Fully Constrained Linear Spectral Unmixing: Analytic Solution Using Fuzzy Sets

José Luis Silván-Cárdenas and Le Wang

***Abstract*—The linear mixture model is a convenient way to describe image pixels as a linear combination of pure spectra – termed endmembers. The fractional contribution from each endmember is calculated through inversion of the linear model. Despite the simplicity of the model, a nonnegativity constraint that is imposed on the fractions leads to an unmixing problem for which it is hard to find a closed analytical solution. Current solutions to this problem involve iterative algorithms, which are computationally intensive and not appropriate for unmixing large number of pixels. This paper presents an algorithm to build fuzzy membership functions that are equivalent to the least square solution of the fully constrained linear spectral unmixing problem. The efficiency and effectiveness of the proposed solution is demonstrated using both simulated and real data.**

***Index Terms*—Fuzzy sets (FSs), linear spectral unmixing (LSU), subpixel fractional cover.**

I. Introduction

LINEAR mixture models assume that mixed spectra from remote sensing images are produced as linear combinations of signatures from distinct materials (endmembers) that are present in the sensor's instantaneous field of view [1]–[3]. The relative contribution of each endmember is, in principle, proportional to its planimetric area, and the process of estimating the relative cover of each material is referred to as linear spectral unmixing (LSU). LSU methods are important for hyperspectral image analysis and have been shown to be effective for the quantification of the relative abundance of materials [4], land cover proportions [5], and species composition [6], [7], or as a precursor for spatial resolution enhancement [8], [9].

For endmember fractions to be meaningful, they must sum to one (sum constraint) and be nonnegative (nonnegativity constraint). The simultaneous consideration of these two constraints during the inversion of the mixture model is termed the fully constrained LSU (FCLSU) problem. Researchers have noted that the sum constraint is easy to implement, but the nonnegativity constraint is not. As a result, most studies implement only the sum constraint and then apply the nonnegativity constraint to the result. Hu *et al.* [10] compared the Lagrange multiplier (LM) optimization method that is proposed in [5] for solving the partially constrained problem and a quadratic programming (QP) procedure for solving the FCLSU problem [11], [12]. The study concluded that the partially constrained approach was preferred over the fully constrained approach based on the computational demands of the QP algorithm alone.

On the other hand, it has been shown that the solution of the partially constrained problem generally yields suboptimal approximations to actual mixing fractions [4]. Because of this, several studies have sought to develop efficient algorithms to solve the fully constrained problem [4], [13], [14]. However, existing solutions involve iterative algorithms, which are computationally intensive and nonsuitable for large-scale applications. Heinz and Chang [4] proposed a solution that is based on the LM optimization method. In that method, the sum constraint is added as an additional equation to the mixture model, and the solution is determined through an iterative algorithm for the nonnegative-constrained problem [13]. In the latter case, the inequalities of the nonnegativity constraint are transformed into equality constraints through slack variables, and the minimization of the residual is carried out through LM optimization. Unfortunately, the adoption of these kinds of approaches has proved unfeasible for large-scale applications. This is made evident by the fact that one of the most popular pieces of software for remote sensing image analysis (ENVI by the ITT Visual Information Solutions, Inc.) does not include a solution for the fully constrained problem. Instead, it provides only a solution to the partially constrained problem with the sum constraint.

This paper shows by construction that the optimal solution to the FCLSU problem can be conveniently expressed in terms of fuzzy set (FS) operations, which results in a highly efficient method when the number of endmembers is less than seven. A summary of relevant definitions from the FS theory and the linear mixture models is first provided in Section II. Then, in Section III, the explicit solution of the FCLSU problem is derived and expressed in terms of FS operations. The FS algorithm is summarized in Section IV. The efficiency of this method was tested and compared with that of a traditional QP algorithm [11] and the partially constrained methods that are available in the ENVI software. The results from the tests are summarized in Section V. A summary of the contribution is provided in Section VI.

Manuscript received December 18, 2009; revised March 11, 2010 and August 20, 2010. Date of current version October 27, 2010. This work was supported in part by the Geography and Geomatic Research Center (CentroGeo). The work of L. Wang was supported in part by the National Science Foundation under Grants BCS-0822489 and DEB-0810933 and in part by the National Key Basic Research and Development Program, China, under Grant 2006CB701304.

J. L. Silván-Cárdenas is with the Geography and Geomatic Research Center (CentroGeo), 14220 Mexico City, Mexico (e-mail: jlsilvan@centrogeo.org.mx).

L. Wang is with the Department of Geography, College of Arts and Sciences, State University of New York, Buffalo, NY 14261 USA (e-mail: lewang@buffalo.edu).

Digital Object Identifier 10.1109/TGRS.2010.2072931

0196-2892/$26.00 © 2010 IEEE

II. Theoretical Considerations

A. FSs

FSs are generalizations of classic (crisp) sets to account for partial membership [15]. Let $\mathcal{F}$ denote the set of FSs in $\mathbb{R}^n$. For any $A \in \mathcal{F}$, a *membership function* $\mu_A : \mathbb{R}^n \mapsto [0,1]$ is defined, so that $\mu_A(x)$ is the grade of membership of x in A. Since the indicator functions of classic sets are special cases of the membership functions, classic sets are also in $\mathcal{F}$.

For two FSs, i.e., A and B, the intersection set is denoted by $A \cap B$, and its membership function is defined as $\mu_{A\cap B} = \mu_A \wedge \mu_B = \min(\mu_A, \mu_B)$. Likewise, the union set is denoted by $A \cup B$, and its membership function is defined as $\mu_{A\cup B} = \mu_A \vee \mu_B = \max(\mu_A, \mu_B)$. The compliment set of A is denoted by $\overline{A}$, and its membership function is defined as $\mu_{\overline{A}} = \neg\mu_A = 1 - \mu_A$. The difference set is denoted by $A - B$, and its membership function is given by $\mu_{A-B} = \mu_A \wedge \neg\mu_B$. The operators $\vee$, $\wedge$, and $\neg$ are referred to as *conjunction*, *disjunction*, and *negation*, respectively. A function $\gamma : [0,1]^m \mapsto [0,1]$ that is expressed in terms of these operators is referred to as a Boolean formula. The set $K(A) = \{x|\mu_A(x) = 1\}$ is called the *kernel* of A. The set $S(A) = \{x|\mu_A(x) > 0\}$ is called the *support* of A. For the two FSs A and B, the crisp set $L(A,B) = \{x|\mu_A(x) = \mu_B(x)\}$ is referred to as the *projection set* of A onto B.

Let $B = \Gamma(A_1,\ldots,A_m)$ be a composite set from the FSs $A_1,\ldots,A_m$ through $\Gamma : \mathcal{F}^m \mapsto \mathcal{F}$. Then, B has a membership of the form

$$\mu_B = \gamma\left(\mu_{A_1},\ldots,\mu_{A_m}\right) \tag{1}$$

where $\gamma : [0,1]^m \mapsto [0,1]$ is a Boolean formula with the same structure as Γ but with set operators (union, intersection, and compliment) that are replaced by Boolean operators (conjunction, disjunction, and negation, respectively). Furthermore, if Γ does not include negation operators, then the kernel and support of B are given by

$$K(B) = \Gamma\left(K(A_1),\ldots,K(A_m)\right) \tag{2}$$

$$S(B) = \Gamma\left(S(A_1),\ldots,S(A_m)\right). \tag{3}$$

A special kind of FSs is comprised by fuzzy half-spaces. Let the hyperplane $h : w^Tx - b = 0$ with normal vector $w \in \mathbb{R}^n$ and bias parameter $b \in \mathbb{R}$. h splits $\mathbb{R}^n$ in three regions: 1) the plane itself; 2) the left-hand side; and 3) the right-hand side. A point $x_0 \in \mathbb{R}^n$ is said to lie on the left half-space if $w^Tx_0 - b < 0$, or it is said to lie on the right half-space if $w^Tx_0 - b > 0$. Hence, the FS H that is specified by a membership function of the form $\mu_\mathrm{H} = \mu(w^Tx - b)$, where

$$\mu(\lambda) = \begin{cases} 1, & \text{for } \lambda \leq 0 \\ 1-\lambda, & \text{for } 0 < \lambda < 1 \\ 0, & \text{for } \lambda \geq 1 \end{cases} \tag{4}$$

is considered here as a fuzzy half-space in $\mathbb{R}^n$. Note that the support and the kernel of H are completely specified by the function $\lambda_\mathrm{H}(x) = w^Tx - b$. Specifically, $x_0 \in K(H)$ if $\lambda_\mathrm{H}(x_0) \leq 0$, and $x_0 \in S(H)$ if $\lambda_\mathrm{H}(x_0) < 1$.

For a given Boolean formula $\gamma : [0,1]^m \mapsto [0,1]$, the membership function of (4) fulfills the following property:

$$\gamma\left(\mu(\lambda_1),\ldots,\mu(\lambda_m)\right) = \mu\left(\bar{\gamma}(\lambda_1,\ldots,\lambda_m)\right) \tag{5}$$

where $\bar{\gamma}$ has the same structure as γ but with conjunction operators replaced by disjunction operators and vice versa. This property can be shown by induction and using the shorter form $\mu(\lambda) = 0 \vee (1 \wedge \neg\lambda)$ for (4).

For a given FS formula $\Gamma : \mathcal{F}^m \mapsto \mathcal{F}$ and its corresponding Boolean formula $\gamma : [0,1]^m \mapsto [0,1]$, the aggregate set $H = \Gamma(H_1,\ldots,H_m)$ of half-spaces H_i with membership functions $\mu_i = \mu(w_i^Tx - b_i) = \mu(\lambda_i)$ for $i = 1,\ldots,p$ has a membership function

$$\mu_\mathrm{H} = \mu\left(\bar{\gamma}(\lambda_1,\ldots,\lambda_m)\right) \tag{6}$$

which follows from (1) and (5). More explicitly, we have

$$\mu_\mathrm{H}(x) = \begin{cases} 1, & x \in K(H) \\ 1-\lambda_1, & x \in (S(H) - K(H)) \bigcap L(H,H_1) \\ \ldots & \\ 1-\lambda_m, & x \in (S(H) - K(H)) \bigcap L(H,H_m) \\ 0, & x \in \overline{S(H)} \end{cases} \tag{7}$$

where the transition region that is defined by the difference set $S(H) - K(H)$ has been split through the projection sets $L(H,H_i)$ that were defined earlier. The aggregate set H shall be considered a fuzzy half-space if $K(H)$, $\overline{S(H)}$, and $S(H) - K(H)$ are all nonempty sets.

B. LSU

Consider an image pixel $p = [p_1,\ldots,p_n]^T$ with n spectral bands. Assume that the spectrum in each pixel is a convex combination of m spectrally distinct materials or endmembers with reflectance spectra $e_1, e_2,\ldots,e_m$ covering the corresponding ground area of the pixel. The fractions of the ground area that is covered by these endmembers are denoted by $f_1, f_2,\ldots,f_m$, respectively. Then, the mixed pixel is expressed as

$$p = Ef + \varepsilon \tag{8}$$

where $f = [f_1,\ldots,f_m]^T$ is a column vector of subpixel fractional cover; $E = [e_1,\ldots,e_m]$ is an $n \times m$ matrix containing the m endmember spectra as column vectors; and ε is the residual that is not explained by the linear model. If p and E are known, f is usually estimated through a least square (LS) optimization. Specifically, f is estimated by solving the following optimization problem:

$$\min_f (p - Ef)^T(p - Ef) \tag{9}$$

that is subject to

$$i. \quad l^Tf = 1$$

$$ii. \quad f \geq 0$$

where l is an m-length column vector of ones, and 0 is an m-length column vector of zeroes. The objective function in

(9) defines spectral similarity in terms of the Euclidean norm. A generalization of the FCLSU problem that considers other measures of the spectral similarity can be found in [14]. In this paper, the Euclidean norm is adopted as a measure of spectral similarity.

The theory of convex geometry shows that a unique solution exists to the optimization problem of (9), provided that $m \leq n+1$, and the differences among endmembers form the basis of the $(m-1)$-dimensional space. It also follows that the solution is error free for points that are inside the convex hull of the endmembers. When constraints are ignored, the unconstrained LS (ULS) solution is written as [16]

$$f_{\mathrm{ULS}} = M^{-1}E^T p \tag{10}$$

where $M = E^T E$. If the sum constraint is considered alone, it is also possible to find an analytical solution through LM optimization [4], [5]. The sum-constrained LS (SLS) solution can be expressed as [16]

$$f_{\mathrm{SLS}} = f_{\mathrm{ULS}} - \frac{l^T f_{\mathrm{ULS}} - 1}{l^T M^{-1} l} M^{-1} l. \tag{11}$$

The fully constrained LS optimization problem previously mentioned can be alternatively written as the QP optimization problem [4], i.e.,

$$\min_f \left(\frac{1}{2} f^T M f + c^T f\right) \tag{12}$$

that is subject to

$$\begin{aligned} &i. \quad A_{\mathrm{eq}} f = b_{\mathrm{eq}} \\ &ii. \quad Af \geq b \end{aligned}$$

where $c = -E^T p$, $A_{\mathrm{eq}} = l$, $b_{\mathrm{eq}} = 1$, $A = -I$, and $b = 0$. In this paper, a QUADPROG function in the optimization toolbox of the software package MATLAB (The MathWorks, Inc. 2002) was used as follows:

$$f_{\mathrm{QP}} = \mathrm{QUADPROG}(M, c, A, b, A_{\mathrm{eq}}, b_{\mathrm{eq}}, 0, l, f_{ULS}) \tag{13}$$

where the last three arguments define the lower and upper bounds of f and the starting solution, respectively. The latter was set to the ULS solution that is given by (10). The QUADPROG function calls a private function QPSUB of the optimization toolbox, which is a general-purpose code that is written in the MATLAB programming language for handling several forms of the general QP problem, i.e., unconstrained QP, QP with equality constraints only, etc. The QPSUB solves the problem in (12) using medium-scale optimization, as described in [12] and [17].

Since QP optimization can be computationally intensive, a reasonable alternative is to first calculate the SLS solution using (11) and then test whether any fractions are negative. If all the fractions fall within the unit interval, then the solution is found, and the search stops; otherwise, QP optimization is invoked. This method is hereafter referred to as partial QP (PQP).

III. FS Solution

This section shows that the optimal solution to the FCLSU problem can be expressed as $f_{\mathrm{FS}} = \mu_{\mathrm{H}}(p)$, where $\mu_{\mathrm{H}}(x) = [\mu_1(x), \ldots, \mu_m(x)]^T$ is composed of membership functions of fuzzy half-spaces. The explicit structure of each of these functions is

$$\mu_i(x) = \bigvee_{k=1}^{K_i} \bigwedge_{j \in J_k} \mu\left(w_j^{(i)} \cdot x - b_j^{(i)}\right) \tag{14}$$

for $i = 1, \ldots, m$; $x, w_j^{(i)} \in \mathbb{R}^n$; $b_j^{(i)} \in \mathbb{R}$; $K_i \in \mathbb{Z}$; $J_k \subset \mathbb{Z}$; and $\mu(\cdot)$ is the membership that is introduced in (4). The symbols $\bigvee$ and $\bigwedge$ in (14) denote the overall minimum and overall maximum of their arguments.

Without loss of generality, only the derivation of $f_1 = \mu_1(p)$ is shown. The derivation of all other fractions would follow an analogous procedure.

A. Preliminaries

Assuming that there are at least two endmembers, write the linear mixture model in the following form:

$$p = f_1 e_1 + f_2 e_2 + \sum_{i=3}^{m} f_i e_i + \varepsilon. \tag{15}$$

Then, subtract e_1 from both sides of the equation, and substitute f_2 by $1 - f_1 - f_3 - \cdots - f_m$ (from the sum constraint) to arrive at the following:

$$f_1(e_2 - e_1) = (e_2 - e_1) - (p - e_1) + \varepsilon + \sum_{i=3}^{m} f_i(e_i - e_2). \tag{16}$$

The last term on the right side of this equation must be taken as zero for $m = 2$. For $m > 2$, this term can be annihilated by multiplying both sides of the equation by an orthogonal vector to the subspace that is spanned by the base $\{e_i - e_2, \text{ for } i = 3, \ldots, m\}$. Such an orthogonal vector can be expressed as $e_1' - e_1$, where e_1' is the projection of the point e_1 onto the hyperplane h_1 containing all endmembers, except e_1. This step is similar to the orthogonal subspace projection approach that is discussed in [18]. Hence, the solution for f_1 can be written as

$$f_1 = 1 - \frac{(e_1' - e_1)^T (p - e_1 - \varepsilon)}{(e_1' - e_1)^T (e_2 - e_1)}. \tag{17}$$

Note that $(e_1' - e_1)^T (e_2 - e_1) = \|e_1' - e_1\|^2$ because both e_1' and e_2 are in h_1, and e_1' is the nearest point in h_1 to e_1. In particular, $e_1' = e_2$ for $m = 2$.

Now, define the following:

$$w_1 = \frac{e_1' - e_1}{\|e_1' - e_1\|^2}, \qquad b_1 = w_1^T e_1$$

so that $h_1 : w_1^T x = b_1 + 1$ is the hyperplane containing the points $e_2, \ldots, e_m$. Using these definitions, rewrite (17) in the

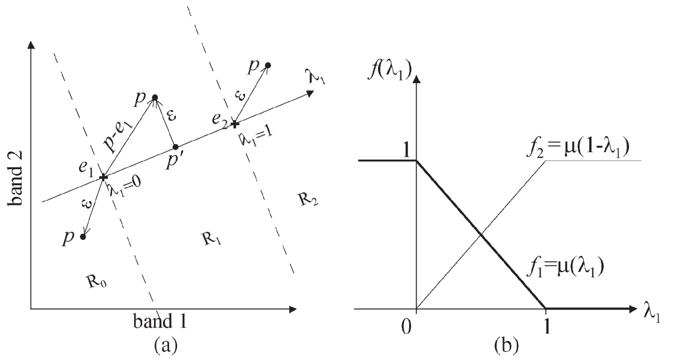


Fig. 1. Two endmembers in two dimensions (e_1 and e_2). (a) Partition of the mixing space with residual vector ε that is indicated for each region (R_0, R_1, and R_2) and the minimum distance axis λ_1. (b) Plot of fractions as a function of λ_1.

shorter form, i.e.,

$$f_1 = 1 - w_1^T(p - \varepsilon) + b_1. \tag{18}$$

The next step is to consider the form of ε. As stated earlier, the set of points that are defined by a linear mixture—with nonnegative fractions that sum to one—defines the convex hull of the set of endmembers. Hence, the minimization of $\|\varepsilon\|$ implies the search for a point $p' = p - \varepsilon$ that is inside the convex hull and is the nearest point to p. For points that are outside the convex hull, the nearest point is the orthogonal projection of point p onto the nearest facet of the hull, where a facet can be a vertex (0-facet), an edge (1-facet), a face (2-facet), a cell (3-facet), or r-facet in general for $r = 0, \ldots, m-2$.

Before deriving the solution for the general case, consider the cases of two and three endmembers.

B. Two Endmembers

As depicted in Fig. 1, the residual vector can have one of three forms, depending on the region where p is located. Each region represents the set of points that share a common nearest facet of the convex hull. By convention, here and in other examples, the boundary points are included in the region that is associated with the smallest number of vertices. The three regions can be characterized by means of the variable $\lambda_1 = w_1^T p - b_1$, varying along the line containing e_1 and e_2. This variable takes negative values for R_0, values between zero and one for R_1 (excluding the boundary points), and values that are greater than one for R_2.

Consider the definitions in Fig. 1(a) to write the explicit expression for the residual as

$$\varepsilon = \begin{cases} p - e_1, & \text{for } \lambda_1 \leq 0 \\ p - e_1 - \lambda_1(e_2 - e_1), & \text{for } 0 < \lambda_1 < 1 \\ p - e_2, & \text{for } \lambda_1 \geq 1. \end{cases} \tag{19}$$

Now, replace ε in (18) to obtain the following:

$$f_1 = \begin{cases} 1, & \text{for } \lambda_1 \leq 0 \\ 1 - \lambda_1, & \text{for } 0 < \lambda_1 < 1 \\ 0, & \text{for } \lambda_1 \geq 1. \end{cases} \tag{20}$$

Finally, use the definitions in (4) and λ_1 to write the following:

$$f_1 = \mu\left(w_1^T p - b_1\right). \tag{21}$$

A similar procedure can be used to show that $f_2 = \mu(\lambda_2)$, where $\lambda_2 = 1 - \lambda_1$. Fig. 1(b) shows a plot of these memberships as functions of λ_1.

C. Three Endmembers

For $m = 3$, there are seven possibilities for ε. Fig. 2(a) illustrates these seven possibilities for two bands. Once again, each region represents the set of points that share a common nearest facet of the convex hull.

Use the definitions of the regions in Fig. 2 to write the residual vector in the following form:

$$\varepsilon = \begin{cases} p - e_1, & \text{for } p \in R_0 \\ 0, & \text{for } p \in R_1 \\ p - e_1 - \lambda_2(e_2 - e_1), & \text{for } p \in R_2 \\ p - e_1 - \lambda_3(e_3 - e_1), & \text{for } p \in R_3 \\ p - e_2, & \text{for } p \in R_4 \\ (\lambda_1 - 1)\left(e_1' - e_1\right), & \text{for } p \in R_5 \\ p - e_3, & \text{for } p \in R_6 \end{cases} \tag{22}$$

where $\lambda_j = w_j^T p - b_j$ with w_j and b_j for $j = 1, 2, 3$ that are respectively being defined as

$$w_1 = \frac{e_1' - e_1}{\|e_1' - e_1\|^2}, \qquad b_1 = w_1^T e_1$$

$$w_2 = \frac{e_2 - e_1}{\|e_2 - e_1\|^2}, \qquad b_2 = w_2^T e_1$$

$$w_3 = \frac{e_3 - e_1}{\|e_3 - e_1\|^2}, \qquad b_3 = w_3^T e_1.$$

These variables (λ's) represent projection axes that are aligned along the vectors $e_1' - e_1$, $e_2 - e_1$, and $e_3 - e_1$, respectively [see Fig. 2(b)].

Now replace ε in (18) to obtain the following:

$$f_1 = \begin{cases} 1, & \text{for } p \in R_0 \\ 1 - \lambda_1, & \text{for } p \in R_1 \\ 1 - \lambda_2, & \text{for } p \in R_2 \\ 1 - \lambda_3, & \text{for } p \in R_3 \\ 0, & \text{for } p \in R_4 \cup R_5 \cup R_6. \end{cases} \tag{23}$$

In this case, f_1 follows the form in (7), i.e., it represents a fuzzy half-space in $\mathbb{R}^n$. As such, the fraction can be expressed in terms of the membership functions of the type in (4).

The next step consists of applying the property (2) to write f_1 in the more compact form of (6). The idea here is to find a formula for the support of f_1, i.e., the region for which $f_1 > 0$, in terms of the support of the fuzzy half-spaces that form it and then replace the set operators by the corresponding Boolean operators. As an example, consider Fig. 2(c). This figure indicates the support of f_1 (shaded region) and the boundary of the half-spaces that form it. The straight lines correspond to $h_j : w_j^T x = b_j + 1$, and the memberships $\mu_{H_j} = \mu(w_j^T x - b_j)$ define the fuzzy half-spaces H_j for

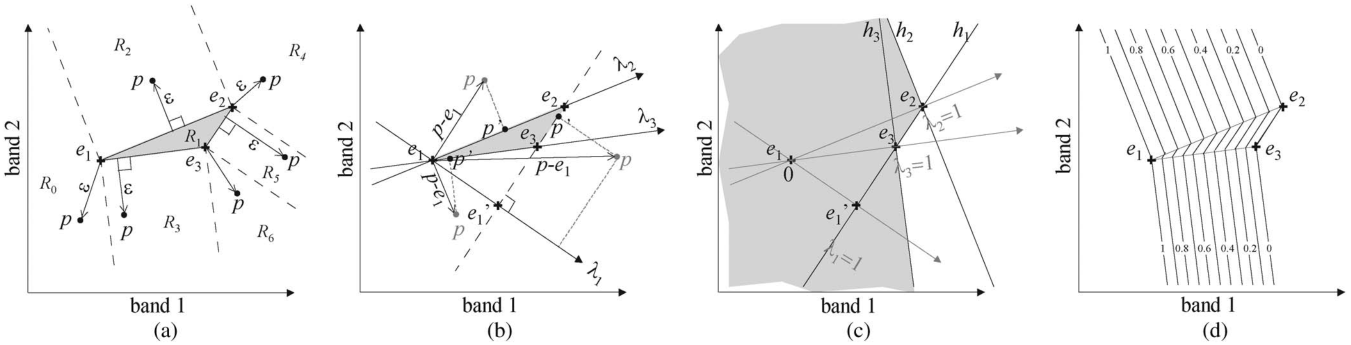


Fig. 2. Three endmembers in two dimensions. (a) Partition of the mixing space that is imposed by the minimization of $\|\varepsilon\|$. (b) Projection of $p - e_1$ along the minimum distance axes λ_1, λ_2, and λ_3. (c) Support for the fraction f_1 (shaded region) with the boundary straight lines ($\lambda_i = 1$, for $i = 1, 2, 3$). (d) Contour lines for the fraction f_1.

$i = 1, 2, 3$ that determine f_1. For this particular case, the support of f_1 can be expressed as

$$S(f_1) = [S(H_1) \cup S(H_3)] \cap S(H_2). \tag{24}$$

Therefore, the fraction results in the following:

$$f_1 = \left[\mu\left(w_1^T p - b_1\right) \vee \mu\left(w_3^T p - b_3\right)\right] \wedge \left(w_2^T p - b_2\right). \tag{25}$$

Fig. 2(d) shows the contour lines for the surface that is produced by (25). The contour lines make evident the transition region of the fuzzy half-space.

D. Any Number of Endmembers

As observed in previous cases, the minimization of $\|\varepsilon\|$ implies a partition of the mixing space into a number of regions, each of which shares a common nearest facet of the convex hull of endmembers. In general, there are $2^m - 1$ facets for $m \leq n + 1$. With so many facets and, hence, regions, it is hard to write an explicit formula for the residual vector of the general case. Instead, it is much easier to inspect the form of the fraction for the following three relevant regions: 1) the core or saturation region; 2) the support minus the core region, i.e., the transition region; and 3) the complement of the support, i.e., the null region. As noted in the following, each of such regions is associated with a distinct set of facets of the convex hull.

Let X denote the set of endmembers and 2^X denote the power set, excluding the empty set. Consider the partition of the power set $2^X = \{e_1\} \cup F \cup G$, where $F = \{A | A \in 2^X, e_1 \in A, |A| > 1\}$ and $G = \{A | A \in 2^X, e_1 \notin A\}$, so that $|F| = |G| = 2^{m-1} - 1$. As it turns out, the core of f_1 is associated with the facet $\{e_1\}$; its transition region is associated with the facets in F, and its support is associated with facets in G. Furthermore, consider an order in F and G, so that $F_j = G_j \cup \{e_1\}$ and $|G_j| \geq |G_k|$, for $j < k$, where F_j and G_j denote the jth facet in F and G, respectively. The ordering is selected, so that F_1 is the facet containing all the vertices and G_1 is the facet containing all the vertices but e_1, e.g., for three endmembers $F_1 = \{e_1, e_2, e_3\}$, $F_2 = \{e_1, e_2\}$, $F_3 = \{e_1, e_3\}$, $G_1 = \{e_2, e_3\}$, $G_2 = \{e_2\}$, and $G_3 = \{e_3\}$.

Now, if the nearest facet to p is e_1, then $p' = e_1 = p - \varepsilon$, and (18) results in $f_1 = 1$. If the nearest facet to p is in G, then p' lies on the hyperplane h_1 containing all endmembers, except e_1, i.e., $w_1^T p' = b_1 + 1$, and (18) results in $f_1 = 0$. If the nearest facet is in F, then there are two possibilities. The first one occurs if p is inside the convex hull, i.e., the nearest facet is F_1 and $\varepsilon = 0$, for which (18) results in $f_1 = 1 - \lambda_1$. The other possibility occurs if the nearest facet is F_j, for $j > 1$. The evaluation of f_1 in this case requires additional definitions.

Let $e_1^{(j)}$ denote the nearest point of e_1 in the subspace that is defined by the points in the facet G_j. If $|G_j| = 1$, the subspace is a point, and the nearest point is the subspace point itself. If $|G_j| > 1$, the nearest point is given by the orthogonal subspace projection as

$$e_1^{(j)} = e_1 + (I - DD^{\#})(e_1 - e) \tag{26}$$

where D is a base matrix that is formed by all linearly independent differences of the vertices in the facet G_j, $D^{\#}$ denotes the pseudo inverse of D, and e is any point in G_j. It can be shown that, regardless of its dimension, the subspace is contained in the hyperplane $h_j : w_j^T x = b_j + 1$ with the following parameters:

$$w_j = \frac{e_1^{(j)} - e_1}{\left\|e_1^{(j)} - e_1\right\|^2}, \qquad b_j = w_j^T e_1. \tag{27}$$

Furthermore, the set of such hyperplanes, for all facets in G, forms the support of f_1, and the function $\lambda_j(x) = w_j^T x - b_j$ takes values between zero and one for the points that are in the transition region of f_1.

Considering the aforementioned definitions, the projection of p onto the facet F_j for $j > 1$, is given by

$$p' = e_1 + \left(e_1^{(j)} - e_1\right)\lambda_j. \tag{28}$$

The substitution of this equation in (18) results in $f_1 = 1 - \lambda_j$, where the identity $(e_1' - e_1)^T(e_1^{(j)} - e_1) = \|e_1' - e_1\|^2$ has been used. The identity holds because e_1' is the nearest point in h_1 to e_1, and $e_1^{(j)}$ is also in h_1.

The result, considering all the cases together, can be written in the following form:

$$f_1 = \begin{cases} 1, & \text{for } p \in R_0 \\ 1 - \lambda_1, & \text{for } p \in R_1 \\ 1 - \lambda_2, & \text{for } p \in R_2 \\ \dots \\ 1 - \lambda_{2^{m-1}-1}, & \text{for } p \in R_{2^{m-1}-1} \\ 0, & \text{for } p \in \bigcup_{j=2^{m-1}}^{2^m} R_j \end{cases} \tag{29}$$

where R_0 is the region that has e_1 as the nearest facet; R_j is the region that has F_j as the nearest facet, for $j = 1, \dots, 2^{m-1} - 1$, or facet $G_{j-2^{m-1}+1}$ for $j = 2^{m-1}, \dots, 2^m$.

The last step is to express (29) in terms of the membership functions $\mu(\lambda_j)$ for $j = 1, \dots, 2^{m-1} - 1$. This is equivalent to building the support of f_1 in terms of the support of the half-spaces that are defined by $\lambda_j(x) < 1$. A general approach is to build the disjunctive normal form (DNF) for the Boolean formula. The idea behind a DNF is that the Boolean formula describes the support as a union of convex sets, which are in turn expressed as intersections of half-spaces.

Let K denote the number of convex sets of the support of f_1 and J_k denote the set of indexes of the half-spaces that are included in the kth convex set for $k = 1, \dots, K$. Then, the DNF formula for the fraction f_1 has the following form:

$$f_1 = \bigvee_{k=1}^{K} \bigwedge_{j \in J_k} \mu\left(w_j^T p - b_j\right). \tag{30}$$

The problem turns now to determine the number of convex sets K and the index sets J_k for $k = 1, \dots, K$. An algorithm to determine these follows.

E. Construction of the DNF Formula

The algorithm to build the Boolean formula in DNF requires at least three endmembers. As shown in a previous section, no Boolean formula is necessary for the case of two endmembers. The idea behind the algorithm that is presented here is to form the first convex set using only the hyperplanes that support the convex hull, i.e., the hyperplanes that do not intersect the interior of the convex hull. For each nonsupport hyperplane, a new convex region is created. The hyperplanes that are included in the new convex set are selected, depending on the position of the projection of e_1 with respect to the facet hyperplanes.

In formal grounds, $h_j : w_j^T x = b_j + 1$ is a *support hyperplane* of the convex hull if

$$w_j^T e_k - b_j \leq 1 \tag{31}$$

for all $k = 1, \dots, m$. This means that all the vertices of the convex hull lie on the left side of a support hyperplane. A *facet hyperplane* is a hyperplane that contains an $(m-1)$ facet of the convex hull, i.e., the hyperplane containing all but one vertex. As such, h_1 is a facet hyperplane but not h_2, h_3, etc.

Given the aforementioned definitions, the algorithm to build the DNF representation for the support of f_1 is summarized as follows:

```
1) Initialize $k = 1$ and $J_1 = \Phi$.
2) For each support hyperplane $h_j$, add $j$ to set $J_1$.
3) For each nonsupport hyperplane $h_j$,
    a) Update the counter $k \leftarrow k + 1$.
    b) Add $j$ to $J_k$.
    c) For each $h_l$, with $l \neq j$,
        i) If $e_1^{(l)}$ lies on the left-hand side, inclusive of all
           facet hyperplanes containing the facet $F_j$, then
           add $l$ to $J_k$.
```

As an example of this algorithm, consider the case in Fig. 2. As shown in Fig. 2(c), the lines h_1 and h_2 are support lines of the convex hull, and thus, $J_1 = \{1, 2\}$. For the nonsupport line h_3, a new convex set is added. The new convex set includes all the lines that have its nearest point to e_1 in the left side of all lines containing the facet $F_3 = \{e_3\}$, i.e., the lines containing the segments $\overline{e_1 e_3}$ and $\overline{e_2 e_3}$. Therefore, h_2 and h_3 are included because their nearest points to e_1 ($e_1'' = e_2$ and $e_1''' = e_3$) lie on the left side, whereas h_1 is excluded because e_1' lies on the other side of one of the facet lines. Then, $J_2 = \{2, 3\}$, and there are no more lines intersecting the convex hull. Using these sets in (30) results in the following:

$$f_1 = [\mu(\lambda_1) \wedge \mu(\lambda_2)] \vee [\mu(\lambda_2) \wedge \mu(\lambda_3)] \tag{32}$$

which can be reduced to (25).

IV. Spectral Unmixing as Multiple Boolean Formula Evaluations

The most straightforward method for spectral unmixing consists of independently building the Boolean formula for each fraction. Then, evaluate each formula for all the mixed pixels. The evaluation of all the formulas can start with the facet hyperplanes. If all the memberships that are associated to the facet hyperplanes fall within the interval (0, 1), then the optimal solution corresponds to such memberships, and no more hyperplane evaluations are required.

It should be noted that the time complexity for the calculation of the membership function is exponential in m because the method involves the calculation of up to $2^{m-1} - 1$ hyperplanes per fraction. However, for a fixed m, the time complexity for the unmixing is linear in the number of pixels because the unmixing consists of multiple evaluations of functions of the form of (14). This characteristic is desirable because, in most practical situations, m is relatively small, whereas the number of mixed pixels can be several thousands or millions. For a relatively large m, a more efficient strategy is desirable. One alternative is to sequentially estimate the fractions, taking advantage of the previously calculated fractions. The rationale of this method is given as follows:

Once f_1 has been obtained, one can subtract $f_1 e_1$ from both sides of (15) and divide by $1 - f_1$ to obtain the following:

$$\frac{p - f_1 e_1}{1 - f_1} = \sum_{i=2}^{m} f_i' e_i + \varepsilon' \tag{33}$$

where $f'_i = f_i/(1 - f_1)$ for $i = 2, \ldots, m$, and $\varepsilon' = \varepsilon/(1 - f_1)$. The problem is exactly the same as previously noted, with the f'_i being the new fractions. However, the number of endmembers is reduced by one, and the number of the hyperplanes that need to be considered is reduced by about one half. One can continue this process until, finally, $f_m = 1 - \sum_{i=1}^{m-1} f_i$ without any further checking. Therefore, instead of evaluating m Boolean formulas, each involving $2^{m-1} - 1$ hyperplanes, one evaluates the $m - 1$ formulas involving the $2^{m-1} - 1$ and $2^{m-2} - 1, \ldots, 3, 1$ hyperplanes.

The two methods that are previously described, which are hereafter referred to as FS and sequential FS (SFS), respectively, were coded in MATLAB v6.5 (The MathWorks, Inc. 2002) and tested for efficiency and sensitivity to numerical errors. Whenever possible, matrix operations were used in order to avoid the overhead of MATLAB loops. As such, the implementations are code compact but not memory compact. A memory-compact implementation would require the use of FOR loops that are time consuming in the MATLAB language. A more efficient memory-compact implementation could be attained with a lower level programming language, such as C or Fortran. In such cases, the evaluation of the Boolean formula could take the advantage of the zeroes and ones from partial evaluations, i.e., if at least one operand of $\bigwedge$ is zero, then all the min term evaluates to zero, or if at least one operand of $\bigvee$ is one, then all the max term evaluates to one. This way, not all the $2^{m-1} - 1$ hyperplanes need to be evaluated.

Since the computation of the hyperplanes that are involved in either aforementioned methods requires that no one endmember is *too close* to the mixing space that is spanned by the other endmembers or else the problem may become ill conditioned, i.e., not amenable for digital computation, the condition number of matrix $E^T E$ can be used for diagnostic purposes. The condition number of a matrix is calculated as the ratio of its maximal to its minimal singular values. A large condition number is indicative of an ill-conditioned problem.

V. Unmixing Tests

The CPU time for various unmixing tests was measured on a laptop computer (Pentium 4 at 2.8 GHz with 512 MB of random access memory), for which unnecessary network connections and background programs had been turned off. The ULS (10), SLS (11), QP (12), and PQP (which were described at the end of Section II-B) solutions were considered here as benchmarks for the proposed FS and SFS methods. All the methods were implemented in the MATLAB programming language. In addition to the MATLAB implementations, the sum-constrained solution, which is available in the commercial software ENVI 4.3 (ITT, Visual Information Solution, Inc.), was also applied. The details on the latter method are provided in [19].

A. Tests From the Simulated Data

Two tests were designed to assess the methods' efficiency as a function of the number of endmembers and the number of pixels. In addition, a test was designed to assess the methods' sensitivity to numerical errors. In all these cases, the mixed spectra were simulated using a previously published spectral library [20]. Sixteen reflectance spectra, which represent a variety of land cover types, were selected from the spectral library. These included three vegetation types (1-conifer, 2-deciduous, 3-grass), three soil types (4-gray silty clay, 5-reddish brown fine sandy loam, 6-dark yellowish brown micaceous loam), water (7-sea water), snow (8-medium granular snow), and eight man-made materials (9-construction concrete, 10-construction asphalt, 11-black gloss paint, 12-pine wood, 13-red smooth-faced brick, 14-aluminum metal, 15-galvanized steel metal, and 16-reddish asphalt roofing shingle). Seven resampled versions of these reflectance spectra were produced using filter functions of distinct satellite sensors. Sensors were selected to represent varying dimensions of the mixing space. These included SPOT1 $(n = 3)$, MSS5 $(n = 4)$, AVHRR7 $(n = 5)$, TM5 $(n = 6)$, SeaWiFS $(n = 8)$, ASTER $(n = 14)$, and MODIS $(n = 35)$ because band 36 fell outside the spectral range of the original spectral library. The endmember fractions were simulated as random numbers with a uniform distribution in the interval [0, 1] and normalized to sum to one. The mixed spectra were simulated through applying the linear mixing model with additive white Gaussian noise with standard deviation $\sigma = 0.1$. In all the tests that are described in the following, simulations were run five times with the same values for n, m, and k, but each time, new fractions and noise components were generated, and new endmembers were randomly selected.

1) Relative Efficiency: In the first test [see Fig. 3(a)], the MODIS pixels $(n = 35)$ were simulated as a mixture of four randomly selected endmembers $(m = 4)$, whereas the number of mixed pixels k varied. Since the memory limitations in MATLAB prevented the use of extremely large data sets, the number of mixed pixels varied as $k = 4^i$ for $i = 2, 3, \ldots, 9$. Hence, the largest simulated MODIS data set was equivalent to an image size of 512 × 512 pixels. Fig. 3(a) shows the median CPU time for unmixing the MODIS pixels with the four methods: the QP, PQP, FS, and SFS. The results indicated that the performance of the QP and the PQP was no substantially different. This is because the fraction of the mixed pixels that fall within the convex hull is generally low under the presence of noise, particularly when $m < n + 1$. Likewise, the performance of the SFS was comparable to that of the FS, with the exception of two extreme points [see Fig. 3(a)]: the smallest number of pixels and the largest number of pixels. The latter case is of most interest in this study. The increase in computer time by the SFS for the largest image size seemed to be caused by the fact that the memory that was used was near its limit. This is because the implementation of the SFS requires the storage of the partial sums of the previously calculated fractions and endmember time fractions, which are not required by the FS. Interestingly, the FS solution took about 1% of the computation time of the QP for unmixing no less than 2^{12} pixels (dotted line). When the number of pixels dropped below this number, this percentage increased, yet it remained below 50%.

In the second test [see Fig. 3(b)], the number of pixels was set to $k = 2^{12}$, and the number and the dimension of the endmembers varied as $m = 2, 3, \ldots, \min(10, n + 1)$, and $n = 3, 4, 5, 6, 8, 14, 35$ (i.e., all the aforementioned sensors),

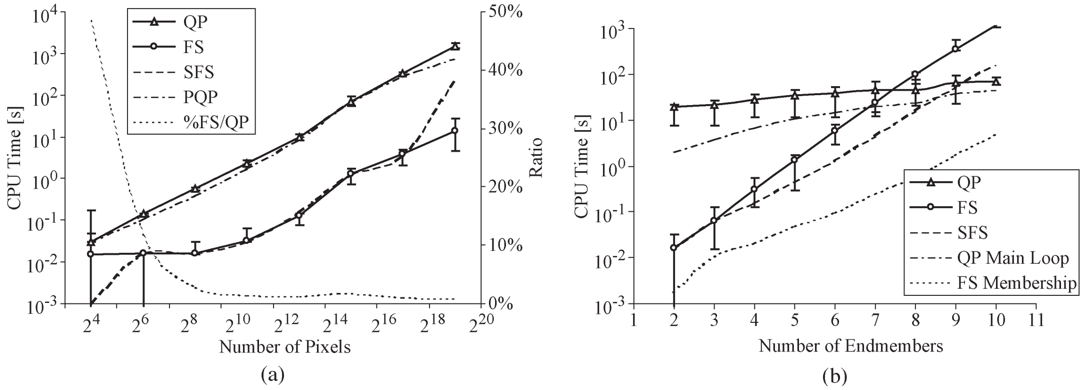


Fig. 3. CPU time using the QP, PQP, FS, and SFS algorithms. (a) The number of endmembers and bands are held fixed ($m = 4$ and $n = 35$, i.e., MODIS), whereas the number of mixed pixels varied ($k = 4^2, 4^3, \ldots, 4^9$). (b) The number of the mixed pixels was held constant ($k = 2^{12}$), whereas the number of endmembers and bands varied ($m = 2, 3, \ldots, \min\{10, n+1\}$ for $n = 3, 4, 5, 6, 8, 14$ and 35). Vertical bars in all cases represent the minimum and maximum CPU times obtained within multiple simulations, whereas all other lines, with the exception of the dotted line in (a), represent the median CPU time from a number of simulations (5–35). The dotted line in (a) represents the percentage of FS to QP CPU median times (use vertical axis to the right).

respectively. It should be noted that the commonality of the endmembers was not exploited by the general QP code that is used in these tests. In order to provide a fairer comparison between the computation times for the FS and QP methods, the CPU time for the main loop of the QP algorithm was also measured. This elapsed time excluded the repeated calculation of the initial active set, which did not vary for a given set of endmembers. Fig. 3(b) provides the median CPU time that is measured for the QP, FS, and SFS and the median elapsed time for the construction of the FS membership functions and for the QP main loop. When the number of endmembers increased, the computation time with FS increased faster than that with the QP. The FS method was more efficient than the QP optimization only for $m < 7$, while the SFS was even more efficient for $m < 9$. The gain in efficiency by the SFS with respect to the FS was more significant for $m > 4$. The results from the PQP were not included in Fig. 3(b) because no significant difference in computation time was observed with respect to the QP. Nonetheless, the CPU time of the main loop of the QP code was about 10% of the total time of the QP for $m = 2$, 40% for $m = 6$, and 70% for $m = 10$. Likewise, the CPU time for the calculation of the FS membership function was about 10% of the total time of FS for $m = 2$, 2% for $m = 6$, and 0.5% for $m = 10$.

2) Numerical Sensitivity: In the third test, the fractions that were derived from the PQP, FS, and SFS methods were compared against those that were derived from the QP method from a sample of $k = 100$ mixed spectra of varying numbers of endmembers and sensors, i.e., $m = 2, 3, \ldots, \min(9, n+1)$ for $n = 3, 4, 5, 6, 8, 14,$ and 35. Endmember matrices with condition numbers that are greater than 100 000 were excluded in the computation of error measures, which is mainly because both the QP and the PQP yielded not a number (NaN) values. Such larger condition numbers generally occurred for $m = n + 1$ and due to the relatively low spectral contrast among certain endmembers. There were 279 sets of simulated pixels ($k = 100$). For each set of simulated pixels, the maximum absolute difference (MAD) between the fractions that are estimated by each method and the QP method was calculated. The median and maximum statistics of the MAD are summarized in Table I.

TABLE I
COMPARISON OF THE PQP, FS, AND SFS WITH RESPECT TO QP. TABLE ENTRIES WERE BASED ON 279 SETS OF THE SIMULATED MIXED PIXELS WITH VARYING ENDMEMBERS AND BANDS (SEE THE TEXT)

Unmixing	Maximum Absolute Difference	
Method	Median	Maximum
QP	0	0
PQP	2.60e-14	7.06e-12
FS	5.37e-12	1.62e-02
SFS	7.45e-01	1.00e-00

Both the PQP and the FS were virtually the same solution as QP optimization. There were, however, a few cases where numerical errors caused slight deviations for the FS method (with maximum difference with respect to the QP fractions of 0.0162). This situation mainly occurred when the condition number was significantly large (i.e., nearly 100 000). In contrast, for the SFS, there were many cases with maximum error around 1. A close inspection of the results for the SFS showed that numerical errors, presumably due to normalization/denormalization of fractions, quickly increased with the number of endmembers, so that, for $m = 4$, the maximum MAD was about 0.45 and, for $m > 4$, the maximum MAD topped at 1.0.

B. Unmixing of Landsat Measurements

A fourth test was based on a Landsat ETM+ image (path 31, row 39) that was acquired on December 19, 2005. The ETM+ image comes with nine bands, of which only six optical bands (with a spatial resolution of 30 m) were used, whereas the panchromatic and thermal bands were excluded. The test area corresponds to a 100-km-length segment of the Rio Grande river channel that runs from Candelaria to Presidio, TX, by a cross section of 5-km width around the main channel of the river. The objective of this test was to assess the improvement of the FS over the unconstrained approach for retrieving subpixel fractions of three land cover types and, at the same time, to assess the relative efficiency with respect to the QP algorithm. A total of $k = 228,800$ Landsat pixels were processed by

TABLE II
PERFORMANCE COMPARISON OF THE LINEAR UNMIXING METHODS THAT ARE TESTED ON LANDSAT DATA

Unmixing	Average RMSE (Std. Dev.)		CPU Time [s]
Method	Reflectance [%]	Fraction [%]	
ULS	0.19 (0.18)	31.48 (23.01)	0.03
QP	1.35 (1.76)	25.07 (14.81)	956.78 (233.88*)
FS	1.35 (1.76)	25.07 (14.81)	1.83

*elapsed time for the main loop of QP code

each method. The land cover types consisted of the following classes: 1) invasive (*Tamarix* species in various phenological stages); 2) native (predominantly *Salix* species and *Prosopis* species); and 3) other (nonwoody, herbaceous vegetation, and bare ground).

An inset of 10 km by 3 km that is located near Candelaria was used for the endmember selection and accuracy assessment. At this site, reference fractions were produced from a high-spatial-resolution (1 m) hyperspectral image (of 61 bands, 10-nm bandwidth, and 400–1000-nm spectral rage) that was acquired with the Airborne Imaging Spectroradiometer for Applications (AISA) sensor on December 21, 2005. Further details on the production of the reference fractions are provided in [6]. The endmembers were produced as the average spectra of up to ten pixels with fractional cover that is greater than 90% in the reference set. Although the endmembers that were used were mixed pixels themselves, they were considered the most spectrally pure representation for the three aforementioned aggregated classes. Random samples ($k = 300$) from the inset were used for accuracy assessment, which was carried out in terms of the root-mean-square error (RMSE) measure.

The three methods (ULS, QP, and FS) were compared in terms of the following: 1) predicted mixed pixel errors; 2) predicted fraction errors; and 3) CPU time. Table II summarizes the results for this test. The results showed that the fully constrained solutions (QP and FS) more accurately retrieve subpixel fractions than the unconstrained approach (RMSE of fraction), despite the higher goodness of the fit (lower RMSE of reflectance) of the latter. Both the QP and the FS yielded the same error measures, but the FS was 130 times faster than the QP (considering the main loop only) and only 60 times slower than the ULS.

C. Unmixing of Hyperion Measurements

The last test was based on a hyperspectral EO-1 Hyperion image (30 m) that was acquired on December 26, 2006 (path 31, row 39). The objective of this test was to compare the relative efficiency of the FS for unmixing hyperspectral data with respect to a partially constrained approach that is available in ENVI. Although the image location concurred with the Landsat data that are previously used, the Hyperion imagery did not overlap with the AISA imagery, and hence, no reference fractions were available for this test.

The image that was acquired was in a processing level (L1Gst) that provides radiometric calibration and orthorectification using a digital elevation model. The Hyperion image contained 198 bands covering the Visible and Near InfraRed (VNIR) and Short-Wavelength InfraRed (SWIR) from 427 to 2394 nm. (Bands 1–7, 58–76, and 225–242 had been eliminated due to atmospheric absorptions.) A subset of 512 lines by 256 samples was selected from the entire scene ($k = 131,072$ pixels). After subsetting, the image digital numbers were converted to at-sensor reflectance values based on [21], and the image noise was reduced by retaining the first ten bands from a minimum noise fraction (MNF) transform [22]. The MNF bands that are beyond band 10 were eliminated because they showed significant stripping artifacts from sensor alignment [21]. After band elimination, the inverse MNF transform yielded a noise-reduced hyperspectral image, and no further atmospheric correction was carried out. Then, the endmembers were automatically generated using a sequentially maximum angle convex cone (SMACC) algorithm that is available in ENVI [23]. The SMACC algorithm identified five endmembers, which were visually interpreted and labeled as Bright Soil, Dry Grass, Water, Green Vegetation, and Dry Woody Vegetation.

The unmixing results from both methods are compared using scatter plots in Fig. 4. As expected, the partially constrained solution yielded negative fractions and never had an RMSE that is greater than the fully constrained solution [except for the small numerical errors; see Fig. 4(f)]. The CPU time for the unmixing algorithm in ENVI could not be measured with the computer clock. Instead, a hand watch was used to measure the elapsed time between the last click to the time when the progress bar ended. The CPU times were approximately 4 min and 25 s for the ENVI algorithm against 10.2 s for the MATLAB implementation of the FS method. The former time may not be representative of the unmixing algorithm that is implemented in ENVI because it may have included unnecessary management processes (e.g., access to disk and progress bar updating), which were not included in the MATLAB implementation. However, the exercise illustrated the viability of the proposed method for hyperspectral image analysis.

VI. SUMMARY

In this paper, the solution of the FCLSU problem has been explicitly derived using geometrical relations among the endmembers and the mixed pixels. The solution was conveniently expressed in terms of the FSs. According to this approach, the fractions of the mixture model are treated as membership functions of the FSs in $\mathbb{R}^n$. As it turned out, such FSs can be built as combinations of half-spaces that are defined by the hyperplanes. The constituent fuzzy half-spaces, as well as the Boolean formula that specifies the way of combining them, depend only on the set of endmembers. This characteristic makes the FS solution more appropriate for large-scale mapping, i.e., when a large number of mixed pixels are to be processed.

The tests from the simulated and the real data sets showed that the proposed FS algorithm was very efficient and viable for unmixing a large number of pixels with few endmembers. An SFS method has been also tested. This method was even more efficient than the FS for larger numbers of endmembers. Unfortunately, this method was also more prone to numerical errors for such large numbers of endmembers. These results

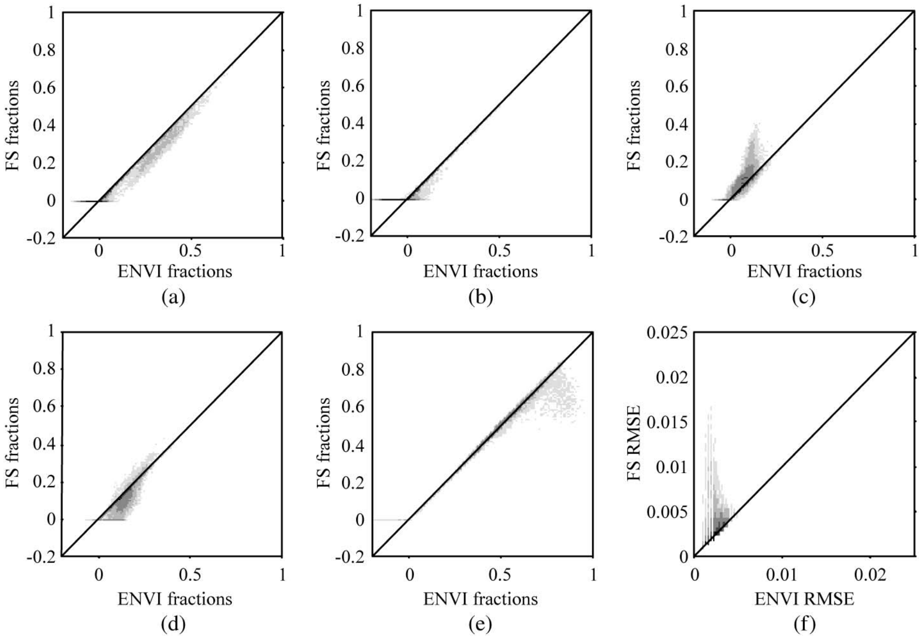


Fig. 4. Comparison of the FS unmixing with respect to a partially constrained method available in the ENVI software. The scatter plots correspond to (a) Bright Soil fractions, (b) Dry Grass fractions, (c) Water fractions, (d) Green Vegetation fractions, (e) Dry Woody Vegetation fractions, and (f) RMSE.

suggested that the best strategy is to use the FS for $m < 7$. For larger values of endmembers, the numerical solution, such as that of the QP optimization, should be used instead.

Although the FSs and the spectral mixture models had been previously independently used for subpixel analysis [24], [25], a theoretical link between the two approaches had not been drawn. Unlike in [26], the explicit solution of the FCLSU problem is a nonlinear transformation of the mixed pixels. In fact, the form of the solution shares similarity with a two-hidden layer neural network, which also suggests the possibility for estimating the endmembers from samples of fractions, as well as a way for generalizing LSU to non-LSU, e.g., by relaxing the linear saturation function [27]. Furthermore, the consideration of other than the Euclidean norm as a measure of spectral similarity, as proposed in [14], could be enabled through the application of the FS method on a nonlinear transformation of the data (e.g., normalization). Before these links can be established, the FS solution for $m > n + 1$ needs to be further investigated.

Acknowledgment

The authors would like to thank the anonymous reviewers for their input. The partial quadratic programming and the sequential fuzzy set implementations were suggested by one of the anonymous reviewers.

References

[1] J. B. Adams, M. O. Smith, and A. R. Gillespie, “Imaging spectroscopy: Interpretation based on spectral mixture analysis,” in *Remote Geochemical Analysis: Elemental and Mineralogical Composition*. Cambridge, MA: Cambridge Univ. Press, 1993, pp. 145–166.

[2] A. R. Gillespie, “Spectral mixture analysis of multispectral thermal infrared images,” *Remote Sens. Environ.*, vol. 42, no. 2, pp. 137–145, Nov. 1992.

[3] E. J. Milton, “Image endmembers and the scene model,” *Can. J. Remote Sens.*, vol. 25, no. 2, pp. 112–120, 1999.

[4] D. C. Heinz and C.-I. Chang, “Fully constrained least square linear spectral unmixing analysis method for material quantification in hyperspectral imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 3, pp. 529–545, Mar. 2001.

[5] J. Settle and N. Drake, “Linear mixing and the estimation of ground cover proportions,” *Int. J. Remote Sens.*, vol. 14, no. 6, pp. 1159–1177, Apr. 1993.

[6] J. L. Silván-Cárdenas and L. Wang, “Retrieval of subpixel Tamarix canopy cover from Landsat data along the forgotten river using linear and nonlinear spectral mixture models,” *Remote Sens. Environ.*, vol. 114, no. 8, pp. 1777–1790, Aug. 2010.

[7] B. Somers, S. Delalieux, W. Verstraeten, J. VerBesselt, S. Lhermitte, and P. Coppin, “Magnitude- and shape-related feature integration in hyperspectral mixture analysis to monitor weeds in citrus orchards,” *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 11, pp. 3630–3642, Nov. 2009.

[8] Y. Gu, Y. Zhang, and J. Zhang, “Integration of spatial–spectral information for resolution enhancement in hyperspectral images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 5, pp. 1347–1358, May 2008.

[9] Y. Ge, S. Li, and V. Lakhan, “Development and testing of a subpixel mapping algorithm,” *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 7, pp. 2155–2164, Jul. 2009.

[10] Y. H. Hu, H. B. Lee, and F. L. Scarpace, “Optimal linear spectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 1, pp. 639–644, Jan. 1999.

[11] P. Gill, W. Murray, and M. Wright, *Practical Optimization*. London, U.K.: Academic, 1981.

[12] P. Gill, W. Murray, and M. Wright, *Numerical Linear Algebra and Optimization*. Redwood City, CA: Addison-Wesley, 1991.

[13] C.-I. Chang and D. C. Heinz, “Constrained subpixel target detection of remotely sensed imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 3, pp. 1114–1159, May 2000.

[14] J. Chen, X. Jia, W. Yang, and B. Matsushita, “Generalization of subpixel analysis for hyperspectral data with flexibility in spectral similarity measures,” *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 7, pp. 2165–2171, Jul. 2009.

[15] L. Zadeh, “Fuzzy sets,” *Inf. Control*, vol. 8, no. 3, pp. 338–353, 1965.

[16] N. Ravishanker and D. Dey, *A First Course in Linear Model Theory*. Boca Raton, FL: Chapman & Hall, 2002.

[17] A. Grace, Optimization toolbox: For use with MATLAB: User’s guide1995.

[18] J. C. Harsanyi and C.-I. Chang, “Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection approach,” *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 4, pp. 779–785, Jul. 1994.

[19] J. Boardman, “Inversion of imaging spectrometry data using singular value decomposition,” in *Proc. IGARSS 12th Can. Symp. Remote*

Sens.—Quantitative Remote Sensing: An Economic Tool for the Nineties, 1989, pp. 2069–2072.

[20] A. Baldridge, S. Hook, C. Grove, and G. Rivera, “The ASTER spectral library version 2.0,” *Remote Sens. Environ.*, vol. 113, no. 4, pp. 711–715, Apr. 2009.

[21] M. Griffin, S. Hsu, H. Burke, S. Orloff, and C. Upham, “Examples of EO-1 Hyperion data analysis,” *Lincoln Lab. J.*, vol. 15, no. 2, p. 271, 2005.

[22] A. Green, M. Berman, P. Switzer, and M. Craig, “A transformation for ordering multispectral data in terms of image quality with implications for noise removal,” *IEEE Trans. Geosci. Remote Sens.*, vol. 26, no. 1, pp. 65–74, Jan. 1988.

[23] J. Gruninger, A. Ratkowski, and M. Hoke, “The sequential maximum angle convex cone (SMACC) endmember model,” in *Proc. SPIE, Algorithms Multispectral Hyperspectral Ultraspectral Imagery*, 2004, vol. 5425, pp. 1–14.

[24] G. Foody and D. Cox, “Sub-pixel land cover composition estimation using a linear mixture model and fuzzy membership functions,” *Int. J. Remote Sens.*, vol. 15, no. 3, pp. 619–631, Feb. 1994.

[25] G. Foody, R. Lucas, P. Curran, and M. Honzak, “Non-linear mixture modelling without end-members using an artificial neural network,” *Int. J. Remote Sens.*, vol. 18, no. 4, pp. 937–953, Mar. 1997.

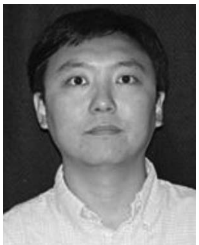
[26] S. Jia and Y. Qian, “Spectral and spatial complexity-based hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 45, pt. 1, no. 12, pp. 3867–3879, Dec. 2007.

[27] J. Silván-Cárdenas, L. Wang, and F. Zhan, “Representing geographical objects with scale-induced indeterminate boundaries: A neural network-based data model,” *Int. J. Geographical Inf. Sci.*, vol. 23, no. 3, pp. 295–318, Mar. 2009.

José Luis Silván-Cárdenas received the B.S. degree in computer engineering and the M.S. degree in electrical engineering from the National Autonomous University of Mexico (UNAM), Mexico City, Mexico, in 1998 and 2002, respectively, and the Ph.D. degree in geographic information sciences from Texas State University, San Marcos, in 2009.

From 2008 to 2010, he was a Postdoctoral Scholar with the State University of New York, Buffalo, and in July 2010, he joined the Geography and Geomatic Research Center (CentroGeo), Mexico City, as an Associate Research Scientist. He has coauthored a dozen of referred journal articles on various topics, including spectral mixture analysis, light detection and ranging (LiDAR) data filtering, subpixel accuracy assessment, land use/cover classification, plant species detection, geographic data representation, applied geostatistics, and population estimation. His research interests include remote sensing digital image processing, feature extraction from LiDAR data, and high spatial and hyperspectral resolution remote sensing techniques.

Dr. Silván-Cárdenas was the recipient of several renowned awards, including the Alfoso Caso Medal in 2002 that was granted by UNAM to the most distinguished graduate students and the Student Honor Paper Competition Award from the Remote Sensing Specialty Group of the Association of American Geographers in 2008.



Le Wang received the B.E. degree from Wuhan Technical University of Surveying and Mapping, Wuhan, China, in 1996, the M.S. degree from Peking University, Beijing, China, in 1999, and the Ph.D. degree in environmental science, policy, and management from the University of California, Berkeley, in 2003.

He is currently an Associate Professor of geography with the Department of Geography, College of Arts and Sciences, State University of New York, Buffalo. He has authored more than 30 reference journal articles and two book chapters. His research interests include the development of advanced remote sensing techniques for estimating small-area urban populations, mapping and monitoring coastal mangrove forests, and mapping and modeling the spread of invasive species.

Prof. Wang was the recipient of the 2008 Early Career Award from the Remote Sensing Specialty Group of the Association of American Geographers.