Lattice Algebra Approach to Color Image Segmentation

Article *in* Journal of Mathematical Imaging and Vision · February 2012

DOI: 10.1007/s10851-011-0302-2 · Source: DBLP

Juan Carlos Valdiviezo-N
Centro de Investigación en Ciencias de Infor…
28 PUBLICATIONS **105** CITATIONS
SEE PROFILE

Gerhard Ritter
University of Florida
146 PUBLICATIONS **3,429** CITATIONS
SEE PROFILE

J Math Imaging Vis (2012) 42:150–162
DOI 10.1007/s10851-011-0302-2

Lattice Algebra Approach to Color Image Segmentation

**Gonzalo Urcid · Juan-Carlos Valdiviezo-N. ·
Gerhard X. Ritter**

Published online: 9 June 2011

Abstract This manuscript describes a new technique for segmenting color images in different color spaces based on geometrical properties of lattice auto-associative memories. Lattice associative memories are artificial neural networks able to store a finite set X of n-dimensional vectors and recall them when a noisy or incomplete input vector is presented. The *canonical* lattice auto-associative memories include the min memory W_{XX} and the max memory M_{XX}, both defined as square matrices of size $n \times n$. The column vectors of W_{XX} and M_{XX}, scaled additively by the components of the minimum and maximum vector bounds of X, are used to determine a set of extreme points whose convex hull encloses X. Specifically, since color images form subsets of a finite geometrical space, the scaled column vectors of each memory will correspond to *saturated* color pixels. Thus, maximal tetrahedrons do exist that enclose proper subsets of pixels in X and such that other color pixels are considered as linear mixtures of extreme points determined from the scaled versions of W_{XX} and M_{XX}. We provide illustrative examples to demonstrate the effectiveness of our method including comparisons with alternative segmentation methods from the literature as well as color separation results in four different color spaces.

Keywords Color image segmentation · Color spaces · Convex sets · Lattice auto-associative memories · Linear mixing model · Pixel based segmentation · Unsupervised clustering

G. Urcid (✉) · J.-C. Valdiviezo-N.
Optics Department, INAOE, Tonantzintla, Pue 72000, Mexico
e-mail: gurcid@inaoep.mx

G.X. Ritter
CISE Department, University of Florida, Gainesville,
FL 32611-6120, USA
e-mail: ritter@cise.ufl.edu

1 Introduction

In several image processing and analysis applications, image segmentation is a preliminary step in the description and representation of regions of interest [1–4]. Segmentation techniques, first developed for grayscale images [5–8], have been extended, enhanced or changed to deal efficiently with color images coded in different color spaces as explained next.

Color image segmentation has been approached from several perspectives that currently are categorized as pixel, area, edge, and physics based segmentation, for which early compendiums appeared in [9, 10]. State-of-the-art surveys are given in [11, 12]. For example, pixel based segmentation includes histogram techniques and cluster analysis in color spaces. Optimal thresholding [13] and the use of a perceptually uniform color space [14] are examples of histogram based techniques. Area based segmentation contemplates region growing as well as split-and-merge techniques, whereas edge based segmentation embodies local methods and extensions of the morphological watershed transformation. This transformation and the flat zone approach to color image segmentation were originally developed, respectively, in [15] and [16]. A seminal work employing Markov random fields for splitting and merging color regions was proposed in [17]. Other recent developments contemplate the fusion of various segmentation techniques such as the application of morphological closing and adaptive dilation to color histogram thresholding [18] or the use of the watershed algorithm for color clustering with Markovian labeling [19]. Physics based segmentation relies on adequate reflection models of material objects such as inhomogeneous dielectrics, plastics, or metals [20, 21]. Nevertheless, its applicability has been limited to finding changes in materials whose reflection properties are well studied and modeled properly.

Recently, soft computing techniques [22] or fuzzy principal component analysis coupled with clustering based on recursive one-dimensional histogram analysis [23], suggest alternative ways to segment a color image. In order to quantify the results obtained from different segmentation schemes, the subject of color image segmentation evaluation has been briefly exposed in [24]. Basic treatment of image segmentation performed in both Hue-Saturation-Intensity (HSI) and RGB color spaces is given in [25, 26]; for a more complete and systematic exposition of color image segmentation methods see [27] or [28]. Also, from the standpoint of lattice algebra, [29, 30] are recent efforts related to the unification of lattice theory based image processing, computational intelligence, modeling, and knowledge representation.

In this paper we present a lattice algebra based technique for image segmentation applied to RGB (Red-Green-Blue) color images transformed to other representative systems, such as the HSI (Hue-Saturation-Intensity), the $I_1I_2I_3$ (principal components approximation), and the L*a*b* (Luminance–redness/greenness–yellowness/blueness) color spaces. The proposed method relies on the min W_{XX} and max M_{XX} lattice auto-associative memories (LAAMs), where X is the set formed by 3D pixel vectors or colors. The scaled column vectors of any memory together with the minimum or maximum bounds of X may form the vertices of tetrahedra enclosing subsets of X, and will correspond to the most saturated color pixels in the image. Image partition into regions of similar colors is realized by linearly unmixing pixels belonging to tetrahedra determined by the columns of the scaled lattice auto-associative memories W and M, and then by thresholding and scaling pixel color fractions obtained numerically by applying a least squares method, such as the *linear least squares* (LLS) method also known as *generalized matrix inversion* [31], or the *non-negative least squares* (NNLS) method [32]. In the final step segmentation results are displayed as grayscale images. The lattice algebra approach to color image segmentation can be categorized as a pixel based unsupervised technique. Preliminary research and computational experiments on the proposed method for segmenting color images were reported in [33, 34].

The paper is organized as follows: Sect. 2 presents background material on image segmentation and a general overview of minimax algebra and lattice auto-associative memories; Sect. 3 develops with some detail the segmentation technique based on the scaled column vectors of LAAMs and briefly describes the linear mixing model used to determine the color fractions composing any pixel vector in the input image. Illustrative examples using synthetic and real images are provided to establish how the proposed method works and how it compares in computational effort, for example, with the c-means and fuzzy c-means clustering techniques. In Sect. 4, we show other segmentation results for additional images represented in the color spaces listed above. Finally, Sect. 5 gives the conclusions and some pertinent comments concerning this research.

2 Mathematical Background

2.1 Image Segmentation

Although there are several approaches to segment a color image, as briefly described in the Introduction, a mathematical description of the segmentation process, common to all approaches, can be given using set theory [1, 3, 4, 25]. In this framework, to segment an image is to divide it into a finite set of disjoint regions whose pixels share well-defined attributes. We recall from basic set theory that a partition of a set is a family of pairwise disjoint subsets covering it. Mathematically, we have

Definition 1 Let X be a finite set with k elements. A *partition* of X is a family $\mathcal{P} = \{R_i\}$ of subsets of X, each with k_i elements for $i = 1, \ldots, q$, that satisfy the following conditions: 1) $R_i \cap R_j = \varnothing$ for $i \neq j$ (pairwise disjoint subsets) and 2) $\bigcup_{i=1}^{q} R_i = X$ where $\sum_{i=1}^{q} k_i = k$ (whole set covering).

Note that the only attribute shared between any two elements of X with respect to a given partition $\mathcal{P}$ is their membership to a single subset R_i of X. Unfortunately, the simple attribute of sharing the same membership is not enough to distinguish or separating objects of interest in a given image. Therefore, Definition 1 must be enriched by imposing other conditions required for image segmentation. Additional attributes shared between pixels (elements of X) can be, for example, spatial contiguity, similar intensity or color, and type of connectedness. All or some of these quantifiable attributes can be gathered into a single *uniformity* criterion specified by a logical predicate. A mathematical statement of our intuitive notion of segmentation follows next.

Definition 2 Let X be a finite set with k elements. A *segmentation* of X is a pair $(\{R_i\}, p)$ composed of a family $\{R_i\}$ of subsets of X each with k_i elements for $i = 1, \ldots, q$, and a logical predicate p specifying a uniformity criterion between elements of X, that satisfy the following conditions: 1) the family $\{R_i\}$ is a partition $\mathcal{P}$ of X, 2) for any i, R_i is a connected subset of X, 3) $\forall i$, $p(R_i) = \textbf{true}$ (elements in a single subset share the same attributes), and 4) for $i \neq j$, $p(R_i \cup R_j) = \textbf{false}$ (elements in a pairwise union of subsets do not share the same attributes).

With respect to condition 2) in Definition 2, we remind that a connected subset R_i is a set where every pair of elements $\{x_s, x_t\} \in R_i$ is connected in the sense that, a sequence of elements, denoted by $(x_s, \ldots, x_r, x_{r+1}, \ldots, x_t)$, exists such that $\{x_r, x_{r+1}\}$ belong to the same spatial neighborhood and all points belong to R_i. A weaker but still useful version of Condition 4) in Definition 2, requires that, R_i and R_j should be neighbor sets. Loosely speaking, a subset $R_i \subset X$ is commonly refer as an image *region*. Whether regions can be disconnected (2nd condition of Definition 2 is not imposed), multi-connected (with holes), should have smooth boundaries, and so forth depends on the application's domain, segmentation technique, and goals. Perceptually, the segmentation process must convey the necessary information to visually recognize or identify the prominent features contained in the image such as color hue, brightness or texture. Hence, adequate segmentation is essential for further description and representation of regions of interest suitable for image analysis or image understanding. We turn now to the description of some basic concepts of minimax algebra as well as some background material about lattice auto-associative memories needed for Sects. 3 and 4.

2.2 Lattice Associative Memories

The basic numerical operations of computing the maximum or minimum of two numbers usually denoted as functions $\max(x, y)$ and $\min(x, y)$ will be written as binary operators using the "join" and "meet" symbols employed in lattice theory, i.e., $x \vee y = \max(x, y)$ and $x \wedge y = \min(x, y)$. We use lattice matrix operations [35, 36] that are defined elementwise using the underlying structure of $\mathbb{R}_{-\infty}$ or $\mathbb{R}_{\infty}$ as semirings. For example, the maximum of two matrices X, Y of the same size $m \times n$ is defined as $(X \vee Y)_{ij} = x_{ij} \vee y_{ij}$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. Inequalities between matrices are also verified elementwise, for example, $X \leq Y$ if and only if $x_{ij} \leq y_{ij}$. Also, the *conjugate matrix* X^* is defined as $-X^t$ where X^t denotes usual matrix transposition. The *max-of-sums* $X \boxdot Y$, of appropriately sized matrices and the *min-of-sums* $X \boxdot Y$, are defined, for $i = 1, \ldots, m$ and $j = 1, \ldots, n$, respectively, as $(X \boxdot Y)_{ij} = \bigvee_{k=1}^{p} (x_{ik} + y_{kj})$ and $(X \boxdot Y)_{ij} = \bigwedge_{k=1}^{p} (x_{ik} + y_{kj})$. For $p = 1$ these lattice matrix operations reduce to the *outer sum* of two vectors $\mathbf{x} = (x_1, \ldots, x_n)^t \in \mathbb{R}^n$ and $\mathbf{y} = (y_1, \ldots, y_m)^t \in \mathbb{R}^m$, given by the $m \times n$ matrix ($i = 1, \ldots, m$ and $j = 1, \ldots, n$)

$$\mathbf{y} \times \mathbf{x}^t = (y_i + x_j) = \begin{pmatrix} y_1 + x_1 & y_1 + x_2 & \cdots & y_1 + x_n \\ y_2 + x_1 & y_2 + x_2 & \cdots & y_2 + x_n \\ \vdots & \vdots & \vdots & \vdots \\ y_m + x_1 & y_m + x_2 & \cdots & y_m + x_n \end{pmatrix}. \tag{1}$$

Henceforth, let $(\mathbf{x}^1, \mathbf{y}^1), \ldots, (\mathbf{x}^k, \mathbf{y}^k)$ be k vector pairs with $\mathbf{x}^\xi = (x_1^\xi, \ldots, x_n^\xi)^t \in \mathbb{R}^n$ and $\mathbf{y}^\xi = (y_1^\xi, \ldots, y_m^\xi)^t \in \mathbb{R}^m$ for $\xi = 1, \ldots, k$. For a given set of vector associations $\{(\mathbf{x}^\xi, \mathbf{y}^\xi) : \xi = 1, \ldots, k\}$ we define a pair of associated matrices (X, Y), where $X = (\mathbf{x}^1, \ldots, \mathbf{x}^k)$ and $Y = (\mathbf{y}^1, \ldots, \mathbf{y}^k)$. Thus, X is of dimension $n \times k$ with i, jth entry x_i^j and Y is of dimension $m \times k$ with i, jth entry y_i^j. To store k vector pairs $(\mathbf{x}^1, \mathbf{y}^1), \ldots, (\mathbf{x}^k, \mathbf{y}^k)$ in an $m \times n$ *lattice associative memory* (LAM), also known as *morphological associative memory* (MAM), a similar approach for vector encoding is used as in a linear or correlation memory but instead of the linear outer product, the lattice outer sum in (1) is applied. The *canonical* LAM's are defined as follows.

Definition 3 The *min-memory* W_{XY} and the *max-memory* M_{XY}, both of size $m \times n$, that store a set of associations (X, Y) are given, respectively, by the expressions

$$W_{XY} = \bigwedge_{\xi=1}^{k} [\mathbf{y}^\xi \times (-\mathbf{x}^\xi)^t]; \qquad w_{ij} = \bigwedge_{\xi=1}^{k} (y_i^\xi - x_j^\xi), \tag{2}$$

$$M_{XY} = \bigvee_{\xi=1}^{k} [\mathbf{y}^\xi \times (-\mathbf{x}^\xi)^t]; \qquad m_{ij} = \bigvee_{\xi=1}^{k} (y_i^\xi - x_j^\xi). \tag{3}$$

We speak of a lattice *hetero-associative* memory (LHAM) if $X \neq Y$ and of a lattice *auto-associative* memory (LAAM) if $X = Y$.

The expressions to the left of (2) and (3) are in matrix form and the right expressions are the ij-th entries that give the network weights of the corresponding associative memory. Note that according to (1), for each ξ, $\mathbf{y}^\xi \times (-\mathbf{x}^\xi)^t$ is a matrix E^ξ of size $m \times n$ that memorizes the association pair $(\mathbf{x}^\xi, \mathbf{y}^\xi)$ hence $W_{XY} = \bigwedge_{\xi=1}^{k} E^\xi$ and $M_{XY} = \bigvee_{\xi=1}^{k} E^\xi$, which suggests the given names. In this paper we will use LAAMs only, i.e., W_{XX} and M_{XX} of size $n \times n$, and if no confusion arises of what the set X stands for, we denote these memories by W and M respectively. In particular, the main diagonals of both matrices, i.e., w_{ii} and m_{ii} consist entirely of zeros. Since $Y = X$, $X \boxdot X^* = (X^*)^* \boxdot X^* = (X \boxdot X^*)^*$, and, therefore, $M = W^*$. Hence, the min-memory and the max-memory are *dual* to each other in the sense of matrix conjugation; consequently, $m_{ij} = -w_{ji}$.

This type of non-linear associative memories, developed from a lattice algebra approach, were introduced as a new paradigm in neural computation to deal with the problem of recalling exemplar patterns from noisy binary or real valued inputs [37–39]. Later, several advancements were achieved including theoretical foundations [40], increased recall capability [41, 42] of exemplar patterns degraded by considerable amounts of random noise, and hyperspectral imagery endmember detection [44–49].

3 LAAMs Approach to Color Image Segmentation

In this section, for illustrative purposes, we consider only images coded in the RGB color space. The first subsection gives a detailed description, in three stages, of the proposed segmentation approach. A brief comment on two fundamental clustering techniques follows in the second subsection as a framework for computational comparisons. The third subsection illustrates the segmentation results on synthetic and real RGB color images obtained by the LAAM's approach together with the results derived from the c-means and fuzzy c-means clustering techniques.

3.1 The Segmentation Process

Segmentation of a color image is performed in stages including: 1) computation of the scaled lattice auto-associative memories, 2) linear unmixing of color pixels using least square methods, and 3) thresholding color fractions to produce color segmentation maps represented as grayscale images. These stages are explained in detail in the following paragraphs.

Given a color image A consisting of $p \times q$ pixels, we build a set X containing all different colors (3-dimensional vectors) present in A. If $|X| = k$ denotes the number of elements in set X, then $k \leq pq = |A|$, where pq is the maximum number of colors available in A. Then, using the right expressions of (2) and (3), the memory matrices min-W_{XX} and max-M_{XX} are computed and to make explicit their respective column vectors, we rewrite them, respectively, as $W = (\mathbf{w}^1, \mathbf{w}^2, \mathbf{w}^3)$ and $M = (\mathbf{m}^1, \mathbf{m}^2, \mathbf{m}^3)$. By definition, the column vectors of W may not necessarily belong to the space $[0, 255]^3$ since W usually has negative entries. The general transformation given in the next definition, will translate the column vectors of W within the color cube.

Definition 4 Let $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^k\}$ be a finite subset of $\mathbb{R}^n$. The *minimum*- and *maximum vector* bounds are given, respectively by $\mathbf{v} = \bigwedge_{\xi=1}^{k} \mathbf{x}^\xi$ and $\mathbf{u} = \bigvee_{\xi=1}^{k} \mathbf{x}^\xi$. Their corresponding entries, for $i = 1, \ldots, n$, are computed as

$$v_i = \bigwedge_{\xi=1}^{k} x_i^\xi; \qquad u_i = \bigvee_{\xi=1}^{k} x_i^\xi. \tag{4}$$

Let $W = (\mathbf{w}^1, \ldots, \mathbf{w}^n)$ and $M = (\mathbf{m}^1, \ldots, \mathbf{m}^n)$ be the sets of column vectors of the min- and max memories relative to X, then *additive scaling* results in two scaled matrices, denoted respectively $\overline{W}$ and $\overline{M}$, whose column vectors are defined, for $j = 1, \ldots, n$, by

$$\overline{\mathbf{w}}^j = \mathbf{w}^j + u_j; \qquad \overline{\mathbf{m}}^j = \mathbf{m}^j + v_j, \tag{5}$$

Note that for $j = 1, \ldots, 3$, $\overline{w}_{jj} = u_j$ and $\overline{m}_{jj} = v_j$. Hence, $\operatorname{diag}(\overline{W}) = \mathbf{u}$ and $\operatorname{diag}(\overline{M}) = \mathbf{v}$.

The first stage of the segmentation process is completed by applying (4) and (5) to X, W, and M. Continuing with the description of the proposed segmentation procedure, use is made of the underlying sets of scaled columns $\overline{W} = \{\overline{\mathbf{w}}^1, \overline{\mathbf{w}}^2, \overline{\mathbf{w}}^3\}$ and $\overline{M} = \{\overline{\mathbf{m}}^1, \overline{\mathbf{m}}^2, \overline{\mathbf{m}}^3\}$ including the extreme vector bounds $\mathbf{v}$ and $\mathbf{u}$. Note that, the vectors belonging to the set $\overline{W} \cup \overline{M} \cup \{\mathbf{v}, \mathbf{u}\}$ provide a way to determine several tetrahedra enclosing specific subsets of X such as, e.g., $\overline{W} \cup \{\mathbf{u}\}$ and $\overline{M} \cup \{\mathbf{v}\}$.

The second stage in the segmentation process is accomplished using concepts from convex set geometry. These concepts make it possible to mix colors in any color space. Recall that X is said to be a *convex set* if the straight line joining any two points in X lies completely within X; also, an n-dimensional *simplex* is the minimal convex set or convex hull whose $n + 1$ vertices (extreme points) are affinely independent vectors in $\mathbb{R}^n$. Since the color cube is a subspace of $\mathbb{R}^3$ a 3-dimensional simplex will correspond to a tetrahedron. Thus, considering pixel vectors in a color image enclosed by some tetrahedron, whose base face is determined by its most saturated colors, an estimation of the fractions in which they appear at any other color pixel can be made. A model commonly used for the analysis of spectral mixtures in hyperspectral images, known as the constrained *linear mixing* (LM) model [43], can readily be adapted to segment noiseless color images by representing each pixel vector as a convex linear combination of the most saturated colors. Its mathematical representation is given by

$$\mathbf{x} = S\mathbf{c} = c_1\mathbf{s}^1 + c_2\mathbf{s}^2 + c_3\mathbf{s}^3, \quad \text{subject to} \tag{6}$$
$$c_1, c_2, c_3 \geq 0 \quad \text{(non-negativity)},$$
$$c_1 + c_2 + c_3 = 1 \quad \text{(full additivity)},$$

where, $\mathbf{x}$ is a 3×1 pixel vector, $S = (\mathbf{s}^1, \mathbf{s}^2, \mathbf{s}^3)$ is a square matrix of size 3×3 whose columns are the extreme colors, and $\mathbf{c}$ is the 3×1 vector of "saturated color fractions" present in $\mathbf{x}$. Notice that the most saturated colors in a given image may easily be equal to the set of primary colors (red, green, blue) or to the set of complementary colors (cyan, magenta, yellow). Therefore, the present step consists of solving (6) to find vector $\mathbf{c}$ given that $S = \overline{W}$ or $S = \overline{M}$ for every $\mathbf{x} \in X$, a procedure known as *linear unmixing*. As mentioned earlier in the introduction, to solve the constrained linear system displayed in (6), one can employ the LLS or NNLS methods imposing the full additivity or the positivity constraint, respectively.

In the third and last stage of the segmentation process, once (6) is solved for every color pixel $\mathbf{x}^\xi \in X$, all c_ξ^j fraction values are assembled to form a vector associated with the saturated color $\mathbf{s}^j$, and the final step is carried out by applying a threshold value, in most cases, between 0.3 and 1 to obtain an adequate segmentation depicting the corresponding image partition (see Definition 2). Additional theoretical background on which the proposed method is based as

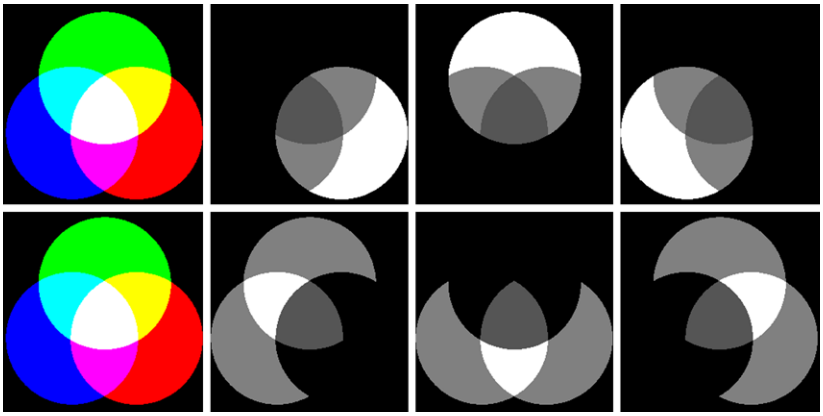

Fig. 1 *1st column*: test RGB color image; *1st row*, *2nd* to *4th cols.*: grayscale images depicting segmented regions containing proportions of red ($\overline{\mathbf{w}}^1$), green ($\overline{\mathbf{w}}^2$), and blue ($\overline{\mathbf{w}}^3$) colors; *2nd row*, *2nd* to *4th cols.*: grayscale images with regions composed of cyan ($\overline{\mathbf{m}}^1$), magenta ($\overline{\mathbf{m}}^2$), and yellow ($\overline{\mathbf{m}}^3$) colors. *Brighter gray tones* correspond to high fractions of saturated colors

well as its application to hyperspectral imagery appears in [47, 49].

3.2 A Comment on Clustering Techniques

Of the many existing approaches to image segmentation [9–12], clustering techniques such as c-means and fuzzy c-means can be applied to color images provided the number of clusters is known beforehand. When using any of these techniques a cluster is interpreted as the mean or average color assigned to an iteratively determined subset of color pixels belonging to X. For an explanation of the basic theory and algorithmic variants concerning the c-means clustering technique cf. [50–52] and similarly, for the fuzzy c-means clustering technique see [53, 54]. In relation to our proposed method based on LAAMs, a comparison with both clustering techniques is immediate since the maximum number of saturated colors determined from W, M, and possibly $\{\mathbf{v}, \mathbf{u}\}$ is always 8, thus the number of clusters is bounded by the interval $[1, 8]$. Furthermore, since any member in the set $\overline{W} \cup \overline{M} \cup \{\mathbf{v}, \mathbf{u}\}$ is an extreme point, we are able to select any two disjoint subsets of three column vectors to form a 3×3 system in order to obtain unique solutions to (6). Therefore, once a pair of triplets is fixed, the number of clusters c can be restricted to the interval $[6, 8]$.

3.3 Segmentation Results and Comparisons

Example 1 (Flat color image) Figure 1 shows in the left column a test RGB color image (primary colors additive mixtures) of size 256×256 pixels that has only 8 different colors. Hence, $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^8\}$ out of a total of 65,536 pixel vectors. The scaled lattice memory matrices and the minimum-, maximum vector bounds are given by

$$\overline{W} = \begin{pmatrix} 255 & 0 & 0 \\ 0 & 255 & 0 \\ 0 & 0 & 255 \end{pmatrix},$$

$$\overline{M} = \begin{pmatrix} 0 & 255 & 255 \\ 255 & 0 & 255 \\ 255 & 255 & 0 \end{pmatrix}, \tag{7}$$

$$\mathbf{v} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \qquad \mathbf{u} = \begin{pmatrix} 255 \\ 255 \\ 255 \end{pmatrix}.$$

For this trivial color image, a simple algebraic analysis yields a closed solution for unmixing color pixels obeying (6). In this case we have

$$\overline{W}^{-1} = \frac{1}{255}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\overline{M}^{-1} = \frac{1}{510}\begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix}. \tag{8}$$

From (8), $\overline{W}^{-1} = I/255$ where I is the 3×3 identity matrix and considering that $x_i^{\xi} \in \{0, 255\}$, $c_i = x_i/255$ verifies trivially the inequalities $0 \leq c_i \leq 1$ for all $i = 1, 2, 3$ and $\xi \in \{1, \ldots, 8\}$. Full additivity is satisfied if $\sum_{i=1}^{3} c_i = \sum_{i=1}^{3} x_i/255 = 1$, therefore color pixel values x_1, x_2, and x_3 lie in the plane $x_1 + x_2 + x_3 = 255$ which occurs only at the points $(255, 0, 0)$, $(0, 255, 0)$, and $(0, 0, 255)$. However, letting $s = x_1 + x_2 + x_3$ the color fractions obtained from the scaled min memory $\overline{W}$ are readily specified by the simple formula

$$c_i = \frac{x_i}{s} = \frac{x_i}{x_1 + x_2 + x_3} \quad \Leftrightarrow \quad s \neq 0, \tag{9}$$

otherwise if $s = 0$ let $c_i = 0$. Similarly, from the inverse matrix $\overline{M}^{-1}$ given in (8), one finds that $c_i = (\sum_{j \neq i} x_j - x_i)/510$ for $i = 1, 2, 3$. However, since $x_i^{\xi} \in \{0, 255\}$ we

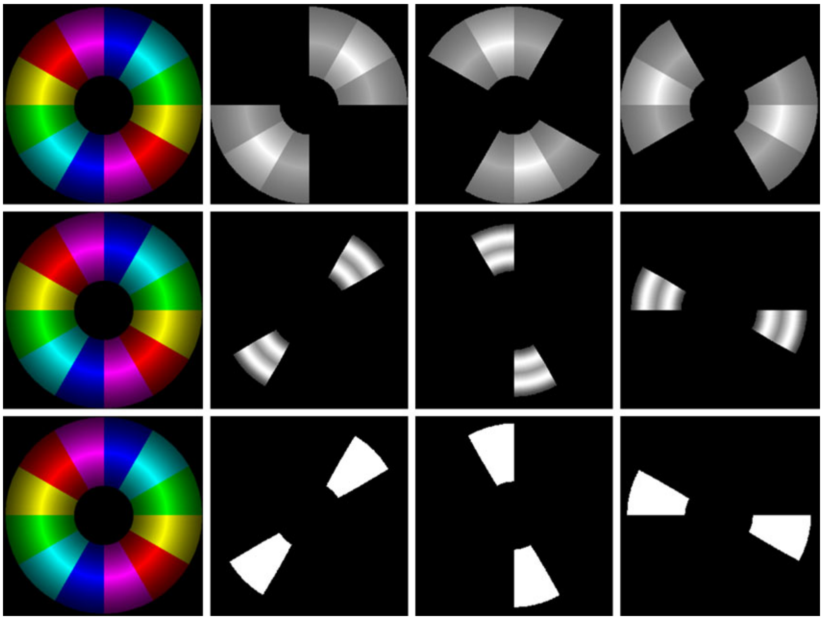

Fig. 2 *1st column*: test RGB color image; *1st row*, *2nd* to *4th cols.*: grayscale images of color fractions obtained by linear unmixing showing the segmentation of cyan ($\overline{\mathbf{m}}^1$), magenta ($\overline{\mathbf{m}}^2$), and yellow ($\overline{\mathbf{m}}^3$) (CMY) colors; *2nd row*, *2nd* to *4th cols.*: fuzzy c-means grayscale images depicting membership distribution in regions of CMY color gradients; *3rd row, 2nd* to *4th cols.*: c-means binary images depicting uniform segmented regions labeled from CMY centroids

Table 1 Fraction values for unmixing pixels of the test RGB color image

Saturated color	Pixel values (x_1, x_2, x_3)	From $\overline{W}$ (c_1, c_2, c_3)	From $\overline{M}$ (c_1, c_2, c_3)
Black	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$
Red	$(255, 0, 0)$	$(1, 0, 0)$	$(0, \frac{1}{2}, \frac{1}{2})$
Green	$(0, 255, 0)$	$(0, 1, 0)$	$(\frac{1}{2}, 0, \frac{1}{2})$
Blue	$(0, 0, 255)$	$(0, 0, 1)$	$(\frac{1}{2}, \frac{1}{2}, 0)$
Cyan	$(0, 255, 255)$	$(0, \frac{1}{2}, \frac{1}{2})$	$(1, 0, 0)$
Magenta	$(255, 0, 255)$	$(\frac{1}{2}, 0, \frac{1}{2})$	$(0, 1, 0)$
Yellow	$(255, 255, 0)$	$(\frac{1}{2}, \frac{1}{2}, 0)$	$(0, 0, 1)$
White	$(255, 255, 255)$	$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$	$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$

have $c_i \in \{-0.5, 0, 0.5, 1\}$; thus, non-negativity is not satisfied for all i. Also, full additivity is verified if $\sum_{i=1}^{3} c_i = \sum_{i=1}^{3}(\sum_{j\neq i} x_j - x_i)/510 = 1$, implying that color pixel values x_1, x_2, and x_3 belong to the plane $x_1 + x_2 + x_3 = 510$, and this can occur only at the points $(0, 255, 255)$, $(255, 0, 255)$, and $(255, 255, 0)$. Therefore, making $s = x_1 + x_2 + x_3$ the color fractions obtained from the scaled max memory $\overline{M}$ are given by the formula

$$c_i = \frac{\sum_{j\neq i} x_j - x_i}{s} = \frac{\sum_{j\neq i} x_j - x_i}{x_1 + x_2 + x_3} \quad \Leftrightarrow \quad s \neq 0, \qquad (10)$$

otherwise if $s = 0$, then $c_i = 0$; also, if $c_i = -1$ (for some i), then set $c_i = 0$ and change c_j to $c_j/2$ for $j \neq i$. Table 1 displays the correspondence between pixel color values and color fractions derived from the scaled LAAMs.

Using the mapping established in Table 1 the color fraction solution vector '**c**' is quickly determined for each one of the 65,536 pixels forming the image, using for S, first the $\overline{W}$ matrix that unmixes the primary colors, then the $\overline{M}$ matrix that unmixes the secondary colors. To the right of the test RGB color image in Fig. 1, the color fraction maps displayed as grayscale images are associated to the saturated colors derived from the column vectors of the scaled LAAMs, except black, considered the image background, and white that results from additive mixture of the three primary colors. Each color fraction segmented image $\mathbf{s}^j$ is visible after a linear scaling from the interval $[0, 1]$ to the grayscale dynamic range $[0, 255]$.

Example 2 (Gradient color image) In Fig. 2, the left column shows a synthetic RGB color image composed by a gradient of primary and secondary colors of size 256 × 256 pixels with 2,400 different colors. Thus, $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^{2400}\}$ (again, from a total of 65,536 color pixels). It turns out that the scaled LAAM matrices and the minimum-, maximum vector bounds are almost the same as those computed in the previous example, (7), except that the numeric value 255 is replaced by 254. Although the given image is rather simple, an algebraic analysis would be impractical for finding a color fractions formula applicable for unmixing every different color present in the image. However, *fast* pixel linear decomposition can be realized, e.g., by generalized matrix inversion (LLS) enforcing full additivity and adequate thresholding of numerical values.

From (6) any $c_q = 1 - c_p - c_r$, where $q = 1, 2, 3$ and $q \neq p < r \neq q$, can be selected to reduce the size of matrix S and vector $\mathbf{c}$. Consequently, computations are simplified by solving for each color pixel the linear system given by

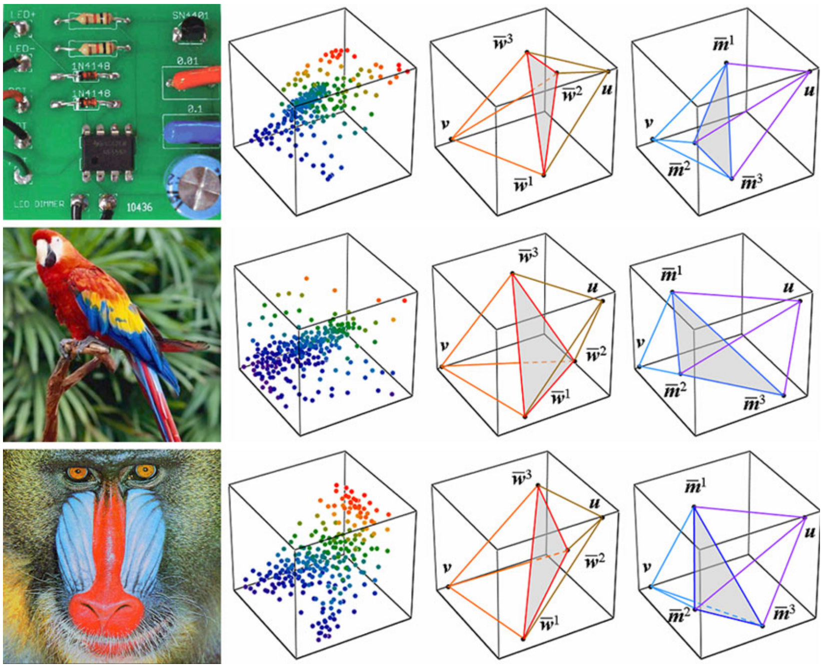

Fig. 3 *1st column*: sample RGB color images; *2nd col.*: scatter plot of a subset of X showing 256 different colors including the most saturated colors determined from $\overline{W}$ and $\overline{M}$; *3rd* and *4th cols.*: tetrahedra determined from proper subsets of $\overline{W} \cup \overline{M} \cup \{\mathbf{v}, \mathbf{u}\}$

$\mathbf{x}_q = S_q \mathbf{c}_q$, where $\mathbf{c}_q = (c_p, c_r)^t$, $S_q = \overline{W}_q$ or $S_q = \overline{M}_q$, and

$$S_q = \begin{pmatrix} s_{1p} - s_{1q} & s_{1r} - s_{1q} \\ s_{2p} - s_{2q} & s_{2r} - s_{2q} \\ s_{3p} - s_{3q} & s_{3r} - s_{3q} \end{pmatrix}, \qquad \mathbf{x}_q = \begin{pmatrix} x_1 - s_{1q} \\ x_2 - s_{2q} \\ x_3 - s_{3q} \end{pmatrix}. \tag{11}$$

In this example we let $q = 1$ and (12) is solved only for $S_1 = \overline{M}_1$. Hence, $c_1 = 1 - c_2 - c_3$ and $\mathbf{c}_1 = (c_2, c_3)^t$. Also, each i-th row of S_1 and entries of the transformed input color vector $\mathbf{x}_1$, for $i = 1, 2, 3$, are given by $(\overline{m}_{i2} - \overline{m}_{i1}, \overline{m}_{i3} - \overline{m}_{i1})$ and $x_i - \overline{m}_{i1}$, respectively. Thresholds applied to fractions values for generating segmented images were computed as

$$u_j = \frac{\tau_j}{256} \bigvee_{\xi=1}^{k} c_\xi^j, \tag{12}$$

where $k = 2{,}400$ and by setting the user defined grayscale threshold $\tau_j = 85$ for all j. The first row in Fig. 2 shows the segmentation produced using $\overline{M}$ (secondary colors), where the brighter gray tones correspond to high fractions of saturated colors. Hence color gradients are preserved as grayscale gradients. Additionally, original color regions composed of some proportion of the saturated colors $\overline{\mathbf{m}}^1$, $\overline{\mathbf{m}}^2$, and $\overline{\mathbf{m}}^3$ appear as middle or dark gray tones. The second row displays the results obtained by applying the fuzzy c-means technique with $c = 7$ and the thresholds values u_j used to cut fuzzy memberships were calculated with (12) setting $\tau_j = 64$ for all $j = 1, \ldots, 7$. Observe that the brighter gray tones are associated with pixels near to fuzzy centroids (high membership values) whereas darker gray tones correspond to pixels far from fuzzy centroids (low membership values); note that original color gradients are not preserved. The third row depicts as black and white binary images the clusters found using the c-means algorithm with $c = 7$ and initial centroids given by the set $\overline{W} \cup \overline{M} \cup \{\mathbf{v}\}$. In this last case thresholds are not needed since the c-means algorithm is a labeling procedure that assigns to all similar colors belonging to a cluster the color value of its centroid. Consequently, a simple labeling procedure is implemented to separate regions of different color.

If $\overline{W}_1$ is selected instead of $\overline{M}_1$ for the system matrix S_1 in (12), similar segmentation results are obtained except that, in this case, red, green, and blue regions would be extracted from the corresponding saturated colors $\overline{\mathbf{w}}^1$, $\overline{\mathbf{w}}^2$, and $\overline{\mathbf{w}}^3$. We remark that Example 2 clearly shows the fundamental difference between the three segmentation methods compared: c-means and fuzzy c-means clustering are statistical and iterative in nature whereas the LAAM's approach coupled with the LM model is a non-iterative geometrical procedure as discussed in Sect. 3.1.

Example 3 (Real color images) Next we provide additional segmentation results for three realistic RGB color images of size 256 × 256 pixels displayed in the first column of Fig. 3 (see Table 2 for image information). For each of these color images, we create a set $X_\ell = \{\mathbf{x}^1, \ldots, \mathbf{x}^{k_\ell}\} \subset [0, 255]^3$ where $\ell = \alpha, \beta, \gamma$, and each vector $\mathbf{x}^\xi \in X_\ell$ is distinct from the others, i.e., $\mathbf{x}^\xi \neq \mathbf{x}^\zeta$ whenever $\xi \neq \zeta$. This is achieved by eliminating pixel vectors of the same color (k_ℓ is given in Table 2). After application of (2)–(3) (LAAMs) and (4) (vector bounds), the scaled matrices $\overline{W}$ and $\overline{M}$ are computed with (5). The numerical entries for the scaled LAAM matrices (3rd column, Table 2) of the sample images are explicitly

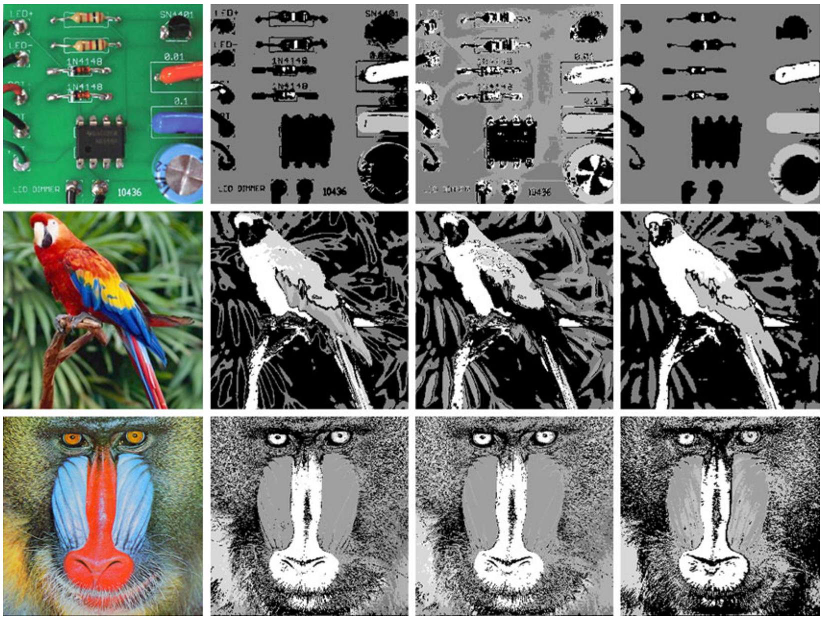

Fig. 4 *1st column*: sample RGB color images; *2nd*, *3rd*, and *4th cols.*: quantized grayscale segmented images composed from results obtained, respectively, with c-means clustering, fuzzy c-means clustering, and scaled LAAMs + LLS linear unmixing techniques

Table 2 Information of sample real RGB color images

Image	Pixels (pq)	Colors ($\lvert X_\ell \rvert = k_\ell$)	Scaled LAAMs
Circuit	65,536	35,932	$\overline{W}_\alpha, \overline{M}_\alpha$
Parrot	65,536	55,347	$\overline{W}_\beta, \overline{M}_\beta$
Baboon	65,536	63,106	$\overline{W}_\gamma, \overline{M}_\gamma$

given below:

$$\overline{W}_\alpha = \begin{pmatrix} 255 & 80 & 101 \\ 71 & 255 & 135 \\ 46 & 154 & 255 \end{pmatrix},$$

$$\overline{M}_\alpha = \begin{pmatrix} 19 & 203 & 228 \\ 194 & 19 & 120 \\ 173 & 139 & 19 \end{pmatrix},$$

$$\overline{W}_\beta = \begin{pmatrix} 255 & 121 & 35 \\ 55 & 251 & 128 \\ 1 & 23 & 255 \end{pmatrix},$$

$$\overline{M}_\beta = \begin{pmatrix} 0 & 200 & 254 \\ 130 & 0 & 228 \\ 220 & 127 & 0 \end{pmatrix},$$

$$\overline{W}_\gamma = \begin{pmatrix} 255 & 129 & 72 \\ 55 & 255 & 156 \\ 0 & 90 & 255 \end{pmatrix},$$

$$\overline{M}_\gamma = \begin{pmatrix} 0 & 200 & 255 \\ 126 & 0 & 165 \\ 183 & 99 & 0 \end{pmatrix}.$$

Notice that the corresponding minimum and maximum vector bounds $\{\mathbf{v}_\ell, \mathbf{u}_\ell\}$ for $\ell = \alpha, \beta, \gamma$ are readily available from the main diagonals of the corresponding LAAM matrices. A 3-D scatter plot of each set X showing only 256 different colors, including the extreme points of the set $\overline{W} \cup \overline{M} \cup \{\mathbf{v}, \mathbf{u}\}$, is depicted in the second column of Fig. 3 for each sample image. Two tetrahedra enclosing points of X are illustrated in the third column of the same figure. The vertices of the left tetrahedron belong to the set $\overline{W} \cup \{\mathbf{v}\}$ and those of the right tetrahedron are in $\overline{W} \cup \{\mathbf{u}\}$; similarly, in the fourth column of Fig. 3, the left tetrahedron has its vertices in the set $\overline{M} \cup \{\mathbf{v}\}$ and the right tetrahedron is formed with the points of $\overline{M} \cup \{\mathbf{u}\}$.

Again, for each RGB color image in Fig. 3, (6) was simplified to (12) setting $q = 1$ and solving it using LLS for each $\mathbf{x} \in X_\ell$, by taking first $\overline{W}_\ell$ and then $\overline{M}_\ell$ as the S matrix for $\ell = \alpha, \beta, \gamma$. It turns out that for the sample images selected the corresponding 3×3 computed scaled LAAMs are non-singular matrices (full rank) and, therefore, the solutions found by the linear unmixing scheme are unique. Since the minimum and maximum bounds $\{\mathbf{v}_\ell, \mathbf{u}_\ell\}$ correspond, respectively, to a "dark" color near black and to a "bright" color near white it is possible to replace a specific column in $\overline{W}$ or $\overline{M}$ with one of these extreme bounds in order to obtain segmentations of dark or bright regions. Thus, final satisfactory segmentation results are produced by an adequate selection of saturated colors $\mathbf{s}^j$ from the set $\overline{W} \cup \overline{M} \cup \{\mathbf{v}, \mathbf{u}\}$. Figure 4 displays the segmentation produced by applying the clustering techniques of c-means, fuzzy c-means, and our proposed LAAMs plus linear unmixing based technique. Results are shown as quantized grayscale images where specific gray tones are associated with selected colors corresponding to cluster centers or extreme points. Table 3 provides the technical information relative to each segmenting algorithm; for example, "runs" is the number of times an al-

Table 3 Technical data used for RGB color image segmentation

Image	c-means	Fuzzy c-means	LAAMs + LLS
Circuit	$c = 8$, runs = 5 distance: squared Euclidean 5th run: 58 iter., 57 sec RGB → 255, 128, 192	$c = 7$, runs = 3 $\exp(U) = 2$, min.imp $= 10^{-5}$ 3rd run: 108 iter., 720 sec RG_1G_2B → 255, 128, 160, 200	$c = 6$, runs = 1 $\overline{W}_\alpha, \overline{M}_\alpha, q = 1$ non-iterative, 30 sec RGB → 255, 128, 192
Parrot	$c = 8$, runs = 5 distance: city block 5th run: 20 iter., 32 sec RGBCY → 255, 128, 160, 192, 216	$c = 6$, runs = 3 $\exp(U) = 2$, min.imp $= 10^{-2}$ 3rd run: 134 iter., 238 sec RG_1G_2Y → 255, 128, 160, 216	$c = 6$, runs = 1 $\overline{W}_\beta, \overline{M}_\beta, q = 1$ non-iterative, 30 sec RGB → 255, 128, 192
Baboon	$c = 7$, runs = 5 distance: city block 5th run: 39 iter., 38 sec RGB_1B_2Y → 255, 128, 160, 176, 216	$c = 6$, runs = 3 $\exp(U) = 2$, min.imp $= 10^{-2}$ 3rd run: 94 iter., 270 sec RGB_1B_2Y → 255, 128, 160, 176, 216	$c = 6$, runs = 1 $\overline{W}_\gamma, \overline{M}_\gamma, q = 1$ non-iterative, 30 sec RGBCY → 255, 128, 160, 192, 216
	procedure: kmeans (Matlab)	procedure: fcm (Matlab)	procedure: geninv (Mathcad)

gorithm is applied to a given image. Specifically, in the Matlab environment, "runs" is equivalent to the "replicates" parameter used for c-means clustering; $\exp(U)$ and min. imp refer to, respectively, the partition matrix exponent and the minimum amount of improvement needed for the objective function to converge in fuzzy c-means clustering. The notation, e.g. RGB → 255, 128, 192, gives the gray levels assigned to the red, green, and blue colors.

4 Segmentation in Different Color Spaces

In this section, for brevity, we will refer to the LAAMs based approach as the WM method. To test the performance of the WM method in different color spaces, besides the standard non-normalized correlated RGB space, we selected as representative alternatives, Ohta's $I_1I_2I_3$ linearly decorrelated RGB color space [12, 17], the HSI non-linear and non-uniform color space [26, 28], and the perceptually uniform color space L*a*b* [14, 28]. Mapping RGB colors to the L*a*b* color space makes use of the linear NTSC illuminant D65 RGB to XYZ conversion matrix.

Example 4 Figure 5 shows in the top left, the "peppers" RGB color image of size 128 × 128 pixels, its HSI transformation displayed as a false RGB color image, and the extreme color pixels determined from $\overline{W}$ (upper row of rectangles) and $\overline{M}$ (lower row of rectangles) in the HSI color space. Here, $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^{13,844}\}$ is reduced from a total of 16,384 pixel vectors. The scatter plot of X is depicted to the left of Fig. 6 together with four tetrahedra enclosing different subsets of X, namely $\overline{W} \cup \{\mathbf{v}\}$ and $\overline{W} \cup \{\mathbf{u}\}$ shown in the middle, or $\overline{M} \cup \{\mathbf{v}\}$ and $\overline{M} \cup \{\mathbf{u}\}$ displayed to the right.The computed scaled memory matrices and vector bounds are

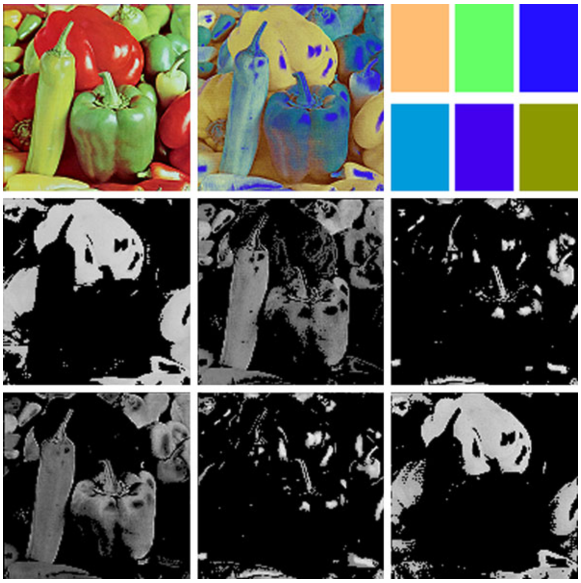

Fig. 5 *1st row*, *left* to *right*: sample RGB color image, transformed HSI color image, saturated colors obtained from $\overline{W}$ (*upper horizontal* array of colored rectangles) and $\overline{M}$ (*lower horizontal* array of colored rectangles); *2nd* and *3rd rows*: grayscale segmented images derived from $\overline{\mathbf{w}}^j$, respectively, $\overline{\mathbf{m}}^j$ for $j = 1, 2, 3$, showing "red/green" pepper regions and bright reflected light regions

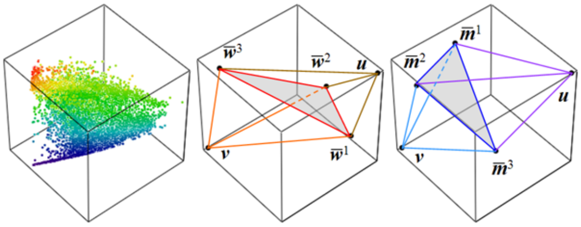

Fig. 6 *Left*: scatter plot of X showing all different colors present in the HSI representation of the "peppers" RGB color image; *middle* and *right*: tetrahedra determined, respectively, from $\overline{W} \cup \{\mathbf{v}, \mathbf{u}\}$ and $\overline{M} \cup \{\mathbf{v}, \mathbf{u}\}$ enclosing four different subsets of X

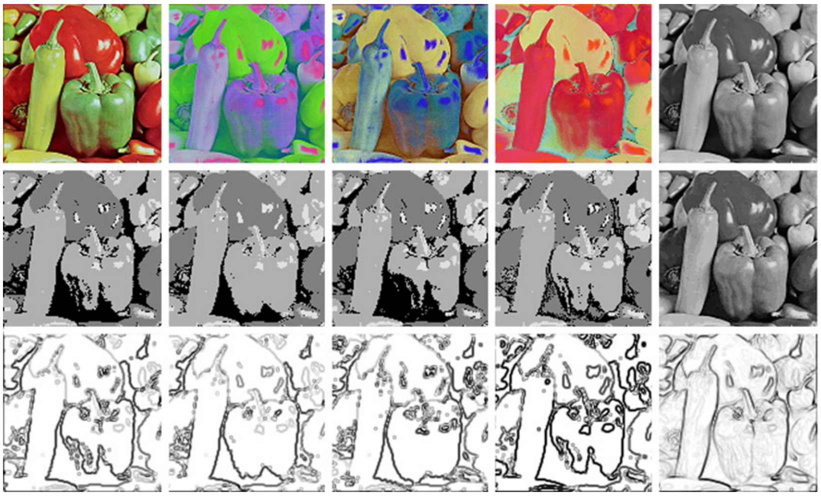

Fig. 7 *1st row*, *1st* to *5th cols.*: "peppers" image in RGB, $I_1I_2I_3$, HSI, L*a*b* color spaces, and NTSC grayscale version; *2nd row*, *1st* to *5th cols.*: grayscale segmented images of "red/green" peppers and bright portions of reflected light corresponding to each color space, and the NTSC grayscale version quantized to 16 levels; *3rd row*, *1st* to *5th cols.*: Sobel edge images corresponding to segmentation methods (1), (2), (5), (6) of Table 4 and the Sobel edge reference image obtained from the quantized NTSC grayscale version

given by

$$\overline{W} = \begin{pmatrix} 255 & 100 & 36 \\ 188 & 255 & 16 \\ 115 & 103 & 255 \end{pmatrix}, \qquad \overline{M} = \begin{pmatrix} 0 & 67 & 140 \\ 155 & 0 & 152 \\ 219 & 239 & 0 \end{pmatrix},$$

$$\mathbf{v} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \qquad \mathbf{u} = \begin{pmatrix} 255 \\ 255 \\ 255 \end{pmatrix}.$$

Using the NNLS numerical method, (6) was solved for every color pixel. The 2nd and 3rd rows in Fig. 5, display the fraction maps obtained from the HSI saturated colors displayed in the top right, whose associated column vectors correspond, respectively, to $\overline{W}$ and $\overline{M}$. As before, thresholds were again computed using (12) with $k = 16{,}384$ and tuning τ_j to adequate values.

For the next example, we recall the mathematical formulas of two measures used for grayscale image comparisons. Specifically, given to matrices $A = (a_{ij})$ and $B = (b_{ij})$ of size $p \times q$ pixels, the *correlation coefficient* $\rho(A, B)$ between A and B, and the *signal to noise ratio* $\mathrm{SNR}(A, B)$ are computed as

$$\rho(A, B) = \frac{\sum_{i=1}^{p} \sum_{j=1}^{q} (a_{ij} - \mu_A)(b_{ij} - \mu_B)}{\sum_{i=1}^{p} \sum_{j=1}^{q} (a_{ij} - \mu_A)^2 \sum_{i=1}^{p} \sum_{j=1}^{q} (b_{ij} - \mu_B)^2}, \tag{13}$$

$$\mathrm{SNR}(A, B) = -10 \log_{10} \frac{\sum_{i=1}^{p} \sum_{j=1}^{q} (a_{ij} - b_{ij})^2}{\sum_{i=1}^{p} \sum_{j=1}^{q} a_{ij}^2}. \tag{14}$$

In (13), the quantities μ_A and μ_B denote the mean value of A and B, respectively.

Example 5 The "peppers" RGB color image and its transformation to the $I_1I_2I_3$, HSI, and L*a*b* color spaces, rendered as false color RGB images, are displayed in the first four columns of row one of Fig. 7. In the 2nd row below each color image, composed thresholded fraction maps selected from $\overline{W} \cup \overline{M}$, depict the segmentation obtained in the corresponding color space, e.g., vectors and fraction thresholds used in RGB color space were $\overline{\mathbf{w}}^1(u_1 = 0.454)$, $\overline{\mathbf{w}}^2(u_2 = 0.363)$, and $\overline{\mathbf{m}}^1(u_1 = 1.561)$; similarly, for the $I_1I_2I_3$ color space, $\overline{\mathbf{m}}^3(u_3 = 0.389)$, $\overline{\mathbf{w}}^3(u_3 = 0.384)$, and $\overline{\mathbf{w}}^1(u_1 = 0.347)$ were chosen. The 3rd row displays Sobel gradient edge images corresponding to the segmentation produced by the WM method in the RGB and $I_1I_2I_3$ color spaces, a clustering method based on Mahalanobis distance, and a hybrid technique employing histograms and morphological watersheds. The 5th column of Fig. 7 shows from top to bottom, the NTSC grayscale version of the original color image, a 16-level quantization produced by an optimized octree nearest color algorithm, and its corresponding Sobel edge image used as reference for quantitative comparisons (see Table 4).

Table 4 Segmentation performance for the "peppers" color image

Segmentation method	Corr. coef. (ρ)	SNR
(1) WM in RGB	0.707	14.179
(2) WM in $I_1I_2I_3$	0.717	14.931
(3) WM in HSI	0.708	14.124
(4) WM in L*a*b*	0.675	14.006
(5) Mahalanobis distance clustering	0.632	12.917
(6) Histograms + Morph. Watersheds	0.594	9.814

Example 6 Figure 8 displays the segmentation results of additional color images. In each row, the source color image in RGB format is shown to the left; to the right, shown as quantized grayscale images, follows the segmentation obtained in the RGB, $I_1I_2I_3$, HSI, and L*a*b* color spaces. For example, the corresponding "bear" grayscale image in the $I_1I_2I_3$

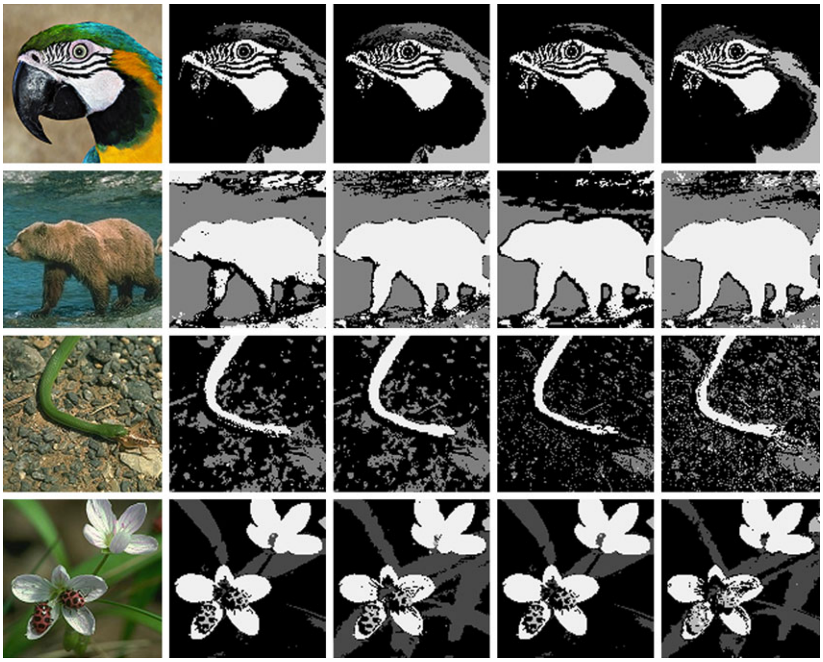

Fig. 8 *1st column*: sample RGB color images; *2nd* to *5th cols.*: compound segmented images obtained with the WM method, respectively, in the RGB, $I_1I_2I_3$, HSI, and L*a*b* color spaces (main regions of interest are quantized)

color space (2nd row, 3rd column) was generated by composing the fraction maps obtained from $\overline{\mathbf{w}}^2$ and $\overline{\mathbf{m}}^2$ after thresholding at low values, respectively, setting $u_2 = 0.387$ and $u_2 = 0.326$.

Based on the example images given here and the performance measure values listed in Table 4, the best segmentation results produced by applying the WM method and semi-constrained LM model occur in the $I_1I_2I_3$ space (cf. again 2nd column of Fig. 7 and 3rd column of Fig. 8).

5 Conclusions

This research work describes a novel pixel based segmentation method for color images in different color spaces based on the W and M lattice auto-associative memories, whose scaled column vectors defines a small finite set of saturated color pixels. These extreme points may form different suitable base sets to perform semi-constrained linear unmixing to determine color fractions of any other pixel in a given input image. Granular segmented images of all saturated pixels are directly produced by scaling the fraction data computed with the LLS or NNLS numerical methods, and coarse segmented images can be obtained by thresholding the corresponding color fraction maps. Examples using synthetic and real RGB color images are given to illustrate visually the results of segmentation. Table 3 summarizes the computational performance of the LAAMs+LLS, the c-means, and the fuzzy c-means techniques, from which the main advantage of the proposed technique is the reduction of processing times due to its non-iterative nature. Similarly, Table 4 gives the computational performance of the LAAMs+NNLS technique in four different color spaces by quantifying the difference between Sobel edge images of segmented grayscale images using the correlation coefficient and the signal to noise ratio. Specifically, color image segmentation carried out in the $I_1I_2I_3$ color space outperformed the results obtained in the RGB space when using a clustering technique based on the Mahalanobis distance between pixels, and a hybrid technique based on histograms and morphological watersheds. We point out that the lattice algebra based technique presented here has been applied so far to still images and further developments are needed for its application to real-time color image segmentation.

Acknowledgements Gonzalo Urcid-S. is grateful with the National System of Researchers (SNI-CONACYT) in Mexico city for partial financial support through grant # 22036; Juan Carlos Valdiviezo-N. thanks the National Council of Science and Technology (CONACYT) in Mexico city for doctoral scholarship # 175027. The authors thank also the anonymous reviewers for their valuable suggestions.

References

1. Ballard, D.H., Brown, C.M.: Computer Vision, pp. 149–150. Prentice Hall, Englewood Cliffs (1982)
2. Haralick, R.M., Shapiro, L.G.: Glossary of computer vision terms. In: Dougherty, E.R. (ed.) Digital Image Processing Methods, p. 439. Dekker, New York (1994)
3. Jain, R., Kasturi, R., Schunck, B.G.: Machine Vision, pp. 73–76. McGraw-Hill, New York (1995)
4. Awcock, G.J., Thomas, R.: Applied Image Processing, pp. 126–129. McGraw-Hill, New York (1996)
5. Pal, N.R., Pal, S.K.: A review on image segmentation techniques. Pattern Recognit. **26**(9), 1277–1294 (1993)

6. Zhu, S.C., Yuille, A.: Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **18**(9), 884–900 (1996)
7. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
8. Chan, T.F., Vese, L.A.: Active contours without edges. IEEE Trans. Image Process. **10**(2), 266–277 (2001)
9. Skarbek, W., Koschan, A.: *Colour image segmentation: a survey*, pp. 1–81. Technical Report 94-32, Technical University of Berlin (1994)
10. Plataniotis, K.N., Venetsanopoulos, A.N.: Color Image Processing and Applications, pp. 237–273. Springer, Berlin (2000)
11. Lucchese, L., Mitra, S.K.: Color image segmentation: A-state-of-the-art-survey. Proc. Indian Natl. Sci. Acad. **67**(2), 207–221 (2001)
12. Cheng, H.D., Jain, X.H., Sun, Y., Wang, J.: Color image segmentation: advances and prospects. Pattern Recognit. **34**(12), 2259–2281 (2001)
13. Celenk, M., Uijt de Haag, M.: Optimal thresholding for color images. In: SPIE Proc., Nonlinear Image Processing IX, San Jose, CA, vol. 3304, pp. 250–259 (1998)
14. Shafarenko, L., Petrou, H., Kittler, J.: Histogram-based segmentation in a perceptually uniform color space. IEEE Trans. Image Process. **7**(9), 1354–1358 (1998)
15. Meyer, F.: Color image segmentation. In: IEEE Proc., 4th Inter. Conf. on Image Processing and Its Applications, pp. 303–306 (1992)
16. Crespo, J., Schafer, R.W.: The flat zone approach and color images. In: Serra, J., Soille, P. (eds.) Mathematical Morphology and Its Applications to Image Processing, pp. 85–92. Kluwer Academic, Dordrecht (1994)
17. Liu, J., Yang, Y.-H.: Multiresolution color image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **16**(7), 689–700 (1994)
18. Park, S.H., Yun, I.D., Lee, S.U.: Color image segmentation based on 3-D clustering: morphological approach. Pattern Recognit. **31**(8), 1061–1076 (1998)
19. Géraud, T., Strub, P.-Y., Darbon, J.: Color image segmentation based on automatic morphological clustering. In: IEEE Proc., Inter. Conf. on Image Processing, Thessaloniki, Greece, vol. 3, pp. 70–73 (2001)
20. Healey, G.E.: Using physical color models in 3-d machine vision. In: SPIE Proc., Perceiving, Measuring and Using Color, San Diego, CA, vol. 1250, pp. 264–275 (1990)
21. Klinker, G.J., Schafer, S.A., Kanade, T.: A physical approach to color image understanding. Int. J. Comput. Vis. **4**(1), 7–38 (1990)
22. Sowmya, B., Sheelanari, B.: Color image segmentation using soft computing techniques. Int. J. Soft Comput. Appl. **4**, 69–80 (2009)
23. Essaqote, H., Zahid, N., Haddaoui, I., Ettouhami, A.: Color image segmentation based on new clustering algorithm and fuzzy eigenspace. Res. J. Appl. Sci. **2**(8), 853–858 (2007)
24. Palus, H., Kotyczka, T.: Evaluation of colour image segmentation results. In: Colour Image Processing Workshop, Erlangen, Germany (2001)
25. Gonzalez, R.C., Woods, R.E.: Digital image processing, 3rd edn., pp. 443–446. Pearson Prentice-Hall, Upper Saddle River (2008)
26. Zhang, C., Wang, P.: A new method for color image segmentation based on intensity and hue clustering. In: IEEE Proc., 15th Inter. Conf. on Pattern Recognition, vol. 3, pp. 613–616 (2000)
27. Palus, H.: Color image segmentation: selected techniques. In: Lukac, R., Plataniotis, K.N. (eds.) Color Image Processing: Methods and Applications, pp. 103–128. CRC Press, Boca Raton (2006)
28. Koschan, A., Abidi, M.: Digital Color Image Processing, pp. 149–174. Wiley, Hoboken (2008)
29. Maragos, P.: Lattice image processing: a unification of morphological and fuzzy algebraic systems. J. Math. Imaging Vis. **22**, 333–353 (2005)
30. Kaburlasos, V.G., Ritter, G.X. (eds.): Computational Intelligence Based on Lattice Theory, vol. 67. Springer, Heidelberg (2007)
31. Ham, F.M., Kostanic, I.: Principles of Neurocomputing for Science and Engineering. McGraw-Hill, New York (1998)
32. Lawson, C.L., Hanson, R.J.: Solving Least Squares Problems. Prentice-Hall, Englewood Cliffs (1974), Chap. 23
33. Urcid, G., Valdiviezo-N., J.C.: Color image segmentation based on lattice auto-associative memories. In: IASTED Proc., 13th Inter. Conf. on Artificial Intelligence and Soft Computing, Palma de Mallorca, Spain, pp. 166–173 (2009)
34. Urcid, G., Valdiviezo-N., J.C., Ritter, G.X.: Lattice associative memories for segmenting color images in different color spaces. In: Lecture Notes in Artificial Intelligence, vol. 6077 (Part II), pp. 359–366. Springer, Berlin (2010)
35. Cuninghame-Green, R.: Minimax, Algebra. Lectures Notes in Economics and Mathematical Systems, vol. 166. Springer, New York (1979)
36. Cuninghame-Green, R.: Minimax algebra and applications. In: Hawkes, P. (ed.) Advances in Imaging and Electron Physics, vol. 90, pp. 1–121. Academic Press, New York (1995)
37. Ritter, G.X., Sussner, P., Diaz de Leon, J.L.: Morphological associative memories. IEEE Trans. Neural Netw. **9**(2), 281–293 (1998)
38. Ritter, G.X., Urcid, G., Iancu, L.: Reconstruction of patterns from noisy inputs using morphological associative memories. J. Math. Imaging Vis. **19**(2), 95–111 (2003)
39. Urcid, G., Ritter, G.X.: Kernel computation in morphological associative memories for grayscale image recollection. In: IASTED Proc., 5th Int. Conf. on Signal and Image Processing, Honolulu, HI, pp. 450–455 (2003)
40. Ritter, G.X., Gader, P.: Fixed points of lattice transforms and lattice associative memories. In: Hawkes, P. (ed.) Advances in Imaging and Electron Physics, vol. 144, pp. 165–242. Elsevier, San Diego (2006)
41. Urcid, G., Ritter, G.X.: Noise masking for pattern recall using a single lattice matrix associative memory. In: Kaburlasos, V.G., Ritter, G.X. (eds.) Computational Intelligence Based on Lattice Theory, vol. 67, pp. 79–98. Springer, Heidelberg (2007)
42. Valle, M.E.: A class of sparsely connected autoassociative morphological memories for large color images. IEEE Trans. Neural Netw. **20**(6), 1045–1050 (2009)
43. Keshava, N.: A survey of spectral unmixing algorithms. Linc. Lab. J. **14**(1), 55–78 (2003)
44. Graña, M., Sussner, P., Ritter, G.X.: Associative morphological memories for endmember determination in spectral unmixing. In: IEEE Proc., Inter. Conf. on Fuzzy Systems, pp. 1285–1290 (2003)
45. Graña, M., Jiménez, J.L., Hernández, C.: Lattice independence, autoassociative morphological memories and unsupervised segmentation of hyperspectral images. In: Proc. 10th Joint Conf. on Information Sciences, pp. 1624–1631 (2007)
46. Valdiviezo, J.C., Urcid, G.: Hyperspectral endmember detection based on strong lattice independence. In: SPIE Proc., Applications of Digital Image Processing XXX, San Diego, CA, vol. 6696, pp. 1–12 (2007)
47. Ritter, G.X., Urcid, G., Schmalz, M.S.: Autonomous single-pass endmember approximation using lattice auto-associative memories. Neurocomputing **72**(10–12), 2101–2110 (2009)
48. Graña, M., Villaverde, I., Maldonado, J.O., Hernández, C.: Two lattice computing approaches for the unsupervised segmentation of hyperspectral images. Neurocomputing **72**(10–12), 2111–2120 (2009)
49. Ritter, G.X., Urcid, G.: Lattice algebra approach to endmember determination in hyperspectral imagery. In: Hawkes, P. (ed.) Ad-

vances in Imaging and Electron Physics, vol. 160, pp. 113–169. Elsevier, Burlington (2006)

50. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proc. 5th Berkeley Symposium on Mathematics, Statistics, and Probabilities, vol. I, pp. 281–297. University of California, Berkeley (1967)
51. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, New York (2000)
52. Elomaa, T., Koivistoinen, H.: On autonomous K-means clustering. In: Proc. 15th Int. Symposium on Methodologies for Intelligent Systems, pp. 228–236 (2005)
53. Bezdeck, J.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum, New York (1982)
54. Lim, Y.W., Lee, S.U.: On the color image segmentation algorithm based on the thresholding and fuzzy c-means techniques. Pattern Recognit. **23**(9), 935–952 (1990)

Gonzalo Urcid received his B.E. (1982) and M.Sc. (1985) both from the University of the Americas in Puebla, Mexico, and his Ph.D. (1999) in optics from the National Institute of Astrophysics, Optics, and Electronics (INAOE) in Tonantzintla, Mexico. He holds the appointment since 2001 of National Researcher from the Mexican National Council of Science and Technology (SNI-CONACYT), and currently is an Associate Professor in the Optics Department at INAOE. His present research interests include applied mathematics, digital and optical image processing, artificial neural networks, and pattern recognition.

Juan-Carlos Valdiviezo-N. is a Ph.D. student in the Optics Department at INAOE, Mexico. He received a B.E. (2005) in electronics engineering from the Tuxtla Institute of Technology in Chiapas, Mexico, and an M.Sc. (2007) in optics from the National Institute of Astrophysics, Optics, and Electronics (INAOE) in Tonantzintla, Mexico. He is the recipient of a Mexican National Council of Science and Technology Fellowship (CONACYT) since 2005, and currently is a member of the mexican SPIE and OSA student chapters. His research interests include digital image processing, spectral analysis, and artificial neural networks.

Gerhard X. Ritter received the B.A. (1966) and Ph.D. (1971) degrees from the University of Wisconsin, Madison. He is currently Professor of Computer Science of the Computer and Information Science and Engineering Department, the Director of the Center for Computer Vision and Visualization, and Professor of Mathematics at the University of Florida. Dr. Ritter is the Chair of the Society of Industrial and Applied Mathematics (SIAM) Activity Group in Imaging Science and of the American Association of Engineering Societies (AAES) R & D task Force. He is the Editor-in-Chief of the Journal of Mathematical Imaging and Vision, and a member of the Editorial Boards for both the Journal of Electronic Imaging and the Journal of Pattern Analysis and Applications. Since 1995 he is a Fellow of SPIE and he was the recipient of the 1998 General Ronald W. Yates Award for Excellence in Technology Transfer, Air Force Research Laboratory and the 1989 International Federation for Information Processing (IFIP) Silver Core Award. He is the author of two books and more than 100 refereed publications in computer vision, image algebra, mathematics, and neural networks. His current research interests include mathematical foundations of digital image processing and computer vision, artificial neural networks, and pattern recognition.