GEOMETRIC SCALE-SPACE FRAMEWORK FOR THE ANALYSIS OF HYPERSPECTRAL IMAGERY

By

Julio Martín Duarte-Carvajalino

A dissertation 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
MAYAGÜEZ CAMPUS
December, 2007

Approved by:

_____	_____
Shawn D. Hunt, Ph.D Member, Graduate Committee	Date
_____	_____
Wilson Rivera Gallego, Ph.D Member, Graduate Committee	Date
_____	_____
Paul E. Castillo, Ph.D Member, Graduate Committee	Date
_____	_____
Guillermo Sapiro, Ph.D Member, Graduate Committee	Date
_____	_____
Miguel Vélez-Reyes, Ph.D President, Graduate Committee	Date
_____	_____
Fernando Gilbes, Ph.D Representative of Graduate Studies	Date
_____	_____
Nestor Rodríguez, Ph.D Program Coordinator	Date

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

GEOMETRIC SCALE-SPACE FRAMEWORK FOR THE ANALYSIS AND PROCESSING OF HYPERSPECTRAL IMAGERY

By

Julio Martín Duarte-Carvajalino

December, 2007

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

ABSTRACT

This work introduces a framework for a fast and algorithmically scalable multiscale representation and segmentation of hyperspectral imagery. The framework is based on the scale-space representation generated by geometric partial differential equations (PDEs) and state of the art numerical methods such as semi-implicit discretization methods, preconditioned conjugated gradient, and multigrid solvers. Multi-scale segmentation of hyperspectral imagery exploits the fact that different image structures exists only at different image scales or resolutions, enabling a better exploitation of the high spatial-spectral information content in hyperspectral imagery. Higher level processes in hyperspectral imagery such as classification, registration, target detection, restoration, and change detection

can improve significatively; by working on the regions (objects) identified by the segmentation process, rather than with the image pixels, as it is traditionally done.

The main contribution of this work is the introduction of a framework, where vector-valued geometric scale-spaces are seamlessly integrated with an algorithm for multiscale segmentation of hyperspectral imagery, in a fast and scalable way that makes feasible an object-oriented approach for higher level processes in hyperspectral image processing.

Resumen de Disertación Presentado a Escuela Graduada
de la Universidad de Puerto Rico como requisito parcial de los
Requerimientos para el grado de Doctor en Filosofía

ESPACIO DE ESCALA GEOMETRICO COMO MARCO TEORICO PARA EL ANALISIS Y PROCESAMIENTO DE IMAGENES HYPERSPECTRALES

Por

Julio Martín Duarte-Carvajalino

Diciembre, 2007

Consejero: Miguel Veléz-Reyes, Ph.D
Departamento: Ciencias e Ingeniería de la Computación y la Información

RESUMEN

Este trabajo presenta una base formal para la representación y segmentación multi-escala de imágenes hiper-espectrales en forma rápida y escalable algorítmicamente. El fundamento de este trabajo se basa en la representación de espacio de escala generada por ecuaciones diferenciales parciales geométricas y métodos numéricos modernos como la discretización semi-implícita, gradiente conjugado precondicionado, y métodos multi-malla. La segmentación multi-escala de imágenes hiper-espectrales explota el hecho de que diferentes estructuras en la imagen existen únicamente a diferentes escalas o resoluciones de la imagen,

permitiendo una mejor explotación del alto contenido espacial y espectral en imágenes hiper-espectrales. Métodos de procesamiento de imágenes hiper-espectrales de mayor nivel tales como clasificación, registración, detección de objetos, restauración de imágenes, y detección de cambio pueden mejorar significativamente; trabajando con las regiones identificadas por la segmentación, en lugar de usar los píxeles de la imagen, como se hace tradicionalmente.

La principal contribución de este trabajo es la introducción de una base formal, donde el espacio de escala para imágenes vectoriales es integrado en forma natural con un algoritmo para la segmentación multi-escala de imágenes hiper-espectrales, con una complejidad computacional que es lineal en el tamaño de las imágenes.

Nobody climbs mountains for scientific reasons. Science is used to raise money for the expeditions, but you really climb for the hell of it.
EDMUND HILLARY

To wonder woman (my wonderful wife) and family . . .

ACKNOWLEDGEMENTS

During the development of my graduate studies in the University of Puerto Rico several persons and institutions collaborated directly and indirectly with my research. Without their support it would be impossible for me to finish my work. That is why I wish to dedicate this section to recognize their support.

I want to start expressing a sincere acknowledgement to my advisor, Dr. Miguel Vélez-Reyes from the Electric and Computer Engineering Department at the UPRM, because he gave me the opportunity to research under his guidance and supervision, both in the academic and personal areas. I received motivation; encouragement and support from him during all my studies. I also express here my sincere gratitude to Dr. Paul Castillo from the Mathematical Sciences Department at UPRM and Dr. Guillermo Sapiro from the Electric and Computer Engineering Department at the University of Minnesota, who helped me to get out of the dead ends and crossroads I found my self during my research and guided me to the satisfactory results we reached in this work. Also I want to thank Dr. Edward Bosch who introduced us to Dr. Guillermo Sapiro and who also listened to our ideas and provide us feedback, with his comments.

I wish also to thank Mr. Larry Biehl for his valuable help in understanding the full potential of Multispec (see Section 3.3), a freeware software developed by D. A. Landgrebe

from Purdue University to analyze multi and hyperspectral imagery and that is consistently used in this thesis as a tool to evaluate our results.

The NSF-EPSCOR fellowship, the Engineering Research Center Program of the National Science Foundation (NSF) under Award Number EEC-9986821, and the National Geospatial Agency (NGA) provided me with the funding and resources needed for the development of this research.

At last, but not least important I would like to thank my wife and family, for their unconditional support, inspiration and love that help me during this stage of my life.

TABLE OF CONTENTS

Table List

Figure List

List of Acronyms

ADI	Alternating Direction Implicit
AMG	Algebraic Multigrid
AOS	Additive Operator Splitting
AVIRIS	Airborne Visible/Infrared Imaging Spectrometer
BIL	Band Interleaved by Line
BIP	Band Interleaved by Pixel
BSQ	Band Sequential
ECHO	Extraction and Classification of Homogeneous Objects
ED	Euclidean Distance
FLD	Fisher Linear Discriminant
GDAL	Geospatial Data Acquisition Library
CG	Conjugated Gradient
GS	Gauss-Seidel
HSI	Hyperspectral Imagery
HYDICE	Hyperspectral Digital Imagery Collection Experiment
LHS	Left Hand Side
LOD	Locally One-Dimensional
MF	Matched Filter
MG	Multigrid
ML	Maximum Likelihood
NASA	National Aeronautics and Space Administration
NW	North Western
PCG	Preconditioned Conjugated Gradient
PDE	Partial Differential Equation
RGB	Red-Green-Blue
RHS	Right Hand Side
SAM	Spectral Angle Mapper
SNR	Signal to Noise Ratio
SSOR	Successive Over-Relaxation
TV	Total Variation
USGS	U.S. Geological Survey

Notation and Operators

$\mathbf{A}$, …, $\mathbf{Z}$	matrices (bold uppercase)
$\mathbf{A}^{-1}$	matrix inverse
$\mathbf{A}^{T}$	matrix transposed
$\mathbf{a}$, …, $\mathbf{z}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\xi}$, $\boldsymbol{\eta}$	vectors (bold lowercase)
$a, ..., z, \alpha, \beta, \eta, \theta, \gamma, s$	variable scalars (italic lowercase)
M, N, S	constants (italic uppercase)
A, …, Z, Ω	sets (uppercase)
$\mathbb{N}$	set of all natural numbers
$\mathbb{Z}$	set of all integers
$\mathbb{R}$	set of all real numbers
$\mathbb{R}^n$	real n-space
i, j, k, l, m	index (cursive subscript)
a_i	vector element
a_{ij}, g_{ij}	matrix element
$[A]_{ij}$	notation to indicate all the elements of matrix **A**
Δ	Laplacian operator
∇	Gradient operator
∇•	Divergence operator
$\|\ \|$	2-norm
$\langle \mathbf{u}, \mathbf{v} \rangle$ or $\mathbf{u} \bullet \mathbf{v}$	dot product between vectors
A \ B	difference between sets A and B
A ∪ B	union between sets A and B
$a \in$ A	a is an element of set A
A ⊂ B	A is a subset of B
∑	summation
O(.)	"Big-Oh" asymptotic upper bound
G(μ,σ)	Gaussian kernel of mean μ and standard deviation σ
$\hat{\imath}, \hat{\jmath}$	unitary vectors along the x and y axis, respectively.

1 INTRODUCTION

> … From then on, the development of multi-dimensional spectroscopy went very fast, inside and outside of our research group.
> RICHARD R. ERNST

Remote sensing imagery is increasingly being employed as a significant component in the evaluation and management of terrestrial and coastal ecosystems. Advantages of this technology include both the qualitative benefits derived from a visual overview, and more importantly, the quantitative abilities for systematic assessment and monitoring the earth ecosystem revealing patterns and relationships unavailable when using traditional data-gathering techniques. From the different remote sensing technologies, hyperspectral imaging has the greatest potential to extract more plentiful and more accurate environmental information, given the enhanced discrimination capabilities of high spectral resolution imagery.

Hyperspectral Remote Sensing provides high-resolution spectral measurements that enable the identification of physical composition and properties of objects and materials from airborne or space-borne platforms. Figure 1.1 illustrates the hyperspectral image concept, where each pixel is a vector containing many samples of the spectral signature at that pixel and at each wavelength an image of the area under study is collected.

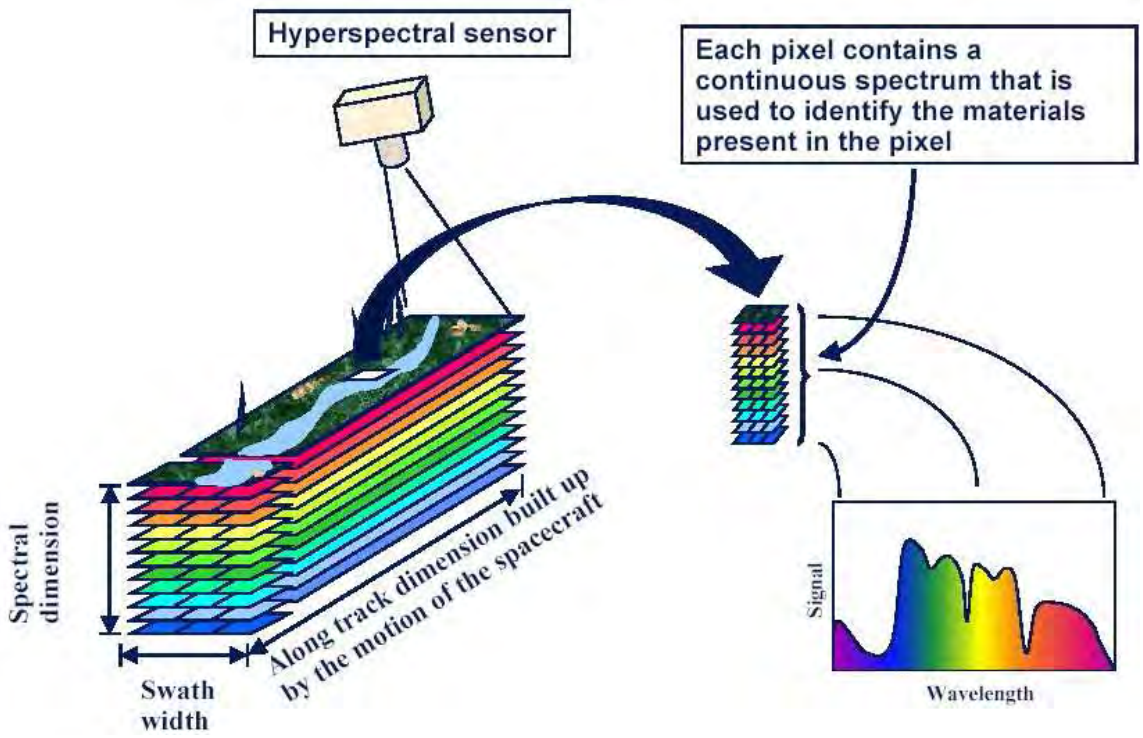


Figure 1.1 Hyperspectral image (taken from [*Davis*, 2000]).

It has been recognized [*Gorte,* 1998; *Baatz and Schäpe*, 2000; *Blaschke et al*, 2000, 2001] that the information required for critical analysis and understanding of remotely sensed imagery is usually not represented in terms of pixels, but in the spatial structures (objects) and their relationships at different image scales. The process of extracting the structures at different image scales is called in image processing, *multiscale image segmentation*. Segmenting an image consists in partitioning the image into non-overlapping homogeneous regions that may correspond to the semantically meaningful structures. Hence, multiscale segmentation is of prime importance in image analysis and understanding of remotely sensed imagery and particularly of hyperspectral imagery, given its higher amount of potential information.

Important higher level image processes in image processing such as classification, target detection, registration, change detection, and restoration that are traditionally performed on a pixel by pixel basis can benefit enormously from a previously segmented image, given the higher statistical and geometrical information content that can be drawn from the structures found.

1.1 Problem Statement

The task of a visual processing system is to extract meaningful information about the outside world, implicitly represented as a set of pixels values that are the result of measurements of an electromagnetic field from a physical scene. However, the vision problem is ill-posed in the sense of Hadamard, since it has no unique solution [*Lindeberg*, 1991]. Different objects may produce the same pixel values under different illumination conditions; hence, two different scenes may produce similar images that become undistinguishable due to the noise introduced by the transmission and sensor systems.

In particular image segmentation, i.e. the extraction of the different structures in the image using only the pixel values, is also an ill-posed problem. The noise in the image, the presence of fuzzy boundaries among the different image structures, occlusions, illumination differences, shadows, and image defects such as optical and electronic blurring due to the sensor system and other variations in the intensity of the image, introduced by variations in the sensor altitude, roll, pitch, and yaw angles; makes possible that different segmentations be equally good, based on metrics that uses the segments found and the pixel values only.

The segmentation problem can be casts into a graph partitioning problem, where the pixels in the image corresponds to nodes in the graph, the edges in the graph connect each

pixel with its nearest neighbors and, associated with the edges, there is a weight function that measures the degree of similarity between two neighboring pixels. In this, setting, the segmentation problem can be expressed as the optimal cut of the graph into a number of disjoints subsets of pixels that maximize the similarity (homogeneity) within each segment and the dissimilarity across segments. From the computational point of view, the segmentation problem is an NP-hard problem, since optimal graph-cut partitioning is NP-hard [*Shi and Malik*, 2000].

Given the importance and computational complexity of the image segmentation task, over 1000 kinds of segmentation approaches have been developed in the past for grayscale and color images [*Zhang*, 2001]. Nevertheless, segmentation algorithms have been introduced relatively late for vector valued images such as multi and hyperspectral imagery given the high dimensionality of the data, the heterogeneity (spatial and spectral) of the image structures, and the difficulty of using model-based methods, such as the classic background-foreground model [*Chen et al*, 2003; *Blaschke et al*, 2000, 2001] used successfully for several grayscale and color segmentation algorithms. In order to have an idea of the dimensionality of the data handled in hyperspectral imagery, let us consider a typical hyperspectral image such as the Cuprite image taken with the AVIRIS sensor over the Cuprite mining district, 2km north of Nevada (see Section 3.2), which consists of 2378 lines, 640 columns and 224 bands, i.e. 640x2378x224 ~ $3x10^9$ intensity values that must be discretized using at least 2 bytes (unsigned integer format) of resolution. Hence, a typical hyperspectral image requires ~1Gb of memory and processing the image would require to perform several operations over each one of the ~10^9 variables in the image, plus storing the

additional variables required by the processing algorithm, which might be also of the same size as the original image.

Hence, traditionally in remote sensing the task of segmentation had been done only with multispectral imagery and using a pool of heuristic recipes, since the dimensionality of the data damps the use of more sophisticated techniques, such as the geometric scale-space, governed by Partial Differential Equations (PDEs). With the advances in computer technology and the use of state of the art numerical methods, the processing of hyperspectral imagery using a well-founded framework such as the scale-space is becoming more and more feasible, nowadays. An important achievement of this work is the extension of state of the art numerical methods to solve fast and accurately the anisotropic diffusion PDE on high dimensional vector-valued spaces making feasible the use of formal scale-spaces to process hyperspectral imagery.

1.2 Geometric Differential Equations

Deterministic differential equations are all embedded in a geometric space, where the independent and dependent variables (including the solution) resides, i.e. the so called jet space (see [*Sapiro*, 2001]). Famous examples of geometric differential equations are Newton's laws, Maxwell equations and the general theory of relativity. Newton's laws, for instance, are embedded in a geometric Euclidean space, where the light always follows a straight line between two points in space that corresponds to the Euclidean distance. On the other hand, the general theory of relativity is embedded in a Riemannian space, where the light travels along the shortest geodesic distance between two points, in the manifold

generated by the gravitational field, hence, the light might seem to bend in Euclidean space [*Faber and Naber*, 1986].

In this work, we use geometric diffusion PDEs that correspond to conservation laws determining the transfer of mass or energy between different points on a fluid, where a gradient of concentration exists. For simplicity, the diffusion PDE that we use in our work is embedded in a Euclidean space and hence, the flux follows the shortest path between two points in Euclidean space. However, a Riemannian space could also be defined, where the diffusion process follows the geodesic lines defined by the manifold that contains the hyperspectral image, in feature space (see for instance [*Bachmann et al*, 2006]). The lector interested in a formal presentation of geometric PDEs for image analysis is referred to the excellent book on geometric PDEs and image analysis made by [*Sapiro*, 2001] and the references therein on differential geometry.

We also use the term geometric PDEs to distinguish them from the stochastic PDEs (SPDEs), where the dependent and independent variables (including the solution) are modeled as random fields. Stochastic PDEs have the advantage of modeling the presence of noise in the data, and the data itself with probabilistic distributions, and they present also some nice properties such as convergence to a non-trivial solution (see for instance [*Unal et al*, 2002], [*Descombes and Zhizhina*, 2003]). However, SPDEs are difficult to solve for any numerical method in high dimensional spaces, as is the case of hyperspectral imagery. The computational complexity of SPDEs increases exponentially with the dimension, for all optimal algorithms [*Novach*, 1988; *Novach and Ritter*, 1997].

1.3 Technical Approach

We propose here to obtain a scale-space representation of hyperspectral imagery using vector-valued geometric partial differential equations (PDEs). This representation regularizes the image by simplifying it, in away that only interesting features are preserved, while unimportant ones are removed [*Tschumperlé and Deriche*, 2005]. In particular, we show [*Duarte et al* 2006, 2007] (see Chapter 3) that the nonlinear diffusion PDE reduces noise and spectral-spatial intra-region variability in hyperspectral imagery, improving classification accuracy figures.

The advantage of using geometric PDEs to generate scale-space representations of hyperspectral images is that they satisfy information-reducing, stability, and invariance properties that have been shown to be of fundamental importance in image processing [*Alvarez et al*, 1992]. Other approaches such as wavelet shrinkage [*Donoho and Johnstone*, 1994] and morphology [*Bosworth and Acton*, 2003] to multiscale image representation have been proven to be equivalent to the continuous scale-space framework generated by PDEs [*Sapiro*, 2001]; but on implementation, many of the geometric properties of the continuous scale-space are missing in the discrete version, due to the propagation of numerical errors that introduce artifacts as the scale increases [*Durand and Froment*, 2003]. On the other hand, the well-founded numerical methods that exist for PDEs extend the properties of the continuous scale-space onto the discrete domain [*Weickert*, 96] making geometric PDEs a good choice to generate a discrete scale-space representation of hyperspectral imagery.

Given that the appropriate scale for a given application is in general unknown, the only reasonable approach is to obtain a hierarchical multi-resolution representation of the image at several scales [*Lindeberg*, 1991]. Hierarchical multiscale representation of images has another important advantage. The computationally expensive graph-partitioning problem in the fine grid of the image can be brought to a coarse scale, where it can be solved with much lower computational cost, and then propagated back to the finest level. In fact, it has been argued that solving the segmentation problem at a coarser scale produce better segmentation results than solving it in the finest scale, where only local information is used [*Sharon et al*, 2000, 2003]. On a hierarchical multiscale representation of the image, statistic and geometric information (shapes) can be gathered from the fine to the coarser levels, so that local and global information are both available to the segmentation process [*Sharon et al*, 2000, 2003]. As we show in [*Duarte et al*, 2007] (see Chapter 4) a multiscale hierarchical segmentation of hyperspectral imagery can be obtained, within the scale-space framework.

The main contribution of this work is the introduction of the geometric scale-space framework to represent and segment multispectral/hyperspectral imagery and the extension of state of the art numerical methods to make this framework computationally feasible. In particular, our contributions are

- The scale-space representation (see Section 2.1) of HSIs increases class separability, which has the potential of improving classification and segmentation accuracies. Besides class separability, scale-space representation of HSIs has

also the potential of improving other image processes such as registration, target detection and image compression.

- Asymptotically optimal algorithms. The scale-space representation and scale-based segmentation of HSIs have been damped in the past due to high computational complexity and the size of the data. Here, we present algorithms that are algorithmically scalable, i.e. their time complexity is linear in the size of the images making the use of the scale-space representation of HIS not only viable but also very attractive computationally.

1.4 Thesis Outline

We first review scale-space theory in Chapter 2. Chapter 3 presents a comparative study of semi-implicit discretization and preconditioned conjugated gradient (PCG) methods to solve the nonlinear diffusion PDE on hyperspectral imagery. The solution of the nonlinear diffusion PDE smoothes undesirable variability in HSIs, which in turn improves class separability. This scale space representation is obtained in the fine fixed grid of the image and hence, no multiscale representation of the images can be obtained. Chapter 4 introduces multigrid methods that allow solving the nonlinear diffusion PDE, with good accuracy, and at the same time provide the necessary structure to segment hyperspectral imagery. Multigrid methods enable a multiscale representation of hyperspectral imagery that in turn facilitates image segmentation, with improved accuracy. The ethical considerations of this work are presented in Chapter 5 and the Conclusions of this work and possible continuations are presented in Chapter 6.

2 BACKGROUND

The skeptic will say: “It may well be true that this system of equations is reasonable from a logical standpoint. But this does not prove that it corresponds to nature.” You are right, dear skeptic. Experience alone can decide on truth. Pure logical thinking cannot yield us any knowledge of the empirical world: all knowledge of reality starts from experience and ends in it.
ALBERT EINSTEIN

2.1 The Scale-Space Concept

(a) **(b)** **(c)**

Figure 2.1 Scale-space concept.

Objects in the world and details on an image only exist and make sense over a limited range of spatial scales or loosely speaking, image resolutions [*Lindeberg*, 1991]. Figure 2.1 shows the same scene, at three spatial resolutions. The finest scale (higher resolution) corresponds to Figure 2.1.a, where the individual leaves and tree’s branches can be distinguished. The

scale increases (resolution becomes coarser) from Figure 2.1.b to Figure 2.1.c, and on the coarser scale, only the basic shape of the tree can be noticed, but the internal details cannot longer be appreciated. From this example, it is clear that the tree object exists only within a range of image scales. At finer scales, say nanometers, an image could only display the tree's molecules, while at the scale of kilometers, the individual trees cannot be distinguished, only the forest. It is also evident that the range of useful scales for a given image is determined by the application itself. A scientist interested in the number and shape of the leaves of a tree would require the resolution depicted on Figure 2.1.a or higher, while a scientist interested only in the number of trees or in the forest, would require the scale depicted on Figure 2.1.c. or another image at a lower spatial resolution.

Here, the concept of scale-space is being made equivalent to the image resolution, which can be expressed as the spatial dimensions that a pixel on the image represents in the physic world, or in terms of pixels of the image itself. Now, we will define the scale-space concept formally.

Let us define a grayscale image as a bounded real function $u(\mathbf{x})$: $\mathbb{R}^n \rightarrow \mathbb{R}$ with $u(\mathbf{x}) \in \mathbb{R}$ being the intensity at the point $\mathbf{x} = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix}^T \in \mathbb{R}^n$, $n \in \mathbb{Z}^+$. A multiscale image analysis is a family of transforms $\{T_t, t \geq 0\}$, where t is the scale parameter, embedding the image $u(\mathbf{x})$ into a family $u(\mathbf{x}, t) = (T_t\, u)(\mathbf{x})$ of gradually simplified versions satisfying architectural axioms, morphological requirements, and stability [*Alvarez et al*, 1993; *Weickert*, 1996]. These axioms and requirements ensure that the multiscale analysis does not introduce artifacts and it is invariant to affine transformations of the image (rescaling, rotation, translation, etc) [*Weickert*, 1996].

The architectural axioms are recursivity, causality, regularity and locality. The recursivity axiom ensures that the scale t can be reached as a sequence of smaller scale steps,

$$T_0 u = u,$$
$$T_t u = T_{t_2}\left(T_{t_1} u\right), \quad t = t_1 + t_2, t_1 \geq 0, t_2 \geq 0.$$

The causality axiom ensures that the transformation depends on the image smoothed at previous scales but not on the image at higher scales, i.e. $T_t u$ can be computed from $T_s u$, $s \leq t$, but not from $s > t$. The regularity axiom requires continuity of the transformation, i.e.

$$\textit{Supremum}\ \{T_t(u+hv) - (T_t(u)+hv)\} \leq \delta ht \ \text{ for } h, t \in [0, 1] \text{ and } u, v \in C^{\infty},$$

where δ depends only on u and v and C^{∞} is the set of all functions that have derivatives of all orders. The locality axiom establishes that $T_{\Delta t} u$ for Δt small is determined by the behavior of u in the close neighborhood of u,

$$Lim_{\Delta t \to 0} \frac{T_{\Delta t} u - T_{\Delta t} \hat{u}}{\Delta t} = 0 \quad \text{for} \quad \frac{\partial^i u}{\partial \mathbf{x}} = \frac{\partial^i \hat{u}}{\partial \mathbf{x}} \quad i \geq 0.$$

Notice that these axioms are also common requisites of realizable transforms in signal processing.

The stability condition is expressed in the maximum principle. If $u \leq v$ then $T_t(u) \leq T_t(v)$ for all $t \geq 0$. This condition ensures that the scale-space transformation does not create additional structures that are not present in the original image (artifacts). Notice that the maximum principle, allows structures to disappear (e.g. under diffusion), since if $u(\mathbf{x}_1) < u(\mathbf{x}_2)$, where $\mathbf{x}_1$ and $\mathbf{x}_2$ are two locations on both sides of a weak edge, the maximum principle does not prohibit that $T_t(u(\mathbf{x}_1)) = T_t(u(\mathbf{x}_2))$, but it does prohibit $T_t(u(\mathbf{x}_1)) > T_t(u(\mathbf{x}_2))$, which would mean an artifact by inversion of the intensity order.

The morphological requirements establish that the scale-space transformation must be invariant to constant changes in the intensity of the image (gray level shift invariance), and affine invariance that includes as particular cases, scale change, rotation and translation of the image,

$$\text{Gray level shift invariance}: T_t(u+a) = T_t(u) + a$$
$$\text{Affine invariance}: T_{t'}(u(\mathbf{Ax})) = T_t(u)(\mathbf{Ax}).$$

where, a is a constant and $\mathbf{A}$ is an $n \times n$ matrix. These requirements ensure that the scale-space analysis is invariant to illumination changes and affine transformations and it depends only on the underlying structures in the image.

The pioneering work of [*Alvarez et al*, 1993] consisted into proving that every scale-space transformation satisfying the cited architectural axioms, invariance properties and maximum principle are governed by parabolic Partial Differential Equations (PDEs) of the form,

$$\frac{\partial u(\mathbf{x},t)}{\partial t} = f\left(\nabla^2 u, \nabla u\right), \quad u(\mathbf{x}, t=0) = u_0, \qquad \textbf{2.1}$$

where, u_0 is the original image, f is a continuous function, $u(\mathbf{x}, t) = T_t(u)$, and

$$\nabla u = \left(\frac{\partial u}{\partial x_1} \quad \frac{\partial u}{\partial x_2} \quad \cdots \quad \frac{\partial u}{x_n} \right)^T, \quad \nabla^2 u = \nabla u \left(\nabla u\right)^T.$$

Notice that even tough any formal scale space is governed by a PDE of the form given by Equation 2.1, not every PDE of this form will generate a valid scale space, as it will be seen in the next section. Notice also that scale in the formal scale-space framework is the level of

image smoothing defined by the evolution of a PDE of the form given by Equation 2.1 that satisfies the formal axioms and requirements of the scale-space.

From now on and for simplicity of the notation, we will drop the explicit dependence on **x** and t of the intensity of the image, u, but it must be always assumed here that u is a function of both space **x** and scale t, also $n = 2$ since we are interested in 2D images only.

2.2 Scale-Space Generated by Geometric PDEs

A simple and well-known particular form of Equation is the heat diffusion equation, given by,

$$\frac{\partial u}{\partial t} = g\Delta u, \quad u(t=0) = u_0, \tag{2.2}$$

where g is the diffusion coefficient, assumed to be constant. The linear scale-space proposed by Lindeberg [1991] is generated by Equation 2.2. This is also called Gaussian scale space, Gaussian blurring or isotropic smoothing, since the analytic solution of Equation 2.2 is given by the convolution of the original image, u_0 with a Gaussian kernel,

$$u(t) = G(0, 2gt) * u_0, \tag{2.3}$$

where, G is a zero mean Gaussian function with variance $2gt$ and * is the convolution operation.

Figure 2.2.a shows the same original image shown on Figure 2.1.a, while Figure 2.2.b and Figure 2.2.c show the same image after Gaussian blurring and sub-sampling by a factor of 4 and 16, respectively. Notice that Figure 2.1.b and Figure 2.1.c present strong visible transitions between the pixels in the image that are due to the sub-sampling process. On the other hand, Figure 2.2.b and Figure 2.2.c have the same spatial resolution than Figure 2.1.b

and Figure 2.1.c, but the transition of the intensity between pixels is very smooth. However, from Figure 2.2, we can also notice the main disadvantage of Gaussian diffusion: not only the noise and the intra-object variability (details) are smoothed, but also the image edges are weakened and dislocated, as it goes from fine to coarser scales.

(a) **(b)** **(c)**

Figure 2.2 Gaussian Blurring.

Equation 2.2 in thermodynamics and other physical sciences corresponds to a conservation law that describes the evolution in time of the concentration of a physical quantity, which can be for instance mass or energy, with constant net mass or energy. In general, the scale-space representation of an image can be viewed as the diffusion of the gray level intensity of the image trough time (scale). The flux of intensity, Φ, is given by [*Weickert*, 1996], $\Phi = -g\nabla u$, where the negative sign comes from the fact that the flux is trying to compensate the intensity gradient. The conservation law stating that net mass or energy cannot be created or destroyed by the diffusion process is expressed as,

$$\frac{\partial u}{\partial t} = -\nabla \bullet \Phi = \nabla \bullet \left(g\nabla u\right), \qquad \textbf{2.4}$$

where, $\nabla\bullet$ is the divergence operator. Equation 2.4 states that concentration (intensity) reduces trough time (scale) as the flux diverges from a point in the image. If we assume that the intensity can flow in all directions (isotropically), without any restriction, the diffusion coefficient *g* is constant and Equation 2.4 reduces to Equation 2.2, since the divergence of the gradient is the Laplacian. The analogy of scale-space as a diffusion process of the image intensity allows us to see why Gaussian blurring destroys the edges in the image. Since the diffusion coefficient is constant (isotropic), the higher the image gradient the higher the flux of intensity, affecting principally the image edges characterized by a high gradient.

This important observation was made by [*Perona and Malik*, 1990], who proposed smoothing the images with a nonlinear diffusion coefficient that prevents diffusion on the image edges. If the image is defined on $\mathbf{x}\in\Omega\subset\mathbb{R}^2$, being $\Omega = (0, a)\times(0, b)$ the image domain, with boundary $\partial\Omega$, and $t \in [0, T]$, the Perona-Malik PDE is given by [*Weickert*, 1996],

$$\begin{aligned} &\frac{\partial u}{\partial t} = \nabla \bullet \left[g\left(\|\nabla u\| \right) \nabla u \right], \quad \text{on} \quad \Omega \times (0,T] \\ &u(t=0) = u_0 \quad \text{on} \quad \Omega \\ &\left[g\left(\|\nabla u\| \right) \nabla u \right] \bullet \hat{n} = 0 \quad \text{on} \quad \partial\Omega \times (0,T], \end{aligned} \tag{2.5}$$

where, $\|\ \|$ is the 2-norm, $\hat{n}$ is the unit normal vector to the boundary $\partial\Omega$, and $\bullet$ is the dot product. The boundary conditions given by the last condition on Equation 2.5 imply that there is no flux in or out the borders of the image. From now on and for simplicity, we will write only the PDE without referring explicitly to the initial or boundary conditions, since they will be the same as in Equation 2.5, i.e. the initial condition of the PDE is the original image and the boundary conditions forbids flux entering or leaving the image.

Perona and Malik (1990) also proposed diffusion coefficients of the form,

$$g\left(\left\|\nabla u\right\|\right)=\frac{1}{1+\left(\frac{\left\|\nabla u\right\|}{\alpha}\right)^2},\quad g\left(\left\|\nabla u\right\|\right)=e^{-\left(\frac{\left\|\nabla u\right\|}{\alpha}\right)^2}, \tag{2.6}$$

where, α is a threshold parameter that allows diffusion in those image regions where the magnitude of the gradient is $\left\|\nabla u\right\|<\alpha$ (within homogeneous regions) and performs backward diffusion, when $\left\|\nabla u\right\|\geq\alpha$. Backward diffusion means that on edges, with a gradient above the threshold, the flux goes from low intensity to high intensity (contrary to forward diffusion), which implies that edges may be enhanced rather than simply preserved. One can see more clearly why this happens by restricting Equation 2.5 to the 1-D case,

$$\frac{\partial u}{\partial t}=-\frac{\partial \Phi}{\partial x}. \tag{2.7}$$

Figure 2.3 shows the derivative of the flux vs. $\partial u/\partial x$, where we had used the exponential diffusion coefficient given on Equation 2.6, with $\alpha = 0.5$. From this figure, it is clear that for $\partial u/\partial x > 0.5$ the rate of flux is negative and hence, the gradient increases (Equation 2.7). The idea of smoothing the image within its homogeneous regions, while enhancing the edges seems like an ideal situation. However, [*Alvarez et al*, 1992] showed that the backward diffusion process makes Equation 2.5 ill-posed. In practice, the discretization of Equation 2.5 introduces some regularization that depends on the discretization scheme itself and the only undesirable effects observed are the introduction of staircasing artifacts and the enhancing of impulsive noise with gradients above α [*Weickert and Benhamouda*, 1997].

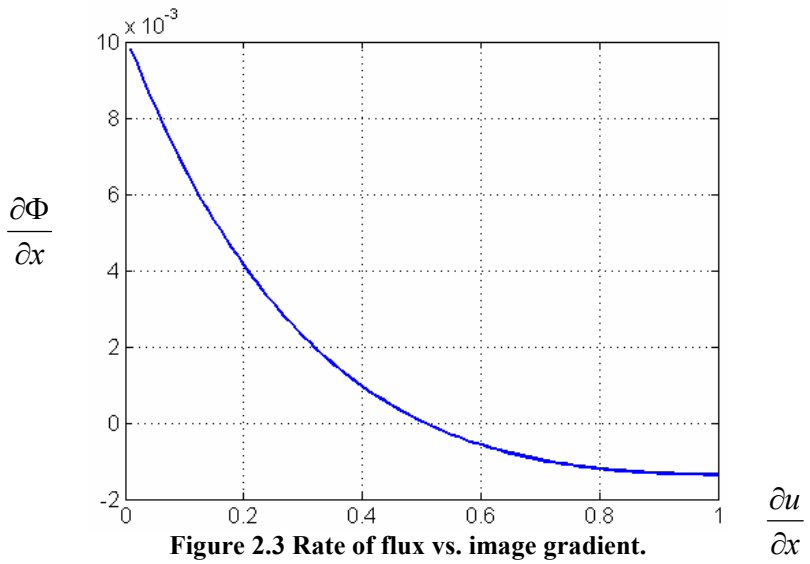


Figure 2.3 Rate of flux vs. image gradient.

The work of [*Catté et al*, 1992] provides a simple way of making Equation 2.5 well-posed, irrespectively of the discretization scheme used, while satisfying all the axioms and properties that a scale-space should posses. The regularized Perona-Malik nonlinear diffusion PDE is given by,

$$\frac{\partial u}{\partial t} = \nabla \bullet \left[g\left(\left| \nabla u_\sigma \right| \right) \nabla u \right], \quad u_\sigma = G\left(0, \sigma^2\right) * u, \tag{2.8}$$

where, $G\left(0, \sigma^2\right)$ is a zero mean Gaussian kernel with variance σ^2. From the previous discussion (Equations 2.2 and 2.3, it can be noticed that u_σ can be obtained by diffusing isotropically the image up to a scale $\sigma^2/2$, with a diffusion coefficient of one. In practice, σ is chosen in such a way that impulsive noise with a high gradient be quickly eliminated without blurring too much the image edges. Weickert proposed later a nonlinear diffusion coefficient for Equation 2.7 that produces segmentation-like results [*Weickert*, 1996], given by,

$$g\left(\left\|\nabla u_\sigma\right\|\right)=\begin{cases}1 & \left\|\nabla u_\sigma\right\|=0\\ 1-e^{-\frac{3.31488}{\left(\left\|\nabla u_\sigma\right\|/\alpha\right)^8}} & \left\|\nabla u_\sigma\right\|>0\end{cases}, \qquad \textbf{2.9}$$

where, constant 3.31488 is the value that makes the flux $g\left(\left\|\nabla u_\sigma\right\|\right)\nabla u$ increasing for $\left\|\nabla u_\sigma\right\|\in(0,\alpha)$ and decreasing for $\left\|\nabla u_\sigma\right\|\in(\alpha,\infty)$. As can be seen from Equations 2.6 and 2.9, $0 < g \leq 1$ as should be expected from a diffusion coefficient.

The major advantage of the linear scale-space approach is that it does not require any prior knowledge on the image and that is why many researchers still use it today, working around its main disadvantage: blurring and dislocation of edges. On the other hand, anisotropic diffusion generates a scale-space that smoothes noise and undesirable variability (small inhomogenities) within objects, while preserving image edges. However, anisotropic diffusion requires the introduction of some prior knowledge represented by the parameters α and σ. In practice, σ and α are set based on the noise level and the strength of the semantically meaningful edges in the image, respectively, which requires the experience of the user with the particular set of images at hand. The automatic selection of α and σ is an open issue that had recently started to being addressed by the scientific community, see for instance [*Black et al*, 1998], [*Mrázek et al*, 2003], and [*Voci et al*, 2004]. In fact, α and σ should also be adaptively selected within the image in such a way that weak edges on a given region might be preserved, while noise and other undesirable low level detail be smoothed out as much as possible.

On the other hand, and even tough Equation 2.8 works fine in practice, it has two main limitations; first, the noise present on strong edges is not eliminated, since diffusion is

reduced there. Second, notice that the diffusion coefficient is nonlinear, but it is the same on all directions, hence, it is locally isotropic and diffusion will be the same in all the directions, on a given point in the image.

An extension of the regularized Perona-Malik PDE was proposed by several authors addressing these issues [*Alvarez et al* 1992, 1994], [*Weickert*, 1996]. Here, we take the general form of the Weickert nonlinear anisotropic diffusion PDE,

$$\frac{\partial u}{\partial t} = \nabla \bullet \left[\mathbf{D}(\nabla u_\sigma)\nabla u\right], \qquad \textbf{2.10}$$

where $\mathbf{D}(\nabla u_\sigma)$ is an 2×2 positive definite diffusion matrix that depends on ∇u_σ. Notice that **D** is the natural extension of the scalar nonlinear diffusion coefficient *g* to a diffusion tensor that can change the flux direction along the direction given by the eigenvectors of **D**. Let $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ be the eigenvectors of **D** and λ_ξ, λ_η their corresponding eigenvalues. If we define,

$$\boldsymbol{\xi} = \frac{\nabla u_\sigma}{|\nabla u_\sigma|}, \langle \boldsymbol{\eta}, \boldsymbol{\xi} \rangle = 0, \lambda_\xi = g(|\nabla u_\sigma|), \lambda_\eta = 1, \qquad \textbf{2.11}$$

then, smoothing along the orthogonal direction to ∇u_σ, given by $\boldsymbol{\eta}$ will be maximal, while smoothing along the gradient direction, given by $\boldsymbol{\xi}$ will be controlled by the nonlinear diffusion coefficient $g(|\nabla u_\sigma|)$ and noise can be eliminated on pixels with high gradient. Notice that smoothing always perpendicular to the gradient direction avoids the destruction of edges in the image, while reducing noise.

Weickert also proposed another PDE that enhance flow-like structures such as fingerprints images, called coherence enhancing diffusion [*Weickert*, 1996],

$$\frac{\partial u}{\partial t} = \nabla \bullet \left[\mathbf{D}\left(J_\rho\right)\nabla u\right], \quad J_\rho = G\left(0,\rho^2\right) * \left(\nabla u_\sigma \nabla u_\sigma^T\right), \tag{2.12}$$

where ∇u_σ^T is the transpose of ∇u_σ, the convolution with the zero mean Gaussian kernel $G\left(0,\rho^2\right)$ is component wise with the elements of the tensor $\nabla u_\sigma \nabla u_\sigma^T$; while, J_ρ is called the structure tensor. Since the eigenvectors of $\nabla u_\sigma \nabla u_\sigma^T$ are $\boldsymbol{\xi}$, $\boldsymbol{\eta}$ (as defined on Equation 2.11, but the eigenvalues are now $\left\|\nabla u_\sigma\right\|^2$ and 0), Equation 2.12 governs a diffusion process that depends on the Gaussian averaged directions $\boldsymbol{\xi}$, $\boldsymbol{\eta}$ within a circular window of size ~3ρ, being ρ the standard deviation of the Gaussian kernel [*Weickert*, 1997]. This PDE considers not only the direction of the image gradient on a single point, but the average within a neighborhood defined by ρ, which allows it to enhance flow-like structures, with a cross-section O(ρ), by defining appropriately the eigenvalues of **D**.

All the previous PDEs have a steady state solution which is a constant image, where all the semantically meaningful structures are lost. Hence, in practice, the user must stop the diffusion process at a final scale $t = T$ that satisfies the requirements of the problem at hand. Alternatively, it has been proposed [*Nördstrom*, 1990] to include a fidelity term on Equation 2.8 that enforces a steady state solution of the PDE, close to the original image,

$$\frac{\partial u}{\partial t} = \nabla \bullet \left[g\left(\left|\nabla u_\sigma\right|\right)\nabla u\right] + \beta\left(u_0 - u\right), \quad u_0 = u(t=0), \quad u_\sigma = G\left(0,\sigma^2\right) * u. \tag{2.13}$$

Notice that on Equation 2.13 the selection of a final scale T is changed for the selection of the parameter β, hence, Equation 2.13 does not avoids the need of selecting a final scale. If for

instance, $\beta \to 0$, we have again Equation 2.8, and if β is large, the steady state of Equation 2.13 would be too close to the original (noisy) image.

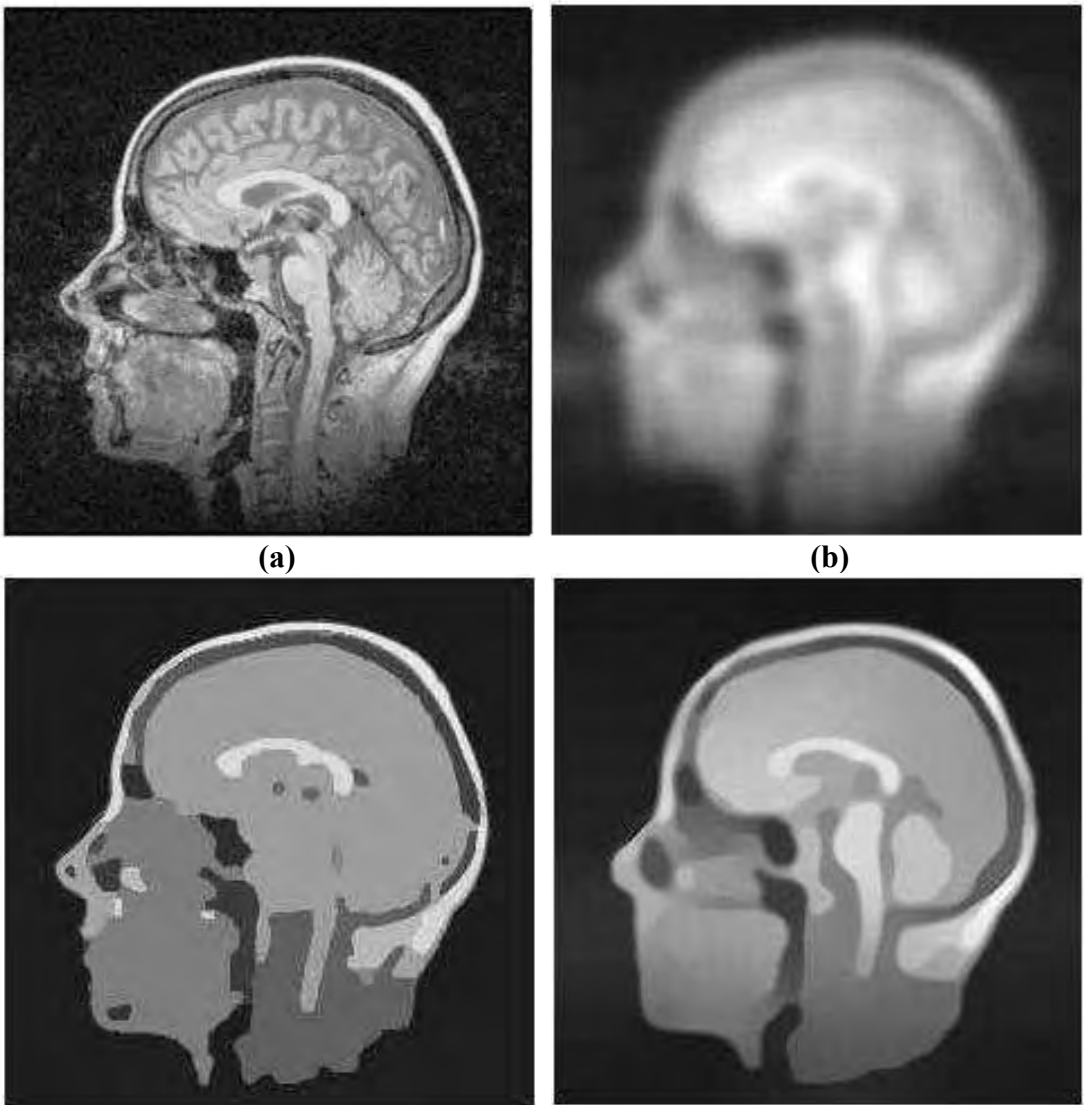
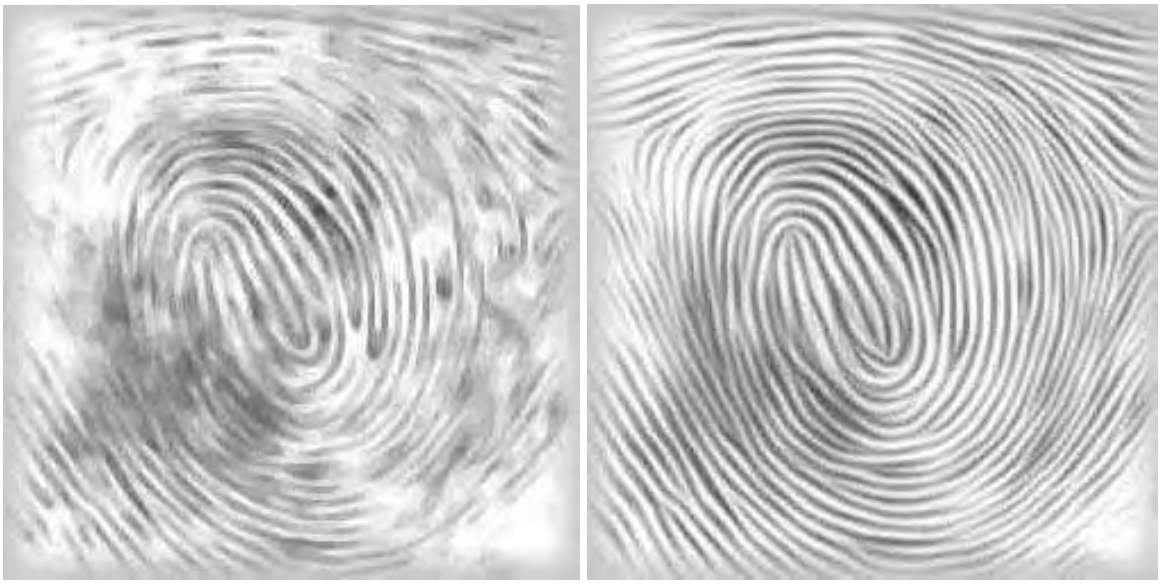


Figure 2.4 a) Original image, smoothed with b) isotropic diffusion, c) nonlinear diffusion, d) nonlinear anisotropic diffusion (taken from [*Weickert*, 1996]).

Figure 2.4 compares the smoothing effect of isotropic diffusion (Equation 2.2), nonlinear diffusion (Equation 2.8), and nonlinear anisotropic diffusion (Equation 2.10). As can be appreciated from these figures, linear isotropic diffusion (Figure 2.4.b) blurs the noise,

but also the edges of the original image (Figure 2.4.a). Nonlinear diffusion (Figure 2.4.c) using the diffusion coefficient given on Equation 2.9 reduces the noise in the image and the smoothed image looks like it were already segmented. However, one can see that some noise remains near the edges on Figure 2.4.c. Figure 2.4.d shows the effect of anisotropic nonlinear diffusion, where noise is reduced the most, but also some features are lost (the shape of the mouth, for instance), while others were preserved better.

Figure 2.5 shows the enhancing effect over flow-like structures that coherence-enhancing diffusion achieves (Equation 2.12).



(a) **(b)**

Figure 2.5 a) Original blurred fingerprint image, b) Enhanced with coherence-diffusion (taken from [*Weickert*, 1996].

There are other PDEs proposed for generating grayscale scale-spaces of the form indicated by Equation 2.1 [*Weickert*, 1996; *Sapiro*, 2001]. However, the equations discussed here are the most widely used, and more importantly, their corresponding discretization schemes have been also studied.

2.3 Image Regularization by Nonlinear Diffusion

The regularizing effect of nonlinear diffusion can be appreciated on Figure 2.6.

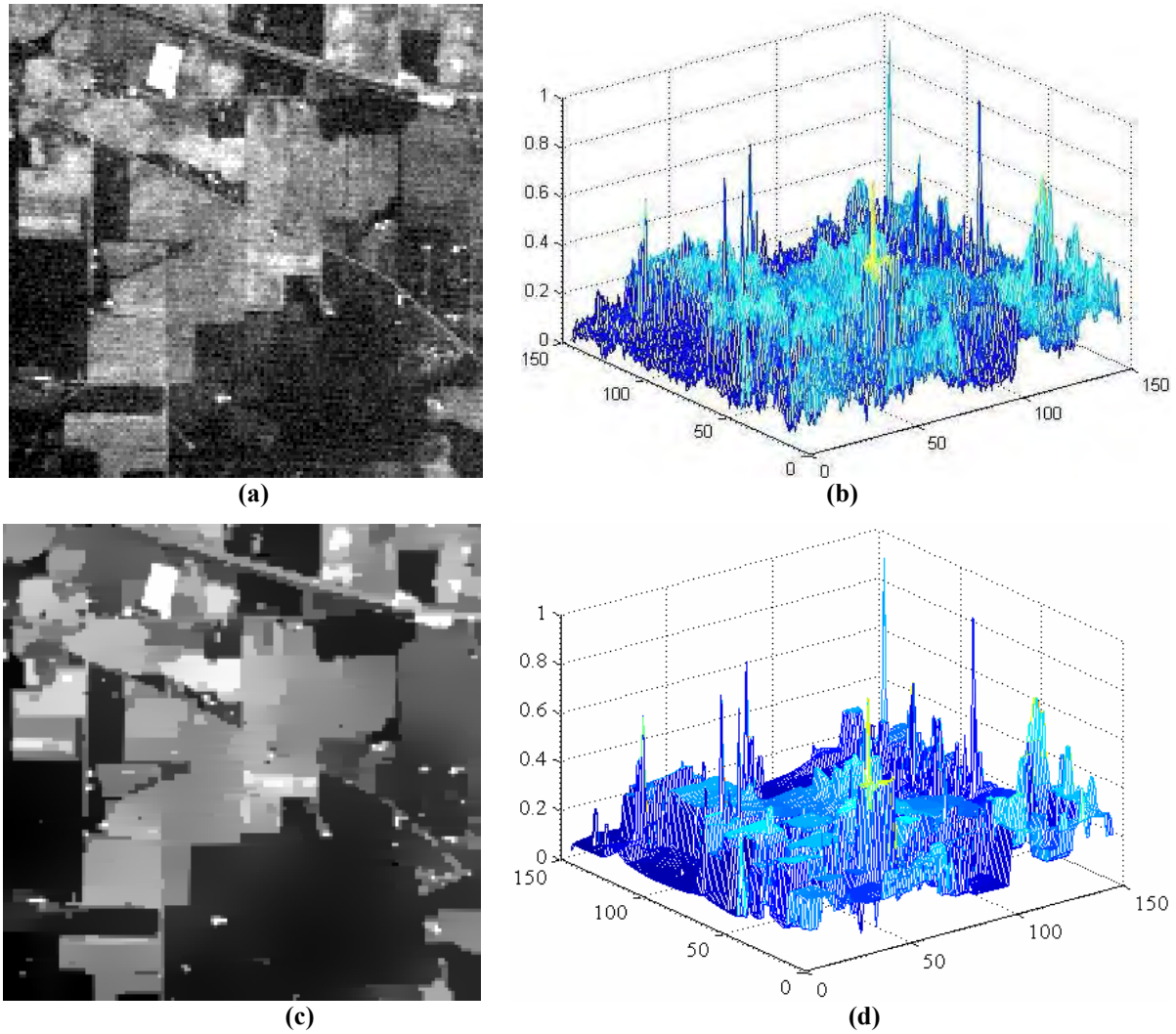


Figure 2.6 Regularization effect of nonlinear diffusion, a) original noisy image, b) intensity of the original image, c) smoothed image, d) intensity of the smoothed image.

Figure 2.6.a shows the noisy spectral band at 419.62 nm, taken with AVIRIS, of the NW Indian Pines test site which consists of 145×145 pixels (see Chapter 3 for further details on this image). Figure 2.6.b shows the surface formed by the grayscale intensity of the original image as a function of the spatial position (in pixels). Figure 2.6.c shows the original noisy

spectral band of Indiana, after nonlinear diffusion (Equations 2.8 and 2.9 with α = 0.022), while Figure 2.6.d shows the surface formed by the intensity of the smoothed image versus the spatial position.

From Figure 2.6.b and Figure 2.6.d it becomes clear that the smoothing effect of the nonlinear diffusion equation can also be interpreted as the nonlinear minimization of the area of the intensity surface.

Regularizing an image can be viewed as the minimization of an energy functional. Let us consider the functional,

$$\|u_0 - u\|^2 + \lambda \int_{\Omega} \hat{g}(u, \mathbf{x}) \partial\Omega . \qquad \textbf{2.14}$$

It is well-known [*Aubert and Kornprobst*, 2002; *Scherzer and Weickert*, 2000; *Weickert*, 1997, 2002; *Sapiro*, 2001] that the Euler-Lagrange equations for the case when $\nabla \hat{g}(u, \mathbf{x}) = g\left(\|\nabla u_\sigma\|^2\right)$ corresponds to the nonlinear diffusion equation (Equation 2.8) and the Euler-Lagrange equations for the case $\hat{g}(u, \mathbf{x}) = \mathbf{x}$, called Tikhonov regularization, corresponds to Gaussian blurring (Equation 2.2). Sapiro and others [*Black et al*, 1998; *Hamza and Krim*, 2001] have also shown that anisotropic diffusion can be seen as a robust estimation procedure that estimates a piecewise smooth image, from the original noisy image. Hence, anisotropic diffusion can be viewed as regularizing an image in such a way that important edges are preserved, while undesirable variations are removed.

2.4 Discretization and Stability of the Nonlinear Diffusion PDE

Solving any of the diffusion equations presented in the previous section requires the discretization of the corresponding PDEs. At first glance this seems a trivial task; however, not every discretization of a PDE generating a continuous scale-space produces an equivalent discrete scale-space, with similar properties as the continuous PDE. Weickert proved the requirements that a discretization of a well-posed continuous scale-space should posses to generate an equivalent discrete scale-space, with similar properties [*Weickert*, 1996, 1998].

The explicit discretization of a diffusion process can be stated, in general, as

$$\begin{aligned} &\mathbf{u}^0 = \mathbf{u}(0), \\ &\mathbf{u}^{k+1} = \mathbf{A}\left(\mathbf{u}_\sigma^k\right)\mathbf{u}^k, \quad k \in [0, K-1] \subset \mathbb{Z}, \end{aligned} \tag{2.15}$$

where, $\mathbf{u}(0)$ is the original image ordered as a vector column $N\times1$, and N is the number of pixels in the image. The scale has been discretized in K steps, such that $t = k\Delta t$, and $T = (K-1)\Delta t$ is the final scale, $\mathbf{A}\left(\mathbf{u}^k\right)$ is an $N\times N$ matrix accounting for the diffusion coefficients among all the pixels in the image. Equation 2.15 is called an explicit discretization of Equation 2.8, because the image at the next scale step, $\mathbf{u}^{k+1}$, can be obtained explicitly from the previous scale step $\mathbf{u}^k$.

Equation 2.15 generates a discrete scale-space if matrix $\mathbf{A}\left(\mathbf{u}^k\right) = [a]_{ij}, i, j \in [0, N-1]$, satisfies the following requirements [*Weickert* 1996, *Weickert and Brox*, 1998], $\forall i, j \in [0, N-1]$,

- Symmetry, $a_{ij} = a_{ji}$.
- Unit row sum, $\sum_{j=0}^{N-1} a_{ij} = 1$.
- Non-negativity, $a_{ij} \geq 0$.
- Positive diagonal, $a_{ii} > 0$.
- Continuity, $\mathbf{A}(\mathbf{u}^k)$ is Lipschitz continuous, i.e. the first derivative of $a_{ij}(u_{ij}^k)$ is bounded (see for instance Equations 2.6, 2.9).
- Irreducibility. A symmetric $N \times N$ matrix can also be viewed in terms of its associated undirected graph, with N vertices and a_{ij} indicates the weight of an edge between vertices i and j. A matrix is irreducible if for each vertex of the associated graph there is a path $l_0, \cdots, l_r$ with $l_0 = i, l_r = j$, such that $a_{l_p, l_{p+1}} > 0$, $0 \leq p < r$, i.e. the graph is fully connected.

These requirements ensure that the discrete scale-space governed by Equation 2.15 is well-posed (see *Weickert*, 1996), satisfies the maximum principle (see Section 2.1), conserves the mean gray value, increases entropy, satisfies consistency with Equation 2.8, and converges to a constant steady state.

The explicit semi-discrete version of Equation 2.8, where only the scale is discretized with a simple forward scheme [*Strikwerda*, 2004] is given by,

$$\frac{u^{k+1} - u^k}{\Delta t} \approx \nabla \bullet \left[g\left(\left\| \nabla u_\sigma^k \right\| \right) \nabla u^k \right] \quad \rightarrow \quad u^{k+1} \approx u^k + \Delta t \left(\nabla \bullet \left[g\left(\left\| \nabla u_\sigma^k \right\| \right) \nabla u^k \right] \right). \qquad \textbf{2.16}$$

It is common to consider $\Delta x = \Delta y = 1$, hence, the spatial discretization of Equation 2.16 leads to the scale-space discretization of Equation 2.8, which can be stated in matrix form as (see *Weickert*, 1998 and Sections 3.2 and 3.3 for further details),

$$\begin{aligned} &\mathbf{u}^0 = \mathbf{u}(0), \\ &\mathbf{u}^{k+1} = \left(\mathbf{I} + \mu \mathbf{G}\left(\mathbf{u}_\sigma^k\right)\right)\mathbf{u}^k, \quad k \in [0, K-1] \subset \mathbb{Z}, \end{aligned} \tag{2.17}$$

where, $\mu = \Delta t$ is the scale-step, $\mathbf{I}$ is the $N \times N$ identity matrix and $\mathbf{G}\left(\mathbf{u}_\sigma^k\right) = [g]_{ij}$ is the $N \times N$ matrix that accounts for the diffusion coefficients and the spatial discretization of the right hand side of Equation 2.16,

$$g_{ij} = \begin{cases} \sum_{k \in \mathrm{N}_i} g_{ik}, & j = i \\ -g_{ij}, & j \in \mathrm{N}_i \\ 0, & else \end{cases},$$

where, N_i is the set of close neighbors to pixel i^{th}.

If we compare Equations 2.15 and 2.17, it is easy to see that $\mathbf{A} = \mathbf{I} + \mu\mathbf{G}$. Hence, $a_{ij} = \mu g_{ij}$ for $i \neq j$ and $a_{ii} = 1 - \mu \sum_{j \neq i} g_{ij}$, which satisfies the unit row sum requirement. The remaining conditions are easy to verify if the diffusion coefficient is defined such that g is Lipschitz continuous, symmetric, and non-negative. Hence, $a_{ij} = \mu g_{ij} > 0$, and $a_{ij} = a_{ji}$ since $g_{ij} = g_{ji}$. The irreducibility comes from the fact that the discretization scheme always connects each pixel with at least four of its nearest neighbors; hence, there is always a path between each pair of pixels and since $g > 0$, the path cannot contain $g = 0$.

However, the condition $a_{ii} > 0$ requires that $1 - \mu \sum_{j \neq i} g_{ij} > 0$, hence,

$$\mu < \frac{1}{\sum_{j \neq i} g_{ij}}. \qquad \textbf{2.18}$$

Since the diffusion coefficient must be $0 < g_{ij} \leq 1$ (see for instance, Equation 2.9) and if we consider only four neighbors, the condition given on Equation 2.18 is always satisfied if $\mu <$ ¼. Notice that if the spatial discretization scheme uses more than four neighbors, Equation 2.18 becomes even more restrictive, for instance if the discretization scheme uses eight neighbors, then $\mu < \frac{1}{8}$.

Equation 2.18 specifies that explicit discretization schemes of the form given by Equation 2.17 cannot have scale-steps larger than ¼. This limitation implies that reaching large scale values requires a large number of iterations, which becomes computationally prohibitive for large data sets, as is the case of hyperspectral imagery.

Alternatively, Equation 2.8 can be discretized in scale as,

$$u^{k+1} - \Delta t\left(\nabla \bullet \left[g\left(\left\|\nabla u_{\sigma}^{k}\right\|\right)\nabla u^{k+1}\right]\right) \approx u^{k}. \qquad \textbf{2.19}$$

Notice that on Equation 2.19 the gradient uses the image at the next iteration step, u^{k+1}, which is unknown, while the diffusion coefficient uses the gradient of the current image, u^{k}, which is known at the start of each iteration. The discretization scheme of equation 2.19 is semi-implicit, since it does not provide explicitly the image at the next scale u^{k+1}. If the diffusion coefficient also uses the unknown image u^{k+1}, then the scheme would be

completely implicit, but it also would be very hard to solve, since Equation 2.19 becomes nonlinear. The advantage of using implicit schemes is that they are very accurate and stable numerically, given that they use as less as possible the (noisy) current value of the image.

Semi-implicit schemes constitute a trade-off between explicit schemes, which are very simple but limited to small scale steps, and implicit schemes which are very accurate and stable at all iteration steps, but hard to solve [*Strickwerda*, 2004]. The scale-space discretization of Equation 2.19 in matrix form is given by,

$$\begin{aligned} &\mathbf{u}^0 = \mathbf{u}(0), \\ &\left(\mathbf{I} - \mu \mathbf{G}\left(\mathbf{u}_\sigma^k\right)\right)\mathbf{u}^{k+1} = \mathbf{u}^k, \quad k \in [0, K-1] \subset \mathbb{Z}. \end{aligned} \tag{2.20}$$

If we compare Equations 2.15 and 2.20, we notice that matrix $\mathbf{A} = (\mathbf{I}\text{-}\mu\mathbf{G})^{-1}$, which means that in order to establish that Equation 2.20 generates a discrete scale-space satisfying the conditions presented earlier, one have to check first that matrix $\mathbf{I}\text{-}\mu\mathbf{G}$ is invertible. If we call $\mathbf{B} = \mathbf{I}\text{-}\mu\mathbf{G} = [b]_{ij}$, then $b_{ij} = -\mu g_{ij}$ for $i \neq j$ and $b_{ii} = 1 + \mu \sum_{i \neq j} g_{ij}$, hence, **B** is strictly diagonally dominant. A well known result from linear algebra is that strictly diagonal dominant matrices are invertible, see e.g. [*Horn and Johnson*, 2006]. Additionally [*Weickert*, 1998], $\mathbf{A} = \mathbf{B}^{-1}$ satisfies all the necessary requirements to create a discrete scale-space, in particular $a_{ii} > 0$ irrespectively of the value of μ, which means that the semi-implicit scheme is numerically stable [*Strickwerda*, 2004] at all iteration steps.

However, the semi-implicit scheme requires solving a linear system (Equation 2.20) at each scale-step and the numerical stability of the semi-implicit schemes does not ensure the accuracy of the computed solution. In fact, the accuracy of the computed solution

degrades as the step size increases, because the diffusion coefficients are taken as the same as the diffusion coefficients of the previous scale, i.e. they are considered as frozen, within each scale step. Hence, a trade-off between the need of using large scale steps to minimize the number of steps to reach a given scale and the need of preserving the accuracy of the computed solution, limits in practice the size of the scale-step that can be used with semi-implicit schemes.

We believe that poor efficiency of the explicit discretization scheme has been the main cause why the formal scale-space has not be used actively to process hyperspectral images. This is the motivation for Chapter 3, where we perform a comparative study of different approximated semi-implicit schemes and the preconditioned conjugated gradient, extended to hyperspectral imagery, in order to explore the feasibility and computational advantages of this approach over traditional explicit schemes.

2.5 Extension to Vector-Valued Images

A hyperspectral image is a especial case of multi-spectral images, where we have hundreds of bands, instead of just tents of bands as in typical multi-spectral images, providing much more spectral information about the physical nature of the underlying scene. A hyperspectral image is of course also a vector-valued image, but a difference of color and multispectral images, the higher information content in the spectral domain increases the possibilities for scale-space representations that reduce progressively the amount of information, preserving as much as possible the semantically meaningful spectral and spatial information at each scale.

The first problem one face trying to extend the scale-space representation of scalar

(grayscale) to vector-valued images is that PDEs governing the scale-space representation of grayscale images rely entirely on the magnitude of the gradient to detect edges on the image. However, the extension of edges in grayscale images to edges in vector-valued images is not straightforward and it is still an open issue.

Early approaches to detecting edges on vector-valued images attempted to combine heuristically the gradient of each spectral band, obtained independently from the other bands [*Sapiro*, 2001]. A well-founded (but certainly, not the only) way to look at gradients on vector-valued images is based on classical Riemannian geometry and was proposed first by [*Di Zenzo*, 1986]. The idea is to consider the square of the infinitesimal Euclidean distance between two vectors as a measure of edge strength. Let $\mathbf{u}(\mathbf{x}):\mathrm{R}^2 \rightarrow \mathrm{R}^M$ be a vector-valued image with M spectral bands. The infinitesimal vectorial distance is given by,

$$d\mathbf{u} = \frac{\partial \mathbf{u}}{\partial x_1} dx_1 + \frac{\partial \mathbf{u}}{\partial x_2} dx_2 ,$$

where, $x_1 = x$, $x_2 = y$. The squared 2-norm of this vector is given by,

$$\left\| d\mathbf{u} \right\|^2 = d\mathbf{u}^T d\mathbf{u} = \sum_{j=1}^{2} \sum_{i=1}^{2} \frac{\partial \mathbf{u}^T}{\partial x_i} \frac{\partial \mathbf{u}}{\partial x_j} dx_i dx_j ,$$

The previous expression can be written in matrix form as,

$$\left\| d\mathbf{u} \right\|^2 = \begin{bmatrix} dx_1 & dx_2 \end{bmatrix} \begin{bmatrix} \sum_{k=1}^{m} \left(\frac{\partial u_k}{\partial x_1} \right)^2 & \sum_{k=1}^{m} \frac{\partial u_k}{\partial x_1} \frac{\partial u_k}{\partial x_2} \\ \sum_{k=1}^{m} \frac{\partial u_k}{\partial x_2} \frac{\partial u_k}{\partial x_1} & \sum_{k=1}^{m} \left(\frac{\partial u_k}{\partial x_2} \right)^2 \end{bmatrix} \begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix} = d\mathbf{x}^T \left(\sum_{i=1}^{m} \nabla u \nabla u^T \right) d\mathbf{x}. \qquad \textbf{2.21}$$

The extremes of $\left\| d\mathbf{u} \right\|^2$ are along the direction given by the eigenvectors of the tensor $\sum_{i=1}^{m} \nabla u \nabla u^T$, defined by the angles θ_+ (maximum change), θ_- (minimum change), while the

values of the extremes are given by the corresponding eigenvalues λ_+ (maximum), and λ_- (minimum) [*Sapiro*, 2001]. Hence, the edge strength can be defined as a function of the extreme values $f(\lambda_+, \lambda_-)$, for instance,

$$f(\lambda_+, \lambda_-) = \lambda_+ + \lambda_- = \sum_{i=1}^{M} \left[\left(\frac{\partial u_i}{\partial x_1} \right)^2 + \left(\frac{\partial u_i}{\partial x_2} \right)^2 \right] = \sum_{i=1}^{M} |\nabla u_i|^2. \quad \textbf{2.22}$$

Notice that Equation 2.22 reduces naturally to $|\nabla u|^2$ for a grayscale image (M = 1). In fact, and as we show on Chapter 3, this similarity metric reduces to the Euclidean distance between vectors. There exists many other possibilities for $f(\lambda_+, \lambda_-)$ as $f(\lambda_+, \lambda_-) = \sqrt{(1+\lambda_+)(1+\lambda_-)}$, which lies within the Beltrami flow framework proposed by [*Kimmel et al*, 2000]. Hence, the extension of the magnitude of the gradient to vector valued images is not unique and it seems more reasonable to speak of similarity metrics between two spectral vectors.

Weickert proposed extensions of Equations 2.8, 2.10, and 2.12 to vector valued images [*Weickert*, 1996, 2002], given by,

$$\frac{\partial u_i}{\partial t} = \nabla \bullet \left[g\left(\sum_{j=1}^{M} |\nabla u_{\sigma,j}|^2 \right) \nabla u_i \right], \quad i = 1, \cdots, M, \quad \textbf{2.23}$$

$$\frac{\partial u_i}{\partial t} = \nabla \bullet \left[\mathbf{D}\left(\sum_{j=1}^{M} \nabla u_{\sigma,j} \nabla u_{\sigma,j}^T \right) \nabla u_i \right], \quad i = 1, \cdots, M, \quad \textbf{2.24}$$

$$\frac{\partial u_i}{\partial t} = \nabla \bullet \left[\mathbf{D}\left(\sum_{j=1}^{M} J_\rho(\nabla u_{\sigma,j}) \right) \nabla u_i \right], \quad i = 1, \cdots, M, \quad \textbf{2.25}$$

respectively, where each band $u_i(\mathbf{x})$: $\mathbb{R}^2 \rightarrow \mathbb{R}$, i = 1, .., M is a grayscale image. A more general anisotropic nonlinear diffusion PDE for vector-valued images, which smoothes the image along θ_-, i.e. the direction of lowest vectorial change was proposed by [*Sapiro and Ringach*, 1996].

$$\frac{\partial u_i}{\partial t} = g\left(f\left(\lambda_+, \lambda_-\right)\right)\frac{\partial^2 u_i}{\partial \theta_-^2}, \quad i = 1, \cdots, M \,. \tag{2.26}$$

Notice that Equation 2.26 can be made equivalent to Equation 2.24, by selecting the eigenvalues of the structure tensor as $g\left(f\left(\lambda_+, \lambda_-\right)\right)$ and zero, such that no diffusion can occur along the direction or greater variability θ_+, while diffusion is allowed along the orthogonal direction given by θ_-. Equations 2.23 to 2.25 are only discrete in the spectral domain; discretization of the scale and spatial domains is needed to solve these PDEs. Details on the discretization of Equation 2.23 are given on Chapter 3.

Another scale-space framework, called direction diffusion, was proposed for vector-valued images [*B. Tang et al*, 2000; *Sapiro*, 2001], based on non-flat manifolds. Basically, the idea is to map the vector valued image $\mathbf{u}(\mathbf{x}, t)$ in $\mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}^M$ to $\hat{\mathbf{u}}(\mathbf{x}, t)$ in $\mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{S}^{M-1}$, where $\mathbb{S}^{M-1}$ is the unit ball in $\mathbb{R}^M$, i.e. we are considering the direction only, not the magnitude of the vectors. Hence, vector valued diffusion becomes direction diffusion, which generates a scale-space with some nice properties, such as discontinuities on the image (edges) are allowed and direction diffusion is not affected by illumination differences on the scene. Further work is required in this area to couple optimally, both magnitude and direction to obtain a scale-space framework for vector-valued images that use all available information in vector-valued images and not just the magnitude or the direction separately.

Notice that on Equations 2.23 to 2.25, the diffusion coefficient or the diffusion tensor is the same for all channels, which prevents that for instance, discontinuities can be created at different image locations, on each channel [*Weickert*, 1996; *Weickert and Brox*, 2002; *Sapiro*, 2001].

2.6 Multiscale Segmentation

The information required for critical image analysis and understanding is usually not represented in terms of pixels, but in the spatial structures, i.e., the homogeneous regions (objects) and their spatial relationships at different image scales [*Gorte*, 1998; *Baatz and Schäpe*, 2000; *Blaschke et al*, 2000, 2001]. Scale-space theory aims to obtain this structure within a formal theory that enables multi-resolution image analysis and multiscale segmentation. Nevertheless, the scale-space theory and the derived multiscale segmentation have been introduced relatively late for multispectral and hyperspectral imagery, in part due to the high dimensionality of the data and heterogeneity (spatial and spectral) of remote sensed images.

In the past few years, several multiscale object-oriented approaches have been proposed for segmenting multispectral imagery, such as the Fractal Net Evolution Approach (FNEA), the linear scale-space of Lindeberg, and Multiscale Object Specific Analysis (MOSA), see [*Hay et al*, 2003] for a review and comparison of these methods. The object-oriented approach consists in generating a multiscale representation of the image based on similarity metrics and hierarchical clustering, but without using a formal definition of scale.

More recently, Bayesian hierarchical clustering using fuzzy trees has been introduced in [*van de Vlag and Stein*, 2007] to segment multispectral images. Adaptive hierarchical

clustering and Support Vector Machines (SVMs) are used in [*Bruzzone and Carlin*, 2006] for supervised classification of high spatial resolution images. Multiscale representations using wavelet shrinkage [*Othman and Qian*, 2006] and morphology [*Plaza et al*, 2007] have been recently extended to hyperspectral imagery. Wavelet shrinkage and morphology have been proven to be equivalent to the continuous scale-space framework generated by PDEs [*Sapiro*, 2001]. However, on implementation many of the geometric properties of the continuous scale-space are missing in the discrete version, by propagation of numerical errors that introduce artifacts as the scale increases [*Durand and Froment*, 2003; *Bosworth and Acton*, 2003] and some nice properties of the formal scale-space as defined on Section 2.2 are lost. Many other algorithms have been also proposed in the past for high spatial resolution multispectral imagery, based on level sets [*Keaton and Brokish*, 2002], histograms [*Silverman et al*, 2004], combination of the spectral and spatial information [*Paclík et al*, 2003] to name just a few.

In this work, we improve and extend to hyperspectral imagery, the fast segmentation algorithm for grayscale images proposed by [*Sharon, et al*, 2000], inspired by Algebraic Multigrid methods (see Chapter 4) and normalized cuts, which is a segmentation algorithm proposed by [*Cox et al*, 1996] and improved later by [*Shi and Malik*, 2000]. Recently, an extension of Sharon's segmentation algorithm has been proposed for multispectral imagery [*Galli and De Candia*, 2005]. Nevertheless, Sharon's segmentation algorithm is based on hierarchical clustering, rather than on the scale-space representation of the image using parabolic (geometric) PDEs. We propose here to integrate the well-founded geometric scale-

space representation of an image using PDEs with a modified version of Sharon's segmentation algorithm that fits perfectly well within this framework.

We integrate in Chapter 4 the well-founded scale-space representation of an image using geometric PDEs, with a modified version of [*Sharon et al*, 2000] segmentation algorithm that fits seamlessly within the scale-space framework.

2.7 Scale Space Representation of Hyperspectral Imagery

The scale space representation of multispectral imagery has been based in the past in hierarchical clustering and the linear scale-space [*Hay et al*, 2003; Navulur, 2007]. Less aware is the remote sensing community of the use of geometric PDEs for the generation of geometric scale-spaces that have well-grounded mathematical and numerical foundations.

Lennon [*Lennon et al*, 2002] used an explicit discretization of the un-regularized version of the Perona-Malik equation (Equation 2.5) extended to vector-valued images, to smooth multispectral imagery. They also show that classification accuracy increases after nonlinear smoothing. In Chapter 3, we extend several well-known semi-implicit methods that enable the extraction of a fast scale-space representation of hyperspectral imagery; based on the semi-implicit discretization of Equation 2.23, see also [*Duarte et al*, 2006, 2007].

2.8 Concluding Remarks

We have introduced in this chapter a formal framework for scale-space image smoothing of grayscale and vectorial images using geometric parabolic PDEs. The scale-space concept is the basis of the object oriented paradigm [Navulur, 2007], however, within this paradigm the scale corresponds to image resolution making this approach heuristic. The formal scale-

space framework presented here provides a well-founded base for the object-oriented paradigm in multi and hyperspectral imagery.

The purpose of the next chapter is to make the formal scale-space framework computationally attractive to process hyperspectral imagery, exploring different numerical methods to solve a slightly modified version of Equation 2.23.

3 Comparative Study of Semi-Implicit Schemes for Nonlinear Diffusion in Hyperspectral Imagery

> I have tried to avoid long numerical computations, thereby following Riemann's postulate that proofs should be given through ideas and not voluminous computations.
> DAVID HILBERT

In this chapter, we show that semi-implicit discretization schemes (see Section 2.4) have better performance (in terms of accuracy and CPU time) than traditional explicit schemes to solve the nonlinear diffusion PDE on hyperspectral imagery. We also show that nonlinear diffusion can be used to reduce the spatial and spectral variability in hyperspectral imagery, improving classification accuracy.

From the different extensions to vector-valued images presented on Section 2.5, we choose the nonlinear diffusion PDE (Equation 2.23) given that is the simplest and most well-known PDE in the literature. It also provides segmentation-like results that facilitate the work of our next objective (Chapter 4), which is segmenting hyperspectral images. We use here a simple variant proposed by [*Perona and Malik*, 1990] for scalar images that introduce some anisotropy in the equation, hence, our PDE can be considered anisotropic. In addition, some of the anisotropic diffusion PDEs proposed in the literature [*Catte et al*, 1992; *Weickert*,

1996] tend to round sharp corners in such a way that, for instance, an object with rectangular shape might be transformed into a rounded rectangle, while in nonlinear diffusion all the edges are preserved. We did not consider the introduction of a fidelity term (Equation 2.13), since it is our belief that penalizing the difference with the original image is not appropriated in general, because it would penalizes the smoothing effect over very noisy images.

3.1 Extension of the Perona-Malik Equation to Vector-Valued Images

Let us represent now a hyperspectral image with M bands and N pixels, in matrix form as,

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_N \end{bmatrix}, \qquad \textbf{3.1}$$

where, each row $\mathbf{u}_i$ corresponds to the sampled spectrum on the i^{th} pixel, $1 \le i \le N$, and the image pixels had been indexed either by rows (row major column format) or by columns (column major format) in such way that matrix $\mathbf{U}$ has N rows corresponding to the number of pixels in the image. Hence, $\mathbf{U}$ is an $N \times M$ matrix.

Now, we will explain in detail the extension of the regularized Perona-Malik equation to vector-valued images. We modify a bit Equation 2.23 as,

$$\frac{\partial \mathbf{u}_i}{\partial t} = \nabla \bullet \left[g\left(\sqrt{\frac{1}{M}\sum_{j=1}^{M} \left\| \nabla u_{\sigma,i,j} \right\|^2} \right) \nabla \mathbf{u}_i \right], \quad i = 1, \cdots, N, \qquad \textbf{3.2}$$

where, we had extended the scalar diffusion coefficient proposed by Weickert (Equation 2.9) to vector-valued images. Notice that by dividing by M on Equation 3.2, we are trying to make the diffusion coefficient independent of the number of bands in the image.

For a 2D vector valued image, we can decompose Equation 3.2 as,

$$\frac{\partial \mathbf{u}_i}{\partial t} = \left(\frac{\partial}{\partial x}\hat{i} + \frac{\partial}{\partial y}\hat{j} \right) \bullet \left[g\left(\sqrt{\frac{1}{M}\sum_{j=1}^{M} \left\| \nabla u_{\sigma,i,j} \right\|^2} \right) \frac{\partial \mathbf{u}_i}{\partial x}\hat{i} + g\left(\sqrt{\frac{1}{M}\sum_{j=1}^{M} \left\| \nabla u_{\sigma,i,j} \right\|^2} \right) \frac{\partial \mathbf{u}_i}{\partial y}\hat{j} \right], \quad i = 1, \cdots, N \,.$$

Perona and Malik [*Perona and Malik*, 1990] proposed to change $\nabla u_{\sigma,i,j}$ by $\frac{\partial u_{\sigma,i,j}}{\partial x}$ on the diffusion coefficient multiplying $\frac{\partial \mathbf{u}_i}{\partial x}$ and by $\frac{\partial u_{\sigma,i,j}}{\partial y}$ on the diffusion coefficient multiplying $\frac{\partial \mathbf{u}_i}{\partial y}$, i.e.,

$$\frac{\partial \mathbf{u}_i}{\partial t} = \frac{\partial}{\partial x}\left(g\left(\sqrt{\frac{1}{M}\sum_{j=1}^{M} \left| \frac{\partial u_{\sigma,i,j}}{\partial x} \right|^2} \right) \frac{\partial \mathbf{u}_i}{\partial x} \right) + \frac{\partial}{\partial y}\left(g\left(\sqrt{\frac{1}{M}\sum_{j=1}^{M} \left| \frac{\partial u_{\sigma,i,j}}{\partial y} \right|^2} \right) \frac{\partial \mathbf{u}_i}{\partial y} \right), \quad i = 1, \cdots, N \,, \tag{3.3}$$

which can be rewritten as,

$$\frac{\partial \mathbf{u}_i}{\partial t} = \frac{\partial}{\partial x}\left(g\left(\left\| \frac{\partial \mathbf{u}_{\sigma,i}}{\partial x} \right\| \right) \frac{\partial \mathbf{u}_i}{\partial x} \right) + \frac{\partial}{\partial y}\left(g\left(\left\| \frac{\partial \mathbf{u}_{\sigma,i}}{\partial y} \right\| \right) \frac{\partial \mathbf{u}_i}{\partial y} \right), \quad i = 1, \cdots, N \,, \tag{3.4}$$

where for simplicity, we had dropped the division by *M*, which will be implicitly assumed from now on. As Perona and Malik argue [*Perona and Malik*, 1990], this change avoids computing the gradient of the image (as in Equation 3.2), which is more expensive than computing the derivative (Equation 3.4) and it introduces some anisotropy to the equation. Indeed, Equation 3.4 can be rewritten as,

$$\frac{\partial \mathbf{u}_i}{\partial t} = \nabla \bullet \left(\begin{bmatrix} g\left(\left\| \frac{\partial \mathbf{u}_{\sigma,i}}{\partial x} \right\| \right) & 0 \\ 0 & g\left(\left\| \frac{\partial \mathbf{u}_{\sigma,i}}{\partial y} \right\| \right) \end{bmatrix} \nabla \mathbf{u}_i \right), \quad i = 1, \cdots, N, \tag{3.5}$$

which is a particular case of the anisotropic diffusion Equation 2.24, with **D** being a diagonal matrix. In particular, we can see that Equation 3.4 has two diffusion components, one along the *x*-axis (first term on the RHS of Equation3.4) and other along the *y*-axis (second term on the RHS of Equation3.4), this allows that the diffusion process vary according to each component independently and hence, diffusion is not isotropic as in 2.23, but also is not as anisotropic as Equation 2.24.

Since the objective of the diffusion coefficient in Equations 3.2 to 3.4 is to prevent diffusion across the boundary between two different regions, the argument θ is not necessarily the image gradient as defined in section 2.5. The diffusion coefficient can be made dependant on any distance measure θ that allows detecting the boundary between two regions. Hence, in general, the diffusion coefficient is given by,

$$g(\theta) = \begin{cases} 1 & \theta = 0 \\ 1 - e^{-\frac{3.31488}{(\theta/\alpha)^8}} & \theta > 0 \end{cases}, \qquad \textbf{3.6}$$

where, $\theta = \sqrt{\|\nabla u_\sigma\|}$ for a grayscale image, or it can be as defined on Equations 3.2 and 3.4, for vector-valued images and α has the same meaning as in Equation 2.9. However, θ could also be a similarity metric that exploits better the higher information content of hyperspectral images, such as the metrics proposed by [*Sweet*, 2003; *Bachman et al*, 2005; *Castrodad et al*, 2007].

In the remainder of this chapter, we will explain in detail the explicit and semi-implicit discretization schemes used to discretize Equation 3.4 and the experiments made to

compare the different discretization schemes in terms of its numerical performance and in terms of their effect on the classification of hyperspectral imagery.

3.2 Discretization Scheme for the Vector-Valued Nonlinear Diffusion Equation

As in [*Perona and Malik*, 1990], we use a forward-time (scale), central-space discretization scheme that is first order accurate in scale and second order accurate in space [*Strikwerda*, 2004] using only the four nearest neighbors. Figure 3.1 shows the finite difference stencil used to discretize Equation 3.4, where the dark circles indicate the pixel (center) and its four nearest neighbors, while the white dashed circles represent intermediate values that are constructions used in the discretization scheme, but do not correspond to real pixels in the image.

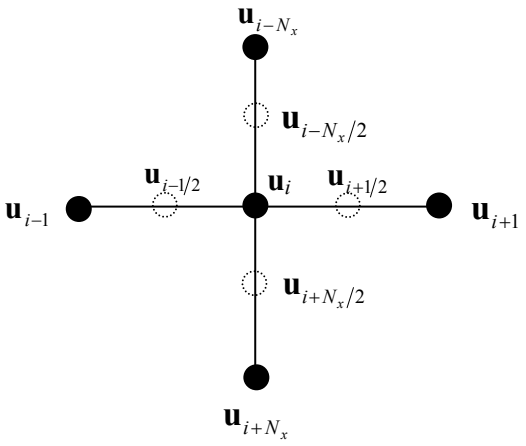


Figure 3.1 Finite Difference Grid of the Discretization Scheme.

Also, we are assuming that the pixels on the image are indexed by rows, where N_y is the number of rows and N_x the number of columns of the image, hence, the left and right neighbors have indices i - 1 and i + 1, respectively, while the up and down neighbors are one

row before, i.e. i - N_x, and one row after, i.e. i + N_x. The intermediate values have non-integer indices that are used only to indicate that they are halfway between real pixels in the image.

Having into account the stencil shown on Figure 3.1, and that $i = 0,\ldots, N\text{-}1$, $k = 0, \ldots, K\text{-}1$, let us discretize Equation 3.4 in two steps. First, using a forward-time, central-space discretization scheme [*Strikwerda*, 2004],

$$\frac{\mathbf{u}_i^{k+1}-\mathbf{u}_i^k}{\Delta t}=\frac{\partial}{\partial x}\left(g\left(\frac{\left\|\mathbf{u}_{\sigma,i+1/2}^k-\mathbf{u}_{\sigma,i-1/2}^k\right\|}{\Delta x}\right)\frac{\mathbf{u}_{i+1/2}^k-\mathbf{u}_{i-1/2}^k}{\Delta x}\right)+\frac{\partial}{\partial y}\left(g\left(\frac{\left\|\mathbf{u}_{\sigma,i+N_x/2}^k-\mathbf{u}_{\sigma,i-N_x/2}^k\right\|}{\Delta y}\right)\frac{\mathbf{u}_{i+N_x/2}^k-\mathbf{u}_{i-N_x/2}^k}{\Delta y}\right).$$

Second, using central-space discretization again for the remaining spatial derivatives,

$$\frac{\mathbf{u}_i^{k+1}-\mathbf{u}_i^k}{\Delta t}=\frac{g\left(\frac{\left\|\mathbf{u}_{\sigma,i+1}^k-\mathbf{u}_{\sigma,i}^k\right\|}{\Delta x}\right)\frac{\mathbf{u}_{i+1}^k-\mathbf{u}_i^k}{\Delta x}-g\left(\frac{\left\|\mathbf{u}_{\sigma,i}^k-\mathbf{u}_{\sigma,i-1}^k\right\|}{\Delta x}\right)\frac{\mathbf{u}_i^k-\mathbf{u}_{i-1}^k}{\Delta x}}{\Delta x}+\frac{g\left(\frac{\left\|\mathbf{u}_{\sigma,i+N_x}^k-\mathbf{u}_{\sigma,i}^k\right\|}{\Delta y}\right)\frac{\mathbf{u}_{i+N_x}^k-\mathbf{u}_i^k}{\Delta y}-g\left(\frac{\left\|\mathbf{u}_{\sigma,i}^k-\mathbf{u}_{\sigma,i-N_x}^k\right\|}{\Delta y}\right)\frac{\mathbf{u}_i^k-\mathbf{u}_{i-N_x}^k}{\Delta y}}{\Delta y}. \quad \textbf{3.7}$$

As it was defined on the previous chapter, the boundary conditions require that there is no flux outgoing or incoming trough the image boundary $\partial\Omega$, hence on implementation, the flux terms $g(\mathbf{.})(\Delta\mathbf{u})$ on Equation 3.7 with calls to indices outside the boundary of the image are made zero.

Now, solving for $\mathbf{u}_i^{k+1}$ on Equation 3.7,

$$\mathbf{u}_i^{k+1}=\mathbf{u}_i^k+\mu\left[g_{i,E}^k\left(\mathbf{u}_{i+1}^k-\mathbf{u}_i^k\right)-g_{i,W}^k\left(\mathbf{u}_i^k-\mathbf{u}_{i-1}^k\right)+g_{i,N}^k\left(\mathbf{u}_{i+N_x}^k-\mathbf{u}_i^k\right)-g_{i,S}^k\left(\mathbf{u}_i^k-\mathbf{u}_{i-N_x}^k\right)\right], \quad \textbf{3.8}$$

where, $\mu = \Delta t$, $\Delta x = \Delta y = 1$, and

$$g_{i,E}^k \equiv g\left(\left\|\mathbf{u}_{\sigma,i+1}^k - \mathbf{u}_{\sigma,i}^k\right\|\right), \quad g_{i,W}^k \equiv g\left(\left\|\mathbf{u}_{\sigma,i}^k - \mathbf{u}_{\sigma,i-1}^k\right\|\right),$$
$$g_{i,N}^k \equiv g\left(\left\|\mathbf{u}_{\sigma,i+N_x}^k - \mathbf{u}_{\sigma,i}^k\right\|\right), \quad g_{i,S}^k \equiv g\left(\left\|\mathbf{u}_{\sigma,i}^k - \mathbf{u}_{\sigma,i-N_x}^k\right\|\right) . \qquad \textbf{3.9}$$

Notice that on Equation 3.9, the argument of the diffusion coefficient (θ) corresponds to the Euclidean distance between vector-valued pixels along the *x* and *y* axis.

Reorganizing Equation 3.9 a bit more,

$$\mathbf{u}_i^{k+1} = \mathbf{u}_i^k\left[1-\mu\left(g_{i,E}^k + g_{i,W}^k + g_{i,N}^k + g_{i,S}^k\right)\right] + \mu\left(g_{i,E}^k\mathbf{u}_{i+1}^k + g_{i,W}^k\mathbf{u}_{i-1}^k + g_{i,N}^k\mathbf{u}_{i+N_y}^k + g_{i,S}^k\mathbf{u}_{i-N_y}^k\right) . \qquad \textbf{3.10}$$

Equation 3.10 corresponds to the explicit discretization of Equation 3.4, which in matrix-vector form can be summarized as,

$$\mathbf{U}^{k+1} = \left(\mathbf{I} + \mu\mathbf{G}^k\right)\mathbf{U}^k, \quad k = 0,\cdots,K-1, \qquad \textbf{3.11}$$

where **I** and **G** are $N\times N$ matrices, being **I** the identity matrix, and matrix **G** accounts for the diffusion coefficients on each pixel. From Equation 3.10, it is easy to see that $\mathbf{I} + \mu\mathbf{G}^k$ is a matrix with only five diagonals, and hence, there is no need to store $\mathbf{I} + \mu\mathbf{G}^k$, only the diffusion coefficients (Equation 3.9).

From Equation 3.10, it is easy to derive the semi-implicit scheme. Let us consider now that the discrete differences on the RHS of Equation 3.8 are taken at the next scale *k* + 1, i.e.

$$\mathbf{u}_i^{k+1} = \mathbf{u}_i^k + \mu\left[g_{i,E}^k\left(\mathbf{u}_{i+1}^{k+1} - \mathbf{u}_i^{k+1}\right) - g_{i,W}^k\left(\mathbf{u}_i^{k+1} - \mathbf{u}_{i-1}^{k+1}\right) + g_{i,N}^k\left(\mathbf{u}_{i+N_y}^{k+1} - \mathbf{u}_i^{k+1}\right) - g_{i,S}^k\left(\mathbf{u}_i^{k+1} - \mathbf{u}_{i-N_y}^{k+1}\right)\right] . \qquad \textbf{3.12}$$

Reorganizing Equation 3.12,

$$\left[1+\mu\left(g_{i,E}^{k}+g_{i,W}^{k}+g_{i,N}^{k}+g_{i,S}^{k}\right)\right]\mathbf{u}_{i}^{k+1}-\mu\left(g_{i,E}^{k}\,\mathbf{u}_{i+1}^{k+1}+g_{i,W}^{k}\,\mathbf{u}_{i-1}^{k+1}+g_{i,N}^{k}\,\mathbf{u}_{i+N_y}^{k+1}+g_{i,S}^{k}\,\mathbf{u}_{i-N_y}^{k+1}\right)=\mathbf{u}_{i}^{k} \quad . \qquad \textbf{3.13}$$

In matrix-vector format,

$$\left(\mathbf{I}-\mu\mathbf{G}^{k}\right)\mathbf{U}^{k+1}=\mathbf{U}^{k},\quad k=0,\cdots,K-1. \qquad \textbf{3.14}$$

which corresponds to the semi-implicit discretization of Equation .

The semi-implicit scheme indicated on Equation 3.14 requires solving a linear system of equations at each scale-step, *k*. Given that matrix $\mathbf{I}-\mu\mathbf{G}^{k}$ have only five non-zero diagonals, some approximations to the exact solution of Equation 3.14 have been proposed in the past for scalar images. Alternatively, Equation 3.14 can be solved using the Preconditioned Conjugated Gradient (PCG) method, which is a fast iterative method that can obtain very accurate solutions, when $\mathbf{I}-\mu\mathbf{G}^{k}$ is positive definite (see Section 3.2.2).

3.2.1 Approximations with Semi-implicit Schemes

Let us decompose the matrix of diffusion coefficients **G** into its components along the *x* and *y* axis, i.e. $\mathbf{G} = \mathbf{G}_x + \mathbf{G}_y$, where $\mathbf{G}_x$ contains only the g_E and g_W diffusion coefficients and $\mathbf{G}_y$ contains only g_N and g_S (see Equation 3.10). Hence, Equation 3.14 can also be expressed as,

$$\left(\mathbf{I}-\mu\mathbf{G}_{x}^{k}-\mu\mathbf{G}_{y}^{k}\right)\mathbf{U}^{k+1}=\mathbf{U}^{k}. \qquad \textbf{3.15}$$

The simplest semi-implicit approximation to Equation 3.15 is given by the Alternating Direction Method, Locally One Dimensional or ADI-LOD [*Strikwerda*, 2004; *Barash et al*, 2003],

$$\left(\mathbf{I}-\mu\mathbf{G}_x^k\right)\left(\mathbf{I}-\mu\mathbf{G}_y^k\right)\approx\mathbf{I}-\mu\mathbf{G}_x^k-\mu\mathbf{G}_y^k.$$

Hence, an approximated solution to Equation 3.14 can be found by solving first a system which considers diffusion only along one direction and then diffusion along the other direction (i.e. locally one-dimensional),

$$\begin{cases}\left(\mathbf{I}-\mu\mathbf{G}_x^k\right)\mathbf{U}^{k+1/2}=\mathbf{U}^k\\ \left(\mathbf{I}-\mu\mathbf{G}_y^k\right)\mathbf{U}^{k+1}\approx\mathbf{U}^{k+1/2}\end{cases}. \qquad \textbf{3.16}$$

Given that $\mathbf{I}-\mu\mathbf{G}_x^k$ and $\mathbf{I}-\mu\mathbf{G}_y^k$ are tri-diagonal matrices, the two systems indicated on Equation 3.16 can be solved in linear time complexity, O(*NM*), using the Thomas algorithm [*Weickert et al*, 1998], which can be extended to vector valued images in a straightforward way (see Appendix A).

Another approximation is given by the Douglas-Rachford ADI method [*Strikwerda*, 2004], which can be extended to vector-valued images as (see Appendix B1 for more details),

$$\begin{cases}\left(\mathbf{I}-\mu\mathbf{G}_x^k\right)\mathbf{U}^{k+1/2}=\left(\mathbf{I}+\mu\mathbf{G}_y^k\right)\mathbf{U}^k\\ \left(\mathbf{I}-\mu\mathbf{G}_y^k\right)\mathbf{U}^{k+1}\approx\mathbf{U}^{k+1/2}-\mu\mathbf{G}_y^k\mathbf{U}^k\end{cases}, \qquad \textbf{3.17}$$

where, the vector-valued Thomas algorithm (Appendix A) can be used again to solve these two equations in linear time complexity (see Section 3.1.3).

Finally, another approximation to Equation 3.14 is given by the Peaceman-Rachford method [*Strikwerda*, 2004; *Barash et al*, 2003], extended here to vector-valued images as (see Appendix B1 for more details),

$$\begin{cases} \left(\mathbf{I}-\frac{1}{2}\mu\mathbf{G}_x^k\right)\mathbf{U}^{k+1/2}=\left(\mathbf{I}+\frac{1}{2}\mu\mathbf{G}_y^k\right)\mathbf{U}^k \\ \left(\mathbf{I}-\frac{1}{2}\mu\mathbf{G}_y^k\right)\mathbf{U}^{k+1}\approx\left(\mathbf{I}+\frac{1}{2}\mu\mathbf{G}_x^k\right)\mathbf{U}^{k+1/2} \end{cases}, \qquad \textbf{3.18}$$

where, the Thomas algorithm can also be employed to solve both tri-diagonal systems.

The previous ADI methods have the disadvantage of being dependant on the order of axis (x or y) chosen to factorize matrix $\mathbf{I}-\mu\mathbf{G}^k$, hence, they are not rotationally invariant [*Barash et al*, 2003] and require solving two linear systems in sequence. A different approach is used by the Additive Operator Splitting (AOS) proposed by [*Weickert et al*, 1998], which has the advantage of being rotationally invariant and it consists of two decoupled systems that can be solved in parallel. The AOS applied to vector-valued images (see Appendix B1) is given by,

$$\begin{aligned} &\left(\mathbf{I}-2\mu\mathbf{G}_x^k\right)\mathbf{U}_x^{k+1}=\mathbf{U}^k, \quad \left(\mathbf{I}-2\mu\mathbf{G}_y^k\right)\mathbf{U}_y^{k+1}=\mathbf{U}^k \\ &\mathbf{U}^{k+1}\approx\frac{\mathbf{U}_x^{k+1}+\mathbf{U}_y^{k+1}}{2} \end{aligned}, \qquad \textbf{3.19}$$

Each of the two systems on Equation 3.19 can be solved efficiently using the Thomas algorithm.

3.2.2 Preconditioned Conjugated Gradient Method

The conjugated gradient (CG) method is an efficient iterative method to solve sparse systems of linear equations whose matrix is symmetric positive definite [*Strikwerda*, 2004]. On Appendix B2, we show that matrix $\mathbf{A}=\mathbf{I}-\mu\mathbf{G}$ on Equation 3.14 is positive definite and hence, the CG method will solve Equation 3.14 efficiently. In this section, we will drop the dependence of **G** and **A** on the scale step k, to avoid confusion with the steps of the PCG

algorithm.

Since **A** is a matrix of size $N \times N$, the CG method will always converge to the exact solution of Equation 3.14 in N iterations [*Saad*, 2003]. However, for large datasets, as is the case of Hyperspectral imagery (see Section 1.1) N can be very large, and in practice, the CG must be stopped when the initial error is reduced below a given tolerance. The reduction of the initial error in the CG method depends on the condition number κ of matrix **A**, as [*Schewchuk*, 1994; *Saad*, 2003],

$$\frac{\left\langle \mathbf{e}^i, \mathbf{A}\mathbf{e}^i \right\rangle}{\left\langle \mathbf{e}^0, \mathbf{A}\mathbf{e}^0 \right\rangle} \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^i, \qquad \textbf{3.20}$$

where $\mathbf{e}^i$ is the error, at the i^{th} iteration. From Equation 3.20, one can see that the number of iterations required to reach a given tolerance increase as the condition number of the matrix increases. As shown on Appendix B2, the condition number of matrix **A** increases proportionally to μ and the threshold parameter α (see Equation 3.6). In practice, for large data sets as is the case of hyperspectral imagery, we want to use large values of μ and hence, the matrix's condition number must be improved by pre-conditioning. The basic idea of preconditioning is to replace the system $\mathbf{A}\mathbf{u}^{k+1} = \mathbf{u}^k$ by [*Strikwerda*, 2004],

$$\mathbf{C}^{-1}\mathbf{A}\mathbf{u}^{k+1} = \mathbf{C}^{-1}\mathbf{u}^k, \qquad \textbf{3.21}$$

where, matrix **C** is called the preconditioner of **A** and $\mathbf{C}^{-1}\mathbf{A}$ is a matrix with better condition number than **A**, such that the conjugated gradient method converges faster. In general, $\mathbf{C}^k$ should approximate **A** such that $\mathbf{C}^{-1}\mathbf{A} \approx \mathbf{I}$, and the product $\mathbf{C}^{-1}\mathbf{u}$ must be easily computed, for

any vector **u**. Since **A** is the same for all image bands on a hyperspectral image, the extension of PCG to hyperspectral images is given by,

$$\mathbf{C}^{-1}\mathbf{A}\mathbf{U}^{k+1} = \mathbf{C}^{-1}\mathbf{U}^{k}. \qquad \textbf{3.22}$$

The problem of preconditioning is to improve convergence enough to make up for the cost of computing $\mathbf{C}^{-1}\mathbf{U}$, on each iteration of the CG method.

Solving Equation 3.22 independently for each band on a hyperspectral image with the CG method is too expensive, since convergence would vary from one band to the next. In order to speed up the process and use better the capabilities of Matlab, we propose here to update all image bands, simultaneously. More precisely, let us consider the general linear system $\mathbf{AX} = \mathbf{B}$, where **X** and **B** are dense matrices of size $N \times M$. In our case, $\mathbf{B} = \mathbf{U}^k$ the smoothed hyperspectral image at scale k, and $\mathbf{X} = \mathbf{U}^{k+1}$ the smoothed hyperspectral image at scale k+1. Hence, **X** and **B** have the form,

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_N \end{pmatrix},$$

where $\mathbf{x}_i$, $\mathbf{b}_i$ are spectral vectors of length M and **A** is a positive definite matrix, which in our case is sparse, containing only five diagonals. The residual $\mathbf{R} = \mathbf{B} - \mathbf{AX}$ has the same structure as **X** and **B** and also the auxiliary matrices **P** and **Q** used in the CG algorithm, see [*Strikwerda*, 2004; *Barret et al*, 1994]. However, we can look **R** (also **P** and **Q**) in terms of the residual of the different grayscale images that make up the hyperspectral image, i.e.

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_M \end{bmatrix},$$

where, $\mathbf{r}_j$ is the residual on the j^{th} band.

The extension of the PCG algorithm to vector-valued images is given on Equation 3.23, where n indicates the iteration step within the CG algorithm. We call this algorithm here as PCG-vectorial. Notice that scalars α, β on the standard PCG algorithm are changed here to vectors of size M, since we are updating M images simultaneously.

$$
\begin{aligned}
&\mathbf{X}^0 = \mathbf{B}, \quad \mathbf{R}^0 = \mathbf{B} - \mathbf{A}\mathbf{X}^0, \quad \overline{\mathbf{b}} = \frac{1}{M}\sum_{\mathbf{j=1}}^{\mathbf{M}} \mathbf{b_j}, \quad \boldsymbol{\rho}^1 = \mathbf{1} \\
&\textit{for } n = 1, 2, \cdots, \max\_\text{iterations} \\
&\quad \mathbf{Z}^{n-1} = \mathbf{C}^{-1}\mathbf{R}^{n-1} \\
&\quad \boldsymbol{\rho}^{n-1} = \boldsymbol{\rho}^n \\
&\quad \boldsymbol{\rho}^n = \left[\rho_j^n\right]_{j=1\ldots M}, \quad \rho_j^n = \left\langle \mathbf{r}_j^{n-1}, \mathbf{z}_j^{n-1} \right\rangle \\
&\quad \textit{if } n = 1 \\
&\qquad \mathbf{P}^1 = \mathbf{Z}^0 \\
&\quad \textit{else} \\
&\qquad \boldsymbol{\beta}^{n-1} = \left[\beta_j^{n-1}\right]_{j=1\ldots M}, \quad \beta_j^{n-1} = \rho_j^n \big/ \rho_j^{n-1} \\
&\qquad \mathbf{P}^n = \left[\mathbf{z}_j^{n-1} + \beta_j^{n-1}\mathbf{p}_j^{n-1}\right]_{j=1\ldots M} \\
&\quad \mathbf{Q}^n = \mathbf{A}\mathbf{P}^n, \\
&\quad \boldsymbol{\alpha}^n = \left[\alpha_j^n\right]_{j=1\ldots M}, \ \alpha_j^n = \frac{\rho_j^n}{\left\langle \mathbf{p}_j^n, \mathbf{q}_j^n \right\rangle} \\
&\quad \mathbf{X}^n = \left[\mathbf{x}_j^{n-1} + \alpha_j^n \mathbf{p}_j^n\right]_{j=1\ldots M} \\
&\quad \mathbf{R}^n = \left[\mathbf{r}_j^{n-1} - \alpha_j^n \mathbf{q}_j^n\right]_{j=1\ldots M} \\
&\quad \text{error} = \left\| \overline{\mathbf{b}} - \mathbf{A}\overline{\mathbf{x}}^n \right\|, \quad \overline{\mathbf{x}} = \frac{1}{M}\sum_{j=1}^{M} \mathbf{x}_j^n, \\
&\quad \text{if error} < \text{tolerance}, \\
&\qquad \text{exit.}
\end{aligned}
\qquad \textbf{3.23}
$$

Another difference is how we compute the error in order to stop the algorithm. The error in Matlab's PCG routine[1] (pcg.m) is computed as $\left\|\mathbf{b}-\mathbf{A}\mathbf{x}^n\right\|$, i.e. the current true residual (which is different from the estimated residual $\mathbf{r}^n$). The straightforward extension of this residual term to vector-valued images would be $\mathbf{R}=\mathbf{B}-\mathbf{A}\mathbf{X}^n$, which is expensive to compute and left us with the problem of how to compute the error. Hence, we decided to stop the algorithm using the mean value of the image, as indicated on Equation 3.23, which is computationally cheaper. Section 3.3.2 shows the speedup achieved by the vectorization of the code, while the accuracy remains as good as running the PCG band by band.

The simplest preconditioner for $\mathbf{A} = \mathbf{I} - \mu\mathbf{G}$ is based on the Symmetric Successive Over-Relaxation method (SSOR) method, which has an explicit formula for the preconditioner $\mathbf{C}$ [*Strikwerda*, 2004],

$$\mathbf{C}=\frac{1}{\omega(2-\omega)}(\mathbf{I}-\omega\mathbf{L})(\mathbf{I}-\omega\mathbf{L}^T), \quad 0<\omega<2, \tag{3.24}$$

where, $\mathbf{A} = \mathbf{I} - \mathbf{L} - \mathbf{L}^T$, and hence, $\mathbf{L}$ is the lower triangular part of $\mu\mathbf{G}$, $\mathbf{L}^T$ the upper triangular part, and $\omega \in (0,2)$ is a parameter that must be settled manually. Hence, $\mathbf{Z}=\mathbf{C}^{-1}\mathbf{R}$ on Equation 3.23 can be obtained as,

$$\frac{1}{\omega(2-\omega)}(\mathbf{I}-\omega\mathbf{L})(\mathbf{I}-\omega\mathbf{L}^T)\mathbf{Z}=\mathbf{R}. \tag{3.25}$$

Since, $\mathbf{I} - \omega\mathbf{L}$ is lower triangular and $\mathbf{I} - \omega\mathbf{L}^T$ is upper triangular, the linear system indicated on Equation 3.25 can be solved in linear time using simple forward and backward substitution as,

[1] The pcg.m routine in Matlab was created by Penny Anderson, 1996, working at Mathworks.

$$\begin{cases} \dfrac{1}{\sqrt{\omega(2-\omega)}}\left(\mathbf{I}-\omega\mathbf{L}\right)\mathbf{Y}=\mathbf{R} \\ \dfrac{1}{\sqrt{\omega(2-\omega)}}\left(\mathbf{I}-\omega\mathbf{L}^{T}\right)\mathbf{Z}=\mathbf{Y} \end{cases}. \qquad \textbf{3.26}$$

Another, preconditioner commonly used in practice is the incomplete Cholesky factorization that approximates $\mathbf{A}$ as $\mathbf{A}\approx\widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^{T}=\mathbf{C}$, being $\widetilde{\mathbf{L}}$ a lower triangular matrix. In our work, we use the incomplete Cholesky factorization with "0" drop tolerance as indicated in [*Saad*, 2003], which means that $\widetilde{\mathbf{L}}$ has the same sparsity pattern as the lower triangular part of $\mathbf{A}$. Since, the numerical factorization of $\mathbf{A}$ might be as expensive as one iteration of the CG method [*Barret et al*, 1994], we use instead the analytical factorization proposed by [*Saad*, 2003] (see Appendix B). Hence, $\mathbf{Z}=\mathbf{C}^{-1}\mathbf{R}$ on Equation 3.23 is computed by forward and backward substitution as,

$$\begin{cases} \widetilde{\mathbf{L}}\mathbf{Y}=\mathbf{R} \\ \widetilde{\mathbf{L}}^{T}\mathbf{Z}=\mathbf{Y} \end{cases}, \qquad \textbf{3.27}$$

ADI-LOD can also provides another preconditioner, since it approximates matrix $\mathbf{A}$, as

$$\mathbf{C}=\left(\mathbf{I}-\mu\mathbf{G}_{x}\right)\left(\mathbf{I}-\mu\mathbf{G}_{y}\right)\approx\mathbf{A}\ .$$

Hence, $\mathbf{Z}=\mathbf{C}^{-1}\mathbf{R}$ can be computed using the ADI-LOD approximation as preconditioner,

$$\begin{cases} \left(\mathbf{I}-\mu\mathbf{G}_{x}\right)\mathbf{Y}=\mathbf{R} \\ \left(\mathbf{I}-\mu\mathbf{G}_{y}\right)\mathbf{Z}=\mathbf{Y} \end{cases}. \qquad \textbf{3.28}$$

Similarly, AOS also can be used as preconditioner, since,

$$\mathbf{C}^{-1} = \frac{1}{2}\left[\left(\mathbf{I} - 2\mu\mathbf{G}_x\right)^{-1} + \left(\mathbf{I} - 2\mu\mathbf{G}_y\right)^{-1}\right] \approx \mathbf{A}^{-1}$$

Hence, $\mathbf{Z} = \mathbf{C}^{-1}\mathbf{R}$ on Equation 3.23 can be computed as,

$$\begin{aligned} &\left(\mathbf{I} - 2\mu\mathbf{G}_x\right)\mathbf{Z}_x = \mathbf{R}, \quad \left(\mathbf{I} - 2\mu\mathbf{G}_y\right)\mathbf{Z}_y = \mathbf{R}, \\ &\mathbf{Z} = \frac{1}{2}\left[\mathbf{Z}_x + \mathbf{Z}_y\right] \end{aligned} \quad . \qquad \textbf{3.29}$$

The Peaceman-Rachford and Douglas-Rachford ADI schemes are more expensive computationally and more sensitive to the scale step than ADI-LOD and AOS, hence, they are not used here as preconditioners.

Additionally, ADI and AOS schemes, used as preconditioners, require the inclusion of a reduction factor ρ [*Castillo and Saad*, 1998], in order to avoid instability on the conjugated gradient method at high scale steps. If we want to find $\mathbf{Z} = \mathbf{C}^{-1}\mathbf{R}$, then

$$\text{ADI} - \text{LOD}: \begin{cases} \left(\mathbf{I} - \rho\mu\mathbf{G}_x\right)\mathbf{Y} = \mathbf{R} \\ \left(\mathbf{I} - \rho\mu\mathbf{G}_y\right)\mathbf{Z} = \mathbf{Y} \end{cases} . \qquad \textbf{3.30}$$

$$\text{AOS}: \begin{cases} \left(\mathbf{I} - 2\rho\mu\mathbf{G}_x\right)\mathbf{Z}_x = \mathbf{R}, \quad \left(\mathbf{I} - 2\rho\mu\mathbf{G}_y\right)\mathbf{Z}_y = \mathbf{R} \\ \mathbf{Z} \approx \frac{1}{2}\left(\mathbf{Z}_x + \mathbf{Z}_y\right) \end{cases} . \qquad \textbf{3.31}$$

The CG method can be accelerated by choosing an initialization close to the actual solution. Given the good performance and low computational cost of ADI-LOD (see Section 3.2.1), we use it to initialize the best PCG method we found (PCG using Cholesky factorization), and we call it here PCG-ADI-LOD (see Section 3.2.1).

3.2.3 Time and Disk Space Complexity

Let us analyze now, the time complexity and disk space requirements of the methods

presented on previous section. Since the number of scale-steps is given by $K = T/\mu$, with T the final scale (see Section 2.4), then the total time complexity is K times the time complexity of a single scale-step. Given the large values that the semi-implicit schemes allows to use for the scale-step, μ, while preserving good accuracy of the computed solution (see Section 3.3), 2-5 scale-steps are enough to smooth common hyperspectral images. Since, in practice $K<<M<<N$ then K can be considered O(1) and the time complexity of the algorithms presented in previous section is dominated by the time complexity at each scale-step.

The computation of the diffusion coefficients is a common step on all the algorithms presented in previous section. The diffusion coefficients in Equation 3.9 can be found in O(M) time, since they only require to compute the 2-norm of the difference between two spectral vectors. Also, the disk space requirements of all the algorithms discussed on previous section are ~2NM, since the hyperspectral image $\mathbf{U}^k$ is of size NM as well as the Gaussian smoothed image $\mathbf{U}_\sigma^k$. Notice that on each scale step $\mathbf{U}^k$ and $\mathbf{U}_\sigma^k$ are overwritten, hence, no extra disk space storage is needed.

Let us analyze the time complexity of the explicit scheme. From Equation 3.10, it is clear that, after computing the diffusion coefficients, the explicit scheme performs a linear combination of spectral vectors, which takes O(M) time. Given that the computations involved on Equation 3.10 are repeated over each one of the N pixels in the image, the time complexity of the explicit scheme is O(NM), i.e. linear in the size of the image.

The disk requirements of the explicit scheme are just ~2MN, since in addition to $\mathbf{U}^k$ and $\mathbf{U}_\sigma^k$, the largest temporal variables needed are the diffusion coefficients, which require

only O(N) disk space. Hence, the disk space requirements by the explicit scheme are O(N) + 2NM ~2NM, since $M >> 1$, as is the case of hyperspectral imagery.

On the other hand, the approximated AOS and ADI semi-implicit schemes are based on the Thomas algorithm, which is run only two times on each method (see Equations 3.16 to 3.19). Since the vector-valued Thomas algorithm runs in O(NM) time (see Appendix A), the time complexity of all these methods is O(NM), given that the remaining operations are sums of two dense matrices of size $N \times M$ (Equation 3.17), which requires O(NM) time, or the multiplication of a dense matrix of size $N \times M$ for a tri-diagonal matrix (Equations 3.17 and 3.18), which is also O(NM). The disk space requirements for the AOS and ADI methods are the same as the explicit method, i.e. ~2NM, since all the temporal variables required in the Thomas algorithm are O(N).

As can be seen from Equation 3.23, the operations in the PCG algorithm are dot products between vectors of size M, which requires O(M) time, dense matrices of size $N \times M$ multiplying vectors of size M, which requires O(NM) time, a sparse matrix (five diagonals of size O(N)) multiplying a dense matrix of size $N \times M$, which requires O(NM) time, since each element of the product is obtained in O(1) time. Finally, the computation of $\mathbf{Z}$ on Equation 3.23 is also O(NM), since it consists of solving first a lower triangular, tri-diagonal system of size NM by forward substitution and then an upper triangular, tri-diagonal system of size NM by backward substitution (see Equations 3.25 to 3.31). Hence, on each iteration, the PCG algorithm takes O(NM) time.

However, the number of iterations of the PCG method needed to reach a given error tolerance is a variable that depends on the scale-step μ and the condition number of the

matrix **A**. Hence, in general the time complexity of the PCG algorithm would be O(ηNM), where η is the number of steps necessary to reach a predefined error level. In our experiments, we found that the maximum number of iterations, which provides good accuracy and competitive running times (relative to ADI and AOS methods) is 30, hence η = O(1) and the PCG algorithm runs in O(NM) time, however, η changes according to the value of μ, since the condition number increases with μ (see Appendix B2).

Finally, let us consider the disk space requirements for the vector-valued PCG algorithm. There are five variables on Equation 3.23 (including $\mathbf{U}^k$) of the same size as the image, i.e. requiring each one NM disk space. The other temporal variables require only O(N) disk space and since $M >> 1$, they can be ignored here. However, notice that variable **Z** on Equation 3.23 can be replaced by **Q** without affecting the algorithm; hence, the PCG method requires ~4NM disk space, given that the space required by $\mathbf{U}_\sigma^k$ is liberated after computing the diffusion coefficients and it is not required by the PCG algorithm.

Table 3.1 summarizes the analysis of the asymptotic time and disk complexity of all the discretization schemes and PCG methods considered here. In Section 3.2, we analyze the speed up achieved with the semi-implicit schemes and PCG methods, which provide us a better comparison between these methods, since in the asymptotic analysis, the constants involved are hidden.

Table 3.1 Summary of Time and Disk Space complexity

Method	Complexity	
	Time	Disk Space
Explicit	$O(NM)$	~$2NM$
AOS	$O(NM)$	~$2NM$
ADI	$O(NN)$	~$2NM$
PCG	$O(\eta NM)$	~$4NM$

3.3 Experiments

In this section, we show that semi-implicit discretization schemes have better performance, in terms of accuracy of the computed solution and CPU time, than traditional explicit schemes to solve the nonlinear diffusion PDE on hyperspectral imagery. We also show that nonlinear diffusion can also be used to reduce the spatial and spectral variability in hyperspectral imagery, improving classification accuracy [*Duarte et al*, 2006, 2007]. The performance of the vector-valued anisotropic diffusion PDE is studied using four hyperspectral images. The first is a synthetic image that allows us to measure the reduction in the spatial and spectral variability and visualize artifacts or the destruction of edges easily. The synthetic and real hyperspectral images are used here for the evaluation of the performance of the implemented methods in terms of speedup and accuracy of the computed solution. The real hyperspectral images are also used here to evaluate the effect of nonlinear diffusion on image classification and its relationship with the accuracy of the computed solution to the PDE.

The explicit and semi-implicit methods indicated in the previous section were all implemented in Matlab 7.0. The classification of the hyperspectral images was performed

with Multispec[2], freeware software developed by D. A. Landgrebe from Purdue University. Also, all the real images were mapped to the [0 1] range, using

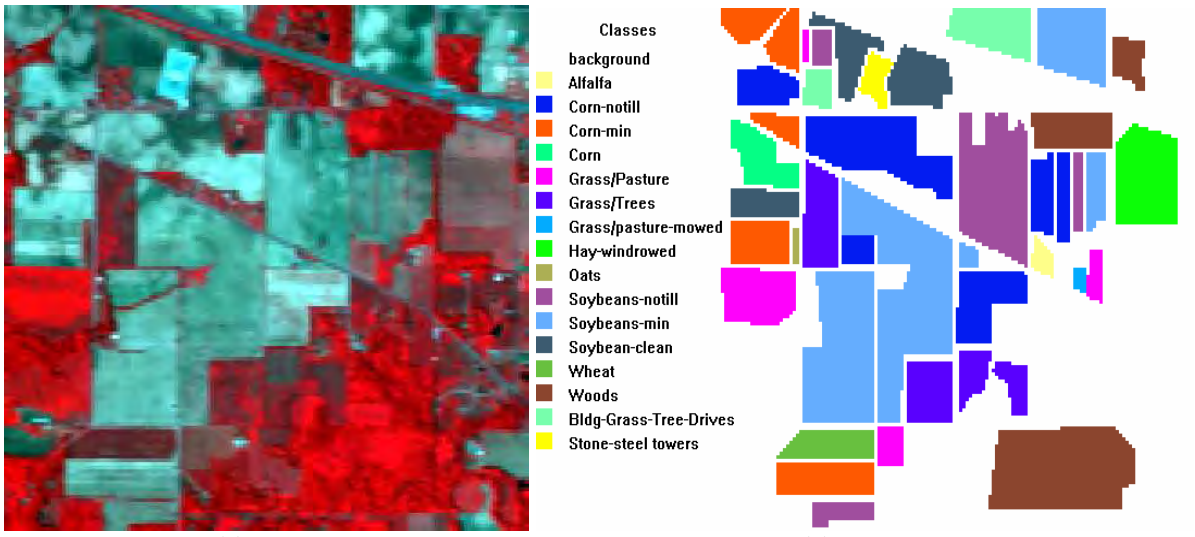
$$\mathbf{U} = [u]_{i,j}, \quad u_{i,j} \leftarrow \frac{\left(u_{i,j} - u_{\min}\right)}{u_{\max}},$$
$$u_{\min} = \min\left\{ u_{i,j} \right\}, u_{\max} = \max\left\{ u_{i,j} \right\}, \quad i = 0, \cdots N-1;\ j = 0, \cdots, M-1.$$

The final scale (K on Equation 3.14) is chosen here as a convenient integer value that facilitates the comparison between the different schemes. The gradient thresholds (α on Equation 3.6 used here for each hyperspectral image, where all found simply by scanning within the range [0.01 0.025] at intervals of 0.001 or larger, and choosing the value of α that produced the best classification accuracies, while the semantically meaningful edges in the image are preserved. This range corresponds to the larger range of variability that we had found for this parameter in all the hyperspectral images we had work with. On the other hand, we must select a standard deviation of the smoothing Gaussian kernel $G(0,\sigma^2)$ (Equation 2.8) such that $\sigma \leq 1/3$, in order to preserve the locality of the diffusion process and given that 99.7% of the Gaussian area is within $\pm 3\sigma$, that is, ± 1 pixel from the center, which corresponds to the stencil used for the discretization of the nonlinear diffusion equation, i.e. a 3×3 grid (Figure 3.1). As Weickert [1998] suggested, the regularization of the image can be done efficiently using isotropic diffusion, on each step with $\mu = \sigma^2/2$.

[2] http://dynamo.ecn.purdue.edu/~biehl/MultiSpec/

By trial and error, we found that $\sigma = 1/5$ is the value that best preserves the edges on all images used so far and at the same time eliminates impulsive noise. Also, for the PCG algorithm we set the maximum number of iterations to 30 given that the maximum number of iterations found in practice, were all lower than 20. We also fixed the error tolerance of the CG algorithm to 10^{-3} (Equation 3.23), which provide us with the best trade off between accuracy and speed such that the PCG methods could compete with the approximated semi-implicit schemes at large scale-steps. The four hyperspectral images used in our experiments are,

1. The NW Indian Pines image (Figure 3.2.a) taken with the AVIRIS[3] (Airborne Visible/Infrared Imaging Spectrometer) sensor, flown by NASA/Ames on June 12, 1992, over an area 6 miles west of West Lafayette, Indiana.



(a) **(b)**

Figure 3.2 a) Indian Pines (RGB shown corresponds to bands 47, 24, and 14), b) Ground truth[2].

This image contains 145x145 pixels and 220 spectral bands in the 400-2500 nm range, for which ground truth exists[2] (Figure 3.2.b). We disregard bands 1-3, 58, 77, 103-

110, 148-166, and 218-220, from the original image either because they correspond to absorption bands, they were too noisy or present strong illumination differences due to the sensor. Hence, the processed image has 145×145 pixels and 185 spectral bands in the 410-2430 nm range.

2. A synthetic hyperspectral test image (Figure 3.3) made from spectral signatures extracted from the Indian Pines image to fill simple geometric figures: triangle, ellipse, donut, and a common background. This image has 150×150 pixels and the same number of bands as the Indian Pines image. The pixels belonging to each geometric figure and background were selected at random and with uniform probability, from the pixels belonging to four different crops in the Indiana Pines image: the Corn-min field (triangle), the Soybeans-notill field (donut), the Soybeans-min field (ellipse), and the Hay-windrowed field (the background).

The spectral variability within each region of the synthetic image can be appreciated by superimposing the spectral signatures of each pixel within each region, as indicated on Figure 3.3.

[3] http://aviris.jpl.nasa.gov/

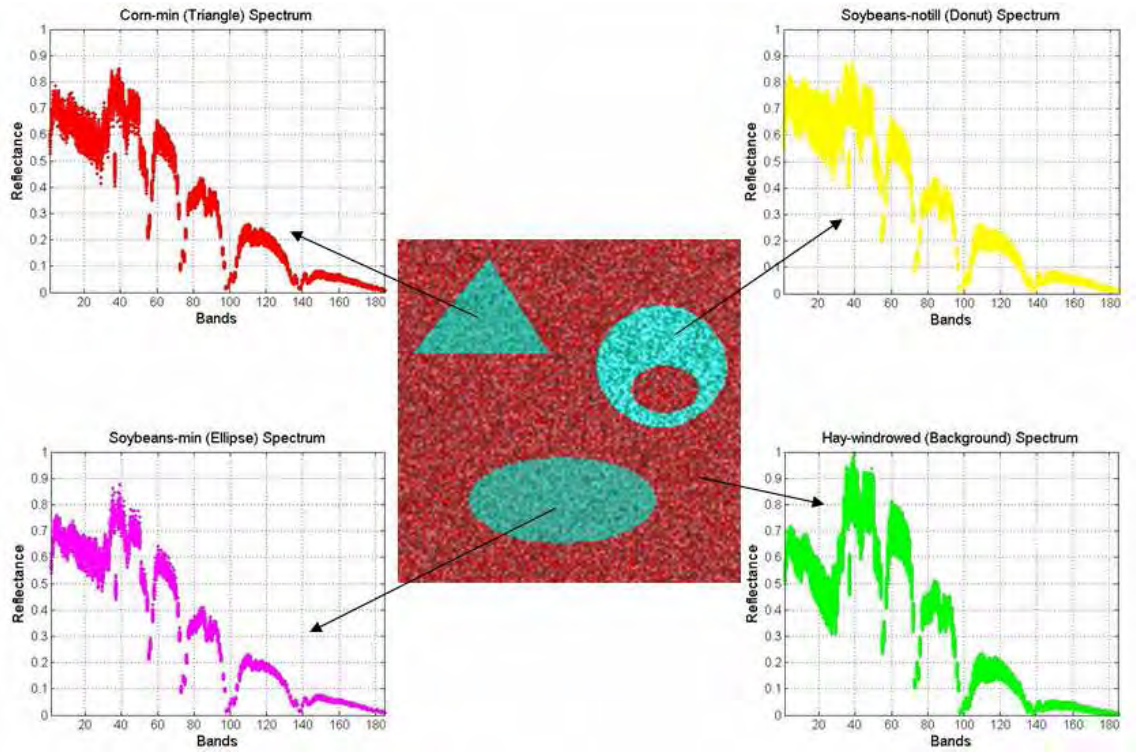


Figure 3.3 Synthetic Hyperspectral image, showing also the spectral variability within each region. (RGB shown corresponds to bands 47, 24, and 14)

3. The Cuprite image (Figure 3.4.a) taken over the cuprite's mining district, 2 km north of Cuprite, Nevada, with the AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) sensor, flown by NASA/Ames[4] 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. We selected a portion of the fourth scene, in the Cuprite image, of size 500×500 pixels that corresponds to a section of the mineral map from the Cuprite mining district, reported by the US Geological Survey (USGS) spectroscopy laboratory in 1995, using the expert system algorithm Tetracorder [*Clark et al*, 2003] and signatures of 60 sampled fields in the region. We use the USGS images as ground

[4] http://aviris.jpl.nasa.gov/html/aviris.freedata.html

truth[5] (Fig. 3.6). We selected from this image 50 bands: 172-221 that correspond to the 2000-2480 nm vibrational absorption region used by the USGS to map minerals in the Cuprite region.

(a) (b)

Figure 3.4 a) RGB composite of Cuprite image using bands 183, 193, and 207 and b) RGB composite of the noisy False Leaves image using bands 90, 68, and 29.

4. The False Leaves indoor image (Figure 3.4.b) of size 640x640 pixels and 120 bands in the 402-908 nm range, collected by the Surface Optics Company[6] using the SOC-700 hyperspectral imager. We selected a portion of this image of size 540×575 pixels that contains all the objects present in the original image. Additionally, and given that this spectrometer has a high spectral resolution, we selected only 30 bands of the original image, by taking one of each four consecutive bands. Since, this image has a high Signal to Noise Ratio (SNR) and none of the atmospheric effects that affect remote sensed images, such as those taken with the AVIRIS sensor; we

[5] http://speclab.cr.usgs.gov/PAPERS/tetracorder
[6] http://surfaceoptics.com

add white Gaussian noise with zero mean and $\sigma = 1$, of amplitude 10% relative to the maximum amplitude in the image.

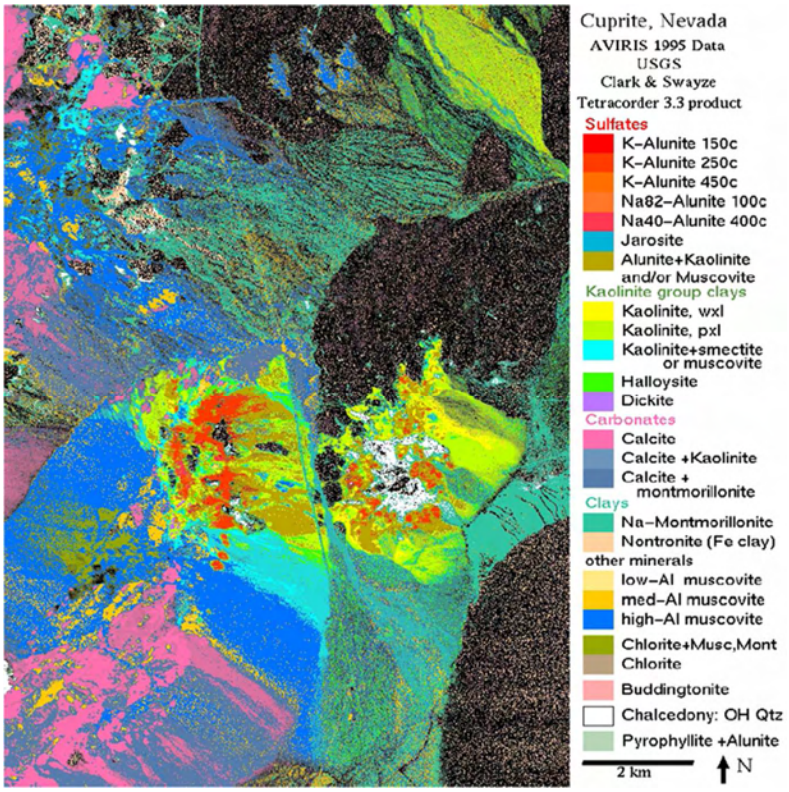


Figure 3.5 Ground truth Cuprite image

3.3.1 Performance in terms of the accuracy of the computed solution

The performance of all the algorithms implemented here is evaluated in terms of the accuracy of the computed solution and in terms of the speedup achieved with respect to the explicit scheme running at his highest stable scale step, i.e. $\mu = ¼$. The accuracy of the computed solution is evaluated as the square error relative to a reference image that has been smoothed with the semi-implicit Crank Nicholson scheme (Equation 3.18), at different scale steps. The Crank Nicholson scheme is second-order accurate, both in scale and space [Strikwerda, 2004] and we ensure high accuracy of the computed solution by using a very small scale-step ($\mu_r =$

0.025). Hence, we generate four reference images, one for each one of the hyperspectral images described in the previous section.

We found that $\alpha = 0.015$ is the highest threshold value that produces the best classification accuracies for the Cuprite and False leaves images, without damaging the semantically meaningful edges, while $\alpha = 0.012$ was the best value found the NW Indiana Pines image, and $\alpha = 0.022$ for the synthetic image. Notice, from Figure 3.3, that the synthetic hyperspectral image has the largest spatial variability, requiring thus a large α threshold. This is due the fact that the synthetic image was constructed by selecting at random pixels from different regions in the NW Indian Pines image. Hence, this image does not have the spatial correlation that appears on natural images and it constitutes by itself a good test for our algorithms, since we had to select a large α value to eliminate this variability but at the time the semantically meaningful edges were preserved (see Figure 3.6).

The number of scale-steps was chosen as the number of steps required by the explicit scheme to smooth the four hyperspectral images and produce the best classification accuracies. Once defined the number of steps of the explicit scheme, the number of steps of the semi-implicit schemes are chosen simply as integer multiples of the largest stable scale step of the explicit scheme, i.e. $\mu_0 = ¼$. We found necessary to run 100 scale-steps of the explicit scheme for the synthetic and False Leaves images, while 50 scale-steps were enough for the NW Indian Pines and Cuprite images. Hence, the final scale for the synthetic and False Leaves images is $100*\mu_0 = 25$ and the final scale for the NW Indian Pines and Cuprite images is $50*\mu_0 = 25$. We can reach these scales faster using the semi-implicit schemes at

larger scale-steps, hence, we select the scale steps for the semi-implicit schemes as $\mu/\mu_0 = 5$, 10, 20 and 50 for the synthetic and False Leaves images and $\mu/\mu_0 = 5$, 10, 25 and 50 for the NW Indian Pines and Cuprite images.

In order to generate the reference images used to evaluate the accuracy of the other methods, we need to run the Crank Nicholson scheme at a much lower scale step, $\mu_r = 0.025$, so that we need $25/\mu_r = 1000$ iterations of the Crank Nicholson scheme for the synthetic and False Leaves images and $12.5/\mu_r = 500$ iterations for the NW Indian Pines and Cuprite images.

The best values for ω in the PCG-SSOR scheme were found, simply by sweeping ω in the 0.01 to 2.0 range at steps of at least 0.01. The values of ω found by this mean were 0.5, 0.4, 0.3, 0.15, and 0.05 for $\mu = \mu_0$, $5\mu_0$, $10\mu_0$, $20\mu_0$ and $50\mu_0$, respectively, and they also correspond to the best values for the synthetic and real hyperspectral images used here. These results indicate that the SSOR preconditioner loses its effectiveness as μ increases, since the preconditioner tends to the identity matrix, when $\omega \to 0$ (Equation 3.26), which means that it cannot do better than the CG alone.

Finally, AOS and ADI-LOD schemes used as preconditioners did worse than the CG method [*Duarte et al*, 2007], which means that it worsened the condition number of matrix **A** and hence, it is not shown in our results here. We believe that the matrices used as approximations of the semi-implicit scheme (see Section 3.2.1) are good approximations to solve the semi-implicit scheme directly, but the error introduced by the approximation results in a strong bias in the CG search and hence, damps the search for an orthogonal basis in CG.

In fact, we could notice that using AOS or ADI as preconditioners, the error does not reduce and it may in fact increase, from iteration to iteration.

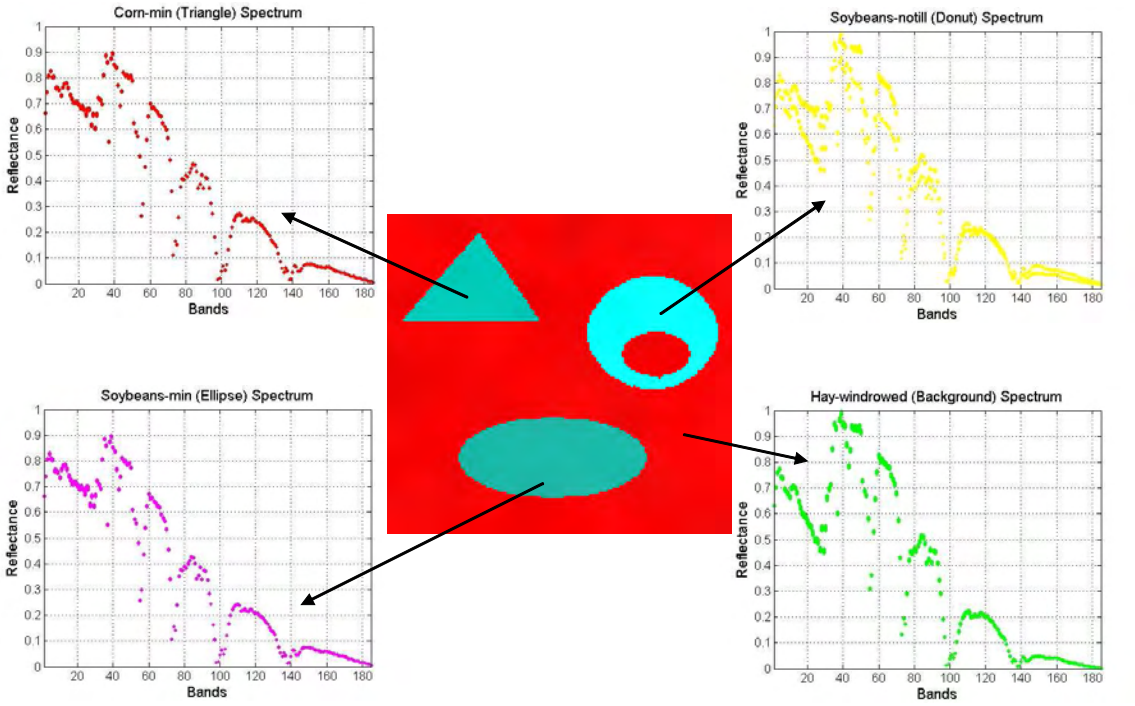
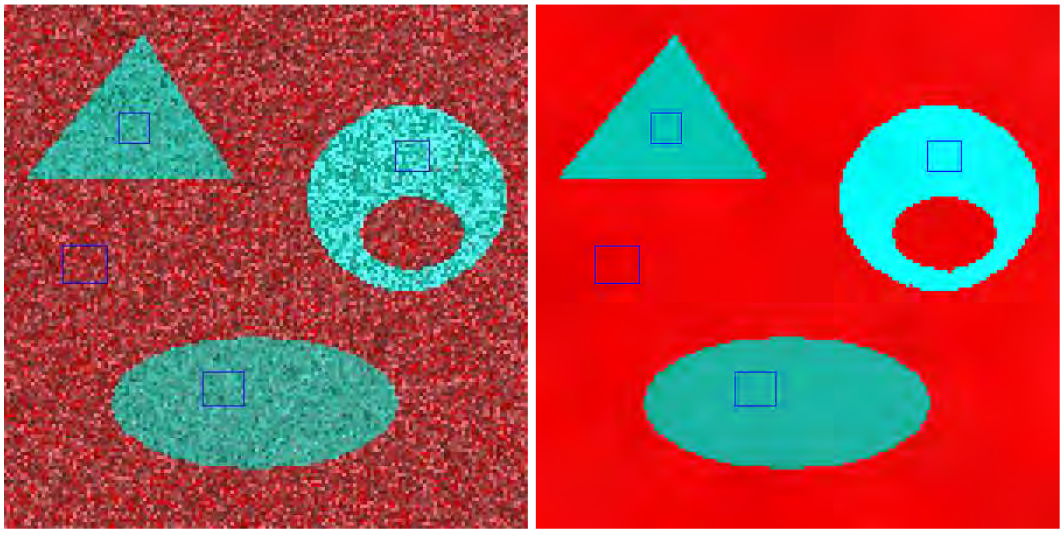


Figure 3.6 Smoothed Synthetic image, showing also the reduction in the spectral variability within each region in the image (RGB shown corresponds to bands 47, 24, and 14).

Figure 3.6 shows the smoothed synthetic hyperspectral image, where it is evident the strong reduction on the spatial and spectral variability within each image region, after nonlinear diffusion, while preserving the semantically meaningful edges. Table 3.2 indicates the reduction on the variance within each image region in the smoothed hyperspectral image.

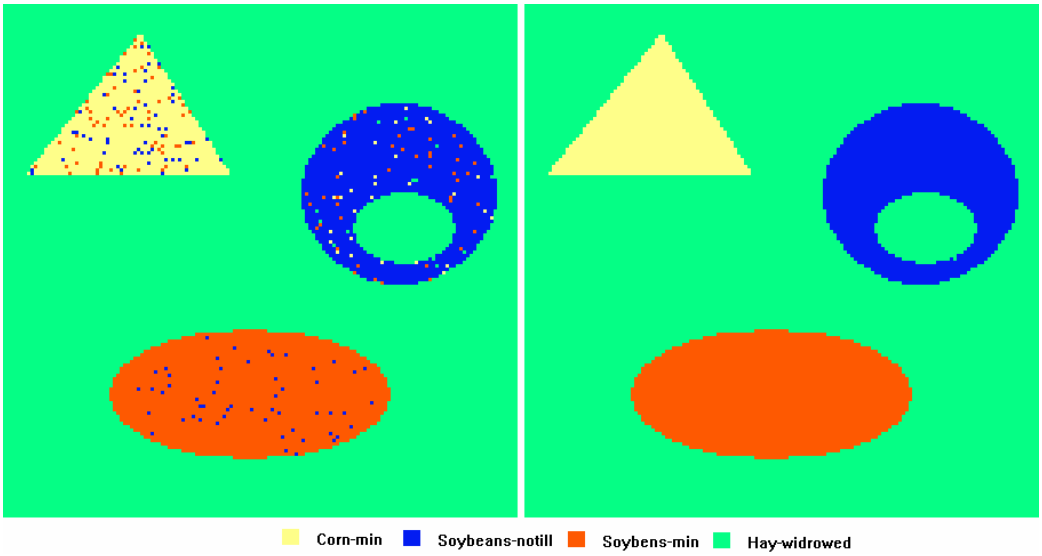
Table 3.2 Reduction the spectral/spatial variability in the smoothed synthetic hyperspectral image

Spectrum	Mean variance		Variance reduction (%)
	Original image	Smoothed image	
Hay-widrowed	2.42E-04	7.81E-07	99.68
Soybeans-min	8.92E-05	5.70E-07	99.36
Corn-min	8.92E-05	4.45E-07	99.50
Soybeans-notill	3.23E-04	3.85E-06	98.81



(a) (b)

Figure 3.7 Training samples (RGB shown uses bands 47, 24, and 14) on a) Original and b)Smoothed Synthetic Hyperspectral image.



(a) (b)

Figure 3.8 SAM Classification a) Original and b) Smoothed synthetic Hyperspectral image.

Figure 3.7 shows the training samples used on both, the original and smoothed synthetic images. The classification results are shown on Figure 3.8 using and all image bands and correlation SAM [*Landgrebe*, 2003]. It can be seen that the smoothed image was classified with 100% of accuracy, meanwhile the original noisy image presented misclassification errors on all the clases, except in the background.

Figures 3.9 to 3.12 show the speedup relative to the explicit scheme for each semi-implicit method used here and for each hyperspectral image, as a function of the relative scale step μ/μ_0. The speedup is computed simply as,

$$S = \frac{\text{CPU time explicit scheme}}{\text{CPU time semi - implicit scheme}}.$$

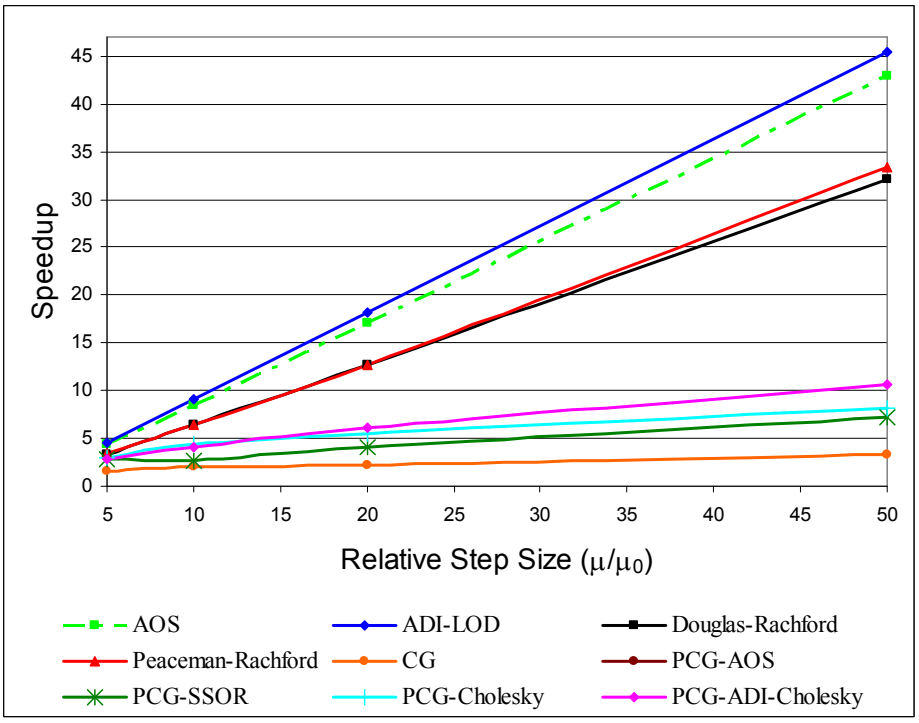


Figure 3.9 Speed-up of the different semi-implicit schemes and PCG methods for the synthetic hyperspectral image.

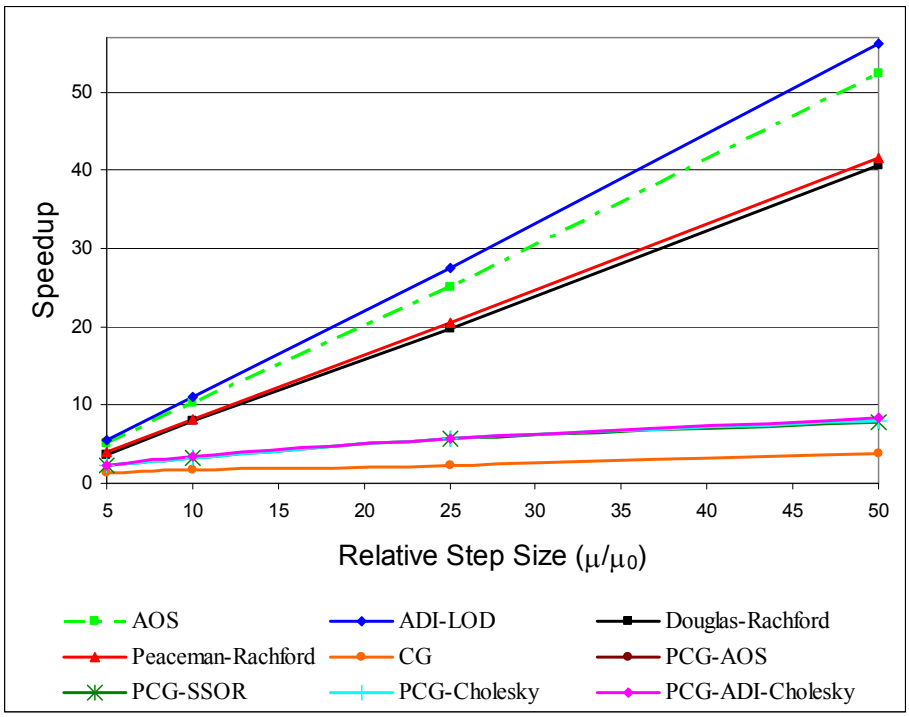


Figure 3.10 S Speed-up of the different semi-implicit schemes and PCG methods for the NW Indian Pines image.

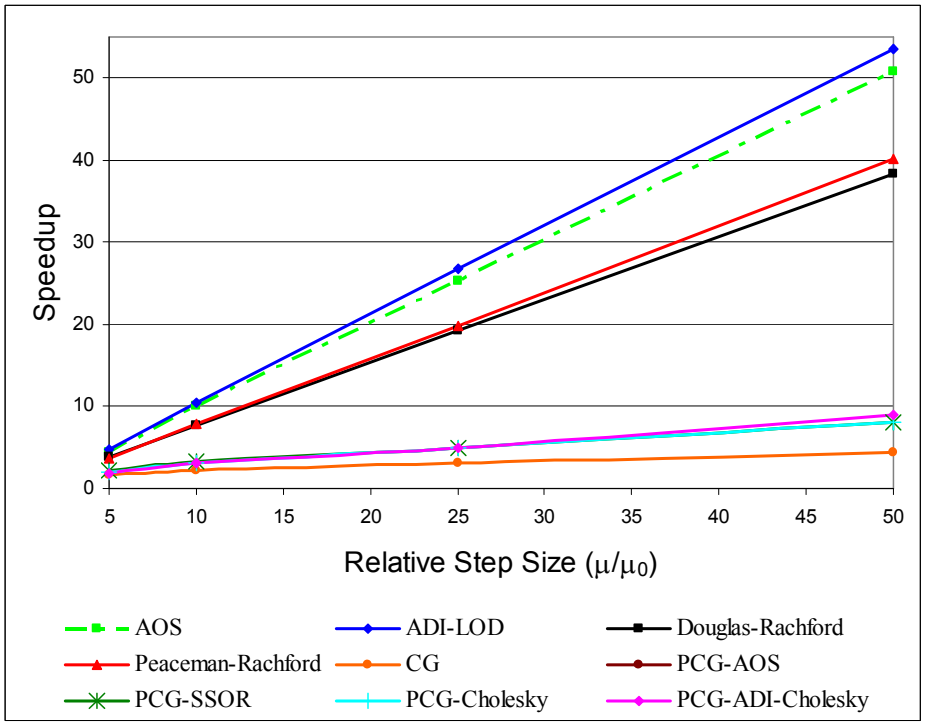


Figure 3.11 Speed-up of the different semi-implicit schemes and PCG methods for the Cuprite image.

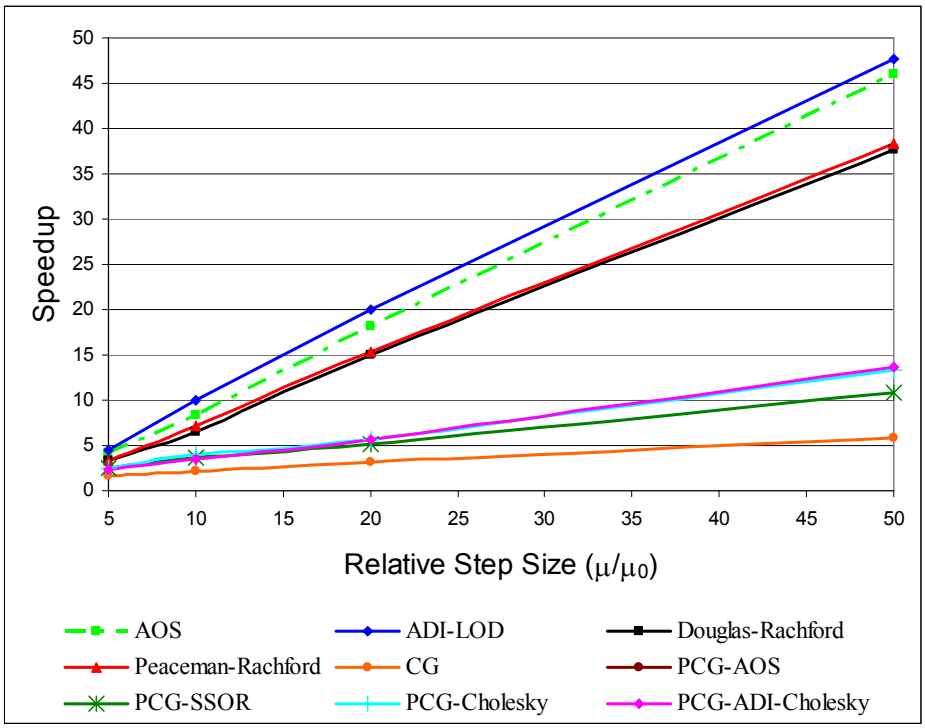


Figure 3.12 Speed-up of the different semi-implicit schemes and PCG methods for the False Leaves image.

From previous figures, we see the same pattern for all images, where the highest speedups are achieved by ADI and AOS, followed closely by Douglas-Rachford and Peaceman-Rachford methods. From these figures, it is clear that the PCG-methods used, only improve the running time of the CG by a factor of two to four times, which is not enough to make them competitive. Finding a good preconditioner is still an art (*Saad*, 2003) and these results only show that the classical preconditioners used here were all inadequate. Further work must be done in this area to find good preconditioners, since as Figures 3.13 to 3.16 show, the PCG methods preserve better the accuracy of the computed solution at large scale-steps.

Figures 3.13 to 3.16 show the percentage of the square error for each one of the numerical methods implemented here and for each hyperspectral image. The square error is computed here as,

$$error = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\left(\hat{u}_{ij} - u_{ij}\right)^2}{\sum_{i=1}^{N}\sum_{j=1}^{M}\hat{u}_{ij}^2}, \qquad \textbf{3.32}$$

where, $\hat{u}$ corresponds to the reference hyperspectral image (smoothed with the Crank Nicholson scheme) and u to the image computed using the explicit, semi-implicit or PCG methods indicated on Figures 3.9 to 3.12.

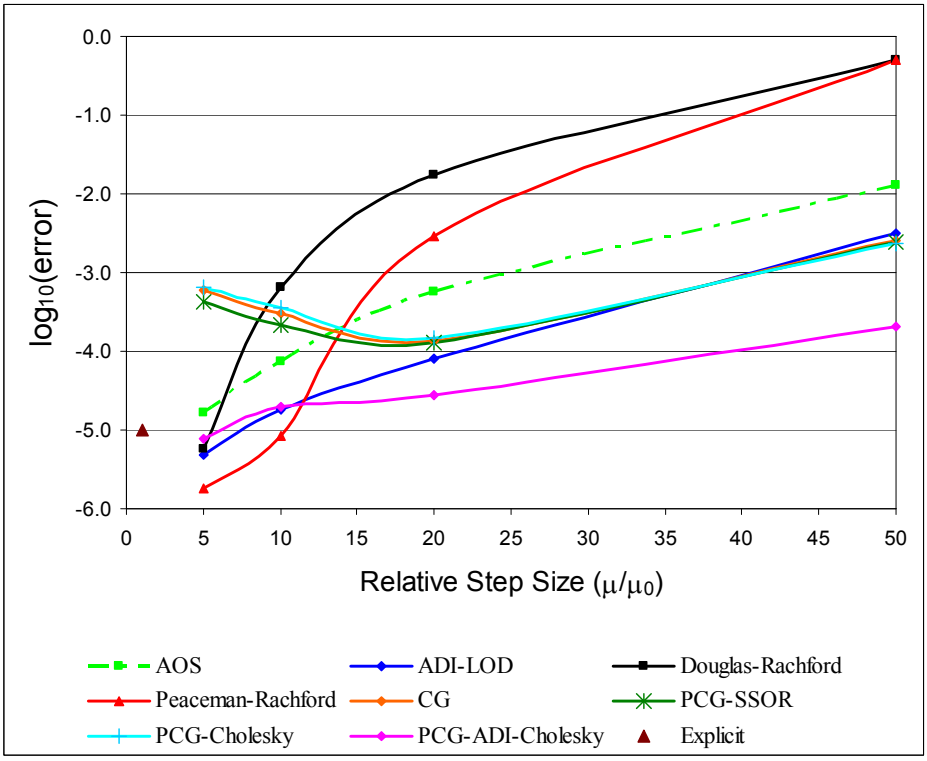


Figure 3.13 Square error of the computed solution for the synthetic hyperspectral image.

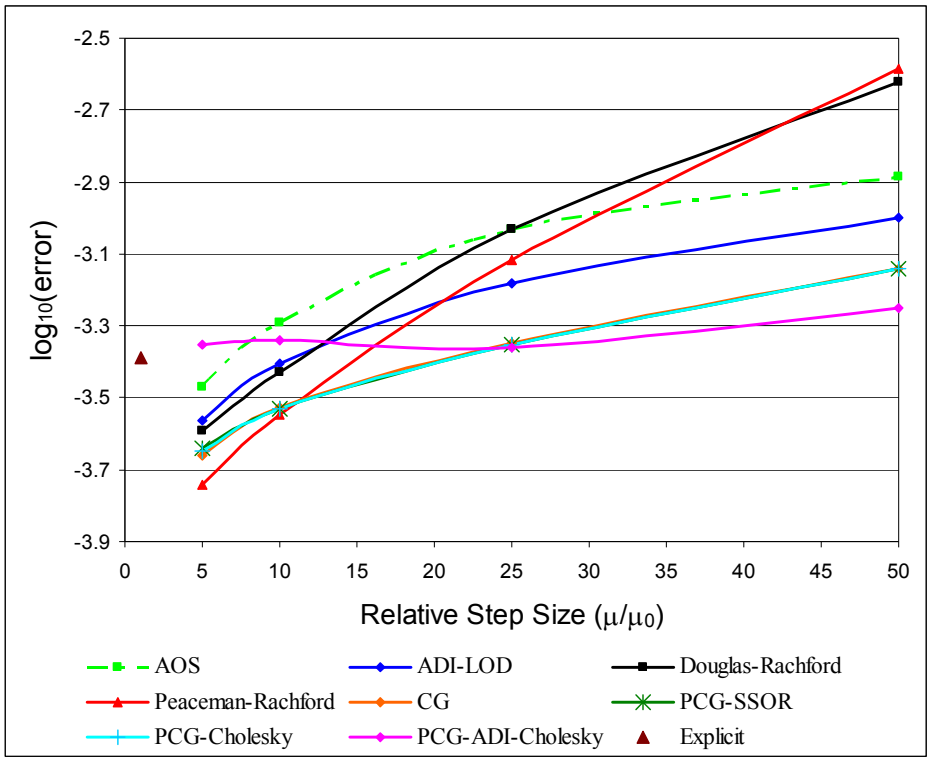


Figure 3.14 Square error of the computed solution for the NW Indian Pines image.

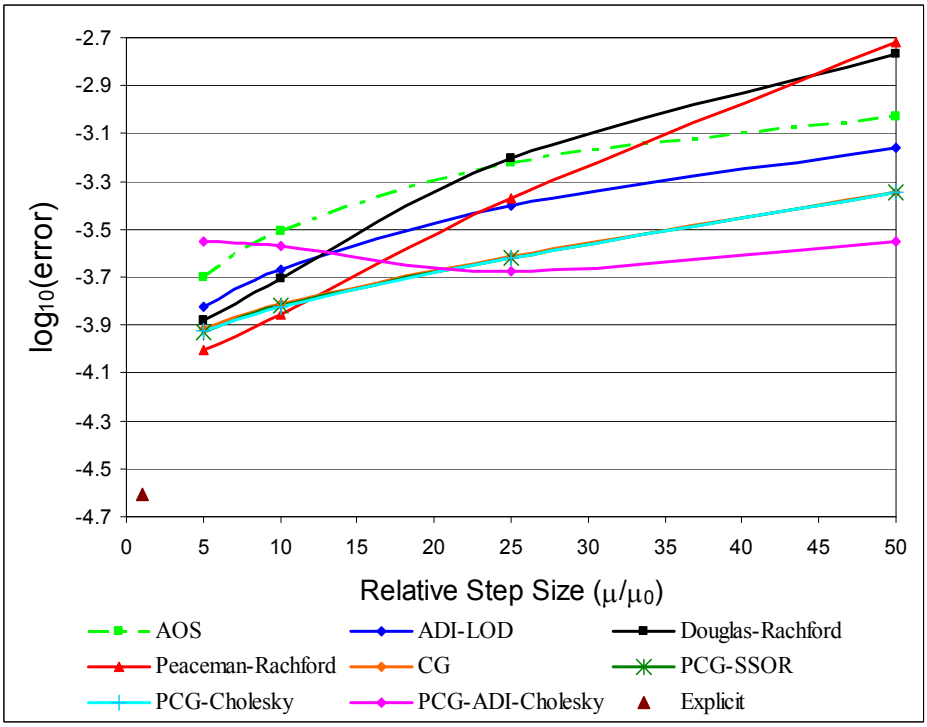


Figure 3.15 Square error of the computed solution for the Cuprite image.

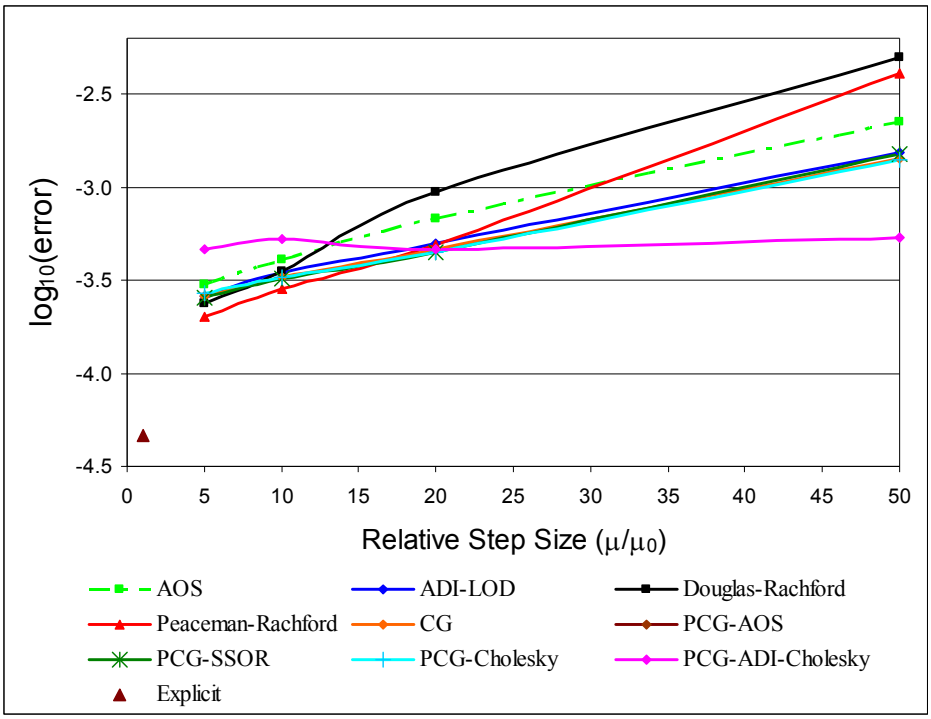
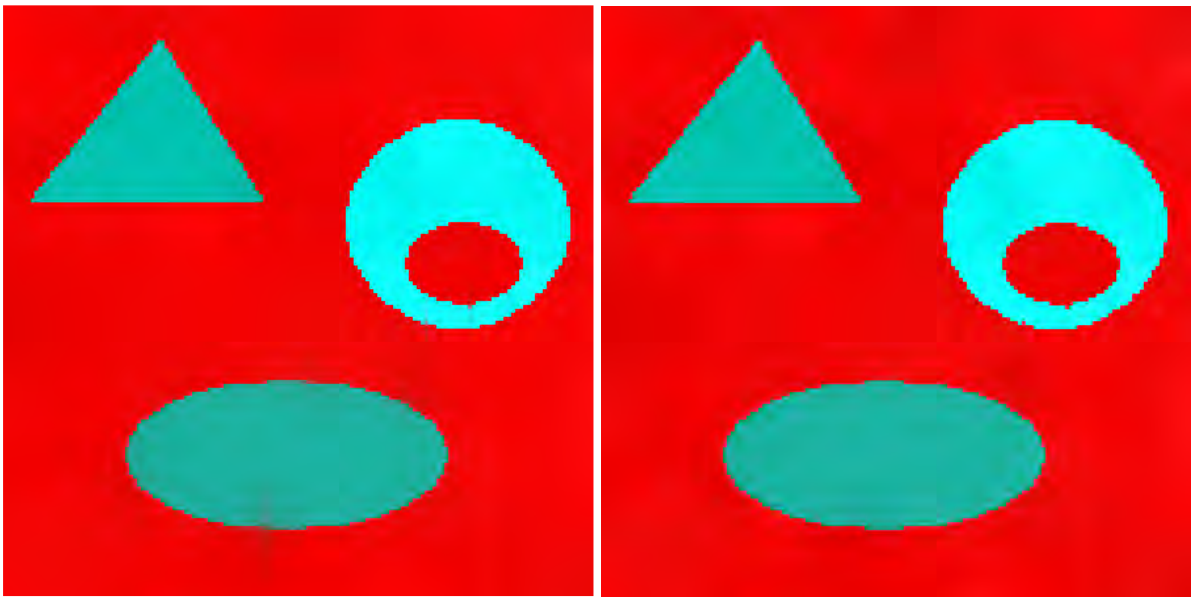


Figure 3.16 Square error of the computed solution for the False Leaves image.

From Figures 3.13 to 3.16 it can be noticed that the error has the same general behavior for all the images considered here. In all theses figures, the PCG methods maintain their accuracy even at large iteration steps. In particular, the CG method initialized with ADI-LOD and preconditioned with incomplete Cholesky factorization (PCG-ADI-CHolesky) is very insensitive to large scale-steps thanks to its initialization with ADI-LOD, which as can be seen from previous figures, has the lowest error of all the approximated semi-implicit methods at scale-steps equal or larger than $20\mu_0$. At moderate iteration steps, the Peaceman-Rachford semi-implicit method has the highest accuracy, while AOS has the largest error. In the experiments, we found that AOS, Douglas-Rachford, and Peaceman-Rachford produce artifacts on the synthetic hyperspectral image for $\mu \geq 20\mu_0$. This result coincides with the

results of [*Weickert et al*, 1998] on the practical range of scale-steps for AOS given by $\mu <$ $20\mu_0$.



(a) **(b)**

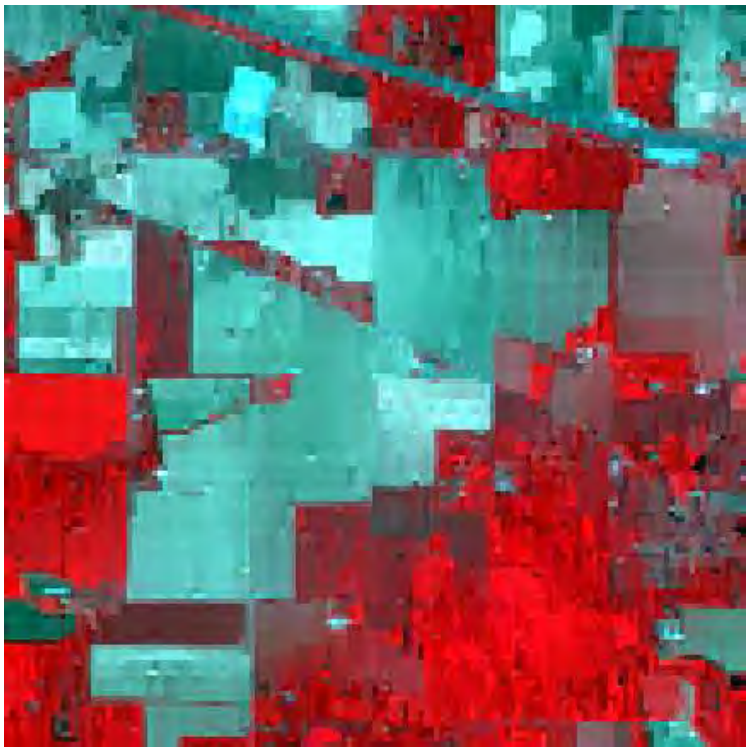
Figure 3.17 Smoothed synthetic image using $\mu = 20\mu_0$ and a) AOS, b)ADI-LOD.

Figure 3.17.a shows the artifacts (seen as gray crosses on the boundaries of the ellipse and donut) produced by AOS on the synthetic image using $\mu = 20\mu_0$, while ADI-LOD (Figure 3.17.b) does not produces any visible artifact. On the other hand, ADI-LOD and the PCG methods did not introduce visible artifacts on any of the hyperspectral images considered here.

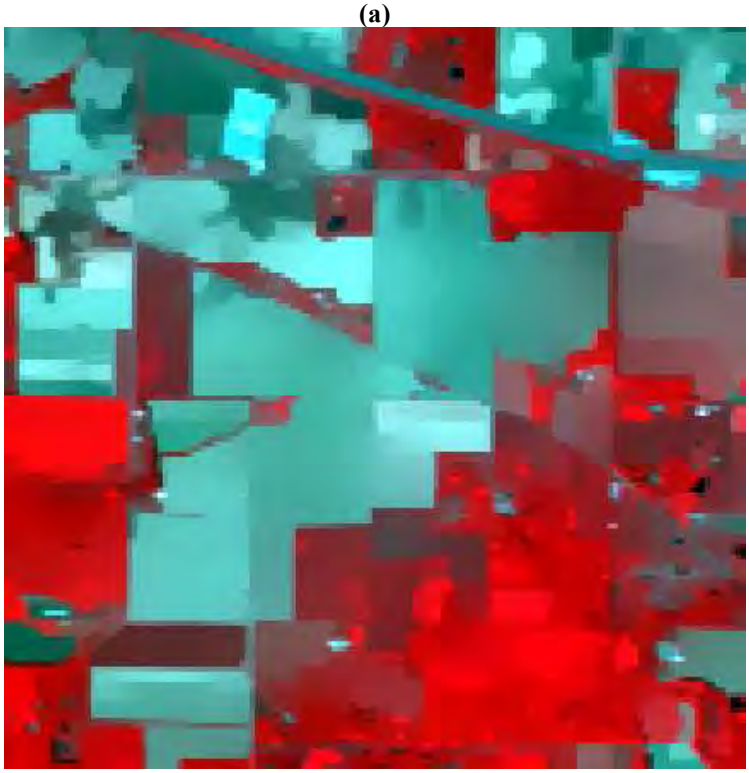
The visual detection of artifacts on real hyperspectral images is not as easy as it was with the synthetic image, because isolated artifacts of size no larger than a few pixels might be easily overlooked among the many different structures in the image. However, in order to illustrate this effect on the real hyperspectral images, we smoothed the NW Indian Pines image using the Peaceman-Rachford method which has the largest square error at $\mu = 50\mu_0$ (see Figure 3.18.a) and compare it with the smoothed image obtained using the PCG-ADI-

Cholesky (Figure 3.18.b) which has the lowest square error at this scale-step. Notice the strange artificial pattern over the entire image that appears on Figure 3.18.a that was not present in the original image (Figure 3.2.a).

As a rule of thumb, the Peaceman-Rachford method have the best accuracies of all the semi-implicit methods considered here at scale-steps $\mu \leq 10\mu_0$, followed closely by the Douglas-Rachford method. Given that they also have good speedups, these two methods should be preferred for scale-steps $\mu \leq 10\mu_0$ if both high accuracy and speedup are of great importance. Given that only AOS and ADI-LOD keep good accuracy at scales $\mu < 20\mu_0$, these two methods should be preferred here, since they also have the largest speedups. After scale steps $\mu > 20\mu_0$, only ADI-LOD and the PCG methods do not introduce artifacts and hence, if both accuracy and speed are required, ADI-LOD predominates over the other methods, given that it is also the fastest.



(a)



(b)

Figure 3.18 Smoothed NW Indian Pines image using $\mu = 50\mu_0$ a) with Peaceman-Rachford (notice the strong artifacts introduced), b) with PCG-Cholesky initialized with ADI-LOD.

Since, usually, we are interested in the largest scale-steps, ADI-LOD would be always preferred since it is the fastest of all the semi-implicit method and, at the same time, have good accuracy. Nevertheless, given the higher parallelism of AOS (see Equation 3.19), it might become faster than ADI-LOD on parallel implementations.

Since we are using a formal scale-space framework here, accuracy is of prime importance, nevertheless in practice, an image that presents artifacts after smoothing might have higher classification accuracies than another smoothed with no artifacts, because reducing intra-object variability can make the statistical model inadequate and hence, smoothing can in fact decrease classification accuracy for those classifiers (see next section). Hence, the degree of artifacts and accepted levels of error with respect to the exact solution of the anisotropic diffusion equation depends heavily on the application and models used. Here, we are interested in generating a valid scale-space for hyperspectral imagery and multiscale segmentation, and hence, good accuracy of the computed solution is always preferred to high speedups, since artifacts can produce segments that do not correspond to real structures in the image.

3.3.2 Comparison of Conjugated Gradient Vectorial vs. band by band

Figures 3.19.a to 3.22.a show the speedup achieved using the vector-valued CG algorithm (Equation 3.23) relative to running the CG on each band, independently.

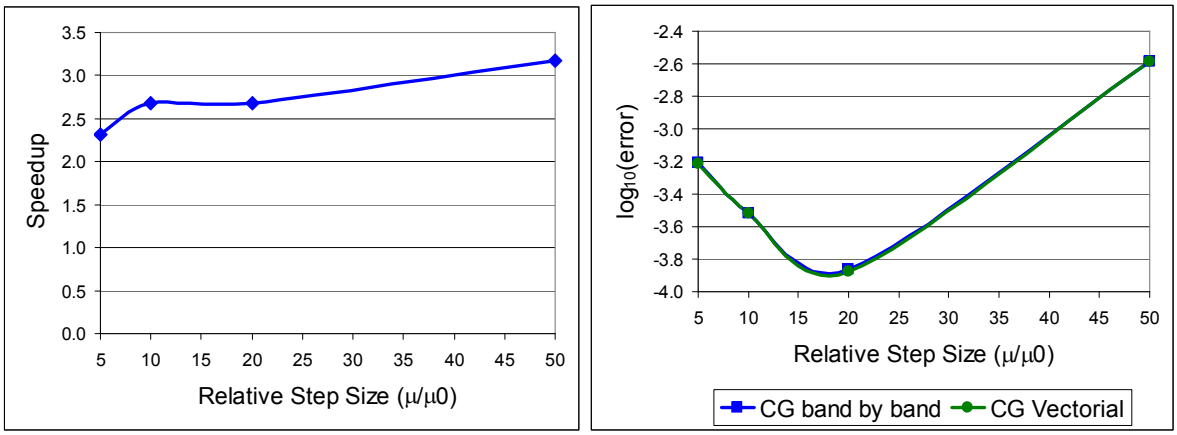


Figure 3.19 Comparison between CG-vectorial vs. CG band by band, in terms of a) Speedup, b) square error, for the synthetic hyperspectral image.

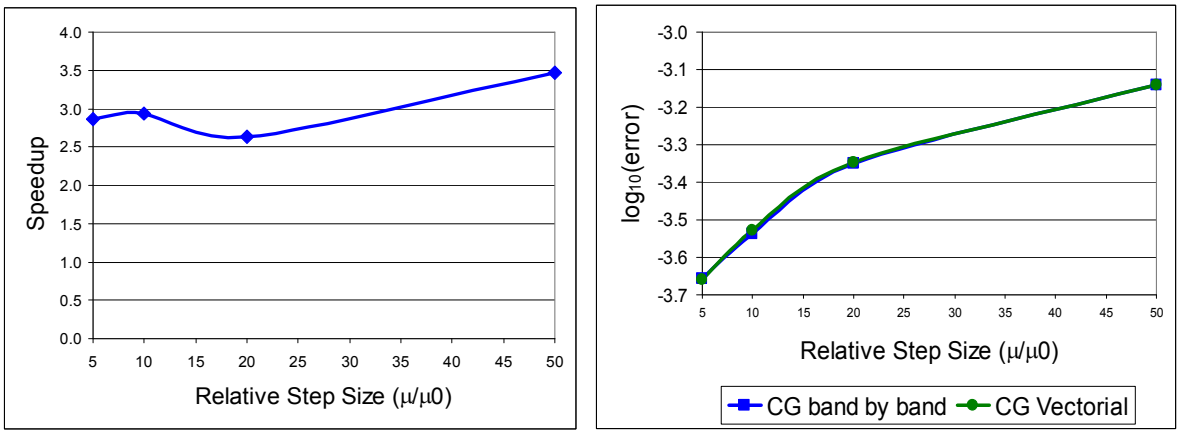


Figure 3.20 Comparison between CG-vectorial vs. CG band by band, in terms of a) Speedup, b) square error, for the NW Indian Pines image.

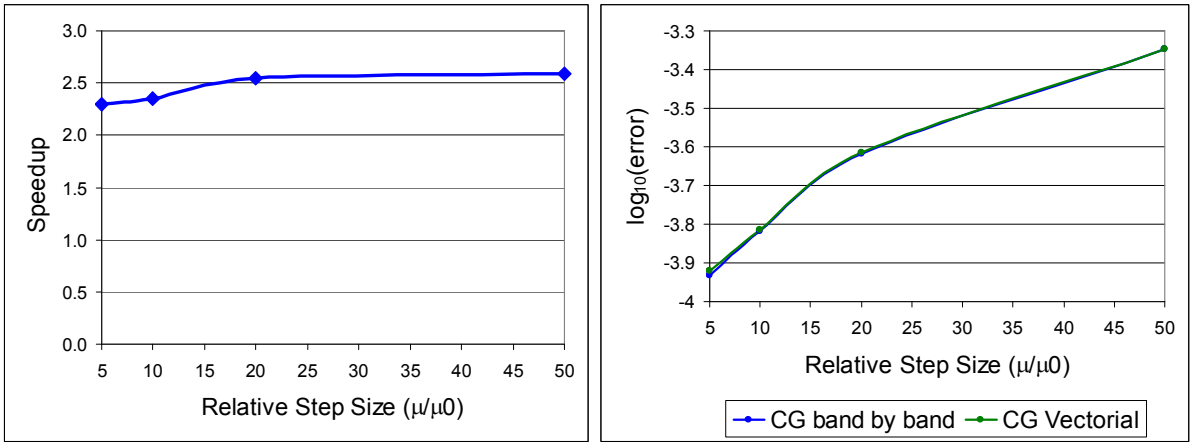


Figure 3.21 Comparison between CG-vectorial vs. CG band by band, in terms of a) Speedup, b) square error, for the Cuprite image.

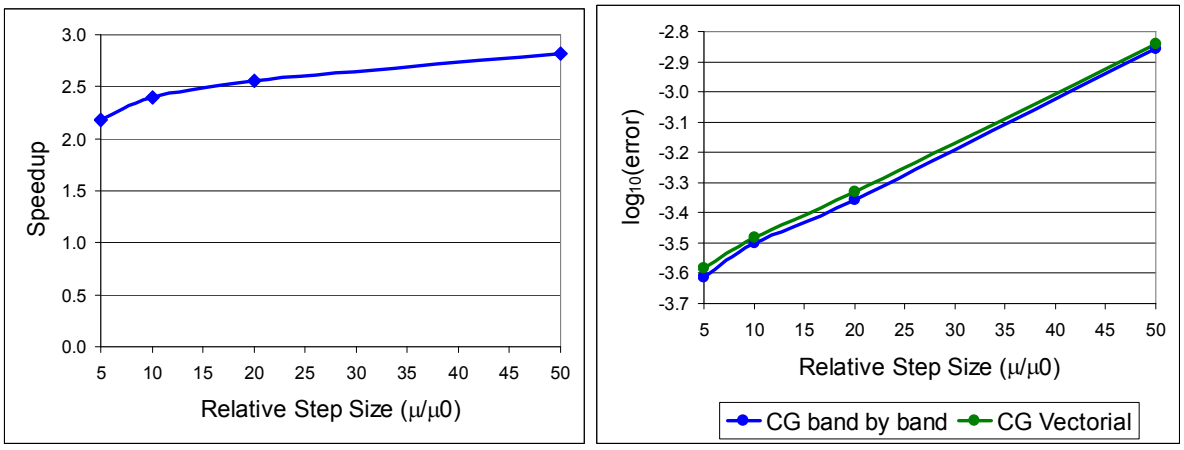


Figure 3.22 Comparison between CG-vectorial vs. CG band by band, in terms of a) Speedup, b) square error, for the False Leaves image.

As can be appreciated from these figures, the speed up ranges from 2.0 to 3.5 times the running time of the band by band CG algorithm, which is a necessary improvement to make it competitive to the vector-valued explicit and semi-implicit methods that are already exploiting the advantage of vectorization in Matlab.

Even tough the last versions of Matlab had improved the performance of for loops trough the use of the Just In Time (JIT) accelerator[7]. Figures 3.19.a to 3.22.a show that vectorization improved the performance of the code. Nevertheless, this vectorial approach is by no means limited to Matlab, and in fact, current high performance implementations exploit the fact that the same processing needs to be performed over hundreds of bands, as is the case of the semi-implicit schemes used here. Of course, full parallelism of all the algorithms used to process hyperspectral imagery would require partitioning the image along the spatial direction [*Plaza et al*, 2006, 2007], given the large spatial size of these images (see Section 1.1) and hence, additional considerations should be made besides vectorization that depend heavily on the architecture. High performance implementations have several

architectural alternatives in hyperspectral imagery such as distributed computing, clusters and the use of hardware implementations using FPGAs [*Plaza et al*, 2006, 2007], however each approach have their advantages and disadvantages and possibly a better alternative is to use a software/hardware codesign that exploits the advantages of hardware implementations with the flexibility and low cost implementation of complex functions [*Filho et al*, 2003]. The study of high performance implementations of all algorithms presented in this and the next chapter is out of the scope of this work, and is a full research project on its own.

3.3.3 Performance in terms of classification accuracy and speedup

In order to test the classification accuracy on the original and smoothed hyperspectral images, we need training and testing samples. The NW Indian Pines and Cuprite images have published ground truth (see Figure 3.2.b and Figure 3.5). Images with ground truth are very scarce in remote sensing, given the costs involved on its acquisition. The False Leaves is an indoor image with objects that can be identified visually. This image owes its name to the fact that there are some plastic leaves that cannot be distinguished from the real ones, in the visible range. Hence, we must use a suitable combination of bands that include the near infrared wavelengths to identify the false leaves and select the corresponding training and testing samples.

From Figure 3.2.b, we can see that the ground truth of the Indian Pines image consists of 16 classes, of which 10 correspond to different kinds of crops, 5 correspond to vegetation and one corresponds to a building.

[7] http://www.mathworks.com/company/newsletters/digest/sept02/accel_matlab.pdf

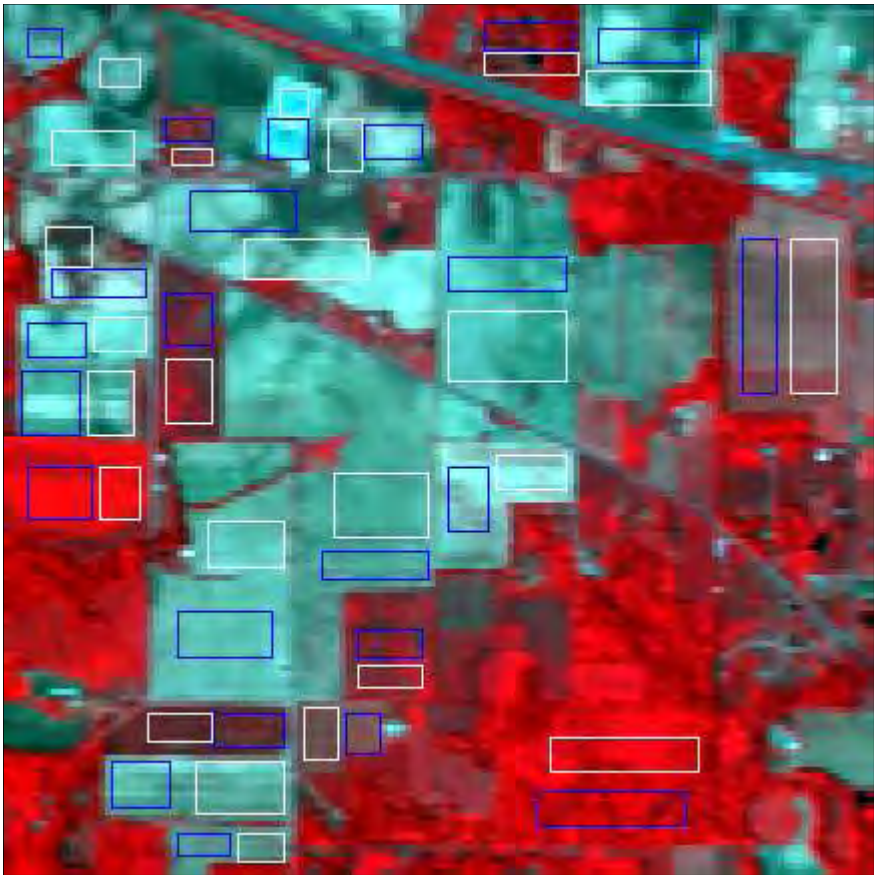


Figure 3.23 Training (blue rectangles) and testing samples (white rectangles) on the NW Indian Pines image. (RGB shown corresponds to bands 47, 24, and 14).

Figure 3.23 shows the training (blue rectangles) and testing samples (white rectangles) selected for 14 of the 16 classes identified on the NW Indian Pines image. The other two classes (Oats and Alfalfa) were not sampled given that there are not enough training and testing samples to perform the classification using classical statistical classification methods.

From Figure 3.5, we can see that the ground truth for the Cuprite image consists of 25 classes of minerals, grouped in five categories: sulfates, carbonates, Kaolinites, Clays, and other minerals.

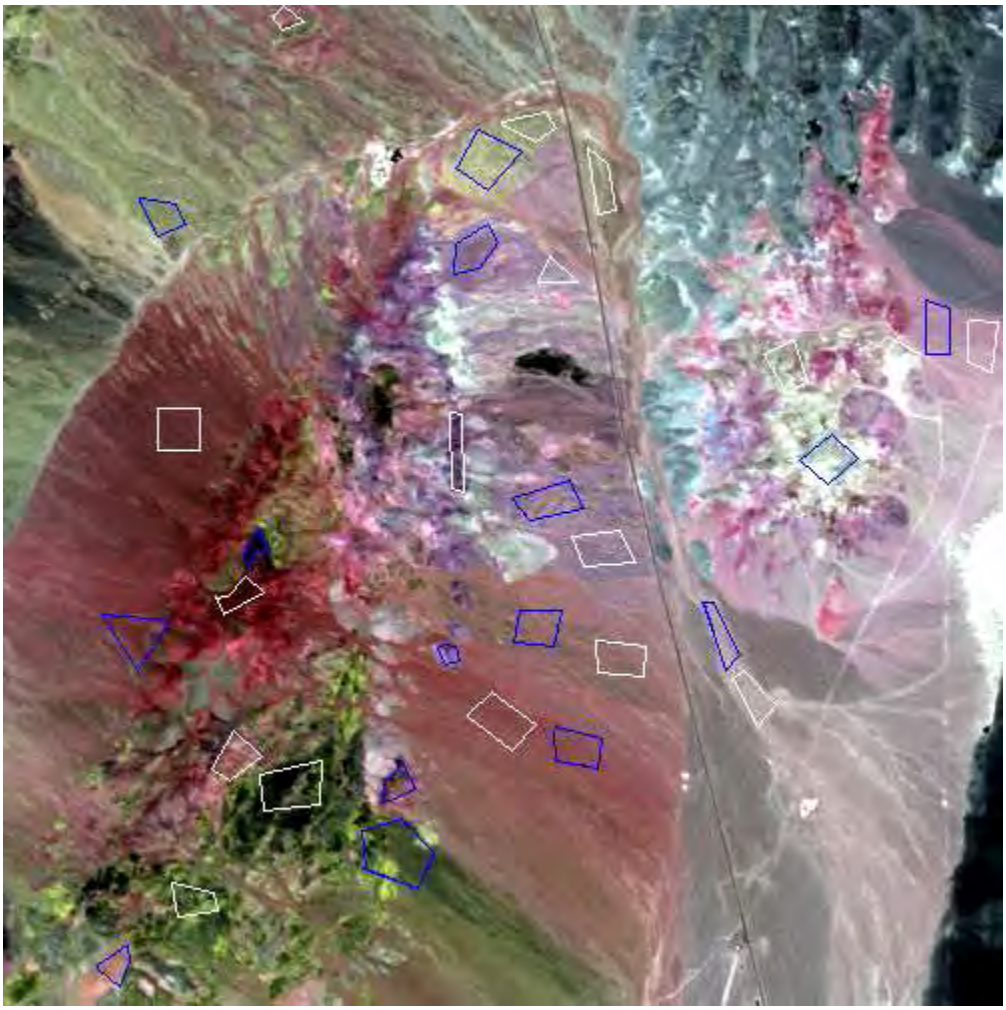


Figure 3.24 Training (blue polygons) and testing samples (white polygons) on the Cuprite image.

Figure 3.24 shows the training (blue polygons) and testing samples (white polygons) selected on 11 classes of the Cuprite image. They are Calcite, Kaolinite and Semectite or Muscovite, K-Alumnite, Kaolinite, Alunite and Kaolinite or Muscovite, Calcite and Kaolinite, Chalcedony, Na-Montmorillonite, Chlorite and Muscovite or Montmorillonite, High-Al Muscovite and Med-Al Muscovite. We consider the different kinds of Alunites as a single class, given that it is extremely difficult to obtain pure training and testing samples for them in this image. The remaining classes were not sampled given that there are not enough training and testing samples or because they were too difficult of localize within the Cuprite

image, even with the help of the wavelengths recommended by the USGS to identify some of the minerals in this image (see Figure 3.4 and Figure 3.5).

Figure 3.25 shows the training (blue polygons) and testing (white polygons) samples on each one of the different objects that can be identified in the Fake Leaves image, where we had used bands 90, 68, 29 from the original image to form the RGB shown, which allow us to distinguish the fake leaves (dark) from the true leaves (red). The classes in this image are the wall, the jar, the flowerpot, the true leaves, the false leaves, the metallic case, plastic label, paper label, and lens cover (dark red) of the featured SOC-700 hyperspectral imager that appears in the image.

We use here all the classifiers available in Multispec [*Landgrebe*, 2003]: Maximum Likelihood (ML), Fisher Linear Discriminant (FLD), Euclidean Distance (ED), Extraction and Classification of Homogeneous Objects (ECHO), Spectral Angle Mapping (SAM), and Matched Filter (MF) to evaluate how each classifier is affected by the nonlinear diffusion process. In fact, ECHO uses the Fisher Linear Discriminant in all the smoothed images, since it produces the highest classification accuracies for all the images.

Figure 3.25 Training (blue polygons) and Testing samples (white polygons) on the False Leaves image.

Since the original and smoothed NW Indian Pines images have 185 spectral bands each and the statistical classifiers employed here require more training pixels than spectral bands in the image [*Landgrebe*, 2003], we selected 20 bands using the SVD band subset selection algorithm [*Vélez-Reyes and Jimenez*, 1998; *Vélez-Reyes et al*, 2000; *Vélez-Reyes and Linares*, 2002] implemented at the UPRM Hyperspectral Image Analysis MATLAB Toolbox [*Arzuaga et al*, 2004] on each one of the smoothed images.

The best classification results for the NW Indian Pines image were obtained using α

= 0.012 and 50 runs of the explicit scheme at $\mu_0 = ¼$. Hence, the semi-implicit methods were also run for $\mu = 5\mu_0$, $10\mu_0$, $25\mu_0$ and $50\mu_0$ that corresponds to $(50\mu_0)/\mu = 10$, 5, 2, and 1 scale-steps, respectively. For the Cuprite image, we found experimentally that a value of α = 0.015 and 50 scale-steps produced good classification results, for the explicit scheme and hence, we use the same values of μ, as in the NW Indian Pines image for the semi-implicit methods. On the other hand, we obtained good classification results in the noisy False Leaves image using $\alpha = 0.015$ and 100 runs of the explicit scheme, hence, the semi-implicit methods were run for this image 20, 10, 5, and 2 scale-steps, respectively.

The classification results are shown in detail from Table 3.3 to Table 3.5 for each image, classifier and numerical method implemented. On these tables, **S** stands for the speedup achieved by the numerical methods implemented, and **PA** and **UA** stand for the producer's and user's accuracies [*Landgrebe*, 2003] respectively. The producer's accuracy indicates the percentage of pixels belonging to a given class that were correctly classified, while the consumer's accuracy indicates the percentage of pixels correctly classified for a given class, relative to all the pixels from the image that were classified (correctly or incorrectly) as belonging to that class[8]. Hence, as consumer's accuracy decreases, the confusion between classes (commission error) increases, even tough a given class might have high producer's accuracy.

The highest classification accuracies and speedups are highlighted on each table, for each method, in bold and cursive. The highest speedups were chosen as the maximum

[8] http://geospatial.amnh.org/remote_sensing/guides/image_interp/accuracy_assessment.html

speedup that keeps the classification accuracy very close or above the classification accuracy achieved with the explicit scheme. Of course, the best performance is for those methods that achieve classification accuracies above the explicit method and high speedups.

Table 3.3 Classification Accuracies NW Indian Pines image.

Numerical method				Classification Accuracy (%)											
				ML		FLL		ED		ECHO		SAM		MF	
Scheme	μ/μ_0	t(s)	S	PA	UA	PA	UA	PA	UA	PA	UA	PA	UA	PA	UA
Original	-	0.0	-	75.3	76.5	63.7	64.9	35.0	41.0	***80.6***	***82.2***	41.0	46.1	47.1	49.8
Explicit	1	118.0	1	89.2	91.4	89.9	90.1	50.6	49.9	***90.1***	***90.3***	50.6	52.2	74.0	75.0
AOS	5	23.2	5	87.5	91.2	90.3	91.1	49.7	52.9	90.9	92.2	49.2	52.5	73.7	76.5
	10	11.6	10	81.2	92.1	91.9	92.5	47.8	51.6	92.7	93.7	48.9	53.2	75.5	79.0
	25	4.7	25	81.7	90.9	90.9	90.6	43.7	49.6	89.7	89.6	47.6	52.8	73.0	74.6
	50	2.3	***52***	80.2	90.0	85.7	85.9	40.5	44.8	***93.7***	***94.1***	44.9	49.8	62.3	65.9
ADI	5	21.2	6	83.9	88.0	92.9	94.1	49.9	49.6	***93.1***	***94.5***	49.5	52.1	74.2	75.2
	10	10.6	11	88.1	92.3	91.1	92.1	49.3	51.4	91.6	93.0	49.6	53.5	74.5	77.8
	25	4.3	***28***	75.6	85.7	91.1	91.5	46.7	51.2	90.9	91.5	48.9	53.0	76.6	79.9
	50	2.1	56	74.1	88.1	91.0	91.0	44.0	50.1	89.9	90.1	47.6	53.1	73.8	76.0
Douglas Rachford	5	32.1	4	92.7	93.6	92.7	93.5	50.7	49.5	93.1	94.2	50.6	54.3	74.2	75.2
	10	15.0	***8***	***94.8***	***95.4***	91.8	92.0	51.0	53.6	91.5	91.7	50.4	54.7	71.8	72.7
	25	6.0	20	88.5	89.4	76.5	77.3	47.0	47.8	77.2	80.1	48.5	51.9	56.1	58.1
	50	2.9	41	77.3	78.4	61.6	62.2	44.4	45.7	71.4	72.8	43.8	48.1	41.2	38.9
Peaceman Rachford	5	29.4	4	89.9	92.5	92.5	93.2	51.0	50.2	92.9	93.8	50.2	53.6	74.1	75.0
	10	14.6	***8***	92.7	94.0	92.4	93.1	50.8	49.7	***92.8***	***93.7***	50.8	55.1	73.5	75.8
	25	5.8	20	87.8	90.8	78.7	79.5	47.7	47.3	82.2	83.8	48.4	50.6	59.8	61.5
	50	2.8	42	75.1	79.6	63.3	63.5	45.7	47.0	70.5	71.9	46.2	48.5	40.0	37.9
PCG-SSOR	5	51.5	2	83.1	87.4	92.5	93.2	51.3	54.6	92.8	93.8	49.6	52.3	74.1	74.9
	10	35.6	3	89.8	91.9	92.0	93.0	50.7	53.9	92.4	93.7	49.5	53.2	73.2	75.6
	25	20.8	6	81.6	87.6	***92.8***	***93.5***	49.3	53.2	91.9	92.4	49.3	52.9	75.8	79.9
	50	15.3	***8***	80.5	90.4	91.6	91.8	45.4	50.0	91.0	91.5	49.4	54.2	74.2	77.0
PCG Cholesky	5	52.4	2	83.2	87.5	92.6	93.3	51.6	54.9	92.9	93.9	49.3	52.3	74.1	75.0
	10	36.1	3	90.8	92.5	92.6	93.7	51.1	54.4	***93.0***	***94.5***	49.3	52.9	73.2	75.6
	25	20.8	6	82.3	88.1	92.5	93.1	49.5	53.5	91.5	92.0	49.1	52.7	75.9	80.2
	50	14.9	***8***	80.6	90.3	91.6	91.8	45.5	50.2	91.0	91.6	49.4	54.3	74.1	76.7
PCG ADI Cholesky	5	52.0	2	92.2	93.8	92.0	92.4	54.0	52.5	92.1	92.6	54.4	57.6	75.3	77.2
	10	35.1	3	92.4	93.9	93.4	94.5	54.9	52.8	93.6	94.8	56.0	60.0	73.5	76.8
	25	20.4	6	89.6	91.7	94.9	95.7	52.5	51.1	***95.0***	***95.7***	54.4	57.5	74.5	74.9
	50	14.2	***8***	76.4	85.2	93.2	94.3	48.4	49.9	93.7	95.1	53.0	56.9	75.4	80.1

From Table 3.3 to Table 3.5, one can see that all the smoothed images achieve higher producer's and user's classification accuracies than the original image, on all classifiers, except for the Maximum Likelihood classifier on the Cuprite and Fake Leaves images, which is due to the fact that the reduced variability of the smoothed images makes inappropriate the statistical model of the ML classifier, since covariances might not be full.

In fact, the scale-space representation of hyperspectral imagery increases in general both classification accuracies, which means that the accuracy by class increases and the confusion between classes decreases. This should be expected from the fact that the scale-space representation reduces the variability within the different regions in the image and hence, reducing the overlapping that exists between the spectrums of different classes due to spectral variability (see Figures 3.27 to 3.30).

Table 3.4 Classification Accuracies Cuprite image.

Numerical method				Classification Accuracy (%)											
				ML		FLL		ED		ECHO		SAM		MF	
Scheme	μ/μ_0	t(s)	S	PA	UA	PA	UA	PA	UA	PA	UA	PA	UA	PA	UA
Original	-	0.00	-	87.3	92.2	92.2	93.1	56.1	62.8	***93.5***	***94.5***	85.9	87.5	66.0	74.8
Explicit	1	543.4	1	86.3	84.8	***97.1***	***97.3***	58.5	63.3	***97.1***	***97.3***	90.6	93.0	79.1	79.5
AOS	5	121.6	4	76.3	87.7	96.8	96.9	58.6	62.9	96.8	96.9	90.8	93.2	79.3	82.5
	10	53.6	***10***	72.6	90.8	***97.0***	***97.0***	58.7	62.8	***97.0***	***97.2***	90.7	93.3	78.3	81.1
	25	21.4	25	76.1	91.2	95.8	96.2	58.4	62.0	95.8	96.2	90.9	93.3	78.5	80.4
	50	10.7	51	79.1	91.5	94.9	95.5	58.3	63.8	94.9	95.5	91.0	93.2	77.6	79.9
ADI	5	112.9	5	77.4	82.9	96.9	97.1	58.2	63.0	96.9	97.1	90.8	93.1	79.5	82.8
	10	51.8	10	73.3	85.7	***97.1***	***97.3***	58.8	63.1	***97.1***	***97.3***	91.1	93.4	79.7	82.8
	25	20.3	***27***	46.9	67.9	96.9	97.1	58.8	62.7	96.9	97.1	90.9	97.1	78.7	81.9
	50	10.2	53	63.4	78.3	95.6	95.9	58.0	61.3	95.6	95.9	91.2	93.5	78.7	81.0
Douglas Rachford	5	143.0	***4***	83.2	91.2	***96.8***	***97.0***	58.6	63.0	***96.8***	***97.0***	91.0	93.3	80.0	83.1
	10	63.8	8	86.4	92.1	95.7	96.5	59.1	63.1	95.8	96.5	91.6	93.7	77.1	78.7
	25	28.3	19	87.0	92.4	94.3	95.4	59.3	64.4	94.4	95.9	91.0	92.8	75.9	81.0
	50	14.2	38	86.0	92.4	93.5	94.3	60.0	66.3	93.5	94.3	86.6	88.6	68.9	77.9
Peaceman Rachford	5	150.9	4	82.1	85.4	96.8	97.0	58.5	63.1	96.8	97.0	90.8	93.1	80.5	84.0
	10	68.8	***8***	79.5	90.8	***97.3***	***97.5***	58.3	62.6	***97.3***	***97.5***	90.9	93.3	79.7	82.7
	25	27.4	20	84.6	91.7	94.5	95.4	59.3	63.3	94.5	95.4	90.9	93.0	77.3	81.7
	50	13.5	40	83.7	91.6	92.8	93.6	59.2	65.1	93.2	94.0	85.7	85.9	68.7	77.4
PCG-SSOR	5	247.2	2	81.5	85.4	96.6	96.7	58.3	62.8	96.6	96.7	90.9	93.2	79.9	83.2
	10	169.0	3	79.8	86.3	96.8	97.0	58.6	62.7	96.8	97.0	91.0	93.3	79.7	83.0
	25	104.0	***5***	77.0	87.8	***97.0***	***97.2***	59.2	62.8	***97.0***	***97.2***	91.2	93.5	79.6	82.3
	50	66.9	8	77.6	91.3	96.3	96.6	58.5	62.2	96.3	96.6	91.3	93.7	78.6	81.2
PCG Cholesky	5	265.2	2	80.4	85.2	96.8	96.9	58.4	62.8	96.8	96.9	90.9	93.2	80.2	83.6
	10	176.1	3	79.8	86.2	96.9	97.1	58.5	62.7	96.9	97.1	91.0	93.3	79.7	83.2
	25	109.9	***5***	77.3	87.9	***97.0***	***97.2***	59.2	62.8	***97.0***	***97.2***	91.3	93.6	79.5	82.1
	50	67.0	8	77.7	91.3	96.3	96.6	58.5	62.2	96.3	96.6	91.3	93.7	78.4	81.1
PCG ADI Cholesky	5	284.3	2	87.5	87.9	95.9	96.2	57.9	62.6	95.9	96.2	92.2	93.9	80.3	83.6
	10	177.9	3	86.0	87.5	95.6	96.0	58.4	62.8	95.6	96.0	92.1	93.8	80.1	83.6
	25	108.2	5	78.4	85.1	96.6	96.9	59.1	63.7	96.6	96.9	91.5	93.6	79.5	83.0
	50	60.0	***9***	47.0	68.6	***96.8***	***97.0***	59.3	63.2	***96.8***	***97.0***	91.0	93.3	79.0	82.4

The bad performance of ML on the Cuprite and False Leaves images can be explained by the fact that these images have less spectral variability (see Figures 3.27 to 3.30) than the NW Indian Pines image and we are using a higher threshold value α (see

Equation 3.6) which means that the diffusion is higher on these images and the variability in the image is reduced more. Hence, the full covariance model used in the ML classifier is not the best statistical model for these two images. On the other hand, the Fisher Linear Discriminant classifier benefits from the reduction in the variability within the image classes [*Landgrebe*, 2003] produced by the scale-space representation of hyperspectral imagery, hence, it has the highest classification accuracies on all the images considered here. A simpler classifier such as FLD worked well, thanks to the reduced variability in the smoothed images

Table 3.5 Classification Accuracies False Leaves image.

Numerical method				Classification Accuracy (%)											
				ML		FLL		ED		ECHO		SAM		MF	
Scheme	μ/μ_0	t(s)	S	PA	UA	PA	UA	PA	UA	PA	UA	PA	UA	PA	UA
Original	-	0.0	-	79.3	81.7	77.1	79.4	55.9	59.7	***82.0***	***84.2***	74.6	76.4	48.7	52.3
Explicit	1	954.2	1	47.7	50.5	93.4	96.1	60.1	65.7	***94.3***	***96.9***	78.1	80.4	73.6	77.2
AOS	5	230.8	4	35.3	50.2	93.1	95.6	60.5	68.0	93.0	95.5	76.2	79.8	74.7	79.1
	10	114.4	8	43.1	54.2	93.5	95.8	60.2	66.7	94.1	96.3	76.8	79.5	74.1	77.5
	20	52.4	***18***	65.8	85.2	94.5	96.2	59.8	65.4	***94.7***	***96.4***	78.1	80.4	72.1	75.1
	50	20.7	46	88.8	92.1	88.5	91.2	58.7	63.7	88.2	91.4	77.9	80.1	64.3	69.3
ADI	5	212.3	4	35.5	50.2	92.7	95.2	60.5	68.4	92.6	95.2	76.1	80.3	74.8	79.4
	10	95.5	10	34.0	50.1	92.9	95.4	60.2	67.2	92.8	95.3	76.4	79.8	75.0	78.8
	20	47.6	***20***	41.2	54.1	93.6	95.8	60.0	66.1	***94.1***	***96.3***	77.8	80.3	74.0	77.0
	50	20.0	48	74.9	89.0	89.7	93.1	58.9	64.0	89.1	93.3	78.4	80.4	67.2	70.6
Douglas Rachford	5	279.7	3	37.6	50.2	92.4	95.0	60.8	69.0	92.3	95.0	76.0	80.0	75.6	79.8
	10	145.6	***7***	81.5	91.9	93.7	96.2	60.6	68.0	***94.1***	***96.6***	76.6	79.9	75.1	79.3
	20	63.3	15	94.6	96.0	90.0	93.6	59.6	65.5	88.8	93.9	77.3	79.7	66.7	73.7
	50	25.4	38	85.0	86.8	83.2	85.6	57.5	62.2	84.1	88.7	76.6	78.9	55.9	58.9
Peaceman Rachford	5	281.9	3	35.5	50.2	92.1	94.7	60.9	68.9	92.1	94.7	75.8	80.0	75.0	79.4
	10	133.8	7	41.3	61.8	92.6	95.1	60.8	68.7	92.4	95.0	76.5	80.4	74.9	79.4
	20	62.5	***15***	85.4	93.3	93.6	95.9	60.4	66.9	***94.4***	***96.6***	77.1	79.7	71.5	76.3
	50	24.9	38	81.5	81.5	84.6	87.4	57.8	63.0	84.5	89.1	77.0	89.1	59.0	62.1
PCG-SSOR	5	381.8	2	36.7	50.2	92.4	95.0	60.9	68.7	92.4	95.0	76.0	80.0	74.9	79.6
	10	255.6	4	37.8	50.2	93.1	95.6	60.7	67.4	93.0	95.5	77.0	80.0	75.2	79.3
	20	181.9	***5***	41.0	50.3	94.0	96.4	60.3	66.3	***94.2***	***96.5***	77.8	80.2	74.4	77.5
	50	88.7	11	80.5	90.5	89.6	93.0	58.9	64.1	88.7	93.1	78.6	80.5	67.2	71.1
PCG Cholesky	5	389.0	2	37.0	50.2	92.3	95.0	60.9	68.7	92.3	94.9	76.0	80.0	75.2	79.8
	10	240.0	4	39.0	50.3	92.9	95.4	60.7	67.3	92.8	95.4	76.8	80.2	75.0	79.1
	20	167.9	***6***	42.3	50.4	93.8	96.2	60.4	66.4	***94.2***	***96.6***	77.5	80.0	74.4	77.6
	50	71.4	13	79.2	90.4	89.9	93.4	59.1	64.3	88.8	93.3	78.7	80.5	67.5	71.1
PCG ADI Cholesky	5	418.1	2	48.3	61.0	87.3	91.2	60.7	68.7	87.3	91.2	75.0	76.5	77.1	83.0
	10	277.5	3	42.4	60.1	88.6	92.4	60.4	69.2	88.6	92.4	74.6	75.2	76.3	81.6
	20	167.9	6	38.0	49.8	89.4	92.8	60.3	69.4	89.4	92.8	74.8	77.0	75.5	80.3
	50	69.5	***14***	42.0	53.0	***94.5***	***96.6***	59.9	66.4	***94.5***	***96.6***	77.9	80.2	74.2	77.5

In general, ECHO is just a little superior than FLD in terms of classification accuracy

on all the images considered here. The difference between ECHO and FLD reduces as the smoothing increases, as can be appreciated on Table 3.4 and Table 3.5, where $\alpha = 0.015$ (see Equation 3.6), meanwhile the difference is higher in the NW Indian Pines image, where $\alpha = 0.012$. ECHO tries to homogenize the image before classifying it, by choosing a small window (2x2 pixels in our simulations). Hence, if the region within the windows is already smooth, due to diffusion, the difference between ECHO and FLD is reduced.

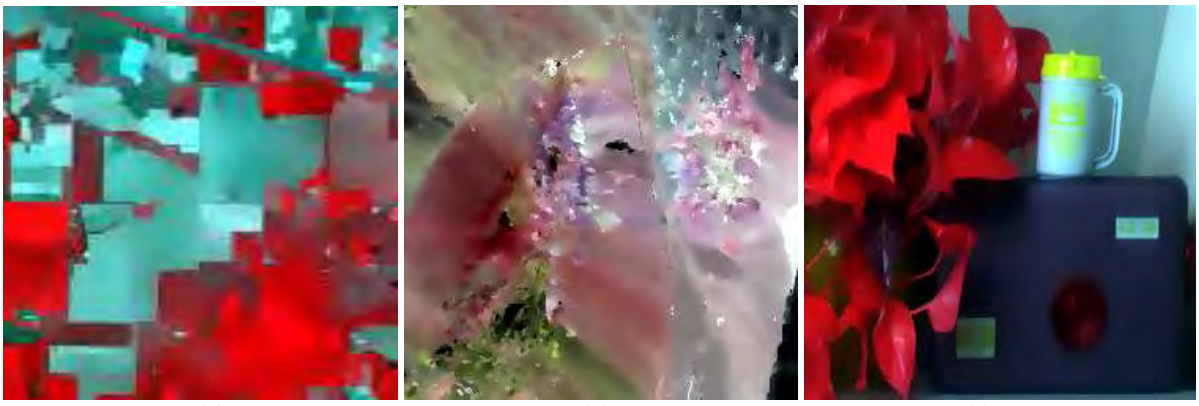
The remaining classifiers, ED, SAM, and MF are in general very insensitive to the scale step, but in general, they do not achieve good classification accuracies, except for the SAM classifier on the Cuprite image. The relative good performance of SAM on this image agrees with the reported studies on mineral classification using the spectral angle and a high number of bands [*Girouard et al*, 2004].

In terms of the implemented numerical methods, AOS and ADI classification accuracies are very insensitive to the scale-step, μ, achieving high speedups and classification accuracies up to $\mu \leq 25\mu_0$ on all the images analyzed here. On the other hand, the Douglas-Rachford and Peaceman-Rachford methods are more sensitive to the scale step, achieving high classification accuracy only up to $\mu \leq 10\mu_0$, which limits their speedup.

PCG methods are very insensitive to the scale step and all behave similarly in terms of classification accuracy. The best classification accuracies and speed-ups are achieved by PCG-Cholesky initialized by ADI-LOD. Notice that with exception of AOS for the NW Indian Pines image, all the speedups that achieved good classification accuracies are within the range defined on previous section, considering the limits imposed by the accuracy of the computed solution and the introduction of artifacts in the images. This means that in general,

we must have good accuracy of the solution to the anisotropic diffusion PDE to achieve good classification accuracies.

It is noteworthy, though, that AOS has such a good performance in terms of classification accuracy (Table 3.3) for the Indian Pines image using a very large step, since we know that AOS is not very accurate at scale steps larger than $20\mu_0$ (see Figures 3.13 to 3.16). We believe that this occurs because α is relatively small here; hence the magnitude of the error is lower. AOS is also symmetric and hence the error introduced can be reduced by a classifier as ECHO, which tends to average out random variations in a small window.



(a) **(b)** **(c)**

Figure 3.26 RGB composite showing the smoothed hyperspectral images, a) Indian Pines (bands 47, 24, and 14), b)Cuprite (bands 183, 193, and 207), and c) False Leaves (bands 90, 68, and 29).

Figure 3.26 shows the smoothed NW Indian Pines, Cuprite and False Leaves images using the numerical methods that achieve the best classification accuracies, indicated from Table 3.3 to Table 3.5. Notice the strong reduction in the spatial variability on each image, while most of the semantically meaningful edges are preserved. In particular, the Indian Pines image looks as it were already segmented and the False Leaves image has no visible noise. The Cuprite image looks more blurred than the other two images, because the edges in this image are not sharp. However, the smoothed Cuprite image does achieves better

classification accuracies than the original image (Table 3.4).

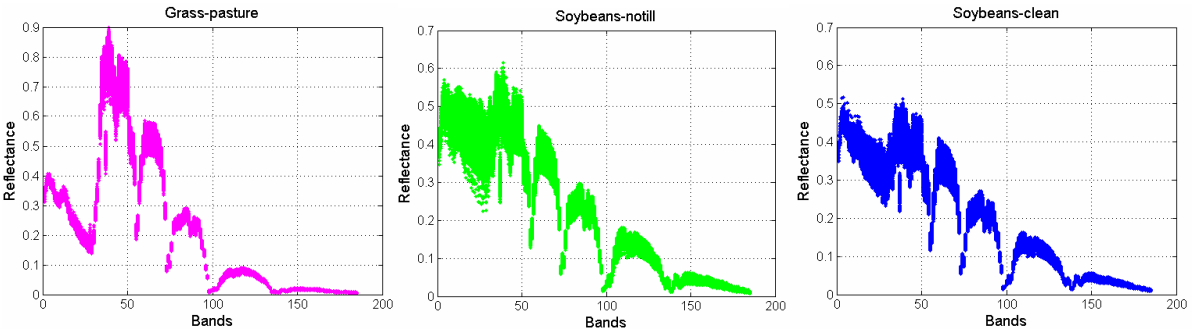


Figure 3.27 Superimposed spectra showing the spectral variability within three crops in the Indian Pines image.

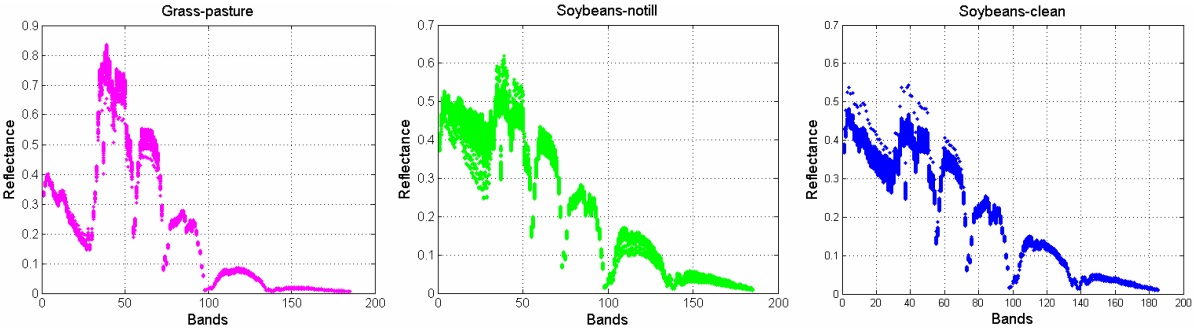
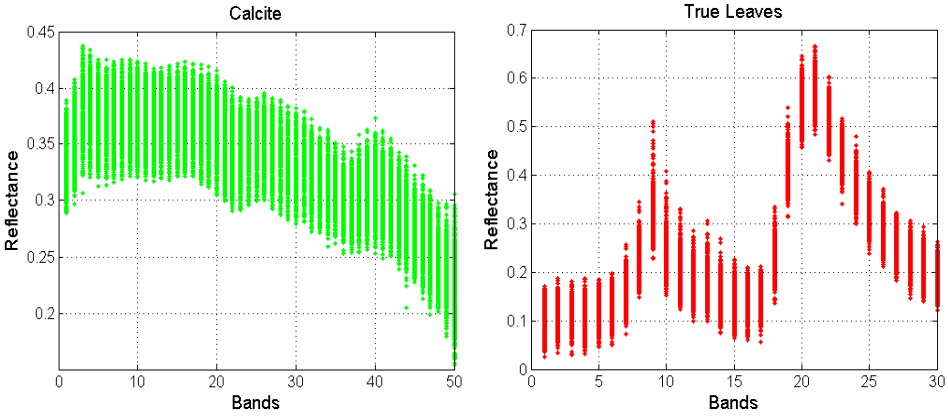


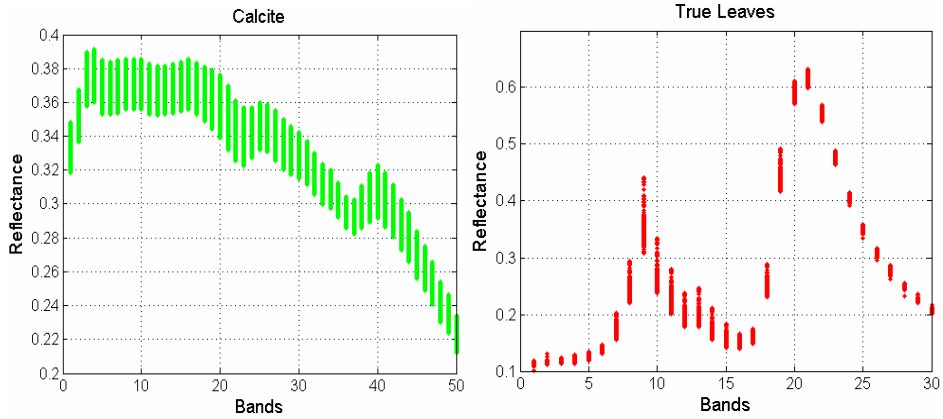
Figure 3.28 Superimposed spectra showing the spectral variability within three crops in the smoothed Indian Pines image.

Figure 3.27 shows the variability in the spectral signature of three selected fields (Grass-pasture, Soybeans-notill, and Soybeans-clean) in the NW Indian Pines image. Figure 3.28 shows the variability in the spectral signatures in the smoothed Indian Pines image for the same fields. Figure 3.29 show the variability in the spectral signature of a region of Calcite within the Cuprite image and of true leaves, within the noisy False Leaves image.



(a) (b)

Figure 3.29 Superimposed spectra showing the spectral variability in the a) Cuprite and b) False Leaves image.



(a) (b)

Figure 3.30 Superimposed spectra showing the spectral variability in the smoothed a) Cuprite and b) False Leaves image.

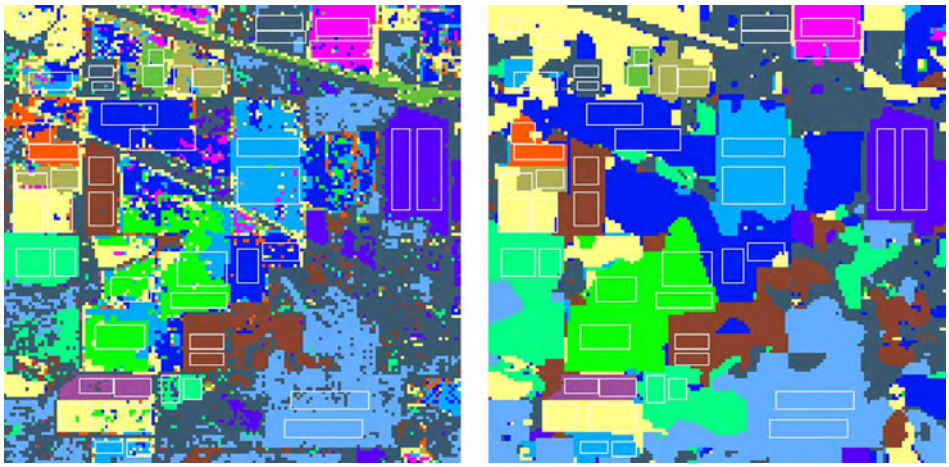
Figure 3.30 shows the spectral variability of the smoothed Cuprite and False leaves images, within the same regions of Calcite and true leaves selected shown in Figure 3.29. Figure 3.31 indicates the regions selected on the Indian Pines, Cuprite and False Leaves images used to construct Figures 3.27 to 3.30.

(a) (b) (c)

Figure 3.31 Selected regions where the spectral signatures where extracted to form Figures 3.27 to 3.30 on a) Indian Pines, b) Cuprite, and c) False Leaves images.

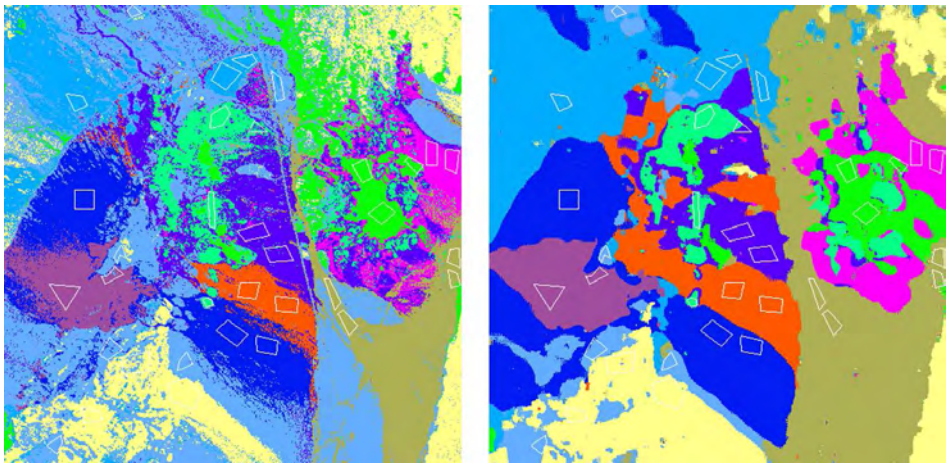
It is remarkable the reduction in the variability of the spectral signatures shown, on the smoothed images. This reduction was enough to increase classification accuracy as reported from Table 3.3 to Table 3.5. Further reduction in the spectral variability would require a higher value of α that might lead to the destruction of important edges in the image.

In order to see the effect of nonlinear smoothing on the classification of the full real hyperspectral images used here, we present from Figures 3.32 to 3.34 the classification maps of the original and smoothed images that achieved the highest classification accuracies (Table 3.3 to Table 3.5). It can be noticed from these figures that not only the testing samples improve their classification accuracy, but also that the smoothed images produce classification maps that look more accurate.



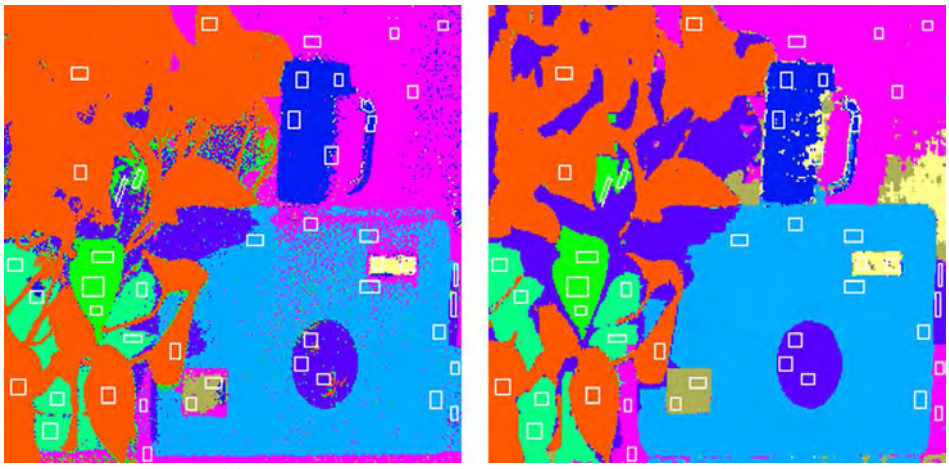
(a) (b)

Figure 3.32 Indian Pines classification map: a) original, b) smoothed.



(a) (b)

Figure 3.33 Cuprite classification map: a) original, b) smoothed.



(a) (b)

Figure 3.34 Fake Leaves classification map: a) original b) smoothed.

3.3.4 Concluding remarks

In this chapter, we showed that nonlinear diffusion can improve significantly image classification accuracies by reducing both, the spatial and spectral variability in hyperspectral imagery. AOS and ADI semi-implicit schemes offer good performance in terms of accuracy and speedup of the computed solution of the nonlinear PDE. PCG linear solvers are less sensitive to the scale step as the approximated ADI and AOS schemes, which mean that higher values of μ can be used. However, PCG methods also require more space, and finding a good preconditioner is still an art.

4 MULTISCALE REPRESENTATION AND SEGMENTATION OF HYPERSPECTRAL IMAGERY

> The challenge is to understand the image really on all the levels simultaneously, and not as an unrelated set of derived images at different levels of blurring.
> JAN J. KOENDERINK

In the previous chapter, we showed that the approximated semi-implicit discretization methods, ADI and AOS, are algorithmically scalable, i.e. they have linear time complexity, achieving acceptable accuracies (no visible artifacts) and speed ups of around 20 times over the explicit scheme. On the other hand, better accuracies can be obtained using the preconditioned conjugated gradient method (PCG). However, the preconditioners used did not scale well and hence the PCG methods implemented were slower than the approximated semi-implicit methods.

Even tough the color composites for the hyperspectral images, smoothed with anisotropic diffusion, look as if they were already segmented (see Figure 3.26), they are not, and we still need a way of segmenting them, within the scale-space framework presented on the previous chapter. From the myriad of segmentation algorithms that exists nowadays for grayscale images [*Zhang*, 2001], a novel and fast segmentation algorithm proposed by [*Sharon, et al*, 2000] called our attention given that it is inspired by Algebraic Multigrid

(AMG) [*Brigss, et al*, 2000], which is a numerical method used to solve, with great accuracy and scalability, discrete PDEs.

We did not use AMG on the previous chapter, because its implementation is much more complicated than the semi-implicit and PCG methods, and we found very good results using the simpler AOS and ADI schemes. Hence, the purpose of this chapter is to integrate the formal scale-space framework introduced on the previous chapter with a segmentation algorithm that is based on this framework.

The segmentation algorithm of [*Sharon, et al*, 2000] can be regarded as hierarchical normalized cuts. Normalized cuts is a discrete-optimization inspired segmentation algorithm proposed by [*Cox et al*, 1996] and improved later by [*Shi and Malik*, 1997]. Normalized cuts translates the segmentation problem into a graph partitioning problem, where the pixels in the image are vertices in the graph and the similarity between neighboring pixels is represented as weighted edges of the graph. The computationally expensive graph-partitioning problem in the fine grid of the image (normalized cuts) can be brought to a coarse scale, where it can be solved with much lower computational cost, and then propagated back to the finest level. In fact, it has been argued that solving the segmentation problem at a coarser scale produces better segmentation results than solving it in the finest scale, where only local information is used [*Sharon et al*, 2000, 2003]. On a hierarchical multiscale representation of the image, statistic and geometric information (shapes) can be gathered from the fine to the coarser levels, so that local and global information are both available to the segmentation process [*Sharon et al*, 2000, 2003]. Recently, an extension of [*Sharon et al*, 2000] algorithm has also been proposed for multispectral imagery [*Gali and de Candia*, 2005].

However, [*Sharon, et al*, 2000; *Gali and de Candia*, 2005] algorithms are inspired by AMG, but they are not properly AMG since they do not solve any PDE generating a scale-space representation of the image. We propose here to integrate the well-founded scale-space representation of an image using geometric PDEs, with a modified version of the AMG-based segmentation algorithm that naturally fits within this framework.

As mentioned before, segmentation can be cast into a graph partitioning problem. An image can be represented by a graph, where the pixels are the vertices and the edges connect each vertex to their closest neighbors (e.g., 4 or 8 neighborhood). Associated to the edges there is a weighting function that indicates the degree of similarity between the vertices. The segmentation problem can be expressed now as removing edges of the graph i.e. finding the graph cut that minimizes the weight of the edges removed [*Shi and Malik*, 1997]. The optimal graph cut is in general an NP-hard problem [*Shi and Malik*, 1997] and hence, fast suboptimal solutions are used. The contribution of [*Sharon, et al*, 2000] consisted of obtaining an approximation to the optimal graph cut, not in the original (large) grid of the image, but, as in AMG, on a much coarser scale, where a suboptimal solution can be found easily and then propagated back to the finest scale. In this way, they achieve image segmentation in linear time complexity, and the quality of the segmentation is often better than the one obtained with normalized cuts [*Sharon, et al*, 2000]. In addition, as the multiscale representation of the image is constructed, statistics can be computed recursively from the different regions in the image, introducing global measures in the segmentation process that are not available at the finest grid [*Sharon et al*, 2003].

This chapter is organized as follows. First, we present the scale-space representation and segmentation of hyperspectral imagery using AMG, and then we present the implementation details of the algorithm and its complexity analysis. Second, we present the performance tests and segmentation results using the four hyperspectral images. Finally, we present some concluding remarks.

4.1 Scale-space representation and segmentation of hyperspectral imagery

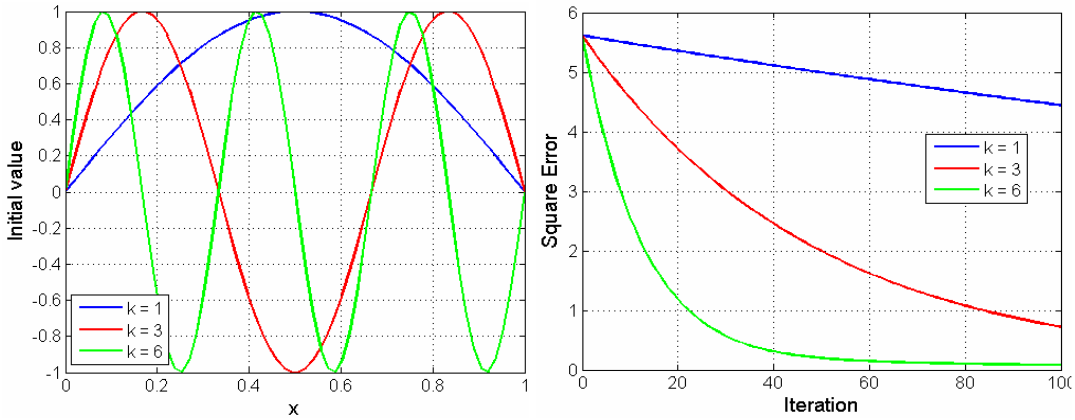
Multigrid methods [*Brigss, et al*, 2000] come from the analysis of classic relaxation methods for solving linear systems of equations. Classic iterative methods reduce efficiently the high frequency components of the error, although they are extremely inefficient to reduce the low frequency components and hence, they converge slowly. To illustrate this, let us consider a simple one-dimensional boundary value example (e.g. the Poisson equation),

$$\frac{\partial^2 u}{\partial x^2} = 0, \quad x \in (0,1), \quad u(0) = u(1) = 0,$$

which can be discretized as, $u_{j-1} - 2u_j + u_{j+1} = 0,\ 1 \le j \le n\text{-}1,\ u_0 = u_n = 0,$ and n is the number of intervals on which the interval (0, 1) has been discretized. The Gauss-Seidel (GS) iterative method solves the previous equation as, $u_j^i = \frac{1}{2}\left(u_{j-1}^i + u_{j+1}^{i-1}\right)$, where $i > 0$ is the iteration number. Let us consider now initial conditions of the form,

$$u_j^0 = \sin\left(\frac{k\pi j}{n}\right),$$

where $k \in \mathbb{N}$ is the wave number. Given that the steady state solution of the Poisson equation is $u = 0$, we can easily compute the error, u^i, as a function of the number of iterations of the GS method.



(a) (b)

Figure 4.1 a) Initial Error at different frequencies, b) Reduction in the error as a function of the number of iterations.

Figure 4.1.a shows the initial error for $k = 1$, 3, and 6, i.e. at increasing frequency values. Figure 4.1.b shows the square error as a function of the iteration number for each one of the three modes indicated on Figure 4.1.a. It can be noticed that the high frequency error ($k = 6$) is quickly reduced by the iterative method, while the medium and low frequency errors are reduced much more slowly. By Fourier analysis, we know that the error term can be expressed as a superposition of sinusoidal components at different frequencies. Hence, as the example illustrates, the high frequency components of the error would be eliminated quickly by relaxation methods such as GS, but the low frequency components would be reduced at a very low rate, causing slow convergence.

Multigrid methods aim to reduce the error components at all frequencies, in linear time complexity. Multigrid involves two complementary processes: relaxation and coarse-

grid correction. Coarse-grid correction consists on transferring information from fine to coarser grids via a sampling operation. The coarsening process is continued until a relatively small grid is reached where the linear system can be solved exactly, with little computational cost. The solution is then propagated back to the finest level via interpolation operations. The coarsening operation displaces the low frequency components of the error to higher frequencies in the coarse grid, where classical relaxation methods reduce them efficiently [*Brigss, et al*, 2000]. The relaxation can be accomplished by a simple iterative method such as Jacobi or Gauss-Seidel.

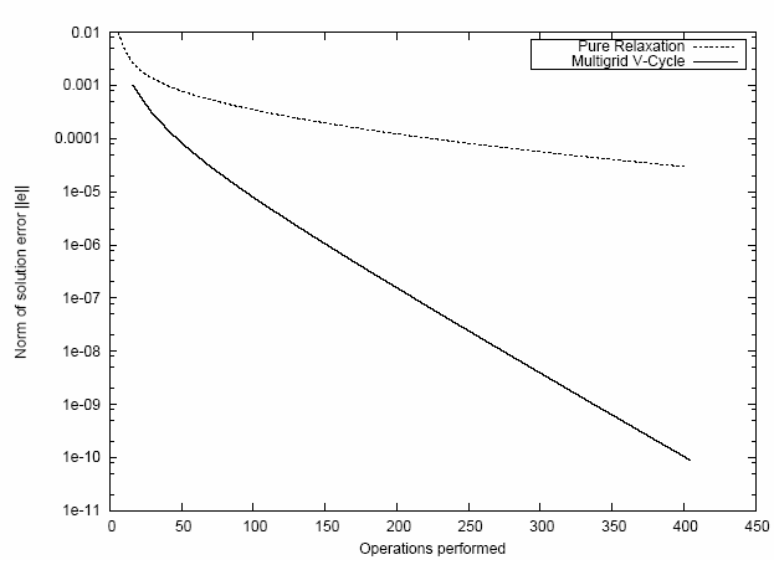


Figure 4.2 Comparison of cost of convergence of multigrid and a pure relaxation method to solve the anisotropic diffusion equation (taken from [*Long*, 2005]).

Figure 4.2 shows the reduction in the error norm obtained by solving the anisotropic diffusion PDE using AMG and pure relaxation (Gauss-Seidel) methods, as a function of the number of pure relaxations made [*Long*, 2005]. The V-cycle indicated on Figure 4.3 will be explained later, in Section 4.1.2. As can be seen from this figure, pure relaxation methods

reduce the error norm very slowly, while AMG reduces it as 10^{-r}, being r the rate of convergence, with a computational cost that increases only linearly.

The method used to coarsen the grid defines if the multigrid method is geometric or algebraic. Geometric multigrid samples the previous grid uniformly. Algebraic multigrid uses an algebraic coarsening, i.e. the grid is sampled non-uniformly, according to the structure of the matrix that defines the linear system, which in our case is **G**, the diffusion matrix. However, it is well known that classical geometric multigrid is not robust on PDEs with highly nonlinear coefficients, as is the case of the nonlinear diffusion PDE [*Brandt et al*, 1992]. As [*Kimmel and Yavneh*, 2003] had shown, algebraic multigrid is more robust for image analysis. Recently, [*Rand and Keenan*, 2003] introduced geometric multigrid and Markov Random Fields to segment hyperspectral imagery. Multigrid is used in [*Rand and Keenan*, 2003] to minimize an energy functional by stochastic relaxation [*Geman and Geman*, 1984]. The main disadvantages of this approach are its high computational cost and the simplifying assumptions needed to make the stochastic approach mathematically tractable.

AMG requires the construction of a multigrid structure that starts with the finest grid of the original image on its base and coarser grids are added "below it" forming an inverted pyramid, as illustrated on Figure 4.3.

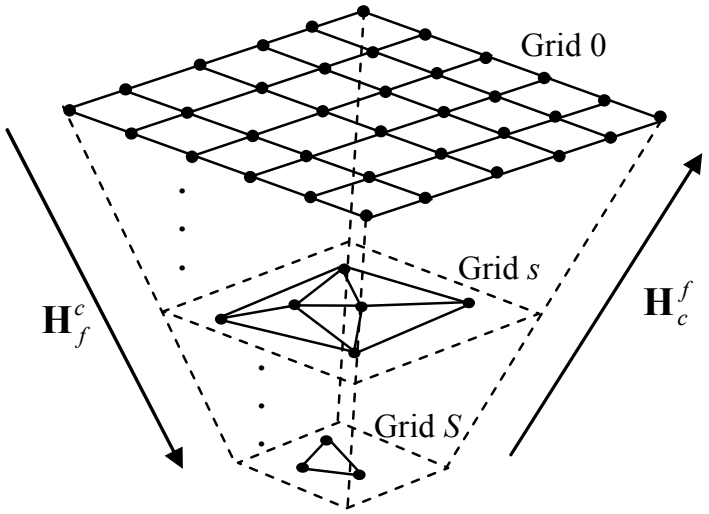


Figure 4.3 Typical multigrid structure. Note that the structure is not necessarily a Cartesian grid.

We use standard graph theory notation (V^s, E^s) to identify the sets of vertices (V^s) and edges (E^s) of the multigrid structure, where the superscript index s indicates the grid level, starting with $s = 0$ for the finest grid and $s = S$ for the coarsest. On this setting, the original hyperspectral image is represented by an undirected graph (V^0, E^0), where the set of vertices V^0 corresponds to the vector-valued pixels in the image, and E^0 is the set of edges connecting each node to its four closest neighbors with weights g_{ij}^0 given by Equation 3.6.

The sampling (restriction) operation, denoted here as $\mathbf{H}_f^c$, and the interpolation (prolongation) operation, denoted as $\mathbf{H}_c^f$, are also indicated on Figure 4.3. Associated to the graph, there is a similarity function g that assigns a weight to each edge $(i, j) \in E^s$ on each grid of the multigrid structure (Figure 4.3), with $0 \leq s \leq S$, being S the coarsest grid. The nonlinear diffusion coefficient, given by Equation 3.6, corresponds to the similarity function g at the finest grid, $s = 0$.

Now, we are going to see in more detail, how the multigrid structure is constructed and how it is used to solve Equation 3.14.

4.1.1 Multigrid structure

The construction of the multigrid structure requires two main steps: selecting the next set of vertices V^{s+1} from the current grid (V^s, E^s), $0 \le s \le S$-1 and connecting the nodes in V^{s+1} to obtain E^{s+1}. In AMG, the vertices V^{s+1} must be sparse in V^s and independent of each other as much as possible. We use the selection mechanism described in [*Sharon et al*, 2000], since it satisfies these requirements. For completeness, we describe the method here in detail.

The mechanism used to select which vertices from (V^s, E^s) will form the next grid is a greedy strategy, where the vertices are first sorted in decreasing order, according to their mass m_i^s, which is a measure of how many pixels in the finest grid can be assigned to a given vertex on a coarse grid. At grid s=0 the mass of all vertices is set $m_i^0 = 1$. The idea of sorting the vertices is that vertices that are representative of a large number of pixels on the finest grid would be more likely selected for the next grid. The selection process consists of the following three steps,

- Sort the set of vertices V^s in decreasing order of mass.
- Initialize $V^{s+1} = \{i_0\}$, where i_0 is the first element in the ordered set V^s.
- For each $i \in V^s \setminus V^{s+1}$:

  $$\text{if } \sum_{j \in V^{s+1}} g_{ij}^s \Big/ \sum_{(i,j) \in E^s} g_{ij}^s \le \tau \Rightarrow V^{s+1} = V^{s+1} \cup \{i\},$$

where, $0 < \tau < 1$ is a threshold value below which we say that vertex i is independent of the selected vertices, V^{s+1}, so far. Notice that the first coarse grid can be obtained now with the previous algorithm, since we have already defined m_i^0, g_{ij}^0 and E^0 and there is no needed to

sort the finest grid, since all the masses are equal. To obtain the coarser grids, we require to compute m_i^s, g_{ij}^s and the set of edges E^s for $s > 0$. We will explain this in detail and the criteria to stop coarsening, after making some important observations about the sorting algorithm.

The sorting algorithm must run in linear time to keep the overall complexity of the algorithm linear. The sorting algorithm used in [*Sharon et al*, 2000] is bucket sort [*Cormen, et al*, 2001], which runs in linear time, on average, assuming a uniform distribution of the mass in the [0 1] range, after normalization. We used instead radix-sort [*Cormen, et al*, 2001], which always runs in linear time, irrespectively of the distribution of the data. Since, radix-sort only works with integer values, we approximate m_i^s to its nearest integer value. This way, radix-sort orders the masses with little selectivity at first, since initially the differences are mainly fractional, but as we coarsen the grid, radix-sort becomes much more selective. We can make more selective radix-sort on the first levels by multiplying the mass by a constant factor of 100, for instance. However, experimentally, we found better segmentation results being less sensitive to the small differences in the first levels, instead of using an absolute ordering of the masses, as bucket sort does. Besides, it is not true that the distribution of mass is uniform as it is assumed by bucket sort, especially at coarser scales, where there can be strong differences between pixels representing large and small regions in the image.

Once the vertices of the first coarse grid are selected, we can compute the dependences of the vertices in $V^s \setminus V^{s+1}$ to the vertices in V^{s+1} and the masses, for $s = 0, \ldots, S\text{-}1$ as,

$$\forall i \in \mathrm{V}^s \setminus \mathrm{V}^{s+1}, j \in \mathrm{V}^{s+1} : w_{ij}^s = w_{ji}^s = \frac{g_{ij}^s}{\sum_{k \in \mathrm{V}^{s+1}} g_{ik}^s}, \quad \textbf{4.1}$$

$$\forall i \in \mathrm{V}^{s+1} : m_i^{s+1} = m_i^s + \sum_{j \in \mathrm{V}^{s+1} \setminus \mathrm{V}^s} w_{ij}^s, \quad \textbf{4.2}$$

where, w_{ij}^s indicates how much vertex $i \in \mathrm{V}^s \setminus \mathrm{V}^{s+1}$ depends on vertex $j \in \mathrm{V}^{s+1}$. Notice that Equation 4.1 enables a multiscale soft-segmentation of the image, where pixels on each grid have a degree of attachment $0 \le w_{ij}^s \le 1$ to pixels selected at coarser levels. Also, notice that if $i, j \in \mathrm{V}^{s+1}$ or $i, j \in \mathrm{V}^s$ then $w_{ij}^s = 0$. This way, the vertices in $\mathrm{V}^s \setminus \mathrm{V}^{s+1}$ depend only on vertices in V^{s+1}, which intends to translate the fine grid problem to the coarse grid.

One can think here in gathering statistics from the previous levels as in [*Sharon et al*, 2003]. However, given the little development of texture measures for hyperspectral imagery and the computational cost of dimension reduction methods (such as principal components) that are employed to make feasible the use of second and higher order statistical discrimination methods (such as maximum likelihood, maximum a posteriori) [*Landgrebe*, 2002], we only use here mean intensities.

Let the hyperspectral image at grid s be $\mathbf{U}^s = \begin{bmatrix} \mathbf{u}_0^s & \mathbf{u}_1^s & \cdots & \mathbf{u}_{\nu_s - 1}^s \end{bmatrix}^{\mathrm{T}}$, where ν_s is the number of vertices at grid s and $\mathbf{u}_i$ is the spectral vector at pixel i^{th}, $1 \le i < \nu_s$. The mean spectral intensity at grid $s + 1$ is given by

$$\forall i \in \mathrm{V}^{s+1} : \mathbf{u}_i^{s+1} = \frac{\mathbf{u}_i^s + \sum_{j \in \mathrm{V}^s \setminus \mathrm{V}^{s+1}} w_{ij}^s \mathbf{u}_j^s}{1 + \sum_{j \in \mathrm{V}^s \setminus \mathrm{V}^{s+1}} w_{ij}^s}. \quad \textbf{4.3}$$

Notice that Equation 4.3 corresponds to the weighted mean vector-valued intensity, where

the spectral signatures of the vertices $j \in V^s \backslash V^{s+1}$ influenced by pixel $i \in V^{s+1}$ are weighted according to their dependence on i. Notice also that Equation 4.3 defines the restriction operation (coarsening of the pyramid), $\mathbf{H}_f^c$, which in matrix format is given by

$$\mathbf{U}^{s+1} = \mathbf{H}_f^c \mathbf{U}^s, \quad \left[H_f^c\right]_{ij} = \frac{w_{ij}}{1 + \sum_{j \in V^s \backslash V^{s+1}} w_{ij}^s}. \tag{4.4}$$

We need now to connect the vertices in V^{s+1}. This is done by first defining the interpolation operator and the corresponding geometric weighting g for all the vertices in the new level $s+1$. By the Garlekin condition [*Brigss, et al*, 2000], $\mathbf{G}^{s+1} = \mathbf{H}_f^c \mathbf{G}^s \mathbf{H}_c^f$, hence, we need to define the interpolation operation, $\mathbf{H}_c^f$. Since, we are working with mean spectrums, the simplest linear interpolation operation is given by,

$$\forall i \in V^s : \mathbf{u}_i^s = \begin{cases} \mathbf{u}_i^{s+1}, \text{if } i \in V^{s+1} \\ \sum_{j \in V^{s+1}} w_{ij}^s \mathbf{u}_j^{s+1}, \text{if } i \notin V^{s+1} \end{cases}, \tag{4.5}$$

which, in matrix-vector notation can be restated as,

$$\mathbf{U}^s = \mathbf{H}_c^f \mathbf{U}^{s+1}, \quad \left[H_c^f\right]_{ij} = w_{ij}^s \ , \tag{4.6}$$

From Equations 4.5 and 4.6,

$$\left[G^{s+1}\right]_{ij} = \frac{1}{1 + \sum_{j \in V^s \backslash V^{s+1}} w_{ij}^s} \sum_{k,l \in V^s} w_{ik}^s g_{kl}^s w_{lj}^s . \tag{4.7}$$

Sharon proposed a very similar equation called Iterated Weighted Aggregation (IWA) to connect the vertices on the coarse grid [*Sharon et al*, 2000]. However, while IWA was proposed as an approximation to $\mathbf{G}^{s+1}$ within a minimization problem, Equation 4.7

corresponds exactly to $\mathbf{G}^{s+1}$ in our AMG setup, as given by the Garlekin condition. It can be noticed that Equation 4.6 only considers local measures accumulated from grid 0 up to the coarser grids, as in IWA.

As in [*Sharon et al*, 2000], we introduced a global measure to steer the segmentation processes, which depends on the mean spectrums computed for each coarse vertex,

$$g_{kl}^{s+1} = \left(\frac{1}{1 + \sum_{j \in \mathrm{V}^s \setminus \mathrm{V}^{s+1}} w_{ij}^s} \sum_{k,l \in \mathrm{V}^s} w_{ik}^s g_{kl}^s w_{lj}^s \right) e^{-\alpha\theta\left(\mathbf{u}_k^{s+1}, \mathbf{u}_l^{s+1}\right)}, \qquad \textbf{4.8}$$

where, θ is a similarity metric as defined in Section 3.1.

Experimentally (see Section 4.3), we found that using Equation 4.8 improves the rate of convergence of AMG over Equation 4.7. This result shows the synergy that exists between the smoothing and segmentation processes. We are translating the PDE and segmentation problems to coarser grids, but on coarser grids, the relationships between the vertices are not completely expressed by local measures and must also include global measures. Notice here that global measures alone are not enough to discriminate between different segments with similar mean spectrums.

Finally, once $\mathbf{G}^{s+1}$ is computed, we can determine the set of edges on grid $s + 1$ as,

$$\mathrm{E}^{s+1} = \left\{ (i,j) : i,j \in \mathrm{V}^{s+1} \wedge g_{ij}^s > 0 \right\}. \qquad \textbf{4.9}$$

4.1.2 AMG solver

Let us restate here Equation 3.14, for greater clarity,

$$\left(\mathbf{I} - \mu\mathbf{G}^k\right)\mathbf{U}^{k+1} = \mathbf{U}^k, \quad k = 0, \cdots, K-1.$$

We will use AMG to solve Equation 3.14 on each scale-step. In the remaining of this section, we will drop the dependence on the scale-step *k*, since the solution sought with AMG will be always $\mathbf{U}^{k+1}$, for $k = 0, \ldots, K\text{-}1$.

Figure 4.4 shows a schematic of the same multigrid structure presented in Figure 4.3 4.1, showing more clearly the V-cycle. As indicated on Figure 4.4, the image at grid s = 0 is coarsened downto grid S, solving then exactly Equation 3.14 at this scale, and then propagating back the solution to the finest grid. Usually a single V-cycle is not enough to obtain good accuracy of the computed solution, $\mathbf{U}_{n+1}$. Hence, the first V-cycle starts with the image $\mathbf{U}_n$, but the next V-cycles starts with the approximation obtained to $\mathbf{U}_{n+1}$, in the previous V-cycle.

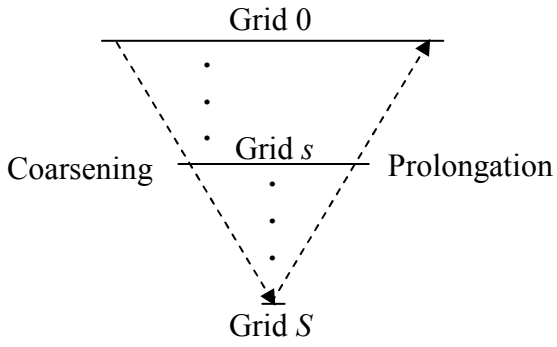


Figure 4.4 Schematics for a V-cycle in Multigrid.

The V-cycle algorithm can be divided in three phases. In the coarsening phase (Figure 4.4), the different components of the error, represented by $\mathbf{X}^s$, $s > 0$, are estimated by relaxation of the residual equation $\left(\mathbf{I} - \mu\mathbf{G}^s\right)\mathbf{X}^s = \mathbf{F}^s$, where $\mathbf{F}^s$, $s > 0$, is the residual [*Brigss, et al*, 2000] at scale *s*. In the coarsest grid, *S*, the component of the error $\mathbf{X}^S$ is computed exactly by Gaussian elimination. In the prolongation phase, the different components of the error are accumulated back to the finest grid as $\mathbf{X}^s = \mathbf{X}^s + \mathbf{H}_c^f \mathbf{X}^{s+1}$, while the residual equation

is relaxed again to approximate the error better. After a V-cycle, $\mathbf{X}^0$ receives the accumulated error from previous grids and the initial estimate of $\mathbf{U}^{k+1}$ can be corrected as $\mathbf{X}^0 = \mathbf{X}^0 + \mathbf{H}_c^f \mathbf{X}^1 \approx \mathbf{U}^{k+1}$. The V-cycle algorithm is presented in detail on Appendix A.

The restriction and prolongation operators, as well as the coarsening of matrix **G**, needed by the V-cycle, were already defined in the previous section. The remaining operations, including relaxation, are simply sparse matrix operations (see Section 4.2 for an analysis of their complexity). The relaxation method chosen here is Gauss-Seidel (GS) since it is simple and always converges for diagonally dominant matrices [*Saad*, 2003] as it is our case (see Chapter 3). We achieved the best rates of convergence for AMG using an implementation that on the finest grid corresponds to a Symmetric-Red-Black GS [*Saad*, 2003]; while on the other grids we alternate the order of relaxation as we did on the finest grid, but based only on the order assigned by the sorting algorithm.

It remains to define now when to stop coarsening the grid. Since, on each coarsening step, we reduce the grid size to less than half the size of the previous grid (assuming sparsity and independence of the new grid), we decided to stop coarsening the grid, when the number of vertices is equal or less than $\log_2 N$.

Notice that on Equation 3.14 we are estimating only one step of the semi-implicit nonlinear diffusion PDE. The solution of the PDE for a given scale may require repeating the process described earlier several times, i.e. for each scale-step we construct the multigrid structure and run several extra V-cycles. However, thanks to the numerical stability of the semi-implicit scheme and the linear time complexity (see Section 4.2) of AMG, we can use large values of the scale-step μ, which means that few scale-steps would suffice for most

applications and the overall complexity remains scalable algorithmically.

4.1.3 AMG-based Segmentation

We can directly use the AMG structure to segment the image. This approach actually works reasonably well, and it is very flexible, since we use the same parameters to solve the PDE and to segment the image. A better approach, however, consists of solving the PDE and then segment the smoothed image using different (updated) parameters to construct the final multigrid structure. We can create an AMG structure over the smoothed image that stops the coarsening process when all the vertices are segment representatives. The basic AMG structure for the segmentation algorithm is constructed as in the previous section, but we can now use the weights given by Equation 3.6 or

$$\forall i, j \in \mathrm{V}^0 : g_{ij} = e^{-\beta\theta\left(\mathbf{u}_i^0, \mathbf{u}_j^0\right)}, \qquad \textbf{4.10}$$

which is the similarity metric proposed by [*Sharon et al*, 2000], extended to hyperspectral imagery. We can also change the parameter α by a parameter γ on the coarsening Equation 4.8, which adds more flexibility to the algorithm.

The saliency Γ_i of a vertex i is determined as in [*Sharon et al*, 2000], for $s = 0, \ldots, S\text{-}1$ as,

$$\forall i \in \mathrm{V}^s : \Gamma_i = \frac{\sum_{j \in \mathrm{V}^s} g_{ij}^s}{m_i^s}. \qquad \textbf{4.11}$$

Equation 4.11 measures the dependence of a vertex i on its neighboring vertices, at a given grid level, normalized by its mass. Hence, a salient segment would be a vertex with very low dependence on its neighborhood, but also influent on the previous grids. Notice that this

measure of saliency is the same used in normalized cuts [*Shi and Malik*, 2000], but on coarse scales. We define a vertex as a salient segment if its saliency is $\Gamma_i \leq \varepsilon$, where ε is a threshold parameter. The coarsening stops as soon as all the vertices in the grid satisfy the saliency criteria.

Once we had detected the representatives at different grid levels, we must go back to the finest grid to segment the image at the highest resolution. This process is called sharpening in [*Sharon et al*, 2000]; a hard segmentation is obtained from the fuzzy dependences that exist between the vertices at the different levels in the AMG structure. The sharpening algorithm of [*Sharon et al*, 2000] works fine if we start from the coarsest grid, but if we start from lower levels (lower scales); the algorithm may leave large regions of the image un-segmented. The reason is that, as we go down, there are much more vertices un-labeled than representatives, in fact, some vertices cannot be labeled on a coarse scale, since they are on islands, i.e. pockets of vertices, isolated from the remaining vertices. Sharon et al recognized this fact in [*Sharon et al*, 2001], where they proposed another approach that includes boundary tracing.

We use here a simpler approach that already produces good segmentation results. Let us call $r_1, r_2, \ldots, r_R$ the R representatives identified at level s, and let $\mathbf{p}_i = \left(p_i^{r_1}, \cdots, p_i^{r_R}\right)$ be a vector of probabilities indicating that vertex $i \in \mathrm{V}^0$ has probabilities $p_i^{r_1}, \ldots, p_i^{r_R}$ of belonging to the segments represented by $r_1, \ldots, r_R$, respectively. Hence, $\mathbf{p}_{r_k} = \left(0, \cdots, p_{r_k}^{r_k}, \cdots, 0\right)$, with $p_{r_k}^{r_k} = 1$, and $\mathbf{p}_i = (0, \cdots, 0)$ for $i \notin \{r_1, \ldots, r_R\}$ at the start of the sharpening algorithm. Let us also call N_i^s the set of vertices in V^s that are close neighbors to vertex i. The sharpening

algorithm is given by,

For levels s down to 0:

- $\forall i \in V^s$: *if* $\max\{\mathbf{p}_i\}<1$, $\mathbf{p}_i = \dfrac{\mathbf{p}_i + \sum_{j \in N_i^s} w_{ij}^s \mathbf{p}_j}{1 + \sum_{j \in N_i^s} w_{ij}^s}$.

  If $\max\{\mathbf{p}_i\} = p_i^{r_k} \geq 1-\delta$ *then* $\mathbf{p}_i = (0, \cdots, p_i^{r_k} = 1, \cdots, 0)$.
- $\forall i \in V^s$: *if* $\max\{\mathbf{p}_i\}<1$, *perform* υ *Gauss-Seidel relaxations of the form,*

$$\mathbf{p}_i = \frac{\mathbf{p}_i + \sum_{j \in N_i^s} g_{ij}^s \mathbf{p}_j}{1 + \sum_{j \in N_i^s} g_{ij}^s}.$$

  If $\max\{\mathbf{p}_i\} = p_i^{r_k} \geq 1-\delta$ *then* $\mathbf{p}_i = (0, \cdots, p_i^{r_k} = 1, \cdots, 0)$.
- $\forall i \in V^s$: *if* $\max\{\mathbf{p}_i\}<1$, *find the closest representative* r_k *with the largest* $g_{ir_k}^s$ *as defined in Equation 4.8 and make* $\mathbf{p}_i = (0, \cdots, p_i^{r_k} = 1, \cdots, 0)$.

The three steps indicated in the sharpening algorithm attempt to assign a segment representative to each vertex at grid *s*. The first step uses the fact that most vertices from grid *s*-1 must be strongly dependant on vertices from grids *s*, hence, $w_{ir_k}^s$ might be high for some representative r_k. However, vertices that were chosen from *s*-1 to the next grid and are not representatives have $w_{ir_k}^s = 0$ for $k = 1, ..., R$, since both *i* and r_k are in V^s. Nevertheless, their neighbors that are in $V^{s-1} \setminus V^s$ might have been labeled in this step. Hence, the next step is the same as in [*Sharon et al*, 2001], we perform υ Gauss-Seidel relaxations allowing the probabilities of the neighbors to affect the probabilities of each vertex, based now on their similarities. As noted in [*Sharon et al*, 2001], two GS relaxations suffice, since a higher number of relaxations does not produce any changes on vertices located on isolated pockets or on vertices that have nearly the same probability of belonging to two different segments. The third step assigns the vertices that have not been labeled yet to the closest representative.

Also as in [*Sharon et al*, 2001], probabilities higher than a given threshold, 1-δ, are set to one; in order to speedup the sharpening process.

4.2 Implementation details and Complexity

Most of the algorithm's parameters are set by trial and eror, e.g. refer to [*Sharon et al*, 2001; *Galli and De Candia*, 2005; *Galun et al*, 2003; *Akselrod-Ballin et al*, 2006]. In particular, we use $\tau = 0.2$, $\delta = 0.2$, $\varepsilon = 10^{-5}$, the number of GS relaxations for the sharpening algorithm is $\upsilon = 2$, and the number of GS relaxations in AMG is simply $\upsilon_0 = \upsilon_1 = \ldots = \upsilon_S = 1$. Experimentally (Section 4.3), we found that two V-cycles suffice to achieve good accuracy for scale-steps $\mu \leq 5$, which corresponds to 20 times the maximum scale-step that can be used with the explicit scheme to achieve good accuracy (see Section 3.3). The remaining parameters α, β, and γ depend on the image and the application itself, since they define the level of smoothing (α) and the threshold in similarity (β, γ) that is acceptable within a homogenous region. In all our experiments, $0.005 \leq \alpha \leq 0.015$, and $\gamma \leq \beta \leq \alpha$, which indicates that the range of variability is relatively small and can be set according to the scene at hand and the scale needed. Notice here that thanks to the smoothing provided by the nonlinear diffusion PDE, β and γ can be set to lower values allowing greater discrimination between the more homogeneous regions in the smoothed image.

We selected α, β, and γ by trial and error, however, we use a simple technique that can be automated. First, we selected the value of α that produces the best overall accuracy for the smoothed image. The scanning of the best α requires only steps of 0.001 or higher within the range indicated before. Sometimes the best α for smoothing is not the best for

segmenting (see next section), but it is a good starting point. If we are smoothing and segmenting, we fix $\alpha = \beta = \gamma$ and vary α at steps of ±0.001, starting from the best value found for smoothing and stop when classification accuracy stops improving. Usually, the best α for smoothing and segmentation is close to the best α for smoothing only. Once we have selected α, the remaining two parameters (β, γ) are set initially to the same value as α. We fix $\gamma = \beta$ and search for the best overall accuracy reducing β at steps of -0.001, since $\beta \leq \alpha$. Finally, we vary γ at steps of -0.001 until a new maximum in classification accuracy is reached.

The advantage of this approach is that classification accuracy change monotonically in all our experiments. Hence, if we find that for a given parameter, say β, a change in -0.001 reduces classification accuracy, then there is no need to continue reducing that parameter because classification accuracy would continue degrading. However, a better approach that can be addressed in the future is the use of parallel genetic algorithms to search the parameter space.

We introduce some changes in Sharon's algorithm that improves the running time and algorithmic scalability of the algorithm for hyperspectral imagery, and overcomes some limitations of the original algorithm. In particular, we made the following changes,

- Sharon's algorithm [*Sharon et al*, 2000], uses state vectors of the same size as the number of pixels in the image. Since at first there are as many segments as pixels in the image, there is an enormous waste of disk space, mostly filled with zeros. A better approach for large sparse graphs is to use Red-Black trees, [*Cormen, et al*,

2001], to store the neighborhood of each vertex, which also provides fast searches within each neighborhood.

- The time complexity of the segmentation algorithm is linear, but the constant of linearity grows exponentially with the size of the neighborhood [*Brandt*, 2000]. We reduce the neighborhood size by two mechanisms. First, we eliminate vertices with weights lower than 0.1. Second, we limit the number of neighbors to 10, significantly reducing the running time of the algorithm without affecting negatively the accuracy of the AMG solver or the segmentation algorithm.
- In [*Sharon et al*, 2000], the pixels are assigned to each representative one at a time. That is, they perform a top-down sharpening on each representative. This segmentation is time consuming (especially with many segments). We sharpen the image, with all the representatives at the same time, as indicated on the sharpening algorithm (Section 4.1.3). The segmentation results are the same, but with an improvement in running time.
- The vector of probabilities indicated on the sharpening algorithm (Section 4.1.3) is not practical for implementation purposes, since most of its entries are always zero. We use instead variable length vectors that store only the indices of the representatives and their corresponding probability. On any scale, a given vertex may be related to a few representatives, and since it is always labeled on that scale, there is no storage overhead.

All the algorithms presented here were implemented in C++ for Linux, under the Cygwin[9] environment. We used the C++ Geospatial Data Abstraction Library (GDAL[10]), which supports more than 50 raster image formats, including BIL, BSQ and BIP formats, commonly used in hyperspectral imagery, without limit in the size of the image. We used GDAL to read the hyperspectral images and to write the smoothed hyperspectral images on disk. We also used LAPACK[11] to obtain an accurate solution of the PDE at the coarsest level *S*, using LU factorization with pivoting first and then Gaussian elimination. An accurate solution of Equation 3.14 can always be found using Gaussian elimination, since the matrix $\mathbf{I}\text{-}\mu\mathbf{G}$ is diagonally dominant [*Golub and Van Loan*, 2000]. Gaussian elimination has time complexity $O(M\nu_s^3)$, where ν_s is the number of vertices at scale *s* and *M* the number of bands in the image, and it is only used in AMG for the coarsest scale, where the number of vertices is $\nu_s = O(\log N)$. We used Gaussian elimination on the finest grid, for comparison purposes only and independently of the AMG framework proposed here (Section 4.1).

From the previous section, it can be seen that in AMG and the segmentation algorithms only a handful of operations are required for each vertex on Equations 4.1 to 4.11. As indicated before, the sorting algorithm runs in linear time on the number of vertices at each grid. Also, since a fixed number of relaxation sweeps are made at each level, relaxation is linear in the number of vertices, at each grid. The matrix operations indicated on the sharpening algorithm (Section 4.1.3) have also linear time complexity since matrix $\mathbf{I}\text{-}\mu\mathbf{G}^s$ is sparse with at most 10 off-diagonal (neighbors) elements and there are ν_s diagonal elements

[9] http://www.cygwin.com/
[10] http://www.gdal.org/

(vertices) at scale s. Hence, the product $(\mathbf{I}-\mu\mathbf{G}^s)\mathbf{X}^s$ takes $\sim 10M\nu_s = O(M\nu_s)$ time. Also, the exact solution of Equation 3.14, using Gaussian elimination takes $O(M \log^3 \nu_S)$ time which is $O(M\nu_S)$, for sufficiently large ν_S. On the other hand, since $\nu_s \leq \frac{1}{2}\nu_{s+1}$, $0 \leq s < S$, with $\nu_0 = N$, the running time of the complete AMG-segmentation algorithm is linear in the number of pixels and spectral bands,

$$\sum_s \kappa M \nu_s \leq \kappa M \sum_s \left(\frac{1}{2}\right)^s N < 2\kappa NM = O(NM), \qquad \textbf{4.12}$$

where, κ is the number of operations on each vertex.

Let us analyze now the storage requirements for the proposed algorithm. AMG requires storing only two matrices of size $\sim M\nu_s$ at each level that dominate the disk storage requirements: **X** and **F**, with the original image stored in $\mathbf{F}^0$. At each scale, storing **X** and **F** require $2M\nu_s$ disk space. By the same reasoning as before, the disk space required is given by

$$\sum_s 2M\nu_s \leq 2M \sum_s \left(\frac{1}{2}\right)^s N < 4NM.$$

Hence, the disk requirements are less than $4MN$, with additional variables of size $O(\nu_s)$ such as $\mathbf{G}^s$ and other temporal variables that can only account for $O(N)$ overall. For sufficiently large M, as is the case of hyperspectral imagery, the disk requirements are dominated by the $4MN$ term, again linear in the number of pixels and hyperspectral bands.

The segmentation algorithm does not require additional storage. The disk requirements for ADI and AOS are $\sim 2MN$, and PCG methods require $\sim 4MN$ (see Chapter 3). Since AMG is

[11] http://www.netlib.org/lapack/

scalable and can have accuracy greater than traditional relaxation methods and the approximated solutions provided by AOS and ADI schemes, we have achieved significant improvement with respect to previous work in terms of scalability, while keeping storage requirements equivalent to PCG methods.

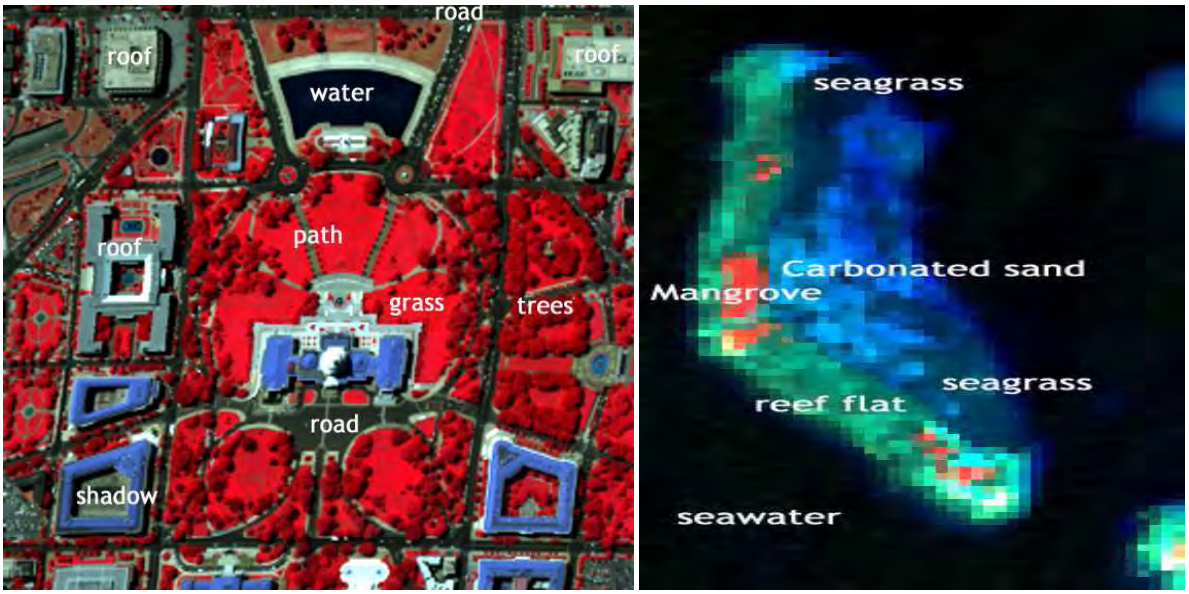
4.3 Experiments

We use four hyperspectral images in our experiments, representing different landscapes,

- Indian Pines image (Figure 3.2.a), described in Section 3.2, with ground truth shown on Figure 3.2.b.
- Cuprite image (Figure 3.4.a), described on Section 3.2, with ground truth shown on Figure 3.5.
- A high spatial-spectral resolution image of the Washington DC Mall area taken by the Hyperspectral Digital Imagery Collection Experiment (HYDICE) sensor on August 23, 1995[12]. This image contains 1280×307 pixels and 224 bands. Several bands are eliminated because they correspond to atmospheric absorption bands or they are too noisy, leaving 191 bands in the 400-2480 nm range. We choose a sub-image of 282×307 pixels and 191 bands (Figure 4.5.a) as representative in our experiments. There is no need for ground truth on this image, given its high spatial resolution (3m) that allows identifying the different objects in the image by simple visual inspection. We use the same classes indicated by previous studies of this image, [*Ball and Bruce*, 2005; *Landgrebe*, 2002] directly marked on Figure 4.5.a.

[12] Available on the accompanying CD of [*Landgrebe*, 2003]

- The Enrique Reef image (Figure 4.5.b), which corresponds to a small part of the AVIRIS image taken over the south-west coast of Puerto Rico in 2006. We use this image because the Enrique Reef environment is a well-known area of study for the marine science department at the UPRM and is part of the seaBED test bed at CenSSIS (*Goodman et al*, 2006). Hence, we used their expertise to identify training and testing samples on the image. The ground truth of this image is directly marked on Figure 4.5.b. We eliminated noisy spectral bands, so that our Enrique reef image consists of 46×90 pixels and 146 bands in the 414-2310 nm range.



(a) **(b)**

Figure 4.5 RGB composite of a) Washington DC (bands 63, 52, and 36) and b) Enrique Reef images (bands 50, 27, and 17), showing also their ground truth.

Classification accuracy of the NW Indian Pines and the Washington DC Mall area have been analyzed recently using a Bayesian MRF approach [*Neher and Srivastava*, 2005]. Also the Cuprite image has been used recently [*Bachmann et al*, 2005] to study the feasibility of dimension reduction using manifold coordinates. However, we would not compare here

our results with previous classifications of these images, since we are using our own set of training and testing samples chosen specifically to test the performance of our segmentation algorithm. Comparisons between different classification methods should always be made using the same images, set of training and testing samples and spectral bands. Since, supervised classification accuracy may vary strongly from one set of training samples to another and also according to the number of bands used.

In Section 4.3.1, we will test the performance of AMG as a solver of Equation 3.14 using a large scale step, $\mu = 5$. This scale-step is typically the largest value for μ, such that the solution obtained does not fall away from the more accurate solution that would be found using a much smaller scale step (see Chapter 3). In Section 4.3.2, we will test the performance of the AMG-based segmentation algorithm, in terms of classification accuracy.

4.3.1 Performance of AMG as a solver

We first compute the sum of square errors between the computed solution of Equation 3.14 using AMG and the solution of Equation 3.14 obtained using LAPACK at the finest grid, using $\alpha = 0.015$ as the threshold in the diffusion coefficient (Equation 3.6), which is the largest value of α we had used in these and previous experiments (Chapter 3). We test AMG using local measures only, i.e. Equation 4.7, and incorporate the mean spectrum, i.e. Equation 4.8, where θ can be either the Euclidean distance or the spectral angle.

Given that Gaussian elimination for banded matrices, as is the case of **G** on the finest grid, requires to store vectors of size $O(N^{3/2})$, it easily overcomes the memory available for large images. In particular, we could not obtain the solution of 3.14 using LAPACK on the finest grid, for the Cuprite image, using a PC with 2Gb of RAM memory. Hence, we use

instead the first 300×300 vector-valued pixels (50 bands) of the Cuprite image, which we call here, small Cuprite.

Figure 4.6 shows, in semi-logarithmic scale the sum of square errors as a function of the number of V-cycles computed using Equation 4.8, where θ is the spectral angle. The error reduces at a rate of $r = 10^{-d}$, where d is the slope of the line shown on Figure 4.6. From this figure, the rate of convergence is in the range r = 0.013-0.032. These results are quite good, since they compare well with the reported convergence rates for well-tuned AMG algorithms (r~0.05) [*Briggs et al*, 2000; *Kimmel and Yavneh*, 2003]. The rates of convergence for $\alpha < 0.015$ could be even better, since the matrix $\mathbf{I}$-$\mu\mathbf{G}$ tends to the identity as α decreases. The rate of convergence indicates that the error is reduced by a factor r on each V-cycle, so that if r = 0.05, the error is 5% of its initial value on the first V-cycle and 0.025% on the next V-cycle. Experimentally, we found that 2-V-cycles are sufficient to provide accuracies superior to the ones obtained using PCG schemes with a tolerance of 10^{-3}. This is the same tolerance used in Chapter 3 with very good results in terms of the accuracy of solution of the geometric PDE and of the classification.

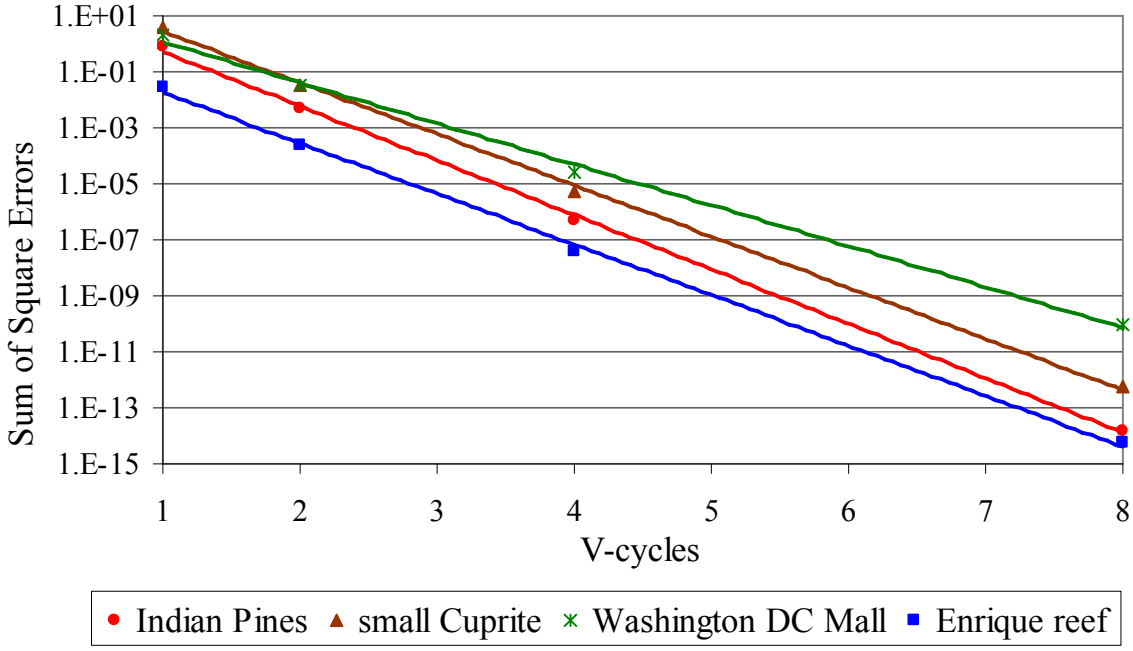


Figure 4.6 Performance of AMG vs. the number of V-cycles, in terms of the square error.

Table 4.1 AMG Rates of Convergence.

Equation	Indian Pines	small Cuprite	Washington DC	Enrique Reef
(12)	0.032	0.051	0.079	0.051
(13)-ED	0.016	0.020	0.050	0.032
(13)-SA	0.013	0.016	0.032	0.016

Table 4.1 compares the rates of convergence of AMG using only accumulated local measures, i.e. Equation 4.7, and the rates of convergence of AMG using local measures and Euclidean distance (ED) or spectral angle (SA) between mean spectrums, i.e. Equation 4.8. It can be noticed that by introducing simple global measures, such as the mean spectral intensity, the error convergence rate is at least two times faster than using accumulated local measures only. It can be also noticed that AMG converges faster using the spectral angle than using Euclidean distances, tough the comparison may be not completely fair, since defining θ as the spectral angle means changing to a PDE, which is not longer Equation 3.3.

Notice also that the slowest rate of convergence corresponds to the Washington DC image, followed by the Enrique reef image. This is due to the high number of objects with strong vectorial boundaries in these images. This implies that $g(\theta)$ varies strongly on a large region within the image, and the simple non-uniform sampling used here is less effective to translate the problem to the coarser grids. Nevertheless, the rates of convergence of the AMG method for these images are still quite good, and there is no need for using more accurate, but computationally expensive, non-uniform sampling methods such as those indicated in [*Brandt*, 2000].

Figure 4.7 compares the sum of square errors with respect to the solution obtained with LAPACK on the finest grid for four methods: ADI and AOS schemes, the Conjugated Gradient method, preconditioned with incomplete Cholesky factorization (PCG-Cholesky, see Chapter 3), and AMG using two V-cycles. It can be noticed from this figure that the sum of squared errors with the proposed AMG is always lower than the error of the other solvers. In particular, the error in AMG is three to four orders of magnitude lower than in ADI and AOS schemes, and even lower than PCG with a tolerance of 10^{-3}, which is the tolerance used in Chapter 3.

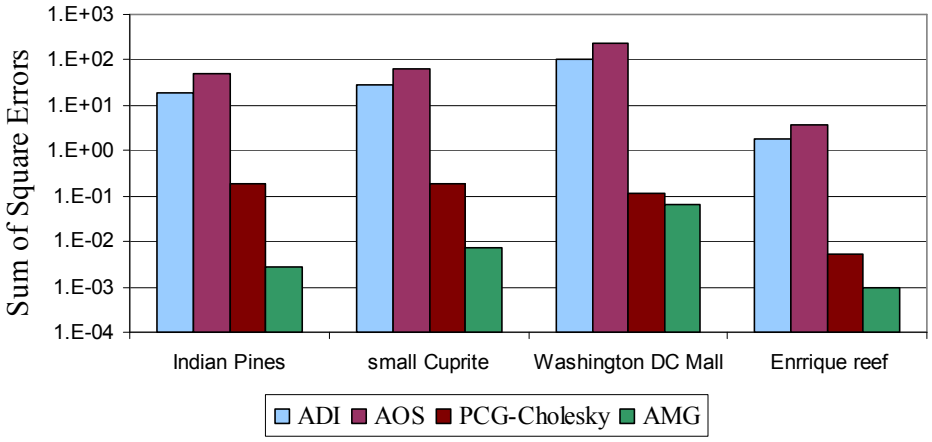


Figure 4.7 Performance of AMG vs. other solvers.

In order to test the performance of AMG in terms of CPU time versus the size of the image (scalability), we selected four sub-images of size 50×50, 100×100, 200×200, and 282×307 pixels from the Washington DC image, with all its 191 bands.

Figure 4.8 shows the CPU time of AMG using $\mu = 5$, $\alpha = 0.015$ and 2 V-cycles, vs. the size of the image, relative to 50×50 image. Figure 4.8 also shows, for comparison purposes, the CPU time required to solve Equation 3.14 for ADI, PCG-Cholesky, and Gaussian elimination. From this figure, we can see that our implementation of AMG is eight times slower than ADI, but AMG is significantly more accurate than ADI and it also naturally enables the segmentation of the image. Further reductions in the running time of AMG can be obtained by using single Red-Black GS relaxation sweeps, instead of the symmetric Red-Black relaxation used here, at the expense of decreasing the convergence rate by a factor of 2 (which is still good, see [*Kimmel and Yavneh*, 2003]). However, we prefer here to trade speed for accuracy of the computed solution for the nonlinear diffusion PDE, since this also may affect classification accuracy, maintaining nevertheless a reasonable computational cost.

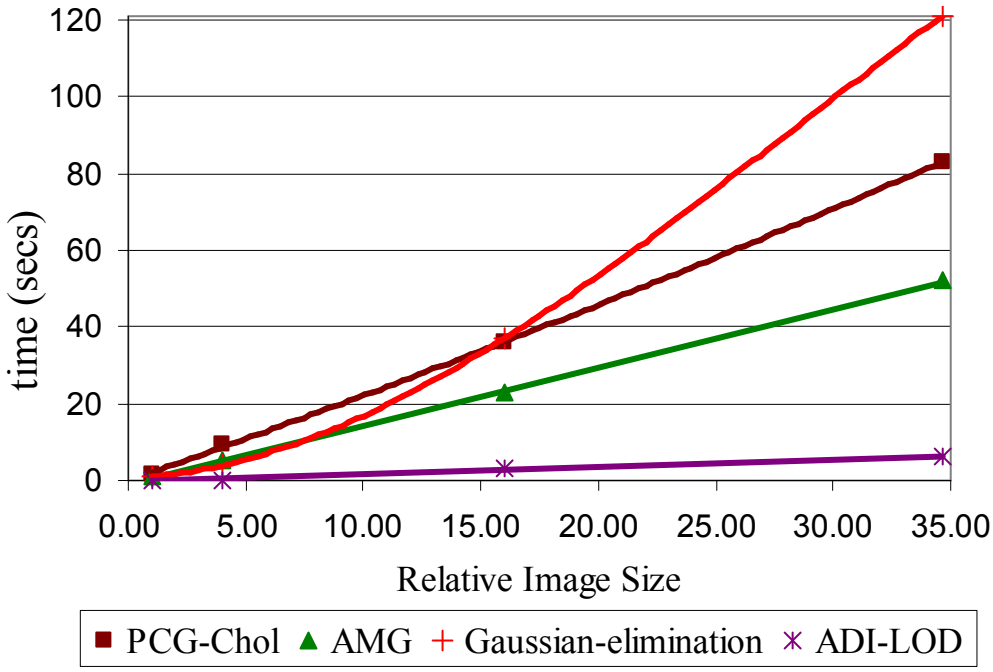


Figure 4.8 CPU time vs. the size of the image.

Figure 4.9 shows the CPU time as a function of the image size for AMG as a solver of Equation 3.14 and to solve both Equation 3.14 and segment the hyperspectral imagery. From this figure, it is clear that solving Equation 3.14 with AMG and segmenting the images has linear time complexity, and the segmentation step takes approximately a 25% of the total smoothing and segmentation time.

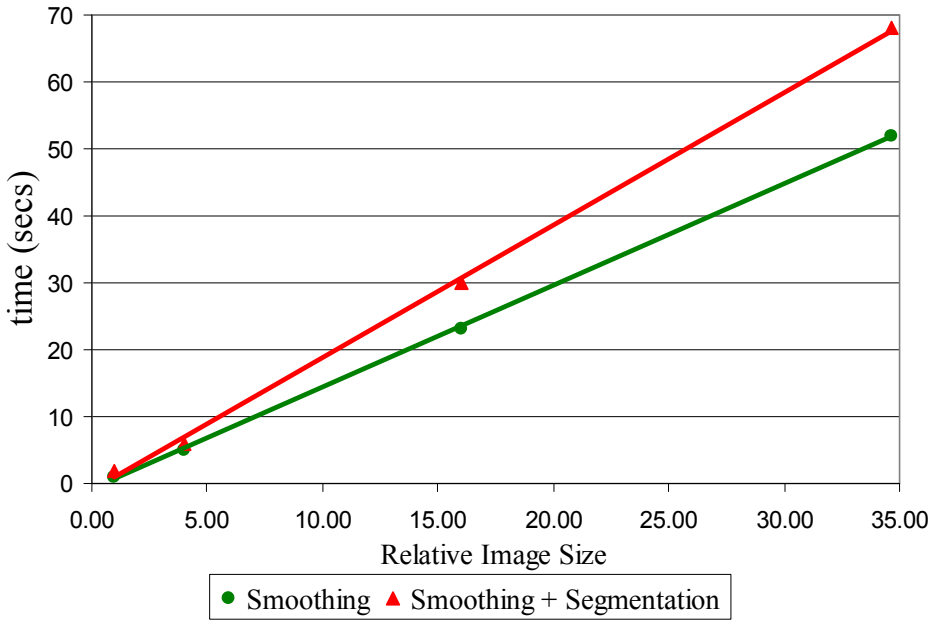


Figure 4.9 CPU time for AMG smoothing and segmentation.

4.3.2 Performance of the AMG-based segmentation

We now evaluate the quality of the segmentation algorithm for classification accuracy. It is clear that over-segmentation affects the accuracy of classification algorithms, since splitting arbitrarily a homogeneous region will produce sub-regions with significantly different statistical characteristics [*Ketting and Landgrebe*, 1976]. On the other hand, under-segmentation might be even worse, since portions of objects belonging to different classes may be passed to the classification algorithm as single objects, precluding the possibility of classifying them correctly. Hence, classification accuracy provides a measure of segmentation quality that corresponds well with the requirements of a good segmentation, and also permits to use real hyperspectral images with ground truth, instead of synthetic test images as often required by current methods that measure the quality of segmented images [*Zhang*, 2001].

We use the segmentation map to produce a piecewise segmented hyperspectral image, where each segment has the spectral signature corresponding to the mean spectrum in the segmented region. We select training (blue polygons) and testing samples (white polygons) on each one of the four hyperspectral images considered here, and shown from Figures 4.10 to 4.12. Notice that the training and testing samples for the NW Indian Pines are different from those used in Chapter 3.

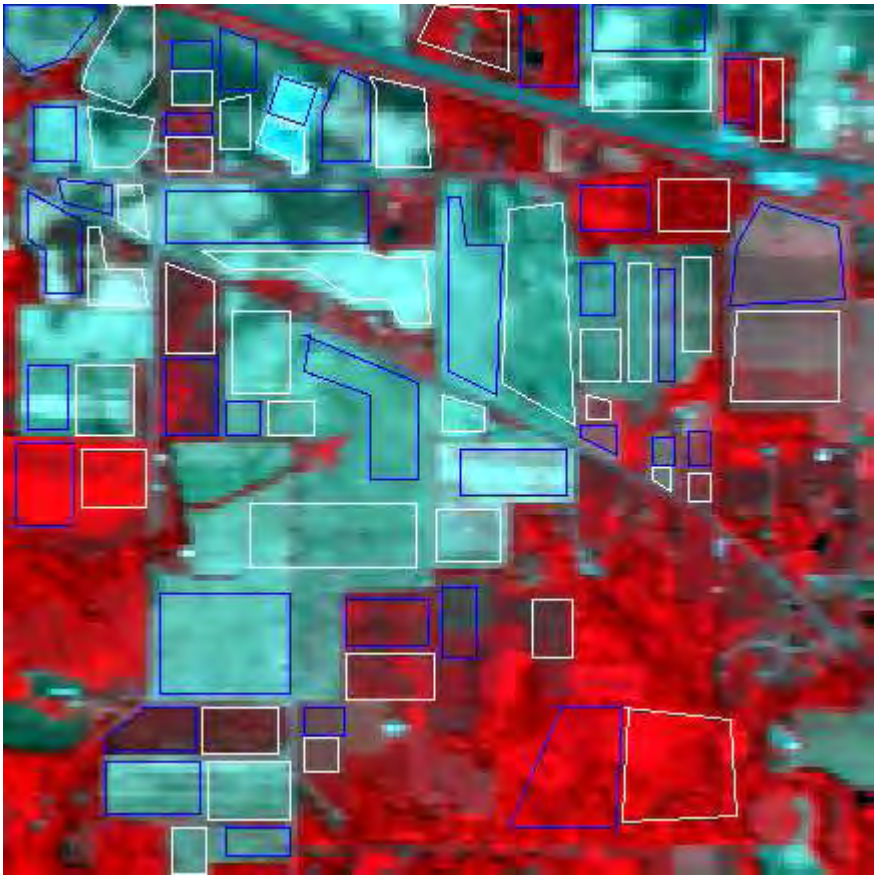


Figure 4.10 Training (blue rectangles) and testing samples (white rectangles) on the NW Indian Pines image (RGB shown corresponds to bands 47, 24, and 14).

The reason is that we are interested now in testing segmentation accuracy trough classification, so larger testing areas are selected in order to penalize over-segmentation. Nevertheless, we use the same Training and Testing samples for the Cuprite image (see

Figure 3.24), since it is difficult to obtain larger areas here without biasing the overall accuracy towards the most abundant minerals.

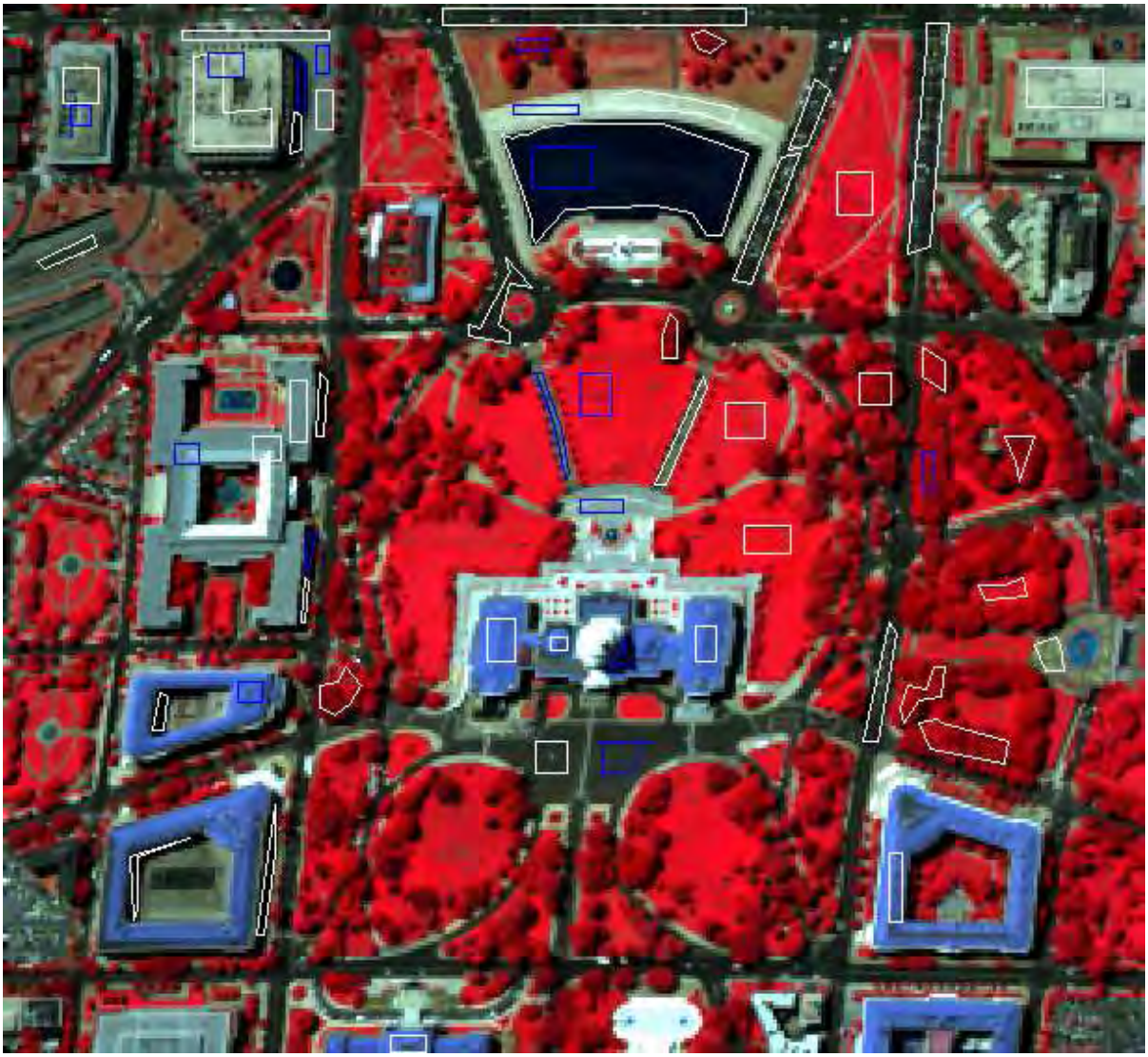


Figure 4.11 Training (blue polygons) and testing samples (white polygons) on the Washington DC mall image (bands 63, 52, and 36).

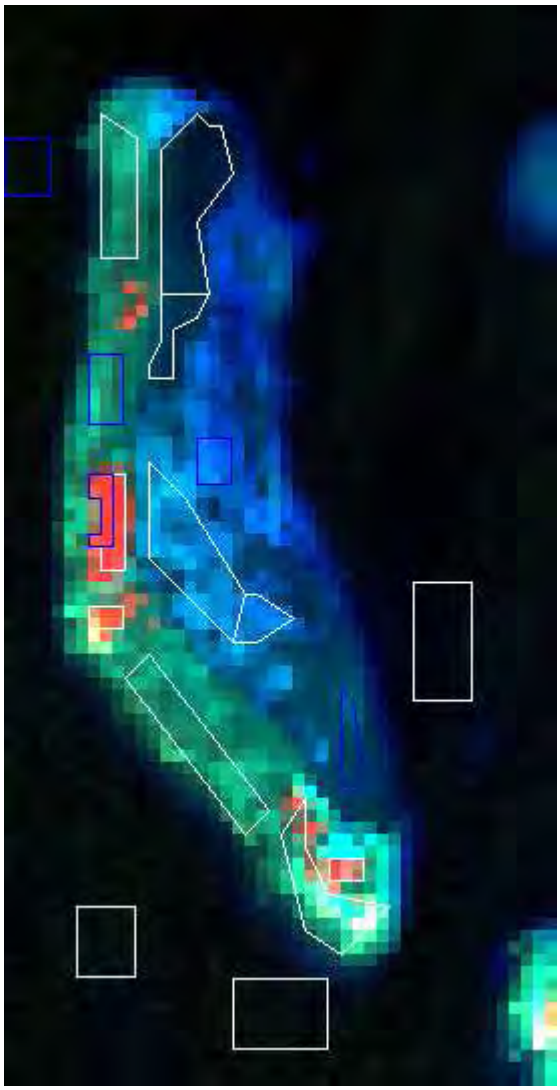


Figure 4.12 Training (blue polygons) and testing samples (white polygons) on the Enrique reef Image (bands 50, 27, and 17).

We choose ECHO, [*Kettig and Landgrebe*, 1976], spectral-spatial as our classifier, provided by MultiSpec. We cannot use simpler classifiers such as Euclidean Distance or Spectral Angle Mapper (SAM), since they do not take into account the spatial domain, which is critical to evaluate the quality of segmentation. Also, we cannot use Maximum Likelihood or other second order statistical classifiers, since they cannot compute accurate covariance matrices with few pixels. Hence, we use ECHO, with a small window of 2×2 pixels that uses

Fisher Linear Discriminant. ECHO clusters the segments into different classes, according to their distance (in terms of Fisher) and according to the homogeneity of the neighborhood, computing likelihoods, whenever possible.

Table 4.2 Overall accuracies in terms of the Kappa statistic.

Classification Accuracy	NW Indian Pines		Cuprite		Washington DC Mall		Enrique Reef	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Original	89.7	68.5	98.7	92.8	100	83.1	100	91.6
Smoothed	99.9	79.2	99.8	96.5	100	84.9	100	94.2
Segmented	97.1	83.0	99.8	95.3	100	83.2	100	95.8
Smoothed and Segmented ED	94.5	87.5	99.7	96.8	100	89.2	100	97.4
Smoothed and Segmented SAM	98.4	***89.0***	99.6	***97.2***	100	***90.5***	100	***98.1***

Table 4.2 shows the best classification accuracies in terms of the Kappa statistic obtained by smoothing with AMG, segmenting with the AMG-based segmentation algorithm, and combining smoothing and segmentation using AMG with Euclidean Distance (ED) or SAM. The Kappa statistics accounts for both the percentage of user's accuracy and the percentage of producer's accuracy (see Section 3.3.3) in a balanced way [*Landgrebe*, 2003]. The Kappa statistic measures the level of agreement between user's and producer's accuracies taking into account that both accuracies may agree simply by chance [*Viera and Garret*, 2005]. A Kappa value of 100% indicates perfect agreement between user's and producer's accuracies, while a Kappa value of 0% indicates that the agreement between user's and producer's accuracies is due only to chance. Hence, obtaining an improvement in the Kappa value indicates that the improvement in classification accuracy was not due to chance.

We could not obtain a classification of the original and smoothed Enrique reef image using ECHO. Hence, the accuracy reported in Table 4.2 for these two images corresponds to the highest classification accuracy, which was obtained using the Spectral Angle Mapper

(SAM), considering all bands. As can be seen from this table, just by smoothing the image, we achieve an improvement in the classification accuracy. Also, segmenting the image usually achieves even better classification accuracies than just smoothing, but not in all the cases. If both smoothing and segmentation are combined, better classification accuracies are obtained than using the smoothing or the segmentation processes alone. The main reason is that nonlinear diffusion reduces the intra-region variability, while keeping the object's boundaries, which improves global separability, while maintaining local information (boundaries) almost intact.

Table 4.3 to Table 4.8 provide detailed information on the percentages of user's accuracy (UA), producer's accuracy (PA), and the number of samples (ns) used for each class and method tested here. At the end of each table, we summarize the total number of samples, the average user's and producer's accuracies, weighted by the corresponding number of samples on each class [*Landgrebe*, 2003], and the Kappa statistic. Smoothing and segmentation using the Euclidean distance is abbreviated in these tables as S&S ED, while the smoothing and segmentation using SAM is abbreviated as S&S SAM. Note that we are not reporting here the accuracies for the training samples of the Washington DC Mall and Enrique Reef images, since as can be seen on Table 4.2, they were always 100%.

Table 4.3 Classification training samples, NW Indian Pines image.

Class	ns	Original		Smoothed		Segmented		S&S ED		S&S SAM	
		UA	PA	UA	PA	UA	PA	UA	PA	UA	PA
Corn-min	397	81.9	71.8	99.7	99.7	99.5	99.0	100.0	90.9	100.0	100.0
Alfalfa	24	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Corn-notill	603	91.7	90.0	100.0	100.0	100.0	94.0	95.3	98.3	100.0	100.0
Corn	105	71.3	97.1	100.0	100.0	96.3	98.1	79.5	100.0	100.0	100.0
Grass/Pasture	199	100.0	96.0	100.0	100.0	100.0	87.9	100.0	91.0	100.0	89.9
Grass/Trees	301	96.2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Grass/pasture-mowed	20	100.0	100.0	100.0	100.0	45.5	100.0	52.6	100.0	51.3	100.0
Hay-windrowed	251	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Soybeans-notill	383	85.8	94.8	99.7	99.2	100.0	90.9	98.0	90.9	99.7	90.9
Soybeans-min	724	90.2	85.6	100.0	99.9	95.3	100.0	99.6	100.0	100.0	100.0
Soybean-clean	149	89.2	100.0	98.0	100.0	81.0	100.0	62.4	38.9	81.0	100.0
Wheat	82	98.8	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Woods	427	99.7	88.8	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Bldg-Grass-Tree-Drives	172	78.6	98.3	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Stone-steel towers	40	100.0	100.0	100.0	100.0	100.0	100.0	36.0	100.0	100.0	100.0
Overall	**3877**	**91.1**	**90.8**	**99.9**	**99.9**	**98.0**	**97.4**	**96.1**	**95.1**	**99.0**	**98.6**
Kappa statistic		**89.7**		**99.9**		**97.1**		**94.5**		**98.4**	

Table 4.4 Classification testing samples, NW Indian Pines image.

Class	ns	Original		Smoothed		Segmented		S&S ED		S&S SAM	
		UA	PA	UA	PA	UA	PA	UA	PA	UA	PA
Corn-min	436	66.3	59.6	83.0	72.7	96.3	78.0	82.7	86.5	100.0	100.0
Alfalfa	12	100.0	58.3	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Corn-notill	572	69.8	72.2	69.6	79.7	82.9	68.5	93.0	85.5	100.0	100.0
Corn	79	47.0	49.4	89.7	65.8	100.0	17.7	61.5	20.3	100.0	100.0
Grass/Pasture	166	59.5	92.8	64.1	100.0	80.7	88.0	85.9	98.8	100.0	89.9
Grass/Trees	295	84.3	96.3	89.1	100.0	100.0	96.3	100.0	96.3	100.0	100.0
Grass/pasture-mowed	10	100.0	100.0	100.0	100.0	8.7	100.0	83.3	100.0	51.3	100.0
Hay-windrowed	255	97.3	69.4	100.0	67.1	100.0	66.7	100.0	100.0	100.0	100.0
Soybeans-notill	436	54.7	90.1	68.3	98.2	74.0	87.6	74.3	89.7	99.7	90.9
Soybeans-min	740	74.2	47.7	90.3	63.9	84.8	88.6	100.0	89.2	100.0	100.0
Soybean-clean	160	90.4	94.4	83.4	97.5	78.5	98.1	60.4	50.6	81.0	100.0
Wheat	104	97.2	100.0	99.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Woods	478	97.7	72.6	95.3	92.9	97.8	100.0	98.7	97.5	100.0	100.0
Bldg-Grass-Tree-Drives	159	45.7	62.9	75.9	55.3	72.9	100.0	89.7	98.7	100.0	100.0
Stone-steel towers	42	100.0	92.9	100.0	97.6	100.0	97.6	34.2	90.5	100.0	100.0
Overall	**3944**	**75.0**	**71.8**	**83.6**	**81.5**	**88.1**	**84.8**	**89.9**	**88.9**	**93.4**	**90.2**
Kappa statistic		**68.5**		**79.2**		**83.0**		**87.5**		**89.0**	

Table 4.5 Classification training samples, Cuprite image.

Class	ns	Original		Smoothed		Segmented		S&S ED		S&S SAM	
		UA	PA	UA	PA	UA	PA	UA	PA	UA	PA
Calcite	821	98.6	100.0	99.5	99.8	99.5	100.0	100.0	100.0	100.0	99.3
High-Al-Muscovite	807	99.1	99.6	100.0	100.0	99.6	100.0	99.3	98.6	100.0	99.5
Kaolinite+Semectite-Muscovite	363	91.4	100.0	100.0	100.0	99.7	100.0	97.1	100.0	97.6	100.0
K-Alumnite	398	99.5	99.5	100.0	99.7	100.0	99.2	100.0	100.0	100.0	98.5
Kaolinite	294	99.7	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Alunite+Kaolinite-Muscovite	406	99.0	94.3	99.8	100.0	100.0	100.0	100.0	100.0	99.3	100.0
Calcite+Kaolinite	754	99.1	100.0	99.7	100.0	100.0	100.0	100.0	100.0	99.5	100.0
Chalcedony	362	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Na-Montmorillonite	450	100.0	99.1	99.6	100.0	100.0	100.0	100.0	100.0	99.8	100.0
Chlorite+Muscovite-Montmorillonite	449	100.0	99.8	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Med-Al-Muscovite	446	100.0	92.4	100.0	98.7	100.0	98.9	100.0	98.7	98.9	98.7
Overall	**5550**	**98.9**	**98.8**	**99.8**	**99.8**	**99.8**	**99.9**	**99.7**	**99.7**	**99.6**	**99.6**
Kappa statistic		**98.7**		**99.8**		**99.8**		**99.7**		**99.6**	

Table 4.6 Classification testing samples, Cuprite image.

Class	ns	Original		Smoothed		Segmented		S&S ED		S&S SAM	
		UA	PA	UA	PA	UA	PA	UA	PA	UA	PA
Calcite	859	99.9	97.0	99.6	97.1	99.8	99.4	99.8	97.2	100.0	99.8
High-Al-Muscovite	479	72.1	99.8	94.9	100.0	96.2	100.0	94.1	100.0	95.6	99.6
Kaolinite+Semectite-Muscovite	376	98.9	100.0	99.5	100.0	99.7	100.0	93.8	100.0	99.5	100.0
K-Alumnite	230	84.0	97.8	83.2	97.0	66.5	92.2	81.3	96.1	77.4	95.2
Kaolinite	325	90.5	100.0	90.5	100.0	92.9	100.0	91.5	100.0	94.2	100.0
Alunite+Kaolinite-Muscovite	521	99.0	91.6	98.6	91.4	96.1	79.5	100.0	90.2	97.4	87.7
Calcite+Kaolinite	473	98.7	99.8	99.6	100.0	96.5	100.0	100.0	100.0	100.0	100.0
Chalcedony	480	99.6	92.9	98.9	92.9	97.2	94.8	98.5	93.8	100.0	95.6
Na-Montmorillonite	417	99.8	99.5	100.0	98.3	100.0	96.9	100.0	98.3	100.0	100.0
Chlorite+Muscovite-Montmorillonite	184	98.8	89.1	100.0	100.0	100.0	89.7	100.0	84.2	100.0	100.0
Med-Al-Muscovite	404	90.6	56.9	94.2	92.8	100.0	96.5	98.0	99.3	100.0	95.0
Overall	**4748**	**94.5**	**93.5**	**97.1**	**96.9**	**96.4**	**95.8**	**97.1**	**96.8**	**97.7**	**97.5**
Kappa statistic		**92.8**		**96.5**		**95.3**		**96.8**		**97.2**	

Table 4.7 Classification testing samples, Washington DC Mall image.

Class	ns	Original		Smoothed		Segmented		S&S ED		S&S SAM	
		UA	PA	UA	PA	UA	PA	UA	PA	UA	PA
Water	1386	99.9	99.1	98.5	100.0	100.0	99.9	100.0	100.0	100.0	100.0
Grass	385	96.9	98.7	100.0	100.0	99.7	100.0	99.7	100.0	99.2	100.0
Trees	552	98.2	97.5	96.0	99.5	99.2	92.6	99.8	99.5	98.7	98.6
Rooftop	814	71.3	67.1	73.0	50.2	70.8	85.6	82.6	68.3	97.2	68.7
Road	1249	92.2	82.4	94.9	94.3	98.9	67.8	98.9	92.2	98.8	91.9
Paths	432	59.0	62.0	47.9	69.7	81.7	72.2	89.0	84.5	56.6	98.6
Shadow	171	49.9	98.8	87.2	100.0	34.6	97.1	37.2	91.2	88.5	90.1
Overall	**4989**	**87.6**	**86.3**	**88.5**	**87.8**	**91.0**	**86.2**	**93.7**	**91.2**	**94.9**	**92.2**
Kappa statistic		**83.1**		**84.9**		**83.2**		**89.2**		**90.5**	

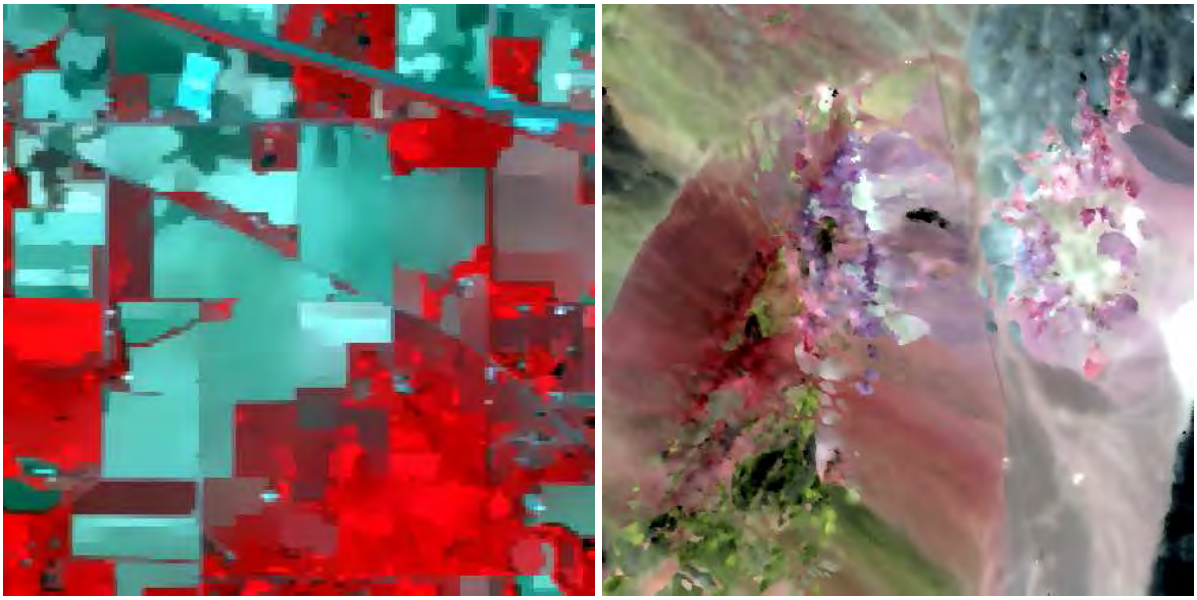
Table 4.8 Classification testing samples, Enrique Reef image.

Class	ns	Original		Smoothed		Segmented		S&S ED		S&S SAM	
		UA	PA	UA	PA	UA	PA	UA	PA	UA	PA
Mangrove	16	62.5	93.8	62.5	93.8	100.0	62.5	100.0	100.0	78.9	93.8
Water	128	98.5	100.0	99.2	100.0	100.0	100.0	97.7	100.0	100.0	100.0
Seagrass	80	84.8	97.5	92.0	100.0	98.8	100.0	100.0	100.0	98.8	100.0
Carbonate Sand	70	100.0	82.9	100.0	91.4	100.0	90.0	100.0	92.9	100.0	100.0
Reef Flat	117	99.1	90.6	99.1	90.6	90.7	100.0	95.8	97.4	99.1	95.7
Overall	**411**	**94.9**	**93.7**	**96.5**	**95.6**	**97.1**	**96.8**	**98.1**	**98.1**	**98.7**	**98.5**
Kappa statistic		**91.6**		**94.2**		**95.8**		**97.4**		**98.1**	

As explained before, the selection of parameters was made here in such a way that the overall classification accuracy (represented by the Kappa statistic) increase, as much as possible using smoothing, segmentation or both smoothing and segmentation. Hence, as can be appreciated from the previous tables, some classes decrease their accuracy in favor of other classes that have a larger number of pixels; see for instance the grass/pasture-mowed class in Table 4.4. Nevertheless, as can be seen from Table 4.3 to Table 4.8, most of the classes always benefit from smoothing, segmentation or both processes. However, if we were interested in improving only the accuracy of a particular class (which is the case of target detection), we must select α, β, and γ towards that objective.

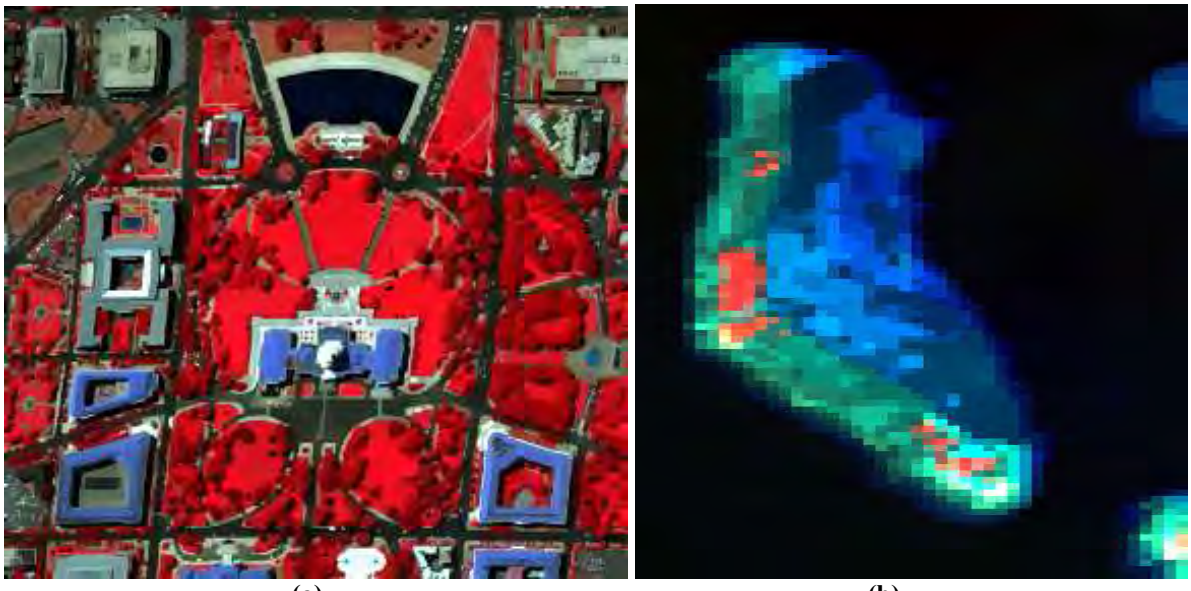
We must emphasize here that we are classifying the piece-wise spectrally-constant hyperspectral images obtained from the segmentation map, with the sole purpose of testing the quality of the segmentation. A more accurate classification of the image would take into account the segmentation maps to extract information from the smoothed image. Future work on this area should also consider the introduction of spectral-spatial similarity metrics or texture and unsupervised classification of the homogeneous regions segmented. Finally, we must also emphasize that the scale space is not only a vehicle to achieve better segmentation results, but also provides smoother images that can provide better results for

other hyperspectral image processing algorithms such as classification, registration, and impainting, in conjunction with the segmentation maps.



(a) (b)

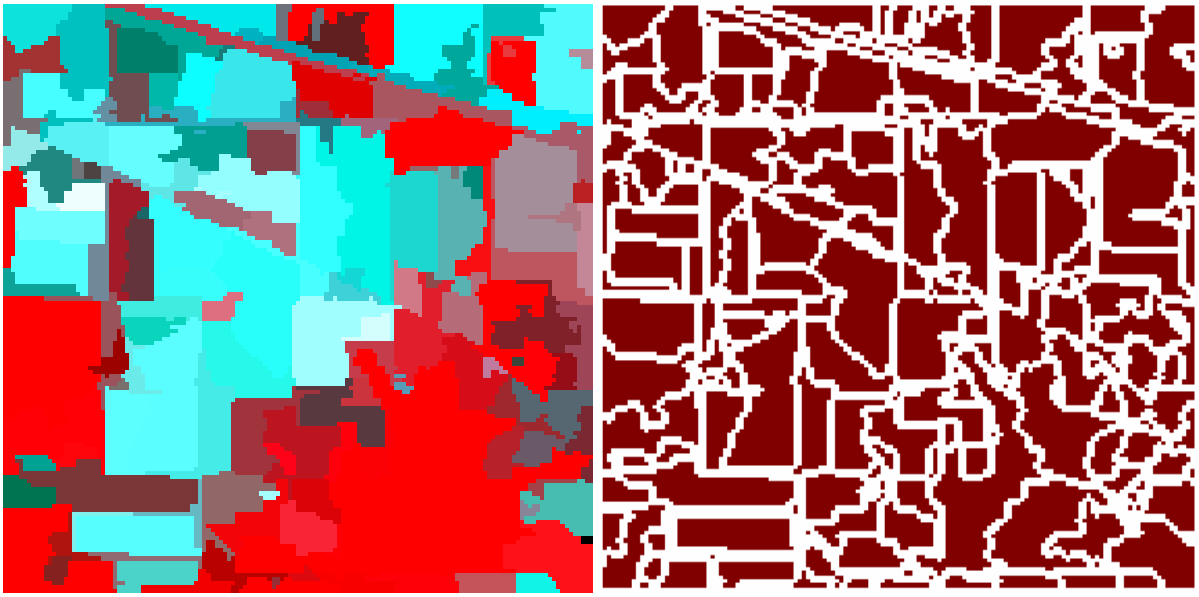
Figure 4.13 Smoothed images with AMG: a) NW Indian Pines (bands 47, 24, and 14) and b) Cuprite (bands 183, 193, and 207).



(a) (b)

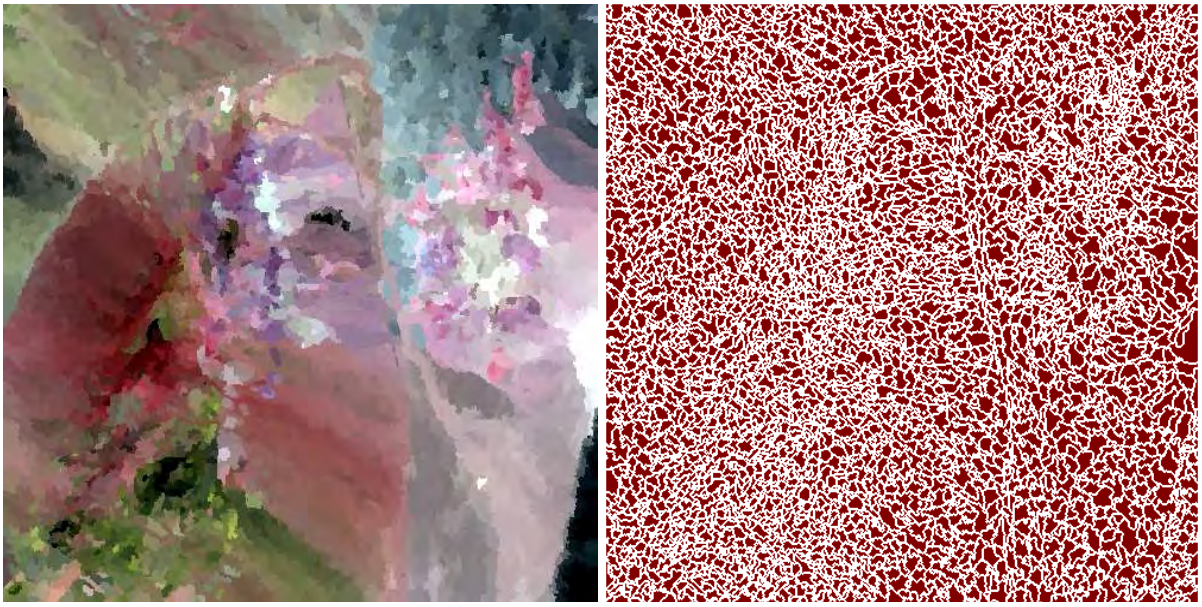
Figure 4.14 AMG Smoothing a) Washington DC (bands 63, 52, and 36) and b) Enrique reef (bands 50, 27, and 17).

Figures 4.13 and 4.14 show RGB composites of the smoothed hyperspectral images that produced the best classification accuracies, as indicated in Table 4.2.



(a) (b)

Figure 4.15 a) Segmented NW Indian Pines image (bands 47, 24, and 14), b) segment boundaries.

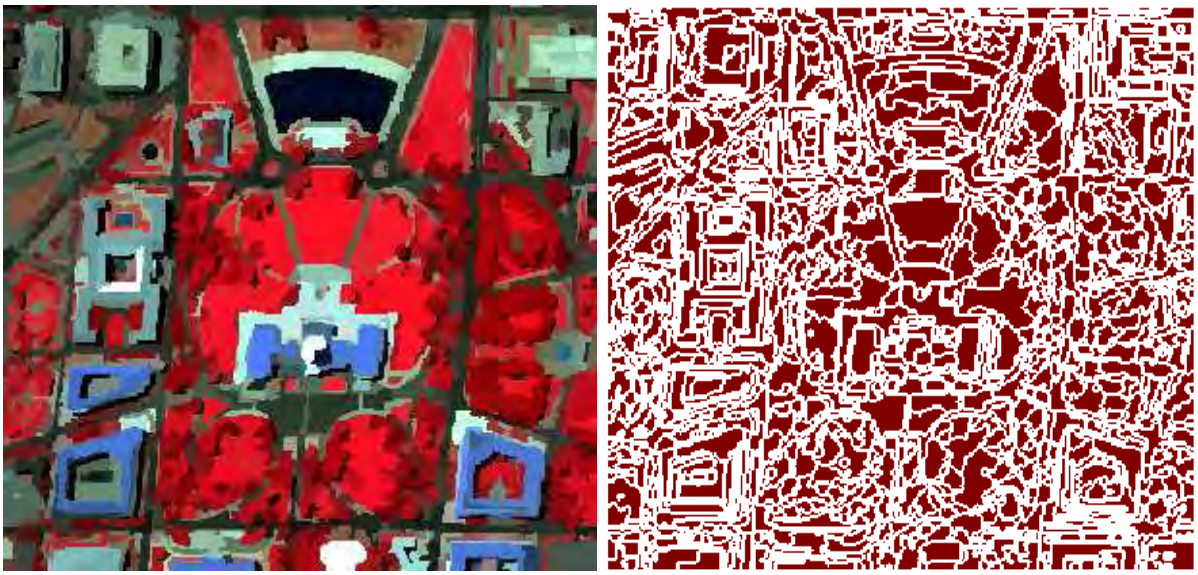


(a) (b)

Figure 4.16 a) Segmented Cuprite image (bands 183, 193, and 207), b) segment boundaries.

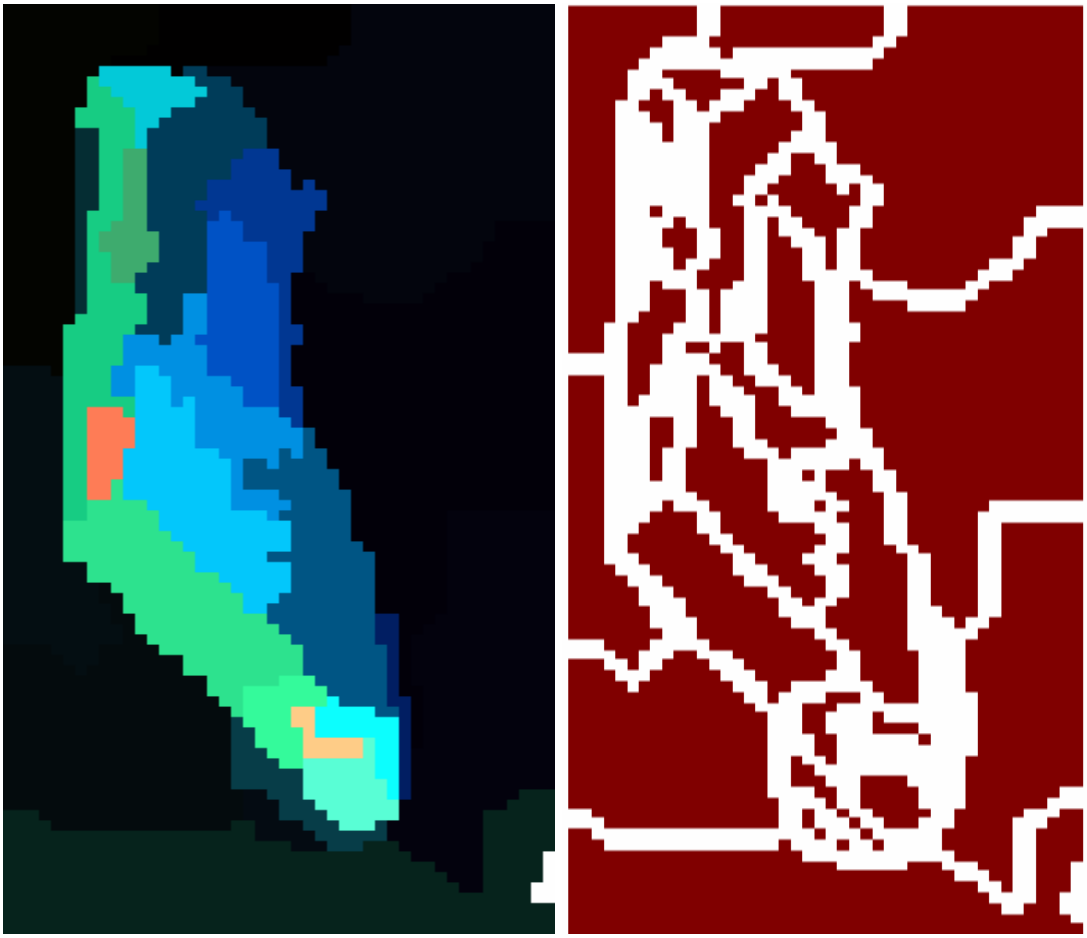
Figures 4.15 to 4.18 show RGB composites of the segmented hyperspectral images that resulted on the best classification accuracies, indicated in Table 4.2, and the corresponding segment boundaries. It should be noticed here that with the exception of the Indian Pines image, the segmented images shown from Figures 4.15 to 4.18 were obtained using smoothed images with a different α value than those in Figures 4.13 and 4.14. The value of α that produces the best classification results using only nonlinear diffusion is not necessarily the best parameter for obtaining the best accuracies using both smoothing and then segmentation.

Notice also that all the images are over-segmented, especially the Cuprite image. As stated before, sub-segmentation is much worse than over-segmentation, since a sub-segmented image will merge two or more segments belonging to two or more classes. Hence, it should be expected that the best classification accuracies corresponds to over-segmented images.



(a) **(b)**

Figure 4.17 a) Segmented Washington DC image (bands 63, 52, and 36), b) segment boundaries.



(a) (b)

Figure 4.18 a) Segmented Enrique Reef image (bands 50, 27, and 17), b) segment boundaries.

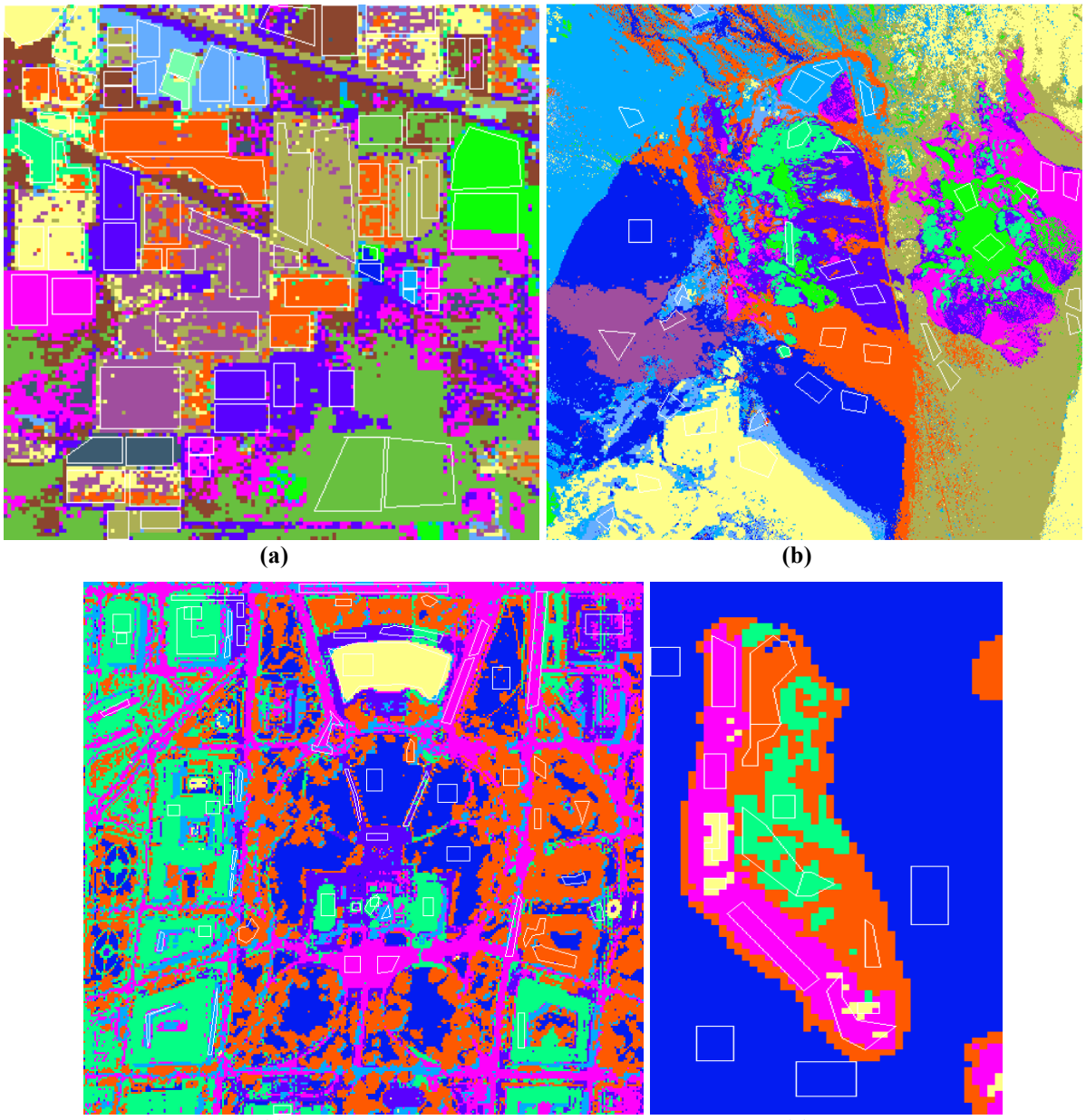
Even more, ECHO would benefit of some variability that allows computing variances within the training samples, and hence, using the Fisher Linear Discriminant. Nevertheless, ECHO penalizes also over-segmentation, since for a given scale that produces the best classification accuracy, the following finest scale always produces lower classification accuracies. Table 4.9 shows the classification accuracies (Kappa statistic) of the scale that produces the best classification accuracies, as reported in Table 4.2 and for comparison purposes, the classification accuracy of the previous coarser scale and the following finer scale. The best classification accuracies for all images occurred at the coarsest level or very

close to it. In fact, the best classification accuracies are within two levels from the coarsest level. In the case of the NW Indian Pines image, the best classification accuracy was for the coarsest level and that is why there is no coarser level on Table 4.9. These results indicate that the parameters selected to stop coarsening the grid (see Section 4.2) are appropriated for all the images used and allows to reduce the search for the best segmentation to the first few coarsest scales and that the classification with ECHO is a good indicator of segmentation accuracy, even tough it may prefer over-segmentation.

Table 4.9 Classification accuracies around the best scale

Classification Accuracy	Indian Pines		Cuprite		Washington DC		Enrique reef	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Best classification result	**98.4**	**89.0**	99.6	**97.2**	**100**	**90.5**	**100**	**98.1**
Previous coarser scale	-	-	**99.7**	94.2	100	85.3	69.3	65.1
Next finer scale	98.4	82.1	99.5	94.2	36.2	55.3	100	96.5

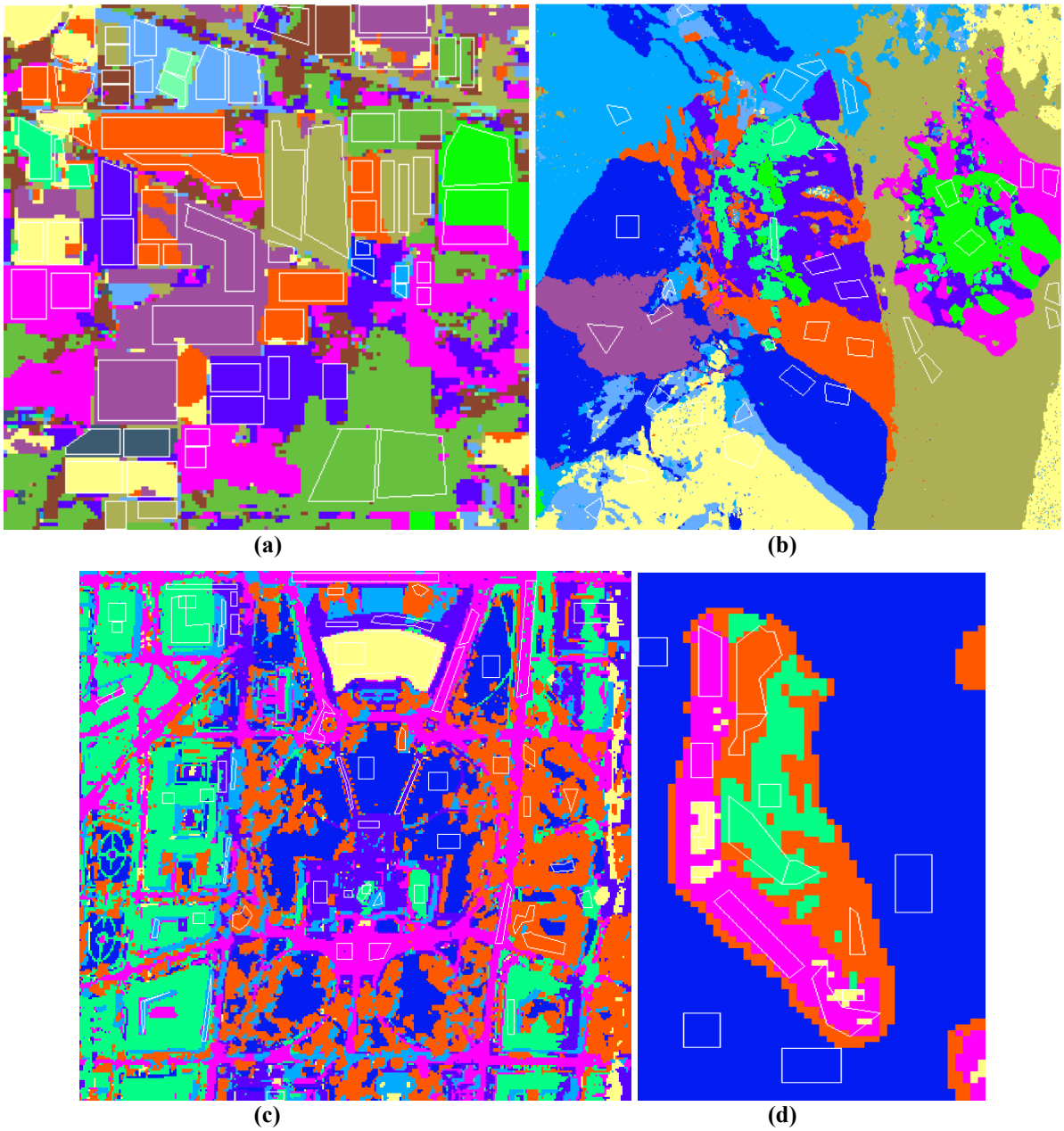
Figure 4.19 shows the classification map for the four original hyperspectral images used here. The areas of the training and testing samples are also indicated to facilitate the visual inspection of the uniformity of the classification within each region. Figure 4.20 shows the classification map for the hyperspectral images smoothed with AMG. It can be noticed here that the classification maps are more uniform within each training and testing region than with the original images (Figure 4.19). Figure 4.21 shows the segmentation maps for the best accuracies obtained using AMG to smooth and segment the four hyperspectral images. By visual comparison is easy to see that Figure 4.21 has the greatest uniformity within each training and testing areas, and in the whole segmentation map.



(a) **(b)**

(c) **(d)**

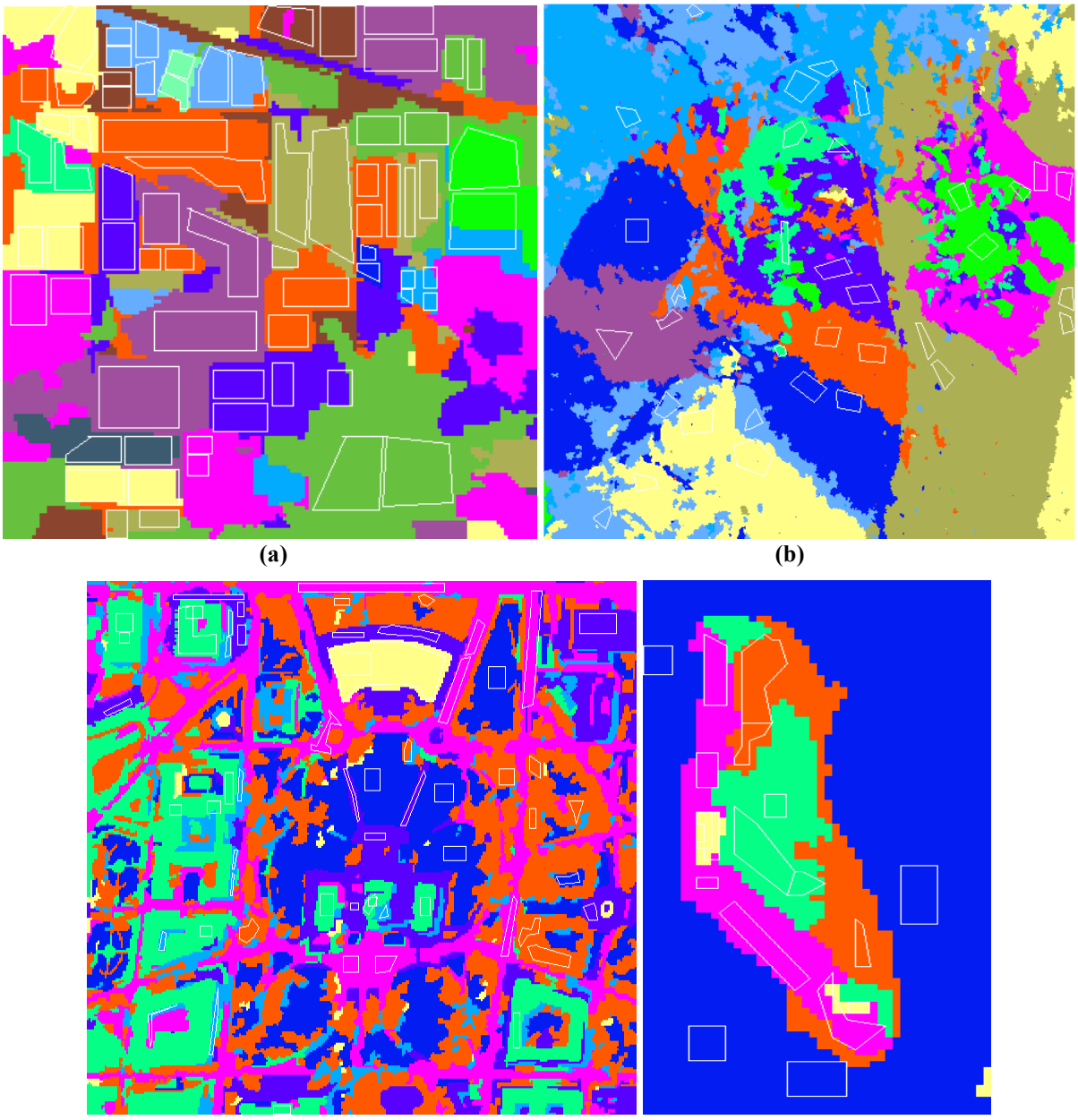
Figure 4.19 Classification maps for the original a) NW Indian Pines, b) Cuprite, c) Washington DC, and d) Enrique Reef images.



(a) **(b)**

(c) **(d)**

Figure 4.20 Classification maps for the smoothed a) NW Indian Pines, b) Cuprite, c) Washington DC, and d) Enrique Reef images.



(a) **(b)**

(c) **(d)**

Figure 4.21 Classification maps for the smoothed and segmented a) NW Indian Pines, b) Cuprite, c) Washington DC, and d) Enrique Reef images.

Table 4.10 shows the parameters corresponding to the results indicated in Table 4.2. It can be noticed that the α value for the best classification accuracies using smoothing only differs slightly from the value of α that produces the best accuracies using both smoothing and segmentation. Also, it can be noticed from Table 4.10 that the range of variability of parameters α, β, and γ is reduced, even tough the four images differ greatly in size and number and type of regions in the image, the level of noise, and the strength of the edges. Finally, and for completeness, we should mention here that the scale used to smooth with nonlinear diffusion all the hyperspectral images was 10, with scale-steps of 5 for all the images. Using this scale, the number of AMG scale-steps required was only two.

Table 4.10 Algorithm parameters.

Algorithm parameters	Indian Pines			Cuprite			Washington DC			Enrique reef		
	α	β	γ	α	β	γ	α	β	γ	α	β	γ
Smoothed	0.010	-	-	0.010	-	-	0.011	-	-	0.008	-	-
Segmented	-	0.008	0.006	-	0.007	0.003	-	0.010	0.007	-	0.008	0.005
Smoothed and Segmented ED	0.012	0.011	0.004	0.008	0.008	0.002	0.012	0.007	0.006	0.010	0.003	0.003
Smoothed and Segmented SA	0.008	0.007	0.003	0.006	0.003	0.001	0.011	0.011	0.009	0.010	0.008	0.003

The Indian Pines image is a patchy image, for which many objects are difficult to differentiate due to the variability of the spectral signatures within each region and the similarity between different classes (see Chapter 3). However, the boundaries of the different regions in the Indian Pines image are relatively strong and help the segmentation process. The separability of classes is higher on the Cuprite image, as can be seen from the training and testing accuracies, but the edges between the different regions are weak. The classes in the Washington DC image are also easier to separate and the edges are strong, but the number of objects in this image is very high, which may present a problem for segmentation. Finally, the Enrique reef image is very easy to classify and segment, but it

contains the highest level of variability (even after eliminating the most noisy bands), which can be appreciated on the visible variability of the seawater in Figure 4.5.b. Hence, even tough each image presents different challenges, we could successfully smooth, segment and classify all of them using the proposed AMG framework, with the parameters indicated on Table 4.10, which shows a relatively low range of variability.

4.3.3 Concluding remarks

We have integrated here geometric scale-space theories and algebraic multigrid solvers for the analysis and processing of hyperspectral images. We have shown that a geometric scale-space representation of hyperspectral images can be efficiently generated combining nonlinear diffusion PDEs and AMG methods, with good accuracy and scalability. Additionally, AMG provides the necessary structure to naturally obtain a hierarchical segmentation of the image. As our results indicate, the segmentation achieved using the smoothed image is better than just segmenting the original image.

We should note that a number of techniques are currently being developed for the fast computation of geometric PDEs, see for example [*Darbon and Siguelle*, 2006] and references there in.

5 CONCLUSIONS AND FUTURE WORK

> Before I came here I was confused about this subject. Having listened to your lecture I am still confused. But on a higher level.
> ENRICO FERMI

5.1 Introduction

PDE-based algorithms for image enhancement, segmentation and restoration have a large history of success for scalar and color images in computer vision, but they have been disregarded in segmentation and classification of hyperspectral imagery. This work showed that PDE-based, image processing methods can improve significantly image enhancing, segmentation and classification for hyperspectral imagery at a low computational cost, using semi-implicit schemes. Traditional statistical classification methods are very robust at low dimensional spaces, but they require an enormous amount of training data for higher dimensional data, as is the case of hyperspectral imagery, which is usually not available. On the other hand, parabolic PDEs offer a well-sounded, common framework to perform image smoothing and object segmentation using all image bands, with improved accuracy.

5.2 Conclusions

Chapter 3 shows that the formal scale-space provides a framework for image processing that can improve significantly image enhancing, and classification in hyperspectral imagery at a low computational cost, using semi-implicit schemes. The scale-space offers a well-founded, common framework to perform image smoothing, object-based segmentation, and classification.

We also showed that in the scale space representation of hyperspectral imagery, smoothing occurs in both the spectral and spatial domain. Not only the spatial features are smoothed out but also the spectral features (by spatial averaging). It was also shown that this framework is computationally feasible for hyperspectral imagery, allowing to process all the bands in the image containing information of the scene of interest. In addition, it was shown that the scale-space framework can be used to improve classification accuracy of hyperspectral imagery, since the smoothing process reduces intra-class variability, which increases class separability. Furthermore, the reduced variability enables high classification accuracy with simple linear classifiers such as the Fisher Linear Discriminant classifier. We also think that the scale space representation can have positive effects in other hyperspectral image processing tasks such as unsupervised classification, registration, image compression, and target recognition.

In particular, AOS and ADI semi-implicit schemes offer high performance in terms of accuracy and speedup of the computed solution of the anisotropic PDE, when scale-steps $\mu \leq 20\mu_0$ are used. If accuracy is of prime importance, the Douglas and Peaceman schemes can achieve higher accuracies at scale steps $\mu \leq 10\mu_0$ sacrificing speed. Even tough, we did not

achieve high speedups with the PCG methods, the initialization with ADI-LOD proved to be a good alternative, with speedups between 8 to 14 (see Table 3.3 to Table 3.5) that are superior to the Douglas and Peaceman-Rachford methods. In fact, we believe that if a better preconditioner is found for Equation 3.14, such that the PCG algorithm would run twice as fast as it did in our experiments, then PCG methods would beat all the approximated semi-implicit schemes seen on Chapter 3, since they have the highest accuracy and their speedup would be higher than the AOS and ADI methods. However, PCG methods also require more disk space, and finding a good preconditioner is still an art, hence, we consider that AOS and ADI methods are the best choice, if we are only interested in image smoothing.

In Chapter 4, we integrated the geometric scale-space theory and algebraic Multigrid solvers for the analysis and processing of hyperspectral images. With this integration, we improved accuracy and algorithmic scalability of the solution to the nonlinear diffusion PDE using the semi-implicit scheme in addition to provide the necessary structure to obtain a multiscale hierarchical segmentation of the image. AMG is slower than the approximated semi-implicit AOS and ADI schemes, but it is faster than PCG methods and much more accurate than the approximated semi-implicit methods studied on Chapter 3. The AMG-based segmentation algorithm enables higher level image processes such as unsupervised classification, registration, image compression, change detection, etc.

The experiments conducted in Chapter 4 indicate that it is always possible to improve the overall classification accuracy of hyperspectral image either by smoothing, segmenting, or combining both processes, irrespectively of the type of landscape the hyperspectral image is imaging. These results also indicate that even tough the rate of convergence of AMG can

be reduced by the presence of many strong transitions, as it happens in urban images, it is as accurate or better than the conjugated gradient method (see Figure 4.7), which we showed to be very accurate in solving the nonlinear diffusion PDE on hyperspectral imagery (Chapter 3).

As we can be seen from Table 4.2, images that have strong spatial or spectral variability and low classification accuracies, such as the NW Indian Pines, benefit more from the nonlinear diffusion PDE and segmentation, thanks to the strong reduction in both dimensions. On the other hand, images that already have good classification accuracies, such as the Cuprite image, would benefit less from the scale-space framework. Additionally, as Table 4.3 to Table 4.8 indicate, not all classes would benefit from smoothing or segmentation, since the dissimilarity metric employed does not necessarily improves the separability of a class with its background. Further work should be dedicated to explore other similarity metrics such as the spectral information divergence (SID) [*Chang*, 2000] and the use of statistics gathered from previous levels to improve the smoothing and segmentation processes.

Since the computational cost of smoothing and segmenting hyperspectral images is not negligible, potential users of this methodology should evaluate first if the original image would benefit from a reduction in the spectral and spatial variability. Besides supervised classification, other higher level processes in hyperspectral imagery could also benefit from the scale-space framework introduced here, such as unsupervised classification, image registration, and image compression.

Finally, the selection of the three parameters indicated on Table 4.10 might seem too complex for hyperspectral imagery, however and as stated in [*Martín-Herrero*, 2007], the tunability of the scale-space framework provides higher flexibility to enhance the performance of multiscale segmentation, as we exploited here. Other anisotropic diffusion PDEs should be also considered in the future for hyperspectral imagery, as it is also indicated in [*Martín-Herrero*, 2007].

We summarize the main contributions of this thesis in the following two statements.

- Even tough, the formal scale-space had already been introduced in the past (see Section 2.7), it has not been actively used in remote sensing, principally due to the fact that early numerical approaches to solve PDEs use simple explicit schemes that require a large number of iterations to reach a given scale. Since typical hyperspectral images consists of large datasets (see Section 1.1), the scale-space approach was not particularly attractive to the remote sensing community. In fact, the few papers that exist on this subject do not present the scale-space framework behind vector-valued PDEs, they only present the nonlinear diffusion PDE as a nonlinear filter that perform denoising on multispectral image. We had made the formal scale-space framework much more attractive computationally, thanks to the extension of semi-implicit schemes and PCG methods that runs 5 to 20 times faster than traditional explicit methods, in fact, according to our own experience, two steps of any of the semi-implicit schemes presented on Chapter 3 and 4 are enough to smooth hyperspectral imagery, with satisfactory accuracy. We also compare the different semi-implicit numerical methods presented here and provide some guidelines that allow the users to select, according to their needs, the best

semi-implicit method to the problem at hand. Up to our knowledge, this thesis is the first introduction of the formal scale-space framework to smooth and segment hyperspectral imagery.

- It has been recognized in the past (see Section 2.6) that the scale-space representation of grayscale and color images might improve segmentation accuracy. However, parabolic PDEs cannot provide hard segmentations, since their solution, at any scale, is always a smooth function, i.e. parabolic PDEs cannot produce discontinuous functions that are required to segment the images. Variational approaches that lie within the Mumford Shah segmentation model allow discontinuous solutions called free discontinuity problems, which solution lies on the space of bounded variations [*Aubert and Kornprobst*, 2002], where non-physical solutions are also possible and leads to challenging theoretical and numerical problems [*Weickert*, 1996]. Our main contribution here is to integrate the formal scale-space governed by conservation laws, with a segmentation algorithm that uses as much as possible the information obtained by the scale-space framework.

Even tough, heuristic approaches to multiscale representation of images can produce acceptable segmentations and have been successfully used to improve classification accuracies in the past (see for instance the scientific papers on behalf of the ECOGNITION suite of algorithms[13]), it is our belief that the introduction of a formal framework can allows us to fully understand and improve the current state of the art in multispectral image processing.

[13] http://www.definiens.com/scientific_papers.php?cat_id=1&link_id=24&sublink_id=25

5.3 Future Work

We propose here the following ideas to continue and improve this research,

- The computational and storage requirements to process hyperspectral imagery far exceed what is possible on a single workstation (see Section 1.1). Hence, there is a need of bringing the approximated AOS and ADI semi-implicit schemes, the PCG methods and the AMG solver and segmentation algorithms to high performance platforms. It is not easy to obtain a scalable parallelization of AMG, which can constitute a research project on its own, since first one need to optimize the sequential code and then minimize the need of communication between the processors in order to obtain scalability on implementation. The huge memory requirements of HSIs might require the use of distributed systems, where the data is spread among the system components. High performance implementations have several architectural alternatives in hyperspectral imagery such as distributed computing, clusters and the use of hardware implementations using FPGAs [*Plaza et al*, 2006, 2007], however each approach have their advantages and disadvantages and possibly a better alternative is to use a software/hardware codesign that exploits the advantages of hardware implementations with the flexibility and low cost implementation of complex functions [*Guilhermino et al*, 2003].

- We have extended the nonlinear diffusion PDE and made it a bit anisotropic, but fully anisotropic PDEs such as those indicated on Equations 2.23 to 2.25 could also be used to generate a scale-space representation of Hyperspectral imagery and be compared with our approach in terms of speed, segmentation and classification accuracy. On the other

hand, some authors [*Bachmann,* 2005, 2006] have argued that the nonlinear structure of hyperspectral imagery can be better exploited by similarity metrics based on the manifold coordinates i.e. using geodesic distances, rather than the usual metrics based on the Euclidean space (see Figure 5.1).

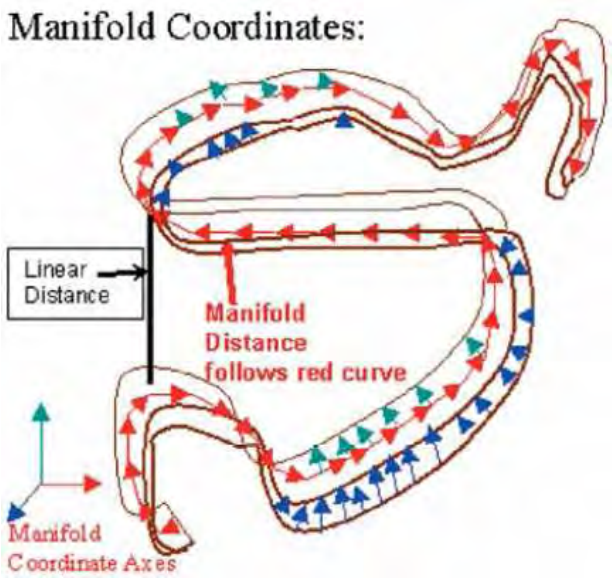


Figure 5.1 Manifold coordinates, taken from [*Bachmann et al*, 2005].

Hence, a possible continuation of this work could be the use of geometric PDEs embedded on non-flat manifolds such as the orientation diffusion PDEs proposed by [*Tang*, 2000; *Sapiro*, 2001] or the Beltrami flow of [*Kimmel*, 2000] (see Section 2.5). The manifold coordinate system can be obtained using the same approach as [*Bachmann et al*, 2005, 2006]. The running time of this approach is $O(N \log^2 N)$, which is quite good, but indicates that PDEs on manifolds are not as scalable as our simpler approach based on the Euclidean space. Nevertheless, it is possible that class separability increases along the manifold coordinates as the work of [*Bachmann et al*, 2005, 2006; *Mohan et al*, 2007] indicates, such that the extra cost could be more than justified.

- Several free and commercial algorithms are available nowadays that provide hierarchical multiscale segmentation of multispectral imagery such as ERDAS [14], EDISON [15], INFOPACK[16], SPRING[17], RHSEG[18] (NASA), GENIE PRO[19], and DEFINIENS[20]. Most of these algorithms are restricted to multispectral imagery and many of them are based on heuristic clustering strategies (see [*Hay et al*, 2003; *Meinel and Neuber*, 2004] for a comparison among them). A comparison of the formal scale-space representation of hyperspectral imagery introduced here, with other state of the art segmentation algorithms is a work that must be addressed in the near future. The results of such comparison could be to demonstrate the advantages of the formal scale-space framework over hierarchical clustering in hyperspectral imagery, or give guidelines on how to improve our approach to make it competitive with current state of the art hierarchical segmentation algorithms.
- From the segmentation map obtained at the coarsest scale, we can obtain statistics for each segment, represented by histograms on each band, or we can obtain generalized vector-valued histograms [*Rubner et al*, 2000] that represent better the fact that segments in hyperspectral images are vector-valued. Extracting statistics at the coarsest level is cheaper at this scale, since the number of objects is O(log$N$), where N is the number of pixels in the image (see Chapter 4). The generalized histograms can be obtained from each segment by vector-quantization [*Linde et al*, 1980; *Nasrabadi and King*, 1988].

[14] http://gi.leica-geosystems.com/LGISub7x384x0.aspx
[15] http://www.caip.rutgers.edu/riul/research/code/EDISON/index.html
[16] http://www.infosar.co.uk/misc/products.html
[17] http://www.dpi.inpe.br/spring/english/index.html
[18] http://techtransfer.gsfc.nasa.gov/RHSEG/index.html

Using these histograms one can determine, by similarity metrics between histograms, if two or more segments belong to the same class. Typical similarity metrics used for hyperspectral imagery are the Chi-square and Kolmogorov-Smirnov distances for the band by band histograms [*Rubner et al*, 2000; *Katartzis et al*, 2004], and the Earth Moving Distance [*Rubner et al*, 2000] for the generalized vector-valued histograms. The histograms can use the real valued spectral signatures at the finest level or we can use Complex Transforms as introduced by [*Castrodad et al*, 2007].

- A necessary step to bring the scale-space framework to the level of usability required by commercial applications consists of providing some help to select the parameters required by the scale-space framework (see Chapter 4). One way to do this (used by most of the commercial segmentation algorithms cited before) is to provide a friendly user interface that allows the users to refine the results interactively. Another possibility consists of determining automatically the parameters needed by the scale-space framework, from the image itself (see Chapter 3) and from tabulated values learned previously according to different user's needs. Another approach could be to explore the parameter space using genetic algorithms that refine the search by maximizing a given objective function that measures the quality of the segmentation, as GENIE PRO does.

[19] http://www.genie.lanl.gov/
[20] http://www.definiens.com/definiens-professional_11_7_9.html

6 ETHICAL CONSIDERATIONS

> Unfortunately, some have begun to pursue scientific research for its own benefit or for profit, without respect for human life.
> NATHAN DEAL

> There is a computer disease that anybody who works with computers knows about. It's a very serious disease and it interferes completely with the work. The trouble with computers is that you "play" with them!
> RICHARD P. FEYNMAN

Ethics in computer science and software engineering is usually associated with viruses, software piracy, privacy and security. Leading computer science professional organizations as IEEE[21] and ACM[22] have published ethic and professional conduct codes, but they are principally oriented towards professional practicers, not researchers [*Honeycutt and Wright*, 2006]. This chapter focuses on the ethical conduct of the researcher in computer and information sciences and engineering. A complete exposition of the subject can be found in the work of [*Wright*, 2006], within the NSF project Land Grant University Research Ethics (LANGURE[23]), to develop a model curriculum in research ethics for doctoral candidates.

[21] http://www.ieee.org/portal/pages/about/whatis/code.html
[22] http://www.acm.org/constitution/code.html
[23] http://www.chass.ncsu.edu/langure/

Computers are routinely used to predict weather, crop productions, disease propagation, financial estimates, etc, affecting directly or indirectly the lives of millions of people, even those who never had use a computer. Faulty software or software that does not meet in application, the performance reported on research papers can cause great lost in terms of money, time and sometimes human lives. Given the strong dependence of modern society on computer systems, it is of prime importance to address the ethical issues related to computer science and computer engineering research. As stated in the ACM code of ethics, "When designing or implementing systems, computing professionals must attempt to ensure that the products of their efforts will be used in socially responsible ways, will meet social needs, and will avoid harmful effects to health and welfare."

In particular, three cornerstone ethical responsibilities must be present on any modern scientific research [*Wright*, 2006]: "the responsible conduct of research, clear and complete recording and reporting of research procedures, results and analysis" (repeatability of the experiments), "respect for those that might be affected by the research."

As in our present work, empirical sciences rely on measurements and observations to support the conclusions of a research. Software implementations created with the purpose of providing a superior performance of one algorithm over another should be as neutral as possible, and when experimental bias exists, this should be clearly expressed and their effect on the results explained. Well-known metrics should be used, whenever possible to compare two or more algorithms. Different programming languages and compilers have different features that allow optimizations that target a specific architecture, and hence, two different algorithms might be incorrectly evaluated if they are implemented and or compiled under

different languages or platforms. Hence, when comparing two or more different algorithms in terms of their performance, they must be coded with the same language and compiled with the same compiler and platform, in order to mitigate the bias introduced by the skills of the programmer with a specific language and the bias introduced by different compilers, compiler settings and platforms. There might exist involuntary bias, if time consuming processes are running in the background, for instance an antivirus action, while one of the algorithms is tested. Hence, and even tough computers are deterministic, many unaccounted factors might bias the result of an experiment and it is responsibility of the researcher to ensure the repeatability of the experiments reported.

As [*Wright*, 2006] states: "the ability to duplicate the work of other researchers is perhaps the most fundamental principle and responsibility of science. Repeating an experiment allows a new result to be corroborated or refuted, as well as providing the means to restate and refine the problem under consideration. Duplicating the prior work of other researchers is often also more than simply recreating the earlier experiment: the later researcher should also be looking for new results that extend or clarify the earlier work. It is the continual refinement of hypotheses that builds credibility of researchers and results alike".

It is a well-known issue in research ethics, the infringement of copyrights and patents of other publications or researchers, upon which the research is based. With the increasing use of the web (and powerful search engines such as Google) in research providing easy and almost unlimited access to the algorithms, figures and the work of other researchers, several ethical issues arise [*Duncan*, 1996], such as invasion of privacy of unknowing subjects, sharing and use of the data without approval, proper credit to the authors, and rushing to

publish incomplete or unreviewed work on the web. Web based research (as it is conducted nowadays by almost all students) has a great potential to harm the rights of unknowing subjects trough search engines. Researchers are nowadays strongly tempted by the possibility of luring data from the web and publish their results, without proper review, to obtain a greater impact of their research. Hence, it is important to obtain the review of a disinterred third party, such as the peer reviewed process required by most of the prestigious proceedings and journals.

In our work, we compared algorithms under the same language and compilers (Matlab code with Matlab code, and C++ code with C++ code with the same compiler settings) and repeat the experiments in order to avoid the presence of processes running on background. We also document here all the theory and formulation necessary to duplicate our work that could not be included in our research papers, due to space limitations. We also make proper references to the ideas and work of previous researchers thorough this document.

APPENDIX A: ALGORITHMS

> Programming is one of the most difficult branches of applied mathematics; the poorer mathematicians had better remain pure mathematicians.
> EDSGER DIJKSTRA

APPENDIX A1 THOMAS ALGORITHM FOR VECTOR-VALUED IMAGES

The following algorithm is a straightforward extension of the algorithm presented in [*Weickert et al*, 1998] for scalar images. Let **GX** = **V**, be a tri-diagonal linear system where **X**, and **V** are $N\times M$ dense matrices (hyperspectral images) and **G** is an $N\times N$ tri-diagonal matrix of the form,

$$\mathbf{G} = \begin{pmatrix} \alpha_1 & & & \beta_1 & & \\ & \alpha_2 & & & \ddots & \\ & & \ddots & & & \beta_{N-d} \\ & & & \ddots & & \\ \gamma_1 & & & & \alpha_{N-1} & \\ & \ddots & & & & \\ & & \gamma_{N-d} & & & \alpha_N \end{pmatrix} \Big\} d$$

where, $1\le d\le N$-1. All the dense matrices are represented here as hyperspectral images of the form,

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{pmatrix},$$

where, $\mathbf{x}_i$ are spectral vectors of length M. The first step is to factorize $\mathbf{G} = \mathbf{LU}$ such that $\mathbf{L}$ and $\mathbf{U}$ are bi-diagonal matrices of the form,

$$\mathbf{L} = \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ l_1 & & & \ddots & & \\ & \ddots & & & 1 & \\ & & l_{N-d} & & & 1 \end{pmatrix}, \quad \mathbf{U} = \begin{pmatrix} m_1 & & & \beta_1 & & \\ & m_2 & & & \ddots & \\ & & \ddots & & & \beta_{N-d} \\ & & & \ddots & & \\ & & & & m_{N-1} & \\ & & & & & m_N \end{pmatrix}$$

LU Decomposition

$for\ i = 1, \cdots, d$
$\quad m_i = \alpha_i$
$for\ i = 1, \cdots, N-d$
$\quad l_i = \gamma_i / m_i$
$\quad m_{i+d} = \alpha_{i+d} - l_i \beta_i$

It is clear that the LU decomposition takes O(N) time and the temporal vectors l and m require only ~2N disk space. Now, the system $\mathbf{LUX} = \mathbf{V}$ can be solved as,

$$\begin{cases} \mathbf{LY} = \mathbf{V} \\ \mathbf{UX} = \mathbf{Y} \end{cases}$$

Forward substitution

We solve here the system **LY** = **V**,

$$
\begin{aligned}
&\textit{for } i = 1, \cdots, d \\
&\qquad \mathbf{y}_i = \mathbf{v}_i \\
&\textit{for } i = d+1, \cdots, N \\
&\qquad \mathbf{y}_i = \mathbf{v}_i - l_{i-d}\mathbf{y}_{i-d}
\end{aligned}
$$

Backward substitution

We solve here the system **UX** = **Y,**

$$
\begin{aligned}
&\textit{for } i = N, \cdots, N-d+1 \\
&\qquad \mathbf{x}_i = \mathbf{y}_i / m_i \\
&\textit{for } i = N-d, \cdots, 1 \\
&\qquad \mathbf{x}_i = \left(\mathbf{y}_i - \beta_i \mathbf{x}_{i+d}\right) / m_i
\end{aligned}
$$

Forward and backward substitution performs N linear combinations of vectors of size M. Hence, the running time of Thomas algorithm for vector-valued images is dominated by these two operations and takes O(NM) time. Notice that the temporal variable **Y** is not required, since we can use **X** instead of **Y** on the forward substitution and overwrite **X** on the backward substitution. Hence, the disk space requirements of the vector-valued Thomas algorithm are O(N) and it does not cause overhead on hyperspectral imagery where $M >> 1$.

Finally, the Thomas algorithm given here is completely general, but for our particular case, $d = 1$ for $\mathbf{G}_x$ and $d = N_x$ for $\mathbf{G}_y$ as were defined on Section 3.1.

APPENDIX A2 ANALYTICAL INCOMPLETE CHOLESKY FACTORIZATION

The following is the analytical incomplete **LU** Factorization that can be found in [*Saad*, 2003], pg. 305 and it is repeated here for completeness. The incomplete **LU** factorization of matrix **A** that comes from a five point discretization of an elliptic PDE, given by,

$$\mathbf{A} = \begin{pmatrix} \delta_1 & \gamma_2 & & \varphi_d & & \\ \beta_2 & \delta_2 & \ddots & & \ddots & \\ & \ddots & \ddots & \gamma_i & & \varphi_N \\ \eta_d & & \beta_i & \delta_i & \ddots & \\ & \ddots & & \ddots & \ddots & \gamma_N \\ & & \eta_N & & \beta_N & \delta_N \end{pmatrix}$$

takes the form $\mathbf{A} \approx (\mathbf{D} - \mathbf{E})\mathbf{D}^{-1}(\mathbf{D} - \mathbf{F})$, where **E** is the lower diagonal part of **A**, **F** is the upper diagonal part of **A**, and **D** can be found recursively by,

$$d_i = \delta_i - \frac{\beta_i \gamma_i}{d_{i-1}} - \frac{\eta_i \varphi_i}{d_{i-m}}, \quad i = 1, \cdots, N.$$

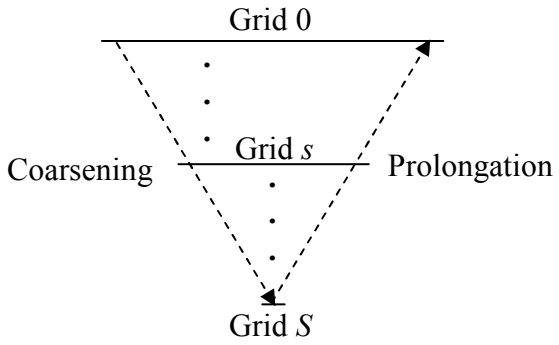
In our particular case, $\mathbf{A} = \mathbf{I} - \mu\mathbf{G}_x - \mu\mathbf{G}_y$ which is symmetric, so $\mathbf{E} = \mathbf{F}$ and the preconditioner is given by,

$$\mathbf{C} = (\mathbf{D} - \mathbf{E})\mathbf{D}^{-1}(\mathbf{D} - \mathbf{E})^T = (\mathbf{D} - \mathbf{E})\mathbf{D}^{-1/2}\left[(\mathbf{D} - \mathbf{E})\mathbf{D}^{-1/2}\right]^T \equiv \widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^T,$$

which is the incomplete Cholesky factorization used in our work.

APPENDIX A3 AMG V-CYCLE

For clarity, we repeat here Figure 4.4



- Grid 0:
  - Relax υ_0 times $(\mathbf{I}-\mu\mathbf{G}^0)\mathbf{X}^0=\mathbf{U}_n$ with initial guess $\mathbf{U}_n$.
  - Compute the error $\mathbf{X}^0=(\mathbf{I}-\mu\mathbf{G}^0)\mathbf{X}^0-\mathbf{U}_n$, the residual $\mathbf{F}^0=(\mathbf{I}-\mu\mathbf{G}^0)\mathbf{X}^0$, and restrict it as $\mathbf{F}^1=\mathbf{H}_c^f\mathbf{F}^0$.

    …
    - Grid s:
      - Relax υ_s times $(\mathbf{I}-\mu\mathbf{G}^s)\mathbf{X}^s=\mathbf{F}^s$, with initial guess $\mathbf{0}$.
      - Compute the error $\mathbf{X}^s=\mathbf{F}^s-(\mathbf{I}-\mu\mathbf{G}^s)\mathbf{X}^s$, the residual $\mathbf{F}^s=(\mathbf{I}-\mu\mathbf{G}^s)\mathbf{X}^s$, and restrict it as $\mathbf{F}^{s+1}=\mathbf{H}_f^c\mathbf{F}^s$.

        …
        - Grid S: Solve exactly $(\mathbf{I}-\mu\mathbf{G}^s)\mathbf{X}^s=\mathbf{F}^s$ to obtain $\mathbf{X}^S$.

        …
    - Grid s:
      - Correct $\mathbf{X}^s=\mathbf{X}^s+\mathbf{H}_c^f\mathbf{X}^{s+1}$.
      - Relax υ_s times $(\mathbf{I}-\mu\mathbf{G}^s)\mathbf{X}^s=\mathbf{F}^s$, with initial guess $\mathbf{X}^s$.

    …
- Grid 0:
  - Correct $\mathbf{X}^0=\mathbf{X}^0+\mathbf{H}_c^f\mathbf{X}^1$.
  - Relax υ_0 times $(\mathbf{I}-\mu\mathbf{G}^0)\mathbf{X}^0-\mathbf{U}_n$ with initial guess $\mathbf{X}^0$ to obtain $\mathbf{X}^0\approx\mathbf{U}_{n+1}$.

APPENDIX B: DETAILS ON SELECTED SUBJECTS

APPENDIX B1: APPROXIMATED SEMI-IMPLICIT METHODS

Douglas-Rachford:

Since $\mathbf{I}-\mu\mathbf{G}_x^k-\mu\mathbf{G}_y^k=\left(\mathbf{I}-\mu\mathbf{G}_x^k\right)\left(\mathbf{I}-\mu\mathbf{G}_y^k\right)-\mu^2\mathbf{G}_x^k\mathbf{G}_y^k$, we can rewrite Equation **Error! Reference source not found.** as,

$$\left(\mathbf{I}-\mu\mathbf{G}_x^k\right)\left(\mathbf{I}-\mu\mathbf{G}_y^k\right)\mathbf{U}^{k+1}=\left(\mathbf{I}+\mu^2\mathbf{G}_x^k\mathbf{G}_y^k\right)\mathbf{U}^k+\mu^2\mathbf{G}_x^k\mathbf{G}_y^k\left(\mathbf{U}^{k+1}-\mathbf{U}^k\right).$$

If $\mathbf{U}^{k+1}$ is close to $\mathbf{U}^k$, then,

$$\left(\mathbf{I}-\mu\mathbf{G}_x^k\right)\left(\mathbf{I}-\mu\mathbf{G}_y^k\right)\mathbf{U}^{k+1}\approx\left(\mathbf{I}+\mu^2\mathbf{G}_x^k\mathbf{G}_y^k\right)\mathbf{U}^k.$$

It is easy to verify that this expression is equivalent to

$$\left(\mathbf{I}-\mu\mathbf{G}_x^k\right)\left[\left(\mathbf{I}-\mu\mathbf{G}_y^k\right)\mathbf{U}^{k+1}+\mu\mathbf{G}_y^k\mathbf{U}^k\right]\approx\left(\mathbf{I}+\mu\mathbf{G}_y^k\right)\mathbf{U}^k$$

which can be solved as,

$$\begin{cases}\left(\mathbf{I}-\mu\mathbf{G}_x^k\right)\mathbf{U}^{k+1/2}=\left(\mathbf{I}+\mu\mathbf{G}_y^k\right)\mathbf{U}^k\\ \left(\mathbf{I}-\mu\mathbf{G}_y^k\right)\mathbf{U}^{k+1}\approx\mathbf{U}^{k+1/2}-\mu\mathbf{G}_y^k\mathbf{U}^k\end{cases}.$$

Peaceman-Rachford:

Having into account the following identities,

$$\mathbf{I}-\mu\mathbf{G}_x^k-\mu\mathbf{G}_y^k=\left(\mathbf{I}-\frac{1}{2}\mu\mathbf{G}_x^k\right)\left(\mathbf{I}-\frac{1}{2}\mu\mathbf{G}_y^k\right)-\left(\frac{1}{2}\mu\mathbf{G}_x^k+\frac{1}{2}\mu\mathbf{G}_y^k+\frac{1}{4}\mu^2\mathbf{G}_x^k\mathbf{G}_y^k\right),$$

$$\mathbf{I}=\left(\mathbf{I}+\frac{1}{2}\mu\mathbf{G}_x^k\right)\left(\mathbf{I}+\frac{1}{2}\mu\mathbf{G}_y^k\right)-\left(\frac{1}{2}\mu\mathbf{G}_x^k+\frac{1}{2}\mu\mathbf{G}_y^k+\frac{1}{4}\mu^2\mathbf{G}_x^k\mathbf{G}_y^k\right).$$

Equation 3.15 can be rewritten as,

$$\left(\mathbf{I}-\frac{1}{2}\mu\mathbf{G}_x^k\right)\left(\mathbf{I}-\frac{1}{2}\mu\mathbf{G}_y^k\right)\mathbf{U}^{k+1}-\left(\frac{1}{2}\mu\mathbf{G}_x^k+\frac{1}{2}\mu\mathbf{G}_y^k+\frac{1}{4}\mu^2\mathbf{G}_x^k\mathbf{G}_y^k\right)\mathbf{U}^{k+1}=$$
$$\left(\mathbf{I}+\frac{1}{2}\mu\mathbf{G}_x^k\right)\left(\mathbf{I}+\frac{1}{2}\mu\mathbf{G}_y^k\right)\mathbf{U}^{k}-\left(\frac{1}{2}\mu\mathbf{G}_x^k+\frac{1}{2}\mu\mathbf{G}_y^k+\frac{1}{4}\mu^2\mathbf{G}_x^k\mathbf{G}_y^k\right)\mathbf{U}^{k},$$

where, if $\mathbf{U}^{k+1}$ is close to $\mathbf{U}^k$, then,

$$\left(\mathbf{I}-\frac{1}{2}\mu\mathbf{G}_x^k\right)\left(\mathbf{I}-\frac{1}{2}\mu\mathbf{G}_y^k\right)\mathbf{U}^{k+1}\approx\left(\mathbf{I}+\frac{1}{2}\mu\mathbf{G}_x^k\right)\left(\mathbf{I}+\frac{1}{2}\mu\mathbf{G}_y^k\right)\mathbf{U}^{k}.$$

This equation is also known as the Crank-Nicholson scheme and it is second order accurate, both in scale and space, which can be solved as,

$$\begin{cases}\left(\mathbf{I}-\frac{1}{2}\mu\mathbf{G}_x^k\right)\mathbf{U}^{k+1/2}=\left(\mathbf{I}+\frac{1}{2}\mu\mathbf{G}_y^k\right)\mathbf{U}^{k}\\ \left(\mathbf{I}-\frac{1}{2}\mu\mathbf{G}_y^k\right)\mathbf{U}^{k+1}\approx\left(\mathbf{I}+\frac{1}{2}\mu\mathbf{G}_x^k\right)\mathbf{U}^{k+1/2}\end{cases}.$$

Additive Opperator Splitting (AOS):

Let us rewrite Equation 3.15 as,

$$\frac{1}{2}\left(\mathbf{I}-2\mu\mathbf{G}_x^k+\mathbf{I}-2\mu\mathbf{G}_y^k\right)\mathbf{U}^{k+1}=\mathbf{U}^{k}.$$

If we pre-multiply both sides of this equation by $\left(\mathbf{I}-2\mu\mathbf{G}_x^k\right)^{-1}$,

$$\frac{1}{2}\left(\mathbf{I}+\left(\mathbf{I}-2\mu\mathbf{G}_x^k\right)^{-1}\left(\mathbf{I}-2\mu\mathbf{G}_y^k\right)\right)\mathbf{U}^{k+1}=\left(\mathbf{I}-2\mu\mathbf{G}_x^k\right)^{-1}\mathbf{U}^{k}.$$

AOS uses the first order Taylor's series approximation $(\mathbf{I}\text{-}\mathbf{X})^{-1}\approx\mathbf{I}+\mathrm{X}$ on the LHS of the previous equation such that

$$\frac{1}{2}\left(\mathbf{I}+\left(\mathbf{I}+2\mu\mathbf{G}_x^k\right)\left(\mathbf{I}-2\mu\mathbf{G}_y^k\right)\right)\mathbf{U}^{k+1}\approx\left(\mathbf{I}-2\mu\mathbf{G}_x^k\right)^{-1}\mathbf{U}^{k},$$

which reduces to

$$\frac{1}{2}\left(2\mathbf{I}+2\mu\left(\mathbf{G}_x^k-\mathbf{G}_y^k\right)-4\mu^2\mathbf{G}_x^k\mathbf{G}_y^k\right)\mathbf{U}^{k+1}\approx\left(\mathbf{I}-2\mu\mathbf{G}_x^k\right)^{-1}\mathbf{U}^k.$$

By the same reasoning, we can pre-multiply both sides of Equation 3.15 by $\left(\mathbf{I}-2\mu\mathbf{G}_y^k\right)^{-1}$ and obtain,

$$\frac{1}{2}\left(2\mathbf{I}+2\mu\left(\mathbf{G}_y^k-\mathbf{G}_x^k\right)-4\mu^2\mathbf{G}_x^k\mathbf{G}_y^k\right)\mathbf{U}^{k+1}\approx\left(\mathbf{I}-2\mu\mathbf{G}_y^k\right)^{-1}\mathbf{U}^k.$$

Adding these two last equations and disregarding the $\mu^2\mathbf{G}_x^k\mathbf{G}_y^k$ term,

$$\mathbf{U}^{k+1}\approx\frac{1}{2}\left[\left(\mathbf{I}-2\mu\mathbf{G}_x^k\right)^{-1}+\left(\mathbf{I}-2\mu\mathbf{G}_y^k\right)^{-1}\right]\mathbf{U}^k,$$

which can be solved as,

$$\begin{aligned}&\left(\mathbf{I}-2\mu\mathbf{G}_x^k\right)\mathbf{U}_x^{k+1}=\mathbf{U}^k,\quad\left(\mathbf{I}-2\mu\mathbf{G}_y^k\right)\mathbf{U}_y^{k+1}=\mathbf{U}^k\\&\mathbf{U}^{k+1}\approx\frac{\mathbf{U}_x^{k+1}+\mathbf{U}_y^{k+1}}{2}\end{aligned}.$$

APPENDIX B2: POSITIVE DEFINITIVENESS AND CONDITION NUMBER OF THE DIFFUSION MATRIX IN THE SEMI-IMPLICIT EQUATION

Let us prove that matrix $\mathbf{A}^k \equiv \mathbf{I}\text{-}\mu\mathbf{G}^k$ is positive definite, so that the CG is an efficient iterative method to solve it. The Gershgorin theorem [*Saad*, 2003] states that for any eigenvalue λ of a matrix $\mathbf{A}^k$ (in our case) there exists $1 \le i \le N$ such that

$$\left|\lambda - a_{ii}^k\right| \le \sum_{j\neq i}\left|a_{ij}^k\right|.$$

Since $\mathbf{A}^k$ is strictly diagonally dominant for all valid semi-implicit discretizations of the nonlinear diffusion equation and the elements in the main diagonal are positive (see Section 2.4),

$$\left|\lambda - a_{ii}^k\right| \leq \sum_{j \neq i} \left|a_{ij}^k\right| < a_{ii}^k ,$$

hence,

$$0 < \lambda < 2a_{ii}^k ,$$

and since all the eigenvalues of $\mathbf{A}^k$ are positive, then $\mathbf{A}^k$ is positive definite [*Horn and Johnson*, 2006]. Notice that until now, we have not used the known structure of our particular matrix given on Equation 3.14 and hence, this result is valid for any other discretizations that satisfy the requirements for a valid discrete scale-space.

We can obtain much more information from the Gershorin theorem for our particular matrix, since we know that (see Equation 3.13) for each eigenvalue, there exists $1 \leq i \leq N$ such that

$$\left|\lambda - 1 - \mu \sum_l g_{i,l}^k\right| \leq \sum_l \left|\mu g_{i,l}^k\right| = \mu \sum_l g_{i,l}^k ,$$

given that $\mu > 0$ and $g_{i,l}^k > 0$ (Section 2.4) and we have defined for convenience,

$$\sum_l g_{i,l}^k = g_{i,E}^k + g_{i,W}^k + g_{i,N}^k + g_{i,S}^k .$$

Gershorin inequality reduces in our case to

$$1 \leq \lambda \leq 1 + 2\mu \sum_l g_{i,l}^k ,$$

This result confirm us that $\mathbf{A}^k$ is positive definite, but also provide us with lower and upper bounds for λ, which is useful to determine an upper bound for the condition number of $\mathbf{A}^k$,

$$\lambda_{\min} \geq 1, \quad \lambda_{\max} \leq 1 + 2\mu \max_i \left\{ \sum_l g_{i,l}^k \right\}.$$

Hence, the condition number will be bounded by,

$$\kappa \leq 1 + 2\mu \max_i \left\{ \sum_l g_{i,l}^k \right\}.$$

From the previous result, we can see that μ affects directly the condition number, so that large values of μ, as those used in Section 3.3, deteriorates the convergence of the CG (see Equation 3.20). Also, we can verify from this result that as stated in [*Weickert et al*, 1998] the threshold parameter α in Equation 3.6 affects the condition number. In order to see this, let us consider the case when $\alpha \to 0$, then $g \to 0$ (see Equation 3.6) and $\kappa \to 1$, but this case is of no practical importance, since if $g \to 0$ there is no diffusion whatsoever. On the other hand, as α increases there is more diffusion and if $\alpha \to 1$ then $g \to 1$ and the condition number is maximal for a given μ. Hence, as α increases the condition number increases too.

REFERENCES

Akselrod-Ballin, A., M. Galun, M. J. Gomori, M. Filippi, P. Valsasina, R. Basri, and A. Brandt, "An integrated segmentation and classification approach applied to multiple sclerosis analysis," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 1122-1129, 2006.

Alvarez, L. , F. Guichard, P. L. Lions, and J. M. Morel, "Axioms and fundamental equations of image processing," *Archives of Rational Mechanics*, vol. 123, pp. 199-257, Sept. 1993.

Alvarez, L., P. L. Lions, and J. M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion II," *SIAM Journal of Numerical Analysis*, vol. 29, no. 3, pp. 845-866, June, 1994.

Arzuaga-Cruz, E., L. O. Jimenez-Rodriguez, D. Kaeli, M. Vélez-Reyes, E. Rodríguez-Díaz, L. Santos-Campis, and C. Santiago, "A MATLAB Toolbox for hyperspectral Image Analysis," *IEEE International Symposium on Geoscience and Remote Sensing*, vol. 7, pp. 4839 – 4842, 2004.

Aubert, G. and P. Kornprobst, *Mathematical problems in image processing: partial differential equations and the calculus of variations*, Applied Mathematical Sciences, vol. 147, New York, NY: Springer-Verlag, 2002.

Baatz, M. and A. Schäpe, "Multiresolution segmentation – an optimization approach for high quality multi-scale image segmentation," in *Angewandte Geographische Informationsverarbeitung*, vol. xii, Strobl, Blaschke, and Griesebner, Eds. Heidelberg: Wichmann-Verlag, pp. 12-23, 2000.

Bachman, C. M., T. L. Ainsworth, and R. A., Fusina, "Exploiting manifold geometry in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, March, 2005.

Bachmann, C. M., Ainsworth, T. L., and Fusina, R. A., "Improved Manifold Coordinate Representations of Large-Scale Hyperspectral Scenes," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 10, pp. 2786-2803, October, 2006.

Ball, J. E. and L. M. Bruce, "Level set segmentation of remotely sensed hyperspectral images," *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, vol. 8, pp. 5638- 5642, 2005.

Barash, D., T. Schlick, M. Israeli, and R. Kimmel, "Multiplicative operator splittings in nonlinear diffusion: from spatial splitting to multiple time steps," *Journal of Mathematical Imaging and Vision*, vol. 19, pp. 33-48, 2003.

Barret, R., M. Berry, T. F. Chan, J. Demmel, J. M. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst, *Templates for the solution of linear systems: building blocks for iterative methods*, Philadelphia, PA: SIAM, pp. 10-14, 25-27, 39-52, 1994.

Black, M. J., G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Transactions on Image processing*, vol. 7, no. 3, pp. 421-431, March 1998.

Blaschke, T., S. Lang, and E. Lorup, "Object-oriented image processing in an integrated GIS/remote sensing environment and perspectives for environmental applications," *Environmental Information for Planning, Politics and the Public*, A. Cremers, and K. Greve, Marburg Eds. Metropolis-Verlag, pp. 555-570, 2000.

Blaschke, T. and J. Strobl, "What's wrong with pixels? Some recent developments interfacing remote sensing and GIS," GIS – *Zeitschrift für Geoinformationssysteme*, vol. 6, pp.12-17, 2001.

Bosworth, J. H. and S. T. Acton, "Morphological scale-space in image processing," *Digital Signal Processing*, vol. 13, no. 2, pp. 338-367, 2003.

Brandt, A., S. F. McCormick, and J. W. Ruge, *Algebraic multigrid for automatic multigrid solutions with application to geodetic computations*, Institute for Computational Studies, Fort Collins, CO Report, Oct., 1982.

Brandt, A., "General highly accurate algebraic coarsening," *Electronic Transactions on Numerical Analysis*, vol. 10, pp. 1-20, Feb., 2000.

Briggs, W. L., V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, 2nd Ed. Philadelphia, PA: SIAM, pp. 7-48, 137-154, 2000.

Bruzzone, L. and L. Carlin, "A Multilevel context-based system for classification of very high spatial resolution images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 9, pp. 2587-2600, Sept., 2006.

Castillo, P. and Saad, Y., "Preconditioning the Matrix Exponential Operator with Applications," *Journal of Scientific Computing*, vol. 13, no. 3, Sept. 1998.

Castrodad, A., E. H. Bosch, and R. Resmini, "Hyperspectral imagery transformations using real and imaginary features for improved classification," *Proceedings of the SPIE*, vol. 6565, pp. 1B1-1B10, May, 2007.

Catté, F., P. L. Lions, J. M. Morel, T. Coll, "Image selective smoothing and edge detection by nonlinear diffusion," *SIAM Journal of Numerical Analysis*, vol. 19, pp. 182-193, February, 1992.

Chang, C. I. "An information-theoretic approach to spectral variability, similarity, and discrimination for hyperspectral image analysis," *IEEE Transactions on Information Theory*, vol. 46, no. 5, pp. 1927 – 1932, Aug., 2000.

Chen, Q.-X., J.-C. Luo, C.-h. Zhou, and T. Pei, "A hybrid multiscale segmentation approach for remotely sensed imagery," *IEEE International Geoscience and Remote Sensing Symposium*, vol. 6, pp: 3416-3419, July, 2003.

Clark, R. N., G. A. Swayze, K. E. Livo, R. F. Kokaly, S. J. Sutley, J. B. Dalton, R. R. McDougal, R. Robert, 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, pp. 5-1 to 5-44, Dec. 2003.

Cormen, T. H., C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd Ed., Cambridge, MA: the MIT press, pp: 168-177, 273-301, 2001.

Cox, I. J., S. B. Rao, and Y. Zhong, "Ratio regions: a technique for image segmentation," in *Proceedings of the International Conference on Pattern Recognition*, vol. B, pp. 557-564, 1996.

Darbon, J. and M. Sigelle, "Image restoration with discrete constrained total variation part I: fast and exact optimization," *Journal of Mathematical Imaging and Vision*, vol. 26, no. 3, pp. 277-291, Dec., 2006.

Davis, C. O., *Hyperspectral Imaging of the littoral battle space*, naval research laboratory report, Washington DC, 2000.
Available online at http://rsd-www.nrl.navy.mil/7212/pdf/7212_overview.pdf

Descombes, X. and E. Zhizhina, *Image denoising using stochastic differential equations*, INRIA report no. 4814, 30 p., May 2003. Available online at http://hal.inria.fr/inria-00071772/en/.

Di Zenzo, S. "A note on the Gradient of a Multi-image," *Computer Vision*, *Graphics*, *and Image Processing*, vol. 33, no. 1, pp. 116-125, Jan. 1986.

Donoho, D. and I. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, pp. 425-455, 1994.

Duarte-Carvajalino, J. M., M. Vélez-Reyes, and P. Castillo, "Scale-space in hyperspectral image analysis," *Proceedings of the SPIE,* vol. 6233, pp. 334-345, 2006.

Duarte-Carvajalino, J. M., P. Castillo, and M. Vélez-Reyes, "Comparative study of semi-implicit schemes for anisotropic diffusion in hyperspectral imagery," *IEEE Transactions on Image processing*, vol. 16, no. 5, pp. 1303-1314, May 2007.

Duarte-Carvajalino, J. M., G. Sapiro, M. Vélez-Reyes, and P. Castillo "Fast multiscale regularization and segmentation of hyperspectral imagery via anisotropic diffusion and algebraic multigrid solvers," *SPIE Defense and Security Symposium* (Orlando, Fl.) vol. 6565(12), March, 2007.

Duarte-Carvajalino, J. M., G. Sapiro, M. Vélez-Reyes, and P. Castillo, "Multiscale representation and segmentation of hyperspectral imagery using geometric partial differential equations and algebraic multigrid methods" approved for publication on *IEEE Transactions on Geoscience and Remote Sensing*, 2007.
Preprint available at http://www.ima.umn.edu/preprints/jun2007/jun2007.html

Durand, S and J. Froment, “Reconstruction of wavelet coefficients using total variation minimization,” *SIAM Journal on Scientific Computation*, vol. 24, no. 5, pp. 1754-1767, 2003.

Faber, R. L. and G. Naber, “Differential Geometry and Relativity Theory: An Introduction,” *American Journal of Physics*, vol. 54, no. 7, pp. 669-670, 1986.

Filho, A. G., A. C. Frery , C. Coêlho de Araújo, H. Alice, J. Cerqueira, J. A. Loureiro, M. Eusebio de Lima , M. G. S. Oliveira, and M. M. Horta, "Hyperspectral images clustering on reconfigurable hardware using the k-means algorithm," *16th IEEE Symp. Integrated Circutis and Systems Design*, pp. 99-104, 2003.

Galli, L. and D. de Candia, “Multispectral image segmentation via iterated weighted aggregation method,” *Proceedings of the SPIE*, vol. 5982, pp. 74-81, 2005.

Galun, M., E. Sharon, R. Basri, and A. Brandt, “Texture segmentation by multiscale aggregation of filter responses and shape elements,” *IEEE International Conference in Computer Vision*, pp. 716-723, 2003.

Geman, S. and D. Geman, “Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 6, pp. 721–741, Nov. 1984.

Girouard, G., A.Bannari, A. El Harti, and A. Desrochers, “Validated Spectral Angle Mapper Algorithm for Geological Mapping: Comparative Study between Quickbird and Landsat-TM,” in *20th ISPRS congress: Geo-Imagery Bridging Continents*, Istanbul-Turkey, 2004, pp. 599-604.

Golub, G. H. and C. F. Van Loan, *Matrix Computations*, Baltimore, MD, USA: The John Hopkins University Press, 1996.

Goodman, J. A., M. Vélez-Reyes, S. Hunt and R. Armstrong, “Development of a field test environment for the validation of coastal remote sensing algorithms: Enrique Reef, Puerto Rico,” *Proceedings of the SPIE*, vol. 6360, September, 2006.

Gorte, B., *Probabilistic segmentation of remotely sensed images*, Enschede: International Institute for Aerospace Survey and Earth Sciences (ITC), 1998.

Hamza, A. B. and H. Krim, "Image denoising: A nonlinear robust statistical approach," *IEEE Transactions on Signal Processing*, vol. 49, pp. 3045–3054, December, 2001.

Hay, G. J., T. Blaschke, D. J. Marceau, and A. Bouchard, "A comparison of three image-object methods, for the multiscale analysis of landscape structure," *Journal of Photogrametry and Remote Sensing*, vol 57, pp. 327-345, April, 2003.

Honeycutt, T. L. and D. R. Wright, "Building bridges: connecting research ethics and computer science," *Proceedings of the 24th ACM annual conference on design of communication*, pp. 1-2, October, 2006.

Horn, R. A. and C. R. Johnson, *Matrix Analysis*, New York, NY: Cambridge University Press, 2006.

Katartzis, A., I. Vanhamel, and H. Sahli, "A Hierarchical markovian model for multiscale region-based classification of vector-valued images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 548-558, March, 2005.

Keaton, T. and J. Brokish, "A level-set method for the extraction of roads from multispectral imagery," *Proceedings of the 31st Applied Imagery Pattern Recognition workshop from color to hyperspectral: advancements in spectral imagery exploitation*, pp. 141-147, 2002.

Kettig, R. L. and D. A. Landgrebe, "Classification of Multispectral Image Data by Extraction and Classification of Homogeneous Objects," *IEEE Transactions on Geoscience Electronics*, vol. 14, no. 1, pp. 19-26, January, 1976.

Kimmel, R., R. Malladi, and N. Sochen, "Images as embedded maps and minimal surfaces: movies, color, texture, and volumetric medical images," *International Journal on Computer Vision*, vol. 39, no. 2, pp. 111-129, September, 2000.

Kimmel, R. and I. Yavneh, "An algebraic multigrid approach for image analysis," *SIAM Journal on Scientific Computing*, vol. 24 no. 4, pp.1218-1231, 2003.

Landgrebe, D. A., "Hyperspectral image data analysis," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 17-28, January, 2002.

Landgrebe, D.A., *Signal Theory Methods in Multispectral Remote Sensing*, Hoboken, NJ: John Wiley and Sons, 2003.

Lennon, M., G. Mercier, and L. Hubert-Moy, "Nonlinear filtering of hyperspectral images with anisotropic diffusion," in *IEEE International Geoscience and Remote Sensing Symposium*, vol. 4, pp. 2477-2479, 2002.

Lennon, M., G. Mercier, and L. Hubert-Moy, "Classification of hyperspectral images with nonlinear filtering and support vector machines," in *IEEE International Geoscience and Remote Sensing Symposium*, vol. 3, pp. 1670-1672, 2002.

Linde, Y., A. Buzo, and R. M. Gray, "An Algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. com-28, no. 1, January, 1980.

Lindeberg, T., *Discrete scale-space theory and the scale-space primal sketch*, Ph.D. thesis, Royal Institute of Technology, Stockholm, Sweden, May, 1991.

Long, S., "Multigrid Methods for Anisotropic Diffusion," *Proceedings of the Australian Pattern Recognition Workshop on Digital Image Computing*, *Medical Imaging*, pp. 175-179, 2005. Available online at http://www.aprs.org.au/wdic2005/papers/35.pdf.

Martín-Herrero, J., "Anisotropic Diffusion in the Hypercube," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 5, pp. 1386-1398, May, 2007.

Mrázek, P. and M. Navara, "Selection of optimal stopping time for nonlinear diffusion filtering," *International Journal on Computer Vision*, vol. 52, no.2-3, pp. 189-203, May-June 2003.

Mohan, A., G. Sapiro, and E. Bosch, "Spatially Coherent Nonlinear Dimensionality Reduction and Segmentation of Hyperspectral Images," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 2, pp. 206-210, April, 2007.

Neher, R. and A. Srivastava, "A Bayesian MRF framework for labeling terrain using hyperspectral imaging," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 6, pp. 1363-1374, June, 2005.

Nasrabadi, N. M., and R. A. King, "Image coding using vector quantization: a review," *IEEE Transactions on Communications*, vol. 36, no. 8, August, 1988,

Navulur, K., *Multispectral Image Analysis Using the Object-Oriented Paradigm*, Boca Raton, FL: CRC Press, 2007.

Nordström, K. Niklas, "Biased anisotropic diffusion: a unified regularization and diffusion approach to edge detection," *Image and Vision Computing*, vol. 8, no. 4, pp. 318-327, November, 1990.

Novach, E., *Deterministic and stochastic error bounds in numerical analysis*, in Lectures notes in Mathematics, vol. 1349, 113 p., Springer-Verlag, 1988.

Novach, E. and K. Ritter, "The curse of dimension and a universal method for numerical integration," in *Multivariate Approximation and Splines*, G. Nürnberger, J. W. Schmidt, G. Walz eds., International Series of Numerical Mathematics, pp. 177-188, 1997.

Othman, H. and S.-E. Qian, "Noise reduction of hyperspectral imagery using hybrid spatial-spectral derivative-domain wavelet shrinkage," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 2, pp. 397-408, Feb., 2006.

Paclík, P., R. P. W. Duina, G. M. P. van Kempenb, and R. Kohlus, "Segmentation of multi-spectral images using the combined classifier approach," Image and Vision Computing, vol. 21, pp. 473–482, Jan. 2003.

Plaza, A., D. Valencia, and J. Plaza, "High-Performance computing in remotely sensed hyperspectral imaging: the pixel purity index algorithm as a case study," *IEEE Parallel and Distributed Processing Symposium*, April, 2006.

Plaza, A. J., and C.-I. Chang, *High Performance Computing in Remote Sensing*, Boca Raton: FL, CRC Computing and Information Science Series, Chapman and Hall, 2007.

Perona, P. and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, July 1990.

Rand, R. S. and D. M. Keenan, "Spatially smooth partitioning of hyperspectral imagery using spectral/spatial measures of disparity," *IEEE Transactions on. Geoscience and Remote Sensing*, vol. 41, no. 6, pp. 1479-1490, June, 2003.

Rubner, Y., C. Tomasi, and L. J. Guibas, "The Earth Mover's Distance as a Metric for Image Retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

Saad, Y., *Iterative methods for sparse linear matrices*, 2nd ed. Philadelphia, PA: SIAM, pp. 300-305, 2003.

Sapiro, G. and Ringach, D.L., "Anisotropic diffusion of multivalued images with applications to color filtering," *IEEE Transactions on Image Processing*, vol. 5, no.11, pp. 1582 - 1586, November, 1996.

Sapiro, G., *Geometric Partial Differential Equations and Image Analysis*, Cambridge, UK: Cambridge University Press, 2001.

Sharon, E., A. Brandt, and R. Basri, "Fast multiscale image segmentation," in *Proceedings of the IEEE Conference in Computer Vision and Pattern Recognition*, vol. 1, pp. 70-77, 2000.

Shi, J. and J. Malik, "Normalized Cuts and Image Segmentation," *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*: 731-737, Puerto Rico, 1997.

Scherzer, O. and J. Weickert, "Relations between regularization and diffusion filtering," *Journal of Mathematical Imaging and Vision*, vol. 12, no. 1, pp. 43-63, February, 2000.

Shewchuk, J. R., *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*, Report, School of Computer Science, Carnegie Mellon University, August 1994. Available online at http://www.cs.cmu.edu/~jrs/jrspapers.html.

Shi, J. and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, August, 2000.

Silverman, J., S. R. Rotman, K. L. Duseau, P. W. Yip, and B. Bukhel, "Refining the Histogram-based segmentation of hyperspectral data," in *Proceedings of the SPIE*, vol. 5546, pp. 334-343, 2004.

Strikwerda, J. C., *Finite difference schemes and partial differential equations*, Philadelphia, PA: SIAM, pp. 145-163, 349, 377-398, 2004.

Sweet, J. N., "The spectral similarity scale and its application to the classification of hyperspectral remote sensing data," *IEEE workshop on Advances in Techniques for Analysis of Remotely Sensed Data*, pp. 92-99, October, 2003.

Tang, B., G. Sapiro, and V. Caselles, "Diffusion of general data on non-flat manifolds via harmonic maps theory: the direction diffusion case," *International Journal of Computer Vision*, vol. 36, no. 2, pp: 149-161, February, 2000.

Tschumperle, D. 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, April 2005.

Unal, G., Krim, H., and Yezzi, A., "Stochastic differential equations and geometric flows," *IEEE Transactions on Image Processing*, vol. 11, no. 2, pp. 1405-1416, 2002.

Van de Vlag, D. E. and A. Stein, "Incorporating uncertainty via hierarchical classification using fuzzy decision trees," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 1, pp. 237-245, Jan., 2007.

Vélez-Reyes, M. and L. O. Jimenez, "Subset selection analysis for the reduction of hyperspectral imagery," in *IEEE International Geoscience and Remote Sensing Symposium*, vol. 3, pp. 1577 – 1581, 1998.

Vélez-Reyes, M., L. O. Jimenez, D. M. Linares, and H.T. Velasquez, "Comparison of matrix factorization algorithms for band selection in hyperspectral imagery," in *Proceedings of the SPIE*, vol. 4049, pp. 288 – 297, 2000.

Vélez-Reyes, and D. M. Linares, "Comparison of Principal-Component-Based Band Selection Methods for Hyperspectral Imagery," in *Proceedings of the SPIE*, vol. 4541, pp. 361 – 369, 2002.

Voci, F. , S. Eiho, N. Sugimoto, and H. Sekiguchi, "Estimating the gradient threshold in the Perona-Malik equation," *IEEE Signal Processing Magazine*, vol. 21, no. 3, pp. 39-65, May 2004.

Weickert, J., *Anisotropic diffusion in image processing*, PhD dissertation, University of Kaiserslautern, Germany, 1996.

Weickert, J. and B. Benhamouda "A semidiscrete nonlinear scale-space theory and its relation to the Perona–Malik paradox," in *Advances in Computer Vision*, Springer, Wien, Germany, pp. 1–10, 1997.

Weickert, J., "A review of nonlinear diffusion filtering," in *Scale-Space Theory in Computer Vision*, Lecture Notes in Computer Science, Kluwer Academic Publishers, Stockholm, Sweeden, vol. 1255, pp. 3–28, 1997.

Weickert, J., *Anisotropic Diffusion in Image Processing*, Teubner-Verlag, Stuttgart, Germany: ECMI Series, 1998.

Weickert, J., B. M. ter Haar 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, March 1998.

Weickert, J. and T. Brox, "Diffusion and regularization of vector- and matrix-valued images," *Contemporary Mathematics*, vol. 313, pp. 251-268, 2002.

Wright, D. R., "Research ethics and computer science: an unconsummated marriage", *Proceedings of the 24th ACM annual conference on Design and Communication*, pp. 196-201, 2006.

Wright, D. R., *Motivation, Design, and Ubiquity: A Discussion of Research Ethics and Computer Science*, Essay, North Carolina State University. Available online at http://arxiv.org/abs/0706.0484

Zhang, Y. J., “A review of recent evaluation methods for image segmentation,” *International Symposium on Signal Processing and its Applications*, pp. 148-151, 2001.