $d = 2$, then from (2.2.2) the tensor $\mathbf{T}$ can be written as:

$$\mathbf{T} = \mu_1 v_1 v_1^T + \mu_2 v_2 v_2^T, \quad \mu_1 \geq \mu_2 \tag{2.2.3}$$

After some algebra manipulations,$\mathbf{T}$ can be decomposed into two parts: A linear part $\mathbf{T}_1$ and the isotropic part $\mathbf{T}_2$ as

$$\mathbf{T} = \mathbf{T}_1 + \mathbf{T}_2,$$

where

$$\mathbf{T}_1 = (\mu_1 - \mu_2)v_1 v_1^T, \tag{2.2.4}$$

$$\mathbf{T}_2 = \mu_2(v_1 v_1^T + v_2 v_2^T). \tag{2.2.5}$$

This decomposition help us interpret and visualize the relative contributions of the basic local structures: Linear and isotropic [54]. Figure 2.2.2 shows the tensor glyph for each case. Figure 2.2.2(a) shows a tensor $\mathbf{T}$ where the linear local part, $\mathbf{T}_1$, is dominant. $\mathbf{T}_1$ is visualized as a red spear whose length is proportional to the magnitude of $\mathbf{T}_1$, see (2.2.4). The yellow circle shows the isotropic part, $\mathbf{T}_2$. Figure 2.2.2(b) shows $\mathbf{T}$ when $\mathbf{T}_2$ is dominant $\mu_1 \approx \mu_2$ and it is visualized as a yellow circle with radius $\mu = \mu_2$. In this case the linear part $\mathbf{T}_1$has magnitude μ_1.

For the three dimensional case $d = 3$, $\mathbf{T}$ is decomposed as:

$$\mathbf{T} = \mu_1 v_1 v_1^T + \mu_2 v_2 v_2^T + \mu_3 v_3 v_3^T, \quad \mu_1 \geq \mu_2 \geq \mu_3 \tag{2.2.6}$$

In the same way as for the 2D case, $\mathbf{T}$ can be decomposed as the contribution of three basic local structures, this is:

$$\mathbf{T} = \mathbf{T_1} + \mathbf{T_2} + \mathbf{T_3}$$

$\mathbf{T}_1$ describe the linear structures, $\mathbf{T}_2$ the planar structures and, $\mathbf{T}_3$ the isotropic structures. They are defined as [54]:

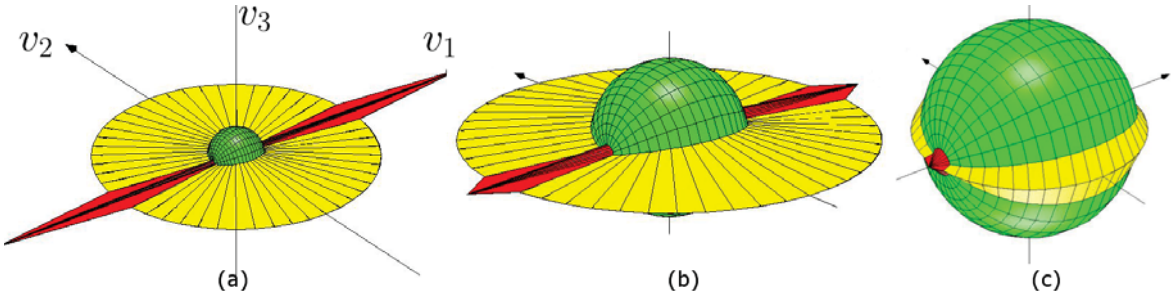

Figure 2.2.3: Representation of basic local structures of a tensor $\mathbf{T} \in \mathbb{R}^{3\times3}$, using tensor glyphs.
Eigenvectors in this example are taken as the canonical base for $\mathbb{R}^3$, $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$(a) $\mathbf{T}_1$ is dominant, the eigenvalues are $\mu_1 = 0.95$, $\mu_2 = 0.5$, and $\bar{}_3 = 0.1$. (b) $\mathbf{T}_2$ is dominant, for this case the eigenvalues are $\mu_1 = 0.95$, $\mu_2 = 0.85$, and $\bar{}_3 = 0.3$. (c) $\mathbf{T}_3$ is dominant, $\mu_1 = 0.68$, $\mu_2 = 0.65$, and $\bar{}_2 = 0.6$. This figure was rendered using the tensor visualization tool T-FLASH [54].

$$T_1 = (\mu_1 - \mu_2) v_1 v_1^T \tag{2.2.7}$$

$$T_2 = (\mu_2 - \mu_3)(v_1 v_1^T + v_2 v_2^T) \tag{2.2.8}$$

$$T_3 = \mu_3 (v_1 v_1^T + v_2 v_2^T + v_3 v_3^T) \tag{2.2.9}$$

Figure 2.2.3 shows the glyph representations of a tensor $\mathbf{T}$ with its decomposition. $\mathbf{T}_1$ is represented with the red spear, $\mathbf{T}_2$ is represented by the yellow sections, and $\mathbf{T}_3$ is represented by the green regions in the glyph. Figure 2.2.3(a) shows when $\mathbf{T}_1$ is dominant with eigenvalues $\mu_1 = 0.95$, $\mu_2 = 0.5$, and $\mu_3 = 0.1$. Figure 2.2.3(b) shows when $\mathbf{T}_2$ is dominant, for this case the eigenvalues are $\mu_1 = 0.95$, $\mu_2 = 0.85$, and $\mu_3 = 0.3$. Figure 2.2.3(c) shows when $\mathbf{T}_3$ is dominant, with $\mu_1 = 0.68$, $\mu_2 = 0.65$, and $\mu_3 = 0.6$.

2.2.2 *Classification of local neighborhoods*

The classification of the eigenvalues of the structure tensor produce very valuable information. The following classification will be based on the rank of the matrix of eigenvalues.

2.2.2.1 *Classification of local neighborhoods: 2-D Case*

The classification of local neighborhoods is done by finding the null space, that is, by looking for eigenvalues that are zero. If the gray values in the direction of an eigenvector v_k do not change then $\mu_k = 0$. The analysis of the eigenvalues for two dimensional and 3 dimensional cases will be presented and aided with results in Section 2.2.1.3. $\mathbf{T}_1$ and $\mathbf{T}_2$ are defined as in (2.2.4) and (2.2.5) respectively.

1. $\mu_1 = \mu_2 = 0$, *rank* 0 *tensor*. This is the null tensor, $\mathbf{T} = 0$. The square Frobenious norm of the gradient $\mu_1^2 + \mu_2^2$ is zero. In this case, the local neighborhood has constant values. It belongs to an object with a homogeneous feature;

2. $\mu_1 > \mu_2 = 0$, *rank* 1 *tensor*. In this case, $\mathbf{T}_2 = 0$ and $\mathbf{T}_1$ is dominant, so this tensor describes linear structures. In an image that could either be the ideal edge (without noise) of an object or an oriented texture;

3. $\mu_1 \geq \mu_2 > 0$, *rank* 2 *tensor*. There are several distinctive sub-cases to this case.

   - $\mu_1 = \mu_2$, In this case $\mathbf{T}_1 = 0$ and $\mathbf{T}_2$ is dominant, so this tensor describes an isotropic structure, i.e., it changes equally in all directions, see Figure 2.2.2(b).

   - $\mu_1 \gg \mu_2 \approx 0$. This is still a rank 2 tensor. This case describe a noisy edge or a noisy oriented texture, as shown in Figure 2.2.2(a). This is the most common case of edges on images extracted from sensors.

2.2.2.2 *Classification of local neighborhoods: 3-D Case*

To describe the analysis of the classification of the eigenvalues in three dimensions, $\mathbf{T}_1$, $\mathbf{T}_2$ and $\mathbf{T}_3$ will be used and defined as in (2.2.7), (2.2.8) and (2.2.9) respectively. Some of the neighborhoods are extension of the two dimensional case.

1. $\mu_1 = \mu_2 = \mu_3 = 0$, *rank* 0 *tensor*. There is neither a preferred orientation of signal variation nor significant variation, which corresponds to homogeneous regions [57].

2. $\mu_1 > \mu_2 = \mu_3 = 0$, *rank* 1 *tensor*. $\mathbf{T}_1$ is dominant with coefficient equal to μ_1, $\mathbf{T}_2$ and $\mathbf{T}_3$ are equal to zero. The signal values change only in the direction of v_1. The neighborhood includes a boundary between two objects (surface) or a layered texture [58]. Using the glyph representation, this will look like Figure 2.2.2(a) but in 3D ;

3. $\mu_1 \geq \mu_2 > \mu_3 = 0$, *rank* 2 *tensor*. In the general case, The signal values, i.e., gray values in the image, change in two directions which generate a plane, and are constant in a third. v_3 gives the direction of the constant gray values. Using the glyph representation this will look like Figure 2.2.3(a) but without the green region ;

   - $\mu_1 \approx \mu_2 > \mu_3 = 0$, $\mathbf{T}_1 \approx 0$, $\mathbf{T}_3 = 0$ and $\mathbf{T}_2$ is dominant with coefficient equal to μ_2. This happens at the edge of a three-dimensional object in a volumetric image [58]. Using the glyph representation this will look like Figure 2.2.3(b) but without the green region.

4. $\mu_1 \geq \mu_2 \geq \mu_3 > 0$, *rank* 3 *tensor*. There is no preferred orientation of signal variation. This is, the gray values change in all three directions. In the general case, the glyph representation this will range from Figure 2.2.3(a) to (c) any of those cases can happens. An special case to that is:

- $\mu_1 \approx \mu_2 \approx \mu_3 > 0$, $\mathbf{T}_3$ is dominant with coefficient equal to μ_3. The signal variation is equal in all directions. This case represents a corner or a junction in 3D or a region with isotropic noise [57, 58]. The glyph representation is given by Figure 2.2.3(c).

In practice, it will not be checked whether the eigenvalues are zero but below a critical threshold that is determined by the noise level in the image. Note that this tensor is suitable to distinguish very well structures that result from signal variation in one direction.

2.3 STRUCTURE TENSOR

The structure tensor in this work will refer to the local structure tensor, that is, the structure tensor defined in a neighborhood. In images, tensors are better suited to find structures inside of a neighborhood in which the gray value only changes in one direction, known as *simple local neighborhood* [59]. In these neighborhoods, oriented structures are formed since the gray values are constant along lines. This property of a neighborhood is known as *local orientation [58].*

2.3.1 *Directions vs. orientation*

The *direction* is defined over the full angle range of 2π (360°). *Orientation* will be used where the angles has range of π (180°). This distinction is done since two patterns that differ by an angle of π are indistinguishable inside a simple neighborhood. On the other hand, two vectors that point in opposite directions, i.e., differ by 180°, are different. An example of this is the gradient vector that

always points in the direction where the gray values are increasing. Thus, the direction of a simple neighborhood is different from the direction of a gradient.

2.3.2 *Estimation of the Local Structure Tensor*

The methods used to estimate the local structure tensor can be classified as:

- ***Gradient methods***: These methods use the spatial gradient to determine the signal orientation [11, 10, 8, 60].
- ***Local-energy method***: This tries to find a representation of the orientation based on the local energy method. The energy of the signal is quantified using quadrature filters then the local structure is determined. These methods are less related to our work and will not be presented here. They are based on the work of Knutsson and his group [59, 61, 62, 63, 64].

2.3.3 *Structure Tensor: The development of Gradient Based methods*

The development of the structure tensor in image processing was driven by the problem of finding the optimal local orientation in a neighborhood [58, 65, 66] and the problem of defining an edge in multichannel images [67].

2.3.3.1 *Optimization Problem*

In this framework [58, 65, 66], The structure tensor results from the solution of an optimization strategy to determine the orientation of a neighborhood; to find the orientation that shows the least deviation from the directions of the gradient. Then the idea is to maximize the squared scalar product between the gradient vector

$g = \nabla I(p')$ and the unit vector representing the local orientation, $\mathbf{r}$, of an image patch centered at location p:

$$\text{maximize}\left(g^T\mathbf{r}\right)^2 = \max_{\theta}\left[\left|g^T\right|^2\cos^2(\theta)\right], \qquad \theta = \angle(g,\mathbf{r}). \tag{2.3.1}$$

The maximum in (2.3.1) is found where g and $\mathbf{r}$ are parallel or anti-parallel, $\theta = 0°$ or $\theta = 180°$ respectively. The following integral is maximized in a local neighborhood of a point p, denoted by $\mathcal{N}(p)$, where $p \in \Omega$:

$$\max_{p'\in\mathcal{N}(p)}\int w(p-p')\left(g^T\mathbf{r}\right)^2 dp' \tag{2.3.2}$$

where the window function w determines the size and shape of the neighborhood around a point p in which the orientation is averaged. The maximization problem must be solved for each point p. In vector notation, (2.3.2) is:

$$\max\left(\mathbf{r}^T J\mathbf{r}\right) \tag{2.3.3}$$

with

$$J = \int_{-\infty}^{\infty} w(p-p')\left(g(p')\,g(p')^T\right)dp'.$$

Note that J is the convolution of the window function w with the outer product of the gradient

$$g(p') = \nabla I(p') = [I_{x_1},\cdots,I_{x_d}],$$

where $I_{x_i} = \frac{\partial I}{\partial x_i}$, for $i = 1,...,d$ and d is the dimension of the neighborhood, then J can be expressed as:

$$J = w * \left(\nabla I(p')\,\nabla I(p')\right)(p).$$

In matrix representation, this is:

$$J = \begin{bmatrix} \left(w * I_{x_1}^2\right)(p) & \cdots & \left(w * I_{x_1} I_{x_d}\right)(p) \\ \vdots & \ddots & \vdots \\ \left(w * I_{x_d} I_{x_1}\right)(p) & \cdots & \left(w * I_{x_d}^2\right)(p) \end{bmatrix}. \tag{2.3.4}$$

It is straightforward to extend the tensor for multichannel signals as:

$$J = \sum_{i=1}^{m} \begin{bmatrix} \left(w_i * I_{x_1,i}^2\right)(p) & \cdots & \left(w_i * I_{x_1,i} I_{x_d,i}\right)(p) \\ \vdots & \ddots & \vdots \\ \left(w_i * I_{x_d,i} I_{x_1,i}\right)(p) & \cdots & \left(w_i * I_{x_d,i}^2\right)(p) \end{bmatrix}.$$

The weighting function w might be different for each channel in order to consider the significance and spatial resolution of a certain channel [58]. A tensor is an adequate first-order representation of a local neighborhood since it only involves first-order derivatives. It is important to note that this tensor only analyzes in which direction(s) the signal variations occur. This is, where the gray values change. More complex structures such as structures with multiple orientations cannot be distinguished.

The problem of maximize (2.3.3), subject to $\mathbf{r}^T\mathbf{r} = 1$, can be solved by using a Lagrange multiplier μ and finding the extremes of

$$\mu\left(1 - \mathbf{r}^T\mathbf{r}\right) + \mathbf{r}^T J\mathbf{r}. \tag{2.3.5}$$

Differentiating with respect to $\mathbf{r}$, and setting the derivative equal to zero results in an *eigenvalue problem*.

$$J\mathbf{r} = \mu\mathbf{r} \tag{2.3.6}$$

This is the problem of finding for which values of μ (2.3.6) has a nonzero solution, i.e., the eigenvalues of J. Substitution in (2.3.5) shows that the solution is the eigenvector corresponding to the largest eigenvalue, μ_1. Then the structure tensor is J with the window function defined by the Gaussian distribution in (2.1.1) and standard deviation ρ. For the discrete case, a Gaussian kernel it is used, see Section 2.1.4.3.

2.3.3.2 *Structure tensor as a covariance matrix*

From a statistical point of view, when signals are regarded as random fields an interesting relationship can be established between the local structure tensor and the covariance matrix Cov.

$$Cov = J - E\left\{\nabla I\left(p\right)\right\} E\left\{\nabla I\left(p\right)^T\right\}$$

where $E\left\{\cdot\right\}$ is the expectation operator and J and $\nabla I\left(p\right)$ are defined as in Section 2.3.3.1. Cov and J are equal when $E\left\{\nabla I\left(p\right)\right\} = 0$. For example on homogeneous areas such as in neighborhood with planes or lines shape, $E\left\{\nabla I\left(p\right)\right\} = 0$, i.e., $E\left\{I_{x_1}\right\} = 0$ and $E\left\{I_{x_2}\right\} = 0$. In those areas, if the dimension of the image is $d = 2$ (scalar or vector valued) and the window function w in (2.3.4) is G_σ a Gaussian with variance σ^2 then the structure tensor can be modeled as the joint distribution of the two partial derivatives, I_{x_1} and I_{x_2}. This is,

$$J = \begin{bmatrix} G_\rho * \left(I_{x_1}^2\right) & G_\rho * \left(I_{x_1} I_{x_2}\right) \\ G_\rho * \left(I_{x_1} I_{x_2}\right) & G_\rho * \left(I_{x_2}^2\right) \end{bmatrix} = \begin{bmatrix} E\left\{I_{x_1}^2\right\} & E\left\{I_{x_1} I_{x_2}\right\} \\ E\left\{I_{x_1} I_{x_2}\right\} & E\left\{I_{x_2}^2\right\} \end{bmatrix} = Cov\left(I_{x_1}, I_{x_2}\right)$$

The convolution corresponds to computing the expectation value with regard to a weighting of the derivatives according to their distance from p.

2.3.3.3 *Using Differential Geometry*

This definition is motivated by finding a definition for an edge in a vector valued image. This framework was presented by Di Zenzo in [67]. The idea is to define a gradient based edge detector for vector-valued images such that the edge information along a given direction in one channel reinforces the edge evidence of the other channels. This is not the only way to do this extension and another definition can lead us to a different framework. Given two pixels p and q in the image f with $f(p)$ and $f(q) \in \mathbb{R}^m$. $df = f(p) - f(q)$ approximates the arc length when $dist(p,q) \to 0$, where $dist$ is the Euclidean distance. Therefore,

$$df = \sum_{i=1}^{2} \frac{\partial f}{\partial x_i} dx_i \tag{2.3.7}$$

and its square Euclidean norm is given by

$$\|df\|^2 = \sum_{i,j=1}^{2} \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} dx_i dx_j = d\mathbf{x}^T \left(\sum_l \nabla f_l \nabla f_l^T \right) d\mathbf{x} \tag{2.3.8}$$

$\|df\|^2$ describes the rate of change of the values (i.e.,intensity or radiance) in the image. Its extremes are along the direction given by the eigenvectors of maximum and minimum change, v_1, v_2 respectively, of the tensor

$$J = \sum_j \nabla f_j \nabla f_j^T. \tag{2.3.9}$$

The magnitude of the extremes are given by μ_1 and μ_2, the largest and smallest eigenvalues of J respectively.

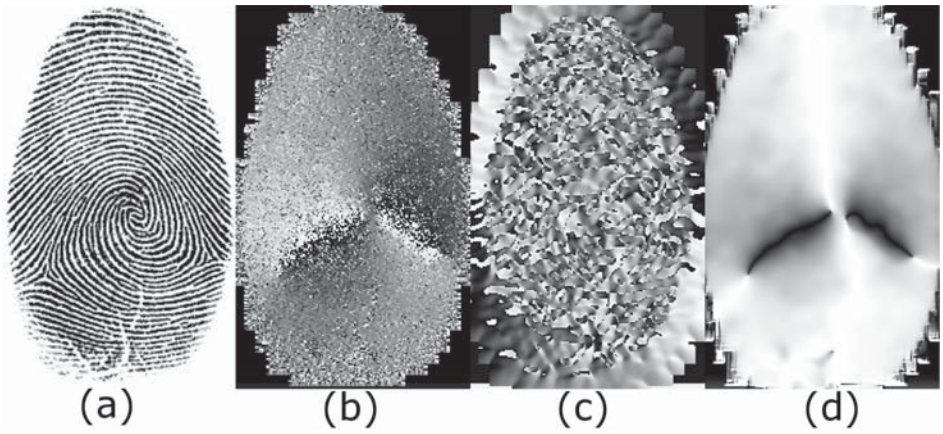

Figure 2.3.1: Smoothed Gradient vs. Structure Tensor. Fingerprint image of size 227×227. (a) Original image. (b) Gradient orientation $\sigma = 0.5$. (c) Gradient orientation $\sigma = 2.5$. (d) Structure Tensor Orientation given by ω_2 with $\sigma = 0.5$, $\rho = 4$.

2.3.4 *Scalar structure tensor*

Finding the local structure of each pixel requires considering the data of its neighborhood [13], which introduces the concept of *the structure tensor*. The task of knowing if a pixel is part of an edge or a corner cannot be accomplished with only its gradient information. There is a need to know if the pixel is similar to its neighbors or not and in which direction. Therefore, the incorporation of its neighbors' gradient information is required.

The local spatial neighborhood information becomes more important when the data is corrupted by noise since the structure has to be estimated in the presence of unreliable data. To incorporate the information within a neighborhood of a pixel, Gaussian convolution with variance σ^2, see Section 2.1.4.3, is used on the original image producing a smoothed image, i.e.,

$$I_\sigma = G_\sigma * I. \tag{2.3.10}$$

This convolution is used to weight the structure information of each pixel's neighbors. It is also used to regularize the structure tensor and make it robust to noise.

For simplicity in notation, I_x and I_y will be used though this work instead of $(I_\sigma)_x$ and $(I_\sigma)_x$ as the horizontal and vertical derivatives of the smoothed image, I_σ, respectively. The problem with the smoothed gradient, $\nabla I_\sigma = \begin{bmatrix} I_x & I_y \end{bmatrix}^T$, is that it can suffer cancellation effects. Figure 2.3.1(b)-(c) illustrates the gradient orientation using gray values where vertical gradients are depicted in black and horizontal ones in white. The gradient orientation is defined as $\tan^{-1}\left(I_y / I_x\right)$. If σ is small as in (b), then high fluctuation of the orientations remains. As σ gets larger, as in (c), then it becomes useless since neighboring gradients with the same orientation but with opposite signs (direction) cancel one another. To avoid that effect, the outer product of the smoothed gradient, which is a symmetric, positive semi-definite matrix, is considered in the structure tensor and known as the initial matrix field [66]:

$$J_0 = \nabla I_\sigma \nabla I_\sigma^T = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}. \tag{2.3.11}$$

Its eigenvalue decomposition is given by:

$$J_0 = \begin{bmatrix} \omega_1 & \omega_2 \end{bmatrix} \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix} \begin{bmatrix} \omega_1 & \omega_2 \end{bmatrix}^T$$

where $\omega_1 \parallel \nabla I_\sigma$ and $\omega_2 \perp \nabla I_\sigma$. It is important to note that $\nabla I_\sigma \neq \nabla I$ for $\sigma > 0$. So, $\omega_1 \nparallel \nabla I$. $\mu_1 = |\nabla I_\sigma|^2 = I_x$ and $\mu_2 = 0$.

Then *the structure tensor* for a neighborhood of scale ρ is computed with a smoothing step performed by a convolution of the components of with a Gaussian kernel G_ρ , which is known as the *local integration step*, as follows

$$J_\rho = G_\rho * J_0 = \begin{bmatrix} G_\rho * J_{0_{11}} & G_\rho * J_{0_{12}} \\ G_\rho * J_{0_{21}} & G_\rho * J_{0_{22}} \end{bmatrix} = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \tag{2.3.12}$$

Its eigen-decomposition is given by:

$$J_\rho = \begin{bmatrix} v_1 & v_2 \end{bmatrix} \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix} \begin{bmatrix} v_1 & v_2 \end{bmatrix}^T,$$

where

$$v_1 \parallel \begin{pmatrix} 2b \\ c - a + \sqrt{(a-c)^2 + 4b^2} \end{pmatrix} \text{ and } v_2 \perp v_1$$

and the corresponding eigenvalues μ_1 and μ_2 given by

$$\mu_1 = \frac{a + c + \sqrt{(a-c)^2 + 4b^2}}{2} \text{ and } \mu_2 = \frac{a + c - \sqrt{(a-c)^2 + 4b^2}}{2}$$

Figure 2.3.1(d) shows eigenvector v_2, which defines the coherence orientation or the structure tensor orientation for the fingerprint image. Figure 2.3.1(d) is the desired average orientation of the lines (compare with Figure 2.3.1(b)). In Figure 2.3.1(d) the fingerprint singularity (minutiae) is well described.

2.3.5 *Vector-Valued Structure Tensor*

For the vector-valued image with m bands, the structural information from all channels, encoded in the initial matrix field, is coupled with a *fusion step*, which is usually a sum over all channels. To make the structure tensor independent of the number of channels, an average is calculated, i.e., all channels are weighted equally. In this case, the initial matrix field for the vector valued case is given by:

$$J_0 = \frac{1}{m} \sum_{i=1}^{m} J_{0_i} = \frac{1}{m} \begin{bmatrix} \sum_{i=1}^{m} I_{x,i}^2 & \sum_{i=1}^{m} I_{x,i} I_{y,i} \\ \sum_{i=1}^{m} I_{x,i} I_{y,i} & \sum_{i=1}^{m} I_{y,i}^2 \end{bmatrix} \tag{2.3.13}$$

Then the structure tensor for a neighborhood of scale ρ is computed as in the scalar case, G_ρ is a Gaussian of variance ρ

$$J_\rho = G_\rho * J_0. \tag{2.3.14}$$

and its eigen-decomposition is given by:

$$J_\rho = \begin{bmatrix} \omega_1 & \omega_2 \end{bmatrix} \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix} \begin{bmatrix} \omega_1 & \omega_2 \end{bmatrix}^T. \tag{2.3.15}$$

2.4 IMAGE PROCESSING USING GEOMETRICAL PDES

A general review of main contributions in image processing using Partial Differential Equations (PDEs) in vector valued images and a review of anisotropic processing of images using divergence-driven PDEs are presented.

PDEs in image processing have been researched and used since 1960 [68]. But around 1990, as a result of a huge interest from scientists around the world, well founded mathematical and numerical methods for PDEs applied to image processing, in particular gray level images, were developed. As a consequence, many faster and efficient algorithms emerged. Processing vector-valued data has become an active research area because of the wider use of instruments that produce vector-valued data and its usage in many fields and applications. Also cheaper computer memory and some new computer architectures have helped to adapt, modify and create new algorithms and methods for this particular kind of data. Some advantages of using PDEs in processing vector-valued Images are [2]:

- A wide and robust mathematical theory of PDEs.
- Efficient and reliable numerical methods to solve PDEs.

- Theory and methods are generalizable to higher dimensions.
- Many of these methods are related to well known physical phenomena such as conservation, diffusion, etc.

Images, in general, result from a uniform sampling of a 2D or 3D signal done by sensors so they are discontinuous by nature. On the other hand, PDE-based image processing theory is developed primarily assuming continuity. Therefore, there is a need for discretization techniques to approximate a solution of the PDEs in this kind of data. The discretization uses a mesh that subdivides the image domain to numerically integrate over each element in the mesh. Since the mesh is determined by the sampling of the pixels in the image, the discretization is restricted to use a structured rectangular mesh without refinement.

2.4.1 *Physical background of diffusion*

The physical concept of diffusion can be found in electromagnetic theory, fluid dynamics, quantum mechanics, and so on. It will be presented in this section in the context of the physical phenomenon of mass transport. Diffusion can be seen as the process that equilibrates concentration of differences without creating or destroying the mass [69]. This equilibrium property is given by Fick's first law:

$$j = -D \cdot \nabla u.$$

The flux j compensates for a concentration gradient ∇u. The relation of j and ∇u is given by the *diffusion tensor* D, which is a symmetric positive definite matrix. The term tensor in this case is used to describe those kinds of matrices, not in the wider sense as in Physics [3]. *Isotropic diffusion* is known as the case where j and ∇u are

parallel and *anisotropic diffusion* if $\jmath$ and ∇u are not. This is the definition adopted by Weickert [2].

Since diffusion only transports mass without destroying it or creating new mass, $\frac{dm}{dt} = 0$, where t denotes time. Application of Reynolds' transport theorem and Gauss' divergence theorem leads to a differential coordinate-free form of the *continuity equation*

$$\partial_t u + \operatorname{div}(\jmath) = \frac{dm}{dt} = 0.$$

To adapt this form to a specific coordinate system, just provide a expression of the divergence operator in that particular coordinate system. Then by substituting Fick's flux into the continuity equation, *the diffusion equation* [69, 70] is obtained .

$$\partial_t u = \operatorname{div}(D \cdot \nabla u). \tag{2.4.1}$$

Equation (2.4.1) is a conservation law that appears when modeling many physical transport phenomena. In general, these processes involve mass or energy transport in a conductive media, without sources or sinks [69]. In the context of image processing, the mass concentration is interpreted as the intensity of the image signal at a certain location. The diffusion direction will change depending on the diffusion tensor D. If D is constant in the whole image domain is known as *Homogeneous diffusion* and a space-dependent filtering is called *inhomogeneous*. A diffusion filter is called *nonlinear* if the diffusion tensor is a function of the differential structure of the evolving image itself, and *linear diffusion* if it does not depend on it [70].

In image processing, there is no real evolution in time, the variable n is a dummy variable that simulates an iterative process as n is gradually increases from $n = 0$ in which $u_i(x, 0) = f_i(x)$ is the original image. Some diffusion processes produce a family $\{T_t f \,|t \geq 0\}$ of gradually smoother versions of f which is known as *scale-space*, see Figure 2.4.1. For a study on scale spaces see [45, 71, 72] and references

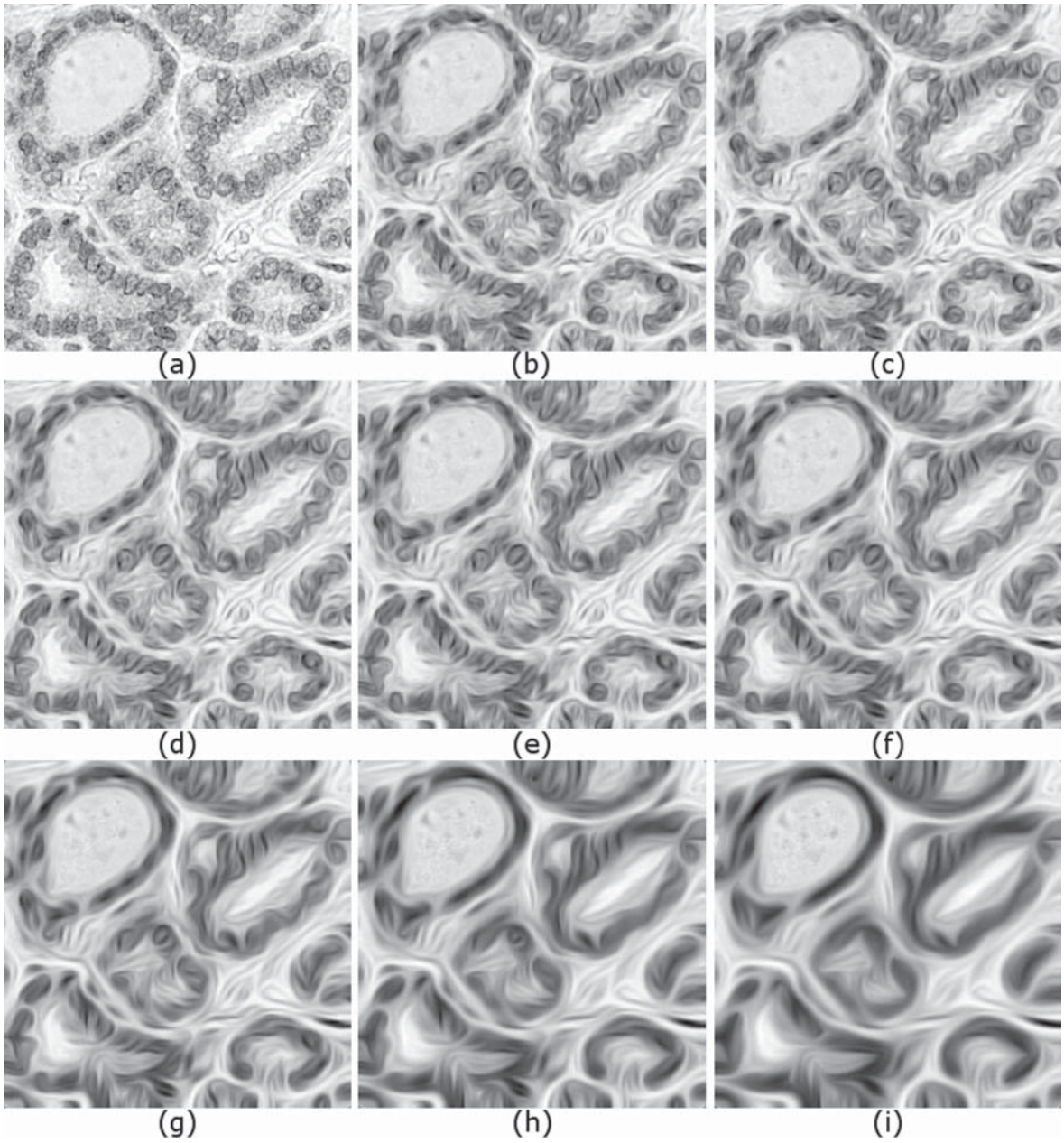

Figure 2.4.1: Scale-space produced by coherence-enhancing diffusion in a thyroid tissue. $\sigma = 0.5$, $\rho = 4$, $\Omega = (0, 252)^2$ $k = 0.1$.(a) Original Image (b)$n = 10$; (c) $n = 15$; (d) $n = 20$; (e) $n = 30$; (f) $n = 40$; (g) $n = 50$; (h) $n = 100$; (i) $n = 200$.

therein. This concept is known from linear diffusion but this filtering dislocates edges when moving from finer to coarser scales. So an edge found in a coarse scale does not match the position of the same edge in the original image. This problem is known as the correspondence problem. Restricting the scale-space idea to be linear leaves only one solution, Gaussian smoothing. Nonlinear diffusion was initially proposed to modify the linear process in order to better capture the geometry of the image itself.

2.5 DIVERGENCE-BASED DIFFUSION

In this section, the vector-valued version of the Divergence-based diffusion equations will be presented. The filtered version $u(x,t)$ of an image $f(x) \in L^\infty(\Omega)$ is the solution to the following initial boundary value problem for the divergence-based diffusion equation with f as the initial condition :

$$\partial_t u_i = div\,(D\nabla u_i)\ i = 1,...,m, \qquad \text{in } \Omega \times T, \tag{2.5.1}$$

$$u_i(x,0) = f_i(x), \qquad \text{in } \Omega, \tag{2.5.2}$$

$$\langle D\nabla u, \overrightarrow{n} \rangle = 0, \qquad \text{on } \partial\Omega \times T \tag{2.5.3}$$

where Ω is the domain of image f, $T := (0,\infty)$ is the time interval in the evolution of dummy variable t, $\overrightarrow{n}$ denotes the outer normal unit vector to boundary $\partial\Omega$, and $\langle \cdot,\cdot \rangle$ is the Euclidean scalar product on $\mathbb{R}^n$. The last equation is a Neumann boundary condition which means that the flux is zero outside the boundary. From now on, all diffusion methods are assumed to have the same initial value (2.5.2) and boundary condition (2.5.3) and Equations (2.5.2) to (2.5.3) will be called **P1**.

2.5.1 *Linear diffusion*

The differences between divergence-based methods is D. If D is a constant, then 2.5.1 is:

$$\partial_t u_i = \Delta u_i \quad i = 1, ..., m, \quad \text{in } \Omega \times T,$$

the solution using the homogeneous diffusion process is [68]:

$$u_i(\mathsf{x}, t) = \begin{cases} f_i(\mathsf{x}) & t = 0 \\ \left(G_{\sqrt{2t}} * f\right)(\mathsf{x}) & t > 0 \end{cases}$$

which is the convolution of the image with the Gaussian filter $G_{\sqrt{2t}}$.

2.5.2 *Perona and Malik Isotropic nonlinear diffusion*

Perona and Malik [73] in 1990 presented the first nonlinear diffusion method that work in images. For that reason, it is considered as a historical work. This was the first adaptive smoothing method, which is based on the idea of applying a process which itself depends on local properties of the image. This process is given by (2.5.1) in which D is the *diffusivity* function g given by:

$$g(s^2) = \frac{1}{1 + s^2/\lambda}, \quad \lambda > 0. \tag{2.5.4}$$

with $s^2 = \frac{1}{m}\sum_j \left\|\nabla u_j\right\|^2$ $j = 1, ..., m$, λ can be set as a percentile P of the image gradient magnitudes at each iteration. Perona and Malik recommended the value $P = 90\%$ [73]. Equation (2.5.1) is solved using a finite difference discretization scheme. This filter was motivated by the idea of adapting the diffusivity g to the

gradient of the actual image $u(x,t)$ instead of the original image $f(x)$ introducing feedback in the process which leads to *nonlinear diffusion*. In 1993, Alvarez [74] showed that there exist PDEs with the original image as the initial condition that govern scale-spaces which satisfy some architectural, information-reducing, and invariance properties. They showed that Perona-Malik diffusion was ill-posed and produced sharpening of edges if the gradient is larger than λ. In practice, the method behaves better than predicted by theory. Latter was shown in [70] that the standard finite difference scheme brought some regularization and that was part of the reason it worked. In 1993, Catte et al [72] showed in an elegant mathematical formulation that the Perona-Malik model could be regularized by replacing the diffusivity $g\left(\frac{1}{m}\sum_j \left\|\nabla u_j\right\|^2\right)$ by $g\left(\frac{1}{m}\sum_j \left\|\nabla u_{\sigma,j}\right\|^2\right)$ with

$$u_{\sigma,j} := G_\sigma * u_j, \quad \text{for } j = 1, ..., m$$

where $G_\sigma(x)$ is defined as in Section 2.1.4. They also showed existence, uniqueness, and regularity of a solution to the initial and boundary value problem. This regularization worked in part since Gaussian smoothing is a low-pass filter that attenuates, in a monotone way, high frequencies. So (2.5.1) becomes

$$\partial_t u_i = \operatorname{div}\left(g\left(\frac{1}{m}\sum_j \left\|\nabla u_{\sigma,j}\right\|^2\right)\nabla u_i\right), \quad \text{for } i = 1, ..., m. \tag{2.5.5}$$

σ will take care of the case when high oscillation in the data can be misinterpreted as edges, and it is viewed as a *noise parameter*. The diffusivity $g(s^2)$ is a monotonically decreasing function of its argument and the flux $\mathbf{\Phi} = g(s^2)s$ is monotonically non-decreasing in s. This last condition is required in order to ensure well-posedness in the nonlinear setting using classical frameworks [21].

2.5.3 *Tensor Anisotropic nonlinear diffusion*

Nonlinear isotropic diffusion uses a scalar diffusivity g which is adapted to the geometry of the image. Anisotropic diffusion uses a diffusion tensor D. The advantage of using this tensor is that diffusion can be directed to any of its (maximum or minimum) directions. The structure tensor when written in its eigenvalue decomposition, see (2.3.15), can be decomposed as the sum of two expressions that describe two basic local structures: linear and isotropic structures, see (2.2.4)-(2.2.5). From the linear structures found using the eigenvector corresponding to the smallest eigenvalue, oriented flow-like structures can be distinguished and enhanced using coherence enhancement. On the other hand, regularization that preserves edges, which consist in smoothing along edges inhibiting the diffusing across them is done by finding the eigenvector corresponding to the largest eigenvalue of the structure tensor.

In this section, the three most used divergence-based anisotropic diffusion methods are presented. The first model will be motivated by the extension of the diffusivity function g from a scalar function to a tensor. It also gives an intuition on how to extend this concept to design filters to enhance 1D features and denoise preserving the edges. The second one is motivated by the necessity, in some applications, to rotate the flux towards the orientation of features of interest. A case for this kind of filtering is the *Coherence-Enhancing Diffusion* (CED). This one is adapted to process 1-D features such as lines and flow-like structures. The third one is motivated as an extension to anisotropic diffusion of the Perona-Malik model. Since this method removes noise from edges it is known as *Edge-Enhancing Diffusion* (EED). CED and EED will be objects of further study.

2.5.3.1 *Anisotropic extension of the isotropic model based on scalar diffusivity*

An extension of a scalar valued function $g(s^2)$ by a tensor $g(J)$ is given by:

$$g(J) = \begin{bmatrix} \theta_+ & \theta_- \end{bmatrix} \begin{bmatrix} g(\mu_+^2) & 0 \\ 0 & g(\mu_-^2) \end{bmatrix} \begin{bmatrix} \theta_+ \\ \theta_- \end{bmatrix} \tag{2.5.6}$$

where μ_-, μ_+ are the smallest and largest eigenvalues of J and θ_-, θ_+ its respective eigenvectors. In this case, $J = J_0$, the initial matrix field for vector valued images, is a natural choice. Using the same diffusivity or diffusion tensor for all bands ensures that the evolution is synchronized. $g(J)$ will be constructed using the process described before. An example of a diffusivity function g is given by [75]:

$$g(s^2) = \alpha + \frac{1}{\sqrt{\beta^2 + s^2}}. \tag{2.5.7}$$

So, $D = g(J)$ for (2.5.1). As a consequence, linear and nonlinear isotropic models can be characterized by their diffusivities $g\left(\sum_j \nabla f_j^T \nabla f_j\right)$ and $g\left(\sum_j \nabla u_j^T \nabla u_j\right)$ respectively, while its anisotropic counterparts are given by $g(\sum_j \nabla f_j \nabla f_j^T)$ and $g(\sum_j \nabla u_j \nabla u_j^T)$, $j = 1, ..., m$.

2.5.3.2 *Coherence-Enhancing Diffusion*

The basic ingredients of Coherence Enhancing Diffusion (CED) are:

- Diffusion is done in the direction of smallest variation of the structure tensor.
- A shape-adapted Gaussian smoothing in a small neighborhood around that direction is used.

In 1993, Catte et al. [76] presented a reaction-diffusion model where the diffusion was along the eigenvector corresponding to the smallest eigenvalue of its diffusion tensor defined as: $\kappa_2 = g(|\nabla u_\sigma|^2)$. But this model did not produce a scale-space,

and its diffusion tensor D had eigen-directions adapted to $\nabla u_{\sigma,i}$ which make them sensible to changes in σ, see Section 2.3.4. In 1998, Weickert [2] presented an approach that is a semi-local analysis by means of the structure tensor combined with 1-D diffusion along one of its eigenvectors which is presented in this section.

CED is basically a 1-D diffusion, where a minimal amount of isotropic smoothing is added for regularization purposes. For this case, D is built upon the structure tensor J_ρ defined for vector-valued images as in (2.3.14). $\mu = (\mu_1 - \mu_2)$ is a measure of local coherence [23]. μ^2 is large for anisotropic structures and tend to zero for isotropic ones. Constant areas are characterized by $\mu_1 = \mu_2 = 0$, straight edges by $\mu_1 \gg \mu_2 = 0$, and corners by $\mu_1 > \mu_2 \gg 0$ [77].

$$D\left(\frac{1}{m}\sum_{i=1}^{m} J_{\rho,i}\right) := \begin{bmatrix} \lambda & \beta \\ \beta & \nu \end{bmatrix} = \begin{bmatrix} v_1 & v_2 \end{bmatrix} \begin{bmatrix} \kappa_1 & 0 \\ 0 & \kappa_1 \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix}. \tag{2.5.8}$$

In (2.5.8) $D(\cdot)$ is the diffusion tensor written as a function to emphasize its dependency on the expression inside the parenthesis. To enhance coherence structures, the process should smooth mainly along the coherence direction v_2. So κ_2 increases with μ^2. Using [20] functions, κ_1, κ_2 are given by :

$$\kappa_1 = \alpha, \qquad \alpha \in (0,1),\ \alpha \ll 1,$$

$$\kappa_2 = g(\mu^2) = \begin{cases} \alpha, & \text{if } \mu_1 = \mu_2, \\ \alpha + (1-\alpha)\exp\left(\frac{-C}{\mu^2}\right),\ C > 0 & \text{else.} \end{cases}$$

Note that by definition $\kappa_2 \in (0,1)$. $\alpha = 10^{-3}$ is the most common value used in the literature. α guarantees that the process never stops, and keeps the diffusion tensor uniformly positive definite [23]. C is a threshold parameter;

$$\text{if } \mu^2 \gg C \text{ then } \kappa_2 \approx 1, \tag{2.5.9}$$

$$\text{if } \mu^2 \ll C \text{ then } \kappa_2 \approx \alpha. \tag{2.5.10}$$

In addition, D satisfies smoothness, symmetry and uniform positive definiteness properties [2]. The smoothness of D come from the convolution with G_σ and G_ρ, which guarantees that the elements of D are C^1 functions [77].

2.5.3.3 *Edge Preserving Diffusion*

In this context, edge-diffusion is presented as special case of coherence-enhancing diffusion in which the integration parameter $\rho = 0$. In (2.5.1), diffusion at the edges is inhibited and as a consequence noise in the edges cannot be eliminated. To solve this problem, the diffusion should be parallel to the edges instead of perpendicular to them. To that end, the diffusion tensor D will depend on $\sum_{i=1}^{m} J_{0_i}$ and eigenvalues κ_1, κ_2 defined as:

$$D\left(\sum_{i=1}^{m} J_{0_i}\right) := \begin{bmatrix} v_1 & v_2 \end{bmatrix} \begin{bmatrix} \kappa_1 & 0 \\ 0 & \kappa_1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{2.5.11}$$

and

$$\kappa_1 := g\left(\sum_i |\nabla u_{\sigma,i}|^2\right),$$

$$\kappa_2 := 1$$

Figure 2.5.1: Comparison between EED and CED. (a) Original image by Monet [78]. (b) Original image after CED $n = 10, \sigma = 0.5, \rho = 3, \psi = 50^{th}$ percentile. (c) Original image after EED,$n = 10, \sigma = 1, \psi = 50^{th}$ percentile.

If σ tends to zero, the isotropic Perona-Malik process is obtained. Many of the edge enhancing diffusion models differ in the function g, the following function was proposed by Weickert [2]:

$$g(s) := \begin{cases} 1, & s \leq 0; \\ 1 - exp\left(\frac{-C_m}{(s/\lambda)^m}\right), & s > 0. \end{cases} \tag{2.5.12}$$

The exponential function is chosen to enforce smoothness. D is positive definite since $\nabla u_{\sigma,i}$ is bounded on $\Omega \times I$ and $|\nabla u_{\sigma,i}|^2 > 0$. C_m is calculated such that the flux $\Phi(s) = g(s) \cdot s$ increases for $s \in [0, \lambda]$ and decreases for $s \in (\lambda, \infty)$. Then the values used are $m = 4$, $C_m = 3.31488$. Therefore, this diffusion can be viewed as an anisotropic regularization of the Perona-Malik model [2].

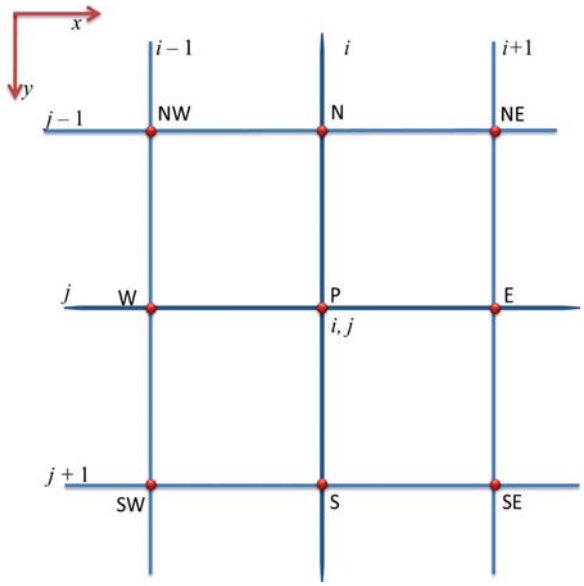

Figure 2.6.1: Spatial grid of the 3 × 3 neighborhood of a pixel P. The red circles indicates the pixels which belong to the 3 × 3 neighborhood of the vertex (or pixel) P $= (i, j)$. N, S, E W, NW, NE, SW, SE corresponds to the geographical coordinates of the 3 × 3 neighborhood around P.

2.5.3.4 *Difference between EED and CED*

The difference between EED and CED is illustrated in Figure 2.5.1. Figure 2.5.1(a) shows Monet's painting "Woman with parasol looking left" [78]. Figure 2.5.1(b) and (c) shows the resulting image after CED and EED processing respectively using the classical structure tensor. The aim of edge enhancement diffusion is to preserve edges while smoothing the inner regions they enclose as shown in the clouds of Figure 2.5.1(c). Coherence enhancing diffusion tries to enhance the 1-D features along ω_2, the direction of minimum change in the structure tensor (see (2.3.15)) which produces the effect of closing lines as seen in the grass and clouds of Figure 2.5.1(b).

2.6 FINITE DIFFERENCE SCHEMES IN IMAGES

Finite difference methods use the strong or differential form of the governing equations (2.5.1 - 2.5.3), they approximate the solution of differential equations by replacing derivative expressions with approximately equivalent difference quotients, which are obtained from an expansion of the function (u in our case) using Taylor polynomials.

For images, it is customary to assume that the pixel size in the x and y direction are 1. So the grid spacing in the x and y directions is equal and $\Delta x = \Delta y = 1$. Figure 2.6.1 shows the grid used in this work. The grid is aligned with the matrix used to represent the image, where the $y-$axis is inverted. Depending on the convention used for the $y-$axis, the values of the indexes can change. To avoid ambiguities, the geographical labels N, S, E W, NW, NE, SW, SE, corresponding to North, South, East, West, etc, are presented for the pixels belonging to the 3×3 neighborhood of pixel $P = (i, j)$, see Figure 2.6.1. The majority of results in this work will be given using the geographical labels.

Let u_p^n denotes the approximation of u at pixel $p = (i, j)$ at time $t_n = n\tau$. The approximation $\delta_x u$ of the derivative $\partial_x u$ can be approximated using different schemes such as:

$$
\begin{aligned}
\delta_x^+ u_{i,j} &= u_{i+1,j}^n - u_{i,j}^n && \text{forward scheme} \\
\partial_x u\,|_{i,j} \approx \delta_x u_{i,j} &= \frac{1}{2}\left(u_{i+1,j}^n - u_{i-1,j}^n\right) && \text{central scheme} \\
\delta_x^- u_{i,j} &= u_{i,j}^n - u_{i-1,j}^n && \text{backward scheme}
\end{aligned}
\tag{2.6.1}
$$

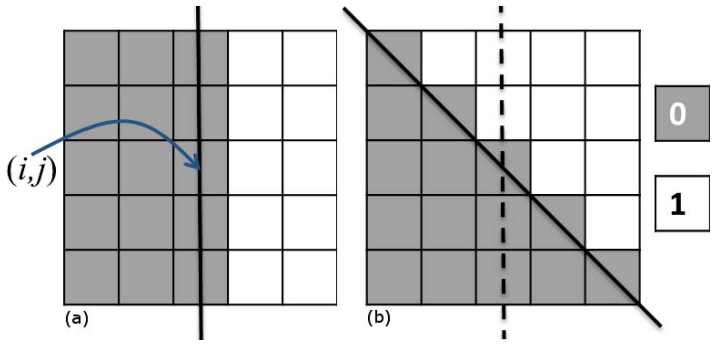

Figure 2.6.2: Rotational invariant property. Example of binary image representing a vertical edge and the same image after a rotation of $\pi/4$ radians. Rotationally invariant operators estimated at the point (i,j) in the middle (indicated by the arrow) should yield the same value in both situations. Taken from [68].

The same is done for $\partial_y u$, in which i is constant and j changes. From these schemes, the one that is symmetric is the central scheme so it is the most used scheme to approximate $\partial_x u$. Those approximations are extensions of the one dimensional case.

It is desired that the discretization of $\partial_x u$ and $\partial_y u$ is rotationally invariant. Since the norm of the gradient, $\|\nabla u\|$, is invariant under rotation. That means, if a rotation with center in (i,j) is applied to the image u with any angle $\theta \in [0, 2\pi[$, then $\|\nabla u\|$ keeps constant for all θ. So the discretization should do the same too [68]. Since this is a discrete setting, it is desirable that $\|\nabla u\|\,|_{i,j}$ keeps constant under rotations of $\pi/4$ as shown in Figure 2.6.2.

It is easy to see that using the central difference scheme in (2.6.1) for the two cases depicted in Figure 2.6.2 $\|\nabla u\|\,|_{i,j} = 1/2$ for (a) and $\sqrt{2}/2$ for (b). A rotationally invariant scheme is proposed in [68] where the derivative is approximated using the following scheme:

$$\begin{aligned}\partial_x u\,|_{i,j} \approx & \frac{a}{2}\left(u^n_{i+1,j} - u^n_{i-1,j}\right) + \\ & \frac{1-a}{4}\left[\left(u^n_{i+1,j+1} - u^n_{i-1,j+1}\right) + \left(u^n_{i+1,j-1} - u^n_{i-1,j-1}\right)\right] \\ \approx & \frac{a}{2}\left(u^n_E - u^n_W\right) + \frac{1-a}{4}\left[\left(u^n_{SE} - u^n_{SW}\right) + \left(u^n_{NE} - u^n_{NW}\right)\right]\end{aligned} \tag{2.6.2}$$

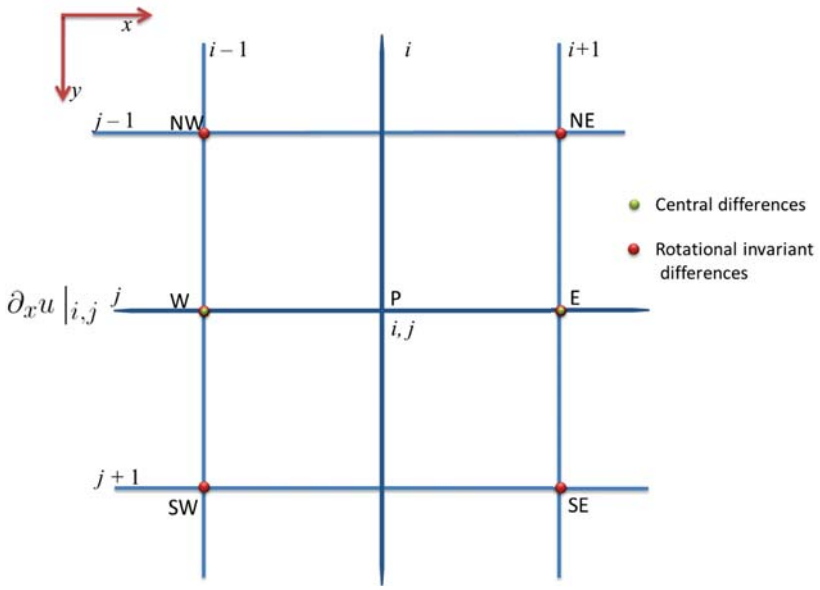

Figure 2.6.3: Pixels involved in the the finite difference schemes.

where $a = \sqrt{2} - 1$ is chosen after applying this derivative to the two cases shown in Figure 2.6.2 and making them equal. Figure 2.6.3 shows the grid pixels used in both schemes.

The usage of this rotational invariant scheme has computational disadvantages in the sense that they require three times the amount of operations per pixel than the classical central difference scheme, which can be an issue with high dimensional images such as vector-valued ones.

3

THE SPECTRALLY WEIGHTED STRUCTURE TENSOR

3.1 STRUCTURE TENSOR

An extension of the structure tensor for vector–valued images, such as color images, was presented by Di Zenzo [67]. His idea was to have one edge descriptor for all bands, so any process following the edges found by the structure tensor will behave the same way in each channel and have the information from all channels. Processing each channel independently has several drawbacks such as edge dislocation and unwanted discontinuities [71]. Di Zenzo's structure tensor is based on the idea that edge information along a given direction in one channel must reinforce edge evidence in the other channels. Hence, all edge information can be fused by using pixel-based averaging along the spectral values of its initial matrix field. Di Zenzo's method assumes that each band provides the same amount of edge information and gives each band the same weight [67]. This may work well with RGB images that only have three bands, but in the case of an HSI with hundreds of bands, edge information may not appear in all bands. Depending on the spectrum interval being sampled and the object's materials composition, some bands of an HSI will show all or parts of its edges and some will not. Therefore, weighting the spectra depending on the amount of edge information they provide may prove a more effective extension of the structure tensor to HSI's than simple averaging.

Structure tensors are classified as linear and nonlinear. Linear structure tensors are defined formally in Section 3.1.2. Nonlinear structure tensors are usually used

in iterative processes where the structure tensor output from iteration n is used as the input for iteration $n+1$. The weighting scheme proposed in this work can be used with both linear and nonlinear structure tensors that are applied to vector-valued data. As with all Di Zenzo's structure tensors, there is a caveat, the range of the data in each channel has to be similar. If not, it is necessary to rescale the data.

In HSI images, the structure tensor can be weighted in the spatial dimensions or the spectral dimensions. Spatially weighted structure tensors have been proposed in [65, 66] mainly to make the structure tensor spatially data-adaptive while a spectrally weighted structure tensor [79] is used to fuse the scalar structure tensor information from many bands. The method proposed in this research belongs to the latter with the added capability to be data–adaptive in the spectral direction.

Structure tensors based on robust statistics are one of the most important spatial weighting techniques since the structure tensor is nonlinear and the weights are based on the data. They were proposed by [65] and further developed in [66, 80, 81]. The structure tensors based on nonlinear diffusion, introduced by [21] and [82], belong to the data-adaptive methods, i.e, methods that try to adapt the structure tensor to the information in a neighborhood. Structure tensors based on robust statistical methods avoid the data adaptive processing by estimating the orientation in a neighborhood using robust error measures.

Spectrally weighted structure tensors have been studied less since much of the structure tensor work focuses on color images with only three bands. Averaging three bands is simple, computationally inexpensive and produces good results. There is, however, a spectrally weighted structure tensor for color images proposed by [79]. In this case, a-priori knowledge of the reliability and noise level in each channel is required, which limits its use. Recently, a weighted zero mean structure tensor for vector valued images has been proposed by [83] using weights based on the median of the data. But it suffers from artifacts if the mean is in the data. In

the following Sections, the classical structure tensor for vector valued images will be defined and some of the best known variants will be described. The spectrally weighted structure tensor will be defined. A weighting process will be presented for the case in which the spectrum of the images are locally highly correlated.

3.1.1 *Di Zenzo Structure Tensor*

Di Zenzo [67] presented a definition of the gradient that uses concepts from classic differential geometry, see (2.3.3.3) given by:

$$J = \sum_j \nabla f_j \nabla f_j^T$$

for an image f in a neighborhood of a pixel at p. *The edge strength* can be defined as a function g of the maximum and minimum eigenvalues of J. For the case of 2D images this will be $\mu_1 \geq \mu_2$. Many functions have been proposed [45, 72, 84]. For example,

$$g(\mu_1, \mu_2) = \mu_1 + \mu_2 = \sum_i^m |\nabla f_i|^2.$$

For the scalar case where $m = 1$, g reduces to $|\nabla f_i|^2$, which is the measure of edge strength for scalar images. For the nonlinear case, the solution u is used instead of f in each iteration n, where for $n = 0$, $u = f$. The following tensor is known as *Di Zenzo's structure tensor*

$$J_0 = \sum_{j=1}^m \nabla u_j \nabla u_j^T. \tag{3.1.1}$$

3.1.2 *Classical structure tensor*

This section will recall some definitions presented in Section 2.3.5 so there is an easier flow of ideas. The extension of the structure tensor to vector valued images is done using Di Zenzo's structure tensor :

$$J_0 = \sum_{i=1}^{m} \nabla u_{\sigma,i} \nabla u_{\sigma,i}^T. \tag{3.1.2}$$

where $\nabla u_{\sigma,i}$ is the smoothed gradient of band i defined in (2.1.2). The fusion of the edge information in each channel is done by averaging Di Zenzo's structure tensor along the spectral direction. Following the notation in Section 2.3.5 the initial matrix field for the vector valued images is obtained by :

$$J_0 = \frac{1}{m} \sum_{i=1}^{m} J_{0_i} = \frac{1}{m} \begin{bmatrix} \sum_{i=1}^{m} I_{x,i}^2 & \sum_{i=1}^{m} I_{x,i} I_{y,i} \\ \sum_{i=1}^{m} I_{x,i} I_{y,i} & \sum_{i=1}^{m} I_{y,i}^2 \end{bmatrix} \tag{3.1.3}$$

Then the structure tensor for a neighborhood of scale ρ is computed as:

$$J_\rho = G_\rho * J_0 = \begin{bmatrix} G_\rho * J_{0_{11}} & G_\rho * J_{0_{12}} \\ G_\rho * J_{0_{12}} & G_\rho * J_{0_{22}} \end{bmatrix} = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \qquad \rho > 0. \tag{3.1.4}$$

Since J_ρ is symmetric positive definite, then it is possible to find an eigen-decomposition with orthonormal eigenvectors ω_1, ω_2 given by:

$$J_\rho = \begin{bmatrix} \omega_1 & \omega_2 \end{bmatrix} \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix} \begin{bmatrix} \omega_1 & \omega_2 \end{bmatrix}^T \qquad \rho \geq 0, \tag{3.1.5}$$

where

$$\omega_1 \parallel \begin{bmatrix} 2b \\ c - a + \sqrt{(a-c)^2 + 4b^2} \end{bmatrix} \text{ and } \omega_2 \parallel \begin{bmatrix} a - c - \sqrt{(a-c)^2 + 4b^2} \\ 2b \end{bmatrix} \quad (3.1.6)$$

the symbol $\parallel$ means 'parallel to' and the corresponding eigenvalues μ_1 and μ_2 are given by

$$\mu_1 = \frac{a + c + \sqrt{(a-c)^2 + 4b^2}}{2} \text{ and } \mu_2 = \frac{a + c - \sqrt{(a-c)^2 + 4b^2}}{2}.$$

J_ρ is the average of the gradient orientations in a neighborhood of size $O(\rho)$. Its eigenvalues measure the average contrast, i.e., the gray value fluctuation for all bands along the eigen-directions within the integration scale ρ. If $\mu_1 > \mu_2$, then ω_1 is the direction of higher average contrast, and ω_2 gives the preferred local orientation or *coherence orientation* [20]. The structure tensor defined in (3.1.4) has two advantages: (i) robustness under noise as a result of smoothing the resulting matrix field, and (ii) additional information is created by averaging the local orientation. It then becomes possible to distinguish areas where structures are oriented uniformly like in the regions between edges. This effect allows estimation of the dominant orientation on those points in the image where the gradient is close to zero [66]. Its major drawback is that it can lead to inaccurate estimation when the local neighborhood is not homogeneous like near the boundary between two different textures or two different moving objects [66], see Section 2.2.2.

3.1.3 *Methods to Enhance the Local Orientation*

The problem of enhancing the local orientation estimation using the structure tensor has been approached from two perspectives:

1. Adapt the neighborhood to the data (i.e., to the image's local geometry) or
2. Choose one of the ambiguous orientations keeping the non-adaptive window [66].

In order to adapt the neighborhood to the data, several methods have been proposed such as those using Kuwahara-Nagao operator [85, 86, 87, 88], adaptive Gaussian filters [89, 90, 91, 92], and nonlinear diffusion [21, 82]. In the last method, the structure tensor is part of an iterative process that instead of applying a Gaussian convolution, as in (2.3.14), applies nonlinear diffusion. The Gaussian convolution is used for the integration step so neighborhoods become data-adaptive. While Gaussian smoothing is equivalent to diffusion with constant diffusivity, nonlinear diffusion reduces the amount of smoothing in the presence of discontinuities in the data, i.e., edges. The discontinuities are determined iteratively in the updated and smoothed data, and therefore, one can integrate data from an arbitrarily shaped neighborhood [66].

The other framework uses *robust statistics* to estimate one of the ambiguous orientations, so it is not necessary to find an adaptive window [65, 66, 80]. Structure tensors based on robust statistics are motivated by a result from [65] that shows that least squares estimation procedures for local orientation reduce to solving an eigenvalue problem for the initial matrix field smoothed by a Gaussian kernel , i.e., the structure tensor, see Section 2.3.3.1. Least squares estimation procedures use a quadratic error measure which is very sensitive to outliers. In [65], the quadratic error measure is changed for a more robust one which imposes smaller penalties on outliers to reduce their influence in the error measure. Robust error measures like Gaussian robust error [65] or Geman-McClure [80] 'clamp' the influence of large outliers to a maximum of one. Then an iterative approach is required to solve for a robust structure-tensor matrix as it becomes non-linearly dependent on the local orientation of the neighborhood.

The next section will present a general definition of the weighted structure tensor. Section 3.2.1 describes weights based on the median.

3.2 THE SPECTRALLY WEIGHTED STRUCTURE TENSOR

The initial matrix field in Equation (3.1.4) describes the fusion step in the classical structure tensor. The proposed method modifies this step. Instead of using the uniform distribution and assigning the same weight to each band, a weighted initial matrix field in each channel will be defined as the scalar tensor product:

$$J_{ok} = [W_k \circ \nabla u_{\sigma,k}] \, [W_k \circ \nabla u_{\sigma,k}]^T$$

where $W_k = [w_{x,k}, w_{y,k}]^T$ are the weights for band k. The initial matrix field for the vector-valued case with weights w_x, w_y, and $w_{xy} = w_x w_y$ is calculated as

$$J_0 = \sum_{k=1}^{m} J_{0k} = \begin{bmatrix} \sum_{k=1}^{m} w_{x,k}^2 \circ u_{x,i}^2 & \sum_{k=1}^{m} w_{xy,k}^2 \circ u_{x,i} u_{y,i} \\ \sum_{k=1}^{m} w_{xy,k}^2 \circ u_{x,i} u_{y,i} & \sum_{k=1}^{m} w_{y,k}^2 \circ u_{y,i}^2 \end{bmatrix} = \begin{bmatrix} A & B \\ B & C \end{bmatrix}, \tag{3.2.1}$$

The structure tensor, J_ρ will be calculated as in (3.1.4). J_ρ for $\rho \geq 0$ is still a tensor as it is a symmetric, positive semi-definite matrix therefore it has the same eigen-decomposition as in (2.3.15).

3.2.1 *Weights Based on the Median.*

The weights presented in this section are taken from [83]. This was the first try to make the structure tensor adaptive in the spectral dimension. These weights are

used on a zero mean smoothed gradient. Let $\overline{I_x}$, and $\overline{I_y}$ be the mean of I_x and I_y respectively, then the components of the zero mean smoothed gradient, $[I_x, I_y]^T$, are given by: $I_x = I_x - \overline{I_x}$ and $I_y = I_y - \overline{I_y}$. Let $p \in \mathbb{R}^m$ be a pixel belogning to the gradient. The weights at p, denoted by W_p, are calculated as follows:

$$Step\,1. \quad W_p = \begin{cases} p & |median\,(p)| < 1 \\ \frac{1}{|median(p)|} & |median\,(p)| > 1 \end{cases} . \tag{3.2.2}$$

In (3.2.2), a threshold of 1 was chosen to avoid division by small numbers. Next W_p is normalized as follows:

$$Step\,2. \quad W_p = \begin{cases} W_p & |max\,(W_p)| < 1 \\ \frac{W_p}{|max(W_p)|} & |max\,(W_p)| > 1 \end{cases}$$

$$Step\,3. \quad W_{p,i} = \begin{cases} 0 & W_{p,i} < a \\ W_{p,i} & W_{p,i} > a \end{cases} \quad for\ a < 0,\ i = 1, ..., m$$

Absolute values are used since the gradient may have negative values. The threshold a is chosen by finding the histogram of W_p and clipping its minimum value at the first bin in the tail with height $\sim 1\%$ of the total of pixels. The idea of the median as a better descriptor of non-symmetric data is behind the definition of these weights. A problem with using zero mean smoothed gradients is that it can produce artifacts in the structure tensor if the mean is part of the data.

3.3 SPECTRALLY WEIGHTED STRUCTURE TENSOR BASED ON THE HEAT OPERATOR

In this section, the adaptive spectrally weighted structure tensor is presented. The classical structure tensor is adaptive to the spatial features in the image. The adaptivity of the structure tensor presented here is in the spectral dimension. This is done by defining weights for each channel of the initial matrix field. The weights will be defined using the heat operator.

3.3.1 *Motivation*

Figure 3.3.1 is used to motivate the proposed spectral weights. It presents scatter plots among several bands of the Indian Pines image (see Section 5.1.1). It also shows that the same correlation holds among the bands of the gradient component I_x. The first row shows gray level images of bands 15, 50, 51, 60, and 110 corresponding to 567.38 nm, 879.53 nm, 889.14 nm, 985.27 nm and 1541.59 nm. Band 45, centered at 831.41 nm, is not shown but looks very similar to band 50. For visualization purposes, all images have been contrast stretched so that the minimum value in the band is black, and the maximum value is white. Note that Figures 3.3.1 (b), (c) and (d) are visually similar. Note that the edges inside the red square do not appear in all bands. The second row of Figure 3.3.1 from left to right shows scatter plots between band 50 and (a) band 15, (b) band 45, (c) band 51, (d) band 60, (e) band 110. The same range $[0, 255]$ is used for the x-axis and y-axis in all scatter plots. As expected, the scatter plots showing the highest correlation are (b), (c) and (d) which correspond to the bands closer to band 50. The scatter plots with bands farther away from band 50 may not show such a correlation as shown in Figures 3.3.1 (a) and (e). The third row of Figure 3.3.1 shows I_x's for the bands in the first

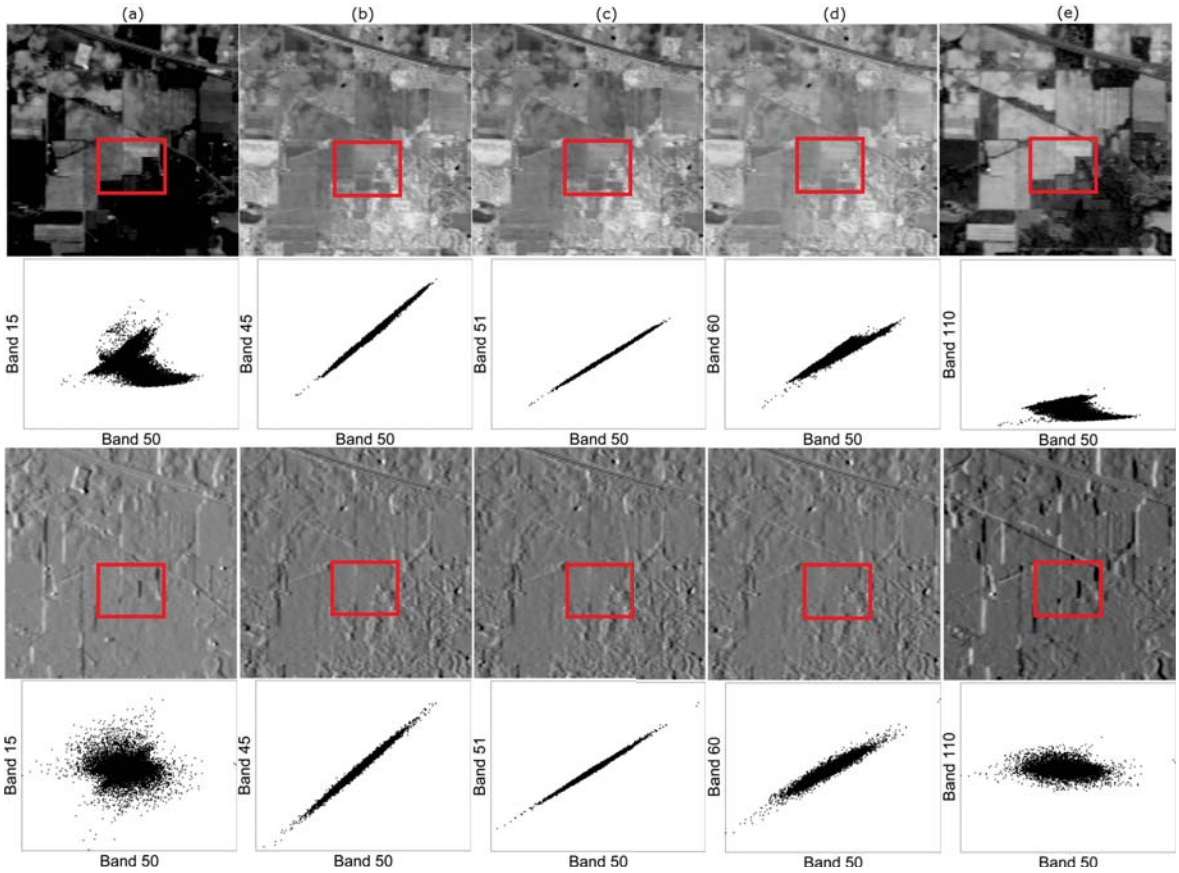

Figure 3.3.1: Bands of the same HSI compared to the respective bands of its gradient component I_x. FIRST ROW (left to right): Indian Pines' bands (a) 15, (b) 50, (c) 51, (d) 60, (e) 110. SECOND ROW: Scatter plots between Indian Pines' band 50 and (left to right) (a) band 15, (b) band 45, (c) band 51, (d) band 60 (e) band 110. THIRD ROW (left to right): I_x's bands (a) 15, (b) 50, (c) 51, (d) 60, (e) 110. FOURTH ROW: Scatter plots between gradient component I_x's band 50 and (left to right) (a) bands 15,(b) band 45, (c) band 51, (d) band 60 (e) band 110.

row of Figure 3.3.1. Visual inspection shows that in the third row band 50 in (b) is very similar to band 51 in (c) and band 60 in (d). The fourth row shows the scatter plots for I_x that correspond to the same bands as the second row. The axes of the scatter plot are equal and set to $[a_i, b_i]$, where $a_i = \min[\min(band\,50),\ \min(band\,i)]$ for $i = 15,\ 45,\ 51,\ 60,\ 110$ and b_i is defined the same using max instead of min. As with the original image, there is a linear correlation between nearby bands and that relation does not hold for bands that are farther away. Similar results can be obtained using I_y (not shown).

Figure 3.3.1 shows that even though ∇I_σ does not provide direct information on the change of the intensity along the spectral dimension, it preserves the correlation between bands. This result is not surprising since spatial derivatives are implemented as derivatives of a Gaussian which are transformations that preserve correlation. The fusion step must weight these local correlations by taking the values of the neighbor spectra into account. So, pixels with similar responses will have similar weights and preserve the differences with other non-similar pixels. An operator that preserves these kinds of local relationships is the heat operator [93]. To use the heat operator, the smoothed gradient is modeled as the initial heat distribution on a compact manifold $\mathbf{M}$ in which each point x in $\mathbf{M}$ is an $m-$dimensional vector.

The next section will present a short overview of the mathematical background for the heat operator and Section 3.3.3 presents the proposed weights.

3.3.2 *The Heat Operator*

The heat operator is used as part of the heat kernel which in turn is used as part of a structure preserving map that finds intrinsic symmetries between shapes [94]. Here it is used as the operator that acts on the spectral signature of each pixel

of the gradient components producing the weights. This section presents a brief introduction to it.

Let $\mathbf{M}$ be a compact Riemannian manifold without boundary. Let $\mathbb{R}^m_+$ be the positive section of the $m-$dimensional Euclidean space, $\mathbb{R}^m$. Let the amount of heat at a point $p \in \mathbf{M}$ at time s for some initial heat distribution $f : \mathbf{M} \to \mathbb{R}^m_+$ at $s = 0$ be defined as $h(p,s) : \mathbf{M} \times \mathbb{R}_+ \to \mathbb{R}^m_+$, where $h(p,s)$ is a solution to the heat equation

$$\partial_s h = -\Delta_{\mathbf{M}} h, \quad \text{and} \quad \lim_{s \to 0} h(p,s) = f(p),$$

and $\Delta_{\mathbf{M}}$is the Laplace-Beltrami operator of $\mathbf{M}$, for an introduction of this operator see [95]. The solution of the heat equation at time s given f, can be computed through the heat operator $H_s : L^2 \to L^2$, where L^2 is the space of all smooth, square integrable functions on $\mathbf{M}$:

$$h(p,s) = (H_s f)(p) = \left(e^{-s\Delta_{\mathbf{M}}} f\right)(p) \tag{3.3.1}$$

These results can be found in [93]. Information about properties and applications of (3.3.1) can be found in [93, 96, 94].

3.3.3 *Weights Based on the Heat Operator*

This section presents the proposed weighting process applied to the classical structure tensor. From Section 3.3.2, $\mathbf{M}$ can be thought of as an abstract space generalization that locally behaves as the Euclidean space and where $\nabla \mathbf{I}_\sigma$ resides. Let p

be a pixel on the smoothed original image $\mathbf{I}_\sigma$. The weights are given by $h(p,s)$ in (3.3.1) with initial heat distribution

$$f(p) = \nabla \mathbf{I}^*_\sigma(p) = \begin{bmatrix} \mathbf{I}^*_x(p) \\ \mathbf{I}^*_y(p) \end{bmatrix}$$

where $\mathbf{I}^*_x(p)$ is $\mathbf{I}_x(p)$ after a convolution with a $[1 \times 3]$ mean filter used to regularize the weighting process. $\mathbf{I}^*_y(p)$ is obtained in the same manner. The weights at p denoted as W_p will be called *heat weights*. Their definition is based on H_s in (3.3.1), but the absolute value of the Laplace-Beltrami operator is used to capture the size of variations independent of their sign. W_p is given as:

$$W_p = \begin{bmatrix} w_{px} \\ w_{py} \end{bmatrix} = \begin{bmatrix} e^{-s|\Delta \mathbf{I}^*_x(p)|} \\ e^{-s|\Delta \mathbf{I}^*_y(p)|} \end{bmatrix}. \tag{3.3.2}$$

Since $\mathbf{I}^*_x(p)$ is an $m-$dimensional vector, its Laplace-Beltrami operator is its second derivative. The same is true for $\mathbf{I}^*_y(p)$. w_{px}, $w_{py} \in \mathbb{R}^m$ are normalized by their respective sums so that $sum(W_p) = [1,1]^T$. In all experiments, $s = 1$ was used. Higher values of s can dominate W_p's behavior, corrupting the information produced by the Laplace-Beltrami operator (see Section 5.3.). This weighting process takes into account the correlation between bands in two ways: (i) using the mean filter and (ii) approximating the Laplace-Beltrami operator using the standard central difference scheme. The spectrally weighted structure tensor with weights based on the heat operator will be called the *heat weighted structure tensor*.

Figure 3.3.2 shows how the weights affect the spectral signatures of the gradient component I_x and how the relations between pixels of the same class are preserved in a HSI. There are two scenarios depicted in Figure 3.3.2 both with pixels in the same class, as shown in (b) and (f). The first scenario, depicted in the left column,

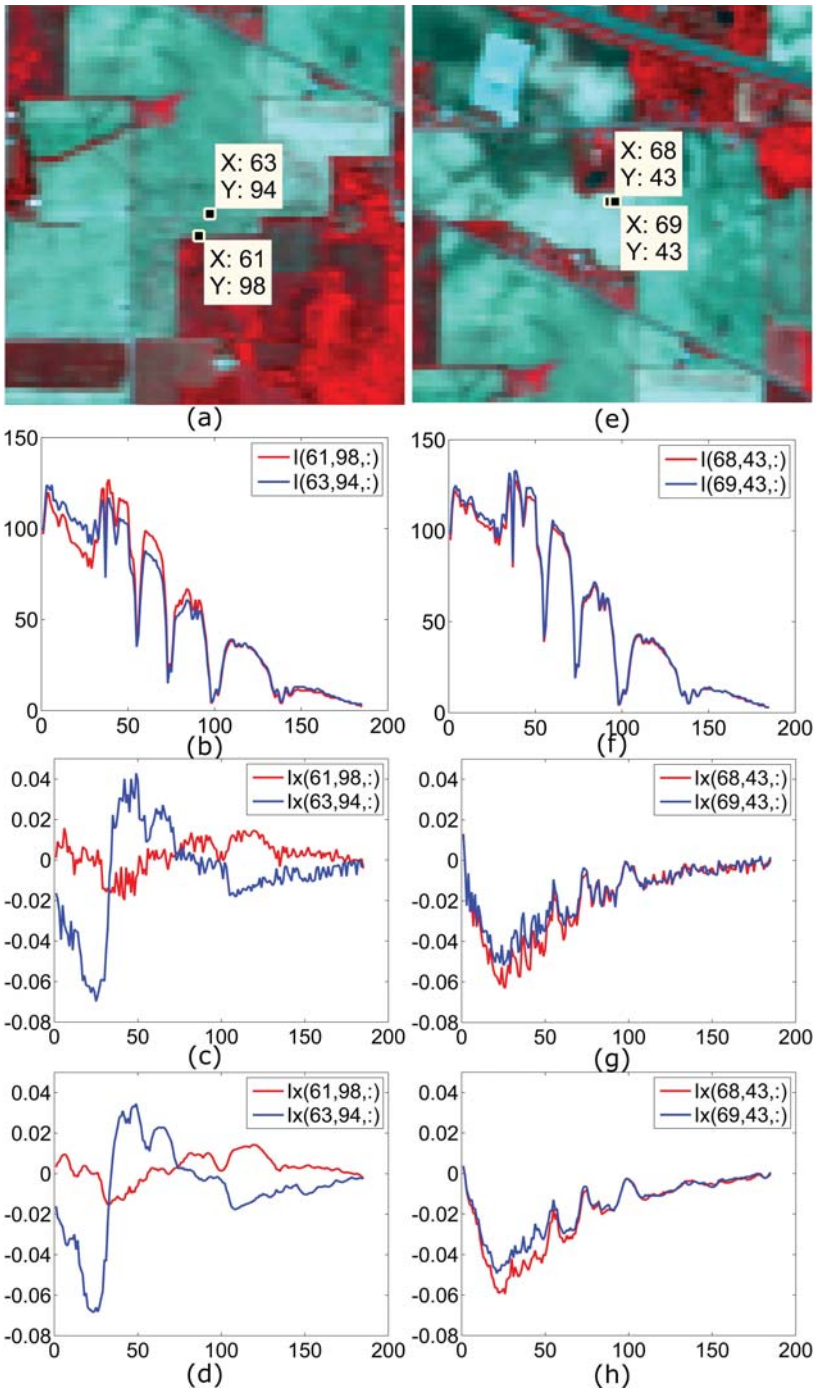

Figure 3.3.2: Weighting effect in pixels on heterogeneous and homogeneous regions of Indian Pines. TOP ROW left to right: (a) Portion of Indian Pines image with a pixel on an edge and a second pixel on an homogeneous region. (b) Spectra of the pixels marked in (a). (c) I_x's signature of the pixels marked in (a) using the uniform weights (d) I_x's initial signature of the pixels marked in (a) using the heat weights. BOTTOM ROW left to right: (e) Portion of Indian Pines image with both pixels in a homogeneous region. (f) Spectra of the pixels marked in (e). (g) I_x's signature of the pixels marked in (e) using the uniform weights (h) I_x's signature of the pixels marked in (e) using the heat weights.

is the comparison of an edge pixel (red) and a pixel from the inside region (blue). The second scenario, shown in the right column, compares two pixels inside region. Figure 3(c) and (d) show I_x's signature from pixels in the same spatial position as in (a). As in (b), the signature of the edge pixel is plotted in red and the non-edge in blue. It can be seen in (c) and (d) that the two signatures are very different as not all spatial neighbors of the edge pixel belong to the same class. In addition, the signature of the edge pixel tends to zero along the spectra. This is expected since it is a horizontal edge and the signature belongs to its horizontal derivative I_x. On the other hand, when the two pixels are on the inside region, their signatures are highly similar, as shown in Figures 3.3.2 (f)–(h). Figures 3.3.2 (c) and (g) show the signature of the pixels on I_x using the uniform weights on both scenarios. Figures 3.3.2 (d) and (h) show the signatures of pixels on I_x using the heat weights. For this example, a comparison of the angle between the signatures produced with the uniform weights, θ_u, and the angle between the signatures produced with the heat weights, θ_h, is done using the relative difference measure given by:

$$\%\,\mathrm{Dif} = \frac{|\theta_h - \theta_u|}{\max\left(|\theta_h|\,, |\theta_u|\right)} \times 100.$$

In the first case, Figures 3.3.2 (c) and (d), θ_h in (d) is 0.47% larger than θ_u in (c). In the second case, Figures 3.3.2 (g) and (h), θ_h in (h) is 4.15% smaller than θ_u in (g). These results suggest that the heat weights, not only produce smoothing, but they also reduce the variability between similar signatures, while maintaining the difference between dissimilar pixels.

From Figure 3.3.2 it is clear that the proposed weighting process produce a smoothing in the spectral response of each pixel of the spatial gradient images. Intuitively the heat operator penalize large abrupt changes in the spectral response of the pixel. In the case of a pixel $\mathbf{p} = [p_1, ..., p_m]$ on I_x^*, the $x-$component of the

spatial gradient, the entry p_i for band i identify the existence or nonexistence of vertical spatial edges in that band. The heat operator applied to $\mathbf{p}$ will penalize large changes in the rate of change of the existence of the spatial edges along the bands at pixel $\mathbf{p}$.

3.3.4 *Parameters used for the Structure Tensor.*

Two parameters for the structure tensor, J_ρ, with $\rho > 0$ were defined in Section 2.3.5. One is known as the noise scale σ, since the smoothed gradient ∇I_σ ,which is an edge detector, ignores details smaller than $O(\sigma)$ [20]. The parameter σ is image dependent, but our experience with several HSIs is that a value of $\sigma \in [0.5, 1.5]$ is enough to regularize the process. For degraded images, larger values of σ are needed [46]. Figure 3.3.3(a) shows the red band of Van Gogh's painting "Road with Cypress and Star" [97]. Figures 3.3.3(b) and (c) are the original image after being processed with TAND-CED (see Section 4.2.2) using the classical structure tensor. To show the impact of the integration scale ρ parameter, all values are kept equal except ρ. When $\rho = 1$ in Figure 3.3.3(b), the filter produces artifacts, while increasing it to $\rho = 4$ in Figure 3.3.3(c) produces flow-like structures that look like completed lines since this larger neighborhood captures the coherence orientation. Increasing it to $\rho = 6$, see Figure 3.3.3(d), produces very similar results to $\rho = 4$. It is important to note that underestimation of ρ is far more critical than overestimation, so it is not difficult to find an estimate that works for the whole image domain [46].

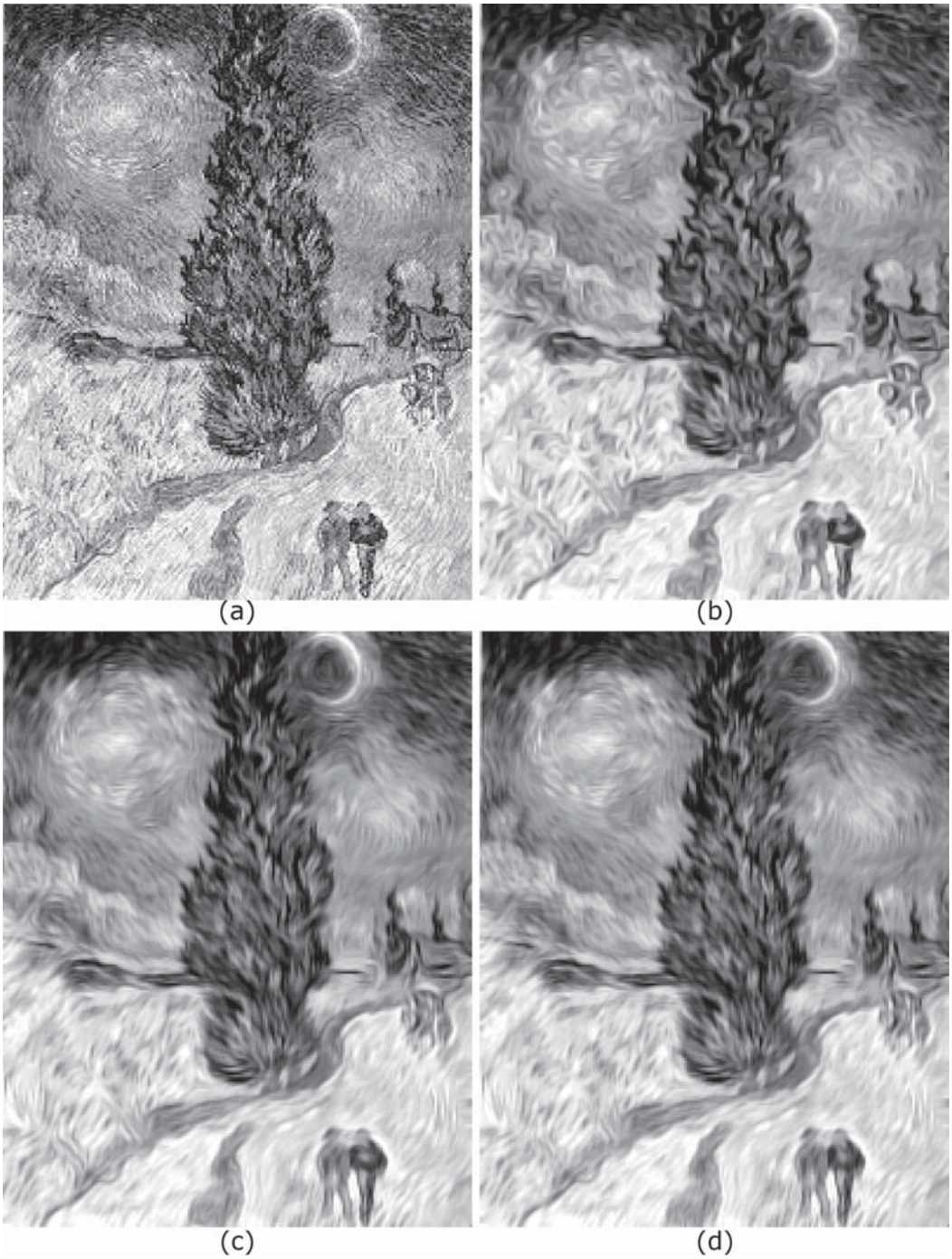

Figure 3.3.3: Integration scale on CED. $\sigma = 0.5$, $n = 4$. Image size is 254×200. (a) original image. (b) After CED with $\rho = 1$. (c) After CED with $\rho = 4$. (d) After CED with $\rho = 6$.

3.4 IMPLEMENTATION DETAILS

The Gaussian derivatives for an $n_1 \times n_2 \times m$ image $\mathbf{I}$ are implemented by truncating the uni-variate Gaussian distribution with variance σ^2 when the smallest values is less than 1×10^{-4}. This produces a kernel G_σ. Then the derivatives $\mathbf{I}_x$, $\mathbf{I}_y$ are implemented using separable kernels. Note that $\mathbf{I}_x$ and $\mathbf{I}_y$ are of the same size as $\mathbf{I}$.

The weights are implemented by converting $\mathbf{I}_x$ and $\mathbf{I}_y$ to matrices in which each row is a pixel. Each of those matrices are convolved with the 1×3 median filter and with the second derivative calculated using symmetric central differences. Then (3.3.2) is calculated and converted back to a cube, producing W_x and W_y. The weighted derivatives are calculated by Hadamard multiplications $(W_x \circ \mathbf{I}_x)$ and $\left(W_y \circ \mathbf{I}_y\right)$, denoted by the small circle $(\circ)$.

Since the derivative is calculated using local neighborhoods all the products used to calculate the components

$$A = \sum_i^m \left[(W_x \circ \mathbf{I}_x) \circ (W_x \circ \mathbf{I}_x)\right]_i, \; B = \sum_i^m (W_x \circ \mathbf{I}_x)_i \circ \left(W_y \circ \mathbf{I}_y\right)_i \text{ and}$$

$$C = \sum_i^m \left(W_y \circ \mathbf{I}_y\right)_i \circ \left(W_y \circ \mathbf{I}_y\right)_i$$

of the initial matrix field are Hadamard too. Each of those components produce a $n_1 \times n_2$ image with the edge information. Vertical edges are found using A. Diagonal edges are found using B and horizontal edges using C. Those A, B and C are used for EED. A, B and C are convolved with G_ρ a Gaussian of variance ρ to calculate the integration step needed for CED's structure tensor. G_ρ is truncated in the same way as G_σ. It is important to note that all the operations needed to implement the structure tensor are pixel-wise, so a parallel implementation is possible. Figure 3.4.1 shows a flow chart of the steps need it to calculate the structure tensor. The

pseudo-code to implement the proposed weighted structure tensor is described in 3.1.

Algorithm 3.1 Weighted Structure Tensor

Inputs : $\mathbf{I}$, σ, ρ
Output: A, B, C

1. Calculate the Gaussian Kernel
   - $thr \leftarrow 0.0001$; $t \leftarrow 1:30$
   - $width := find(\exp\left(-(t.^2)/2\sigma^2\right) > thr; x \leftarrow -width : width$
   - $G_\sigma \leftarrow \exp\left(-(x \circ x)/2\sigma^2\right)$
   - $G_\sigma \leftarrow G_\sigma / sum(G_\sigma)$
2. Calculate $\mathbf{I}_x$, $\mathbf{I}_y$
   - $\mathbf{I}_x \leftarrow (\mathbf{I} * (-x \circ G_\sigma)) * G_\sigma^T$
   - $\mathbf{I}_y \leftarrow \left(\mathbf{I} * (-x \circ G_\sigma)^T\right) * G_\sigma$
3. Calculate $(W_x \circ \mathbf{I}_x)$ and $\left(W_y \circ \mathbf{I}_y\right)$ using (3.3.2).
4. Calculate $A = \sum_i^m \left[(W_x \circ \mathbf{I}_x) \circ (W_x \circ \mathbf{I}_x)\right]_i$
5. Calculate $B = \sum_i^m \left(W_x \circ \mathbf{I}_x\right)_i \circ \left(W_y \circ \mathbf{I}_y\right)_i$
6. Calculate $C = \sum_i^m \left[\left(W_y \circ \mathbf{I}_y\right) \circ \left(W_y \circ \mathbf{I}_y\right)\right]_i$
7. if $\rho \neq 0$ % for CED
8. $A \leftarrow A * G_\rho$
9. $B \leftarrow B * G_\rho$
10. $C \leftarrow C * G_\rho$
11. endIf

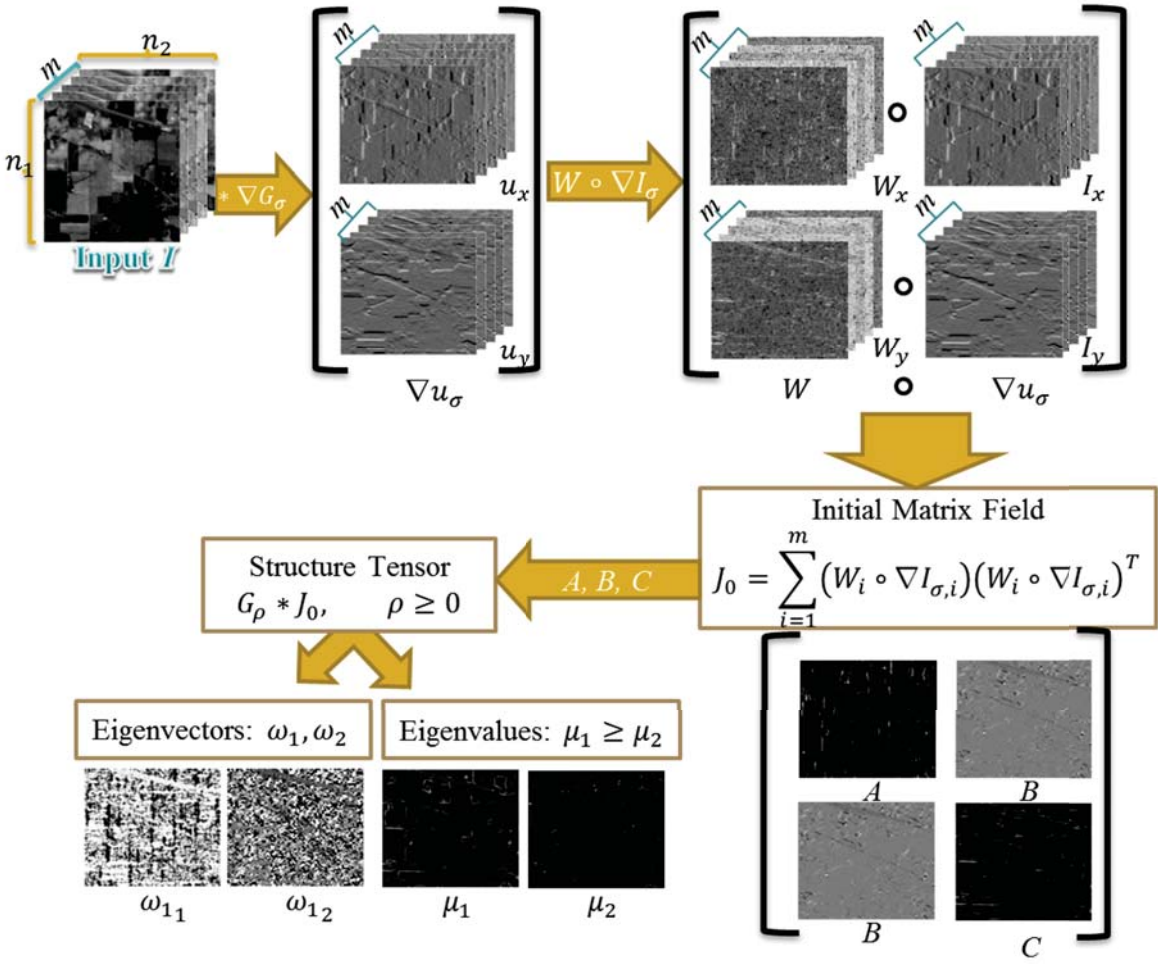

Figure 3.4.1: Flow chart describing the proposed Structure Tensor.

3.5 CONCLUDING REMARKS

In this chapter a framework for the spectrally adapted structure tensor for vector-valued images has been presented. This framework consist in a weighting process that takes into account the spectral characteristics of those images. A spectrally adapted structure tensor for the case in which the spectrum of the vector-valued images is locally highly correlated have been developed. HSI are an example of those images. The heat operator give the mathematical framework to use the spectral characteristics of the images for the weighting process.

4

TENSOR ANISOTROPIC NONLINEAR DIFFUSION

This chapter will describe the proposed TAND diffusion tensors. Those tensors are build based on the structure tensor defined in Section 2.5.3. Extending TAND to vector-valued images that are locally highly correlated in the spectral direction is the main motivation to propose them. The diffusion tensors used in the proposed TAND filters will use the eigenvectors of the spectrally adaptive structure tensor proposed in the previous chapter. The eigenvalues will be defined so TAND will take advantage of the edge information obtained by the heat weighted structure tensor. Experimental results on the usage of the heat weighted structure tensor on TAND and comparison of the proposed TAND with the state of the art proposed by Weickert will be presented.

4.1 INTRODUCTION

TAND is described by the partial differential equation (PDE):

$$\frac{\partial u}{\partial t} - \nabla \cdot (D \nabla u) = 0, \qquad in\ \Omega \tag{4.1.1}$$

where u (the filtered version of image $f(x)$ with domain Ω) is the solution of the initial boundary value problem for the diffusion equation with f as the initial con-

dition, and Neumann boundary conditions. D is a positive semi definite symmetric matrix known as the *diffusion tensor* D is defined as:

$$D = \begin{bmatrix} \omega_1 & \omega_2 \end{bmatrix} \begin{bmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} \lambda & \beta \\ \beta & \nu \end{bmatrix} \tag{4.1.2}$$

where, $\omega_1 = [\omega_{1_1}, \omega_{1_2}]^T \perp \omega_2 = [-\omega_{1_2}, \omega_{1_1}]^T$ are defined as in (3.1.5). κ_1 and κ_2 are functions on the largest and smallest eigenvalues of J_ρ, μ_1 and μ_2 respectively. λ, β and ν can be written in terms of the eigenvalues and eigenvectors of D as follows:

$$\lambda = \kappa_1 \omega_{1_1}^2 + \kappa_2 \omega_{1_2}^2 \tag{4.1.3}$$

$$\beta = (\kappa_1 - \kappa_2)\, \omega_{1_1} \omega_{1_2} \tag{4.1.4}$$

$$\nu = \kappa_1 \omega_{1_2}^2 + \kappa_2 \omega_{1_1}^2, \tag{4.1.5}$$

when, $\rho = 0$, (3.1.3) is the structure tensor used for EED and when $\rho > 0$, (3.1.4) is the structure tensor used for CED. J_0 defined in (3.1.3) is used for the classical structure tensor while J_0 defined in (3.2.1) is used for the spectrally weighted tensor. The next section will present the functions used to define the eigenvalues κ_1 and κ_2 of D for the proposed TAND.

4.2 EIGENVALUES OF THE PROPOSED DIFFUSION TENSORS

This section will give some motivation to define new diffusion tensors for TAND-EED and TAND-CED that take advantage of the information produced by the heat weighted structure tensor. A discussion on why the eigenvalues where defined using those functions is provided.

4.2.1 *EED diffusion tensor*

For EED, the diffusion is directed along the edges, avoiding diffusion across them and on them. So, both κ_1 and κ_2 must get close to zero in the presence of an edge and be maximal in homogeneous regions. There are several functions proposed in the literature used for the eigenvalues κ_1 and κ_2. A review of them can be found in [45]. The proposed κ_1 and κ_2 are given by :

$$\kappa_1 = \begin{cases} 1 & \mu_1 = 0 \\ 1 - \exp\left(\frac{-C_r}{(\mu_1/\psi)^r}\right) & \mu_1 > 0 \end{cases} \tag{4.2.1}$$

$$\kappa_2 = \begin{cases} 1 & \mu_1 \leq \psi \\ \frac{1}{\xi\left(\kappa_1^1\right)^4+1} & \mu_1 > \psi \end{cases} \tag{4.2.2}$$

κ_1 is defined as the one proposed for EED by Weickert in [45] with $r = 4$, $C_r = 3.315$ (see [2] §5.1), where $1 = \exp(-C_r)(1 + 2C_r \cdot r)$ and ψ is a contrast parameter. For the edges $(\mu_1 > \psi)$, κ_2 is a constant defined in terms of the entropy of κ_1 at iteration $n = 1$, $\xi\left(\kappa_1^1\right)$ and 1 otherwise. κ_1^1 is κ_1 calculated using the original image. The entropy $\xi\left(\kappa_1\right)$ is defined as in [98]:

$$\xi = -\sum_{i=1}^{255} q_i \log\left(q_i\right)$$

where q_i is the probability of that a pixel, belonging to κ_1, has intensity i and $q_i \log\left(q_i\right) = 0$ if $q_i = 0$, . In practice, q_i is the histogram count of the i^{th} bin of κ_1's global intensity histogram. Weickert's in [45] uses κ_2 equal to 1. The problem of letting $\kappa_2 = 1$ is that it does not adapt to the structures in the image. The

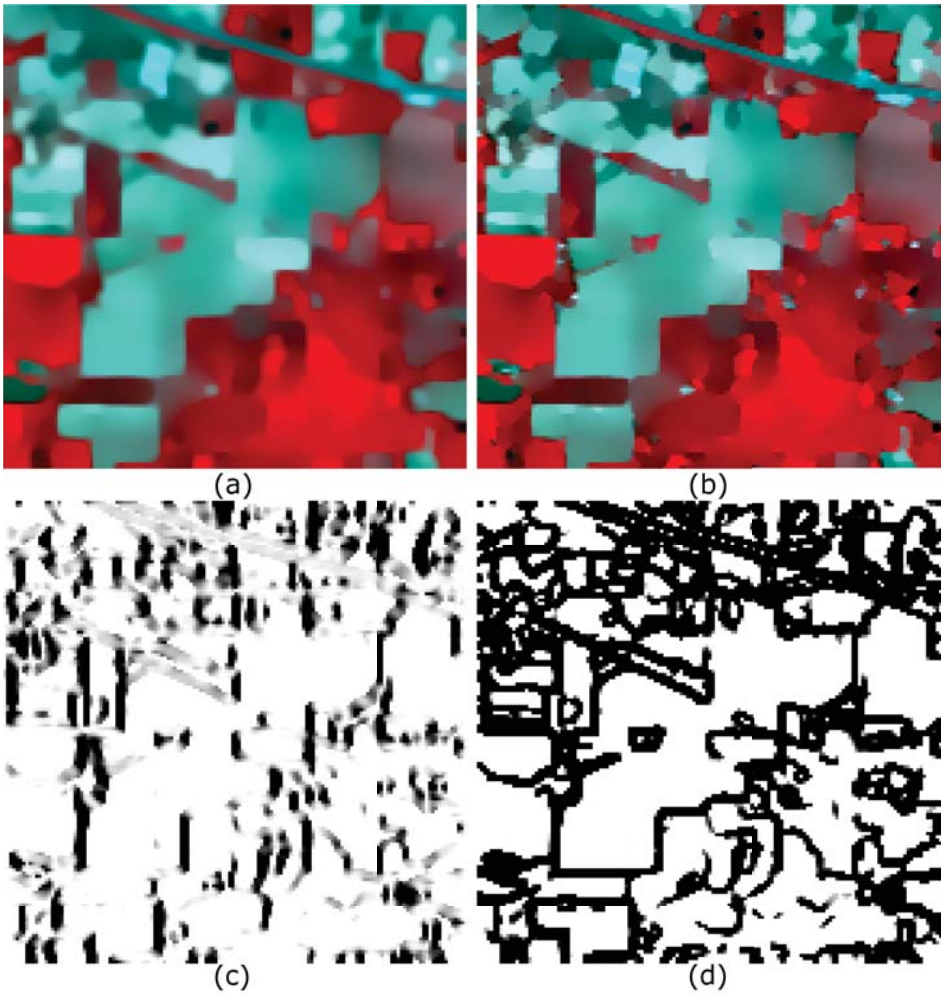

Figure 4.2.1: Comparison of the classical structure tensor using $\kappa_2 = 1$ and κ_2 defined as in (4.2.2). (a) RGB composition of HSI Indian Pines after EED using classical structure tensor with $\kappa_2 = 1$, $n = 4$ iterations; (b) same as (a) with $\kappa_2 = 1/\left[\xi\left(\kappa_1^1\right)^4 + 1\right]$, $n = 4$ for edges, otherwise 1; (c) $\lambda-$component of the diffusion tensor in (a); (d) $\lambda-$component of the diffusion tensor in (b).

same smoothing is applied in the direction of ω_2 independently of the structures present in the image. $\kappa_2 = 1$ produces a strong smoothing in the edges as seen in Figure 4.2.1(a). The value of κ_2 in (4.2.2) will decrease in the presence of an edge ($\mu_1 > \psi$), in which case $\kappa_1 \approx 0$ and $\kappa_2 \approx 1/\left[\xi\left(\kappa_1^1\right)^4 + 1\right]$. κ_2 is maximal in the interior of a region ($\mu_1 \leq \psi$), in which case $\kappa_1 \approx 1$ and $\kappa_2 \approx 1$. This definition mitigates the blurring effect in the edges, while keeping the smoothing in the homogeneous regions as seen in Figure 4.2.1(b). Figure 4.2.1(c) shows the $\lambda-$component of the diffusion tensor based on the classical structure tensor with $\kappa_2 = 1$. Clearly this component only preserves the vertical edges (dark edges representing intensity values close to zero). There is a strong smoothing in all the other edges (light gray edges). While, in Figure 4.2.1(d), where κ_2 is defined as in (4.2.2), edges are preserved. κ_2 needs to have the same value for all edges to prevent the same edge from having strong smoothing in some places and lesser in others. So, κ_2 was chosen to have a dependency on a global characteristic of κ_1 that does not depend directly on the intensity scale. So, the entropy of κ_1 was a natural choice. The entropy is calculated using the original image since the entropy of the filtered image u decreases at each iteration, i.e., $\xi\left(u^{n+1}\right) \leq \xi\left(u^n\right)$ [46]. So, $\xi\left(\kappa_1^1\right)$ is constant for all iterations n.

4.2.2 *CED Diffusion tensor*

When $\rho > 0$, the structure tensor J_ρ is used to enhance one dimensional features in the image. In this case, the eigenvalue κ_1 is equal to a very small constant α used to guarantee that the diffusion process never stops and to keep the diffusion tensor uniformly positive definite [46]. κ_2 will depend on the difference in the eigenvalues of J_ρ, i.e., $\mu = \mu_1 - \mu_2$. The eigenvalues are defined as in [20]

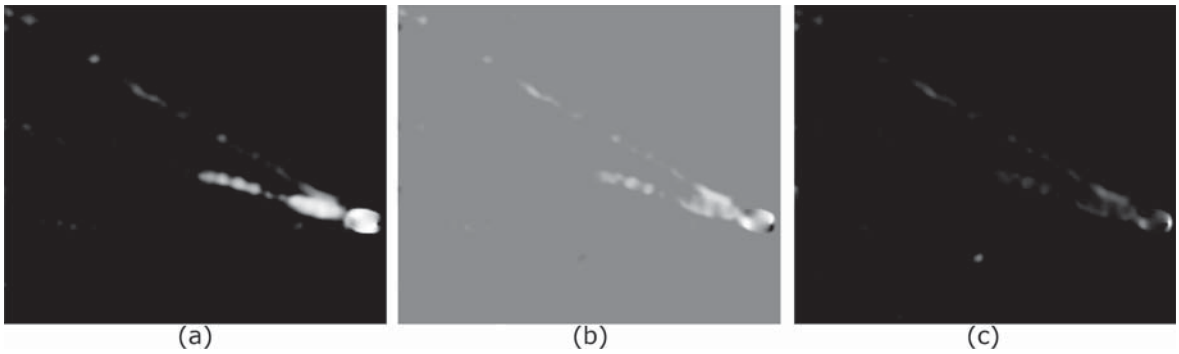

Figure 4.2.2: κ_2 using (4.2.3). Components of the diffusion tensor using $C = 1$ in (4.2.3) after $n = 1$ iterations.

$$\kappa_1 = \alpha \qquad 0 < \alpha \ll 1$$

$$\kappa_2 = \begin{cases} \alpha & \mu = 0 \\ \alpha + (1-\alpha)\exp\left(\frac{-C}{\mu^2}\right) & \textit{otherwise} \end{cases} \tag{4.2.3}$$

$C = 1$ is used. This constant assignment does not work well. Figure 4.2.2 shows the λ, β, and ν components of the diffusion tensor (see (4.1.2)) using κ_2 as defined in (4.2.3) with $C = 1$ of an AVIRIS image [99]. In this image, only part of the center wake is enhanced. The idea of using TAND-CED in this image is to enhance the wakes by eliminating the radial texture traversing them. If we let C be a variable then C works as a contrast parameter of μ^2 and not of μ as shown in Figure 4.2.4(b). This can presents problems for images where the majority of the edges responses in μ are less than one. So it will be better to define κ_2 in term of the edges responses of μ and not on μ^2.

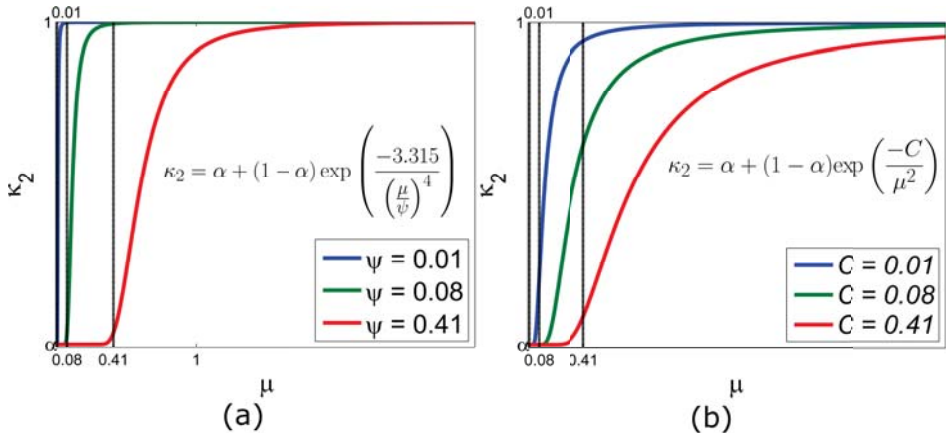

Figure 4.2.3: Comparing κ_2 using (4.2.3) and using (4.2.4). (a) Plot of κ_2 in (4.2.4), varying the value of ψ. (b) Plot of κ_2 in (4.2.3), varying the value of C.

The eigenvalues of the diffusion tensor can also be defined using the following expression [100] with ψ as in (4.2.1):

$$\begin{aligned} \kappa_1 &= \alpha && 0 < \alpha \ll 1 \\ \kappa_2 &= \begin{cases} \alpha & \mu = 0 \\ \alpha + (1-\alpha)\exp\left(\frac{-3.315}{(\mu/\psi)^4}\right) & \textit{otherwise.} \end{cases} \end{aligned} \tag{4.2.4}$$

The value of the contrast parameter ψ in (4.2.4) defines low contrast regions as regions where $\mu < \psi$ making $\kappa_2 \approx \alpha$, and high contrast regions where $\mu > \psi$ making $\kappa_2 \approx 1$ as shown in Figure 4.2.3(a). The threshold parameter ψ in (4.2.4) is a contrast parameter of μ instead of μ^2. Figure 4.2.4 shows the AVIRIS image [99], see Section 5.1.2, processed with heat weighted structure tensor with $n = 2$, $\sigma = 1.2$ $\rho = 2.5$. That figure shows the components of the diffusion tensor, λ, β, and ν using (4.2.4) as eigenvalue function for κ_2 and its sensibility to changes in the value of ψ for $\psi = 70^{th}$, 92^{th} and 98^{th} percentile of μ. It is clear that varying ψ produces enhancement in different features of the image. If C in (4.2.3) is taken as a contrast parameter and the majority of the values of μ are greater than 1 then (4.2.3) and (4.2.4) can produce very similar results. For all CED experiment presented in this

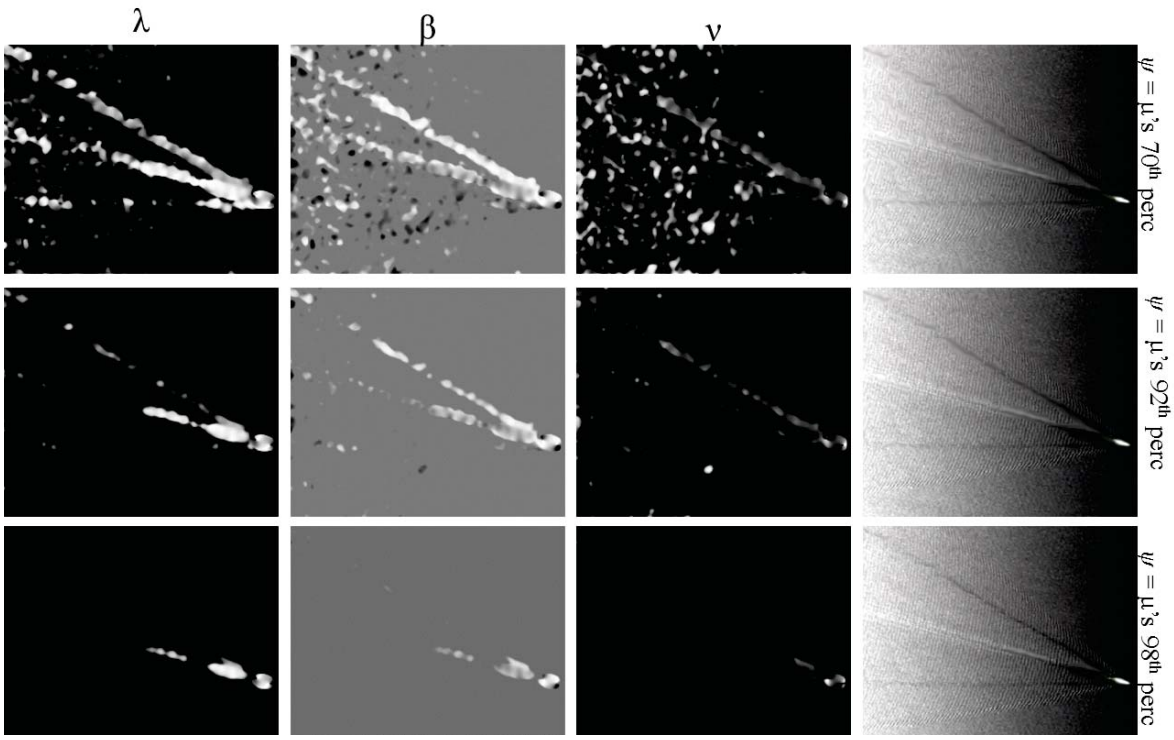

Figure 4.2.4: Proposed diffusion tensor for CED and its sensibility to ψ. AVIRIS image [99] was process with heat weighted structure tensor with $n = 2$, $\sigma = 1.2$ $\rho = 2.5$. Components of the diffusion tensor, λ, β, and ν using (4.2.4) as eigenvalue function for κ_2 vs. ψ are shown. $\psi = 70^{th}$, 92^{th} and 98^{th} percentile of μ.

work the eigenvalues of the diffusion tensor will be defined as in (4.2.4), otherwise will be specified. For CED, the value of the noise scale σ needs to be set small enough so the orientation information is not destroyed.

4.3 TAND AS AN APPROXIMATION TO LOCAL DIFFUSION

One of the most studied diffusion PDE-based methods is TAND [21, 2, 20, 46]. TAND was proposed by Weickert in [71]. This filter belongs to a family of regularized anisotropic nonlinear filters that have many proved properties such as well-posedness (existence and uniqueness of results), continuous dependence of the solution on the initial image and scale space properties such as maximum–minimum

principles, invariance, Lyapunov functionals, and convergence to a constant steady-state. In addition, a scale space discrete theory has been developed [2].

Tschumperlé and Derriche in [18] found that the divergence operator in TAND can be decomposed as a trace-base operator on the Hessian, **H**, plus a term that appears connected to the spatial variation of the tensor field produced by D, i.e.,

$$\operatorname{div}(D\nabla I_i) = \operatorname{trace}(DH_i) + \nabla I_i^T \overrightarrow{div}(D) \qquad i = 1, \ldots m \tag{4.3.1}$$

where

$$\overrightarrow{div}(D) = \begin{pmatrix} \operatorname{div}\left[\begin{pmatrix} \lambda & \beta \end{pmatrix}^T\right] \\ \operatorname{div}\left[\begin{pmatrix} \beta & \nu \end{pmatrix}^T\right] \end{pmatrix}$$

and λ, β and ν are the components of D, $i = 1, \ldots, m$. This is because they see D in (4.1.1) as a global operator (i.e., a tensor field on Ω), and not a local operator at a neighborhood of a pixel $p \in \Omega$ as in (4.4.1). So, λ, β and ν are no longer constants at p, they are now functions from $\Omega \to \mathbb{R}$. Then, it is concluded in [12] that the first term in (4.3.1) corresponds to local smoothing directed by the elements of D and that the second term perturbs the smoothing behavior steered by D in the first term. In this work, the second term in the right hand side of (4.3.1) will be called *the disturbance term*. They express (4.3.1) as a sum of two trace terms. To show all these calculations, they use the initial matrix in (3.1.3) ([12]§3.2-3.3). For the general framework ([12]§3.4) based on a generalization of the operator trace(DH_i), they choose to use the structure tensor in (3.1.4).

All the examples presented in this framework [12, 3] were compared with TAND–CED. This comparison appears to be a natural choice since this method uses an integration parameter $\rho > 0$, just as in TAND–CED. However, by observing how the eigenvalues of both methods are selected, it becomes clear that, for their diffusion tensor, they are selected to do very different processes. Hence, a comparison of the two

may not be appropriate. Perhaps a better comparison with Weickert's TAND–EED. This is also supported in the derivation of their unifying framework ([12] §3.3) since they use the initial matrix field there. In addition, TAND–EED is designed with the same purpose as the trace–based formulation. Both aim to preserve and produce small enhancement on the edges with strong smoothing of the homogeneous areas. CED, on the other hand, aims to enhance the edges by completing them with almost no smoothing of the homogeneous regions. To briefly explain this, both κ_1 and κ_2, in the trace-based formulation, use the decreasing functions given by [12]:

$$\kappa_1 = \frac{1}{1 + (\sqrt{\mu_1 + \mu_2})^2}, \; \kappa_1 = \frac{1}{\sqrt{1 + (\sqrt{\mu_1 + \mu_2})^2}} \tag{4.3.2}$$

In the presence of a homogeneous region, $\mu_1 \approx \mu_2 \approx 0$, $\kappa_1 \approx \kappa_2 \approx 1$ produce a strong smoothing. This is true for κ_1 and κ_2 in (4.3.2) for the trace based formulation and also for κ_1 in (4.2.1) and $\kappa_2 = 1$ for Weickert's TAND–EED. For TAND–CED, $\kappa_1 \approx \kappa_2 \approx \alpha = 1 \times 10^{-3}$, produces almost no smoothing. In the case of edges, $\mu_1 \gg \mu_2 \approx 0$, κ_1 and κ_2 in (4.3.2) vanish to zero. κ_2 vanishes slower than κ_1 , producing a very small enhancement along the edges. κ_1 in (4.2.1) for TAND–EED behaves similarly to κ_1 in (4.3.2) and produces a small enhancement for edges with intensity greater than ψ, and $\kappa_2 = 1$. In TAND–CED, κ_1 is a small constant, $\alpha = 1 \times 10^{-3}$, and κ_2 reaches its maximum value of 1 for edges with intensity greater than ψ. In general, the diffusion tensor for the trace–based operator behaves more similar to TAND for EED than CED.

Empirically speaking, in the experience with TAND processing of HSIs, the influence of the second term in the RHS of (4.3.1) is not noticeable in the sense that TAND will still enhance the features as expected, (see Section 5.5.1.2). Hence, the experiments performed indicate that the computational cost justifies the orientation and magnitude inaccuracies introduced by this term, if any is really introduced.

However, further study comparing TAND–EED and the trace-based formulation is needed as well as a study of the properties of this general trace based formulation to see if the properties of TAND are extensible to this more general approach. Neither are addressed in this thesis.

Computationally speaking, to calculate the trace-based operator, it is necessary to calculate the gradient and Hessian at each pixel entry. There are two ways to approximate the Hessian: (i) using a finite difference approach or (ii) using spatially varying normalized Gaussian kernels. The approximation that gives results comparable with TAND–EED using semi-implicit methods is the one using Gaussian kernels. This approximation has a high computational cost since Gaussian kernels need to be calculated at each pixel entry and then a local convolution is performed. But it is accurate and this scheme guarantees that the trace–based formulation preserves the maximum principle. The heat weighted structure tensor can also be used in the trace-based formulation. TAND was chosen since two kind of processing can be done with it, regularization and structure enhancement, and given that HSI are large data sets, a tradeoff between accuracy and computational cost is also necessary.

4.4 DISCRETIZATION OF THE DIVERGENCE-BASED DIFFUSION EQUATION USING A TENSOR

The anisotropic diffusion filter in (4.1.1) can be written as

$$\begin{aligned}\frac{\partial u}{\partial t} &= \operatorname{div}\left(D\nabla u\right)\\ \frac{\partial u}{\partial t}\Big|_{\mathcal{N}(p)} &= \operatorname{div}\begin{bmatrix}\lambda\partial_x u + \beta\partial_y u\\ \beta\partial_y u + \nu\partial_y u\end{bmatrix}\\ &= \partial_x\left(\lambda\partial_x u\right) + \partial_x\left(\beta\partial_y u\right) + \partial_y\left(\beta\partial_y u\right) + \partial_y\left(\nu\partial_y u\right)\end{aligned} \tag{4.4.1}$$

where $\mathcal{N}(p)$ is a neighborhood around pixel p. Equation (4.4.1) cannot be solved analytically. It needs to be approximated by numerical methods. Finite difference methods are most commonly used, see (Section 2.6). They are easy to implement and they work well with a regular rectangular pixel grid, such as those present in digital images (see [101]).

For images, it is customary to assume that the pixel size in the x and y direction is 1. Let τ be the time step size, and denote $U_{i,j}^n$ as the approximation of u at pixel $p = (i, j)$ at time $t_n = n\tau$. A simple finite difference approximation to $\partial_t u$ at pixel p for iteration n is given by:

$$\partial_t u = \frac{U_p^{n+1} - U_p^n}{\tau} \tag{4.4.2}$$

To include the neighbors in the 3×3 neighborhood and to produce a symmetric derivative, the spatial derivatives are approximated as follows:

$$\delta_x^* U_{i,j} = U_{i+\frac{1}{2},j}^n - U_{i-\frac{1}{2},j}^n, \text{ and } \delta_y^* U_{i,j} = U_{i,j+\frac{1}{2}}^n - U_{i,j-\frac{1}{2}}^n, \tag{4.4.3}$$

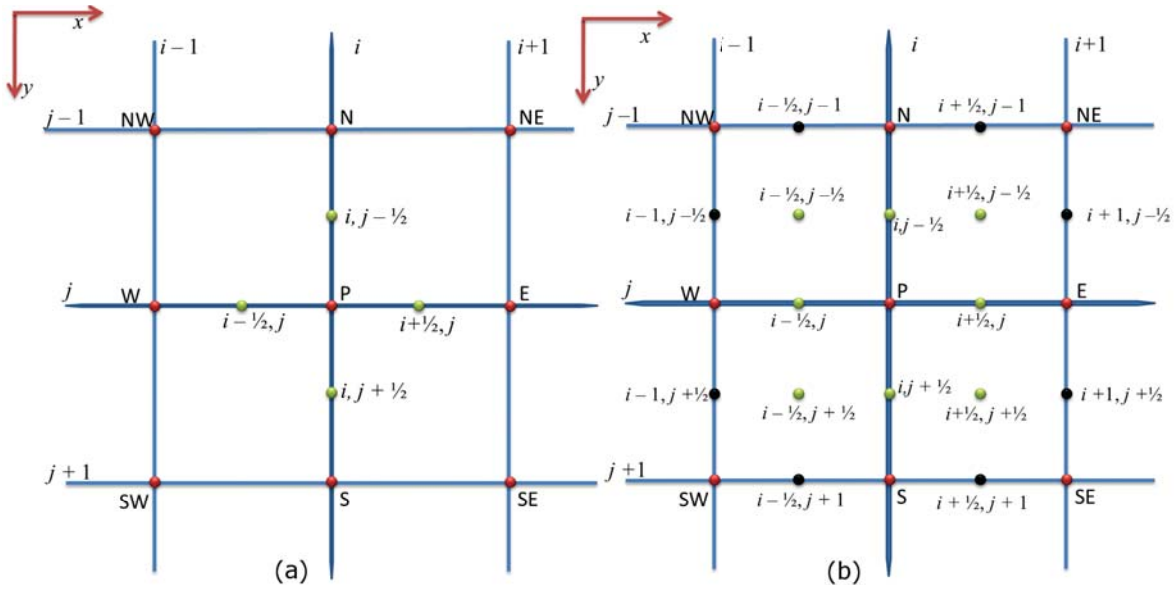

Figure 4.4.1: Pixels involved in the approximation of the divergence term for the classical and rotational invariant difference schemes

which corresponds to the green dots in Figure 4.4.1(a). Then, the right hand side of (4.4.1) is approximated as:

$$\begin{aligned}\partial_x\left(\lambda^n\partial_x u^n\right)\big|_{i,j} \approx& \left(\lambda^n_{i,j}\delta^*_x U^n_{i,j}\right)_{i+\frac{1}{2},j} - \left(\lambda^n_{i,j}\delta^*_x U^n_{i,j}\right)_{i-\frac{1}{2},j}\\ \approx& \lambda^n_{i+\frac{1}{2},j}\left(U^n_{i+\frac{1}{2},j} - U^n_{i-\frac{1}{2},j}\right)_{i+\frac{1}{2},j} - \lambda^n_{i-\frac{1}{2},j}\left(U^n_{i+\frac{1}{2},j} - U^n_{i-\frac{1}{2},j}\right)_{i-\frac{1}{2},j}\end{aligned}$$

$$\partial_x\left(\lambda^n\partial_x u^n\right)\big|_{i,j} \approx \frac{\lambda^n_E + \lambda^n_P}{2}\left(U_E - U_P\right) + \frac{\lambda^n_P + \lambda^n_W}{2}\left(U_W - U_P\right).$$

In an analogous form it is found that $\partial_y\left(\nu\partial_y u\right)\big|_{i,j}$ given by:

$$\partial_y\left(\nu^n\partial_y u^n\right)\big|_{i,j} \approx \frac{\nu^n_S + \nu_P}{2}\left(U_S - U_P\right) + \frac{\nu^n_P + \nu^n_N}{2}\left(U_N - U_P\right).$$

For the mixed derivatives, there are several possibilities:

SCHEME 1. Using the standard central differences in (2.6.1) :

Table 4.4.1: 3×3 averaging mask $\mathbf{A}\left(U_{i,j}^{n}\right)$ for the case in which the mixed derivatives are calculated using the standard central differences defined by δ_x in 2.6.1

$\frac{\beta_N+\beta_W}{4}$	$\frac{\nu_N+\nu_P}{2}$	$\frac{\beta_E+\beta_N}{4}$
$\frac{\lambda_W+\lambda_P}{2}$	$-\begin{pmatrix} \frac{\lambda_E+\lambda_P}{2}+\frac{\lambda_W+\lambda_P}{2} \\ \\ \frac{\nu_N+\nu_P}{2}+\frac{\nu_S+\nu_P}{2} \end{pmatrix}$	$\frac{\lambda_E+\lambda_P}{2}$
$\frac{\beta_W+\beta_S}{4}$	$\frac{\nu_S+\nu_P}{2}$	$\frac{\beta_E+\beta_S}{4}$

$$\partial_x\left(\beta^n \partial_y u^n\right)\big|_{i,j} \approx \frac{\left(\beta_{i,j}^n \delta_y U_{i,j}^n\right)_{i+1,j} - \left(\beta_{i,j}^n \delta_y U_{i,j}^n\right)_{i-1,j}}{2}$$

$$\approx \frac{\beta_{i+1,j}^n}{4}\left(U_{i,j+1}^n - U_{i,j-1}^n\right)_{i+1,j} - \frac{\beta_{i-1,j}^n}{4}\left(U_{i,j+1}^n - U_{i,j-1}^n\right)_{i-1,j}$$

$$\approx \frac{\beta_{i+1,j}^n}{4} U_{i+1,j+1}^n - U_{i+1,j-1}^n - \frac{\beta_{i-1,j}^n}{4} U_{i-1,j+1}^n - U_{i-1,j-1}^n$$

$$\approx \frac{\beta_E^n}{4}\left(U_{SE}^n - U_{NE}^n\right) + \frac{\beta_W^n}{4}\left(U_{NW}^n - U_{SW}^n\right)$$

In an analogous manner, the other mixed derivative is calculated

$$\partial_y\left(\beta^n \partial_x u\right)\big|_{i,j} \approx \frac{\beta_S^n}{4}\left(U_{SE}^n - U_{SW}^n\right) + \frac{\beta_N^n}{4}\left(U_{NW}^n - U_{NE}^n\right)$$

The preceding approximations lead to the scheme:

$$\begin{aligned}
\frac{U_P^{n+1} - U_P^n}{\tau} =& \frac{\lambda_E^n + \lambda_P^n}{2}\left(U_E - U_P\right) + \frac{\lambda_P^n + \lambda_W^n}{2}\left(U_W - U_P\right) \\
&+ \frac{\nu_S^n + \nu_P^n}{2}\left(U_S - U_P\right) + \frac{\nu_P^n + \nu_N^n}{2}\left(U_N - U_P\right) \\
&+ \frac{\beta_E^n}{4}\left(U_{SE}^n - U_{NE}^n\right) + \frac{\beta_W^n}{4}\left(U_{NW}^n - U_{SW}^n\right) \\
&+ \frac{\beta_S^n}{4}\left(U_{SE}^n - U_{SW}^n\right) + \frac{\beta_N^n}{4}\left(U_{NW}^n - U_{NE}^n\right)
\end{aligned}$$

Table 4.4.2: 3×3 averaging mask $\mathbf{A}\left(U_{i,j}^n\right)$ for the case in which the mixed derivatives are calculated using the standard central differences defined by δ_x^* in (4.4.3).

$\frac{\beta_W+\beta_N+2\beta_P}{8}$	$\frac{1}{2}(\nu_N+\nu_P)+$ $\frac{1}{8}(\beta_W-\beta_E)$	$\frac{\beta_E+\beta_N+2\beta_P}{8}$
$\frac{1}{2}(\lambda_W+\lambda_P)+$ $\frac{1}{8}(\beta_N-\beta_S)$	$-\begin{pmatrix}\frac{\lambda_E+\lambda_W}{2}+\lambda_P \\ \frac{\nu_N+\nu_S}{2}+\nu_P\end{pmatrix}$	$\frac{1}{2}(\lambda_E+\lambda_P)+$ $\frac{1}{8}(\beta_S-\beta_N)$
$\frac{\beta_W+\beta_S+2\beta_P}{8}$	$\frac{1}{2}(\nu_S+\nu_P)+$ $\frac{1}{8}(\beta_E-\beta_W)$	$\frac{\beta_E+\beta_S+2\beta_P}{8}$

As a result, the unknown U_P^{n+1} can be calculated explicitly as a weighted mean of known values at level n. In stencil notation, the averaging mask is given by Table 4.4.1.

SCHEME 2. Using the central differences in (4.4.3):

$$
\begin{aligned}
\partial_x\left(\beta^n\partial_y u^n\right)\big|_{i,j} \approx& \left(\beta_{i,j}^n\delta_x^* U_{i,j}^n\right)_{i+\frac{1}{2},j}-\left(\beta_{i,j}^n\delta_x^* U_{i,j}^n\right)_{i-\frac{1}{2},j}\\
\approx& \beta_{i+\frac{1}{2},j}^n\left(U_{i,j+\frac{1}{2}}^n-U_{i,j-\frac{1}{2}}^n\right)_{i+\frac{1}{2},j}-\beta_{i-\frac{1}{2},j}^n\left(U_{i,j+\frac{1}{2}}^n-U_{i,j-\frac{1}{2}}^n\right)_{i-\frac{1}{2},j}\\
\approx& \frac{\beta_E^n+\beta_P^n}{2}\left(U_{i+\frac{1}{2},j+\frac{1}{2}}^n-U_{i+\frac{1}{2},j-\frac{1}{2}}^n\right)-\frac{\beta_W^n+\beta_P^n}{2}\left(U_{i-\frac{1}{2},j+\frac{1}{2}}^n-U_{i-\frac{1}{2},j-\frac{1}{2}}^n\right)\\
\approx& \frac{\beta_E^n+\beta_P^n}{8}\left(U_S^n+U_{SE}^n-U_N^n-U_{NE}^n\right)\\
&+\frac{\beta_W^n+\beta_P^n}{8}\left(U_N^n+U_{NW}^n-U_S^n-U_{SW}^n\right)
\end{aligned}
$$

In an analogous manner, the other mixed derivative is calculated

$$
\begin{aligned}
\partial_y\left(\beta^n\partial_x u^n\right)\big|_{i,j} \approx& \frac{\beta_S^n+\beta_P^n}{8}\left(U_E^n+U_{SE}^n-U_{SW}^n-U_W^n\right)\\
&+\frac{\beta_N^n+\beta_P^n}{8}\left(U_W^n+U_{NW}^n-U_E^n-U_{NE}^n\right)
\end{aligned}
$$

Note that for this scheme, interpolations for β's and U's are necessary. To use this scheme on the mixed derivatives, the values on the green dots in Figure 4.4.1(b)

need to be interpolated. They are interpolated using all the neighbors of pixel P and P itself, that is, the red dots in Figure 4.4.1(b). e.g.,

$$\begin{aligned} U^n_{i+\frac{1}{2},j+\frac{1}{2}} =& \frac{1}{4}\left(U^n_{i,j} + U^n_{i+1,j} + U^n_{i+1,j+1} + U^n_{i,j+1}\right) \\ =& \frac{1}{4}\left(U^n_P + U^n_E + U^n_{SE} + U^n_S\right) \end{aligned}$$

For this scheme the averaging mask is given in Table 4.4.2.

SCHEME 3. Using the rotationally invariant scheme (2.6.2):

The inclusion of NW, SW, NE, SE neighbors of pixel P produces a grid shown in Figure 4.4.1(b). The black dots are additional interpolations needed to calculate the rotationally invariant scheme. The coefficients for the rotationally invariant scheme were calculated using Matlab's symbolic package. The expressions are too long to be displayed here.

It was found, that Schemes 1 and 2 produced similar results. Scheme 3 that includes the eight neighbors produced less blurry diagonal edges. But as TAND iterates, some other important edges where smoothed, while the other schemes did not produce such results. Therefore, due to less computational complexity, fewer interpolations and the quality of the results Scheme 1 was chosen, i.e, the mixed derivatives used the standard central difference scheme.

After solving all the spatial derivatives approximations, (4.4.2) can be expressed as a product of U^n_p and an averaging mask, $\mathbf{A}(U^n_p)$, written as a function of U^n_p (see [2]). Equation (4.4.1) in matrix-vector notation becomes:

$$\mathbf{U}^{n+1} = \left[\mathbf{Id} + \tau \mathbf{A}\left(\mathbf{U}^n\right)\right]\mathbf{U}^n,$$

where **Id** is the identity matrix. This scheme is known as the *explicit scheme*. The problem with this scheme is that it is only experimentally stable for step sizes $\tau \leq ¼$

which makes the discretization impractical [2]. There is another discretization of (4.4.1) that is numerically stable for any size of τ since $[\mathbf{Id} - \tau\mathbf{A}(\mathbf{U}^n)]$ is diagonal dominant, known as *semi implicit*,

$$\mathbf{U}^n = [\mathbf{Id} - \tau\mathbf{A}(\mathbf{U}^n)]\,\mathbf{U}^{n+1}$$

The accuracy of the solution depends on the choice of τ. With a large τ, the solution can be obtained within few iterations but the accuracy will suffer to the point that is useless. A trade-off between the number of iterations n and τ's size is necessary. This semi-implicit finite-difference scheme produces a nine diagonal linear system that is not symmetric and can be solved with any of the Krylov subspace methods for non-symmetric sparse matrices[102]. The most used are Generalized Minimum Residual Method (GMRES) or conjugated gradient stabilized method (BICGStab).

4.5 SOLVER AND PRECONDITIONER USED

The $145 \times 145 \times 185$ Indian Pines HSI was used in experiments to determine whether to choose BiCGStab or GMRES. TAND was iterated until $n = 8$, the methods were set to stop with a tolerance of 10^{-3} and the time step for the discretization was $\tau = 2$. As TAND iterates the number of iterations and time it needs to solve the linear system was decreasing. The experiments where carried out with the three discretization schemes described in the last section. BiCGStab without preconditioning was three times faster than GMRES in all of them for the first iteration. For all the other iterations it was at least 1.5 times faster than GMRES. So BiCGStab was chosen as the solver. Scheme 3 has the best performance in terms of the time taken to converge to a solution and with fewer iterations. But the discretization step took 1.5 times longer than with any of the other schemes. Scheme 1 and 2 had

similar performance, being Scheme 1 slightly better than Scheme 2. So Scheme 1 was chosen.

A preconditioner based on Peric's direct method to solve nine - diagonal linear systems [103] was developed. The details of this preconditioner are in Appendix A.2. This preconditioner was compared with the ILU(0) preconditioner and the Jacobi explicit diagonalization scheme. ILU was almost 2.8 times faster than Jacobi diagonalization and 2.25 times faster than our proposed preconditioner. Those times were calculated in the first iteration of TAND.

4.6 TAND ALGORITHM

TAND algorithm is depicted using a flow chart in Figure 4.6.1, its pseudo-code is given by Algorithm 4.1.

4.7 CONCLUDING REMARKS

A Tensor Anisotropic Nonlinear Diffusion (TAND) have been proposed and studied. Diffusion tensors were proposed for TAND for Edge Enhancement Diffusion (EED) and for Coherence Enhancement Diffusion (CED). Those proposed diffusion tensors used the orientation and eigenvalues of heat weighted structure tensor developed in Chapter 3. This structure tensor make TAND adaptive to the spectral characteristics of HSI. The proposed diffusion tensor for TAND-CED is more sensitive to the values of the contrast parameter used to define the edges, while Weickert's one is sensitive to its square.

All aspects of TAND implementation have been studied. After implementing three methods to discretize the derivatives, the standard central difference scheme used to discretize the mixed derivatives ∂_{xy}, ∂_{yx}, and the standard central differ-

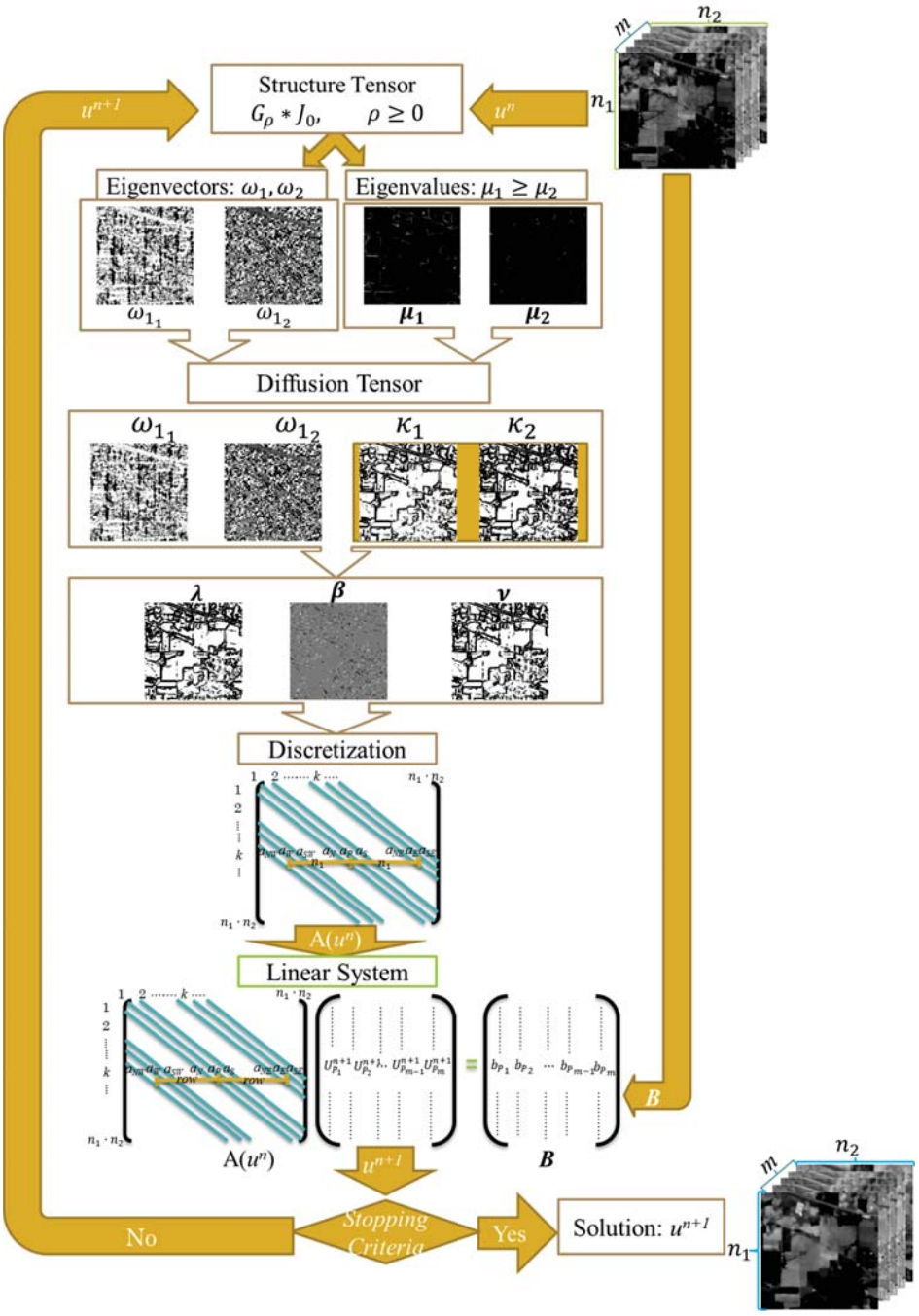

Figure 4.6.1: Flow chart of the TAND algorithm

Algorithm 4.1 Tensor Anisotropic Nonlinear Diffusion

INPUT:

Image U^n

Information from U^n's structure tensor to be used.

- Eignvectors of structure tensor ω_1^n ω_2^n.
- For EED: The largest eigenvalue, μ_1^n.
- For CED The difference between the eigenvalues, $\mu = \mu_1^n - \mu_2^n$.

Contrast parameter ψ

OUTPUT: Image U^{n+1}

1. Calculate κ_1, κ_2
   - For EED use (4.2.1) and (4.2.2)
   - For CED use (4.2.4)
2. Calculate the components of the diffusion tensor λ^n, β^n, ν^n using (4.1.3), (4.1.4) and (4.1.5) respectively
3. Construct the averaging mask A using λ, β, ν. %See the section above
4. Apply boundary conditions
5. Solve the linear system $AU^{n+1} = B$

ence scheme applied to half distances to discretize ∂_x^2 and ∂_y^2 obtained good results in term of interpolations, less computational time and good results. To solve the linear system it was studied the performance of two methods to solve non-symmetric linear systems, BiCGStab and GMRES. BiCGStab was chosen since it needed less iterations and less time to find a solution. A preconditioner was developed and tested against the ILU(0) and the Jacobi explicit diagonalization preconditioner. The ILU(0) was choosen since it needed less iterations and less time to find the solution.

5

EXPERIMENTAL RESULTS

This chapter presents the data used for the experiments, and an empirical guideline to set the parameters used in TAND. Comparison between Nonlinear diffusion and TAND is performed to see the importance of using a tensor. An experimental comparison of the performance of the structure tensors using TAND is presented. To that end, the proposed TAND for EED and CED are used and the comparison is performed by only changing the structure tensor. The aim of these experiments is to show how a weighted structure tensor improves the proposed TAND. Finally a comparison of Weickert's TAND with the proposed TAND is presented.

All methods were implemented in Matlab. The heat weighted structure tensor was implemented as described in Algorithm 3.1. The classical structure tensor was implemented as described in Subsection 3.1. TAND was implemented as described in Algorithm 4.1. All HSIs bands were normalized to the [0, 255] range (8-bits). The diffusion tensor for TAND–CED is defined using (4.2.4) and for TAND–EED using (4.2.1) and (4.2.2). They will show comparisons with the classical structure tensor. In all experiments, a semi-implicit scheme was used, which is solved using BICGStab with a tolerance of 10^{-4} and the discretization time step was $\tau = 5$, any change on those parameters will be specified. For some images, those methods were slow to converge. So, to accelerate the convergence the ILU(0) preconditioner was used.

5.1 DATA

5.1.1 *Data for EED*

For EED, the following HSI were used:

1. The **Indian Pines** image taken with the AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) sensor, flown by NASA/Ames on June 12, 1992 over an area 6 miles west of West Lafayette, IN. This image contains 145×145 pixels at ~ 20 m of spatial resolution and 220 spectral bands in the $400 - 2500\,nm$ range, at $\sim 10\,nm$ of spectral resolution, for which ground truth exists. Fifteen bands were discarded following [5]. Therefore only 185 spectral bands of the Indian Pines image were used in this work. A false color composite is shown in Figure 5.1.1(a). Figure 5.1.1(b) shows its ground reference image[105]

2. A $120 \times 120 \times 145$ region of interest (ROI) of the **Forest Radiance I** was collected by the airborne imaging spectrometer HYDICE over a forest region at the U.S. Army Aberdeen Proving Ground in 1995 by the HYMSMO (Hyperspectral MASINT Support to Military Operations) program. This ROI uses 145 of 224 spectral bands in the $400 - 2500\,nm$ range. The sensor altitude was $5146.33\,ft$. Bands with low signal to noise ratio were rejected as in [106]. This region of interest was used since targets are small and discriminating targets edges are difficult because the image has a strong vertical texture. Ground truth about target location and signatures is available. A false color composite of this ROI is shown in Figure 5.1.1(d) with its ground reference image map shown in Figure 5.1.1(e).

3. The **A. P. Hill** image captured over Fort. A. P. Hill, Virginia in September 2001 using AVIRIS sensor with ~ 3.5 m spatial resolution, and 224 spectral

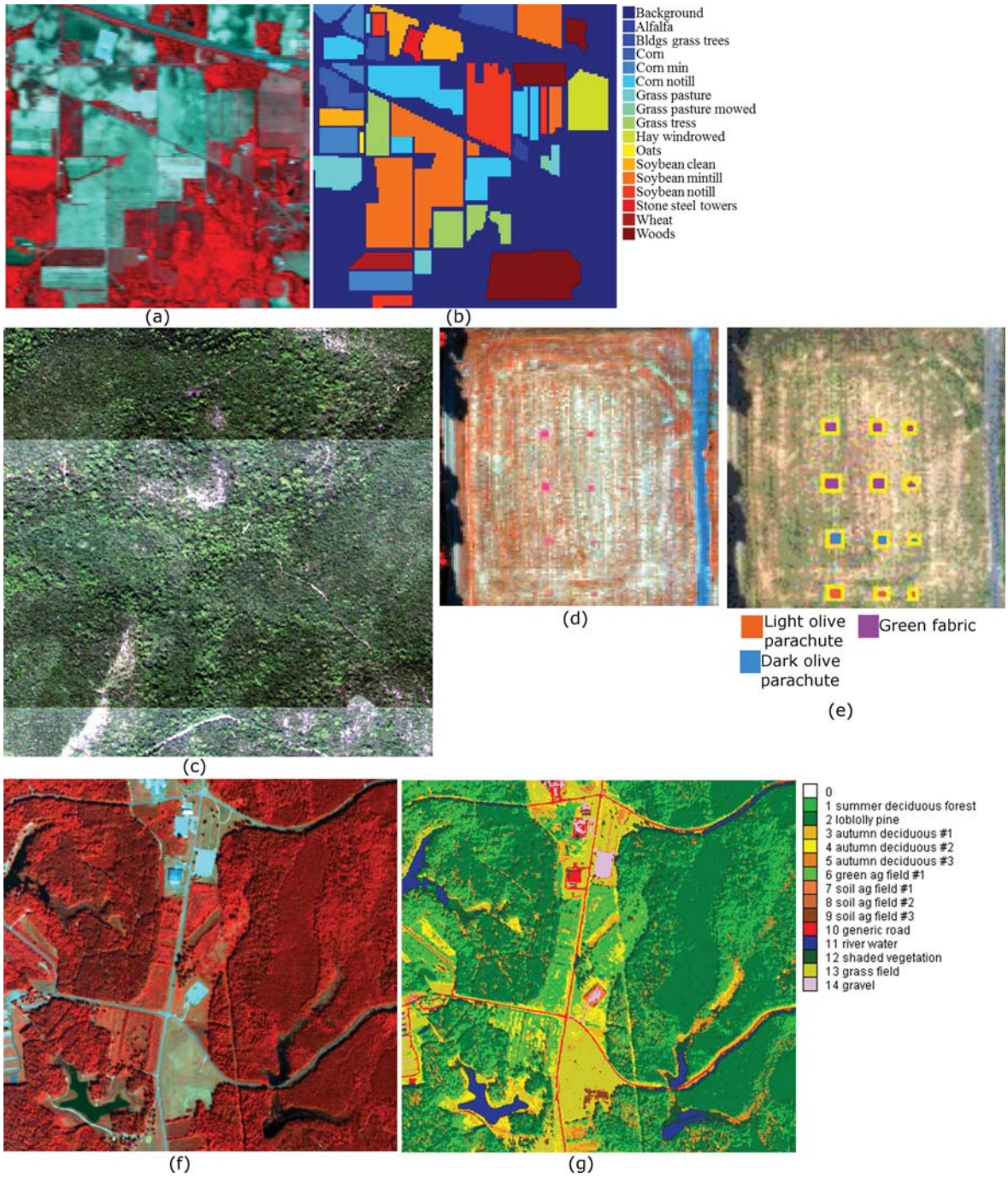

Figure 5.1.1: Hyperspectral images used for EED and their classification maps.(a) False color composite of AVIRIS Indian Pines image with bands [50, 25, 17]. (b) Ground reference map for the Indian Pines image (c) True color composite of AISA Eagle Guanica dry forest $800 \times 800 \times 128$ with bands [54, 35, 15].(d) False color composite of HYDICE Forest Radiance image with bands [75, 37, 18]. (e) Ground reference map for the Forest Radiance image (f) False color composite of AVIRIS A. P. Hill image with bands [45, 24, 12]. (g) Classification map for the Composite of Cuprite image with bands [183, 119, 207]. (e)

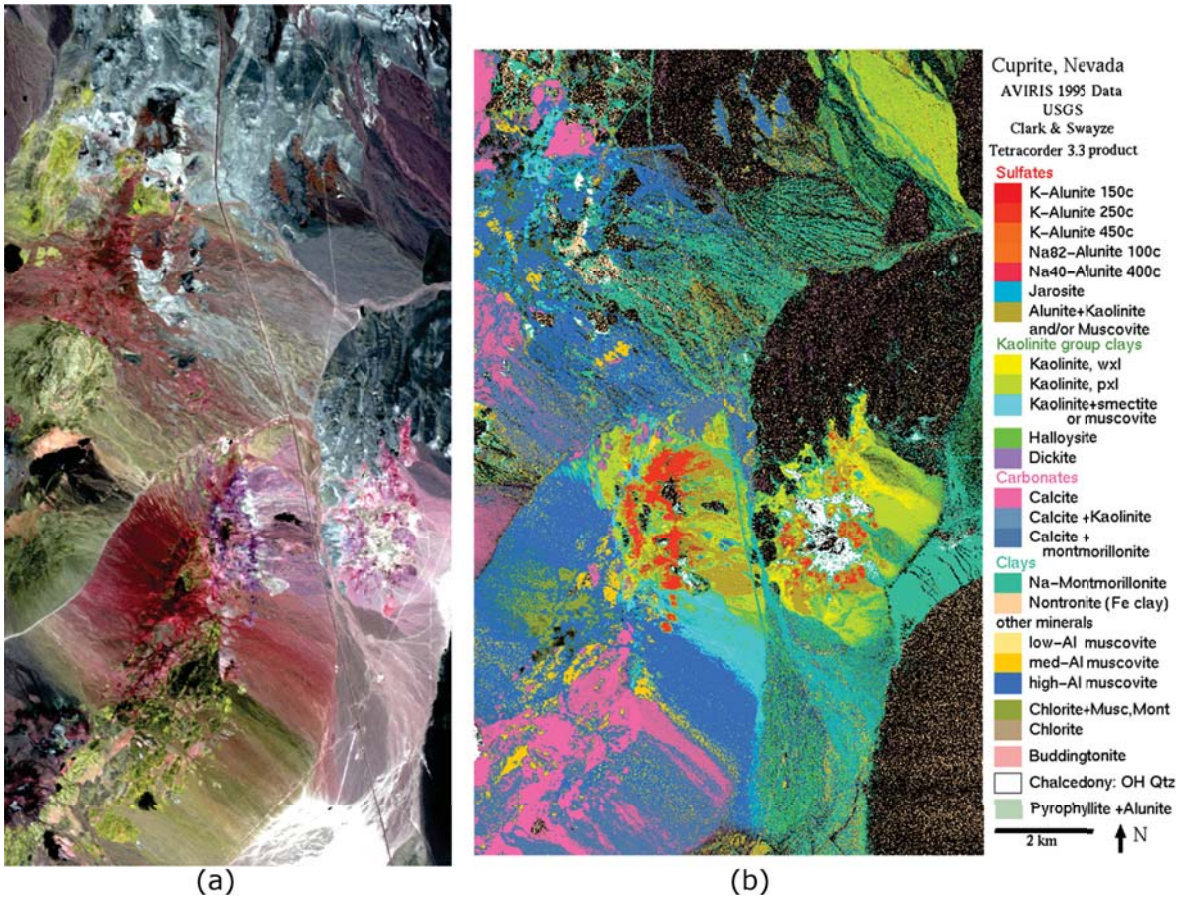

Figure 5.1.2: Cuprite image and its (a) RGB composition of AVIRIS Cuprite image with bands [183, 193, 207] (b) Ground reference map of a ROI of (a) [104].

bands in the $400 - 2500\,nm$ range at $\sim 10\ nm$ of spectral resolution. A false color composite of a $512 \times 540 \times 224$ ROI of this image is shown in Figure 5.1.1(f) and a classification map obtained by minimum distance classifier [107] is shown in Figure 5.1.1(g)

4. The **Cuprite** image taken over the mining district, 2 km north of Cuprite, Nevada, with the AVIRIS sensor, flown by NASA/Ames on June 19, 1997. This image contains five scenes for a total of 640×2378 pixels and 224 bands in the $370 - 2500\ nm$ range. A ROI of the fourth scene of size $512 \times 614 \times 224$ pixels was selected, a composite of this ROI is shown in Figure 5.1.2(a). A ground reference map created using the Tetracorder from USGS [104] is shown in Figure 5.1.2(b).

5. **Guanica** dry forest images taken in the Southwestern Puerto Rico captured with the AISA Eagle sensor. Figure 5.1.1(c) presents a true color composite of a $800 \times 800 \times 128$ ROI of a registered image composed by three pieces of flight lines of AISA image. The registration was done using ENVI software. This image was captured on December of 2007 with a spatial resolution of 1 meter and 128 spectral bands from 397 nm to 995 nm with a spectral resolution of $\sim$ 4.5 nm. Due to its size, this image will be used to test parameters and calculate number of iterations of the solver. TAND-EED denoising performance will not be tested with this image.

5.1.2 *Data for CED*

In this chapter, CED will be used to extract the chain of thyroid cells from a microscopy HSI and to extract ship wakes from an airborne HSI. All the images where

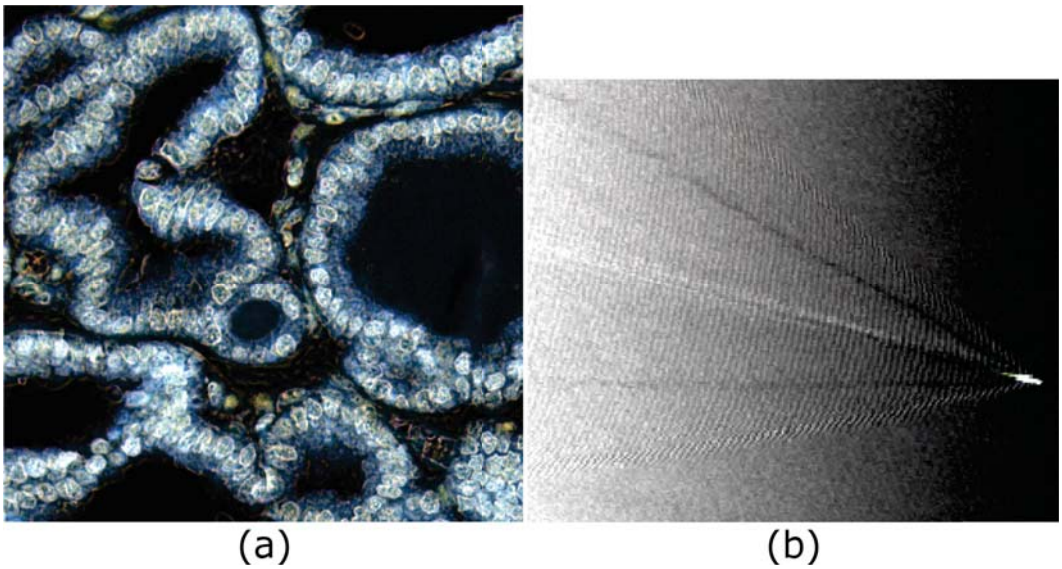

Figure 5.1.3: Original Images used for CED. (a) True color of a ROI of a Thyroid cells image with bands [13 8 3].(b) True color of a ROI of an AVIRIS image [99] with bands [27 17 7].

processed using κ_2 as in (4.2.4). The following two HSI have been used for this section:

1. A ROI of a Thyroid cells image, shown in Figure 5.1.3(a), of size 395×398 pixels that was collected with a Citoviva hyperspectral microscope with range from $420 - 720$ *nm* with 16 bands each of 20 *nm* spectral resolution.

2. A ROI of an AVIRIS image [99], shown in Figure 5.1.3(b), of size 222×266 pixels with 224 bands or channels taken from the Deep Horizon Gulf Of Mexico Oil Spill on May 17, 2010. This image was used in Chapter 4 to show how the components of the proposed diffusion tensor for CED behave.

5.2 PARAMETER SELECTION

This section gives some empirical guidelines to select the parameters used in TAND in both cases. The proposed TAND needs two parameters: 1. The contrast param-

eter ψ; 2. The stoping time $t_n = n\tau$, related to the number of iterations, n and the time step size τ.

1. Finding an "optimal" value for the contrast parameter ψ is usually problem dependent. Perona-Malik [73] suggested using the cumulative histogram of μ_1 in (4.2.1) or μ in (4.2.4) and then setting ψ to a certain percentile. They choose the 90^{th} percentile, i.e., 90% of all gradients are smaller than ψ. In the experiments carry out during this work, values close to the 50^{th} percentile as suggested by Weickert [20] produce better results.

2. It is important to note that the *stopping time* $t_n = n\tau$ gains importance when the diffusion process is a restoration method, as in EED. For CED, the evolution as the iterations of TAND increase is of most interest. In contrast with the isotropic case where $\sigma = \sqrt{2t_n}$ [45], the smoothing in nonlinear diffusion in general is non-uniform and the time t_n is not directly related to a spatial scale. Searching for a stopping criterion has been an active topic. For a review of the latest techniques see [108, 109, 23]. For Semi-implicit methods, the size of τ is only restricted by the quality of the solution [5]. Values of τ in $[0.25, 10]$ have been tried. $\tau = 5$ produce good results with a relative residual error tolerance of BiCGStab set to 10^{-4}. Given that, the following stopping criteria using the *relative entropy of* κ_1, the ratio between κ_1's entropy at iteration n, $\xi(\kappa_1^n)$ and κ_1's entropy of the original image, $\xi(\kappa_1^1)$, is proposed:

$$\frac{\xi(\kappa_1^n)}{\xi(\kappa_1^1)} \leq thr.$$

where thr is a user defined threshold. $\xi(\kappa_1^n)/\xi(\kappa_1^1)$ decreases monotonically from 1 to 0 during TAND-EED as seen in Figure 5.2.2. Figure 5.2.3 shows the images resulting using the thresholds defined by the horizontal red lines in Figure 5.2.2. Informally speaking, the processing in TAND-EED is driven

by the 'edge information' encoded by κ_1. So, a threshold *thr* based on the desired reduction factor of the uncertainty associated with the 'edge information' in κ_1 with respect to that of the original image will be more appropriated. In each iteration of TAND-EED, the uncertainty associated with κ_1 is reduced as shown in Figure 5.2.1. This behavior occurs independently of the preconditioner used as shown in Figure 5.2.1(b), the time step size, τ and the complexity of the features in the image as the images used have different complexity. Figure 5.2.3 shows the original images and its stopped versions using $thr \leq 0.5$. The stopping criteria produce denoised images that preserving the majority of the main features in them.

5.3 ABOUT THE WEIGHTS

In the definition of the heat weights given in (3.3.2), there is a parameter s that is fixed to a value of 1. Experiments were conducted on the heat weighted structure tensor to see if varying the value of s with the number of iterations, n, of TAND, will give some advantage over using a fixed value of 1. A value of $n = 50$ iterations of TAND–EED was used for the Indian Pines image and $n = 200$ TAND–CED was used for the thyroid tissue image. Figure (5.3.1)(a) and (b) show that varying s with n for TAND–EED produces stronger diffusion without blurring some important edges. In general, the results are not counterproductive. Figure 5.3.1(c) and (d) shows TAND–CED after $n = 200$ iterations for $s = 1$ and varying s with n respectively. From those figures, it is clear that varying s with n was counterproductive as it changed the orientation of the structure tensor resulting in cells shaped like rectangles.

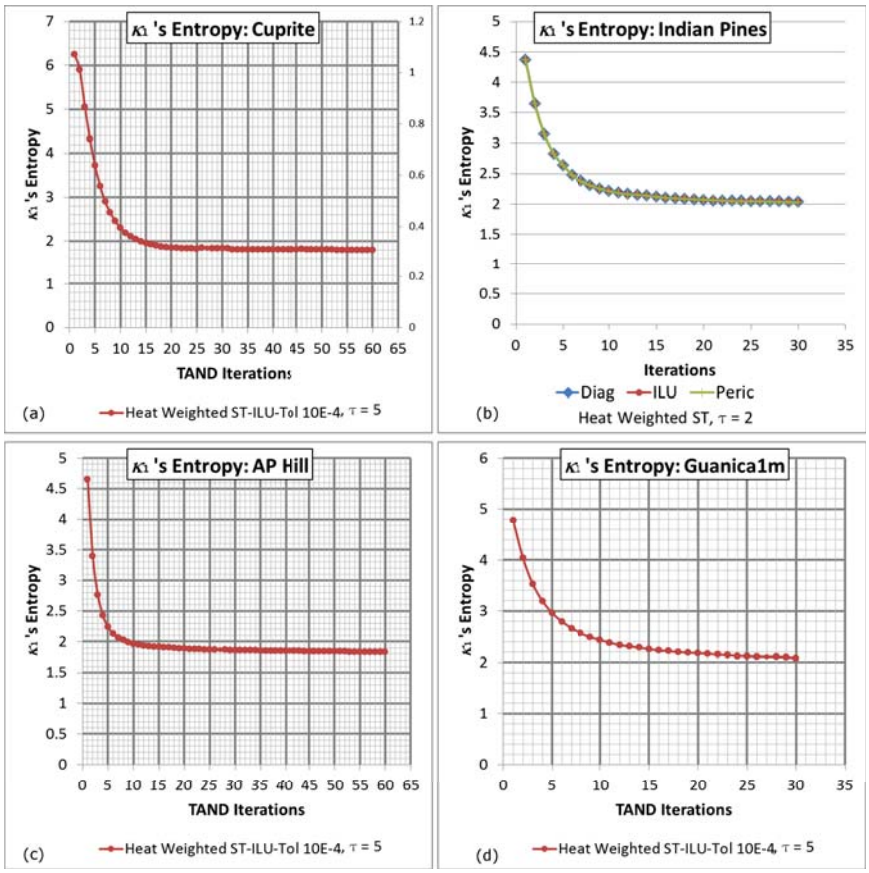

Figure 5.2.1: κ_1's Entropy vs. TAND iterations. (a) Evolution of κ_1's Entropy of Cuprite image as TAND-EED iterate. ILU(0), the heat weighted structure tensor, $\sigma = 0.8$, $\psi = 40^{th}$ percentile of μ_1 and $\tau = 5$ are used (b) Comparison of the evolution of κ_1's Entropy of Indian Pines image using different preconditioners to solve TAND-EED. The heat weighted structure tensor, $\sigma = 0.8$, $\psi = 55^{th}$ percentile of μ_1 and $\tau = 2$ are used. (c) Evolution of κ_1's Entropy of AP Hill image as TAND-EED iterate. ILU(0), the heat weighted structure tensor, $\sigma = 1$ $\psi = 45^{th}$ percentile of μ_1 and $\tau = 5$ are used. (d) Evolution of κ_1's Entropy of Guanica $800 \times 800 \times 128$ image as TAND-EED iterate. ILU(0), the heat weighted structure tensor, $\sigma = 1.5$, $\psi = 45^{th}$ percentile of μ_1 and $\tau = 5$ are used.

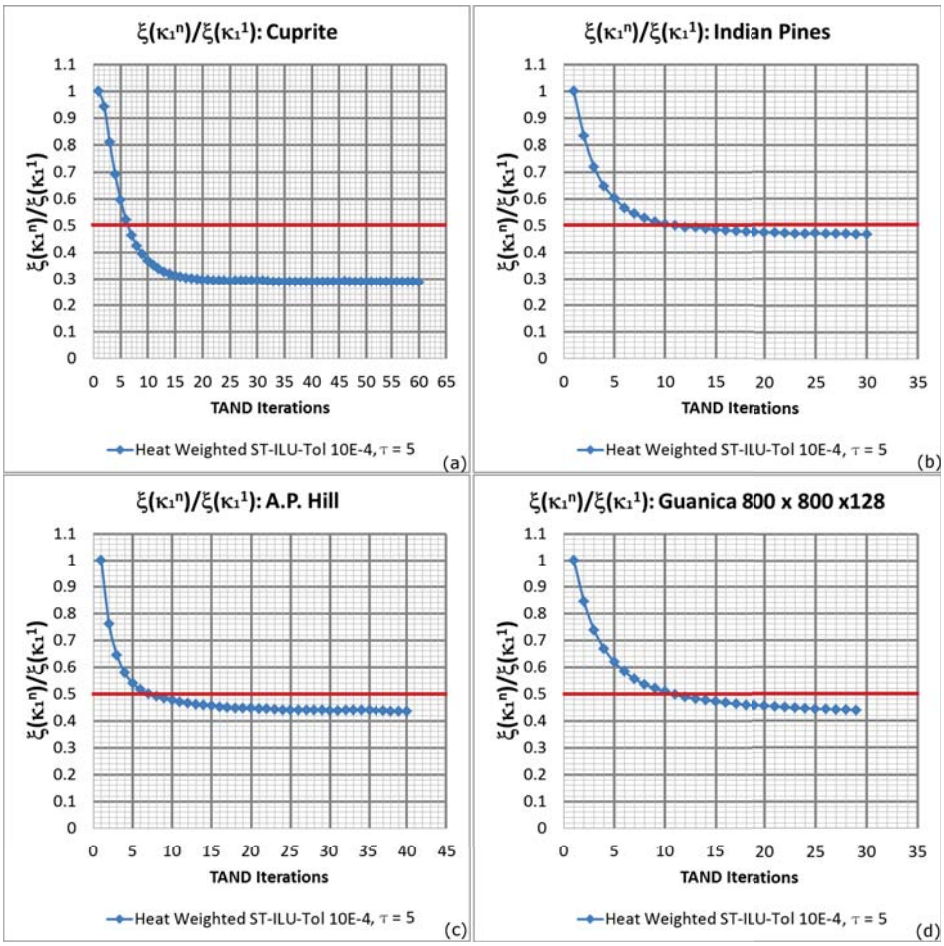

Figure 5.2.2: Relative entropy of κ_1 of the curves in Figure 5.2.1. (a) Cuprite image (b) Indian Pines; (c) A. P. Hill (c) Guanica $800 \times 800 \times 128$. The red horizontal lines are proposed thresholds for those images.

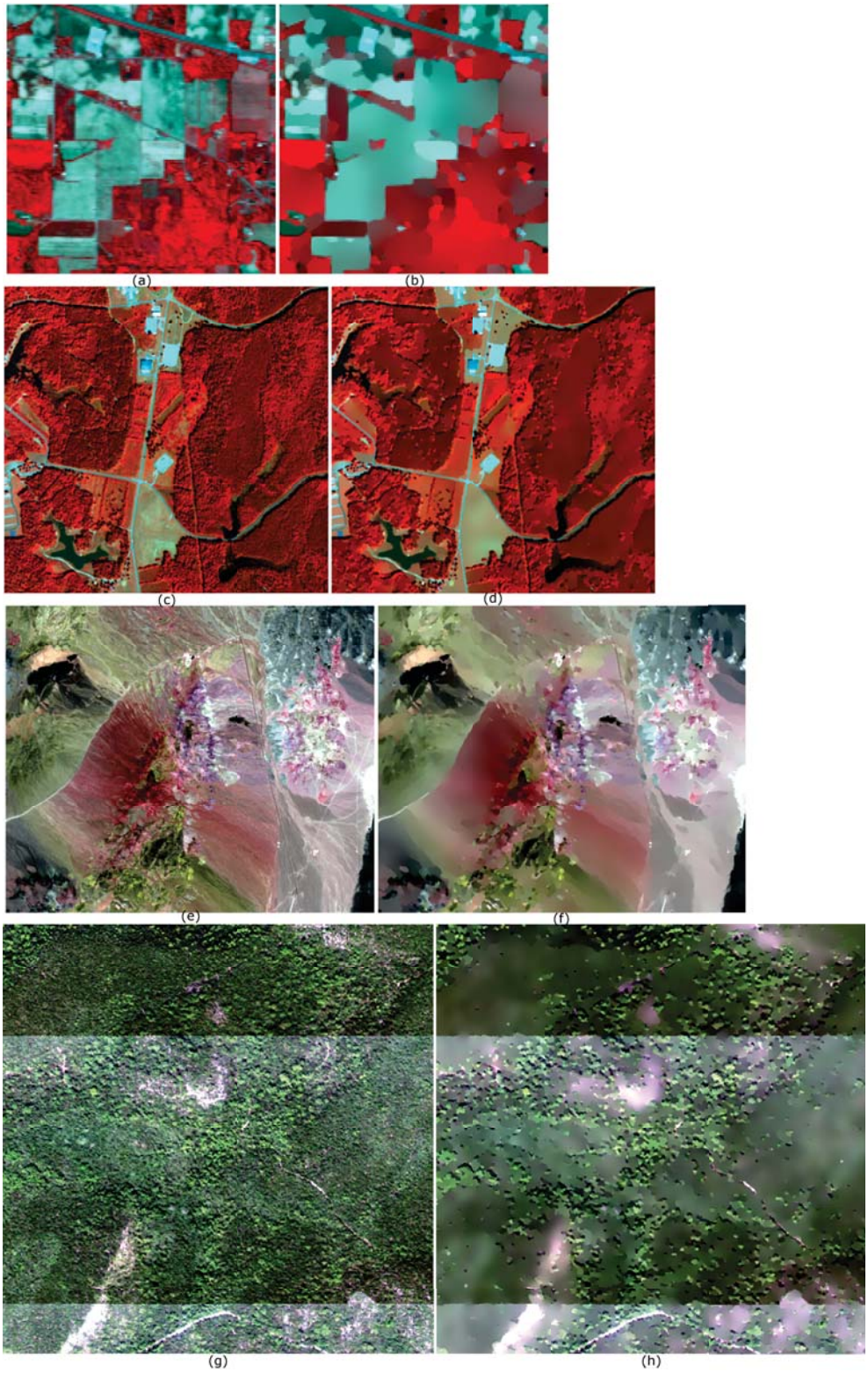

Figure 5.2.3: Images produced with relative entropy of $\kappa_1$as the stopping criteria with thresholds defined with the horizontal red lines in Figure 5.2.2. (a) $\xi(\kappa_1^1)$ is calculated using Indian Pines Original image ;(b) Indian Pines image after TAND-EED stopped when $\xi(\kappa_1^n) = 0.5 \cdot \xi(\kappa_1^1)$, $n = 10$. (c) A.P. Hill original image (d) same as (b) but using A.P. Hill image, $n = 7$ (e) Cuprite Original image. (f) Cuprite after TAND-EED stopped when $\xi(\kappa_1^n) = 0.4 \cdot \xi(\kappa_1^1)$, $n = 10$. (g) Guanica original Image $800 \times 800 \times 128$ (h) same as (b) but using A.P. Hill image, $n = 10$.

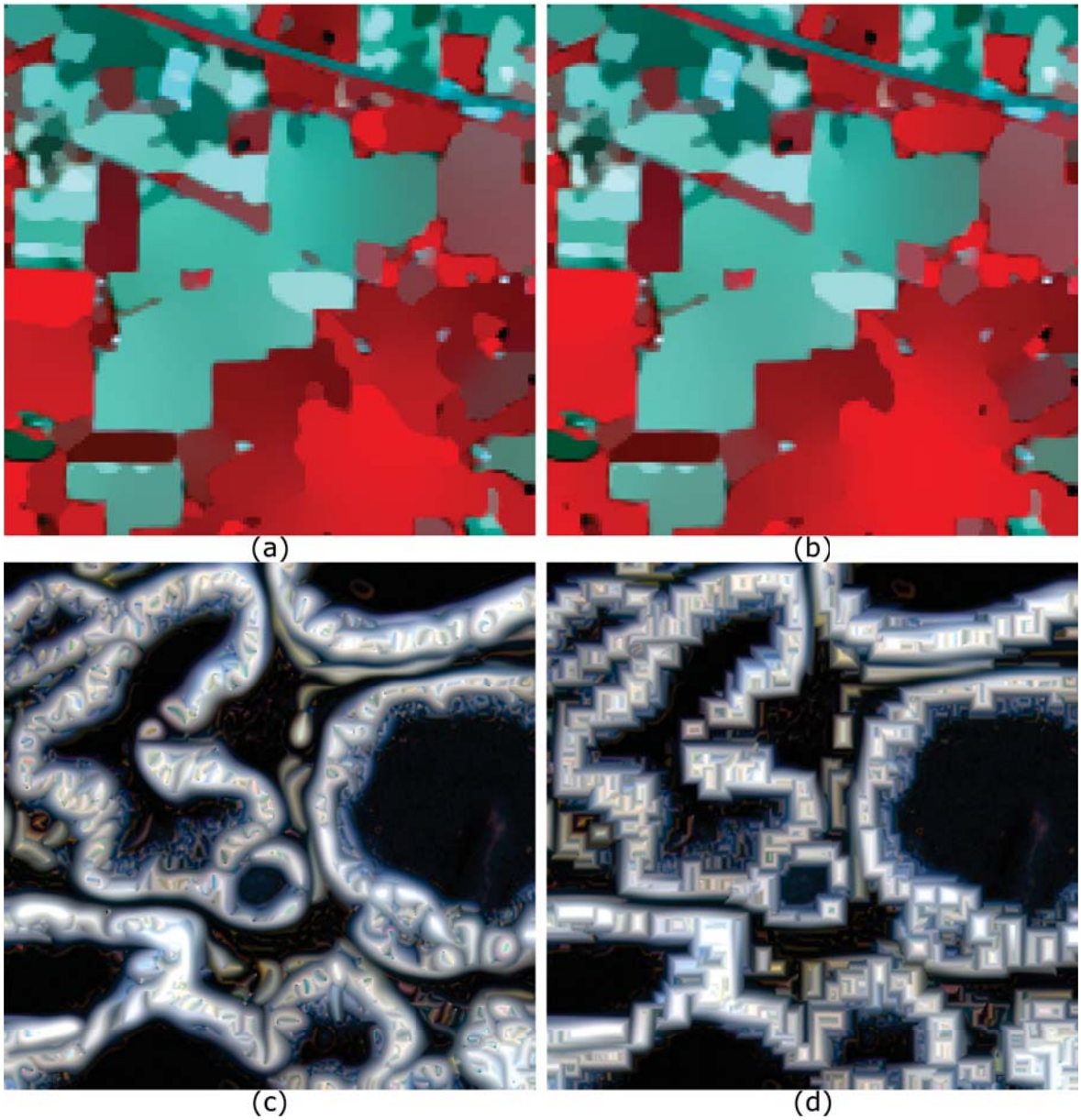

Figure 5.3.1: Varying s in (3.3.2) with TAND's iterations n. (a) True color of AVIRIS Indian Pines image after TAND–EED. with $s = 1$, $n = 50$ iterations (b) True color of AVIRIS Indian Pines image after TAND–EED. Varying s with n, $n = 50$ iterations (c) True color of to the thyroid tissue after TAND–CED with $s = 1$, $n = 200$ iterations (d) True color of to the thyroid tissue after TAND–CED varying s with n, $n = 200$ iterations

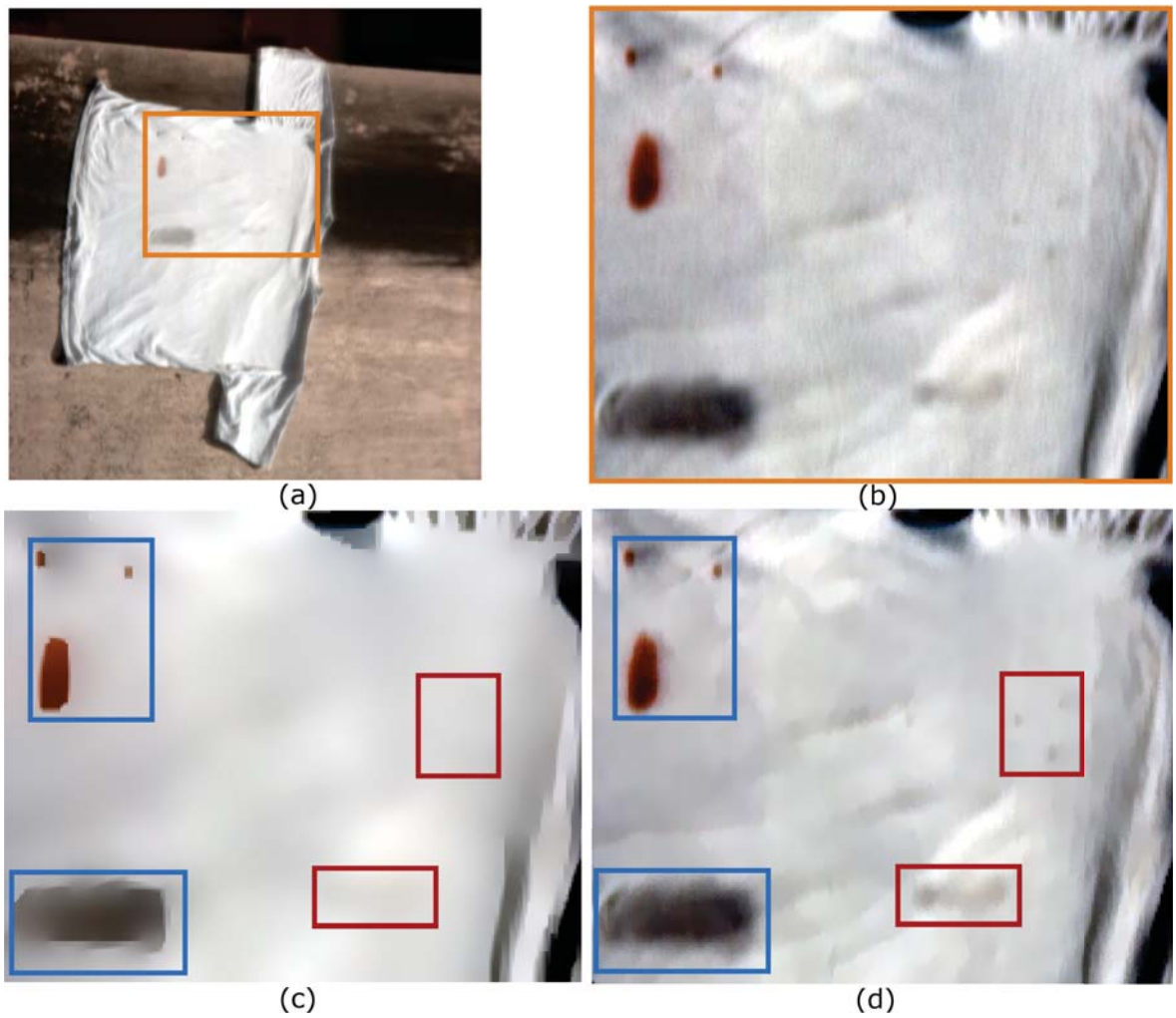

Figure 5.4.1: Comparison between nonlinear diffusion and tensor anisotropic nonlinear diffusion. (a) Original HSI (b) ROI of (a). (c) Nonlinear diffusion as proposed in [5]; $n = 2$, $\gamma = 6.25$, $\psi = 1.2\,10^{-2}$ (d) TAND using the heat weighted structure tensor, $n = 2$, $\sigma = 1$, $\psi = 55^{th}$ percentile of μ_1.

5.4 NONLINEAR DIFFUSION VS. TENSOR ANISOTROPIC DIFFUSION

Figure 5.4.1 illustrates the difference between using nonlinear diffusion with a diffusivity function [5] and using a diffusion tensor based on the heat weighted structure tensor. Figure 5.4.1(a) shows a RGB composite of a 640×640 HSI with 120 bands in the $400 - 900\,nm$ range of a shirt with stains of oil and ketchup. This image was taken with a SOC-700 hyperspectral imager from Surface Optics Company.

Figure 5.4.1(b) shows the region of interest (ROI) inside of the orange square in Figure 5.4.1(a) of size 184×225 where the stains are.

Figure 5.4.1(c) shows the ROI in Figure 5.4.1(b) after being denoised with the PDE-based nonlinear diffusion proposed in [5]. This method uses the PDE in (2.4.1) but instead of using a diffusion tensor D, it uses the diffusivity function in (4.2.1). The PDE is discretized with a semi-implicit scheme that produces a five diagonal linear system. This system is solved with a conjugate gradient method preconditioned with incomplete Cholesky. The filtered image was produced with a Matlab implementation provided by the authors of [5]. Figure 5.4.1(d) shows the denoised image with TAND-EED using the heat weighted structure tensor. Figure 5.4.1(c) used $n = 2$, $\gamma = 6.25$, $\psi = 1.2 \times 10^{-2}$ and Figure 5.4.1(d) used $n = 2$, $\sigma = 1$, $\psi = 55^{th}$ percentile of μ_1.

Usually PDE-based nonlinear diffusion is very good at preserving high contrast edges as in the regions highlighted with blue squares in Figure 5.4.1(c). But low contrast edges such as the ones highlighted in red are not preserved. Both kinds of edges are preserved with the proposed TAND and in general there is a more natural look to the image which preserves some wrinkles as well. Nonlinear diffusion is the PDE-based diffusion algorithm of choice since it is computationally less expensive than TAND, but the extra cost of TAND is justified by the quality of the results.

5.5 COMPARISON OF THE STRUCTURE TENSORS USING TAND

5.5.1 *TAND for Edge Enhancing Diffusion.*

5.5.1.1 *Experiments with A.P. Hill Image*

Figure 5.5.1 shows the comparison of the structure tensors in TAND-EED using A. P. Hill image with parameters$\sigma = 1,\ \tau = 5,\ \psi = \mu_1'$s 45 percentile. The left column shows the results using the heat weighted ST and the right column using the classical structure tensor. Figure 5.5.1(a) shows the $\lambda-$component of the diffusion tensor for iteration $n = 1$, Figure 5.5.1 (b) shows the same component for iteration $n = 35$, and Figure 5.5.1(c) shows the A. P. Hill image resulting after TAND-EED with $n = 35$ iterations using the heat weighted structure tensor. Figure 5.5.1(d), (e), and (f) show same images as in (a), (b), and (c) respectively but using the classical structure tensor. From Figure 5.5.1(a) to (b) and Figure 5.5.1(d) to (e) it is clear that the $\lambda-$component of the diffusion tensor have been smoothed with both structure tensors and both preserve main features in the image. The blue squares in Figure 5.5.1(b) show features that the heat weighted structure tensor preserves while the classical one does not. The results are very similar for both structure tensors in the vegetation but the zoomed section of Figure 5.5.1(b) and (c) shows that the heat weighted structure tensor preserves the features of the road for more iterations while the classical structure tensor, shown in Figure 5.5.1(e) and (f), does not.

5.5.1.2 *Experiments with Indian Pines image*

Figure 5.5.2 shows the results of using TAND–EED with the classical structure tensor (top row) and the heat weighted structure tensor (bottom row) for iteration $n = 1$ (left column) and $n = 4$ (right column). With $n = 4$, TAND produces an over-smoothed image that helps to visualize the time evolution of the diffusion

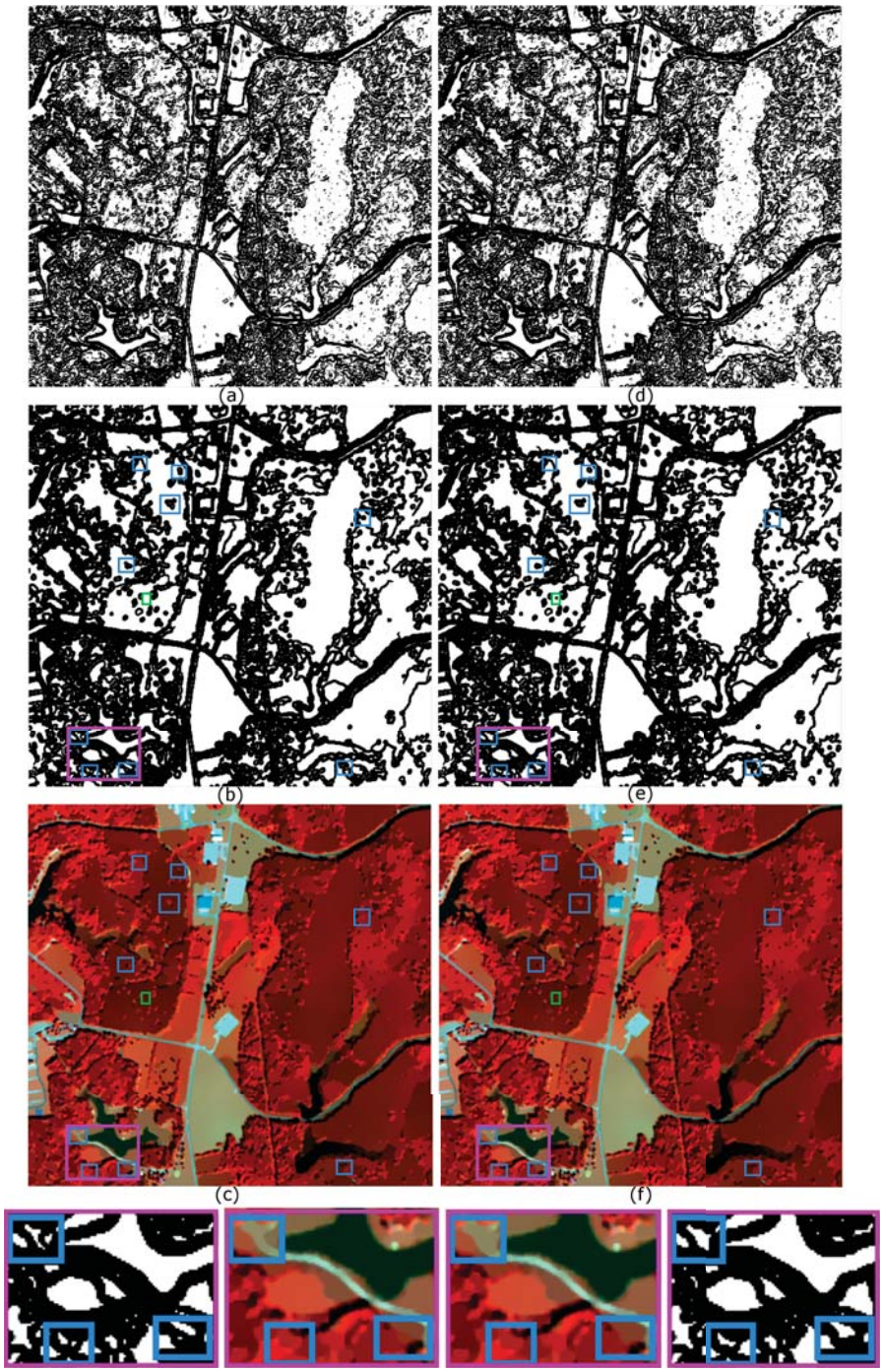

Figure 5.5.1: Comparison of the effect of the structure tensors in TAND-EED using A. P. Hill image. $\sigma = 1$, $\tau = 5$, $\psi = \mu_1'\text{s}\,45\,\text{percentile}$ (a) $\lambda-$component of the diffusion tensor for iteration $n = 1$ using the heat weighted structure tensor. (b) $\lambda-$component of the diffusion tensor for iteration $n = 35$ using the heat weighted structure tensor. (c) A. P. Hill after TAND-EED $n = 35$. (d), (e), and (f) same as (a), (b), and (c) respectively but using the classical structure tensor.

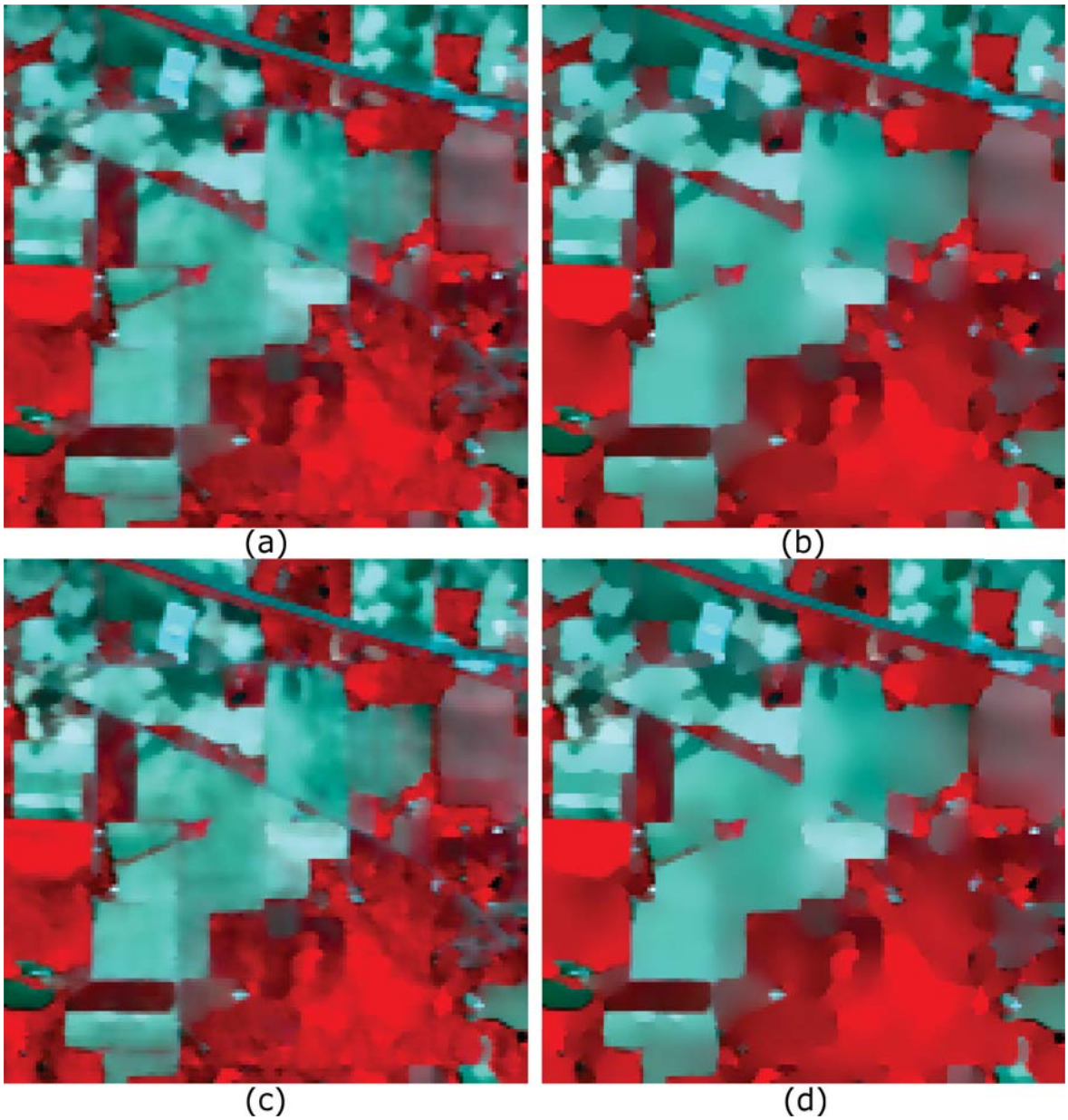

Figure 5.5.2: False color of Indian Pines image after TAND-EED. $\sigma = 0.8$, $\rho = 2$, $\psi = \mu_1'$s 55^{th} percentile; (a) with the classical structure tensor, $n = 1$. (b) with the classical structure tensor, $n = 4$. (c) with the heat weighted structure tensor, $n = 1$. (d) with the heat weighted structure tensor, $n = 4$.

Figure 5.5.3: Comparison of the effect of the structure tensor using the time evolution after iterations n of the λ –entry of the diffusion tensor of TAND-EED applied to Indian Pines image. $\sigma = 0.8$, $\tau = 2$, $\psi = \mu_1' \mathrm{s}\, 55^{th}$ percentile. (a) λ using the classical structure tensor, $n = 1$. (b) λ using the classical structure tensor, $n = 4$. (c) and (d) same as (a) and (b) using the heat weighted structure tensor.

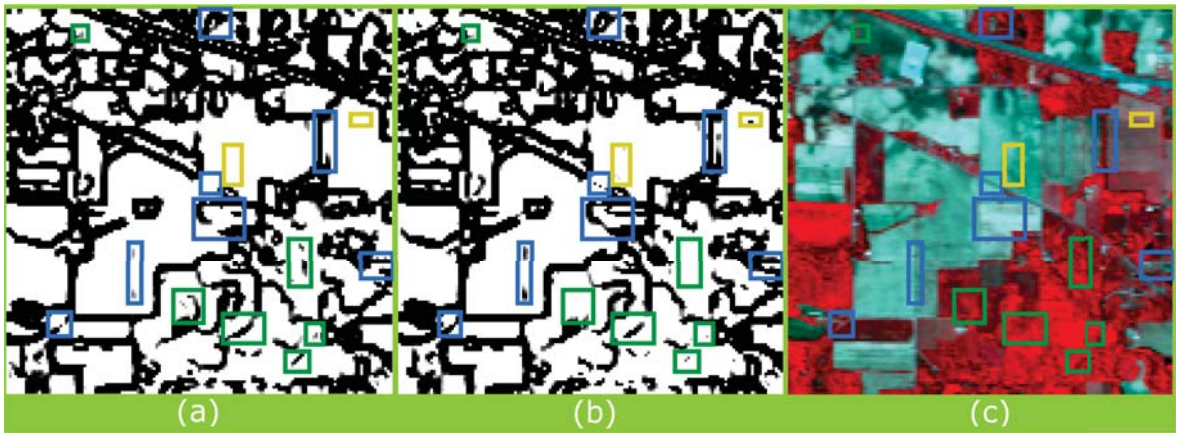

Figure 5.5.4: Time evolution of the λ –entry of the diffusion tensor. $\sigma = 0.8$, $\tau = 2$, $\psi = \mu_1'$s 55^{th} percentile, $n = 3$. (a) λ using the classical structure tensor; (b) λ using the heat weighted structure tensor; (c) original image with the mark up sections.

tensor. From Figure 5.5.2, it can be seen that both methods produce a smoothing preserving the majority of the main edges. To better observe that the heat weighted structure tensor produces better discrimination of structures that need to be persistent through the iterations than the classical tensor, the $\lambda-$entry of the diffusion tensor using both structure tensors will be shown and analyzed. Figure 5.5.3 shows the evolution of the $\lambda-$entry of the diffusion tensor (see (4.1.2)) using the classical and the proposed structure tensor. Both diffusion tensors use $\sigma = 0.8$, $\rho = 2$, $\psi = 55^{th}$ percentile of μ_1, and the eigenvalues defined in (4.2.1) and (4.2.2). Figure 5.5.3(a) and (b) show results using the classical structure tensor with $n = 1$ and $n = 4$ respectively. Figure 5.5.3(c) and (d) show results using the heat weighted structure tensor with $n = 1$ and $n = 4$ respectively.

It can be concluded that as a result of the nonlinear nature of TAND, the $\lambda-$entry of the diffusion tensor has been smoothed: preserving the most important edges and eliminating small variations in the homogeneous regions, as well as showing the nonlinear nature of the structure tensor. Figure 5.5.4 shows a comparison between the $\lambda-$entry of the diffusion tensor after $n = 3$ iterations using the classical and the proposed structure tensor. In both diffusion tensors, the same parameters

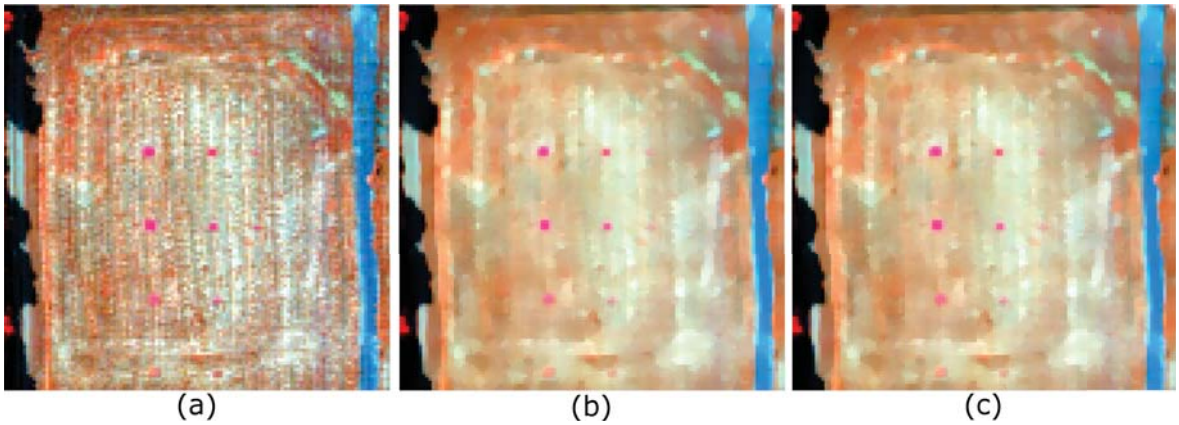

Figure 5.5.5: TAND–EED applied to FR1. $\sigma = 0.5$, $\tau = 1.2$, $\psi = \mu_1'\text{s}\,50^{th}$ percentile, $n = 1$ (a) original image; (b) using the classical structure tensor; (c) using the heat weighted structure tensor.

as in Figure 5.5.3 are used. Figure 5.5.4(a) shows λ using the classical structure tensor. Figure 5.5.4(b) shows λ using the heat weighted structure tensor. Figure 5.5.4(c) shows the original image. The highlighted regions in the images correspond in all three images. Highlighted regions in blue show that the diffusion tensor using the heat weighted ST can preserve important structures longer, i.e., for more iterations. Highlighted regions in green show several structures produced by illumination changes preserved by λ using the classical structure tensor. Several of those structures are still preserved in the next iteration; see Figure 5.5.3(b). Highlighted regions in yellow show structures produced by illumination changes preserved by λ using the heat weighted ST. Those pixels disappear in the next iteration as can be seen in Figure 5.5.3(d). From Figure 5.5.4, it can be concluded that the heat weighted structure tensor enhances the capabilities of the diffusion tensor to discriminate which edges are more important to keep.

5.5.1.3 *Experiments with Forest Radiance* 1 *Image*

To process FR1, it is necessary to smooth the image, since there are vertical textures from the grass. But the smoothing cannot be too strong or it will erase the subpixel targets. Figure 5.5.5 shows false color images resulting from TAND for EED

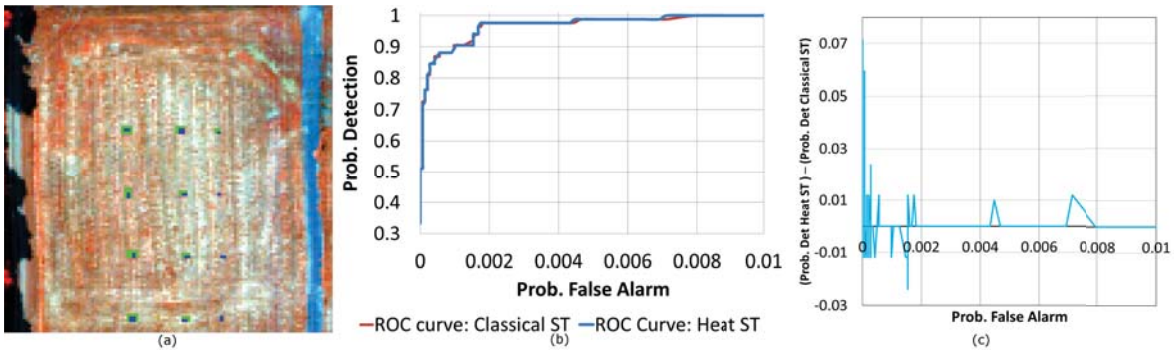

Figure 5.5.6: Forrest Radiance 1 detection results. (a) Original Image with pixels used to estimate the target signature (blue) and the ground truth (blue+green). (b) Comparison of ROC curves generated using the Matched Filter target detection method applied to images shown in Figure 5.5.5(b) using the classical structure tensor, (red line) and Figure 5.5.5(c) the heat weighted structure tensor, (blue line). (c) Difference plot between probabilities of detection using the heat weighted structure tensor and the classical structure tensor.

using the classical and heat weighted structure tensor both with $\sigma = 0.5$, $\psi = \mu_1'$s 50^{th} percentile, $n = 1$. Figure 5.5.6(a) shows 26 target pixels (blue) from 84 pixels of the ground truth (green plus blue) used to generate the detection rules. Target pixels were used to estimate the target spectra using the mean of the 26 pixels. Figure 5.5.6(b) shows a comparison of the ROC curves of the FR1 image after one iteration of TAND for EED using the classical structure tensor (red line) and the heat weighted structure tensor (blue line). Both cases used the matched filter target detection rule for each of the images in Figure 5.5.5(b) and (c), and were compared with the ground truth pixels. It is clear that the results are visually similar and the detection performance is almost identical. From Figure 5.5.6(c), it can be seen that the proposed heat weighted structure tensor improves the detection between 1 to 7 percent for certain probabilities of false alarm over the classical structure tensor. In this case, the similarity of the results is due to the fact that the targets are only 84 pixels from 21025 in total. This shows that, under this circumstance, the heat weighted structure tensor is at least as good as the classic one in terms of the ROC curve. The Matched Filter target detection method and the ROC curves

values were generated using the ENVI software. The next section will show that for a more challenging problem, the heat weighted structure tensor produces better results than the classic structure tensor.

5.5.1.4 *Experiment with Cuprite Image*

Figure 5.5.7 shows the $\lambda-$component of the diffusion tensor for the Cuprite image. Left column shows results of the proposed TAND-EED using the heat operator ST and the right column shows results of the proposed TAND-EED using the classical ST. Comparing for $n = 1$ in Figure 5.5.7 (a) and (d) the heat weighted structure tensor helps the diffusion tensor to preserve the roads and other important features while the classical ST does not. When the proposed TAND-EED using the heat weighted ST iterates, as shown in Figure 5.5.7(b) for $n = 5$ and (d) for $n = 15$, preserves the roads and the different materials smoothing its homogeneous regions and not its edges. On the other hand, the proposed TAND-EED using the classical structure tensor, shown in Figure 5.5.7(e) for $n = 5$ and (f) for $n = 15$ does not. This fact is more notorious in the time evolution shown in Figure 5.5.8 where the results of the proposed TAND-EED are shown for the same images as in Figure 5.5.7

5.5.2 *TAND for Coherence Enhancing Diffusion.*

5.5.2.1 *Thyroid Cells Image.*

To show the performance of CED with the heat weighted structure tensor for an image with complex small structures, the thyroid tissue image is used. It is difficult to evaluate this image by inspecting the diffusion entries of the diffusion tensor so the evaluation will be done by studying its evolution through the iterations. Figure 5.5.9 shows the thyroid tissue image processed using the classical structure tensor in the left column and the heat weighted structure tensor in the right column. The

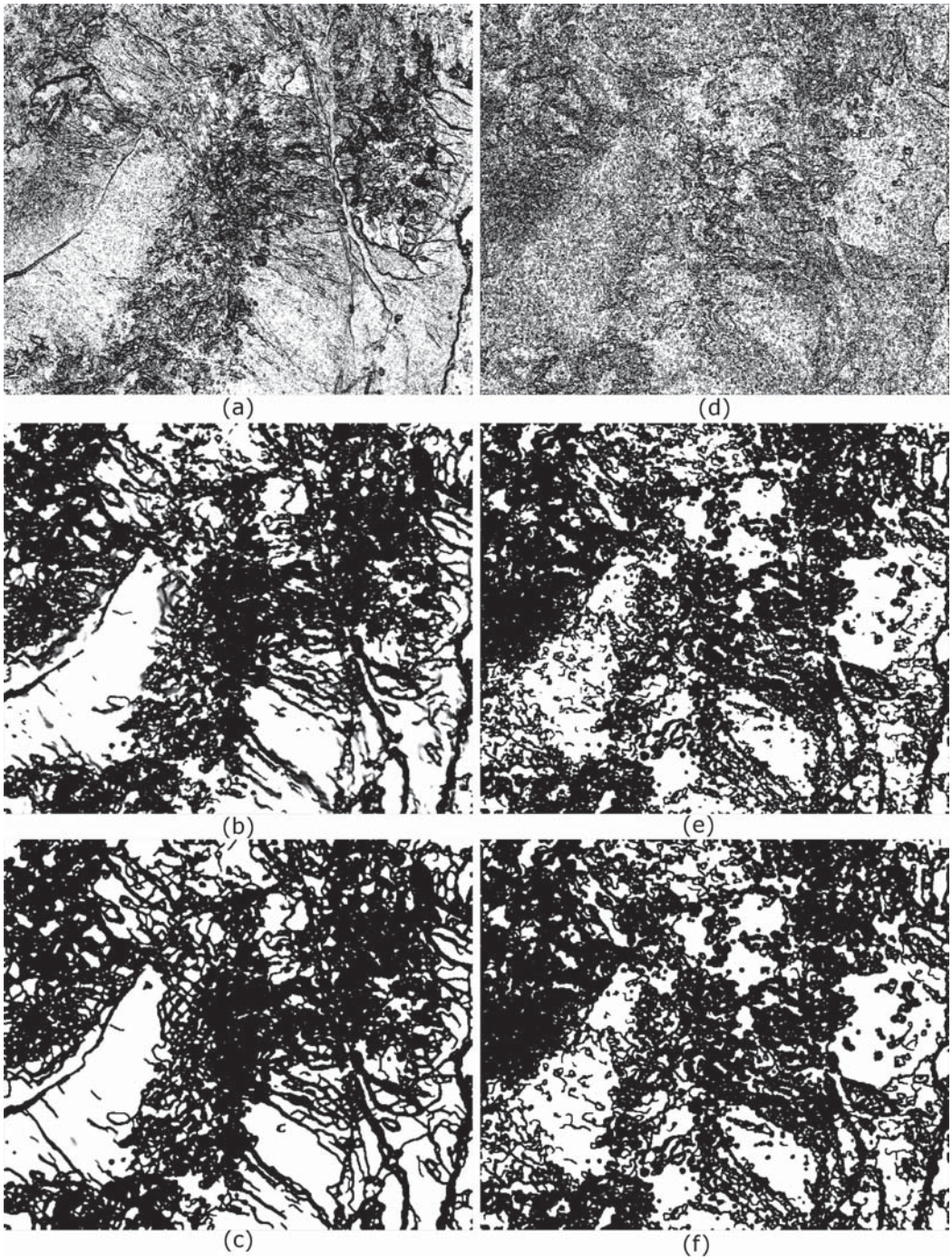

Figure 5.5.7: Comparison of the effect of the structure tensor using the time evolution of the $\lambda-$entry of the diffusion tensor of TAND-EED applied to Cuprite image. $\sigma = 1$, $\tau = 5$, $\psi = \mu_1'$s 45 percentile (a) $n = 1$ using the heat weighted structure tensor. (b) $n = 5$ using the heat weighted structure tensor. (c) $n = 15$ using the heat weighted structure tensor. (d), (e), and (f), are the same as (a), (b), and (c) respectively but using the classical structure tensor.

Figure 5.5.8: Comparison of the effect of the structure tensor using the time evolution of the resulting images of TAND-EED applied to Cuprite image. (a) $n = 1$ using the heat weighted ST; (b) $n = 5$ using the heat weighted ST; (c) $n = 10$ using the heat weighted ST; (d) $n = 15$ using the heat weighted ST; (e), (f), (g), and (h) same as (a), (b), (c), and (d) respectively but using the classical ST.

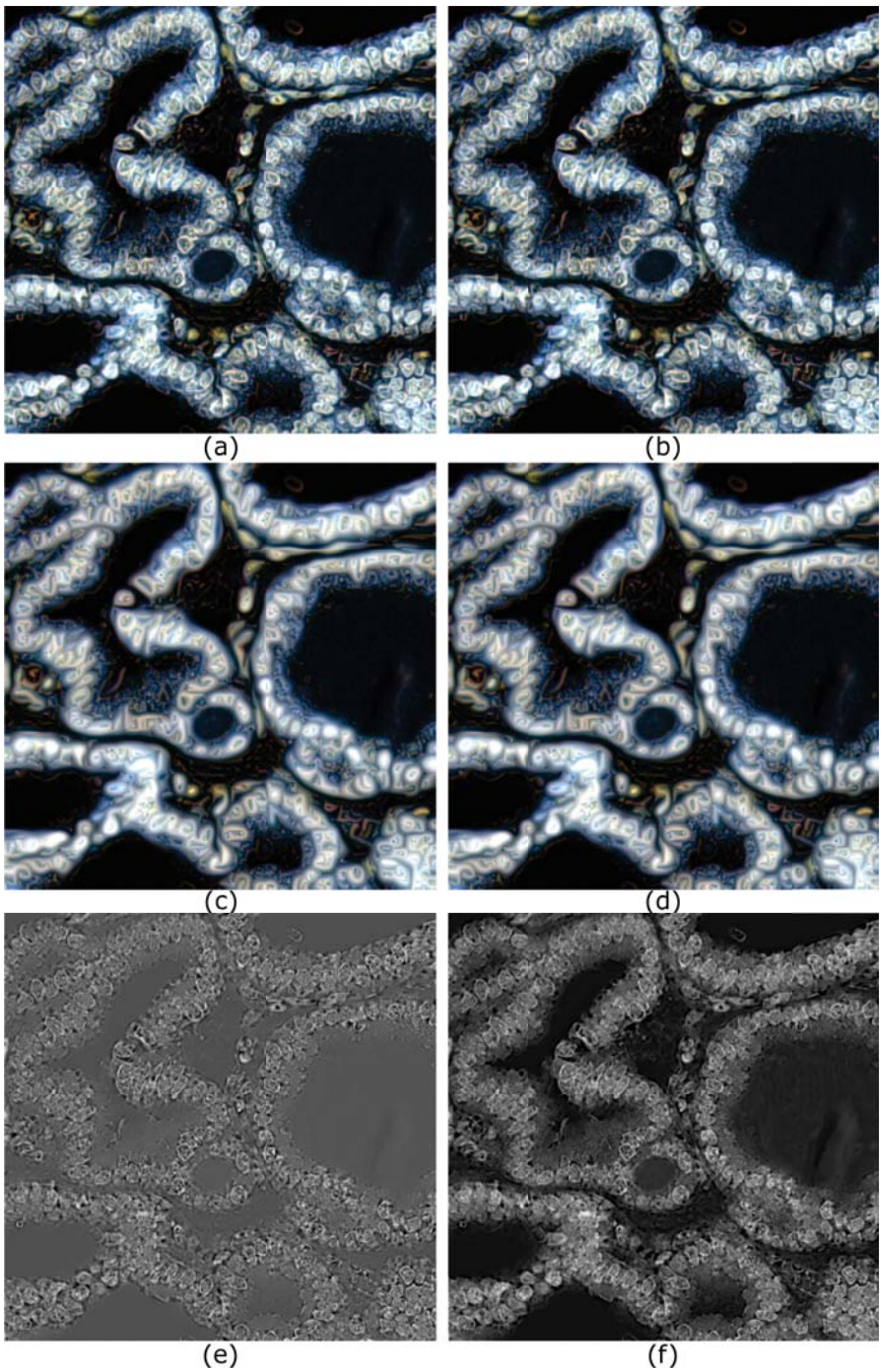

Figure 5.5.9: Time evolution of TAND-CED of the thyroid tissue image. $\sigma = 0.5$, $\rho = 1.5$ $\tau = 1.5$, $\psi = \mu'\text{s}\, 45^{th}$ percentile. (a) CED with classical structure tensor, $n = 1$. (b) CED with heat weighted structure tensor, $n = 1$. (c) CED with classical structure tensor, $n = 30$. (d) CED with heat weighted structure tensor, $n = 30$. e) $\overline{\mathbf{I} - u_{30}}$where u_{30} is obtained by processing $\mathbf{I}$ with TAND-CED using the classical structure tensor and $n = 30$. (f) Same as (e) but using the heat weighted structure tensor.

first row shows results for iteration $n = 1$. The middle row shows results for $n = 30$. Notice that, from $n = 1$ to $n = 30$, the edges of the cells have been thickened and the homogeneous regions show a light smoothing. Subtracting the image resulting from CED after $n = 30$ iterations, u_{30}, from the original image $\mathbf{I}$, produces an image with 224 bands. This image is averaged over all bands, i.e., $\overline{\mathbf{I} - u_{30}}$. Figure 5.5.9(e) shows it using the classical structure tensor and Figure 5.5.9(f) using the heat weighted structure tensor. These images are displayed with the intensity range set to $[\min \overline{\mathbf{I} - u_{30}}, \max \overline{\mathbf{I} - u_{30}}]$. Figure 5.5.9(f) has higher contrast between cell edges and dark regions. The next step is to use a threshold, ϕ, equal to a user defined percentile of the intensity values of $\overline{\mathbf{I} - u_{30}}$, i.e. for $n = 30$,

$$I_{n,\phi} = \overline{\mathbf{I} - u_{n|\phi}}. \tag{5.5.1}$$

The results, using $\phi = 75^{th}$ percentile, are the binary images shown in Figure 5.5.10. Figure 5.5.10(a) shows $I_{30,75}$ using the classical structure tensor while Figure 14(c) shows it using the heat weighted structure tensor. Figure (5.5.10)(b) and (d) show $I_{30,75}$ after eliminating all the connected components with area less than or equal to 15 pixels (called *granules*) using the morphological operator open in the Matlab function `bwareaopen`. Figure 5.5.10(b) uses the classical structure tensor and Figure 5.5.10(d) uses the heat weighted structure tensor. The images in Figure 5.5.10(b) and (d) show a better definition of the individual cells. Comparing the values of $I_{30,75}$ in Figure 5.5.10(a) and (c), it can be seen that $I_{30,75}$ using the heat weighted structure tensor (bottom row) has fewer granules, and the cells look more complete. To quantify these observations, a test was designed in which $I_{30,\phi}$ is calculated with values of ϕ taken every second percentile from the 30^{th} to the 98^{th} percentile. Values of ϕ less than the 30^{th} percentile show too much detail to effectively count the cells. For each value of ϕ, the number of connected components

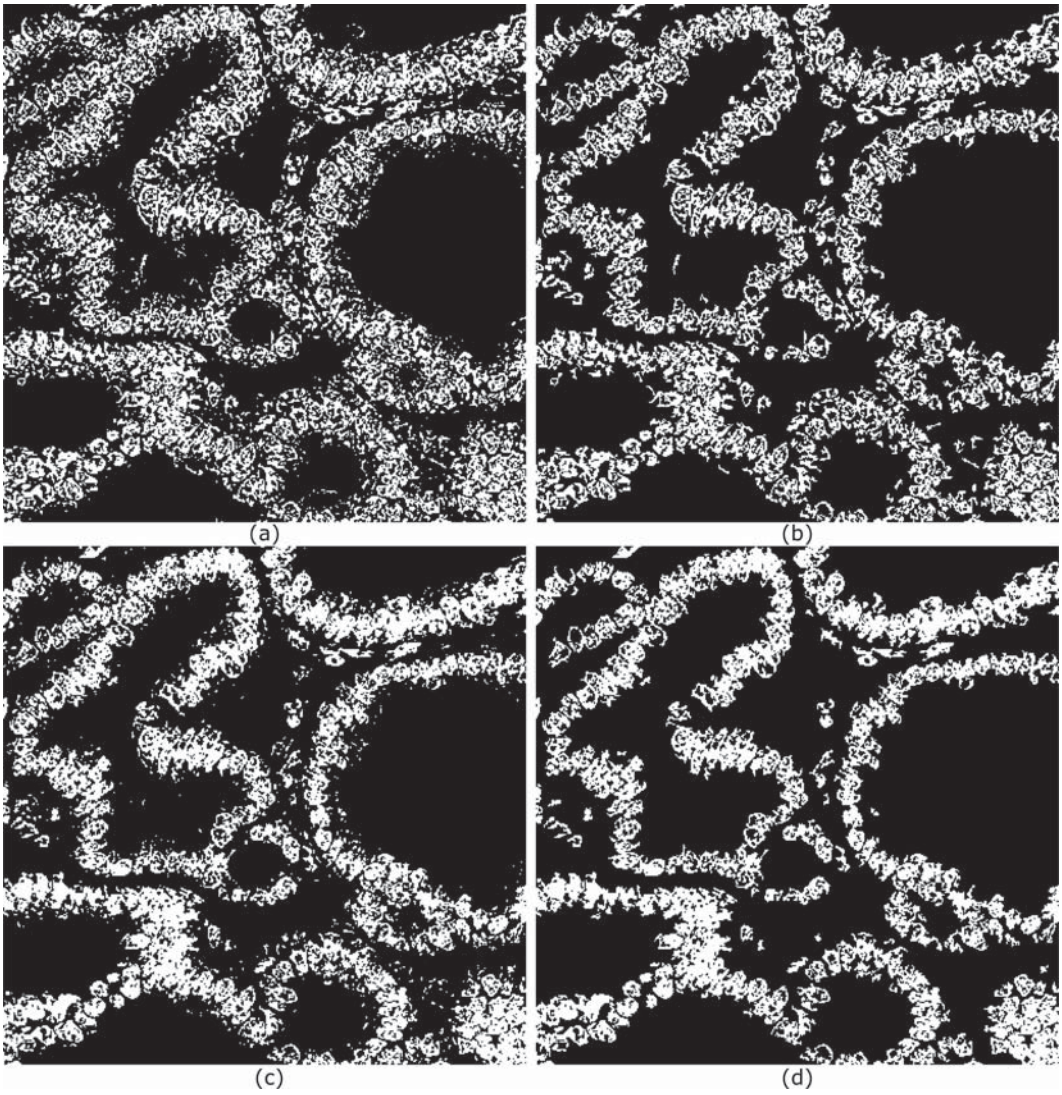

Figure 5.5.10: Edges enhanced by TAND–CED. (a) using the classical structure tensor, $I_{30,75}$ as defined in (5.5.1). (b) Same as (a) after eliminating all granules of area 15 pixels. (c) Same as (a), but using the heat weighted structure tensor. (d) Same as (b), but using the heat weighted structure tensor.

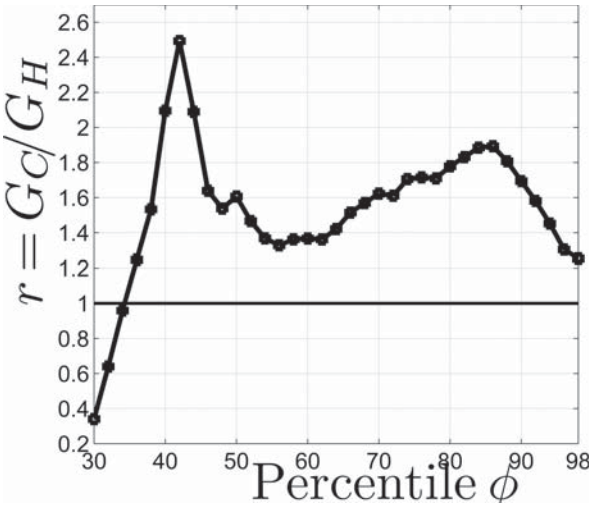

Figure 5.5.11: Comparison of number of granules of $I_{30,\phi}$ with area less than equal to 15 pixels found using the classical structure tensor vs. the heat weighted structure tensor.

with an area of 15 pixels or less is counted. Then a ratio, r between the number of granules found in $I_{30,\phi}$ for the classical structure tensor, G_C, and the number of granules in $I_{30,\phi}$ for the heat weighted structure tensor, G_H, is calculated, i.e.,

$$r = G_C/G_H.$$

If $G_C = G_H$ then $r = 1$; if $G_C > G_H$ then $r > 1$, and if $G_C < G_H$ then $0 < r < 1$. The results are shown in Figure 5.5.11.

After the 36^{th} percentile, the number of granules for the classical structure tensor is higher than the number of granules for the heat weighted structure tensor. At the 42^{nd} percentile, the one based in the classical structure tensor has almost 2.5 times the number of granules than the proposed method. This decreases to 1.4 times as many at the 60^{th} percentile before growing to almost 1.9 at the 86^{th} percentile. Cell counting is an important application for these images. To that end, ϕ in $[70^{th}, 85^{th}]$ percentile is one of the most critical intervals since $I_{n,\phi}$ will begin to show the cells separating from each other. As the value of ϕ gets higher, many cells will look incomplete and the resulting images are less critical.

From Figure 5.5.11, it can be seen that between the 70^{th} and 85^{th} percentile, $I_{30,\phi}$ for the classical structure tensor has between 1.6 to 1.9 times the number of granules of the proposed method. Therefore, weighting the structure tensor using the weights based on the heat operator helps TAND to make a better distinction between the cells edges and their surrounding features.

5.6 COMPARISON BETWEEN THE SPECTRALLY ADAPTED AND WEICKERT'S SPATIAL TAND.

In the previous section, comparisons were done preserving the proposed TAND and only changing the structure tensor. The idea was to study the performance of the proposed structure tensor compared with the classical one. In this section, TAND with the spectrally adapted structure tensor and the proposed diffusion tensors will be compared to the state of the art TAND proposed by Weickert [2].

5.6.1 *TAND for Edge Enhancing Diffusion*

A time step $\tau = 5$ and number of iteration, $n = 4$ will be used for EED. That is equivalent to 80 iterations of the explicit scheme which is empirically stable for time step size of at most $\tau = 0.25$ [46]. Figure 5.6.1 shows the results of this comparison. Figure 5.6.1(b), (d), (e) and (f) show results from Weickert's TAND and Figure 5.6.1(c), (g), (h) and (i) show the proposed TAND. Weickert's TAND shows a strong smoothing in the whole image preserving only the strongest edges. The edges look blurry and the small features are hardly visible. In contrast, the proposed TAND the edges are not blurred and the smoothing in the inside regions is visually comparable to Weickert's. The small features are still visible and remind with the same color composition than in the original image. A shock filter to

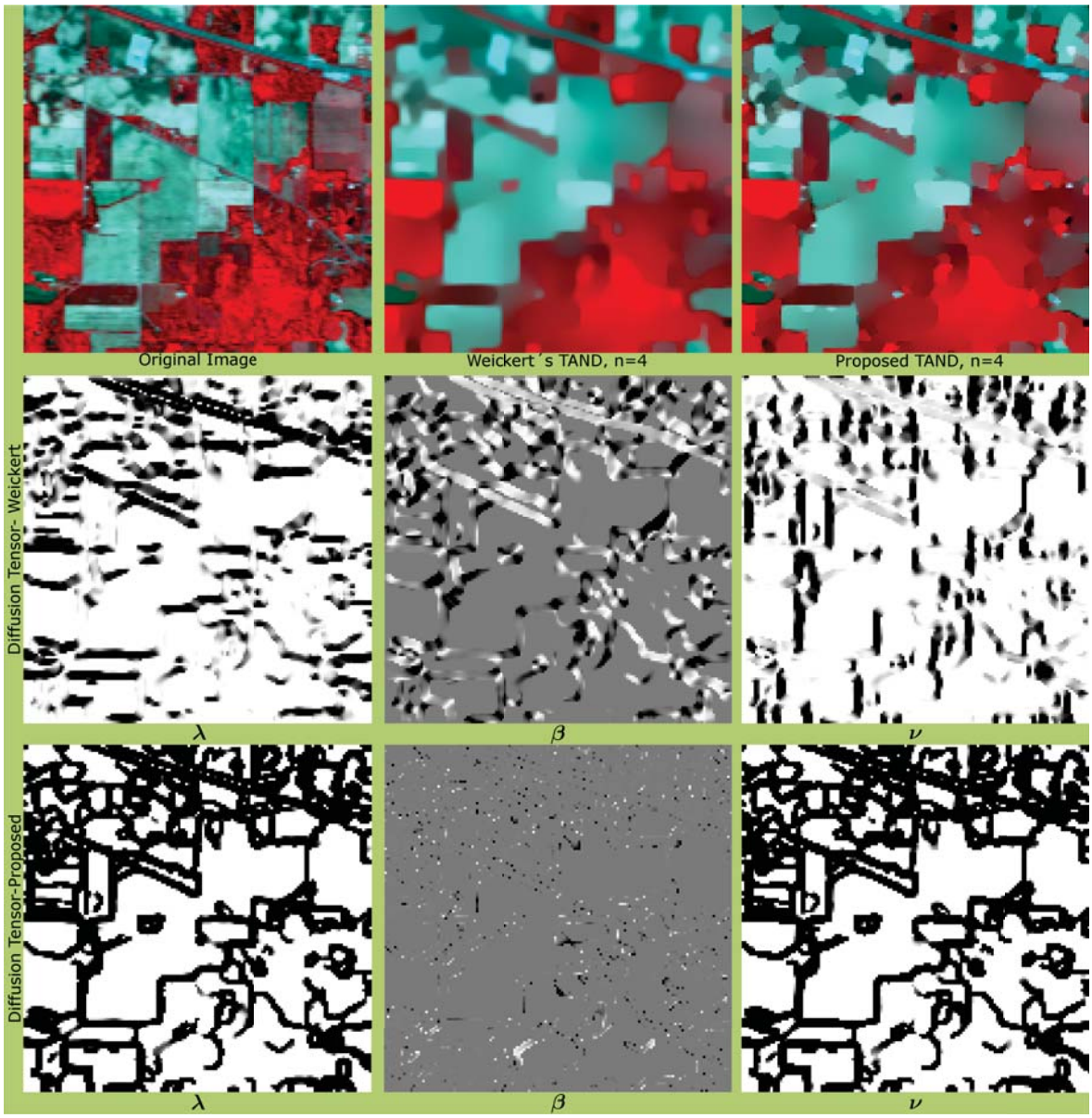

Figure 5.6.1: Comparison of the regularization from proposed TAND and Weickert's TAND. $\sigma = 0.8$, $\tau = 5$, $\psi = \mu_1'$s 55^{th} percentile, $n = 4$. (a) RGB Indian Pines image (b) after Weickert's TAND (c) after proposed TAND; (d), (e) and (f) are λ, β, and ν components of Weickert diffusion tensor. (g), (h) and (i) are the corresponding components of the proposed diffusion tensor.

unsharp the edges is usually used after diffusion . For the case of the proposed TAND, this is not need it. This difference is explained with the components of their respective diffusion tensors. Figure 5.6.1(e) and (h) shows the $\beta-$ component of both methods. It is clear that, for Weickert, this component plays an important role. While in the case of the proposed TAND does not. According to the analysis of the divergence-based PDE diffusion carry out by [3] β produce an inter channel coupling which is reduced in the proposed TAND. λ and $\nu-$ components of the proposed TAND looks very similar but their values for the horizontal and vertical edges are different. So depending in the application, preserving the edges can help to obtain better results for the next processing in the image. Comparison with Figure 5.6.2 and Figure 5.6.3 show similar conclusions than the ones with Figure 5.6.1.

5.6.2 *TAND for Coherence Enhancing Diffusion*

The thyroid cell tissue will be used for the comparisons for CED. The parameters used are: time step size $\tau = 5$, $\sigma = 0.5$, $\rho = 2.5$ and $n = 30$. Figure 5.6.4 shows the thyroid tissue being processed with Weickert's TAND and the proposed TAND. It is clear the the proposed TAND produce less enhancement of the area around the cell chain and produce less granules as shown in Figure 5.6.5.

Figure 5.6.2: Comparison of the proposed TAND-EED with Weickert's TAND-EED applied to Cuprite image. (a) $n = 1$ using the proposed TAND; (b) $n = 5$ using the the proposed TAND-EED; (c) $n = 15$ using the proposed TAND; (d), (e), and (f) are the same as (a), (b), and (c) respectively but using the Weickert's TAND.

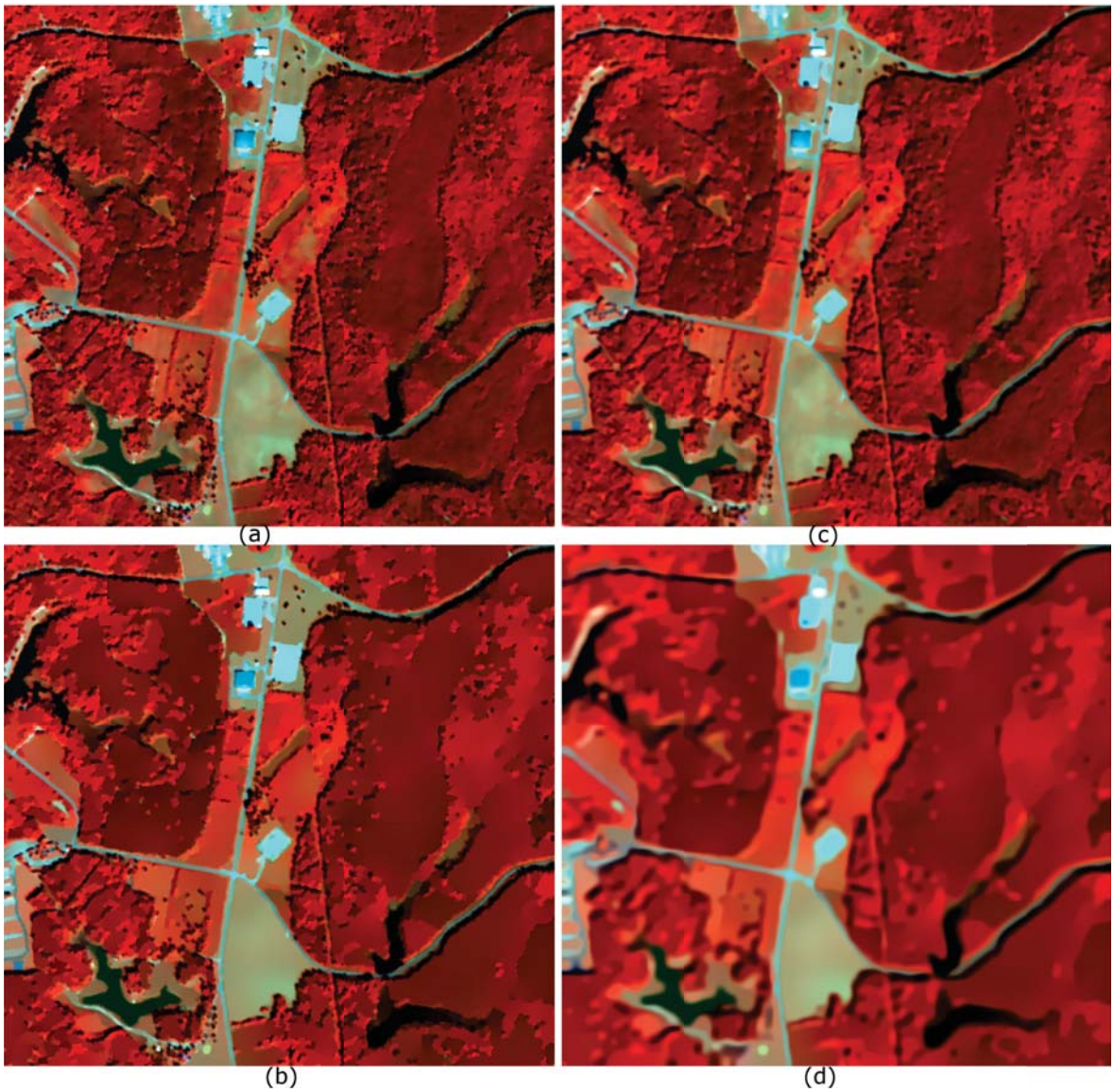

Figure 5.6.3: Comparison of the proposed TAND-EED with Weickert's TAND-EED applied to A. P. Hill image. Both methods uses $\sigma = 1.5$, $\tau = 5$, $\psi = \mu_1'$s 45 percentile. (a) $n = 1$ using the proposed TAND; (b) $n = 15$ using the the proposed TAND-EED; (c), and (d) same as (a) and (b) respectively but using the Weickert's TAND.

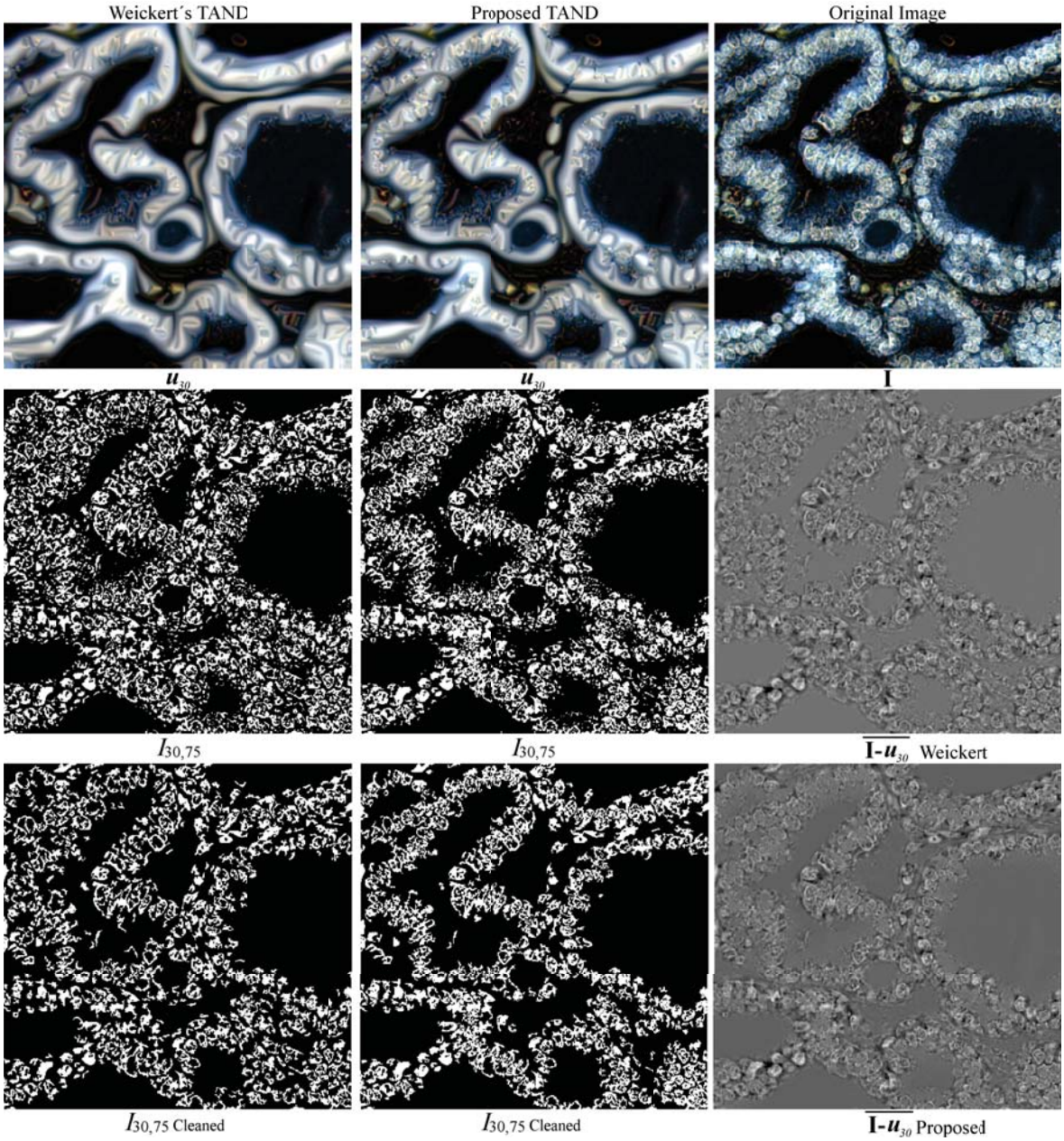

Figure 5.6.4: Comparison of cell chains obtained with the spectrally adapted TAND and Weickert's TAND. $\tau = 5$, $\sigma = 0.5$, $\rho = 2.5$ and $n = 30$. (a) $I_{30,75}$ using the classical structure tensor, as defined in (5.5.1). (b) Same as (a), after eliminating all granules of area 15 pixels. (c) $\overline{\mathbf{I} - u_{30}}$ with Weickert's TAND-CED (d) $I_{30,75}$ using the heat weighted structure tensor. (e) Same as (d) after eliminating all granules of area 15 pixels. (f) $\overline{\mathbf{I} - u_{30}}$ with proposed TAND-CED, (g) Weickert's TAND-CED (h) Proposed TAND-CED (i) Original Image

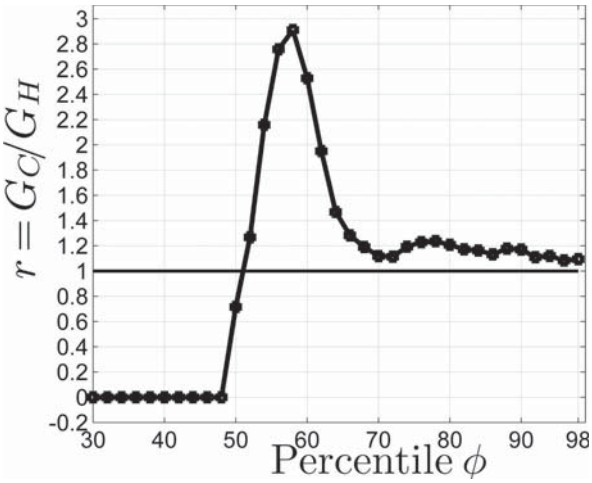

Figure 5.6.5: Comparison of number of granules of $I_{30,\phi}$ with area less than equal to 15 pixels found using the classical structure tensor vs. the heat weighted structure tensor.

5.7 CONCLUDING REMARKS

The experiments in this chapter show that defining the weights with $s = 1$ produces more stable results for CED than variying s with the number iterations, n, of TAND. They also show that using an anisotropic diffusion based on a tensor helps the diffusion process to discriminate and ultimately preserve low contrast edges. Comparing the ST using TAND show that the heat weighted structure tensor helps the diffusion tensor TAND-EED to better discriminate which edges to keep longer as TAND-EED iterate. This result was achieved by comparing the components of the diffusion tensor of TAND-EED produced using the proposed heat weighted and the classical structure tensor. The proposed ST also helps TAND-CED to produce less broken edges and to obtain a better structure enhancement than using the classical structure tensor. From the comparison between the proposed TAND with Weickert's TAND, it can be concluded that the proposed ST produce better regularization results since the edges are not blurred while the inner regions are strongly smoothed but preserving the small features in the images. Comparing

both TAND-CED can be concluded that using the heat weighted ST applied to the thyroid cell image produces less broken edges and the chain of cells can be better discriminated than using Weickert's TAND-CED.

6

ON PUBLICLY AVAILABLE REMOTE SENSING IMAGERY

The work presented in this thesis belongs to the Hyperspectral Remote Sensing field. The discussion in this chapter will be expanded to include aerial photography and satellite imaging in general. This chapter will reflect on and present issues about the impact of publicly available remotely sensed imagery.

A brief overview of the advances in technology and some events that helped reach the availability of remotely sensed data to the public will be presented. It is not by any means complete. It is done with the aim of placing ideas in perspective. The first launch of a non-weather satellite for civilian use occurred in 1972 when NASA put Landsat into orbit to monitor the planet's land masses, tracking everything from desertification to changes in agriculture. During the 1980s, hyperspectral remote sensors were developed. Satellites with the capability of acquiring HSIs were launched in the subsequent decades, bringing more analytic capabilities to the data collected. At the same time geospatial data began to be developed but it was not until the mid-2000s that this technology began have its greatest impact on our society. Geospatial data is the combination of several technologies such as remote sensed data, GIS and global positioning systems (GPS). Geospatial data add high positional accuracy capabilities to the remotely sensed data, in that way integrating separate or loosely related technologies producing a powerful system [110]. Another factor was the the creation of the global information infrastructure, i.e., the development and public access to the Internet for the majority of people on the planet. The internet allows access to information in nearly real time. With the de-

velopment of computer hardware, cheap computer memories and wide broadband internet access, now it is possible to download a high volume of data in several minutes.

Fourteen years ago, in 1998, several ethical issues were raised by [111] related to the advances in (*i*) information management, data control of remotely sensed imagery, and (*ii*) the development of a digital, global information infrastructure, such as the Internet. There were concerns about the usage of remotely sensed data given the general evolution of the remote sensing infrastructure from government to commerce and from domestic to international. This paper also warned about the lack of comprehensive policy development with respect to high-resolution remote sensing technology. The relaxed policies about remotely sensed data paved the way for Google Earth that was launched in 2005.

This brought about the change from centralized government owned remotely sensed data to freely acquired aerial imagery using Google Earth and from data produced by trained experts and then processed to be used for another set of trained experts to data openly available to anybody who knows how to use the internet. Google Earth uses high spatial resolution data provided by commercial satellite image providers like DigitalGlobe and GeoEye as well as data provided by local municipalities and governmental agencies [112]. Not only Google can acquire remotely sensed imagery, nowdays it is possible to market, sell and buy high spatial and spectral resolution global imagery. The USA, Russian, French, Japanese, Indian, Chinise-Brazilian, Israel and Canadian governments, and even non-goverment parties sell this kind of imagery acquired by their orbiting imaging system over the Internet [110, 113]. According to a United Nation resolution for remotely sensed data [114], data rights, pricing, and distribution policies reside with the governments owning the spacecraft.

All that gives rise to issues about privacy. Those images are acquired without taking into account the owner of the land being sensed or any wishes for privacy that a person or institution might have. Of course, if it were necessary to have the written permission of every owner of a piece of land in the Earth, Remote Sensing would not exist. Despite the fact that global remote sensing requires some form of global rules, the addition of profitable businesses ruled by market variables and civilian and military interests make it very difficult to have meaningful rules governing it [113]. So, now it is possible to have access to high spatial resolution photographs of military installations (national or foreign) and study its development through the years [115]. It is also possible to acquire high spatial and spectral HSI images detecting hidden precious metal deposits and to then sell them to the highest bidder without the owner of the land even knowing it. This is somewhat extreme as you normally need some idea of where to look but it is possible. Definitely it is possible to open Google Earth in any random place in the US click Street View and maybe within five minutes or so you can see the state of the neighborhood and then send advertisements of your products. It is also possible to find a picture of a car clearly showing its license plate in the driveway of a random house. All the before mentioned factors have contributed to us living in a surveillance society. For example questions, such as: Can the IRS use those images to look for unreported property renovation or extension? Can the law enforcement and environmental agencies use these publicly available images to do surveillance without a warrant? The answer is yes at least in the US [111, 113] and Canada [116].

Another interesting issue raised by [115] related to Google Earth and to some extend to all remotely sensed imagery is the idea of a 'global panopticon' in which the viewer of data sees all, but those who are viewed see nothing and, importantly, do not know if and when they are being watched. Furthermore, the majority of those being viewed are not even aware of the mechanism that might be watching

them and correspondingly do not act as if they are being watched. So there are two faces of the coin, the liberating and revelatory aspects of this technology unveiling governments and military to all and in contrast the possibly repressive aspects of a panoptic technology of surveillance.

On a brighter side, public access to remote sensing imagery has allowed some non-environmentally or non-military related projects that are looking for the greater good for humanity. The use of remotely sensed images as evidence of human rights violations, such as the ones in Darfur and Zimbabwe [117]. They have also been used to investigate accusations of human rights violations, such as government sponsored forced relocation, location of massive graves, and to document the extent of conflicts in many parts of the world [118, 119]. HSIs have been used in environmental applications since its development. But now not only satellite imagery is used but high resolution spacial aerial images found freely in Google Earth have also been used to make virtual discoveries. Hundreds of new species in dense and difficult to access forests in different parts of the world have been found using the available imagery, which then led to an in-site discovery. The internet and remotely sensed imagery is also used to create awareness to all that are willing to read about the speed of deforestation on some other part of the planet [120].

6.1 CONCLUDING REMARKS

In conclusion, publicly available remote sensing imagery has issues related to privacy, data management and control of who owns the information. The reality of being sensed from so many feet over our heads and not knowing it produces the idea of a panoptic society in which even the governments cannot hide its secrets places. On the other hand, the fact that everybody who owns a computer and uses internet can have access to it also gives some liberating sense. In general, it is diffi-

cult to control the flow of remotely sensed information since many parties produce it and distribute it. On the other hand, it has been shown that there is more availability of data for the experts and for people who just like to be informed. It is also expected advances in the creation of sensors with more capabilities, now that there is a wider use of remotely sensed technology.

7

CONCLUSIONS AND FUTURE WORK

7.1 CONCLUSIONS

The structure tensor is the backbone in many algorithms used in computer vision to find the spatial local structure in color and gray value images. Its extension to vector valued images with more than 3 bands has been done without taking into account the spectral characteristics of the data. The structure tensor is a useful tool that arises in many frameworks such as statistics, differential geometry, and optimization. This work showed that including the spectral information in the structure tensor makes a difference and it can improve significantly image enhance methods such as TAND.

A spectrally weighted structure tensor framework has been presented. Also, a spectrally weighted structure tensor for the case of vector-value images that locally are highly correlated in the spectral dimension has been proposed. The weights are based on the heat operator acting on each pixel of the gradient. This structure tensor is data-based and it does not introduce new parameters to the model.

Diffusion tensors were proposed and it was shown that they take full advantage of the edge information produced by the spectrally weighted structure tensor. The proposed diffusion tensors were used to design a Tensor Anisotropic Nonlinear Diffusion (TAND) for Edge Enhancing Diffusion (EED) and for Coherence Enhancing Diffusion (CED). TAND-EED was used to regularize HSIs, while TAND-CED was used to do structure enhancement. Experiments with HSI of different spatial and

spectral resolutions were used. The images were taken using sensors in airborne, microscope, and in-situ hyperspectral camera. The experiments compare the proposed TAND using the heat weighted structure tensor with the proposed TAND using the classical structure tensor.

The experiments performed in this work showed that the proposed weighted structure tensor enhanced the ability of the diffusion tensor of TAND–EED to discriminate edges that must persist longer through the iterations while discarding edges that are less important. To the best to my knowledge, TAND–CED has not been used before to do structure enhancement of HSIs. Neither is used to do structure enhancement of HSI as treated in this work. TAND–CED using the heat weighted structure tensor produced less broken edges than the one using the classical structure tensor. In this case, two different problems from different fields were treated. One was extracting the chain of cells in a thyroid tissue image and the other was finding the wakes of a boat in an open ocean image. This was achieved by averaging along the spectral direction the image of the difference between the original image and the TAND-CED processed image at some iteration n. Counting the number of connected components of small area of the resulting image was used as a measure of how broken the edges were. Then the ratio, r, between the numbers of granules found using the classical ST over the number of granules found using the heat weighted one was used to assess performance. The graph of r vs. ϕ (the threshold for the intensity values of the averaged image) in both problems showed that TAND-CED using the heat weighted ST produces less broken edges.

About the convergence of the nine diagonal linear system it was found that ILU(0) was enough to accelerate the convergence of the solution.

7.2 FUTURE WORK

The spectrally adapted structure tensor presented in Chapter 3 is designed for vector-valued images that are locally highly correlated, such as HSI. Finding a way to spectrally adapt other kind of vector-valued images, such as MSI, that do not fully or partially fulfill this condition is important, so TAND can be adapted to them. The spectrally adapted structure tensor can be studied also for 3D images, i.e., volumetric images, such as MRI medical images.

Finding a tensor for TAND that produce less or none rounding effect in the corners is another topic to study. This effect is due to the fact that the structure tensor for EED does not distinguish between right and left directions. This is a known problem, it have been studied by Steidl and Teuber for gray value images of shear and rotated rectangles [121]. In this case, they modify the eigenvectors of the structure tensor to make them sensible to directions, and not only orientations. A diffusion tensor that mitigates this problem for images with shear and rotated rectangles is also proposed. So, an study of an spectrally adapted structure tensor that it is sensible to directions to try the correct the rounding of the corners can also be done. Another direction of study is the usage of structure tensor-based corners detection methods and used it to correct the lack of detection of the corners by the structure tensor itself.

From the implementation of the structure tensor, it is clear that all the operations are done pixel-wise. To calculate the structure tensor per pixel, it is only necessary the pixel and its 3×3 neighborhood around itself. This produces an embarrassing parallel scenario. So a parallel implementation can be done. TAND on the other hand has the same characteristics until the step in which the linear system need to be solved. There are parallel implementations of BiCGStab for distributed memory architectures. In [122], a parallel implementation using Compute Unified Device

Architecture (CUDA) in Graphics Processing Unit (GPU) have been proposed for solving finite difference linear systems with explicit preconditioned Biconjugate Conjugate Gradient type methods . So a parallel implementation of the structure tensor and TAND can be achieved.

Studying the effect of the disturbance term presented in Section 4.5 thru the iterations and its real impact in the divergence-based diffusion can be another study direction. A study of advantage and disadvantage of using TAND-EED versus using the regularization obtained using the trace-base diffusion operator presented by [123, 3] for HSI can be useful to study in which situation is better to use either method.

APPENDIX

A

SELECTED TOPICS

A.1 SOME DEFINITIONS FROM LINEAR ALGEBRA AND NUMERICAL ANALYSIS

The definitions in this section are based on the ones appearing in [102].

A.1.1 *Rank of a Matrix*

The rank of a matrix A is defined as the dimension of the subspace for which $\mathbf{T}v \neq 0$ and denoted by *rank*. The space for which $\mathbf{T}v = 0$ is known as the *null space*. The dimension of the null space is known as *nullity*, denoted by *null*. One of the most important theorems in linear algebra, known as *dimension theorem* establish that:

$$rank(A) + null(A) = d$$

where d is the number of columns of A, which in the case of symmetric matrices is the same as its number of rows.

A.1.2 *Sparse and dense Matrices*

A *sparse matrix* is a matrix in which the majority of entries are zero. In contrast, a *dense matrix* is one in which the majority of the entrances are non-zero. A *structured sparse matrix* is one whose nonzero entries form a regular pattern, often along a

small number of diagonals. A matrix with irregularly located entries is said to be *irregularly structured*. One way to produce sparse matrices is from the discretization of a PDE or an Ordinary differential equation (ODE) to fit the continuous theory into the discrete data.

A.1.3 *Projection Methods to solve large linear systems*

Given a linear system

$$Ax = b \tag{A.1.1}$$

where A is a $k \times k$ real matrix. A can also denote the linear mapping in $\mathbb{R}^k$ that it represents. b is a vector with the same number of rows as A, and x is the solution vector to be found when solving the linear system (A.1.1) . The idea of *projection methods* is to extract an approximate solution to the above problem from a subspace $\mathcal{K}$ of $\mathbb{R}^k$.

The *Petrov-Galerkin conditions* are m (independent) orthogonality conditions imposed over the the search subspace $\mathcal{K}$ of dimension m so an approximate solution of (A.1.1) can be found in $\mathcal{K}$. Specifically, the residual vector $b - Ax$ is constrained to be orthogonal to m linearly independent vectors. This defines another subspace $\mathcal{L}$ of dimension m which will be called the *subspace of constraints* . This is,

$$\text{Find } \tilde{x} \in \mathcal{K}, \text{ such that } b - A\tilde{x} \perp \mathcal{L}$$

when an initial guess x_0 to the solution exists then the approximation must sought in the affine space $x_0 + \mathcal{K}$ instead of the homogeneous vector space $\mathcal{K}$. Then the problem becomes

$$\text{Find } \tilde{x} \in x_0 + \mathcal{K}, \text{ such that } b - A\tilde{x} \perp \mathcal{L}.$$

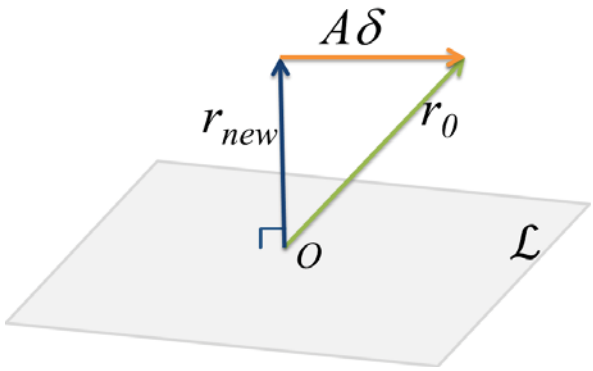

Figure A.1.1: Interpretation of the orthogonality condition. Taken from [102]

$\tilde{x} \in x_0 + \mathcal{K}$ means that $\tilde{x}$ can be written as $\tilde{x} = x_0 + \delta$, for some $\delta \in \mathcal{K}$. Let the residual vector r_0 be defined as :

$$r_0 = b - Ax_0, \tag{A.1.2}$$

then

$$b - A\left(x_0 + \delta\right) \perp \mathcal{L} \quad \text{or} \quad r_0 - A\delta \perp \mathcal{L}$$

then by definition of orthogonality the approximate solution can be defined as:

$$\tilde{x} = x_0 + \delta, \text{ for } \delta \in \mathcal{K}, \text{ such that } \langle r_0 - A\delta, w\rangle = 0, \ \forall w \in \mathcal{L}$$

where $\langle \cdot, \cdot \rangle$ is the inner product defined in $\mathbb{R}^m$.

Many of the standard techniques use a succession of these projections. Typically, a new projection step uses a new pair of subspaces $\mathcal{K}$ and $\mathcal{L}$ and an initial guess x_0 equal to the most recent approximation obtained from the previous projection step [102]. When $\mathcal{K} = \mathcal{L}$ then the projection method is known as *orthogonal*, otherwise *oblique*. This distinction is rather important and gives rise to different types of algorithms.

A set of methods known as *Krylov subspace methods* defines the search subspace $\mathcal{K}$ of dimension m as

$$\mathcal{K}_m(A, r_0) = span\left\{r_0,\ Ar_0,\ A^2r_0, ..., A^{m-1}r_0\right\}$$

where r_0 is defined as in (A.1.2). The different Krylov methods distinguish one from another by the definition of $\mathcal{L}_m$ and the use of preconditioners.

For completeness in the presentation the three most used Krylov methods used for non-symmetric linear systems will be presented in the next subsections all the algorithms are taken from [102].

A.1.3.1 *BiCG*

Bi-conjugated gradient, BiCG, is a Krylov method that solves non-symmetric linear systems by defining a projection process onto

$$\mathcal{K}_m = span\left\{v_1,\ Av_1,\ A^2v_1, ..., A^{m-1}v_1\right\}$$

orthogonal to

$$\mathcal{L}_m = span\left\{w_1,\ A^Tw_1,\ \left(A^T\right)^2 w_1, ..., \left(A^T\right)^{m-1} v_w\right\}$$

with $v_1 = r_0/\left\|r_0\right\|_2$ and w_1 an arbitrary vector that fulfills $\langle v_1, w_1\rangle \neq 0$. $w_1 = v_1$ is often chosen. If there is a dual system $A^Tx^* = b^*$ to be solved with A^T then $v_1 = r_0^*/\left\|r_0^*\right\|_2$ where $r_0^* = b^* - A^Tx_0^*$. The algorithms is presented in Algorithm A.1 [102]. If a dual system with A^T is being solved then $r_0^* = b^* - A^Tx_0^*$ and the update $x_{j+1}^* := x_j^* + \alpha_j p_j^*$ to the dual approximate solution must be added after line 5 in Algorithm A.1. There is a variant of this algorithm that avoid the products with the transpose matrix which is a more efficient computationally.

Algorithm A.1 Bi-Conjugate Gradient (BiCG)

1. Compute $r_0 = b - Ax_0$. Choose r_0^* such that $\langle r_0, r_0^* \rangle \neq 0$.
2. Set, $p_0 = r_0$, $p_0^* = r_0^*$
3. For $j = 0, 1, ...,$ until convergence do
4. $\quad \alpha_j := \frac{\langle r_j, r_j^* \rangle}{\langle Ap_j, p_j^* \rangle}$
5. $\quad x_{j+1} := x_j + \alpha_j p_j$
6. $\quad r_{j+1} := r_j - \alpha_j A p_j$
7. $\quad r_{j+1}^* := r_j^* - \alpha_j A p_j^*$
8. $\quad \beta_j := \frac{\langle r_{j+1}, r_{j+1}^* \rangle}{\langle r_j, r_j^* \rangle}$
9. $\quad p_{j+1} := r_{j+1} + \beta_j p_j$
10. $\quad p_{j+1}^* := r_{j+1}^* + \beta_j p_j^*$
11. EndDo

Algorithm A.2 BiCGStab

1. Compute $r_0 = b - Ax_0$. Choose r_0^* arbitrarily
2. Set, $p_0 = r_0$
3. For $j = 0, 1, ...,$ until convergence do
4. $\quad \alpha_j := \frac{\langle r_j, r_j^* \rangle}{\langle Ap_j, r_0^* \rangle}$
5. $\quad s_j := r_j - \alpha_j A p_j$
6. $\quad \omega_j := \frac{\langle As_j, s_j \rangle}{\langle As_j, As_j \rangle}$
7. $\quad x_{j+1} := x_j + \alpha_j p_j + \omega_j s_j$
8. $\quad r_{j+1} := s_j - \omega_j A s_j$
9. $\quad \beta_j := \frac{\langle r_{j+1}, r_0^* \rangle}{\langle r_j, r_0^* \rangle} \times \frac{\alpha_j}{\omega_j}$
10. $\quad p_{j+1} := r_{j+1} + \beta_j \left(p_j - \omega_j A p_j \right)$
11. EndDo

A.1.3.2 *BiCGStab*

Bi-Conjugate gradient method was developed to avoid the usage of the transpose of A in BiCG and to gain faster convergence. This method check for convergence twice in one iteration, so the method can converge in the middle of an iteration. The algorithm is presented in Algorithm (A.2)

A.1.3.3 *GMRES*

The Generalized Minimum Residual Method (GMRES) is a projection method based on taking $\mathcal{K} = \mathcal{K}_m$ and $\mathcal{L} = A\mathcal{K}_m$ in which $\mathcal{K}_m$ is the $m - th$ Krylov subspace with

Algorithm A.3 GMRES

1. Compute $r_0 = b - Ax_0$. $\beta := \|r_0\|_2$, and $v_1 = r_0/\beta$
2. For $j = 0, 1, ..., m$ Do:
3. Compute $w_j := Av_j$
4. For $i = 0, 1, ..., j$ Do:
5. $h_{ij} := \langle w_j, v_i \rangle$
6. $w_j := w_j - h_{ij} v_i$
7. EndDo
8. $h_{j+1} := \|w_j\|_2$. If $h_{j+1,j} = 0$ set $m := j$ and go to 11
9. $v_{j+1} = w_j / h_{j+1,j}$
10. EndDo
11. Define the $(m+1) \times m$ Hessenberg matrix $\bar{H}_m = \{h_{ij}\}_{1 \leq i \leq m+1, 1 \leq j \leq m}$.
12. Compute y_m the minimizer of $\|\beta e_1 - \bar{H}_m y\|_2$ and $x_m = x_0 + V_m y_m$.

$v_1 = r_0 / \|r_0\|_2$. This technique minimizes the residual norm over all vectors in $x_0 + \mathcal{K}_m$. Any vector in $x_0 + \mathcal{K}_m$ can be written as

$$x = x_0 + V_m y,$$

where y is an $m-$vector.The simplest version of this algorithm is presented in Table (A.3) [102].

B

A PRECONDITIONER FOR SOLVING THE FINITE DIFFERENCE DISCRETIZATION OF PDES APPLIED TO IMAGES

B.1 LINEAR SYSTEMS AND PRECONDITIONERS USED TO SOLVE THEM

Semi-implicit schemes are the most common methods used in PDE-based image processing [2, 5, 4]. These schemes produce a linear system that need to be solved in each iteration. Depending on the discretization those system can suffer from slow convergence. The mathematical guarantees are, at best, that the solution can be found in linear time that depend on the number of entries in the image. For high dimensional data such as HSI, this can be really slow. Therefore the most common practice to accelerate convergence is preconditioning the coefficient matrix. *Preconditioning* are a set of methods that find equivalent linear systems with the same solution that the original linear system. This is done by transforming the original linear system into one that has the same solution, but it is likely to be easier to solve with an iterative solver [102].

For example, the system $Ax = b$ can be transformed by :

$$M^{-1}Ax = M^{-1}b \tag{B.1.1}$$

or

$$AM^{-1}u = bM^{-1} \tag{B.1.2}$$

where M^{-1} can be obtained using some analytical method or the solution of a subsidiary linear system $u = Mx$. (B.1.1) is known as *left-preconditioner* and (B.1.2) as *right-preconditioner.* Usually explicit computation of M and $M^{-1}A$ will be avoided. Instead the iterative processes operate with A and M^{-1} whenever it is necessary.

The most simple explicit preconditioner technique is given by the *Jacobi or diagonal preconditioner.* This preconditioner consist in setting the preconditioner $M = diag(A)$, assuming that $A_{ii} \neq 0$ for all i. The system is solved by a Krylov method. The standard preconditioner is the one obtained by the *incomplete factorization* of the original matrix A. The coefficient matrix A is factored into the product of the lower triangular and upper triangular parts of A denoted respectively as L and U, and the residual or error of the factorization R is subtracted, that is:

$$A = LU - R$$

this factorization is known as ILU(0). It is rather popular since it is easy and inexpensive to compute. But often the approximation can be so crude that the Krylov method used will require more iterations to converge [102]. More sophisticated ILU factorization will require a higher computational cost but it makes the Krylov method converge with fewer iterations.

When the coefficient matrix A is symmetric and positive definite it is always possible to find a lower triangular matrix L such that $A = LL^T$, this is known as the Cholesky factorization . An *incomplete Cholesky factorization* finds a sparse lower triangular matrix K that is an approximation to L in some sense. Then a preconditioner M can be defined as:

$$M = KK^T$$

K can be found by finding the exact Cholesky decomposition except that any entry is set to zero if the corresponding entry in A is also zero.

Several standard preconditioners have been used to process HSI, such as Incomplete Cholesky, for the divergence-based diffusion PDE using a diffusivity function [5], and the Incomplete LU factorization (ILU(0)), for the trace-based diffusion PDE using oriented Gaussian masks [124].

B.1.1 *Properties of the Linear System arising from TAND.*

TAND discretized with a 3×3 spatial neighborhood and using semi-implicit methods produces a nine diagonal coefficient matrix. This is, it produces a structured sparse coefficient matrix as shown in Figure B.1.1(b). Let I be the image processed with TAND with size $n_1 \times n_2 \times m$. Then the coefficient matrix in Figure B.1.1(b) arise from the calculation of the 3×3 averaging mask $A\left(U_p^n\right)$ at pixel $P \in \mathbb{R}^m$ as shown in Figure B.1.1(a). The averaging mask at each pixel $P = (i, j, 1 : m)$ only has 9 nonzero entries. This mask is vectorized and becomes the row $k = i + n_1(j-1)$. After the averaging mask is calculated for each image pixel, the $N \times N$ coefficient matrix $\mathbf{A}$ is created, where $N = n_1 \cdot n_2$. Since the diffusion tensor is the same for all the bands, the linear system can be seen in two ways :

1. A linear system in which the right hand side (RHS) is a matrix, that is

$$A\mathbf{U}^{n+1} = \mathbf{B}$$

where each column of $\mathbf{B}$ is a band of the image $\mathbf{U}^n$, see Figure B.2.1. In this case, the operations in the Krylov methods are the same. To keep the relations between the bands of a HSI and have the same amount of diffusion in each band, it is important not to solve this system as a multiple Right

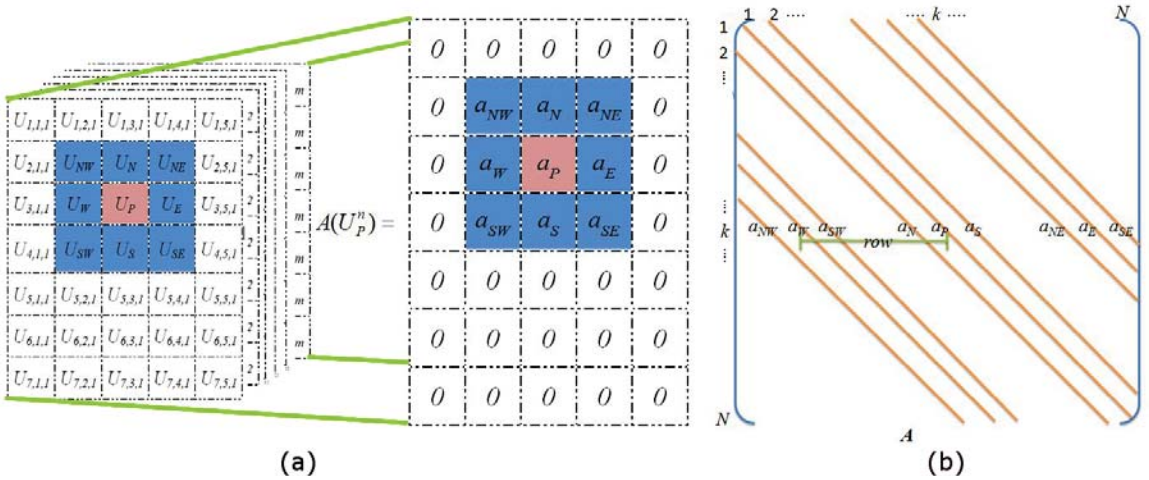

Figure B.1.1: Averaging mask and the nine diagonal linear system produced by solving TAND with the finite difference method.

Hand Side (RHS) system. Instead, the residuals of the whole system are calculated by creating a vector of residuals and finding the residual of each band in a loop over the number of bands. The results in this thesis were calculated by modifying Matlab's BiCGStab method in that way. The system is preconditioned by finding a preconditioner for matrix A.

2. It can also be seen as a regular linear system

$$\mathcal{A}\mathbf{U}^{n+1} = \mathbf{b}$$

in which the coefficient matrix $\mathcal{A}$ is a matrix with m blocks in its diagonal, and each block is the A matrix as shown in Figure B.1.1(b). $\mathbf{b}$ is a vector created from the vectorization of $\mathbf{U}^n$. In this case, a preconditioner for the whole system can be constructed by creating a matrix with the same structure as $\mathcal{A}$ but each block is the preconditioner designed for A

Therefore in any of the cases it is only required to find a preconditioner for the matrix A shown in Figure B.1.1(b).

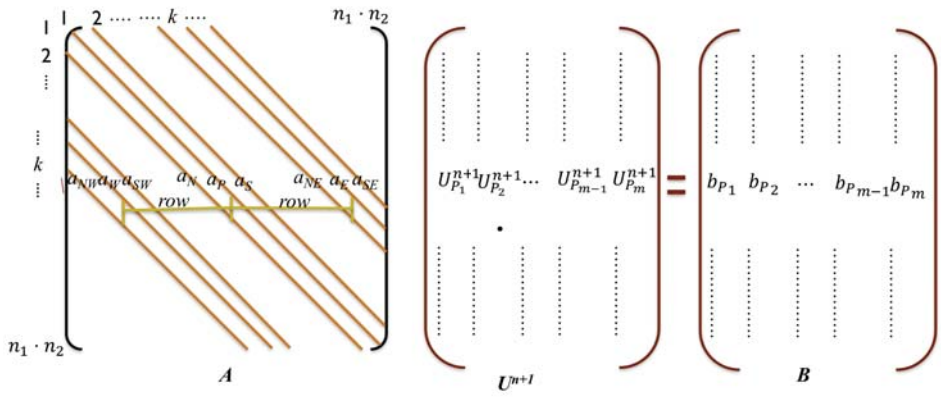

Figure B.2.1: Schematic representation of the nine diagonal linear system resulting from TAND.

B.2 PERIC PRECONDITIONER

Having a priory knowledge of the sparsity pattern of the coefficient matrix is an advantage in order to design preconditioners based on the data. The preconditioner presented in this section can be extended for any of the cases presented in Section B.1.1. Let

$$A\phi = S \tag{B.2.1}$$

be the linear system, where A is a nine diagonal matrix, ϕ is the vector of variables and S is the RHS. The system is depicted in Figure B.2.1.

Since each row for the linear system only has nine values different from zero then the system at row k is:

$$\begin{aligned} a_{nw}\phi_{nw} + a_w\phi_w + a_{sw}\phi_{sw} + a_n\phi_n + a_p\phi_p \\ + a_s\phi_s + a_{nw}\phi_{nw} + a_e\phi_e + a_{se}\phi_{se} = S_p \end{aligned} \tag{B.2.2}$$

Since the sparse pattern of matrix A is from a discretization from a regular grid, a matrix $C = L \cdot U$ with the same diagonals as matrix A can be found. L and U have nonzero coefficients on seven diagonals only, which coincide with the correspond-

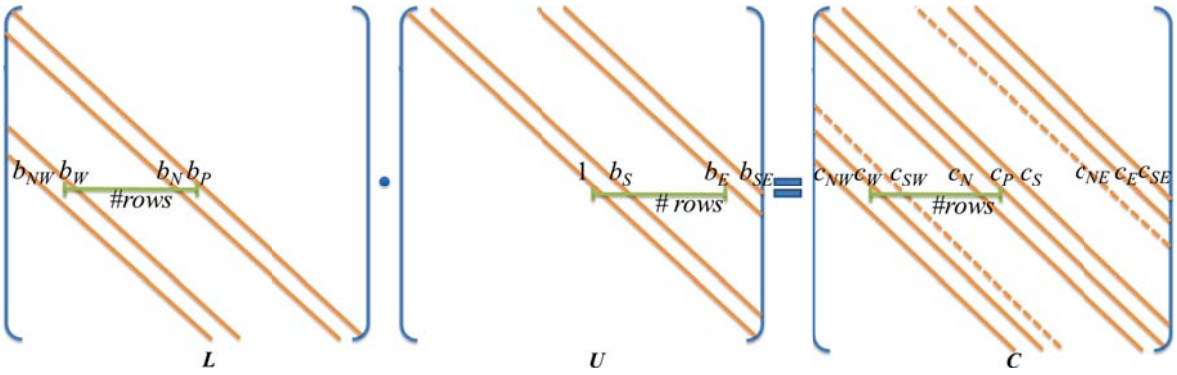

Figure B.2.2: Schematic representation of the proposed matrices L, U and their product matrix C

ing seven nonzero diagonals of A. The two diagonals left out correspond to two opposite corner points a_{NE}, a_{SW} of the averaging mask shown in Figure B.1.1(a), since the column major format is used to arrange ϕ. Otherwise they are a_{NW}, a_{SE}. C with its decomposition is shown in Figure (B.2.1)

The following is the resulting coefficient of the LU product matrix, C, for a row k:

$$\begin{aligned}
c^k_{nw} &= b^k_{nw};\\
c^k_{w} &= b^k_{w} + b^k_{nw} b^{k-r-1}_{s};\\
c^k_{sw} &= b^{k-r}_{s} b^k_{w};\\
c^k_{n} &= b^k_{n} + b^{k-r-1}_{e} b^k_{nw};\\
c^k_{p} &= b^k_{p} + b^{kr}_{e} b^k_{w} + b^k_{n} b^{k-1}_{s} + b^k_{nw} b^{k-r-1}_{se};\\
c^k_{s} &= b^k_{p} b^k_{s} + b^{kr}_{se} b^k_{w};\\
c^k_{ne} &= b^{k-1}_{e} b^k_{n};\\
c^k_{e} &= b^k_{e} b^k_{p} + b^k_{n} b^{k-1}_{se};\\
c^k_{se} &= b^k_{p} b^k_{se};
\end{aligned} \tag{B.2.3}$$

For now on, the superscript k will be dropped and all the calculations will be assumed to be at row k of the corresponding matrix, where $r = n_1$.

Since L and U do not use ϕ_{ne} and ϕ_{sw}, they can be approximated by using the same bands used in their definition on the averaging mask and subtract them from the center value, as follows:

$$\begin{aligned} \phi_{ne} &\approx \alpha \left(\phi_n + \phi_e - \phi_p\right) \\ \phi_{sw} &\approx \alpha \left(\phi_s + \phi_w - \phi_p\right) \end{aligned} \tag{B.2.4}$$

Substituting Equations (B.2.4) in (B.2.2) and after some algebra :

$$\begin{aligned} (a_n + \alpha a_{ne})\phi_n + a_{nw}\phi_{nw} + (a_p - \alpha a_{ne} - \alpha a_{sw})\phi_p + (a_s + \alpha a_{sw})\phi_s \\ + a_{se}\phi_{se} + (a_w + \alpha a_{sw})\phi_w + (a_e + \alpha a_{ne})\phi_e = S_p \end{aligned} \tag{B.2.5}$$

is obtained. This equation produces a linear system

$$\tilde{A}\phi = S \tag{B.2.6}$$

where the coefficient for the k^{th} row is given by:

$$\begin{aligned} \tilde{a}_{nw} &= a_{nw} \\ \tilde{a}_w &= a_w + \alpha a_{sw} \\ \tilde{a}_n &= a_n + \alpha a_{ne} \\ \tilde{a}_p &= a_p - \alpha a_{ne} - \alpha a_{sw} \\ \tilde{a}_s &= a_s + \alpha a_{sw} \\ \tilde{a}_e &= a_e + \alpha a_{ne} \\ \tilde{a}_{se} &= a_{se} \end{aligned}$$

Note that (B.2.6) only have seven diagonals and not nine as $C = L \cdot U$. Assuming that the equations (B.2.4) holds then

$$\phi_{ne} - \alpha\left(\phi_n + \phi_e - \phi_p\right) = \phi_{sw} - \alpha\left(\phi_s + \phi_w - \phi_p\right) \approx 0 \tag{B.2.7}$$

so the following system will be approximately equal to (B.2.6)

$$\begin{aligned} &\tilde{a}_{nw}\phi_{nw} + \tilde{a}_w\phi_w + \tilde{a}_n\phi_n + \tilde{a}_p\phi_p + \tilde{a}_s\phi_s + \tilde{a}_e\phi_e + \tilde{a}_{se}\phi_{se} \\ &+c_{sw}\left[\phi_{sw} - \alpha\left(\phi_s + \phi_w - \phi_p\right)\right] + c_{nw}\left[\phi_{ne} - \alpha\left(\phi_n + \phi_e - \phi_p\right)\right] = S_p \end{aligned} \tag{B.2.8}$$

Replacing the values of $\tilde{a}$'s in (B.2.8) and factoring for the ϕ's, a linear system is obtained. Then, making the corresponding ϕ's coefficients in (B.2.8) equal to corresponding C's coefficients in (B.2.3), results in:

$$\begin{aligned} b_{nw} =& a_{nw} \\ b_w =& \frac{a_w + \alpha a_{sw} - b_{nw} b_s^{k-r-1}}{\alpha b_s^{k-r} + 1} \\ b_n =& \frac{a_n + \alpha a_{ne} - b_e^{k-r-1} b_{nw}}{\alpha b_e^{k-1} + 1} \\ b_p =& a_p - b_n b_s^{k-1} - b_e^{k-r} b_w - b_{nw} b_{se}^{k-r-1} \\ &+ \alpha(b_e^{k-1} b_n - a_{sw} - a_{ne} + b_s^{k-r} b_w) \\ b_s =& \frac{a_s + \alpha a_{sw} - b_{se}^{k-r} b_w - \alpha b_s^{k-r} b_w}{b_p} \\ b_e =& \frac{a_e + \alpha a_{ne} - b_n b_{se}^{k-1} - \alpha b_e^{k-1} b_n}{b_p} \\ b_{se} =& \frac{a_{se}}{b_p} \end{aligned}$$

which will be the coefficients of the diagonals in L and U in (B.2.3). The terms without superscript are assumed at row k. All those calculations were done using Matlab's symbolic toolbox. This preconditioner was inspired by a method [103]

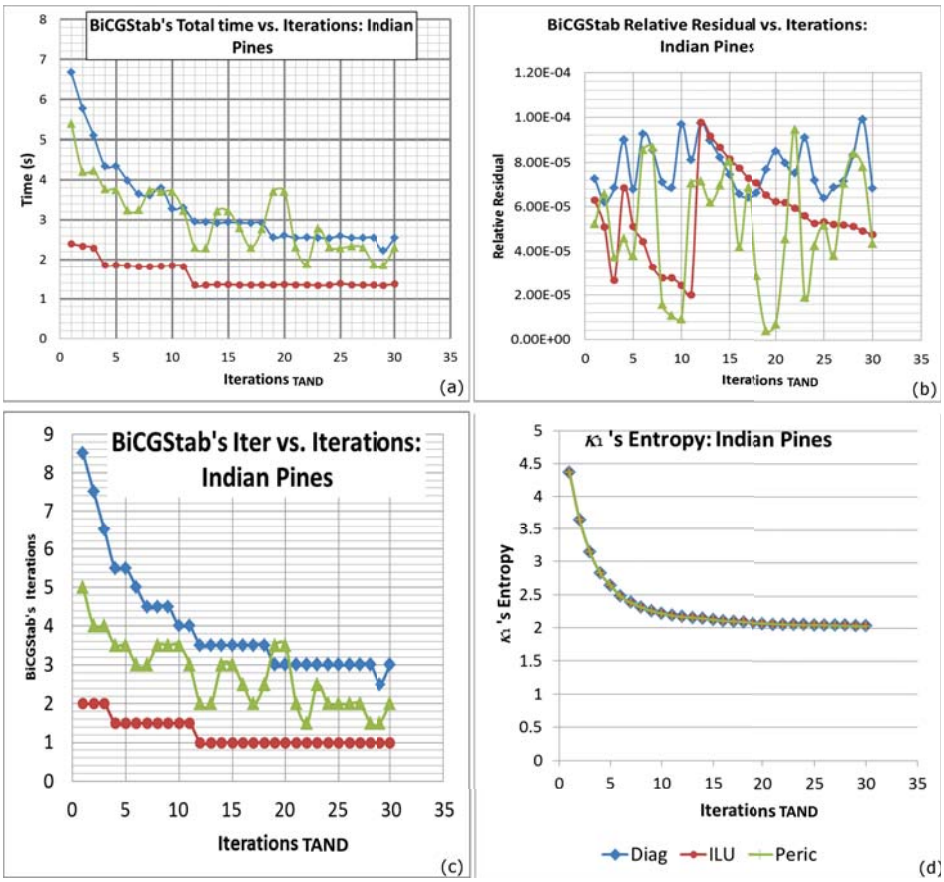

Figure B.3.1: Comparison in terms of (a) total time, (b) relative residual and (c) number of iterations that BiCGStab needs to converge after being preconditioned with ILU(0), Jacobi diagonal and the proposed Peric Preconditioners. The linear system results from appliying TAND-EED to Indian Pines. (d) shows $\kappa_1's$ entropy vs. number of iterations of TAND

proposed to solve the linear system in (B.2.1). But the coefficients for C result from the approximation of (B.2.6), which in turn is the approximation to (B.2.1), and does not inspire much reliability for a direct solver. But could be used as a preconditioner. The results showed in the next section use $\alpha = 0.02$.

B.3 EXPERIMENTAL RESULTS

Figure B.3.1 shows the comparison of preconditioning BiCGStab using ILU(0), Jacobi diagonal and the proposed Peric Preconditioners. Figure B.3.1(a) shows the

total time for a preconditioned BiCGStab to obtain a solution. This time is obtained after adding the time to create the preconditioner and solve the system. As TAND-EED regularize the image the time needed to obtain a solution decrease in all the preconditioners, except in Peric, in some intervals the Peric preconditioner is slowing down the solution. From Figure B.3.1(a) is clear that the quickest preconditioner is the ILU(0). Figure B.3.1(b) shows the results for the relative residual error. BiCGStab was set to stop with a tolerance of 10^{-4}.The different preconditioned BiCGStabs produce a non monotone decreasing sequences of errors as the number of iterations of TAND increases, except ILU that after TAND's iteration 12, its errors decrease. Figure B.3.1(c) shows the number of iterations that BiCGStab needs to converge vs. the number of iterations of TAND. The shapes of the curves for Figure B.3.1(a) and (c) are equal since the time need to solve the system is proportional to the number of iterations that the preconditioned BiCGStab need to converge. Figure B.3.1(d) shows $\kappa_1's$ entropy, This value is almost identical for each of the preconditioned BiCGStab in each iteration of TAND, showing that the resulting images are very similar. $\kappa_1's$ entropy was calculated

BIBLIOGRAPHY

[1] H. Z. M. Shafri, A. Suhaili, and S. Mansor, "The performance of maximum likelihood, spectral angle mapper, neural network and decision tree classifiers in hyperspectral image analysis," *Journal of Computer Science*, vol. 3, no. 6, pp. 419–423, Jun. 2007.

[2] J. Weickert, *Anisotropic Diffusion in Image Processing*. Teubner-Verlag, 1998.

[3] D. Tschumperle and R. Deriche, "Diffusion PDEs on vector-valued images," *IEEE Signal Processing Magazine*, vol. 19, no. 5, pp. 16– 25, Sep. 2002.

[4] J. Martin – Herrero, "Anisotropic diffusion in the hypercube," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 5, pp. 1386–1398, May 2007.

[5] J. M. Duarte-Carvajalino, P. E. Castillo, and M. Velez-Reyes, "Comparative study of Semi-Implicit schemes for nonlinear diffusion in hyperspectral imagery," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1303 –1314, May 2007.

[6] J. Weickert, "Multiscale texture enhancement," in *Computer Analysis of Images and Patterns*, ser. Lecture Notes in Computer Science, V. Hlavác and R. Šára, Eds. Springer Berlin / Heidelberg, 1995, vol. 970, pp. 230–237.

[7] T. Brox and J. Weickert, "Nonlinear matrix diffusion for optic flow estimation," in *Proceedings of the 24th DAGM Symposium on Pattern Recognition*. London, UK: Springer-Verlag, 2002, pp. 446–453.

[8] W. Fróstner and E. Gülch, "A fast operator for detection and precise location of distinct points, corners and centers of circular features," in *Proceedings of the ISPRS Intercommission Workshop on Fast Processing of Photogrammetric Data*, 1987, pp. 281–305. [Online]. Available: http://www-cse.ucsd.edu/classes/sp02/cse252/foerstner/foerstner.pdf

[9] T. Lindeberg, *Scale-Space Theory in Computer Vision*, 1st ed. Springer, Dec. 1993.

[10] J. Bigün, G. H. Granlund, and J. Wiklund, "Multidimensional orientation estimation with applications to texture analysis and optical flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 775–790, Aug. 1991.

[11] A. Rao and B. Schunck, "Computing oriented texture fields," in *CVGIP: Graphical Models and Image Processing*, vol. 53, San Diego, CA, USA, 1991, pp. 157–185.

[12] D. Tschumperlé, "PDE's based regularization of multivalued images and applications," Ph.D. dissertation, Université de Nice-Sophia Antipolis, Dec. 2002.

[13] T. Preusser and M. Rumpf, "Anisotropic nonlinear diffusion in flow visualization," in *Proceedings Visualization*, Oct. 1999, pp. 325 –539.

[14] H. Scharr and H. Spies, "Accurate optical flow in noisy image sequences using flow adapted anisotropic diffusion," *Signal Processing: Image Communication*, vol. 20, no. 6, pp. 537 – 553, 2005, special Issue on Advanced Aspects of Motion Estimation.

[15] B. Bouda, L. Masmoudi, and D. Aboutajdine, "CVVEFM: cubical voxels and virtual electric field model for edge detection in color images," *Signal*

Processing, vol. 88, no. 4, pp. 905–915, Apr. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.sigpro.2007.10.006

[16] M. Lee and L. Bruce, "Applying cellular automata to hyperspectral edge detection," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Jul. 2010, pp. 2202 –2205.

[17] X. Chen and H. Chen, "A novel color edge detection algorithm in RGB color space," in *IEEE 10th International Conference on Signal Processing (ICSP)*, Oct. 2010, pp. 793 –796.

[18] D. Tschumperlé and R. Deriche, "Vector-valued image regularization with PDEs: a common framework for different applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 506–517, 2005.

[19] D. Tschumperlé, "Fast anisotropic smoothing of Multi-Valued images using Curvature-Preserving PDE's," *International Journal of Computer Vision*, vol. 68, pp. 65–82, 2006, 10.1007/s11263-006-5631-z.

[20] J. Weickert, "Coherence-Enhancing diffusion filtering," *International Journal of Computer Vision*, vol. 31, no. 2-3, pp. 111–127, 1999.

[21] J. Weickert and T. Brox, "Diffusion and regularization of vector- and matrix-valued images," Tech. Rep. preprint no. 58, Tech. Rep., 2002.

[22] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349–366, Feb. 2007.

[23] J. Weickert, "Coherence-enhancing diffusion of colour images," *Image Vision Comput.*, vol. 17, no. 3-4, pp. 201–212, 1999.

[24] O. Drblíková, A. Handlovičová, and K. Mikula, "Error estimates of the finite volume scheme for the nonlinear tensor-driven anisotropic diffusion," *Appl. Numer. Math.*, vol. 59, no. 10, pp. 2548–2570, 2009.

[25] K. O. Riedel, "Corner-preserving anisotropic diffusion and junction detection using the structure tensor," in *DAGM-Symposium*, 1999, pp. 164–171.

[26] B. Burgeth, S. Didas, and J. Weickert, "A general structure tensor concept and coherence-enhancing diffusion filtering for matrix fields," *Visualization and Processing of Tensor Fields*, pp. 305–323, 2009.

[27] J. R. Schott, *Remote Sensing: The Image Chain Approach*. Oxford University Press, May 2007.

[28] R. A. Schowengerdt, *Remote Sensing: Models And Methods for Image Processing*. Academic Press, Aug. 2006.

[29] D. Manolakis, D. Marden, and G. A. Shaw, "Hyperspectral image processing for automatic target detection applications," *Lincoln Laboratory Journal*, vol. 14, no. 1, pp. 79–116, 2003.

[30] R. Resmini, *Dimension Reduction of Hyperspectal Image (HSI) Data*, Jul. 2005, published: IPAM Graduate Summer School.

[31] U. o. C. S. Barbara, *Hawaii's Hyperspectral Remote Sensing as an Environmental Monitoring Tool for Shallow Coral Reef Ecosystems*. [Online]. Available: http://emerald.ucsc.edu/hyperwww/hawaii.html

[32] H. Erives and G. J. Fitzgerald, "Automated registration of hyperspectral images for precision agriculture," *Comput. Electron. Agric.*, vol. 47, no. 2, pp. 103–119, 2005.

[33] B. Datt, T. R. McVicar, T. G. Van Niel, D. L. Jupp, and J. S. Pearlman, "Preprocessing EO-1 hyperion hyperspectral data to support the application of agricultural indexes," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 6, pp. 1246–1259, 2003.

[34] D. G. Ferris, R. A. Lawhead, E. D. Dickman, N. Holtzapple, J. A. Miller, S. Grogan, S. Bambot, A. Agrawal, and M. L. Faupel, "Multimodal hyperspectral imaging for the noninvasive diagnosis of cervical neoplasia," *Journal of Lower Genital Tract Disease*, vol. 5, no. 2, pp. 65–72, 2001.

[35] Z. Liu, Q. Li, J. Yan, and Q. Tang, "A novel hyperspectral medical sensor for tongue diagnosis," *Sensor Review*, vol. 27, no. 1, pp. 57–60, 2007.

[36] K. M. Rajpoot, N. M. Rajpoot, and M. Turner, "Hyperspectral colon tissue cell classification," *SPIE Medical Imaging (MI)*, 2004.

[37] S. J. Hamilton and R. A. Lodder, "Hyperspectral imaging technology for pharmaceutical analysis," in *Proc Soc Photo-Optic Instrum Eng*, 2002, pp. 2002–4626.

[38] D. B. Malkoff and W. R. Oliver, "Hyperspectral imaging applied to forensic medicine," in *Proceedings of SPIE*, vol. 3920, 2000, p. 108.

[39] A. A. Gowen, C. P. O'Donnell, P. J. Cullen, G. Downey, and J. M. Frias, "Hyperspectral imaging-an emerging process analytical tool for food quality and safety control," *Trends in Food Science & Technology*, vol. 18, no. 12, pp. 590–598, 2007.

[40] J. M. Duarte-Carvajalino, G. Sapiro, M. Velez-Reyes, and P. E. Castillo, "Multiscale representation and segmentation of hyperspectral imagery using geometric partial differential equations and algebraic multigrid methods," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 8, pp. 2418 –2434, Aug. 2008.

[41] R. M. Cavalli, F. Colosi, A. Palombo, S. Pignatti, and M. Poscolieri, "Remote hyperspectral imagery as a support to archaeological prospection," *Journal of Cultural Heritage*, vol. 8, no. 3, pp. 272–283, 2007.

[42] R. C. González and R. E. Woods, *Digital image processing*, 3rd ed. Prentice Hall, 2007.

[43] H. Ren and C. I. Chang, "Automatic spectral target recognition in hyperspectral imagery," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 39, no. 4, pp. 1232–1249, 2004.

[44] C. Bazán, M. Miller, and P. Blomgren, "Structure enhancement diffusion and contour extraction for electron tomography of mitochondria," *Journal of Structural Biology*, vol. 166, no. 2, pp. 144–155, May 2009. [Online]. Available: http://www.sciencedirect.com/science/article/B6WM5-4VR246P-1/2/7941e8ba9aa4e71fdd5038c422fbb9fo

[45] J. Weickert, B. M. Romeny, and M. A. Viergever, "Efficient and reliable schemes for nonlinear diffusion filtering," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 398 –410, Mar. 1998.

[46] J. Weickert, "Nonlinear diffusion filtering," in *Handbook on Computer Vision and Applications: Signal Processing and Pattern Recognition*. Academic Press, 1999, vol. 2, pp. 423–450.

[47] L. J. Grady and J. R. Polimeni, *Discrete Calculus: Applied Analysis on Graphs for Computational Science*. Springer, Aug. 2010.

[48] L. Florac, *Image Structure*, ser. Computational Imaging and Vision, 1997, vol. 10. [Online]. Available: http://www.springer.com/computer/image+processing/book/978-0-7923-4808-5

[49] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM REVIEW*, vol. 51, no. 3, pp. 455–500, 2009.

[50] S. Aja-Fernández, R. d. L. Garcia, D. Tao, and X. Li, *Tensors in Image Processing and Computer Vision*, 1st ed. Springer, Jul. 2009.

[51] J. Bigün and G. H. Granlund, "Optimal orientation detection of linear symmetry," in *IEEE First International Conference on Computer Vision*, London, Great Britain, Jun. 1987, pp. 433–438.

[52] Wikipedia contributors, "Structure tensor," May 2012, page Version ID: 490314716. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Structure_tensor&oldid=490314716

[53] M. Nicolescu and G. Medioni, "Motion segmentation with accurate boundaries: a tensor voting approach," in *Proceedings of the 2003 IEEE computer society conference on Computer vision and pattern recognition*, ser. CVPR'03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 382–389. [Online]. Available: http://dl.acm.org/citation.cfm?id=1965841.1965890

[54] J. Wiklund, V. Nicolas, P. Rondao, M. Andersson, and H. Knutsson, "T-flash: Tensor visualization in medical studio," *Tensors in Image Processing and Computer Vision*, pp. 455–466, 2009.

[55] G. Kindlmann, "Superquadric tensor glyphs," in *Proceedings of IEEE TVCG/EG Symposium on Visualization*, vol. 2004, 2004, pp. 147–154.

[56] A. Brun and H. Knutsson, "Tensor glyph warping: Visualizing metric tensor fields using riemannian exponential maps," *Visualization and Processing of Tensor Fields*, pp. 139–160, 2009.

[57] L. Cammoun, C. Castaño-Moraga, E. Muñoz-Moreno, D. Sosa-Cabrera, B. Acar, M. Rodriguez-Florido, A. Brun, H. Knutsson, and J. P. Thiran, "A review of tensors and tensor signal processing," in *Tensors in Image Processing and Computer Vision*, ser. Advances in Pattern Recognition, S. Aja-Fernández, R. Luis García, D. Tao, and X. Li, Eds. Springer London, 2009, pp. 1–32.

[58] B. Jähne, "Local structure," in *Handbook of Computer Vision and Applications*, ser. Signal Processing and Pattern Recognition. Academic Press, 1999, vol. 2, pp. 209–238.

[59] G. H. Granlund and H. Knutsson, *Signal processing for computer vision*. Springer, 1995.

[60] M. Kass and A. Witkin, "Analyzing oriented patterns," *Comput. Vision Graph. Image Process.*, vol. 37, no. 3, pp. 362–385, Mar. 1987.

[61] H. Knutsson, "Representing local structure using tensors," in *The 6th Scandinavian Conference on Image Analysis*, Oulu, Finland, June 1989, pp. 244–251, report LiTH–ISY–I–1019, Computer Vision Laboratory, Linköping University, Sweden, 1989.

[62] H. Knutsson, C.-F. Westin, and M. Andersson, "Representing local structure using tensors ii," in *Proceedings of the 17th Scandinavian conference on Image analysis*, ser. SCIA-11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 545–556.

[63] H. Knutsson and M. Andersson, "What's so good about quadrature filters?" in *Proceedings of the International Conference on Image Processing*, vol. 3, Sep. 2003, pp. III – 61–4 vol.2.

[64] B. Rieger and L. J. V. Vliet, "A systematic approach to nD orientation representation," *Image Vis. Comput*, vol. 22, pp. 453–459, 2004.

[65] v. d. R. Boomgaard and v. d. J. Weijer, "Robust estimation of orientation for texture analysis," in *The 2nd international workshop on texture analysis and synthesis*, 2002, pp. 135–138. [Online]. Available: http://dare.uva.nl/record/104550

[66] T. Brox, R. Boomgaard, F. Lauze, J. Weijer, J. Weickert, P. Mrázek, and P. Kornprobst, "Adaptive structure tensors and their applications," in *Visualization and Processing of Tensor Fields*, ser. Mathematics and Visualization, J. Weickert and H. Hagen, Eds. Springer Berlin Heidelberg, 2006, pp. 17–47, 10.1007/3-540-31272-2_2. [Online]. Available: http://dx.doi.org/10.1007/3-540-31272-2_2

[67] S. Di Zenzo, "A note on the gradient of a multi-image," *Computer Vision, Graphics, and Image Processing*, vol. 33, no. 1, pp. 116 – 125, 1986.

[68] G. Aubert and P. Kornprobst, *Mathematical problems in image processing: partial differential equations and the calculus of variations*. Springer-Verlag New York Inc, 2006.

[69] J. H. Ferziger and M. Perić, *Computational methods for fluid dynamics*. Springer Berlin, 1999.

[70] J. Weickert, "A review of nonlinear diffusion filtering," *Scale-space theory in computer vision*, pp. 1–28, 1997.

[71] ——, "Scale-Space properties of nonlinear diffusion filtering with a diffusion tensor," Laboratory of Technomathematics, University of Kaiserslautern, P.O, Tech. Rep., 1994.

[72] F. Catte, P. L. Lions, J. M. Morel, and T. Coll, "Image selective smoothing and edge detection by nonlinear diffusion," *SIAM J. Numer. Anal.*, vol. 29, no. 1, pp. 182–193, 1992.

[73] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629 –639, Jul. 1990.

[74] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel, "Axioms and fundamental equations of image processing," *Archive for Rational Mechanics and Analysis*, vol. 123, no. 3, pp. 199–257, 1993.

[75] M. Z. Nashed and O. Scherzer, "Least squares and bounded variation regularization with nondifferentiable functionals," *Numerical Functional Analysis and Optimization*, vol. 19, no. 7, pp. 873–901, 1998.

[76] G. H. Cottet and L. Germain, "Image processing through reaction combined with nonlinear diffusion," *Mathematics of Computation*, vol. 61, no. 204, pp. 659–673, 1993.

[77] O. Drblíková and K. Mikula, "Convergence analysis of finite volume scheme for nonlinear tensor anisotropic diffusion in image processing," *SIAM J. Numer. Anal.*, vol. 46, no. 1, pp. 37–60, 2007.

[78] C. Monet, "Woman with a parasol, looking left," 1886, published: Musée d'Orsay, Paris.

[79] R. Brügelmann and W. Förstner, "Noise estimation for color edge extraction," in *Robust Computer Vision*, W. Förstner and S. Ruwiedel, Eds. Wichmann, Karlsruhe, 1992, pp. 90–107.

[80] S. Nath and K. Palaniappan, "Adaptive robust structure tensors for orientation estimation and image segmentation," in *Advances in Visual Computing*, ser. Lecture Notes in Computer Science, G. Bebis, R. Boyle, D. Koracin, and B. Parvin, Eds. Springer Berlin / Heidelberg, 2005, vol. 3804, pp. 445–453, 10.1007/11595755_54.

[81] P. Mrazek, J. Weickert, and A. Bruhn, "On robust estimation and smoothing with spatial and tonal kernels," in *Geometric Properties for Incomplete data*, R. Klette, R. Kozera, L. Noakes, and J. Weickert, Eds. Dordrecht: Kluwer Academic Publishers, 2006, vol. 31, pp. 335–352. [Online]. Available: http://www.springerlink.com/content/g552624mnkrr2306/export-citation/

[82] T. Brox, J. Weickert, B. Burgeth, and P. Mrazek, "Nonlinear structure tensors," *Image and Vision Computing*, vol. 24, no. 1, pp. 41–55, Jan. 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1709276

[83] M. Marin-McGee and M. Velez-Reyes, "A structure tensor for hyperspectral images," in *3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. IEEE, Jun. 2011, pp. 1–4.

[84] R. Kimmel, R. Malladi, and N. Sochen, "Image processing via the beltrami operator," in *In Proc. of 3-rd Asian Conf. on Computer Vision, Hong Kong*, 1998, pp. 574–581.

[85] P. Bakker, L. J. van Vliet, and P. W. Verbeek, "Edge preserving orientation adaptive filtering," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 1999, p. 540 Vol. 1.

[86] R. v. d. Boomgaard, *Decomposition of the Kuwahara-Nagao Operator in Terms of a Linear Smoothing and a Morphological Sharpening*, R. B. H. Talbot, Ed., 2002.

[87] M. Kuwahara, K. Hachimura, S. Eiho, and M. Kinoshita, *Processing of RI-angiocardiographic images*, J. K. Preston and M. Onoe, Eds. Plenum Press, 1976.

[88] M. Nagao and T. Matsuyama, "Edge preserving smoothing," *Computer Graphics and Image Processing*, vol. 9, no. 4, pp. 394–407, Apr. 1979.

[89] W. Wang, J. Gao, and K. Li, "Structure-Adaptive anisotropic filter with local structure tensors," in *Second International Symposium on Intelligent Information Technology Application, IITA '08.*, vol. 2, Dec. 2008, pp. 1005 –1010.

[90] M. Middendorf and H. Nagel, "Empirically convergent adaptive estimation of grayvalue structure tensors," in *Pattern Recognition*, L. Gool, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, vol. 2449, pp. 66–74. [Online]. Available: http://www.springerlink.com/content/fbbrynkyy8yhfd4q/

[91] H. H. Nagel and A. Gehrke, "Spatiotemporally adaptive estimation and segmentation of OF-fields," in *Computer Vision ECCV-98*, H. Burkhardt and B. Neumann, Eds. Berlin/Heidelberg: Springer-Verlag, vol. 1407, pp. 86–102.

[92] G. Z. Yang, P. Burger, D. N. Firmin, S. R. Underwood, and D. B. Longmore, "Structure adaptive anisotropic filtering," in *Fifth International Conference on Image Processing and its Applications*, Jul. 1995, pp. 717 –721.

[93] A. Grigor'yan, "Heat kernels on weighted manifolds and applications," *The ubiquitous heat kernel: AMS special session, the ubiquitous heat kernel, October 2-4, 2003, Boulder, Colorado*, vol. 398, p. 93, 2006.

[94] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. Guibas, "One point isometric matching with the heat kernel," *Computer Graphics Forum*, vol. 29, no. 5, pp. 1555–1564, 2010.

[95] S. Rosenberg, *The Laplacian on a Riemannian Manifold: An Introduction to Analysis on Manifolds*. Cambridge University Press, Jan. 1997.

[96] P. Auscher, T. Coulhon, and A. Grigor'yan, *Heat kernels and analysis on manifolds, graphs, and metric spaces: lecture notes from a quarter program on heat kernels, random walks, and analysis on manifolds and graphs : April 16-July 13,*

2002, Emile Borel Centre of the Henri Poincaré Institute, Paris, France. American Mathematical Soc., 2003.

[97] V. Van Gogh, "Road with cypress and star," 1890, published: Auvers-sur-Oise.

[98] C. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 27, 1948.

[99] Jet Propulsion Laboratory, NASA, "AVIRIS line of deep horizon gulf of mexico oil spill. flight ID =f100517t01 and runId= p00_r06," http://aviris.jpl.nasa.gov/ql/10qlook/f100517t01p00r06_geo.jpeg, 2010. [Online]. Available: http://aviris.jpl.nasa.gov/ql/10qlook/f100517t01p00r06_geo.jpeg

[100] M. Marin-McGee and M. Velez-Reyes, "Enhancement of flow-like structures in hyperspectral imagery using tensor nonlinear anisotropic diffusion," in *Proc. SPIE 8048*, 2011, pp. 80 480E–80 480E–9.

[101] J. C. Strikwerda, *Finite difference schemes and partial differential equations*. Society for Industrial Mathematics, 2004.

[102] Y. Saad, *Iterative Methods for Sparse Linear Systems, Second Edition*, 2nd ed. Society for Industrial and Applied Mathematics, Apr. 2003.

[103] M. Peric, "Efficient semi-implicit solving algorithm for nine-diagonal coefficient matrix," *Numerical Heat Transfer*, vol. 11, no. 3, pp. 251–279, 1987.

[104] R. N. Clark, G. A. Swayze, K. E. Livo, R. F. Kokaly, S. J. Sutley, J. B. Dalton, R. R. McDougal, and C. A. Gent, "Imaging spectroscopy: Earth and planetary remote sensing with the USGS tetracorder and expert systems," *Journal of Geophysical Research*, vol. 108, no. E12, p. 5131, 2003.

[105] Purdue Research Foundation, "MultiSpec | tutorials," https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html, 2012. [Online]. Available: https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html

[106] J. Kerekes, A. Cisz, and R. Simmons, "A comparative evaluation of spectral quality metrics for hyperspectral imagery," *Proceedings of Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XI*, pp. 469–480, 2005.

[107] J. J. Cipar, "Background spectral library for fort A.P. hill, virginia," vol. 5544. SPIE, 2004, pp. 35–46.

[108] J. J. Fernández and S. Li, "An improved algorithm for anisotropic nonlinear diffusion for denoising cryo-tomograms," *Journal of Structural Biology*, vol. 144, no. 1-2, pp. 152–161, 2003.

[109] P. Mrázek and M. Navara, "Selection of optimal stopping time for nonlinear diffusion filtering," *Int. J. Comput. Vision*, vol. 52, no. 2-3, pp. 189–203, May 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=641431.641463

[110] J. B. Campbell and R. H. Wynne, *Introduction to Remote Sensing*, 5th ed. Guilford Press, Jun. 2011.

[111] E. T. Slonecker, D. M. Shaw, and T. M. Lillesand, "Emerging legal and ethical issues in advanced remote sensing technology," *Photogrammetric engineering and remote sensing*, vol. 64, no. 6, pp. 589–595, 1998. [Online]. Available: http://cat.inist.fr/?aModele=afficheN&cpsidt=2241832

[112] N. Rastogi, "Google's blind spots," *Slate*, Dec. 2008. [Online]. Available: http://www.slate.com/articles/news_and_politics/explainer/2008/12/googles_blind_spots.html

[113] S. Hitchings, "Policy assessment of the impacts of remote-sensing technology," *Space Policy*, vol. 19, no. 2, pp. 119–125, May 2003. [Online]. Available: http://library.uprm.edu:2051/science/article/pii/S0265964603000171

[114] United Nations, "Declaration of principles regarding the remote sensing of earth from space." 1987.

[115] "Camp delta, google earth and the ethics of remote sensing in archaeology," vol. 42. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/00438243.2010.498640

[116] S. McKee, "Remote sensing issues at the supreme court of canada," *BULLETIN-ASSOCIATION OF CANADIAN MAP LIBRARIES AND ARCHIVES*, vol. 123, p. 3, 2005.

[117] "Satellite evidence of human rights violations (1): Darfur andÂ zimbabwe," http://filipspagnoli.wordpress.com/2009/01/31/satellite-evidence-of-human-rights-violations/, Jan. 2009. [Online]. Available: http://filipspagnoli.wordpress.com/2009/01/31/satellite-evidence-of-human-rights-violations/

[118] American Association for the Advancement of Science, "AAAS - geospatial technologies and human rights," http://shr.aaas.org/geotech/index.shtml. [Online]. Available: http://shr.aaas.org/geotech/index.shtml

[119] "Satellite sentinel project," http://www.satsentinel.org/. [Online]. Available: http://www.satsentinel.org/

[120] Rhett Butler, "Satellites and google earth prove potent conservation tool by rhett butler: Yale environment 360," http://e360.yale.edu/content/feature.msp?id=2134, Mar. 2009. [Online]. Available: http://e360.yale.edu/content/feature.msp?id=2134

[121] G. Steidl and T. Teuber, "Diffusion tensors for processing sheared and rotated rectangles," *Image Processing, IEEE Transactions on*, vol. 18, no. 12, pp. 2640 – 2648, dec. 2009.

[122] G. Gravvanis, C. Filelis-Papadopoulos, and K. Giannoutakis, "Solving finite difference linear systems on GPUs: CUDA based parallel explicit preconditioned biconjugate conjugate gradient type methods," *The Journal of Supercomputing*, pp. 1–15.

[123] R. Mendez-Rial, M. Calvino-Cancela, and J. Martin-Herrero, "Accurate implementation of anisotropic diffusion in the hypercube," *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 4, pp. 870–874, Oct. 2010.

[124] R. Mendez-Rial and J. Martin-Herrero, "Efficiency of Semi-Implicit schemes for anisotropic diffusion in the hypercube," *IEEE Transactions on Image Processing*, 2011.