Chapter XIII

Parallel Segmentation of Multi-Channel Images Using Multi-Dimensional Mathematical Morphology

Antonio Plaza, University of Extremadura, Spain

Javier Plaza, University of Extremadura, Spain

David Valencia, University of Extremadura, Spain

Pablo Martinez, University of Extremadura, Spain

ABSTRACT

Multichannel images are characteristic of certain applications, such as medical imaging or remotely sensed data analysis. Mathematical morphology-based segmentation of multichannel imagery has not been fully accomplished yet, mainly due to the lack of vector-based strategies to extend classic morphological operations to multidimensional imagery. For instance, the most important morphological approach for image segmentation is the watershed transformation, a hybrid of seeded region growing and edge detection. In this chapter, we describe a vector-preserving framework to extend morphological operations to multichannel images, and further propose a fully automatic multichannel watershed segmentation algorithm that naturally combines spatial and spectral/temporal information. Due to the large data volumes often associated with multichannel imaging, this chapter also develops a parallel implementation strategy to speed up performance. The proposed parallel algorithm is evaluated using magnetic resonance images and remotely sensed hyperspectral scenes collected by the NASA Jet Propulsion Laboratory Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS).

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

INTRODUCTION

The segmentation of an image can be defined as its partition into different regions, each having certain properties (Zhang, 1996). In mathematical terms, a segmentation of an image f is a partition of its definition domain D_f into m disjoint, non-empty sets $s_1,s_2,...,s_m$ called segments, so that $\bigcup_{i=1}^{m} S_i = D_f$ and $S_i \cap S_j = \varnothing, \forall i \neq j$. Segmentation of intensity images in the spatial domain usually involves four main approaches (Haralick & Shapiro, 1985). Thresholding techniques assume that all pixels whose value lies within a certain range belong to the same class. Boundary-based methods assume that the pixel values change rapidly at the boundary between two regions. Region-based segmentation algorithms postulate that neighboring pixels within the same region have similar intensity values, of which the split-and-merge technique is probably the most well known. Hybrid methods combine one or more of the above-mentioned criteria. This class includes variable-order surface fitting and active contour methods.

One of the most successful hybrid segmentation approaches is the morphological watershed transformation (Beucher, 1994), which consists of a combination of seeded region growing (Adams & Bischof, 1994; Mehnert & Jackway, 1997) and edge detection. It relies on a marker-controlled approach (Fan, et al., 2001) that considers the image data as imaginary topographic relief; the brighter the intensity, the higher the corresponding elevation. Let us assume that a drop of water falls on such a topographic surface. The drop will flow down along the steepest slope path until it reaches a minimum. The set of points of the surface whose steepest slope path reachs a given minimum constitutes the *catchment basin* associated with that minimum, while the watersheds are the zones dividing adjacent catchment basins. Another way of visualizing the watershed concept is by analogy to immersion (Vincent & Soille, 1991). Starting from every minimum, the surface is progressively flooded until water coming from two different minima meet. At this point, a watershed line is erected. The watershed transformation can successfully partition the image into meaningful regions, provided that minima corresponding to relevant image objects, along with object boundaries, are available (Shafarenko, et al., 1997). Despite its encouraging results in many applications, morphological techniques have not been fully exploited in applications that involve multichannel imagery, where a vector of values rather than a single value is associated with each pixel location.

Many types of multichannel images exist depending on the type of information collected for each pixel. For instance, color images are multichannel images with three channels, one for each primary color in the RGB space. Images optically acquired in more than one spectral or wavelength interval are called multispectral. These images are characteristic in satellite imaging and aerial reconnaissance applications. The number of spectral channels can be extremely high, as in the case of hyperspectral images produced by imaging spectrometers (Chang, 2003). Finally, all image types above can be extended to the class of multitemporal images or image sequences, which consist of series of images defined over the same definition domain, but collected at more than a single time. Examples include magnetic resonance (MR) images in medical applications and video sequences.

Segmentation of multichannel imagery has usually been accomplished in the spectral/temporal domain of the data only. Techniques include well known data clustering algorithms such as ISODATA (Richards & Jia, 1999). Other techniques, such as

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Soille's watershed-based multichannel segmentation (Soille, 1996), are based on an initial spectral clustering followed by a post-classification using spatial information. This approach separates spatial information from spectral information, and thus the two types of information are not treated simultaneously.

In this chapter, we develop a novel watershed-based segmentation technique that naturally combines spatial and spectral/temporal information in simultaneous fashion. While such an integrated approach holds great promise in several applications, it also creates new processing challenges (Tilton, 1999). In particular, the price paid for the wealth of spatial and spectral/temporal information is an enormous amount of data to be processed. For that purpose, we develop a parallel implementation of the proposed segmentation algorithm that allows processing of high-dimensional images quickly enough for practical use. The chapter is structured as follows. The following section provides a mathematical formulation for multidimensional morphological operations, and relates the proposed framework to other existing approaches in the literature. A multi-dimensional watershed-based segmentation approach is described next, along with its parallel implementation. A quantitative segmentation evaluation comparison with regard to standard segmentation techniques is then provided, using both MR brain images and remotely sensed hyperspectral data collected by the 224-channel NASA Jet Propulsion Laboratory Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS) system. The chapter concludes with some remarks.

MULTIDIMENSIONAL MATHEMATICAL MORPHOLOGY

Our attention in this section focuses primarily on the development of a mechanism to extend basic morphological operations to multichannel imagery. In the following, we provide a mathematical formulation for classic and extended morphological operations.

Classic Morphological Operations

Following a usual notation (Soille, 2003), let us consider a grayscale image f, defined on a space E. Typically, E is the 2-D continuous space R^2 or the 2-D discrete space Z^2. The classic erosion of f by $B \subset Z^2$ is given by the following expression:

$$(f \ominus B)(x, y) = \min_{(s,t) \in Z^2(B)} \{f(x+s, y+t)\}, \qquad (x, y) \in Z^2 \qquad (1)$$

where $Z^2(B)$ denotes the set of discrete spatial coordinates associated to pixels lying within the neighborhood defined by a "flat" SE, designed by B. The term "flat" indicates that the SE is defined in the x-y plane. Similarly, the classic dilation of f by B is given by:

$$(f \oplus B)(x, y) = \max_{(s,t) \in Z^2(B)} \{f(x-s, y-t)\} \qquad (x, y) \in Z^2 \qquad (2)$$

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Vector-Based Mathematical Morphology

Let us denote a multichannel image with n channels as $\mathbf{f}$, where the values of each pixel $\mathbf{f}(x,y)$ of the definition domain D_f are given by an n-dimensional vector $\mathbf{f}(x,y)=(f_1(x,y), f_1(x,y),...,f_n(x,y))$. Extension of monochannel erosion and dilation above the images defined on the n-dimensional space is not straightforward. A simple approach consists in applying monochannel techniques to each image channel separately, an approach usually referred to as "marginal" morphology in the literature. This approach is unacceptable in many applications because, when morphological techniques are applied independently to each image channel, there is a possibility for loss or corruption of information of the image due to the probable fact that new pixel vectors —not present in the original image— may be created as a result of processing the channels separately. In addition, no correlation between spectral/temporal components is taken into account. An alternative way to approach the problem of multidimensional morphology is to treat the data at each pixel as a vector. Unfortunately, there is no unambiguous means of defining the minimum and maximum values between two vectors of more than one dimension, and thus it is important to define an appropriate arrangement of vectors in the selected vector space.

Several vector-ordering schemes have been discussed in the literature (Plaza, et al., 2004). Four classes of ordering will be shortly outlined here for illustrative purposes. Let us now consider an n-dimensional image $\mathbf{f}$ and let $\mathbf{f}(x,y)$ and $\mathbf{f}(x',y')$ denote two pixel vectors at spatial locations (x,y) and (x',y') respectively, with $\mathbf{f}(x,y)=(f_1(x,y), ...,f_n(x,y))$ and $\mathbf{f}(x',y')=f_1(x',y'),...,f_n(x',y'))$. In marginal ordering (M-ordering), each pair of observations $f_i(x,y)$ and $f_i(x',y')$ would be ordered independently along each of the n channels. In reduced ordering (R-ordering), a scalar parameter function g would be computed for each pixel of the image, and the ordering would be performed according to the resulting scalar values. The ordered vectors would satisfy the relationship $\mathbf{f}(x,y) \le \mathbf{f}(x',y') \Rightarrow g[\mathbf{f}(x,y)] \le g[\mathbf{f}(x',y')]$. In partial ordering (P-ordering), the input multivariate samples would be partitioned into smaller groups, which would then be ordered. In conditional ordering (C-ordering), the pixel vectors would be initially ordered according to the ordered values of one of their components, for example, the first component, $f_1(x,y)$ and $f_1(x',y')$. As a second step, vectors with the same value for the first component would be ordered according to the ordered values of another component, e.g., the second component, $f_2(x,y)$ and $f_2(x',y')$, and so on.

In this chapter, we adopt a new distance-based vector ordering technique (D-ordering), where each pixel vector is ordered according to its distance from other neighboring pixel vectors in the data (Plaza, et al., 2002). This type of ordering, which can be seen as a special class of R-ordering, has demonstrated success in the definition of multichannel morphological operations in previous work. Specifically, we define a cumulative distance between one particular pixel $\mathbf{f}(x,y)$ and all the pixel vectors in the spatial neighborhood given by B (B-neighborhood) as follows:

$$C_B(\mathbf{f}(x,y)) = \sum_{(s,t)} \mathrm{Dist}(\mathbf{f}(x,y), \mathbf{f}(s,t)), \qquad \forall (s,t) \in Z^2(B), \tag{3}$$

where Dist is a linear point-wise distance measure between two N-dimensional vectors. As a result, $C_B(\mathbf{f}(x,y))$ is given by the sum of Dist scores between $\mathbf{f}(x,y)$ and every other pixel vector in the B-neighborhood. To be able to define the standard morphological

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

operators in a complete lattice framework, we need to be able to define a *supremum* and an *infimum*, given an arbitrary set of vectors $S=\{v_1,v_2,...,v_p\}$, where p is the number of vectors in the set. This can be achieved by computing $C_B(S)=C_B(\boldsymbol{v}_1),C_B(\boldsymbol{v}_2),...,C_B(\boldsymbol{v}_p)\}$ and selecting $\boldsymbol{v}_j$, such that $C_B(\boldsymbol{v}_j)$ is the minimum of $C_B(S)$, with $1\leq j\leq p$. In similar fashion, we can select $\boldsymbol{v}_k$, such that $C_B(\boldsymbol{v}_k)$ is the maximum of $C_B(S)$, with $1\leq k\leq p$. Based on the simple definitions above, the flat extended erosion of $\boldsymbol{f}$ by B consists of selecting of the B-neighborhood pixel vector that produces the minimum C_B value:

$$(\boldsymbol{f}\ominus B)(x,y)=\left\{\boldsymbol{f}(x+s',y+t'),(s',t')=\arg\min_{(s,t)\in Z^2(B)}\left\{C_B\left(\boldsymbol{f}(x+s,y+t)\right)\right\}\right\},\ (x,y)\in Z^2, \tag{4}$$

where the $\arg\min$ operator selects the pixel vector that is most similar, according to the distance Dist, to all the other pixels in the in the B-neighborhood. On other hand, the flat extended dilation of $\boldsymbol{f}$ by B selects the B-neighborhood pixel vector that produces the maximum value for C_B:

$$(\boldsymbol{f}\oplus B)(x,y)=\left\{\boldsymbol{f}(x-s',y-t'),(s',t')=\arg\max_{(s,t)\in Z^2(B)}\left\{C_B\left(\boldsymbol{f}(x-s,y-t)\right)\right\}\right\},\ (x,y)\in Z^2, \tag{5}$$

where the $\arg\max$ operator selects the pixel vector that is most different, according to Dist, from all the other pixels in the B-neighborhood. Using the above notation, the multichannel morphological gradient at pixel $\boldsymbol{f}$ (x,y) using B can be simply defined as follows:

$$G_B(\boldsymbol{f}(x,y))=Dist((\boldsymbol{f}\oplus B)(x,y),(\boldsymbol{f}\otimes B)(x,y)). \tag{6}$$

It should be noted that the proposed basic multidimensional operators are vector-preserving in the sense that vectors which are not present in the input data cannot be generated as a result of the extension process (Plaza, et al., 2002). Obviously, the choice of Dist is a key topic in the resulting multichannel ordering relation. A common choice in remote sensing applications is the spectral angle (SAD), an illumination-insensitive metric defined between two vectors $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ as follows:

$$SAD(\boldsymbol{s}_i,\boldsymbol{s}_j)=\cos^{-1}\left(\boldsymbol{s}_i\cdot\boldsymbol{s}_j/\|\boldsymbol{s}_i\|\|\boldsymbol{s}_j\|\right)=\cos^{-1}\left(\sum_{l=1}^{N}s_{il}s_{jl}\Big/\left[\sum_{l=1}^{N}s_{il}^2\right]^{1/2}\left[\sum_{l=1}^{N}s_{jl}^2\right]^{1/2}\right) \tag{7}$$

In medical imaging, for instance, illumination effects not as relevant as noise or other types of interferers. In those cases, a most common choice is the Euclidean distance (ED). In the following, we respectively adopt SAD and ED as the baseline distances for remote sensing and medical imaging experiments discussed in this chapter.

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

MULTICHANNEL WATERSHED SEGMENTATION ALGORITHM

The segmentation paradigm of our multichannel watershed segmentation algorithm consists of three stages. First, multidimensional morphological operations are used to collect a set of minima according to some measure of minima importance. Starting from the selected minima and using the multidimensional morphological gradient as a reference, a multichannel watershed transformation by flooding is then applied. Finally, watershed regions are iteratively merged, according to a similarity criterion, to obtain the final segmentation.

Minima Selection

The key of an accurate segmentation resides in the first step, that is, the selection of "markers," or minima, from which the transform is started. Following a recent work (Malpica, et al., 2003), we hierarchically order all minima according to their deepness, and then select only those above a threshold. This approach has the advantage that it provides an intuitive selection scheme controlled by a single parameter. The concept can be easily explained using the immersion simulation. The deepness of a basin would be the level the water would reach, coming in through the minimum of the basin, before the water would overflow into a neighboring basin, that is, the height from the minimum to the lowest point in the watershed line of the basin.

Deepness can be computed using morphological reconstruction applied to the multichannel gradient in equation (6). Reconstruction is a special class of morphological transformation that does not introduce discontinuities (Vincent, 1993). Given a "flat" SE of minimal size, designed by B, and the multichannel gradient $G_B(f)$ of an n-dimensional image $\boldsymbol{f}$, morphological reconstruction by erosion of $G_B(f)$ using B can be defined as follows:

$$(G_B(\boldsymbol{f})\otimes B)^t(x,y)=\bigvee_{k\geq 1}\left[\delta_B^t(G_B(\boldsymbol{f})\otimes B \mid G_B(\boldsymbol{f}))\right](x,y), \tag{8}$$

where,

$$\left[\delta_B^t(G_B(\boldsymbol{f})\otimes B \mid G_B(\boldsymbol{f}))\right](x,y)=\left[\overbrace{\delta_B\delta_B\cdots\delta_B}^{t \text{ times}}(G_B(\boldsymbol{f})\otimes B \mid G_B(\boldsymbol{f}))\right](x,y) \tag{9}$$

and,

$$\left[\delta_B(G_B(\boldsymbol{f})\otimes B \mid G_B(\boldsymbol{f}))\right](x,y)=\bigvee\{(G_B(\boldsymbol{f})\otimes B)(x,y), G_B(\boldsymbol{f}(x,y))\} \tag{10}$$

In the above operation, $G_B(\boldsymbol{f})\otimes B$ is the standard erosion of the multichannel gradient image, which acts as a "marker" image for the reconstruction, while $G_B(\boldsymbol{f})$ acts as a "mask" image. Reconstruction transformations always converge after a finite number of iterations t, that is, until the propagation of the marker image is totally impeded by the mask image. It can be proven that the morphological reconstruction $(G_B(\boldsymbol{f})\otimes B)^t$ of $G_B(\boldsymbol{f})$ from

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

$G_B(\boldsymbol{f})\otimes B$ will have a watershed transform in which the regions with deepness lower than a certain value v have been joined to the neighboring region with closer spectral properties, that is, parameter v is a minima selection threshold.

Flooding

In this section, we formalize the flooding process following a standard notation (Soille, 2003). Let the set $P=\{\boldsymbol{p}_1,\boldsymbol{p}_2,...,\boldsymbol{p}_k\}$ denote the set of k minimum pixel vectors selected after multidimensional minima selection. Similarly, let the catchment basin associated with a minimum pixel $\boldsymbol{p}_i$ be denoted by $CB(\boldsymbol{p}_i)$. The points of this catchment basin which have an altitude less than or equal to a certain deepness score d (Malpica, et al., 2003) are denoted by:

$$CB_d(\boldsymbol{p}_i)=\{\boldsymbol{f}(x,y)\in CB(\boldsymbol{p}_i)\,|\,Dist(\boldsymbol{p}_i,\boldsymbol{f}(x,y))\le d\} \quad (11)$$

We also denote by $\boldsymbol{X}_d=\bigcup_{i=1}^{k} CB_d(\boldsymbol{p}_i)$ the subset of all catchment basins which contain a pixel vector with a deepness value less than or equal to d. Finally, the set of points belonging to the regional minima of deepness d are denoted by $RMIN_d(\boldsymbol{f}(x,y))$. The catchment basins are now progressively created by simulating the flooding process. The first pixel vectors reached by water are the points of highest deepness score. These points belong to $RMIN_{pj}(\boldsymbol{f}(x,y))=\boldsymbol{X}_{D_B(\boldsymbol{p}_j)}$, where $\boldsymbol{p}_j$ is the deepest pixel in P, that is, $D_B(\boldsymbol{p}_j)$ is the minimum, with $1\le j\le k$. From now on, the water either expands the region of the catchment basin already reached by water, or starts to flood the catchment basin whose minima have a deepness equal to $D_B(\boldsymbol{p}_l)$, where $\boldsymbol{p}_l$ is the deepest pixel in the set of $P-\{\boldsymbol{p}_j\}$. This operation is repeated until $P=\varnothing$. At each iteration, there are three possible relations of inclusion between a connected component $\boldsymbol{Y}$ and $\boldsymbol{Y}\cap\boldsymbol{X}_{D_B(\boldsymbol{p}_j)}$:

(1) If $\boldsymbol{Y}\cap\boldsymbol{X}_{D_B(\boldsymbol{p}_j)}=\varnothing$ then it follows that a new minimum $\boldsymbol{Y}$ has been discovered at level $D_B(\boldsymbol{p}_l)$. In this case, the set of all minima at level $D_B(\boldsymbol{p}_l)$, that is, $RMIN_{p_l}(\boldsymbol{f}(x,y))$ will be used for defining $\boldsymbol{X}_{D_B(\boldsymbol{p}_l)}$.

(2) If $\boldsymbol{Y}\cap\boldsymbol{X}_{D_B(\boldsymbol{p}_j)}\neq\varnothing$ and is connected, then the flooded region is expanding and $\boldsymbol{Y}$ corresponds to the pixels belonging to the catchment basin associated with the minimum and having a deepness score less than or equal to $D_B(\boldsymbol{p}_l)$, that is, $\boldsymbol{Y}=CB_{D_B(\boldsymbol{p}_l)}(\boldsymbol{Y}\cap\boldsymbol{X}_{D_B(\boldsymbol{p}_j)})$.

(3) Finally, if $\boldsymbol{Y}\cap\boldsymbol{X}_{D_B(\boldsymbol{p}_j)}\neq\varnothing$ and is not connected, then the flooded regions of the catchment basins of two distinct minima at level $D_B(\boldsymbol{p}_j)$ are expanding and merged together.

Once all levels have been flooded, the set of catchment basins of a multidimensional image $\boldsymbol{f}$ is equal to the set $\boldsymbol{X}_{D_B(\boldsymbol{p}_m)}$, where $\boldsymbol{p}_m$ is the least deep pixel in P, that is, $D_B(\boldsymbol{p}_m)$ is the maximum, with $1\le m\le k$. The set of catchment basins after multidimensional watershed can be represented as a set $\{CB(\boldsymbol{p}_i)\}_{i=1}^{k}$, where each element corresponds to the catchment basin of a regional minimum of the multichannel input image $\boldsymbol{f}$. This is the final segmentation output for the algorithm. A parallel algorithm for implementing the proposed flooding simulation is described in the following section.

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Region Merging

To obtain the final segmentation, some of the regions $\{CB(\boldsymbol{p}_i)\}_{i=1}^{k}$ resulting from the watershed can be merged to reduce the number of regions (Le Moigne & Tilton, 1995). This section briefly explains the region merging criteria and method employed. First, all regions are ordered into a region adjacency graph (RAG). The RAG is an undirected graph G=(V,E), where $V = \{CB(\boldsymbol{p}_i)\}_{i=1}^{k}$ such that each region $CB(\boldsymbol{p}_i)$ is represented by a node, and $e(\boldsymbol{p}_i, \boldsymbol{p}_j) \in E$ if:

(1) $\boldsymbol{p}_i, \boldsymbol{p}_j \in V$, and;
(2) $CB(\boldsymbol{p}_i), CB(\boldsymbol{p}_j)$ are adjacent, or
(3) $Dist(\boldsymbol{p}_i, \boldsymbol{p}_j) < S$, where S is a pixel vector similarity threshold.

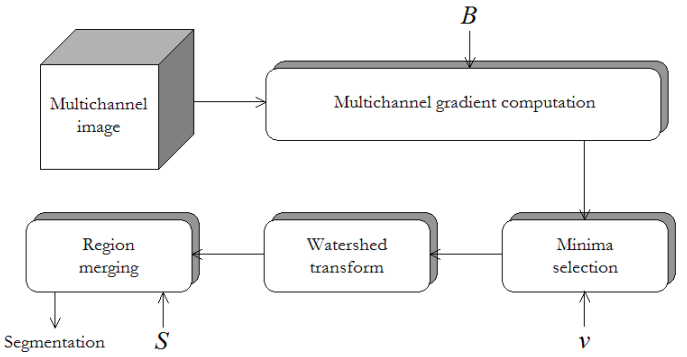
The merging process is based on graph G, where the weight of an edge $e(\boldsymbol{p}_i, \boldsymbol{p}_j)$ is the value of $Dist(\boldsymbol{p}_i, \boldsymbol{p}_j)$. Regions $CB(\boldsymbol{p}_i)$, $CB(\boldsymbol{p}_j)$ can be merged attending to spatial properties in the case of adjacent regions, and also according to pixel vector similarity criteria in the case of non-adjacent regions. Similar merging procedures have been successfully used before in the literature (Tilton, 1999). Finally, Kruskal's algorithm can be applied to generate the minimum spanning tree, denoted as *T*, by adding one edge at a time. Initially, the edges of G are sorted in a non-decreasing order of their weights. Then, the edges in the sorted list are examined one-by-one and checked to determine whether adding the edge that is currently being examined creates a cycle with the edges that were already added to *T*. If it does not, it is added to *T*; otherwise, it is discarded. It should be noted that adding $e(\boldsymbol{p}_i, \boldsymbol{p}_j)$ to *T* represents the merge of its two regions $CB(\boldsymbol{p}_i)$ and $CB(\boldsymbol{p}_j)$. On other hand, adding the edge with the minimum weight one-by-one in an increasing order to *T* using the sorted list is equivalent to the merge of the two most similar regions. Finally, when an edge is rejected because it creates a cycle in *T*, no merge is performed because its two regions have already been merged into one. The process is terminated when *T* contains *k* edges.

In order to summarize the different stages and parameters involved, Figure 1 shows a block diagram depicting our multidimensional morphological algorithm for segmentation of multichannel images.

As noted, there are three input parameters: *B*, a "flat" SE of minimal size used in the morphological operations; *v*, a minima selection threshold used in the minima selection process and *S*, a pixel vector similarity threshold used in the region merging stage. First, a multichannel gradient computation is performed by taking advantage of extended morphological operations. This step works as a multichannel edge detector. Second, minima are selected from the resulting output by using the concept of deepness. Third, flooding from markers is accomplished by utilizing the spectral angle between pixel vectors. This operation makes use of the full spectral information (as opposed to traditional watershed-based segmentation algorithms), thus avoiding the problem of band selection from the input data. Finally, the resulting segmentation is refined by a region-merging procedure that integrates the spatial and spectral information. As can be deduced from the description above, one of the main contributions of the proposed algorithm is the fact that it naturally combines spatial/spectral information in all steps. The algorithm is fully automatic, and produces a segmentation output given by a set of

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Figure 1. Block diagram summarizing the multichannel watershed segmentation algorithm



watershed regions after region merging that we will denote from now on as $\{WS_i\}_{i=1}^{m}$, with $\bigcup_{i=1}^{m} WS_i = D_f$ and $WS_i \cap WS_j \neq \varnothing, \forall i \neq j$.

PARALLEL IMPLEMENTATION

Parallelization of watershed algorithms that simulate flooding is not a straightforward task. From a computational point of view, these algorithms are representative of the class of irregular and dynamic problems (Moga & Gabbouj, 1997). Moreover, the watershed process has an extremely volatile behavior, starting with a high degree of parallelism that very rapidly diminishes to a much lower degree. In this section, our goal is to develop an efficient and scalable parallel implementation of the algorithm proposed in the previous section.

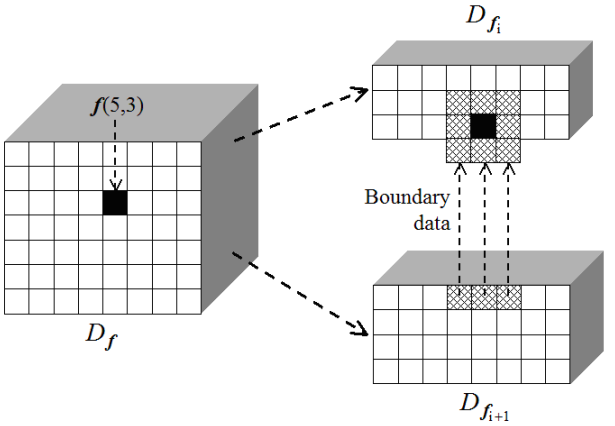
Partitioning

Two types of partitioning strategies can be applied to image segmentation problems (Seinstra, et al., 2002). One may decide whether the computation associated with the given problem should be split into pieces (functional decomposition), or the data on which the computation is applied (domain decomposition). Functional decomposition is inappropriate for segmentation with watersheds, where a sequence of operators are applied in a chain to the entire image. Our parallel algorithm uses domain decomposition, that is, the original multichannel image $\boldsymbol{f}$ is decomposed into subimages. It is important to note that the subimages are made up of entire pixel vectors, that is, a single pixel vector is never split amongst several processing elements (PEs). If the computations for each pixel vector need to originate from several PEs, then they would require intensive interprocessor message passing (Montoya, et al., 2003). Thus, the global domain D_f is split among P processors in disjoint subdomains as follows:

$$D_f = D_{f_0} \bigcup D_{f_1} \bigcup \cdots \bigcup D_{f_{P-1}}, \text{ with } D_{f_i} \cap D_{f_j} = \varnothing, \forall i \neq j \quad (12)$$

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Figure 2. Morphological structuring element computation split between two processors



Task Replication

An important issue in SE-based morphological image processing operations is that accesses to pixels outside the domain D_f of the input image is possible. For instance, when the SE is centered on a pixel located in the border of the original image, a simple border-handling strategy can be applied (Seinstra, et al., 2003). On the other hand, additional inter-processor communications may be required when the structuring element computation needs to be split amongst several different processing nodes, as shown by Figure 2. In the example, the computations for the pixel vector at spatial location (5,3) needs to originate from two processors. In order to avoid such an overhead, edge/corner pixels are replicated in the neighboring processors whose subdomains are thus enlarged with a so-called extension area. The extended subdomains are overlapping, and can be defined as follows:

$$D_{f_i}^e = \left\{ \boldsymbol{f}(x,y) \in D_f \mid N_B(\boldsymbol{f}(x,y)) \cap D_{f_i} \neq \varnothing \right\}, \tag{13}$$

where $N_B(\boldsymbol{f}(x,y))$ is the B-neighborhood of $\boldsymbol{f}(x,y)$. Using the above notation, we can further denote the neighboring subimages of $\boldsymbol{f}_i$ as the set $N_B(\boldsymbol{f}_i) = \left\{ \boldsymbol{f}_j \mid D_{f_j} \cap D_{f_i}^e \neq \varnothing \right\}$. It should be noted that the task-replication strategy above enhances code reusability, which is highly recommended in order to build a robust parallel algorithm. As will be shown by experiments, the amount of redundant information introduced by the proposed framework can greatly reduce communication time.

Implementation

Our implementation of the parallel multichannel watershed algorithm uses a simple master-slave model. The master processor divides the multichannel image $\boldsymbol{f}$ into a set of subimages $\boldsymbol{f}_i$ which are sent to different processors, so that the domain of each subimage is an extended subdomain given by $D_{f_i}^e$. The slave processors run the segmentation

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

algorithm on the respective subimages and also exchange data among themselves for uniform segmentation. After the segmented regions become stable, the slaves send the output to the master, which combines all of them in a proper way and provides the final segmentation. If we assume that the parallel system has p processors available, then one of the processors is reserved to act as the master, while each of the remaining p-1 processors create a local queue Q_i with $1 \leq i \leq p$-1. The minima selection algorithm is run locally at each processor to obtain a set of minima pixels surrounded by non-minima, which are then used to initialize each queue Q_i. Flooding is then performed locally in each processor as in the serial algorithm. However, due to the image division, flooding is confined only to the local subdomain. Therefore, there may exist parts of the subimage that cannot be reached by flooding since they are contained in other subimages. Our approach to deal with this problem is to first flood locally at every deepness score in the subimage. Once the local flooding is finished, each processor exchanges segmentation labels of pixels in the boundary with appropriate neighboring processors. Subsequently, a processor can receive segmentation labels corresponding to pixels in the extended subdomain. The processor must now "reflood" the local subdomain from those pixels, a procedure that may introduce changes in segmentation labels of the local subdomain. Communication and reflooding are again repeated until stabilization (i.e., no more changes occur). When the flood-reflood process is finished, each slave processor sends the final segmentation labels to the master processor, which combines them together and performs region merging to produce final set of segmentation labels.

To conclude this section, we emphasize that although the flooding-reflooding scheme above seems complex, we have experimentally tested that this parallelization strategy is more effective than other approaches that exchange segmentation labels without first flooding locally at every deepness score. The proposed parallel framework guarantees that the processors are not tightly synchronized (Moga & Gabbouj, 1998). In addition, the processors execute a similar amount of work at approximately the same time, thus achieving load balance. Performance data for the parallel algorithm are given in the following subsection.

EXPERIMENTAL RESULTS

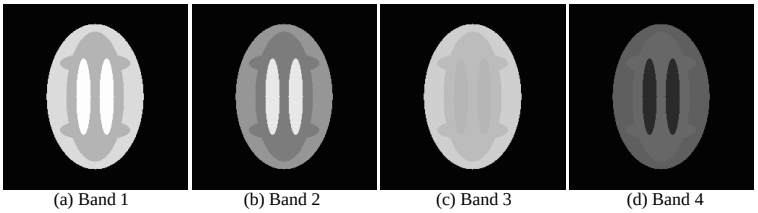
This section reports on the effectiveness of the proposed parallel segmentation algorithm in two specific applications. In the first one, phantom and real MR brain images are used to investigate the accuracy of multichannel watershed segmentation in computer-aided medical diagnoses. In the second application, hyperspectral data collected by NASA's Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS) are used to illustrate a remote sensing classification scenario.

MRI Brain Image Experiments

In medical applications, the detection and outlining of boundaries of organs and tumors in magnetic resonance imaging (MRI) are prerequisite. This is one of the most important steps in computer-aided surgery. In this section, we present two sets of experiments, one set of computer-generated phantom images and another set of real MRI images, to show that the proposed algorithm has a good capability of segmentation.

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Figure 3. Four band test phantoms for MRI simulation study



(a) Band 1 (b) Band 2 (c) Band 3 (d) Band 4

Table 1. Gray level values of the tissues of each band of the multichannel MRI phantom image.

Band	GM	WM	CSF
1	209	180	253
2	150	124	232
3	207	188	182
4	95	94	42

Computer Simulations for Phantom Experiments

In this subsection, computer simulations are used to conduct a quantitative study and performance analysis of the proposed multichannel watershed algorithm. The computer-generated phantom MRI images, shown in Figure 3, consist of four bands. The ellipses represent structural areas of three interesting cerebral tissues corresponding to gray matter (GM), white matter (WM) and cerebral spinal fluid (CSF). From the periphery to the center, the distribution of tissues is simulated as follows: background (BKG), GM, WM and CSF, given by the gray level values in Table 1. The gray level values of these areas in each band were simulated in such a fashion that these values reflect the average values of their respective tissues in real MRI images. A zero-mean Gaussian noise was added to the phantom images in Figure 3 to achieve different levels of signal-to-noise ratios (SNRs) ranging from 10 dB to 30 dB. Despite the fact that such MRI phantom images may be unrealistic, they only serve for the purpose of illustration of the proposed algorithm. This is done by using available absolute ground-truth information at a pixel level, known from the controlled simulation scenario in which the data were simulated.

In order to assess contour-fitting precision of the proposed multichannel watershed algorithm, we use the following statistical measures (Hoover, et al., 1996): correct detection, over-segmentation, under-segmentation and missed and noise region. Let D be the total number of regions detected by the algorithm, and let G be the number of ground-truth regions (four in the phantom example). Let the number of pixels in each detected region, D_i, be denoted as P_{D_i}. Similarly, let the number of pixels in each ground-truth region, G_i, be denoted as P_{G_i}. Let $0_{D_i G_i} = P_{D_i} \bigcap P_{G_i}$ be the number of overlapped pixels between P_{D_i} and P_{G_i}. Thus, if there is no overlap between P_{D_i} and P_{G_i}, then $0_{D_i G_i} = \varnothing$, while if there is complete overlap, then $0_{D_i G_i} = P_{D_i} = P_{G_i}$. Let a threshold value T be a measure of

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

the strictness of the definition desired. With the above definitions in mind, the following segmentation accuracy metrics can be defined:

(1) A pair made up of a detected region D_i and a ground-truth region G_i are classified as an instance of correct detection if $0_{D_i G_i} / P_{D_i} \geq T$, that is, at least T percent of the pixels in the detected region D_i are overlapped with the ground-truth region G_i;

(2) A ground-truth region G_i and a set of detected regions D_j, j=1,...,n are classified as an instance of under-segmentation if: $\sum_{j=1}^{n} O_{D_j G_i} / P_{G_i} \geq T$, that is, the ground-truth region G_i is at least T-percent overlapped with the composition of the n detected regions, and, $\forall j \in [1,\cdots,n], O_{D_j G_i} / P_{G_i} \geq T$, that is, all of the detected regions D_j are at least T-percent overlapped with the ground-truth region G_i;

(3) A set of ground-truth regions G_j, j=1,...,m and a detected region D_i are classified as an instance of over-segmentation if: $\sum_{j=1}^{m} O_{D_i G_j} / P_{D_i} \geq T$, that is, the detected region D_i is at least T-percent overlapped with the composition of the m ground-truth regions, and, $\forall j \in [1,\cdots,m], O_{D_i G_j} / P_{D_i} \geq T$, that is, all of the ground truth regions G_j are at least T-percent overlapped with the detected region D_i;

(4) A detected region D_i, not participating in any instance of correct detection, over-segmentation or under-segmentation, is classified as a missed region; and

(5) A ground-truth region G_j not participating in any instance of correct detection, over-segmentation or under-segmentation is classified as noise region.

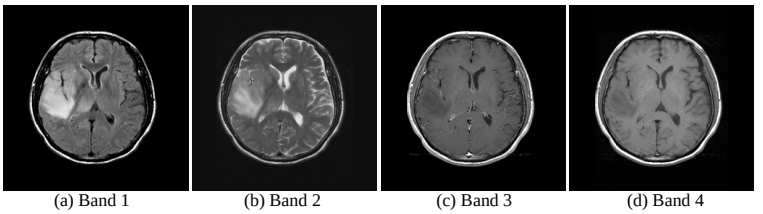
Using the five segmentation accuracy metrics above, Table 2 shows the number of correct detections, over-segmentations, under-segmentations, missed and noise regions obtained after applying the multichannel watershed algorithm to the phantom image in Figure 3, corrupted by Gaussian noise in different proportions. The parameters used in experimental results were $B = B_5^{(\text{disk})}$, that is, a disk-shaped SE of radius equal to 5 pixels; v, a minima selection threshold automatically calculated from the data using the multi-level Otsu method (Plaza, et al., 2002) and a pixel vector similarity threshold S that was set to 0.01 in experiments. The above values were selected empirically, although we

Table 2. Number of correct detections (C), over-segmentations (O), under-segmentations (U), missed (M) and noise (N) regions obtained after applying the multichannel watershed algorithm, using different tolerance (T) values, to the phantom image in Fig. 3 corrupted by noise in different proportions.

	SNR = 30 dB					SNR = 20 dB					SNR = 10 dB				
T	C	O	U	M	N	C	O	U	M	N	C	O	U	M	N
95%	4	0	0	0	0	2	1	1	0	2	2	0	2	0	4
90%	4	0	0	0	0	3	0	1	0	2	3	0	1	0	3
80%	4	0	0	0	0	4	0	0	0	1	3	0	1	0	1
70%	4	0	0	0	0	4	0	0	0	0	3	0	1	0	0
60%	4	0	0	0	0	4	0	0	0	0	4	0	0	0	0
50%	4	0	0	0	0	4	0	0	0	0	4	0	0	0	0

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Figure 4. Four spectral bands of a real MRI brain image.



(a) Band 1 (b) Band 2 (c) Band 3 (d) Band 4

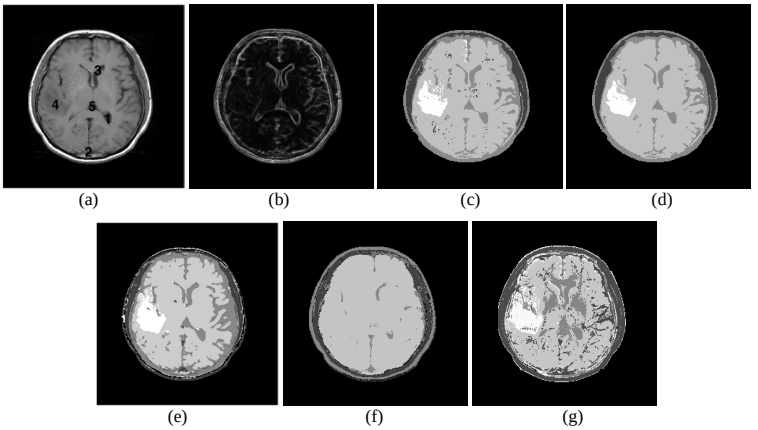
experimentally observed that the algorithm behaved similarly with other parameter settings. In addition, several tolerance threshold values were considered in the computation of the five statistical metrics above. As shown by the table, most of the regions detected by the multichannel watershed algorithm were labeled as correct detections, even for very high tolerance thresholds or signal-to-noise ratios. The above results demonstrated that the proposed algorithm could produce accurate results in the presence of noise by integrating the information available from all the available channels in combined fashion.

Real MRI Image Experiments

In the following experiments, a real MRI multichannel image was used for performance evaluation. Figure 4 shows the four different slices of the considered case study. Since in many MRI applications the three cerebral tissues, GM, WM and CSF, are of major interest, Figure 5(a) shows five markers with higher *deepness* associated with these regions. They are shown as numbers superimposed on band 3 of the real image, where the order relates to their deepness score (the pixel labeled as "1" has the highest deepness). On other hand, Figure 5(b) shows the morphological gradient obtained for this case study. The result of the multichannel flooding-based watershed algorithm (before region merging) from the markers in Figure 5(a) is given in Figure 5(c), where the considered parameter values were the same as those used for phantom experiments in the previous subsection. The final result after the region merging step is given in Figure 5(d). For illustrative purposes, Figures 5(e-g), respectively, show the segmentation result obtained by other standard algorithms. The first one is a standard single-channel watershed algorithm (Rajapakse, et al., 1997) applied to the band with higher contrast, that is, band 2 in Figure 4(b). The second one is a watershed clustering-based technique applied to the spectral feature space (Soille, 1996). This approach differs from our proposed combined method in that it separates the spectral and the spatial information. It first clusters the data in spectral space and then segments objects assuming that the pixels in a given class should have relatively homogeneous spectral intensities. The third approach is the well known ISODATA segmentation procedure that is regarded as a benchmark for most unsupervised segmentation algorithms. The algorithm uses the Euclidean distance as a similarity measure to cluster data elements into different classes (Richards & Jia, 1999).

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Figure 5.



(a) Five markers with highest deepness score, represented by numbers superimposed on band 3; (b) Multichannel morphological gradient; (c) Multichannel watershed segmentation (before merging); (d) Multichannel watershed segmentation (after merging); (e) Single-channel watershed segmentation; (f) Soille's watershed-based segmentation; (g) ISODATA segmentation.

As observed in Figure 5, the multichannel watershed algorithm in Figures 5(c-d) has several advantages over the other segmentation approaches tested. First, the proposed method was able to impose smoothness in the segmentation result. The segmentations produced by other methods often lack spatial consistency. For instance, the single-channel watershed algorithm in Figure 5(e) produced watershed lines which were rather noisy and jagged, even after region merging. This was because only one band was used, and the segmentation was dependent on intensity values at that particular band. In this work we selected the band with higher contrast for the single-channel watershed run; this selection might not be optimal in other cases. On the other hand, the watershed-based clustering approach in Figure 5(f) produced a smoother segmentation as a consequence of the (separate) use of spatial and spectral information. However, this approach missed some important regions clearly visible in the results produced by all other algorithms. Finally, the ISODATA segmentation in Figure 5(g) was produced using the spectral information only, that is, a pixel was classified depending on its spectral values whatever those of its neighbors. This resulted in a rather noisy and disconnected output. As can be seen in Figures 5(c) and 5(d), the region merging stage implemented in the proposed method improved the segmentation by associating together some disconnected regions resulting from the flooding. Overall, results in Figure 5 demonstrate the importance of using spatial and spectral information in simultaneous fashion.

Finally, it should be noted that no quantitative analysis was conducted for the real MRI experiments due to the lack of ground-truth information. However, a visual evaluation of the results in Figure 5 by an expert radiologist indicated a "better

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

delineation of objects and superior spatial consistency of multichannel watershed over the other tested approaches." As for computational complexity, we used a PC with an AMD Athlon 2.6 GHz processor and 512 Megabytes of RAM to run all the experiments in this section, and it was found that the multichannel watershed algorithm produced segmentation results in less than 30 seconds in all cases. This was mainly due to the limited number of bands available. Subsequently, no parallelization strategies were deemed necessary for computer-aided diagnosis of MRI image experiments. However, multichannel image data in other applications, such as hyperspectral images in remote sensing, are characterized by high-dimensional images with hundreds of spectral bands. As a result, further experimentation using real hyperspectral data sets with high dimensionality is pertinent.

Remotely Sensed Hyperspectral Data Experiments

The image data set used in experiments is a hyperspectral scene collected by the 224-band Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS), a remote sensing instrument operated by NASA's Jet Propulsion Laboratory (Green, et al., 1998). The data set was collected over an agricultural test site located in Salinas Valley, California, and represents a challenging segmentation problem. Fortunately, extensive ground-truth information is available for the area, allowing a quantitative assessment in terms of segmentation accuracy. Figure 6(a) shows the entire scene and a sub-scene of the dataset (called hereinafter Salinas A), dominated by directional regions. Figure 6(b) shows the 15 available ground-truth regions. The available data volume (over 50 Mb) creates the need for parallel watershed-based analysis able to produce segmentation results quickly enough for practical use.

Table 3 displays the number of correct detections, over-segmentations, under-segmentations and missed regions obtained after applying the proposed multichannel segmentation algorithm using disk-shaped SEs with different radiuses, measured using different tolerance thresholds. For illustrative purposes, results by two other standard algorithms, ISODATA and Soille's watershed-based clustering, are also displayed. It should be noted that the statistics for noise regions are not provided on purpose, due to the fact that available ground-truth information displayed in Figure 6(b) is not absolute. In all cases, parameter v was set automatically using the multi-level Otsu method, and parameter S was set to 0.01 empirically. As shown by Table 3, the use of appropriate structuring element sizes in the proposed method produced segmentation results which were superior to those found by ISODATA and Soille's watershed-based clustering algorithm. In particular, the best results were obtained when a disk-shaped SE $B = B_{15}^{(\text{disk})}$ was used. This is mainly due to the relation between the SE and the spatial properties of regions of interest in the scene. Specifically, the usage of $B_{15}^{(\text{disk})}$ resulted in a number of correct detections (11) which was the highest one in experiments, while the scores for all error metrics were minimized. Interestingly, no under-segmentation or missed instances were obtained in this case, while only one case of over-segmentation was observed. This case comprised the four lettuce romaine regions contained in the Salinas A subscene, which are at different weeks since planting (4, 5, 6 and 7 weeks, respectively), and covering the soil in different proportions. These four ground-truth regions were always detected as a single, over-segmented region, which is a reasonable

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Figure 6. (a) Spectral band at 488 nm of an AVIRIS hyperspectral image comprising several agricultural fields in Salinas Valley, California, and a sub-scene of the dataset (Salinas A), outlined by a white rectangle. (b) Land-cover ground truth regions.

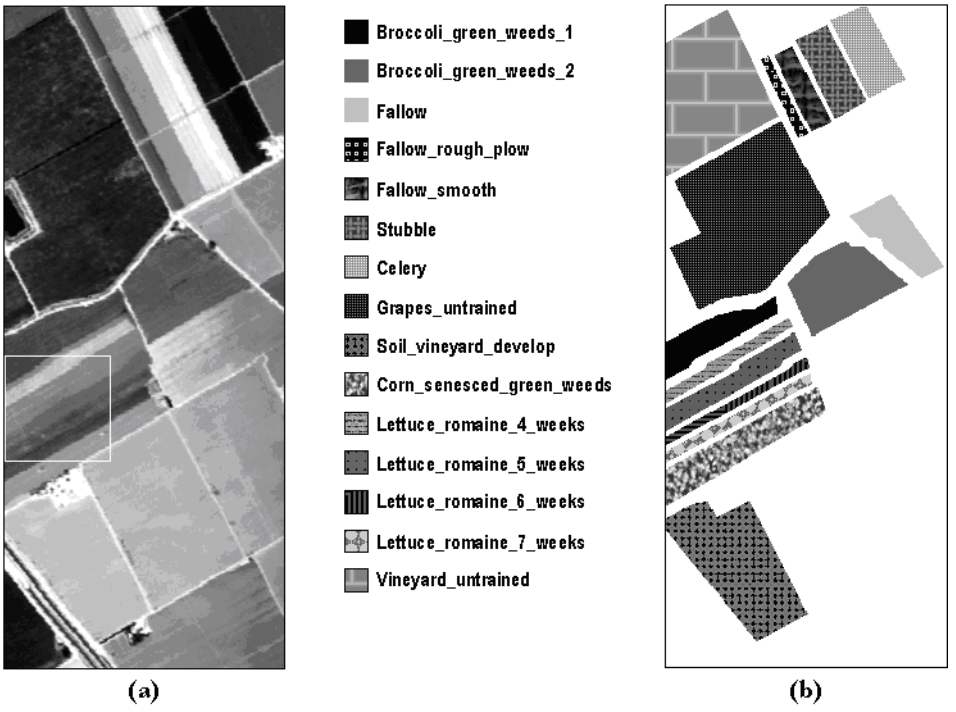


Table 3. Number of correct detections (C), over-segmentations (O), under-segmentations (U) and missed (M) regions obtained after applying the proposed multichannel watershed algorithm, Soille's watershed-based clustering, and ISODATA to the AVIRIS Salinas scene in Fig. 6(a) using different tolerance (T) values.

Method		$T = 80\%$				$T = 90\%$				$T = 95\%$			
		C	O	U	M	C	O	U	M	C	O	U	M
Multichannel watershed Algorithm	$B_3^{(\text{disk})}$	2	2	5	6	2	2	2	9	1	3	2	9
	$B_7^{(\text{disk})}$	6	2	3	5	5	2	3	6	5	2	2	7
	$B_{11}^{(\text{disk})}$	10	0	0	6	10	0	1	5	9	0	1	6
	$B_{15}^{(\text{disk})}$	11	1	0	0	11	1	0	0	11	1	0	0
Watershed-based clustering		7	1	2	5	6	1	1	6	5	1	1	8
ISODATA		3	2	2	8	2	1	2	10	2	0	1	12

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

segmentation result given the slight differences in the spectral characteristics of the four lettuce fields. Overall, the results shown in Table 3 reveal that the proposed algorithm can achieve very accurate segmentation results in a complex analysis scenario given by agricultural classes with very similar spectral features.

It should be noted that the proposed algorithm required several hours of computation in the same computing environment described in MRI experiments, which created the need for a parallel implementation. In order to investigate the efficiency of our parallel multichannel watershed implementation, we coded the algorithm using the C++ programming language with calls to a message passing interface (MPI). The parallel code was tested on a high-performance Beowulf cluster (Brightwell, et al., 2000) available at NASA's Goddard Space Flight Center in Maryland. The system used in experiments, called Thunderhead, is composed of 256 dual 2.4 Ghz Intel Xeon nodes with 1 Gigabyte of local memory and 80 Gigabytes of main memory.

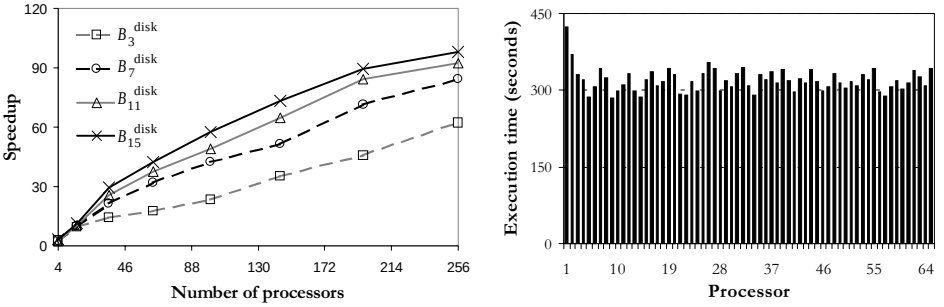
Table 4 shows execution times in seconds of the parallel algorithm with the AVIRIS scene for several combinations of SE sizes and number of processors. Processing times were considerably reduced as the number of processors was increased. For instance, for the case of using $B_{15}^{(\text{disk})}$, which resulted in the most accurate segmentation results as shown in Table 4, 36 processors were required to produce a segmentation result in less than ten minutes. If the application under study can tolerate less accurate segmentations, such as those obtained using $B_{3}^{(\text{disk})}$, then the number of required processors to produce the output in about five minutes was only 16. In order to further analyze the scalability of the parallel code, Figure 7(a) plots the speed-up factors achieved by the parallel algorithm over a single-processor run of the algorithm as a function of the number of processors used in the parallel computation. The achieved speed-up factors were better

Table 4. Execution time in seconds, T(K)*, and speed-up,* S_K*, achieved by the proposed multichannel watershed algorithm with the AVIRIS Salinas scene in Fig. 6(a) for several combinations of structuring element sizes and number of processors (*K*).*

	$B_3^{(\text{disk})}$		$B_7^{(\text{disk})}$		$B_{11}^{(\text{disk})}$		$B_{15}^{(\text{disk})}$	
K	T(K)	S_K	T(K)	S_K	T(K)	S_K	T(K)	S_K
1	3145	1.00	7898	1.00	11756	1.00	16234	1.00
4	1195	2.63	2668	2.96	5090	2.31	5026	3.23
16	329	9.56	772	10.23	1112	10.58	1445	11.23
36	220	14.34	372	21.22	459	25.56	551	29.45
64	180	17.43	245	32.11	313	37.45	424	39.34
100	134	23.45	186	42.45	238	49.21	282	57.45
144	89	35.10	154	51.25	182	64.56	221	73.28
196	68	45.67	110	71.23	139	84.32	181	89.34
256	49	62.45	93	84.39	127	92.34	165	98.23

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Figure 7. (a) Speed-up factors achieved by the parallel algorithm as a function of the number of processors. (b) Execution times in seconds achieved for each of the processors using $B_{15}^{(\text{disk})}$ as structuring element and 64 processors.



for large SEs, a fact that reveals that the proposed parallel implementation is more effective as the volume of computations increases.

Finally, in order to investigate load balance, Figure 7(b) shows the execution times of the parallel algorithm for each of the processors on Thunderhead for a case study where 64 processors were used in the parallel computation; one processor (master or root processor) presents a slightly higher computational load as compared to the other processors. This comes as no surprise because the root processor is in charge of data partitioning, and also combines the partial results provided by every processor. It can be seen, however, that load balance is much better among the rest of the processors. Summarizing, we can conclude that the proposed multichannel watershed segmentation algorithm, implemented on a commodity cluster of PCs, achieved good results in terms of segmentation accuracy, speed-up, scalability and load balance in the context of a high-dimensional image analysis application, dominated by large data volumes and complex patterns of communication and calculation.

CONCLUSIONS

This chapter has developed an approach to generalize the concepts of mathematical morphology to multichannel image data. A new vector organization scheme was described, and fundamental morphological vector operations were defined by extension. Theoretical definitions of extended morphological operations were then used in the formal definition of a multichannel watershed-based segmentation algorithm, which naturally combines the spatial and spectral/temporal information present in multichannel images in simultaneous fashion. While such an integrated approach holds great promise in several applications, it also creates new processing challenges. For that purpose, this chapter also developed a parallel implementation which allows processing of large images quickly enough for practical use. A quantitative segmentation evaluation

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

comparison with regard to standard techniques, using both MRI brain images and remotely sensed hyperspectral data collected by the NASA's Jet Propulsion Laboratory AVIRIS imaging spectrometer, revealed that the proposed parallel algorithm can produce highly accurate segmentation results in reasonable computation times, even when the computational requirements introduced by multichannel imagery are extremely large.

ACKNOWLEDGMENTS

The research in this chapter was supported by the European Commission through the project entitled "Performance analysis of endmember extraction and hyperspectral analysis algorithms," (contract no. HPRI–1999–00057). The authors would like to thank Professor Chein-I Chang for providing the MRI data and Dr. J. Anthony Gualtieri for providing the hyperspectral data. A. Plaza would also like to acknowledge support received from the Spanish Ministry of Education and Science (Fellowship PR2003-0360), which allowed him to conduct research as postdoctoral scientist at NASA's Goddard Space Flight Center and University of Maryland, Baltimore County.

REFERENCES

Adams, R. & Bischof, L. (1994). Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *16*, 641-647.

Beucher, S. (1994). Watershed, hierarchical segmentation and waterfall algorithm. In E. Dougherty (Ed.). *Mathematical morphology and its applications to image processing*. Boston: Kluwer.

Brightwell, R., Fisk, L. A., Greenberg, D. S., Hudson, T., Levenhagen, M., Maccabe, et al. (2000). Massively parallel computing using commodity components. *Parallel Computing*, *26*, 243-266.

Chang, C.-I (2003). *Hyperspectral imaging: Techniques for spectral detection and classification*. New York: Kluwer.

Fan, J., Yau, D.K.Y., Elmargamid, A. K., & Aref, W. G. (2001). Automatic image segmentation by integrating color-edge extraction and seeded region growing. *IEEE Transactions on Image Processing*, *10*, 1454-1466.

Green, R.O., et al. (1998). Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS). *Remote Sensing of Environment*, *65*, 227–248.

Haralick, R., & Shapiro, L. (1985). Image segmentation techniques. *Computer Vision, Graphics and Image Processing*, *29*, 100-132.

Hoover, A., et al. (1996). An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*, 673-689.

Le Moigne, J., & Tilton, J. C. (1995). Refining image segmentation by integration of edge and region data. *IEEE Transactions on Geoscience and Remote Sensing*, *33*, 605–615.

Malpica, N., Ortuño, J. E., & Santos, A. (2003). A multichannel watershed-based algorithm for supervised texture segmentation. *Pattern Recognition Letters*, *24*, 1545-1554.

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Mehnert, A., & Jackway, P. (1997). An improved seeded region growing algorithm. *Pattern Recognition Letters*, *18*, 1065-1071.

Moga, A.N., & Gabbouj, M. (1997). Parallel image component labeling with watershed transformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*, 441-450.

Moga, A.N., & Gabbouj, M. (1998). Parallel marker-based image segmentation with watershed transformation. *Journal of Parallel and Distributed Computing*, *51*, 27-45.

Montoya, M. G., Gil, C., & García, I. (2003). The load unbalancing problem for region growing image segmentation algorithms. *Journal of Parallel and Distributed Computing*, *63*, 387-395.

Plaza, A., Martinez, P., Perez, R., & Plaza, J. (2002). Spatial/spectral endmember extraction by multidimensional morphological operations. *IEEE Transactions on Geoscience and Remote Sensing*, *40*(9), 2025-2041.

Plaza, A., Martinez, P., Perez, R., & Plaza, J. (2004). A new approach to mixed pixel classification of hyperspectral imagery based on extended morphological profiles. *Pattern Recognition, 37*, 1097-1116.

Rajapakse, J., Giedd, J., & Rapaport, J. (1997). Statistical approach to segmentation of single-channel cerebral MR images. *IEEE Transactions on Medical Imaging*, *16*, 176-186.

Richards, J., & Jia, X. (1999). *Remote sensing digital image analysis* (*3rd ed*). Berlin: Springer.

Seinstra, F. J., Koelma, D., & Geusebroek, J. M. (2002). A software architecture for user transparent parallel image processing. *Parallel Computing*, *28*, 967-993.

Shafarenko, L., Petrou, M., & Kittler, J. Automatic watershed segmentation of randomly textured color images. *IEEE Transactions on Image Processing*, *6*, 1530-1544.

Soille, P. (1996). Morphological partitioning of multispectral images. *Journal of Electronic Imaging*, *5*, 252-265.

Soille, P. (2003). *Morphological image analysis, principles and applications* (*2nd ed.*). Berlin: Springer.

Tilton, J.C. (1999). A recursive PVM implementation of an image segmentation algorithm with performance results comparing the HIVE and the Cray T3E. In *Proceedings of the 7th Symposium on the Frontiers of Massively Parallel Computation*, Annapolis, Maryland.

Vincent, L., & Soille, P. (1991). Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*, 583-598.

Vincent, L. (1993). Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE Transactions on Image Processing, 2*, 176-201.

Zhang, Y. J. (1996). A survey on evaluation methods for image segmentation. *Pattern Recognition*, *29*, 1335-1346.

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Section V: Special Segmentation Applications

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.