MULTISCALE REPRESENTATION AND SEGMENTATION OF HYPERSPECTRAL IMAGERY USING GEOMETRIC PARTIAL DIFFERENTIAL EQUATIONS AND ALGEBRAIC MULTIGRID METHODS

By

Julio M. Duarte-Carvajalino

Guillermo Sapiro

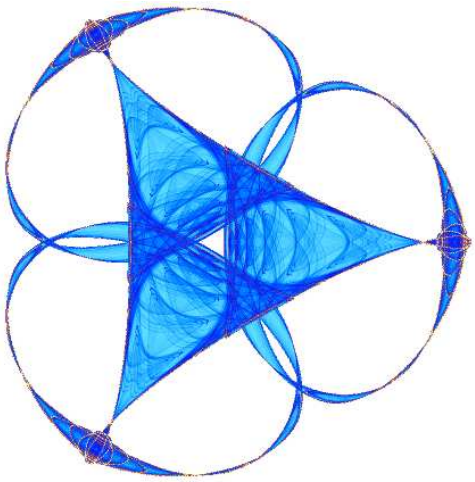
Minguel Vélez-Reyes

and

Paul E. Castillo

IMA Preprint Series # 2167

(June 2007)



INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA

400 Lind Hall
207 Church Street S.E.
Minneapolis, Minnesota 55455–0436

Phone: 612-624-6066 Fax: 612-626-7370
URL: http://www.ima.umn.edu

Multiscale Representation and Segmentation of Hyperspectral Imagery using Geometric Partial Differential Equations and Algebraic Multigrid Methods

Julio M. Duarte-Carvajalino, *Student Member, IEEE*, Guillermo Sapiro, *Senior Member, IEEE*, Miguel Vélez-Reyes, *Senior Member, IEEE*, and Paul E. Castillo

***Abstract*—A fast algorithm for multiscale representation and segmentation of hyperspectral imagery is introduced in this paper. The multiscale/scale-space representation is obtained by solving a nonlinear diffusion Partial Differential Equation (PDE) for vector-valued images. We use Algebraic Multigrid (AMG) techniques to obtain a fast and scalable solution of the PDE and to segment the hyperspectral image following the intrinsic multigrid structure. We test our algorithm on four standard hyperspectral images that represent different environments commonly found in remote sensing applications: agricultural, urban, mining, and marine. The experimental results show that the segmented images lead to better classification than using the original data directly, in spite of the use of simple similarity metrics and piecewise constant approximations obtained from the segmentation maps.**

***Index Terms*—Multigrid, multiscale, geometric partial differential equations, segmentation, hyperspectral images.**

I. Introduction

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. Scale-space theory (multiscale analysis) aims to obtain this structure within a formal and well-sounded 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 multi and hyperspectral imagery, in part due to the high dimensionality of the data and heterogeneity (spatial and spectral) of remote sensed images.

This paper introduces a fast and scalable algorithm for multiscale representation and segmentation of hyperspectral imagery. The scale-space representation of the image is obtained as the solution of a vector-valued nonlinear diffusion Partial Differential Equation (PDE), with the image as its initial condition. We use Algebraic Multigrid (AMG) methods to solve the nonlinear diffusion PDE, with good accuracy and scalability. AMG also provides a multiscale representation of the image that enables its subsequent multiscale segmentation. In addition to the presentation of the new framework, we evaluate the performance of AMG as a solver of the nonlinear diffusion equation and the quality of the segmentation obtained. For this purpose, we use four hyperspectral images, the Indian Pines test site (NW Indiana), the Cuprite mining district (Nevada), the Enrique Cay image (SW Puerto Rico) taken with the AVIRIS and the Washington DC mall area taken with the HYDICE sensor. These images represent very different image environments often found in remote sensing: agricultural, mining, marine, and urban.

The main contribution of this work consists in the introduction of AMG to solve the vector-valued nonlinear diffusion PDE and from there to naturally obtain a multiscale segmentation of hyperspectral imagery. To the best of our knowledge, this is the first time that AMG is used to obtain a multiscale representation and segmentation of hyperspectral imagery.

The remainder of this paper is organized as follows. Section II defines in general terms the notation used here, Section III presents a brief review of the state of the art on nonlinear diffusion of hyperspectral imagery, Algebraic Multigrid, and segmentation of hyperspectral imagery, with emphasis on AMG-based segmentation. Section IV introduces the algorithm for nonlinear diffusion PDE and segmentation of hyperspectral imagery. Section V presents implementation details of the algorithm and complexity analysis. Section VI presents the performance tests and segmentation results using the four hyperspectral images mentioned above. The conclusions of this work are presented in Section VII.

II. Notation

We use lower case bold (as in **u**) for vectors (such as a hyperspectral pixel), and uppercase bold (as in **U**) for matrices (such as the whole hyperspectral image). The individual elements of a matrix are noted with the same letter as the

M. Vélez-Reyes (Miguel.Velez@ece.uprm.edu) and J. M. Duarte-Carvajalino (jmartin@ece.uprm.edu) are with the Laboratory of Applied Remote Sensing and Image Processing (LARSIP), UPRM, Mayaguez, PR 00681-9048, USA.

G. Sapiro (guille@umn.edu) is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455-0436 USA.

P. Castillo (castillo@math.uprm.edu) is with the Department of Mathematical Sciences, UPRM, Puerto Rico, Mayaguez 00681-9018 USA.

matrix, but in lowercase cursive with the row and column indices as subscripts in cursive, for instance, g_{ij} is an element of matrix **G**. Matrix **G** can also be represented in terms of its elements as $\mathbf{G} = [g]_{ij}$.

All variables and parameters are in lowercase cursive (as variable indices *i*, *j*, parameters α, μ) some parameters that are considered as constant such as the image dimensions (*M*, *N*) and fixed labels are in uppercase cursive (*S* representing the coarsest grid). The sets are always represented here in uppercase, and the usual set operations between sets A an B are the union $A \cup B$, the intersection $A \cap B$ and the set difference, $A \backslash B$. The set elements are considered as variables, hence, we say $i \in V$.

Finally, subscripts and superscripts follows the same notation indicated here. Superscripts are used here mainly to indicate a grid, within the multigrid structure, for instance V^s means the set of vertices V at grid *s*.

III. Background and State of the Art

A. Nonlinear Diffusion for Hyperspectral imagery

The classical nonlinear diffusion PDE for vector-valued images is given by [1, 2],

$$\frac{\partial \mathbf{u}_i(\mathbf{x},t)}{\partial t} = \nabla \bullet \left(g(\theta) \nabla \mathbf{u}_i(\mathbf{x},t) \right), \quad i = 1, \cdots, M, \qquad (1)$$

where, *g* is the diffusion function (coefficient), $\mathbf{U} = [\mathbf{u}_1\ \mathbf{u}_2\ \ldots\ \mathbf{u}_M]$ is the *M*x*N* matrix representation of an *M*-band image with *N* vector valued pixels, *t* is the scale parameter and θ is a measure of the vector-valued edge strength, which is given [1, 3, 4] as,

$$\theta = \sqrt{\frac{1}{M} \sum_{j=1}^{M} \left| \nabla u_{\sigma,j}(\mathbf{x},t) \right|^2}. \qquad (2)$$

where, u_σ is a smoothed version of *u* obtained by convolving *u* with a Gaussian of standard deviation σ [1]. For simplicity, from now on, we will omit the subscript σ on *u* in θ. By a convenient selection of the diffusion coefficient in (1), the intensity of the image is allowed to diffuse within the image structures, reducing the intra-object variability, while preventing diffusion across the edges, characterized by a high vector gradient.

Alvarez et al. [5] showed that all scale-spaces satisfying natural physical principles are governed by parabolic Partial Differential Equations (PDEs), as is the case of (1). The scale-space generated by a parabolic PDE can be seen as a continuous transformation of the image into a space of progressively "smoother" images, identified by the parameter or scale *t*. Adequate selection of the scale reduces nuisance variability in the image, facilitating the extraction of homogeneous regions, i.e., segmenting the image.

The explicit discretization of (1) is given by [3, 4, 6, 7],

$$\mathbf{U}_{n+1} = \left(\mathbf{I} + \mu \mathbf{G}_n \right) \mathbf{U}_n, \qquad (3)$$

where, $\mu = \Delta t / (\Delta x \Delta y)$, being Δt the discretization of the scale, Δx and Δy the discretization of the spatial coordinates, **I** the identity matrix, **G** is the matrix of diffusion coefficients, and *n* is shortcut notation for the discretized scale, $n\Delta t$. Lennon et al. [8, 9] use (3) to smooth multispectral imagery showing that classification accuracy increases after the nonlinear smoothing. However, Equation (3) is limited to small scale steps, $\mu \leq ¼$, due to numerical stability, which constitutes a serious limitation for most practical scales and large data sets, as is the case of hyperspectral imagery.

Another option to discretize (1) is to use semi-implicit schemes [10], which are numerically stable for all values of μ. The semi-implicit discretization of (1) is given by [3, 4, 6, 7],

$$\left(\mathbf{I} - \mu \mathbf{G}_n \right) \mathbf{U}_{n+1} = \mathbf{U}_n. \qquad (4)$$

However, the numerical stability of (4) comes at a price, we have to solve a linear system of equations at each iteration step and the accuracy of the computed solution decreases as μ increase. As Duarte et al. [3, 4] showed, semi-implicit schemes such as Alternating Direction Implicit (ADI) and Additive Operator Splitting (AOS) schemes can solve the nonlinear diffusion in hyperspectral imagery 20 times faster than using explicit schemes. ADI and AOS schemes provide an approximation to the exact solution of (4) that can be obtained in linear time complexity. They also showed that (4) can be solved with higher accuracy using Preconditioned Conjugated Gradient (PCG) methods. Even tough the preconditioners analyzed did not scale well; they obtained the best classification accuracies with this method. Thereby, solving (4) with higher accuracy might be worth the effort. The solution presented here is fast and accurate, and naturally leads to multiscale segmentation as discussed later.

Regarding *g*, several diffusion coefficients have been proposed in the past for the nonlinear diffusion PDE [2]. In our experience, the diffusion coefficient proposed by Weickert [7] produces segmentation-like images, and it is given by,

$$g(\theta) = \begin{cases} 1 & \theta = 0 \\ 1 - e^{-\frac{3.31488}{(|\nabla u|/\alpha)^8}} & \theta > 0 \end{cases}, \qquad (5)$$

where, 3.31488 is the value that makes the flux $g(\theta)\theta$ increasing for $\theta \in [0, \alpha]$ and decreasing for $\theta \in (\alpha, \infty)$, α is a threshold parameter that controls the amount of diffusion in terms of θ, which in general can be viewed as a similarity metric, for vector valued images. As shown in [3, 4], the discretization of θ corresponds to the Euclidean distance between spectral vectors. On the other hand, we can use a different similarity metric for vector-valued images like the spectral angle [11] (see also [36]). From now on, whenever we refer to θ as the Euclidean distance, we are referring to the discrete version of the vector valued gradient as defined in (2).

B. Algebraic Multigrid Methods

Multigrid methods [12] 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. Multigrid methods aim to reduce the error at all frequencies, in linear

time complexity. Multigrid includes two complementary processes: relaxation and coarse-grid correction. Coarse-grid correction involves transferring information from a fine to a coarse grid 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 finer 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 [12]. The relaxation can be accomplished by a simple iterative method such as Jacobi or Gauss-Seidel.

The method used to coarsening 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 a matrix, which in our case is the diffusion matrix **G**. 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 [13]. As Kimmel and Yavneh [14] had shown, algebraic multigrid is more robust for image analysis using geometric PDEs.

We then use an algebraic multigrid (AMG) method to solve the semi-implicit Equation (4), i.e. the nonlinear diffusion PDE on hyperspectral imagery. The complete description of our AMG algorithm is presented in detail on Section IV.

C. Segmentation of Hyperspectral Imagery

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) [15]. The object-oriented approach consists in generating a scale-space representation of the image based on similarity metrics and hierarchical clustering. The practical application of object-based segmentation has been limited so far to high spatial resolution images from IKONOS2 (4 bands) and multispectral images from LANDSAT (7 bands)[1]. In addition to the object-oriented approach, other algorithms have been proposed in the past for high spatial resolution multispectral imagery, based on level sets [16], Markov Random Fields [17], and histograms-based segmentation [18], to name just a few.

Here, we use a modified version of a fast segmentation algorithm for grayscale images proposed by E. Sharon, et al. [19], inspired by AMG and normalized cuts, the later a segmentation algorithm proposed by Cox et al. [20] and improved later by Shi and Malik [21]. Recently, an extension of Sharon's segmentation algorithm has also been proposed for multispectral imagery [22].

The segmentation algorithm in [19] is based on hierarchical clustering, rather than on the formal scale-space representation of the image using geometric PDEs. 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.

The segmentation problem can be cast into the problem of graph partitioning. 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 weight function that indicates the degree of similarity between the vertices. The segmentation problem can be expressed now as finding the optimal graph cut that minimizes the weight of the edges removed [21]. The optimal graph cut is in general an NP-hard problem [21] and hence, fast suboptimal solutions are used. The contribution of Sharon et al. [19] consists into 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 finer level. In this way, they achieved image segmentation in linear time complexity, and it is often better than those obtained with (single-scale) normalized cuts [19]. 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 finer grid [23].

IV. AMG-Based Scale-Space Representation And Segmentation of Hyperspectral Imagery

AMG requires the construction of a multigrid structure that starts with the finer grid of the original image on its base and coarser grids are added "below it" forming an inverted pyramid, as shown on Figure 1.

We use standard graph theory notation (V^s, E^s) to identify the set 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 finer grid and $s = S$ for the coarsest one. 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 (5).

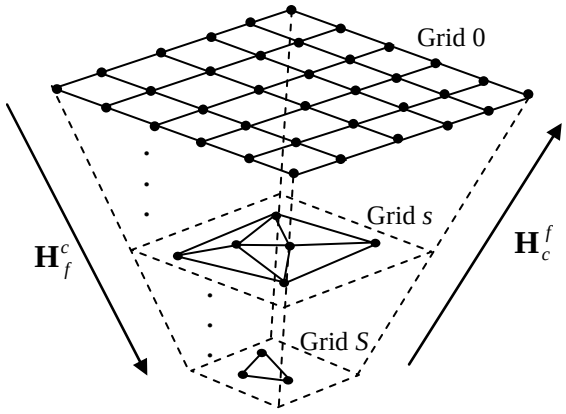


Figure 1 Typical Multigrid structure. Note that the structure is not necessary of a Cartesian grid.

[1] See http://www.definiens.com/documents/publications_earth.php

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 1. Associated to the graph, there is a similarity function g that assigns a weight to each edge $(i, j) \in \mathrm{E}^s$ on each grid with $0 \le s \le S$, being S the coarsest grid. The nonlinear diffusion coefficient, given by (4), corresponds to the similarity function g at the finest grid, $s = 0$.

A. Multigrid Structure

The construction of the multigrid structure requires two main steps: selection of the next set of vertices V^{s+1} from the current grid $(\mathrm{V}^s, \mathrm{E}^s)$, $0 \le s \le S$-1, and the connection of 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 [19], since it satisfies these requirements. For completeness, we describe it here in detail.

The mechanism used to select which vertices from $(\mathrm{V}^s, \mathrm{E}^s)$ 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. The mass is a measure of how many pixels in the finer grid can be assigned to a given vertex on a coarse grid. At grid s=0 the mass of all vertices is defined as $m_i^0 = 1$. The idea of sorting the vertices is that vertices that are representative of a large number of pixels on the finer grid; would be more likely selected for the next grid. The selection process consists of the following three steps,

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

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

where, $0 < \tau < 1$ is a threshold value below which we say that vertex i is independent of the vertices selected, 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 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 introducing first some important comments on the sorting algorithm.

The sorting algorithm must run in linear time to keep the overall complexity of the algorithm also linear. The sorting algorithm used in [19] is a bucket sort [24], 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 [24], 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 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.

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

$$\forall i \in \mathrm{V}^s \backslash \mathrm{V}^{s+1}, j \in \mathrm{V}^{s+1} : w_{ij}^s = w_{ji}^s = \frac{g_{ij}^s}{\sum_{(i,k) \in \mathrm{E}^s} g_{ik}^s}, \qquad (6)$$

$$\forall i \in \mathrm{V}^{s+1} : m_i^{s+1} = m_i^s + \sum_{j \in \mathrm{V}^{s+1} \backslash \mathrm{V}^s} w_{ij}^s, \qquad (7)$$

where, w_{ij}^s indicates how much vertex $i \in \mathrm{V}^s \backslash \mathrm{V}^{s+1}$ depends on vertex $j \in \mathrm{V}^{s+1}$. Notice that (6) naturally 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 $\mathrm{V}^s \backslash \mathrm{V}^{s+1}$ 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 [23]. However, given the little development of texture measures for hyperspectral imagery and the difficulty of obtaining second and higher order statistics for hyperspectral data from small, fuzzy segments, we only use first order statistics, i.e. mean intensities.

Let the hyperspectral image at grid s be $\mathbf{U}^s = \begin{bmatrix} \mathbf{u}_1^s & \mathbf{u}_2^s & \cdots & \mathbf{u}_{\nu_s}^s \end{bmatrix}^{\mathrm{T}}$, where ν_s is the number of vertices at grid s and $\mathbf{u}_i$ is the spectral vector at pixel i^{th}. The mean spectral intensity at grid s+1 is given by

$$\forall i \in \mathrm{V}^{s+1} : \mathbf{u}_i^{s+1} = \frac{\mathbf{u}_i^s + \sum_{j \in \mathrm{V}^s \backslash \mathrm{V}^{s+1}} w_{ij}^s \mathbf{u}_j^s}{1 + \sum_{j \in \mathrm{V}^s \backslash \mathrm{V}^{s+1}} w_{ij}^s}. \qquad (8)$$

Notice that (8) corresponds to the weighted mean vector-valued intensity, where the spectral signatures of vertices $j \in \mathrm{V}^s \backslash \mathrm{V}^{s+1}$ influenced by pixel $i \in \mathrm{V}^{s+1}$ are weighted according to their dependence on i. Notice also that (8) defines the restriction operation (coarsening of the pyramid), $\mathbf{H}_f^c$, which in matrix format is given by

$$\mathbf{U}^{s+1} = \mathbf{H}_f^c \mathbf{U}^s, \quad \left[h_f^c\right]_{ij} = \frac{w_{ij}}{\sum_{j \in \mathrm{V}^s \backslash \mathrm{V}^{s+1} \cup \{i\}} w_{ij}^s}, \qquad (9)$$

where, we had defined, for convenience, $w_{ii}^s = 1$, since it

provides a more compact representation for $\mathbf{H}_f^c$.

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 [12], $\mathbf{G}^{s+1} = \mathbf{H}_f^c \mathbf{G}^s \mathbf{H}_c^f$, so 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,

$$\forall i \in \mathrm{V}^s : \mathbf{u}_i^s = \begin{cases} \mathbf{u}_j^{s+1}, \text{if } i \in \mathrm{V}^{s+1} \\ \sum_{j \in \mathrm{V}^{s+1}} w_{ij}^s \mathbf{u}_j^s, \text{if } i \notin \mathrm{V}^{s+1} \end{cases} \quad (10)$$

which, in matrix-vector notation is given by,

$$\mathbf{U}^s = \mathbf{H}_c^f \mathbf{U}^{s+1}, \qquad \left[h_c^f\right]_{ij} = \left[w_{ij}^s\right], \quad (11)$$

From (9) and (10),

$$\mathbf{G}^{s+1} = \left[g^{s+1}\right]_{kl} = \frac{1}{\sum_{j \in \mathrm{V}^s \setminus \mathrm{V}^{s+1} \cup \{i\}} w_{ij}^s} \sum_{k,l \in \mathrm{V}^s} w_{ik}^s g_{kl}^s w_{lj}^s. \quad (12)$$

Sharon et al. [19] proposed a very similar equation called Iterated Weighted Aggregation (IWA) to connect the vertices on the coarse grid. However, while IWA was proposed as an approximation to $\mathbf{G}^{s+1}$ within a minimization problem, Equation (12) corresponds exactly to $\mathbf{G}^{s+1}$, as given by the Garlekin condition, in our AMG setup. It can be noticed that as in IWA, (12) only considers local measures accumulated from grid 0 up to the coarser grids.

As in [19], we introduce a global measure to steer the segmentation processes, which depends on the mean spectrums computed for each coarse vertex,

$$g_{kl}^{s+1} = \left(\frac{1}{\sum_{j \in \mathrm{V}^s \setminus \mathrm{V}^{s+1} \cup \{i\}} w_{ij}^s} \sum_{k,l \in \mathrm{V}^s} w_{ik}^s g_{kl}^s w_{lj}^s \right) e^{-\alpha\theta\left(\mathbf{u}_k^{s+1}, \mathbf{u}_l^{s+1}\right)}, \quad (13)$$

where, θ is in general a similarity metric that depends on vectors $\mathbf{u}_k^{s+1}, \mathbf{u}_l^{s+1}$ and corresponds to the discretization of θ as defined on Section III.

Experimentally (see Section VI), we found that using (13) improves the rate of convergence of AMG over (12). This result demostrates 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 relationship between the vertices are not completely expressed by local measures, and 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,

$$\mathrm{E}^{s+1} = \left\{ (i,j) : i,j \in \mathrm{V}^{s+1} \wedge g_{ij}^s > 0 \right\}. \quad (14)$$

B. AMG Solver

Figure 2 shows a schematic of the same multigrid structure presented in Figure 1, showing more clearly the V-cycle. As Figure 2 shows, the V-cycle consists in coarsening the image downto grid *S*, solves exactly for the error, and then propagates back the solution to the finer grid. In the following, we call $\mathbf{X}^0$ the approximation to the exact solution we are looking for (see Equation 3), $\mathbf{U}_{n+1}$, and $\mathbf{X}^s$ the approximation to the error at grids $s > 0$. We omit subscript *n* on $\mathbf{G}_n^s$ and call it simply $\mathbf{G}^s$.

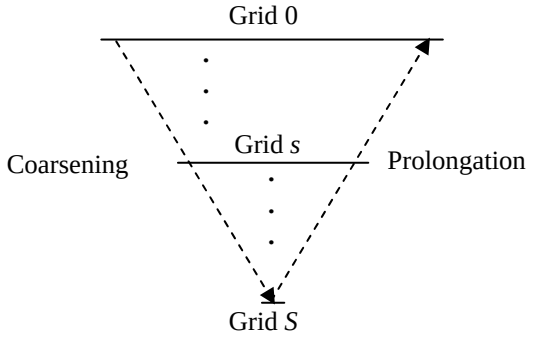


Figure 2 Schematics for a V-cycle in Multigrid.

The V-cycle algorithm is given by,

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

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

    …
    - *Grid S: Solve exactly* $\left(\mathbf{I} - \mu\mathbf{G}^S\right)\mathbf{X}^S = \mathbf{F}^S$ *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* $\left(\mathbf{I} - \mu\mathbf{G}^s\right)\mathbf{X}^s = \mathbf{F}^s$, *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* $\left(\mathbf{I} - \mu\mathbf{G}^0\right)\mathbf{X}^0 - \mathbf{U}_n$ *with initial guess* $\mathbf{X}^0$ *to obtain* $\mathbf{X}^0 \approx \mathbf{U}_{n+1}$.

The V-cycle algorithm can be divided in three phases. In the coarsening phase [12] (Figure 2), 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. 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 better approximate the error. After a V-cycle, $\mathbf{X}^0$ receives the accumulated error from previous grids and the initial estimate of $\mathbf{U}_{n+1}$ can be corrected as $\mathbf{X}^0=\mathbf{X}^0+\mathbf{H}_c^f\mathbf{X}^1\approx\mathbf{U}_{n+1}$. Usually, a single V-cycle is not enough to achieve good accuracy, and a few extra V-cycles might be needed. However, the extra V-cycles are computationally faster than the first one, since they use the AMG structure constructed on the first V-cycle.

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 VI for an analysis of their complexity). The relaxation method chosen here is Gauss-Seidel (GS) [25]. We achieved the best rates of convergence for AMG using an implementation that on the finest grid corresponds to a Symmetric-Red-Black GS, 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 we 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 can say that grid S would have ~ $\log_2 N$ vertices, where N is the number of vertices at grid 0 (the original picture size). Hence, we decided to stop coarsening the grid, when the number of vertices is equal or less than $\log_2 N$.

Notice that on Equation (4) 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 (Section VI) of AMG, we can use large values of μ, which means that few steps would suffice for most applications and the overall complexity remains scalable algorithmically.

C. Segmentation Algorithm

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 is to solve 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 explained before, but we now use (3) or

$$\forall i,j\in \mathrm{V}^0 : g_{ij}=e^{-\beta\theta\left(\mathbf{u}_i^0,\mathbf{u}_j^0\right)}, \tag{15}$$

which is the similarity metric proposed by [19] extended to hyperspectral imagery. We can also change the parameter α by a parameter γ on the coarsening Equation (13).

The saliency Γ of a vertex is determined as in [19], for s=0, …, S-1 as,

$$\forall i\in \mathrm{V}^s : \Gamma_i=\sum_{j\in \mathrm{V}^s} g_{ij}^s \Big/ m_i^s . \tag{16}$$

Equation (16) 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 [21], but on coarse scales. We define a vertex as a salient segment if its saliency is $\Gamma_i \le \varepsilon$, 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 finest resolution. This process is called sharpening in [19], 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 [19] 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 [26], 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, …, r_K the K representatives accumulated from levels s, $s+1$,…, S, and let $\mathbf{p}_i=\left(p_i^{r_1},\cdots,p_i^{r_K}\right)$ be a vector of probabilities indicating that vertex $i\in \mathrm{V}^0$ has probabilities $p_i^{r_1},\cdots,p_i^{r_K}$ of belonging to the segments represented by r_1,…,r_K, 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 $\mathbf{p}_i=\left(0,\cdots,0\right)$ for $i\notin\{r_1,\ldots,r_K\}$. 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:

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

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

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

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} *as defined in (13) 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* must be strongly dependant on vertices from grids *s*+1 to *S*, hence, w_{ir_k} might be high for some representative r_k. However, vertices that were chosen from *s* to the next grids and are not representatives have $w_{ir_k} = 0$ for *k* = 1,.., *K*, since both *i* and r_k are in V^{s+1}. Nevertheless, their neighbors that are in $V^s \backslash V^{s+1}$ might have been labeled in this step. Hence, the next step is the same as in [19], we perform υ Gauss-Seidel relaxations allowing than the probabilities of the neighbors affect the probabilities of each vertex, based now on their similarities. As noted in [19], υ=2 GS relaxations suffices, since a higher number of relaxations does not produce any change 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, in terms of similarity. Also as in [19], probabilities higher than a given threshold, 1-δ, are set to one, in order to speedup the sharpening process.

V. Implementation Details and Complexity

Most of the algorithm's parameters are set by experimentation, see for e.g. [19, 22, 23, 27]. In particular we use τ = 0.2, δ = 0.2, $\varepsilon = 10^{-5}$, the number of GS relaxations for the sharpening algorithm is υ=2, and the number of GS relaxations in AMG is simply $\upsilon_0 = \upsilon_1 = \ldots = \upsilon_S = 1$. Experimentally (see next section), we also found that two V-cycles suffice to achieve good accuracy for scale-steps μ ≤ 5, which corresponds to 20 times the maximum stable scale-step that can be used with the explicit scheme. The remaining parameters α, β, and γ depends 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 ≤ α ≤ 0.015, γ ≤ β ≤ α, 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 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 introduce some additional changes in the implementation in order to improve the running time and scalability of the algorithm for hyperspectral imagery. In particular, we made the following changes,

- Sharon's algorithm, [19], 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, [24], 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 [28]. 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, reducing considerably the running time of the algorithm, without affecting negatively the accuracy of the AMG solver or the segmentation algorithm.
- In [19], 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 Figure 4. The segmentation results are the same, but with an improvement in running time.
- The vector of probabilities indicated on Figure 4 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 were implemented in C++ on a Linux platform, under the Cygwin[2] environment. We used the Geospatial Data Abstraction Library (GDAL[3]) 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 and segmentation results. We also used LAPACK[4] 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 (4) can be always found using Gaussian elimination, since the matrix **I**-μ**G** is diagonally dominant [29]. 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 for the coarsest scale, where the number of vertices is $\nu_s = O(\log N)$. We used Gaussian elimination on the finest grid, but just for comparison purposes and independently of the AMG framework here proposed (see Section VI).

From the previous section, it can be seen that for both the AMG and the segmentation algorithm, there are just a few number of operations for each vertex on Equations (4)-(14). As indicated before, the sorting algorithm runs in linear time

[2] http://www.cygwin.com/
[3] http://www.gdal.org/
[4] http://www.netlib.org/lapack/

on the number of vertices at each grid. Also, and since a fixed number of relaxation sweeps are made on each level, relaxation is linear in the number of vertices, at each grid. The matrix operations indicated on Figure 4 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 (vertices) at scale s. Hence, the product $(\mathbf{I}\text{-}\mu\mathbf{G}^s)\mathbf{X}^s$ takes $\sim 10M\nu_s = O(M\nu_s)$ time. Also, the exact solution of (3), 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 hyperspectral bands,

$$\sum_s \kappa M \nu_s \leq \kappa M \sum_s \left(\frac{1}{2}\right)^s N \leq 2\kappa MN = O(MN) \qquad (17)$$

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

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

$$\sum_s 2M\nu_s \leq 2M \sum_s \left(\frac{1}{2}\right)^s N \leq 4MN. \qquad (18)$$

Hence, the disk requirements are $\sim 4MN$, with additional temporal variables of size $O(N)$. For sufficiently large M, as is the case of hyperspectral imagery, the disk requirements are dominated by the $\sim 4MN$ 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 [4] for details). Since, AMG is scalable and can have greater accuracy 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. It should be also noticed here that even tough the storage requirements of AMG can be excessive to process large hyperspectral images in a PC; proper processing of these images requires parallel processing capabilities and storage far beyond the capability of a single PC, where the extra storage needed by AMG can be satisfied, making this approach practical for large datasets.

VI. Experiments

We use four hyperspectral images in our experiments, representing different environments in remote sensing,

- Indian Pines image (Figure 3a), taken with the AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) sensor, on June 12, 1992, over an agricultural area 6 miles west of West Lafayette. This image contains 145x145 pixels and 220 spectral bands in the 400-2500 nm range, for which ground truth exists[5] (see Figure 4a). We disregard 35 bands from the original image, either because they were too noisy or because they present strong illumination differences. Hence, our Indian Pines image has 145x145 pixels and 185 spectral bands in the 410-2430 nm range.
- Cuprite image (Figure 3b), taken over a mining district, 2 km north of Cuprite, Nevada, with the AVIRIS sensor, flown on June 19, 1997. We selected a portion of the image of size 500x500 pixels and 224 bands that corresponds to part of the mineral mapping reported by the US Geological Survey (USGS) spectroscopy laboratory in 1995, using the expert system algorithm Tetracorder [30] and signatures of 60 sampled fields in the region[6]. We use the classification map reported by the USGS as our ground truth (see Figure 4b). We selected from this image 50 bands, 172-221, that corresponds to the 2000-2480 nm vibrational absorption region used by the USGS for mapping minerals in the Cuprite image.
- 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. This image contains 1280x307 pixels and 224 bands. Several bands are eliminated because they do not contain information or they are too noisy, leaving 191 bands in the 400-2480 nm range. We choose a sub-image of 282x307 pixels and 191 bands (Figure 5a) 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, [31], [32] directly marked on Figure 5a.
- The Enrique Reef image (Figure 5b), that corresponds to a small part of the AVIRIS image taken over the south-west coast of Puerto Rico in 2005. We use this image because the Enrique Reef environment is a well-known area of study for the marine science department at the UPRM. 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 5b. We eliminated spectral bands in this image that contained just noise, so that our Enrique reef image consists of 46x90 pixels and 146 bands in the 414-2310 nm range.

Our first set of experiments consist into test the performance of AMG as a solver of (4) 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 [3, 4].

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

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

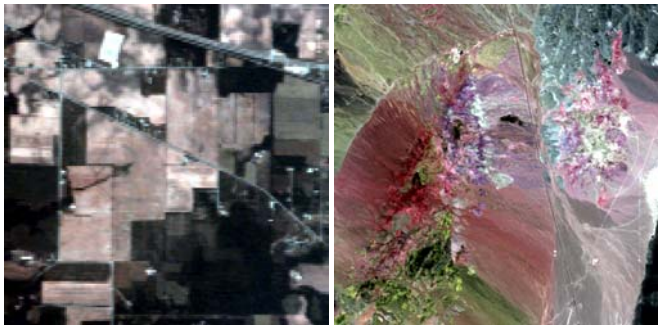


Figure 3 RGB composite of a) Indian Pines and b) Cuprite images.

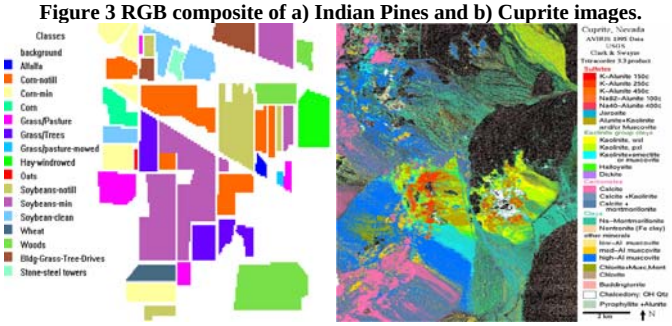


Figure 4 Ground truth of a) Indian Pines and b) Cuprite images.

Figure 5 RGB composite of a) Washington DC and b) Enrique Reef images, showing also their ground truth.

A. Performance of the AMG as a Solver

We compute first the sum of square errors between the computed solution of (4) using AMG and the solution obtained using LAPACK at the finest grid, with α=0.015 in (4), which is the largest value of α we had used in these and previous experiments [3, 4]. We test AMG using local measures only, i.e. Equation (12), and incorporating the mean spectrum, i.e. Equation (13), where θ can be either the Euclidean distance or the spectral angle.

Given that Gauss-Seidel elimination for banded matrices, as is the case of **G** on the finer grid, requires to store vectors of size $O(N^{3/2})$, it easily overcomes the memory available on an average PC. In particular, we could not obtain the solution of (4) using LAPACK on the finer grid, for the Cuprite image, given its size. Hence, we use instead the first 300x300 pixels (50 bands) of the Cuprite image, which we call here, small Cuprite.

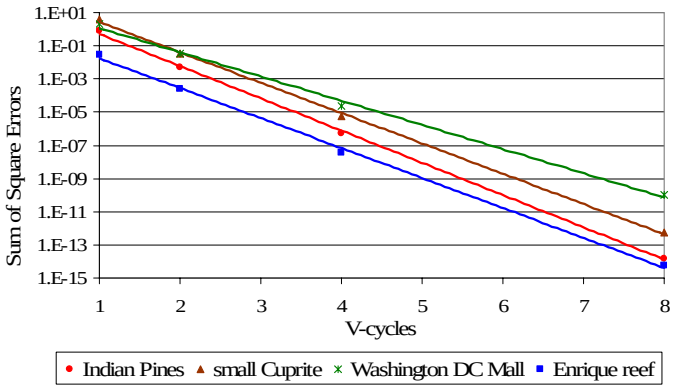


Figure 6 Performance of AMG vs. the number of V-cycles.

First, Figure 6 shows, in semi-logarithmic scale, the sum of square errors as a function of the number of V-cycles, using (13) and θ being the spectral angle. The error reduces at a rate of $r = 10^{-d}$, where d is the slope of the line shown on Figure 6. From this figure, the rate-of-convergence range is r = 0.013-0.032. This is quite good, since it compare well with the reported convergence rates for well-tuned AMG algorithms (r~0.05) [12], [14]. The rates of convergence for α<0.015 should 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 (see Figure 7) with a tolerance of 10^{-3}, which is the same tolerance used in [3, 4], with very good results in terms of the solution of the geometric PDE and classification accuracy.

Table 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 1 compares the rates of convergence of AMG using only accumulated local measures, i.e. Equation (12), and the rates of convergence of AMG using local measures and Euclidean distance (ED) or spectral angle (SA) between mean spectrums, i.e. Equation (13). It can be noticed that introducing simple global measures as the mean spectral intensity reduces the error at least two times faster than using accumulated local measures only. It can be also noticed that the spectral angle converges faster than using Euclidean distances, tough the comparison may be not completely fair, since defining θ as the spectral angle means changing the PDE, which is not longer the exact Equation (1).

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 wide 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 proposed AMG for these images are still quite good, and there is no need for

using more precise, but computationally expensive, non-uniform sampling methods such as those indicated in [28].

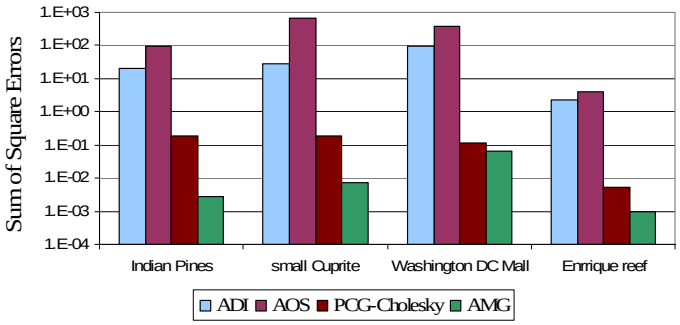


Figure 7 Performance of AMG vs. other solvers

Figure 7 compares the sum of square errors relative to the exact solution obtained with LAPACK on the finer grid, for four methods: ADI and AOS schemes, the Conjugated Gradient method, preconditioned with incomplete Cholesky factorization (PCG-Cholesky), 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 [3, 4].

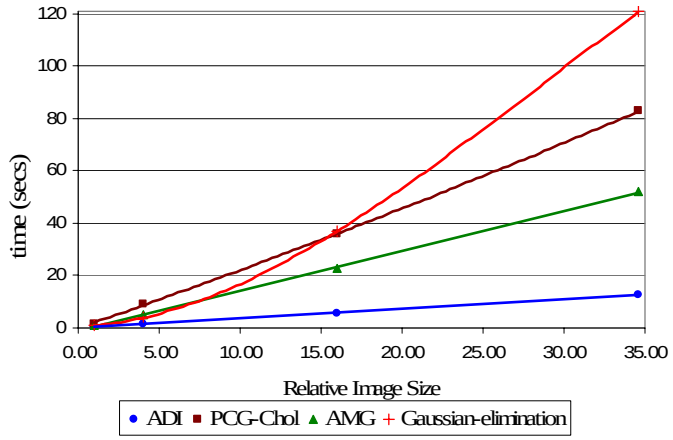


Figure 8 CPU time vs. the size of the image

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 50x50, 100x100, 200x200, and 282x307 pixels from the Washington DC image, with all its 191 bands. Figure 8 shows the CPU time of AMG for a scale-step of $\mu = 5$, $\alpha=0.015$ and 2 V-cycles, vs. the size of the image, relative to the smallest image, i.e. the image of 50x50 pixels.

Figure 8 shows, for comparison purposes, the CPU time required to solve (4) for ADI, PCG-Cholesky, and Gaussian elimination. From this figure, we can say that our implementation of AMG is four 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 increasing the convergence rates by a factor of 2 (which are still good, see [14]). However, we prefer here to trade speed for accuracy of the computed solution for the nonlinear diffusion PDE, since this also affects classification accuracy, maintaining nevertheless a reasonable computational cost.

Figure 9 shows the CPU time of AMG as a solver of (3) and the total CPU time of AMG to both solve (4) and segment the hyperspectral imagery, as a function of the image size. From this figure it is clear that solving (4) with AMG and segmenting the images has linear time complexity, and that the segmentation step takes approximately a 25% of the total smoothing and segmentation time.

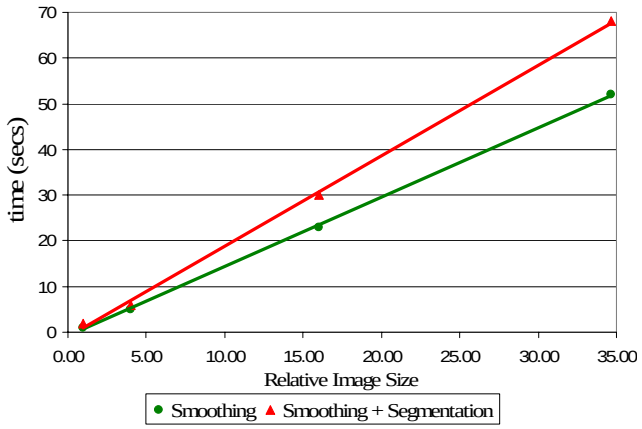


Figure 9 CPU time for AMG smoothing and segmentation

B. 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 quality of classification algorithms, since small regions provide less statistical information than larger ones. 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 classification, instead of synthetic test images as often required by current methods that measure the quality of segmented images [33].

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 and testing samples on each one of the four hyperspectral images considered here (see Figures 10 and 11). The labels on Figures 10 and 11 indicates if the fields correspond to training (tr) or testing (te) samples and the numbers indicated are used simply to distinguish them. We choose ECHO, [34], spectral-spatial as our classifier, provided by MultiSpec[7], freeware software developed by D. A. Landgrebe.

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

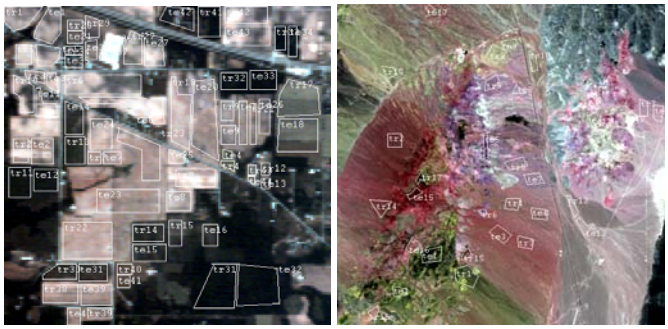


Figure 10 Training and testing samples on a) Indian Pines and b) Cuprite images

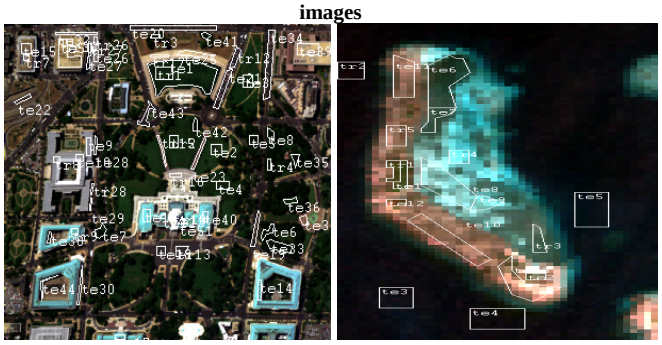


Figure 11 Training and testing samples on a) Washington DC mall and b) Enrique reef images

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 2x2 pixels that uses Fisher Linear Discriminant. ECHO clusters the segments into different classes, according to their distance (in terms of the Fisher) and according to the homogeneity of the neighborhood, computing likelihoods, whenever possible.

Table 2 Best classification accuracies

Classification Accuracy (%)	Indian Pines		Cuprite		Washington DC		Enrique reef	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Original	90.8	71.8	98.8	93.5	100	86.2	100	93.7*
Smoothed	99.9	81.3	99.9	95.2	100	87.8	100	95.6
Segmented	97.4	85.6	99.9	95.8	100	85.5	100	96.8
Smoothed and Segmented ED	95.1	88.8	99.7	96.8	100	91.2	100	98.1
Smoothed and Segmented SA	98.6	**90.2**	99.7	**97.5**	100	**92.2**	100	**98.5**

Table 2 shows the best classification accuracies obtained by simply smoothing with AMG, segmenting only with the AMG-based segmentation algorithm, and smoothing first and then segmentation using AMG. Table 2 also includes the classification accuracies of the original image, for comparison purposes. We could not obtain a classification of the original Enrique reef image using ECHO, probably due to the noise present in the image. Hence, the accuracy reported in Table 2 for this image 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 already 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 performed, 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.

We must emphasize here that we are classifying the piece-wise spectrally-constant hyperspectral images with the sole purpose of testing the quality of the segmentation. A more advanced 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 reported segmentation maps.

Figures 12 and 13 show RGB composites of the smoothed hyperspectral images that produced the best classification accuracies, as indicated in Table 2.

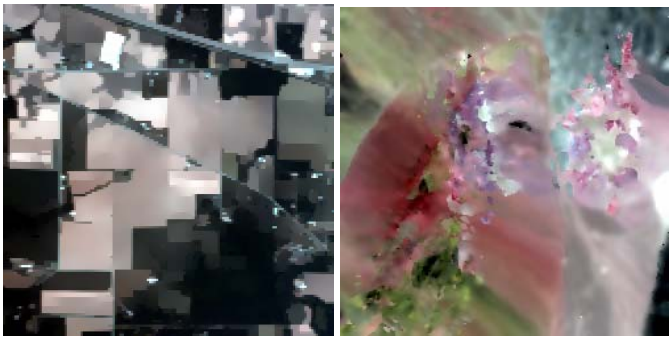


Figure 12 Smoothed a) Indian Pines and b) Cuprite images with AMG.

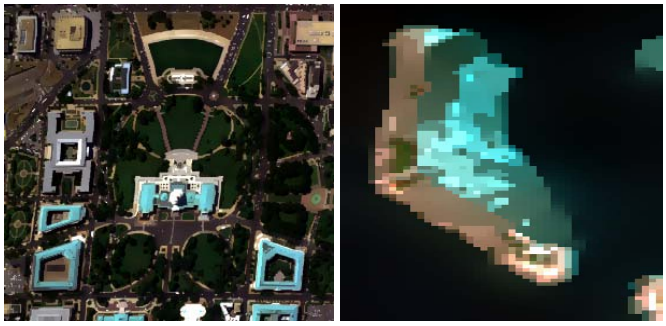


Figure 13 Smoothed a) Washington DC and b) Enrique reef with AMG.

Figures 14 and 15 show the RGB composites of the segmented hyperspectral images that resulted on the best classification accuracies, indicated in Table 2. It should be noticed here that with exception of the Indian Pines image, the segmented images shown in Figures 14 and 15 were obtained using smoothed images with a different α value than those in Figures 12 and 13. The α value 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.

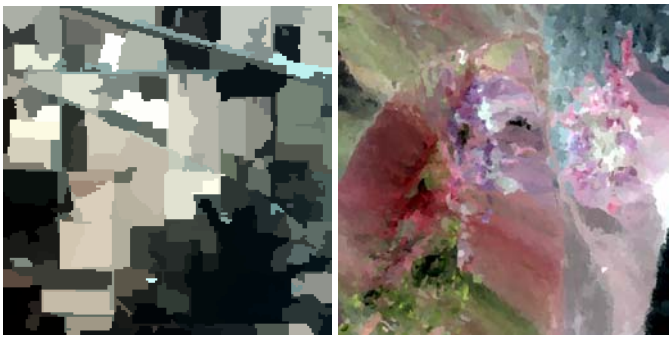


Figure 14 Segmented a) Indian Pines and b) Cuprite images with AMG.

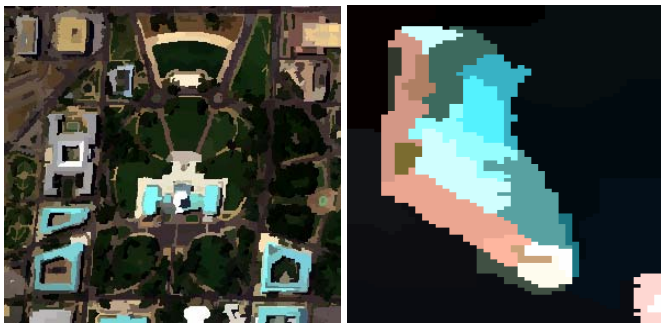


Figure 15 Segmented a) Washington DC and b) Enrique reef with AMG.

Table 3 shows the parameters corresponding to the results indicated in Table 2. It can be noticed that the value of α for the best classification accuracies using only smoothing differs slightly from the value of α that produces the best accuracies using both smoothing and segmentation. Also, it can be noticed from Table 1 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.

Table 3 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

Finally, and for completeness, we should mention that the scale used to nonlinearly smooth all the 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.

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 [3] for more detail in this regard). 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 noise (even after eliminating noisy bands), which can be appreciated on the visible variability of the seawater in Figure 5b. 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 3, which shows a relatively low range of variability.

VII. Conclusion

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 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, probably noisy, image.

We should note that a number of techniques are currently being developed for the fast computation of geometric PDEs, see for example [35] and references there in. The extension of those approaches for hyperspectral data, as well as the use of our proposed framework for generically solving such PDEs, is the subject of current efforts in our group.

Acknowledgment

This work made use of Engineering Research Centers Shared Facilities supported by the National Science Foundation under Award Number EEC-9986821. Mr. Duarte was also supported by a NSF-EPSCOR fellowship, from the PR EPSCOR program. Guillermo Sapiro is supported by the National Geospatial-Intelligence Agency (NGA), as well as NSF, ONR, and DARPA. Miguel Velez-Reyes was also supported by NGA and DoD. The authors wish to thank Dr. D. A. Landgrebe for making some of the hyperspectral data used in this paper and MultiSpec publicly available, and Dr. Edward Bosch from the NGA for introducing our groups at UPRM and UMN, leading to the collaboration that resulted in the work reported here.

References

[1] J. Weickert and T. Brox, “Diffusion and regularization of vector- and matrix-valued images,” in *Inverse Problems, Image Analysis, and Medical Imaging*, vol. 313 of Contemporary Mathematics, M. Z. Nashed and O. Scherzer, Ed. AMS, Providence, 2002, pp. 251–268.

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

[3] J. M. Duarte-Carvajalino, P. Castillo, and M. Velez-Reyes, “Comparative study of semi-implicit schemes for nonlinear diffusion in hyperspectral imagery,” *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1303- 1314, May 2007.

[4] J. M. Duarte-Carvajalino, M., Vélez-Reyes, and P. Castillo, “Scale-space in hyperspectral image analysis,” in *SPIE Defense and Security Symposium*, 2006, vol. 6233, pp. 334-345.

[5] L. Alvarez, P. L. Lions, and J. M. Morel, “Image selective smoothing and edge detection by non-linear diffusion,” *SIAM J. Num. Anal.*, vol. 29, no. 1, pp. 182–193, June 1992.

[6] J. Weickert, *Anisotropic diffusion in image processing*, Ph.D. dissertation, Dept. of Mathematics, University of Kaiserslautern, Germany, January 1996.

[7] J. Weickert, B. M. ter Haar Romeny, and M. A. Viergever, “Efficient and reliable schemes for nonlinear diffusion filtering,” *IEEE Trans. Image Process*. vol. 7, no. 3, pp. 398–410, March 1998.

[8] M. Lennon, G. Mercier, and L. Hubert-Moy, "Nonlinear filtering of hyperspectral images with anisotropic diffusion," in *IEEE Intl. Geosci. and Remote Sens. Symposium*, 2002, vol. 4, pp. 2477-2479.

[9] M. Lennon, G. Mercier, and L. Hubert-Moy, "Classification of hyperspectral images with nonlinear filtering and support vector machines," in *IEEE Intl. Geosci. and Remote Sens. Symposium*, 2002, vol. 3, pp. 1670-1672.

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

[11] R.G. Resmini, "Hyperspectral/spatial detection of edges (HySPADE): an algorithm for spatial and spectral analysis of hyperspectral information", *Proc. SPIE*, vol. 5425, 2004, pp. 433-442.

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

[13] A. Brandt, 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.

[14] R. Kimmel and I. Yavneh, "An algebraic multigrid approach for image analysis," *SIAM J. Scientific Comp.*, vol. 24 no. 4, pp.1218-1231, 2003.

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

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

[17] I.B. Kerfoot and Y. Bresler, "Theoretical analysis of multispectral image segmentation criteria," *IEEE Trans. Image Process.*, vol. 8, no. 6, pp. 798-820, June, 1999.

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

[19] E. Sharon, A. Brandt, and R. Basri, "Fast multiscale image segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2000, vol. 1, pp. 70-77.

[20] I. J. Cox, S. B. Rao, and Y. Zhong, "Ratio regions: a technique for image segmentation," in *Proc. Intl. Conf. Pattern Recognition*, 1996, vol. B, pp. 557-564.

[21] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 888-905, Aug. 2000.

[22] L. Galli and D. de Candia, "Multispectral image segmentation via iterated weighted aggregation method," in *SPIE symposium on Image and Signal Process. for Remote Sens.*, 2005, vol. 5982, pp. 74-81.

[23] M. Galun, E. Sharon, R. Basri and A. Brandt, "Texture segmentation by multiscale aggregation of filter responses and shape elements", in *IEEE Intl. Conf. Computer Vision*, 2003, pp. 716-723.

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

[25] Y. Saad, *Iterative methods for sparse linear matrices*, 2nd ed. Philadelphia, PA: SIAM, 2003.

[26] E. Sharon, A. Brandt, and R. Basri, "Segmentation and boundary detection using multiscale intensity measurements," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2001, vol.1, pp. 469-476.

[27] A. Akselrod-Ballin, 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," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, vol. 1, pp. 1122-1129.

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

[29] A. George and K. D. Ikramov, "Gaussian elimination is stable for the inverse of a diagonally dominant matrix," *Mathematics of Computation*, vol. 73, no. 246, pp. 653-657, 2004.

[30] R. N. Clark , et al., "Imaging spectroscopy: earth and planetary remote sensing with the USGS Tetracorder and expert systems", *J. Geophysical Research*, vol. 108, no. E12, pp. 5-1 to 5-44, Dec. 2003.

[31] J. E. Ball, and L. M. Bruce, "Level set segmentation of remotely sensed hyperspectral images," in *Proc. of the IEEE Intl. Geosc. and Remote Sensing Symp*., 2005, vol. 8, pp. 5638- 5642.

[32] D. Landgrebe, "Hyperspectral image data analysis," *IEEE Signal Process. Mag*., vol. 19, no. 1, pp. 17-28, Jan., 2002.

[33] Y. J. Zhang, "A review of recent evaluation methods for image segmentation," in *Intl. Symp.on Signal Processing and its Applications*, 2001, pp. 148-151.

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

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

[36] A. Castrodad, E. H. Bosch, and R. Resmini, "Hyperspectral imagery transformations using real and imaginary features for improved classification," *SPIE Defense and Security Symposium 2007*, vol 6565, 2007.