A Stepwise Analytical Projected Gradient Descent Search for Hyperspectral Unmixing and Its Code Vectorization

Fadi Kizel, Maxim Shoshany, Nathan S. Netanyahu, Gilad Even-Tzur, and Jón Atli Benediktsson

***Abstract*— We present in this paper a new methodology for spectral unmixing, where a vector of fractions, corresponding to a set of endmembers (EMs), is estimated for each pixel in the image. The process first provides an initial estimate of the fraction vector, followed by an iterative procedure that converges to an optimal solution. Specifically, *projected gradient descent* (PGD) optimization is applied to (a variant of) the *spectral angle mapper* (SAM) objective function, so as to reduce significantly the estimation error due to amplitude (i.e., magnitude) variations in EM spectra, caused by the illumination change effect. To improve the computational efficiency of our method over a commonly used gradient descent technique, we have derived analytically the objective function's gradient and the optimal step size (used in each iteration). To gain further improvement, we have implemented our unmixing module via code vectorization, where the entire process is "folded" into a single loop, and the fractions for all of the pixels are solved for simultaneously. We call this new parallel scheme *vectorized code projected gradient descent unmixing* (VPGDU). VPGDU has the advantage of solving (simultaneously) an independent optimization problem per image pixel, exactly as other pixel-wise algorithms, but significantly faster. Its performance was compared to the commonly used *fully constrained least squares unmixing* (FCLSU), the *generalized bilinear model* (GBM) method for hyperspectral unmixng, and the fast state-of-the-art methods, *sparse unmixing by variable splitting and augmented Lagrangian* (SUnSAL) and *collaborative SUnSAL* (CLSUnSAL) based on the *alternating direction method of multipliers* (ADMM). Considering all of the prospective EMs of a scene at each pixel (i.e., without a priori knowledge which/how many EMs are actually present in a given pixel), we demonstrate that the accuracy due to VPGDU is considerably higher than that obtained by FCLSU, GBM, SUnSAL, and CLSUnSAL under varying illumination, and is otherwise comparable with respect to these methods. However, while our method is significantly faster than FCLSU and GBM, it is slower than SUnSAL and CLSUnSAL by roughly an order of magnitude.**

***Index Terms*— Hyperspectral imaging, spectral unmixing, Gradient methods, Optimization**

Manuscript received _____; revised _____; accepted _____. Date of publication _____; date of current version _____. This work was supported in part by the Israel Science Ministry, Scientific Infrastructure Research Grant Scheme 3-8144 (2011-2014), in part by the Helen and Norman Asher Space Research Grant Scheme (211-2012), in part by the Technion PhD Scholarship (2009-2013), in part by the new England fund, Technion, and in part by the Environmental Mapping and Monitoring of Iceland by Remote Sensing (EMMIRS) project.

Fadi Kizel is affiliated with the Department of Mapping and Geo-Information Engineering, Technion–Israel Institute of Technology, Haifa 32000, Israel, where he carried out this research as part of his Ph.D. thesis, and is currently with the Faculty of Electrical and Computer Engineering, University of Iceland, Hjardarhagi 2-6, 107 Reykjavik, Iceland (e-mail: fqzl@campus.technion.ac.il; fadikizel@hi.is).

Maxim Shoshany and Gilad Even-Tzur are with the Department of Mapping and Geo-Information Engineering, Technion–Israel Institute of Technology, Haifa 32000, Israel (e-mail: maximsh@technion.ac.il; eventzur@tecnion.ac.il).

Nathan S. Netanyahu is with the Department of Computer Science, Bar-Ilan University, Ramat Gan 52900, Israel, and is also affiliated with the Center for Automation Research, University of Maryland, College Park, MD 20742 USA (e-mail: nathan@cs.biu.ac.il; nathan@cfar.umd.edu).

Jón Atli Benediktsson is with the Faculty of Electrical and Computer Engineering, University of Iceland, Hjardarhagi 2-6, 107 Reykjavik, Iceland

I. Introduction

GIVEN a (hyper)spectral image, the linear mixture model assumes that the collected spectra in a given pixel is formed as a linear combination of a set of pure spectral signatures, known as endmembers (EMs). Only a few pixels in an image are essentially "pure" [1], while the rest – especially in remotely sensed images – contain more than one material. Thus, reliable analysis of acquired spectral data requires the process of *spectral unmixing*, where a vector of fractions (abundances), corresponding to the set of EMs, is estimated for each pixel in the scene. (See, e.g., [2], [3] and [4] for detailed surveys.) The recent growing availability of airborne and satellite hyperspectral (HS) remote sensing platforms poses new challenges vis-à-vis the utilization of HS imagery in a wide range of applications. Such applications may include the important processing of urban, agricultural, and natural image regions, which requires the detection of a large number of biotic, a-biotic, and man-made materials. To distinguish between a large number of EMs, one needs to address the following main issues that are inherently associated with HS imagery: (1) Spectral similarities between different materials (with differences only in a few spectral features), (2) variations in illumination angles (topographic effects) resulting in different spectral reflectance distributions for the same surface cover materials, and (3) high processing time as the number of prospective EMs increases.

In tackling the above issues, it is essential to first determine the most relevant set of EMs and then employ an appropriate

unmixing strategy. Methods for EM finding include, e.g., the manual EM selection tool (MEST) [5], as well as various automated algorithms, based on multidimensional geometric and statistical principles [6]. Early automated methods, e.g., the N-FINDR [7] and the improved version presented in [8], generally seek pure pixels that represent the EMs, while more recent methods [9], [10] do not assume the presence of pure pixels and try to estimate, instead, the EM spectra as the simplex vertices of the data cloud based on the principle of the *minimum volume enclosing simplex* (MVES). In addition, recently developed methods [11], [12] try to overcome the problem of non-present pure pixels by using *sparse regression* techniques [13]; a large number of library spectra is used to model the mixed pixels and the most suitable subset of EMs is found for each pixel during the unmixing process. See [14], for a detailed survey of EM extraction methods, and [15]–[17], for more recent implementation approaches.

As noted, once an adequate EM set is determined, an appropriate unmixing strategy is employed to find an optimal abundance vector in fraction space. Numerous unmixing methods have been pursued over the years to meet this objective. For example, least squares-based approaches ([18], [19]) have been used in an iterative manner to provide fully constrained solutions. This was further refined by stepwise search strategies, such as *quadratic programming* [20]–[22], *gradient descent optimization* [23]–[27], and *sequential quadratic programming* (SQP) [28]. One of the disadvantages, however, of a typical search algorithm is its low computational efficiency. This deficiency can be tackled successfully due to [11], [12]. In an attempt to further alleviate this issue, it is of interest to derive semi-analytical solutions for gradient descent methods. A comprehensive overview of unmixing methods with an emphasis on fraction estimation can be found in [29], [30]. More recent methods consider solution sparsity [31]–[33], spatial information [34], [35], or both elements [36], [37] to further enhance the fraction accuracy. In addition to the wide use of the LMM in the majority of existing methods, nonlinear models have been recently introduced in various works. (See, e.g., [38], [39] for detailed surveys.) In particular, the *generalized bilinear model* (GBM) presented in [26] is a generalization of the bilinear model and the ordinary LMM.

The choice of an objective function is naturally a crucial component of the unmixing process. The *Euclidean minimum distance* (EMD), known for its convenient integration with a constrained least squares framework, has been commonly used for unmixing. Unfortunately, the performance of EMD unmixing is highly affected by the illumination change effect (which causes magnitude variations in the reflected spectra due to shadow and different topography [40]). Exploiting, on the other hand, the special geometric properties of the *spectral angle mapper* (SAM) measure can reduce significantly the resulting bias and improve accordingly the unmixing results; see, e.g., [28], [41]. We employ in this paper *projected gradient descent* (PGD), which projects the estimate (in each iteration) onto the feasible subspace defined by the required constraints, for solving efficiently the constrained optimization problem in question.

The use of PGD for constrained optimization problems was originally proposed in [42] (some works also refer to [43]). In practice, PGD is a subcase of the proximal gradient decent methods [44], which are more suitable for non-smooth objective functions. It is also considered a subcase of the *forward-backward splitting algorithm* [45], where a forward step is applied by a single progress of the ordinary gradient descent, and a backward step is applied by projecting the result onto the feasible region. (See [46] for a survey and detailed discussion on these methods.) In general, the performance of gradient-based methods is highly influenced by the choice of the step size [46], [47]. Given a function ϕ to be minimized, with gradient $\mathbf{p}$ at point $\mathbf{x}$, the ideal (current) step size, γ, is the global minimizer (in case of a minimization problem) of the univariate function defined by [48]

$$\varphi(\gamma) = \phi(\mathbf{x} + \gamma\mathbf{p}).$$

Identifying γ according to the above could be very expensive, though, in most cases. An alternative, efficient step size computation may be achieved by a backtracking line search, subject to the *Armijo rule*. The latter ensures an effective (but not necessarily optimal) reduction at each iteration [46], [47], and was found to be satisfactory in terms of both processing time and convergence. Still, it would be highly desirable to derive an optimal step size for achieving maximal progress at each iteration of the process in an analytical (non-expensive) manner.

We employ PGD for spectral unmixing, while providing analytical, closed-form expressions for the gradient of the SAM-like objective function selected and the step size at each iteration, to improve the unmixing process in terms of: (1) Robustness to the varying illumination effect, (2) scalability to a large number of EMs, and (3) computational speedup. Furthermore, the analytic stepwise PGD framework results also in a simple parallelization of the entire process by *code vectorization*. (Using vectorized code reduces significantly the run-time of the entire process by "loop-unrolling", where in the case of image unmixing, all the inner loops of the different pixels are run simultaneously using array (instead of scalar) operations; it also has the advantage of solving a separate optimization problem for each individual pixel within a parallel run.)

In summary, the study presented here aims at developing an accurate and computationally efficient approach for unmixing, especially under varying illumination, by applying an analytical projected gradient descent formulation to a variant of the SAM objective function. A detailed formulation of the analytical expressions for the objective function's gradient and step size in each iteration is provided. We supplement these derived expressions with a fast algorithm for projection onto the constraint simplex [49] to yield a stepwise analytical framework for fully constrained unmixing. The newly derived method is highly robust to illumination change, can handle a

relatively large number of EMs, and is adaptable to any linear fraction constraint. To overcome the high processing time, associated typically with gradient descent, we take advantage of the closed-form analytical expressions derived and the simplicity of the framework's components. As noted, we implemented our proposed scheme via code vectorization, which results in significant speedup on raster images. We call this unmixing scheme *vectorized code projected gradient descent unmixing* (VPGDU). VPGDU performs essentially unmixing for a given image by solving simultaneously a whole set of independent optimization problems, where each problem is associated with an image pixel. Owing to this advantage, VPGDU can be invoked simultaneously with different parameters at each pixel, e.g., likelihood of purity, a good initial fraction estimate, upper and lower bounds of the various estimated fractions, different termination criteria (e.g., number of iterations and stopping threshold), etc.

The developed methodology was tested extensively on real data, such as the well-known Cuprite reflectance image and an AISA image over a mixed natural-urban region, as well as synthetic data, e.g., EM signatures extracted automatically/manually (from these real images) or selected from a spectral library. In particular, it was compared against the commonly used FCLSU method [19], the GBM method [26], and the fast state-of-the-art methods, SUnSAL [11] and CLSUnSAL [11]-[12]. (The latter two methods provide an abundance estimate based on the *alternating direction method of multipliers* (ADMM) [50], by solving the so-called *constrained sparse regression* (CSR) problem.)

The rest of the paper is organized as follows. Section II presents the linear mixture model used in this work. Section III presents the main concepts of our framework. Section IV gives a detailed analytical derivation of the proposed projected gradient descent unmixing methodology, including a detailed discussion of its parallel implementation via code vectorization (VPGDU). Section V presents detailed experimental results, including comparative performance evaluation of VPGDU versus FCLSU, GBM, SUnSAL, and CLSunSAL. Finally, Section VI makes concluding remarks.

II. The Linear Mixture Model

Assuming a linear mixture model (LMM), each pixel signature, $\mathbf{m} = [m_1, ..., m_\lambda]^T$, in a hyperspectral image with λ bands, can be expressed as a linear combination of L EM spectra as follows:

$$\mathbf{m} = \mathbf{E}\mathbf{f} + \mathbf{n}, \tag{1}$$

where $\mathbf{E}$ is a $(\lambda \times L)$ matrix whose columns are the EM spectral signatures, $\mathbf{f}$ is an $(L \times 1)$ vector containing the true fractions of the EMs, and n is a $(\lambda \times 1)$ vector, assumed to be a zero-mean Gaussian representing the system noise. During the unmixing process an estimated fraction vector, $\hat{\mathbf{f}}$, is calculated for each pixel in the image. The LMM represents the relation between the EMs, their fractions and the mixture, and it constitutes the basis for the mathematical terminology and formulations to be used during the unmixing process.

III. Framework Concepts

A. Searching in Endmember Fraction Space for Spectral Unmixing

Let $\mathbf{S} \in \mathbb{R}^L$ be a scalar field spanned by the orthogonal set $\{(f_1, 0, ..., 0), (0, f_2, ..., 0), ..., (0, ..., f_L)\}$, where each coordinate axis in $\mathbf{S}$ is represented by a single EM fraction. Each point in $\mathbf{S}$ represents a fraction combination which reconstructs a different mixture due to (1). A solution of the unmixing problem can be achieved by seeking the point (i.e., fraction combination) in $\mathbf{S}$ that optimally reconstructs the pixel's spectral signature. Ranking points in $\mathbf{S}$ as prospective solutions for the spectral unmixing requires a definition of an objective function that measures the spectral distance between the reconstructed mixture and the pixel signature. Once an objective function is defined, an optimal solution can be achieved by finding the global maximum/minimum (depending on the kind of objective function). Searching for an optimal, fully constrained solution should be carried out by a examining a solution in the feasible region bounded by the non-negativity and sum-to-one constraints. Search methods for spectral unmixing should combine an objective function and an optimization method; a nonlinear objective function requires an initial estimate before applying an iterative search process. Although iterative search methods are more flexible and adaptive to different conditions and constraints, they tend to be very slow, especially when some required parameters are determined empirically. Thus the desired goal is to derive an analytical stepwise search method that would result in enhanced unmixing performance. The following three main components will be employed in our methodology:

1. *An initial estimate:*

We employ the preprocessing model presented in [51] to generate an initial estimate of the fraction vector based on the relationship between the fractions and the spectral angle mapper (SAM) values between the EMs and the mixture.

2. *Objective function:*

The objective function picked is based on the SAM measure for its crucial advantage in reducing significantly the illumination effect. The parameters of the search process presented in this paper will be analytically derived with respect to the SAM objective function.

3. *Stepwise analytical projected gradient descent with vectorized code:*

A projected gradient descent procedure is derived in fraction space seeking for a constrained optimal solution. Also, code vectorization was implemented for speeding up the entire process.

The above three components are integrated within a stepwise analytical framework, where the initial estimate is arrived at empirically. Following the initial estimate, an iterative search procedure is employed. In each step the result is assessed with respect to the objective function and the fraction change. The search is repeated until the objective function is satisfied or the maximal fraction change becomes smaller than a pre-defined threshold. Fig. 1 conceptually demonstrates the algorithm's framework. In the next section we describe in detail each of these methodological components.

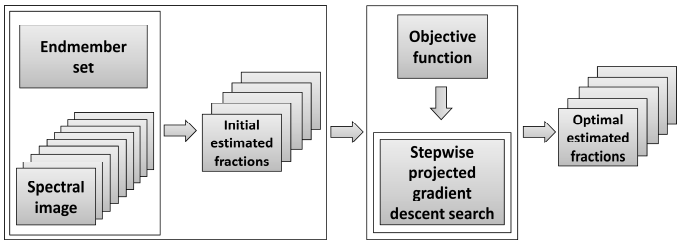

Fig. 1. Conceptual framework of unmixing methodology.

IV. Detailed Derivation Of Stepwise Analytical Projected Gradient Descent Spectral Unmixing

A. *Initial Estimate*

We provide an overview of this stage according to the presentation in [51]. Given two spectral signatures $\mathbf{s_1}$ and $\mathbf{s_2}$, their *SAM* similarity measure is defined as

$$SAM\left(\mathbf{s_1},\mathbf{s_2}\right)=\cos^{-1}\left(\frac{\mathbf{s_1}^T\mathbf{s_2}}{\|\mathbf{s_1}\|\cdot\|\mathbf{s_2}\|}\right), \tag{2}$$

where $\|\cdot\|$ denotes the ℓ_2-norm. Logically, the larger the EM's fraction f_i is, the smaller the *SAM* measure is between its signature and the mixture signature. Actually, this relationship is a function of all of the EMs and their fractions in a given mixture and it can be mapped in order to create a basis for generating an initial estimate of the fraction vector. Given a pixel signature $\mathbf{m}$ that is a mixture over the set of EMs, the *normalized SAM* of the i^{th} EM, denoted by NS_i, is defined as:

$$NS_i=\frac{SAM(\mathbf{E}_i,\mathbf{m})}{\sum_{j=1}^{L}SAM(\mathbf{E}_j,\mathbf{m})}, \tag{3}$$

where $\mathbf{E}_i$ denotes the signature of the i^{th} EM. The relationship between a fraction and its *NS* value is intrinsic and can be described by fitting a linear function whose coefficients can be estimated according to [51]. Specifically, we carry out the following steps:

Step 1: Simulate a set of known fractions in the range $0\le f_i\le 1$ and create for each fraction a controlled mixture $\mathbf{m}$ according to the expression

$$m^{(d)}=f_iE_i^{(d)}+\sum_{j\neq i}f_jE_j^{(d)}, \tag{4}$$

where $m^{(d)}$ and $E_j^{(d)}$ denote, respectively, the reflectance of $\mathbf{m}$ and $\mathbf{E}_i$ in the d^{th} spectral band, and the f_j's ($j\neq i$) are picked at random subject to $\sum_{j\neq i}f_j=1-f_i$.

Step 2: For each fraction f_i and its corresponding controlled mixture**,** compute a corresponding NS_i according to (3)**.**

Step 3: Fit a linear function to estimate the relationship between the true fractions and their corresponding NS_i values.

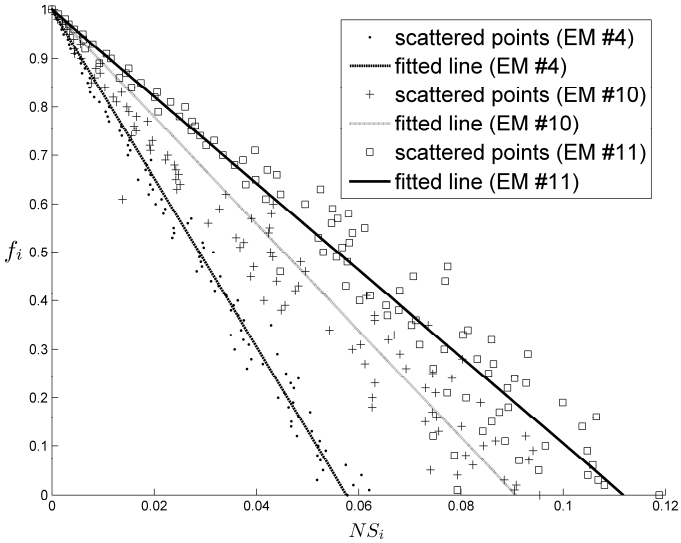

Fig. 2. Scatter plot of f_i vs. NS_i for several EMs, with fitted linear functions.

Fig. 2 presents the scatter of the fractions versus their corresponding *NS* values and the estimated linear functions for three different EMs taken from a set of EMs to be presented (see Fig. 14). Each assessed coefficient is essentially a function of the EM set, regardless of the mixture to be solved; once preprocessing is applied, its results can be used for any mixture containing these EMs. Let $\boldsymbol{\alpha}=[\alpha_1,\alpha_2,\ldots,\alpha_L]$ and $\boldsymbol{\beta}=[\beta_1,\beta_2,\ldots,\beta_L]$ denote slope and intercept vectors, respectively, where α_i and β_i are the estimated coefficients of the fitted linear function for the i^{th} EM. Also, let $\mathbf{NS}=[NS_1,NS_2,\ldots,NS_L]$ denote the vector of normalized *SAMs* computed according to (3) for a given mixture $\mathbf{m}$. Using our method, an initial estimate of the fraction vector is given by

$$\hat{\mathbf{f}}^0=\boldsymbol{\alpha}^T\mathbf{NS}+\boldsymbol{\beta}. \tag{5}$$

As can be seen from Fig. 2, the initial estimate could provide negative values for EMs with actual small abundances. Since this step provides merely an initial estimate, negative values are set to zero. Note that in case the objective function is convex, devising a specific initial estimate might not be essential, although it would be beneficial for faster convergence to the optimal solution.

B. Projected gradient descent for fully constrained spectral unmixing

Gradient descent is a standard, commonly used method for nonlinear optimization. Using it for the unmixing problem, we start with an initial estimate of the fraction vector $\hat{\mathbf{f}}^0$. Then, a stepwise computation towards the optimal solution is applied according to:

$$\hat{\mathbf{f}}^{k+1} = \hat{\mathbf{f}}^k + \gamma_k \nabla\phi\left(\hat{\mathbf{f}}^k\right), \tag{6}$$

where ϕ is the objective function, k is the iteration number, γ_k is the optimal step size in the gradient direction (giving a maximal change in ϕ), and $\nabla\phi\left(\hat{\mathbf{f}}^k\right)$ is the gradient of ϕ at $\hat{\mathbf{f}}^k$, or the derivative of the objective function with respect to the fraction vector, i.e., $\nabla\phi = \frac{\partial\phi}{\partial\hat{\mathbf{f}}}$. For a differentiable objective function ϕ, $\nabla\phi$ can be expressed analytically or can be calculated numerically. The gradient points in the direction which maximizes ϕ, but the change quantity in ϕ itself still depends on the step size γ. Finding the optimal step size, i.e., the step size that yields the highest change in ϕ (for the current iteration) is done by solving the optimization problem:

$$\gamma_k = \arg\max\{\varphi\}, \tag{7}$$

where

$$\varphi = \phi\left(\hat{\mathbf{f}}^k + \gamma_k \nabla\phi\left(\hat{\mathbf{f}}^k\right)\right). \tag{8}$$

It would be desirable to find an explicit analytical solution by solving $\frac{\partial\varphi}{\partial\gamma_k} = 0$. Otherwise, a numerical solution may be applied.

Requiring valid, feasible unmixing results restricts the optimization process by the well-known *abundance non-negativity constraint* (ANC) and the *abundance sum-to-one constraint* (ASC), i.e., $\hat{\mathbf{f}} \geq 0$ and $\mathbf{1}^{\mathbf{T}}\hat{\mathbf{f}} = 1$, respectively. Given an unmixing problem with *L* EMs, the feasible solution region defined by the ANC and ASC constraints is the positive simplex of the ℓ_1 ball in $\mathbb{R}^L$, e.g., the feasible region for the 2-EM problem is the line segment joining the points (1,0) and (0,1) in $\mathbb{R}^2$, and that of the 3-EM problem is the planar (triangular) segment defined by the points (1,0,0), (0,1,0) and (0,0,1) in $\mathbb{R}^3$. The iterative procedure given by (6-8) follows the steepest descent direction in each step. However, $\hat{\mathbf{f}}^{k+1}$ may not necessarily meet the ANC and ASC constraints. Specifically, to keep the solution inside the feasible region in each iteration, we apply the following projected gradient descent process:

$$\hat{\mathbf{f}}^{k+1} = \mathrm{P}_\Omega\left(\hat{\mathbf{f}}^k + \gamma_k \nabla\phi\left(\hat{\mathbf{f}}^k\right)\right), \tag{9}$$

where $\mathbf{x} = \mathrm{P}_\Omega(\mathbf{y})$ denotes the projection of $\mathbf{y} \in \mathbb{R}^L$ onto the convex set Ω and is defined by

$$\mathbf{x} = \arg\min_{\mathbf{x}\in\Omega} \left\|\mathbf{x} - \mathbf{y}\right\| \tag{10}$$

For the problem in question we will denote $\Omega = \Omega^L$ as the canonical simplex defined by

$$\Omega^L = \left\{\mathbf{x} = \left(x_1, x_1 \cdots x_L\right)^{\mathrm{T}} \in \mathbb{R} : 0 \leq x_i \leq 1 \text{and} \sum_{i=1}^{L} x_i = 1\right\} \tag{11}$$

In other words, although the calculated gradient $\nabla\phi\left(\hat{\mathbf{f}}^k\right)$ in conjunction with the step size γ_k might yield a new point that is out of the feasible region, the Euclidean projection presented in (9—11) ensures that the next point would belong to the feasible region by finding the closest point to the simplex defined in (11). Fig. 3 presents a scheme of the projected gradient descent process for a general objective function ϕ and a convex set Ω.

To solve the minimization problem in (10), we use the very fast algorithm presented in [49]. (A related method for projection onto a simplex, as part of an FCLSU can be found in [52].) Note that by modifying Ω^L such that $\sum_{i=1}^{L} x_i \leq 1$, the ASC constraint can be easily replaced, if required, by a sum-less-than-one constraint [53].

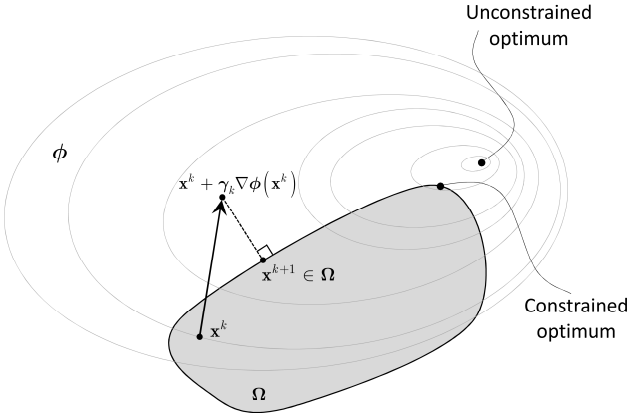

Fig. 3. Illustration of the projected gradient descent process for a general objective function ϕ and convex set Ω.

C. The objective function

Among numerous available spectral unmixing methods, the objective function used mostly is the Euclidean minimum distance (EMD). One drawback of the EMD is its sensitivity to change in radiometry. Given that an EM spectral shape is fairly consistent while its amplitude varies significantly [54], it would be of interest to employ a spectral measure that minimizes the amplitude variation effect. A measure that meets this requirement is the spectral angle mapper (SAM). As indicated before, the beneficial use of the SAM as an objective function for the spectral unmixing was proved and shown in [28], [28]. In view of the previous assumptions, the objective function can be defined as

$$\psi = SAM\left(\mathbf{m}, \mathbf{E}\hat{\mathbf{f}}\right) = \cos^{-1}\left(\frac{\mathbf{m}^T\mathbf{E}\hat{\mathbf{f}}}{\|\mathbf{m}\|\cdot\|\mathbf{E}\hat{\mathbf{f}}\|}\right) \tag{12}$$

subject to

$$\sum_{i=1}^{L}\hat{f}_i \le 1 \quad \text{and} \quad 0 \le \hat{f}_i \le 1.$$

An optimal estimation of $\hat{\mathbf{f}}$ can be achieved by minimizing ψ. Carrying out a gradient descent optimization requires the gradient (at any point) of the objective function. For convenience, we write:

$$\psi = \cos^{-1}\phi \tag{13}$$

The gradient of ψ can be derived by taking

$$\nabla\psi\left(\hat{\mathbf{f}}^k\right) = \frac{\partial\psi}{\partial\hat{\mathbf{f}}} = -\frac{\nabla\phi\left(\hat{\mathbf{f}}^k\right)}{\sqrt{1-\phi^2}}. \tag{14}$$

We notice that the gradient of ψ is undefined at the minimum, where $\phi = 1$, and so a gradient optimization could provide unstable results when applied to ψ. Thus, we take ϕ as an alternative objective function, i.e., we want to maximize

$$\phi = \frac{\mathbf{m}^T\mathbf{E}\hat{\mathbf{f}}}{\|\mathbf{m}\|\cdot\|\mathbf{E}\hat{\mathbf{f}}\|}, \tag{15}$$

subject to

$$\sum_{i=1}^{L}\hat{f}_i \le 1 \quad \text{and} \quad 0 \le \hat{f}_i \le 1$$

In other words, an optimal estimation of the fractions should satisfy

$$\hat{\mathbf{f}} = \arg\max\{\phi\}. \tag{16}$$

To derive the gradient of ϕ, let us express it as $\phi = \frac{\mu}{\vartheta}$, where

$$\mu = \mathbf{m}^T\cdot\mathbf{E}\hat{\mathbf{f}} \quad \text{and} \quad \vartheta = \|\mathbf{m}\|\cdot\|\mathbf{E}\hat{\mathbf{f}}\|.$$

It can be easily shown that the derivatives of μ and ϑ with respect to $\hat{\mathbf{f}}$ are $\frac{\partial\mu}{\partial\hat{\mathbf{f}}} = \mathbf{E}^T\mathbf{m}$ and $\frac{\partial\vartheta}{\partial\hat{\mathbf{f}}} = \frac{\mathbf{E}^T\mathbf{E}\hat{\mathbf{f}}}{\|\mathbf{E}\hat{\mathbf{f}}\|}\cdot\|\mathbf{m}\|$, respectively. Thus, the gradient of ϕ can be derived as follows:

$$\begin{aligned}\nabla\phi\left(\hat{\mathbf{f}}\right) &= \frac{\partial\phi}{\partial\hat{\mathbf{f}}} = \frac{\frac{\partial\mu}{\partial\hat{\mathbf{f}}}\vartheta - \frac{\partial\vartheta}{\partial\hat{\mathbf{f}}}\mu}{\vartheta^2} \\ &= \frac{\mathbf{E}^T\mathbf{m}\cdot\|\mathbf{m}\|\cdot\|\mathbf{E}\hat{\mathbf{f}}\| - \frac{\mathbf{E}^T\mathbf{E}\hat{\mathbf{f}}}{\|\mathbf{E}\hat{\mathbf{f}}\|}\cdot\mathbf{m}^T\mathbf{E}\hat{\mathbf{f}}\cdot\|\mathbf{m}\|}{\left(\|\mathbf{m}\|\cdot\|\mathbf{E}\hat{\mathbf{f}}\|\right)^2}.\end{aligned} \tag{17}$$

Simplifying, somewhat, we obtain

$$\nabla\phi\left(\hat{\mathbf{f}}^k\right) = \frac{\mathbf{E}^T\mathbf{m}\cdot\|\mathbf{E}\hat{\mathbf{f}}\|^2 - \mathbf{E}^T\mathbf{E}\hat{\mathbf{f}}\cdot\mathbf{m}^T\mathbf{E}\hat{\mathbf{f}}}{\|\mathbf{m}\|\cdot\|\mathbf{E}\hat{\mathbf{f}}\|^3}. \tag{18}$$

As mentioned previously, the gradient points in the direction that maximizes the objective function; the amount of change in the objective function still depends on the step size γ. An optimal step size can be achieved by differentiating $\varphi = \phi\left(\hat{\mathbf{f}}^k + \gamma_k\nabla\phi\left(\hat{\mathbf{f}}^k\right)\right)$ with respect to γ_k and requiring that $\frac{\partial\varphi}{\partial\gamma_k} = 0$. Following (12), and omitting the index notations, such that, $\hat{\mathbf{f}} = \hat{\mathbf{f}}^k$, $\gamma = \gamma_k$, and $\nabla = \nabla\phi\left(\hat{\mathbf{f}}^k\right)$, we get

$$\varphi = \phi\left(\hat{\mathbf{f}} + \gamma\nabla\right) = \frac{\mathbf{m}^T\mathbf{E}\left(\hat{\mathbf{f}} + \gamma\nabla\right)}{\|\mathbf{m}\|\cdot\|\mathbf{E}\left(\hat{\mathbf{f}} + \gamma\nabla\right)\|}. \tag{19}$$

Using the differentiation chain rule, we obtain

$$\frac{\partial\varphi}{\partial\gamma} = \frac{\partial\phi\left(\hat{\mathbf{f}} + \gamma\nabla\right)}{\partial\gamma} = \frac{\partial\phi\left(\hat{\mathbf{f}} + \gamma\nabla\right)}{\partial\left(\hat{\mathbf{f}} + \gamma\nabla\right)}\cdot\frac{\partial\left(\hat{\mathbf{f}} + \gamma\nabla\right)}{\partial\gamma}. \tag{20}$$

Following (15), and using again the shorthand index notation, we obtain the derivative of φ with respect to $\left(\hat{\mathbf{f}} + \gamma\nabla\right)$ as

$$\begin{aligned}&\frac{\partial\varphi}{\partial\left(\hat{\mathbf{f}} + \gamma\nabla\right)} = \\ &\frac{\mathbf{E}^T\mathbf{m}\cdot\|\mathbf{E}\left(\hat{\mathbf{f}} + \gamma\nabla\right)\|^2 - \mathbf{E}^T\mathbf{E}\left(\hat{\mathbf{f}} + \gamma\nabla\right)\cdot\mathbf{m}^T\mathbf{E}\left(\hat{\mathbf{f}} + \gamma\nabla\right)}{\|\mathbf{m}\|\cdot\|\mathbf{E}\left(\hat{\mathbf{f}} + \gamma\nabla\right)\|^3}.\end{aligned} \tag{21}$$

Also, the derivative of $\left(\hat{\mathbf{f}}+\gamma\nabla\right)$ with respect to γ is:

$$\frac{\partial\left(\hat{\mathbf{f}}+\gamma\nabla\right)}{\partial\gamma}=\nabla^T \tag{22}$$

Requiring that $\frac{\partial\phi}{\partial\gamma}=0$ yields

$$\nabla^T\mathbf{E}^T\mathbf{m}\cdot\left\|\mathbf{E}\left(\hat{\mathbf{f}}+\gamma\nabla\right)\right\|^2=\nabla^T\mathbf{E}^T\mathbf{E}\left(\hat{\mathbf{f}}+\gamma\nabla\right)\cdot\mathbf{m}^T\mathbf{E}\left(\hat{\mathbf{f}}+\gamma\nabla\right) \tag{23}$$

This can be rewritten as

$$\nabla^T\mathbf{E}^T\mathbf{m}\left(\left(\hat{\mathbf{f}}+\gamma\nabla\right)^T\mathbf{E}^T\mathbf{E}\left(\hat{\mathbf{f}}+\gamma\nabla\right)\right)=\nabla^T\mathbf{E}^T\mathbf{E}\left(\hat{\mathbf{f}}+\gamma\nabla\right)\cdot\mathbf{m}^T\mathbf{E}\left(\hat{\mathbf{f}}+\gamma\nabla\right) \tag{24}$$

Further manipulations give the equations below:

$$\begin{aligned}&\nabla^T\mathbf{E}^T\mathbf{m}\left(\hat{\mathbf{f}}^T\mathbf{E}^T\mathbf{E}\hat{\mathbf{f}}+\gamma\nabla^T\mathbf{E}^T\mathbf{E}\hat{\mathbf{f}}+\gamma\hat{\mathbf{f}}^T\mathbf{E}^T\mathbf{E}\nabla+\gamma^2\nabla^T\mathbf{E}^T\mathbf{E}\nabla\right)\\&=\nabla^T\mathbf{E}^T\mathbf{E}\left(\hat{\mathbf{f}}\mathbf{m}^T\mathbf{E}\hat{\mathbf{f}}+\gamma\nabla\mathbf{m}^T\mathbf{E}\hat{\mathbf{f}}+\gamma\hat{\mathbf{f}}\mathbf{m}^T\mathbf{E}\nabla+\gamma^2\nabla\mathbf{m}^T\mathbf{E}\nabla\right)\end{aligned} \tag{25}$$

or

$$\begin{aligned}&\nabla^T\mathbf{E}^T\mathbf{m}\hat{\mathbf{f}}^T\mathbf{E}^T\mathbf{E}\hat{\mathbf{f}}+\gamma\nabla^T\mathbf{E}^T\mathbf{m}\nabla^T\mathbf{E}^T\mathbf{E}\hat{\mathbf{f}}+\gamma\nabla^T\mathbf{E}^T\mathbf{m}\hat{\mathbf{f}}^T\mathbf{E}^T\mathbf{E}\nabla+\gamma^2\nabla^T\mathbf{E}^T\mathbf{m}\nabla^T\mathbf{E}^T\mathbf{E}\nabla\\&=\nabla^T\mathbf{E}^T\mathbf{E}\hat{\mathbf{f}}\mathbf{m}^T\mathbf{E}\hat{\mathbf{f}}+\gamma\nabla^T\mathbf{E}^T\mathbf{E}\nabla\mathbf{m}^T\mathbf{E}\hat{\mathbf{f}}+\gamma\nabla^T\mathbf{E}^T\mathbf{E}\hat{\mathbf{f}}\mathbf{m}^T\mathbf{E}\nabla+\gamma^2\nabla^T\mathbf{E}^T\mathbf{E}\nabla\mathbf{m}^T\mathbf{E}\nabla\end{aligned} \tag{26}$$

Since all of the terms in (23) are scalars, the following hold true:

$$\begin{aligned}&\nabla^T\mathbf{E}^T\mathbf{m}\hat{\mathbf{f}}^T\mathbf{E}^T\mathbf{E}\nabla=\left(\nabla^T\mathbf{E}^T\mathbf{m}\hat{\mathbf{f}}^T\mathbf{E}^T\mathbf{E}\nabla\right)^T=\nabla^T\mathbf{E}^T\mathbf{E}\hat{\mathbf{f}}\mathbf{m}^T\mathbf{E}\nabla\\&\nabla^T\mathbf{E}^T\mathbf{m}\nabla^T\mathbf{E}^T\mathbf{E}\nabla=\left(\nabla^T\mathbf{E}^T\mathbf{m}\nabla^T\mathbf{E}^T\mathbf{E}\nabla\right)^T=\nabla^T\mathbf{E}^T\mathbf{E}\nabla\mathbf{m}^T\mathbf{E}\nabla\end{aligned} \tag{27}$$

Using these two facts, the last two terms on the left-hand side cancel out the last two terms on the right-hand side in (26), respectively, so we are left with the following equation:

$$\nabla^T\mathbf{E}^T\mathbf{m}\hat{\mathbf{f}}^T\mathbf{E}^T\mathbf{E}\hat{\mathbf{f}}+\gamma\nabla^T\mathbf{E}^T\mathbf{m}\nabla^T\mathbf{E}^T\mathbf{E}\hat{\mathbf{f}}=\nabla^T\mathbf{E}^T\mathbf{E}\hat{\mathbf{f}}\mathbf{m}^T\mathbf{E}\hat{\mathbf{f}}+\gamma\nabla^T\mathbf{E}^T\mathbf{E}\nabla\mathbf{m}^T\mathbf{E}\hat{\mathbf{f}} \tag{28}$$

Finally, we can now extract the following analytical expression for γ:

$$\gamma=\frac{\nabla^T\mathbf{E}^T\mathbf{E}\hat{\mathbf{f}}\mathbf{m}^T\mathbf{E}\hat{\mathbf{f}}-\nabla^T\mathbf{E}^T\mathbf{m}\hat{\mathbf{f}}^T\mathbf{E}^T\mathbf{E}\hat{\mathbf{f}}}{\nabla^T\mathbf{E}^T\mathbf{m}\nabla^T\mathbf{E}^T\mathbf{E}\hat{\mathbf{f}}-\nabla^T\mathbf{E}^T\mathbf{E}\nabla\mathbf{m}^T\mathbf{E}\hat{\mathbf{f}}}. \tag{29}$$

The analytically derived framework is described by the following pseudo-code:

Algorithm 1: Projected Gradient Descent Unmixing (PGDU)

Inputs: $\mathbf{E}(\lambda \times L)$: Matrix of set of EMs

$\mathbf{H}(r \times c \times \lambda)$: Cube of spectral image to be unmixed

$\hat{\mathbf{F}}^0(r \times c \times L)$: Cube of initial estimate of fraction image

1) **for** each row in image (r times)
2) **for** each column in image (c times)
3) **while** $\max\left(\left|\hat{\mathbf{f}}^k - \hat{\mathbf{f}}^{k-t}\right|\right) > threshold$ *
4) Calculate $\nabla\phi\left(\hat{\mathbf{f}}^k\right)$ and γ_k by (18) and (29), respectively;
5) Set $\hat{\mathbf{f}}^{k+1} = \mathrm{P}_\Omega\left(\hat{\mathbf{f}}^k + \gamma_k \nabla\phi\left(\hat{\mathbf{f}}^k\right)\right)$;
6) **end**
7) **end**
8) **end**

* The term $\left|\hat{\mathbf{f}}^k - \hat{\mathbf{f}}^{k-t}\right|$ indicates the amount of change in fraction values during the last t iterations, where t is a predefined parameter

A convergence proof of the PGDU method (with supporting empirical evidence) is provided in the Appendix.

Special property of the objective function:

The effect of varying illumination on the measured mixture signature is usually represented by scalar multiplication. The advantage of the suggested objective function ϕ (which is a shape similarity measure) is its invariance to scalar multiplication as $\phi(\mathbf{m}, \mathbf{Ef}) = \phi(\mathbf{m}, a\mathbf{Ef})$, for any scalar $a \neq 0$. Geometrically, the global maximum of ϕ, for a given unmixing problem, extends along a straight line in fraction space (Fig. 4), i.e., all the points on this line represent the same fraction combination multiplied by a different scalar. Thus, using ϕ as an objective function for the unmixing process can significantly reduce the bias on the estimated fractions. Fig. 4 presents conceptually the advantage of using ϕ as an objective function compared with a least squares solution under varying illumination.

While the results obtained by VPGDU are influenced only slightly by the varying illumination effect, the biased results (due to the same effect) using SUnSAL are clearly evident (see Fig. 4). We further underscore empirically this advantage in Section V.

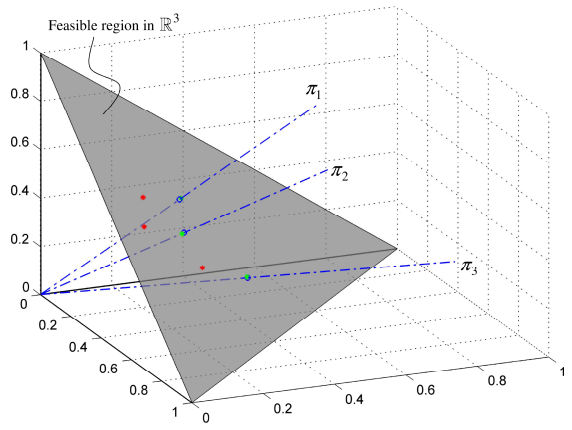

Fig. 4. Illustration for three pixel solutions obtained by SUnSAL and VPGDU under varying illumination effect; shaded triangle (ℓ_1 positive simplex) represents the feasible region for solution in $\mathbb{R}^3$, and each dashed-dotted line (π_1, π_2, π_3) represents the optimum line of the objective function for a pixel in question; green stars represent the true fraction values, and red stars and blue circles represent their SUnSAL and VPGDU estimates, respectively.

D. Code vectorization for speedup

A main drawback of standard gradient descent optimization is its typical slow convergence. Projected gradient descent performs faster since the solution is kept inside the feasible region during the entire process. However, the running time of our proposed *projected gradient descent unmixing* (PGDU) is still on the same order of FCLSU and GBM, which is considerably higher than that of SUnSAL and CLSUnSAL. We ran the procedures on 19 synthetic datasets of 200,000 mixed pixels, where each pixel in a given dataset contains (a subset of) 2, 3, …, 20 EMs. This was repeated 10 times (for each dataset).

Fig. 5 shows the median run-times per-pixel obtained running Matlab R2012b on a Microsoft 64-bit Windows 8 operating system with 32GB of RAM.

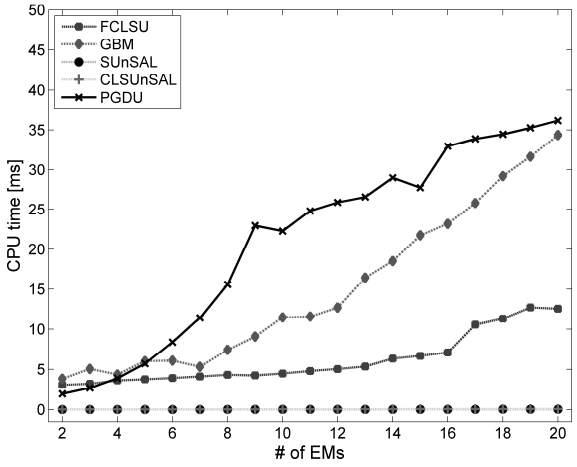

Fig. 5. Median run-times per-pixel (over 10 simulations) for FCLSU, GBM, SUnSAL, CLSUnSAL, and PGDU algorithms applied to 200,000 mixed pixels as a function of number of EMs.

Using the basic form of the projected gradient descent method, the running time rapidly increases as the number of EMs exceeds five. The process applied (to the entire image) iterates over the image rows and columns, until a pre-defined convergence criterion is met.

Pixel-wise algorithms as FCLSU, GBM, and PGDU suffer from a high computational cost and could prove impractical when applied to very large images. On the other hand, SUnSAL and CLSUnSAL each solve a single optimization problem with respect to the entire image; this coupled with efficient optimization considerably reduces their run-times. An efficient computation of our projected gradient descent method should exploit the analytical expression of the gradient and the step size, as well as the method's simple mathematical form, which does not involve a complex operation such as matrix inversion. Thus, the new module is adaptable to code vectorization, which compresses the entire process into a single loop of converging iterations; a new fraction vector is calculated in each iteration for all of the image pixels simultaneously by using the vectorized form of (18) and (29).

In addition to the gained speedup of the process, the suggested vectorized code has the advantage of applying an independent optimization problem per pixel in the image exactly as the pixelwise algorithm. This advantage can be very useful in cases where different constraints are applied on the different pixels, what is not over methods that apply a single large optimization problem.

We present below some matrix operations to be used for this mechanism:

Let $\underset{l\times 1}{\mathbf{1}} = [1,1,...,1]^T$ and let $\mathbf{U},\mathbf{V} \in \mathbb{R}^{l\times s}$ be the matrices

$$\mathbf{U} = \begin{bmatrix} u_{1,1} & u_{1,2} & . & u_{1,s} \\ u_{2,1} & u_{2,2} & . & . \\ . & . & . & . \\ u_{l,1} & . & . & u_{l,s} \end{bmatrix} \quad \text{and} \quad \mathbf{V} = \begin{bmatrix} v_{1,1} & v_{1,2} & . & v_{1,s} \\ v_{2,1} & v_{2,2} & . & . \\ . & . & . & . \\ v_{l,1} & . & . & v_{l,s} \end{bmatrix}$$

The Hadamard (entry-wise) product is defined as:

$$\mathbf{U} \circ \mathbf{V} = \begin{bmatrix} u_{1,1}v_{1,1} & u_{1,2}v_{1,2} & . & u_{1,s}v_{1,s} \\ u_{2,1}v_{2,1} & u_{2,2}v_{2,2} & . & . \\ . & . & . & . \\ u_{l,1}v_{l,1} & . & . & u_{l,s}v_{l,s} \end{bmatrix} \tag{30}$$

We now present the following set of array operations for vectorizing the code:

Operation 1: Array multiplication $(.*)$, the same as the Hadamard product.

Operation 2: Array right division $(./)$, that is,

$$\mathbf{U}./\mathbf{V} = \begin{bmatrix} \frac{u_{1,1}}{v_{1,1}} & \frac{u_{1,2}}{v_{1,2}} & . & \frac{u_{1,s}}{v_{1,s}} \\ \frac{u_{2,1}}{v_{2,1}} & \frac{u_{2,2}}{v_{2,2}} & . & . \\ . & . & . & . \\ \frac{u_{l,1}}{v_{l,1}} & . & . & \frac{u_{l,s}}{v_{l,s}} \end{bmatrix}$$

Operation 3: Array power $(.\hat{})$, that is,

$$\mathbf{U}.\hat{}\, q = \begin{bmatrix} {u_{1,1}}^q & {u_{1,2}}^q & . & {u_{1,s}}^q \\ {u_{2,1}}^q & {u_{2,2}}^q & . & . \\ . & . & . & . \\ {u_{l,1}}^q & . & . & {u_{l,s}}^q \end{bmatrix}$$

Creating a row vector containing the sums of matrix columns can be simply done as

$$\mathbf{1}^T\mathbf{U} = \left[\sum_{i=1}^{l} u_{i,1}, \sum_{i=1}^{l} u_{i,2}, \ldots, \sum_{i=1}^{l} u_{i,s}\right] \tag{31}$$

Converting a row vector into a matrix with the same vector duplicated along the rows: Letting $\mathbf{w} = [w_1, w_2, ...w_s]$, we have

$$\mathbf{1w} = \begin{bmatrix} 1 \\ 1 \\ . \\ . \\ 1 \end{bmatrix} [w_1, w_2, \ldots, w_s] = \begin{bmatrix} w_1, w_2, \ldots, w_s \\ w_1, w_2, \ldots, w_s \\ . \quad\quad . \\ . \quad\quad . \\ w_1, w_2, \ldots, w_s \end{bmatrix} \tag{32}$$

Using the above operations enables code vectorization of the gradient descent unmixing process. We are given a spectral image (with r rows, c columns, and λ bands), a matrix of L EMs, and a fraction image (with r rows, c columns and L bands) obtained by the initial estimation process. A matrix $\mathbf{M}$ (with λ rows and $r \cdot c$ columns) can then be created by permuting the spectral image as shown in Fig. 6. The same operation is applied to the estimated fraction image to create the matrix $\hat{\mathbf{F}}$ (with L rows and $r \cdot c$ columns); each column in $\mathbf{M}$ and $\hat{\mathbf{F}}$ contains the spectral signature and estimated fraction vector of the corresponding pixel in the spectral image and the fraction image, respectively.

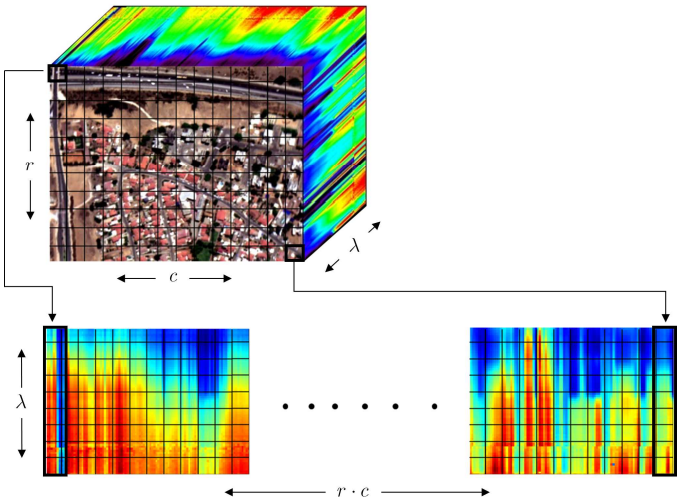

Fig. 6. Spectral cube permutation; each pixel in the 3D image is converted to a column in a 2D matrix; code vectorization is applied based on matrix operations.

Having defined $\mathbf{M}$, $\mathbf{E}$, and $\hat{\mathbf{F}}$, we now implement the *vectorization code of the projected gradient descent unmixing* (VPGDU). Specifically, the gradient $\nabla\phi\left(\hat{\mathbf{f}}^k\right)$ and the optimal step size γ_k can be simultaneously calculated for all pixels by the following operations:

$$\nabla_1 = \mathbf{E}^T\mathbf{M} \tag{33}$$

$$\nabla_2 = \mathbf{1}\cdot\mathbf{1}^T\left(\left(\mathbf{E}\hat{\mathbf{F}}^k\right)\circ\left(\mathbf{E}\hat{\mathbf{F}}^k\right)\right) \tag{34}$$

$$\nabla_3 = \mathbf{E}^T\mathbf{E}\hat{\mathbf{F}}^k \tag{35}$$

$$\nabla_4 = \mathbf{1}\cdot\mathbf{1}^T\left(\mathbf{M}\circ\left(\mathbf{E}\hat{\mathbf{F}}^k\right)\right) \tag{36}$$

$$\nabla_5 = \mathbf{1}\cdot\mathbf{1}^T\left(\mathbf{M}\circ\mathbf{M}\right).\wedge 0.5 \tag{37}$$

$$\nabla_n = \nabla_1\circ\nabla_2 - \nabla_3\circ\nabla_4 \tag{38}$$

$$\nabla_d = \nabla_5\circ\left(\nabla_2.\wedge 1.5\right) \tag{39}$$

Thus, the gradient for each pixel in the image is given by:

$$\underset{entire\ image}{\nabla_{\hat{\mathbf{F}}^k}} = \nabla_n./\nabla_d \tag{40}$$

To obtain the step size, we compute the following:

$$\mathbf{\Gamma}_1 = \mathbf{1}^T\left(\mathbf{M}\circ\left(\mathbf{E}\hat{\mathbf{F}}^k\right)\right) \tag{41}$$

$$\mathbf{\Gamma}_2 = \mathbf{1}^T\left(\nabla_{\hat{\mathbf{F}}^k}\circ\left(\mathbf{E}^T\mathbf{E}\hat{\mathbf{F}}^k\right)\right) \tag{42}$$

$$\mathbf{\Gamma}_3 = \mathbf{1}^T\left(\hat{\mathbf{F}}^k\circ\left(\mathbf{E}^T\mathbf{E}\hat{\mathbf{F}}^k\right)\right) \tag{43}$$

$$\mathbf{\Gamma}_4 = \mathbf{1}^T\left(\nabla_{\hat{\mathbf{F}}^k}\circ\left(\mathbf{E}^T\mathbf{E}\nabla\right)\right) \tag{44}$$

$$\mathbf{\Gamma}_5 = \mathbf{1}^T\left(\nabla_{\hat{\mathbf{F}}^k}\circ\left(\mathbf{E}^T\mathbf{M}\right)\right) \tag{45}$$

$$\mathbf{\Gamma}_n = \mathbf{\Gamma}_2\circ\mathbf{\Gamma}_1 - \mathbf{\Gamma}_5\circ\mathbf{\Gamma}_3 \tag{46}$$

$$\mathbf{\Gamma}_d = \mathbf{\Gamma}_5\circ\mathbf{\Gamma}_2 - \mathbf{\Gamma}_4\circ\mathbf{\Gamma}_1 \tag{47}$$

And the step size for each pixel in the image is given by

$$\mathbf{\Gamma}_k = \mathbf{1}\left(\mathbf{\Gamma}_n./\mathbf{\Gamma}_d\right) \tag{48}$$

The iterative step at each pixel is calculated simultaneously by

$$\hat{\mathbf{F}}^{k+1} = \mathbf{P}_\Omega\left(\hat{\mathbf{F}}^k + \mathbf{\Gamma}_k\circ\nabla_{\hat{\mathbf{F}}^k}\right) \tag{49}$$

We used the vectorized code provided in [49] for the projection operator $\mathbf{P}_\Omega$.

Testing the running time of VPGDU relatively to FCLSU, GBM, SUnSAL, and CLSUnSAL reveals a significant improvement. The results are shown in Fig. 7.

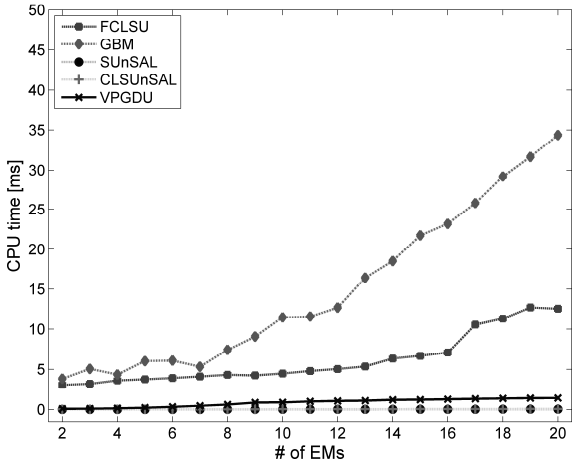

Fig. 7. Median run-times per-pixel (over 10 simulations) for FCLSU, GBM, SUnSAL, CLSUnSAL, and VPGDU algorithms applied to 200,000 mixed pixels as a function of number of EMs (FCLSU and GBM are not vectorized).

The vectorized code of the PGDU algorithm is described by the pseudo-code below.

Algorithm 2: Vectorized Code Projected Gradient Descent Unmixing (VPGDU)

Inputs: $\mathbf{E}(\lambda\times L)$: Matrix of set of EMs

$\mathbf{H}(r\times c\times\lambda)$: Cube of spectral image to be unmixed

$\hat{\mathbf{F}}^0(r\times c\times L)$: Cube of initial estimate of fraction image

1) Create $\mathbf{M}$ and $\hat{\mathbf{F}}$ by permuting $\mathbf{H}$ and $\hat{\mathbf{F}}^0$, respectively;

2) **while** $\left(\left(\max\left(\left|\hat{\mathbf{F}}^k - \hat{\mathbf{F}}^{k-t}\right|\right) > threshold\right)\vee\left(\mathbf{M}\neq\varnothing\right)\right)$

3) Compute $\nabla_{\hat{\mathbf{F}}^k}$ and $\mathbf{\Gamma}_k$ by (33)—(40) and (41)—(48), respectively;

4) Set $\hat{\mathbf{F}}^{k+1} = \mathbf{P}_\Omega\left(\hat{\mathbf{F}}^k + \mathbf{\Gamma}_k\circ\nabla_{\hat{\mathbf{F}}^k}\right)$;

5) Remove from $\mathbf{M}$ all pixels for which process converged;

6) **end**

* The term $\left|\hat{\mathbf{F}}^k - \hat{\mathbf{F}}^{k-t}\right|$ indicates the amount of change in fraction values during the last t iterations, where t is a predefined parameter

The results clearly reveal the superior efficiency of SUnSAL and CLSUnSAL, whose run-times are almost

invariant with respect to the number of EMs used in the unmixing. Thus, these methods can be considered as a reference of efficiency with respect to newly proposed methods. Although the run-time of VPGDU is considerably faster than (the original PGDU and) the standard, off-the-shelf FCLSU, it is about an order of magnitude slower than SUnSAL and CLSUnSAL. VPGDU, as well as SUnSAL and CLSUnSAL use parallel processing to solve for all the image pixels. However, whereas the latter solve a single optimization problem applied to the entire image, VPGDU solves for each pixel an independent optimization problem, running concurrently on all pixels.

V. Experimental Results

A comparative performance evaluation of VPGDU was carried out relatively to the standard FCLSU, GBM [25], and state-of-the-art SUnSAL [11][1] and CLSUnSAL [11][2]. We experimented with the following datasets:

Set A: Contains 20 spectral signatures (with 224 bands) selected from the USGS digital spectral library.

Set B: Contains 20 spectral signatures extracted automatically, using the fast VCA algorithm [55], from a 250×190 sub-image (containing 188 out of 224 bands after removal of noisy bands); the image was selected from the well-known Cuprite reflectance image acquired by the AVIRIS sensor in 1997 (Fig. 13). The data for creating both sets A and B are available online from [55].

Set C: Contains a real 72-band AISA image of size 234×284 acquired over Hadera, Israel in 2006 (Fig. 14) and 14 EMs derived manually from the image itself.

Set D: Contains the Cuprite real reflectance image and 12 EMs extracted automatically from the image itself, using the VCA algorithm [55].

In addition to experimenting with the real image, four different synthetic tests were created using data sets A and B; see Fig. 8.

1) Experiment 1

A set of 10,000 synthetic mixed pixels was generated using the 20 EMs of Set A. The fractions in each simulated pixel follow a Dirichlet distribution [55] and fulfill the ANC and ASC constraints. An additive white Gaussian noise was added to the mixed spectra with SNRs of 30dB and 20dB. The well-known *root mean square error* (RMSE) [56] for each EM was calculated as follows:

$$RMSE_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(f_{i,j}-\hat{f}_{i,j}\right)^2}, \qquad (50)$$

where i denotes the i^{th} EM, N =10,000 is the number of pixels in the set, and $f_{i,j}$ and $\hat{f}_{i,j}$ are the true and estimated fractions, respectively, of the i^{th} EM in the j^{th} pixel. The experiment was applied to both Sets A and B; the results are presented in Tables I and II.

2) Experiment 2

This experiment is similar to the previous one, except that each generated spectral mixture was multiplied by a random number between 0.7 and 1 to simulate a varying illumination effect. We evaluated again the unmixing performance, for each EM, by the RMSE expression in (50). The results are presented in Tables III and IV.

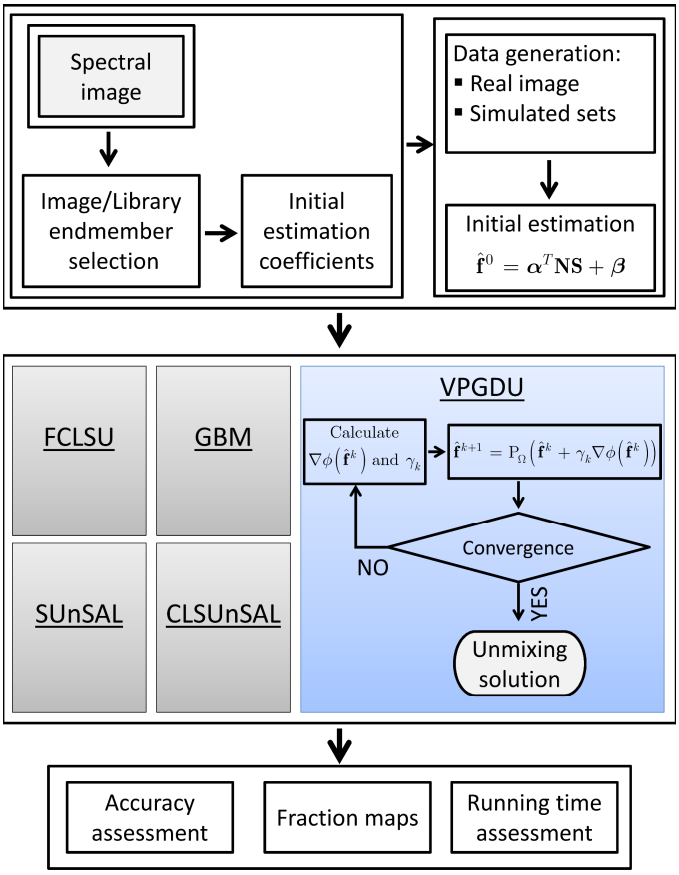

Fig. 8. Overview of comparative experimental study.

[1] with 'lambda' = 1, 'POSITIVITY' = yes, 'ADDONE' = yes, and 'AL_ITERS' = 1000

[2] with 'POSITIVITY'=yes, 'ADDONE'=yes, 'lambda'= 3E-4, 'AL_ITERS'=2000, 'TOL'=1E-8)

TABLE I
RMSE values (per each EM from Set A) for FCLSU, GBM, SUnSAL, CLSUnSAL, and VPGDU

EM#	SNR=20dB					SNR=30dB				
	FCLSU	**GBM**	**SUnSAL**	**CLSUnSAL**	**VPGDU**	**FCLSU**	**GBM**	**SUnSAL**	**CLSUnSAL**	**VPGDU**
1	0.0411	0.0446	0.0411	0.0411	0.0425	0.0193	0.0192	0.0194	0.0193	**0.0177**
2	0.0356	0.0362	0.0356	0.0356	**0.0353**	0.0165	0.0167	0.0165	0.0165	**0.0147**
3	0.0357	0.0364	0.0357	0.0357	0.0458	0.0214	0.0214	0.0215	0.0214	0.0219
4	0.0355	0.0356	0.0355	0.0355	0.0469	0.0192	0.0187	0.0192	0.0192	0.0197
5	0.0307	0.0306	0.0307	0.0307	0.0329	0.0164	0.0160	0.0164	0.0164	**0.0143**
6	0.0331	0.0349	0.0331	0.0331	0.0348	0.0157	0.0159	0.0157	0.0157	**0.0143**
7	0.0310	0.0311	0.0310	0.0310	0.0599	0.0186	0.0183	0.0186	0.0186	0.0241
8	0.0480	0.0484	0.0[]0	0.0480	0.0543	0.0257	0.0255	0.0258	0.0257	**0.0239**
9	0.0195	0.0191	0.0195	0.0195	**0.0190**	0.0100	0.0095	0.0100	0.0100	**0.0081**
10	0.0307	0.0310	0.0307	0.0307	0.0334	0.0152	0.0152	0.0152	0.0152	**0.0138**
11	0.0344	0.0362	0.0344	0.0344	**0.0342**	0.0168	0.0167	0.0169	0.0168	**0.0157**
12	0.0474	0.0480	0.0474	0.0474	0.0539	0.0279	0.0275	0.0280	0.0279	**0.0261**
13	0.0610	0.0605	0.0610	0.0610	0.0711	0.0319	0.0313	0.0320	0.0319	0.0338
14	0.0482	0.0529	0.0482	0.0482	0.0773	0.0398	0.0440	0.0399	0.0398	0.0429
15	0.0342	0.0348	0.0342	0.0342	0.0453	0.0193	0.0194	0.0194	0.0193	0.0201
16	0.0293	0.0306	0.0293	0.0293	0.0368	0.0172	0.0174	0.0173	0.0172	0.0174
17	0.0269	0.0271	0.0269	0.0269	**0.0268**	0.0121	0.0122	0.0121	0.0121	**0.0105**
18	0.0279	0.0282	0.0279	0.0279	0.0304	0.0132	0.0131	0.0132	0.0132	0.0141
19	0.0319	0.0338	0.0319	0.0319	0.0344	0.0160	0.0162	0.0161	0.0160	**0.0144**
20	0.0355	0.0359	0.0355	0.0355	0.0366	0.0164	0.0167	0.0164	0.0164	0.0168
mean	0.0359	0.0368	0.0359	0.0359	0.0426	0.0194	0.0195	0.0195	0.0194	**0.0192**

TABLE II
RMSE values (per each EM from Set B) for FCLSU, GBM, SUnSAL, CLSUnSAL, and VPGDU

EM#	SNR=20dB					SNR=30dB				
	FCLSU	**GBM**	**SUnSAL**	**CLSUnSAL**	**VPGDU**	**FCLSU**	**GBM**	**SUnSAL**	**CLSUnSAL**	**VPGDU**
1	0.0566	0.0627	0.0566	0.0566	0.0593	0.0276	0.0284	0.0277	0.0277	**0.0268**
2	0.0904	0.0921	0.0906	0.0904	0.1025	0.0447	0.0450	0.0453	0.0447	0.0477
3	0.0263	0.0274	0.0263	0.0263	0.0395	0.0133	0.0134	0.0133	0.0133	0.0170
4	0.0529	0.0606	0.0529	0.0529	0.0559	0.0256	0.0262	0.0256	0.0256	**0.0247**
5	0.0549	0.0563	0.0549	0.0549	0.0638	0.0242	0.0243	0.0242	0.0242	0.0251
6	0.0494	0.0523	0.0494	0.0494	0.0627	0.0222	0.0222	0.0223	0.0222	0.0262
7	0.0746	0.0754	0.0746	0.0746	**0.0743**	0.0336	0.0338	0.0336	0.0336	**0.0333**
8	0.0760	0.0803	0.0760	0.0760	0.0783	0.0344	0.0348	0.0345	0.0344	**0.0342**
9	0.0938	0.0947	0.0937	0.0938	0.0967	0.0487	0.0486	0.0489	0.0487	**0.0471**
10	0.0642	0.0664	0.0642	0.0642	0.0708	0.0277	0.0280	0.0277	0.0277	0.0284
11	0.0906	0.1041	0.0906	0.0906	0.0973	0.0472	0.0493	0.0476	0.0472	0.0488
12	0.0833	0.0840	0.0833	0.0833	0.0914	0.0372	0.0372	0.0372	0.0372	0.0392
13	0.0909	0.0938	0.0909	0.0909	0.1060	0.0481	0.0482	0.0483	0.0482	0.0497
14	0.1079	0.1096	0.1082	0.1079	0.1189	0.0680	0.0696	0.0690	0.0681	**0.0612**
15	0.0904	0.0917	0.0903	0.0904	0.0926	0.0474	0.0472	0.0477	0.0474	0.0475
16	0.1417	0.1419	0.1409	0.1416	**0.1358**	0.0894	0.0886	0.0958	0.0894	**0.0781**
17	0.1213	0.1264	0.1209	0.1213	0.1294	0.0656	0.0656	0.0665	0.0656	**0.0650**
18	0.0524	0.0664	0.0523	0.0524	0.0657	0.0274	0.0291	0.0275	0.0274	0.0297
19	0.1449	0.1470	0.1444	0.1449	**0.1443**	0.0876	0.0889	0.0994	0.0877	**0.0875**
20	0.1364	0.1365	0.1360	0.1364	0.1384	0.0813	0.0813	0.0850	0.0815	**0.0785**
mean	0.0849	0.0885	0.0849	0.0849	0.0912	0.0450	0.0455	0.0464	0.0451	**0.0448**

.

TABLE III
RMSE values (per each EM from Set A), under varying illumination effect, for FCLSU, GBM, SUnSAL, CLSUnSAL, and VPGDU

EM#	SNR=20dB					SNR=30dB				
	FCLSU	**GBM**	**SUnSAL**	**CLSUnSAL**	**VPGDU**	**FCLSU**	**GBM**	**SUnSAL**	**CLSUnSAL**	**VPGDU**
1	0.0775	0.0745	0.0775	0.0775	**0.0413**	0.0737	0.0675	0.0737	0.0737	**0.0179**
2	0.0742	0.0741	0.0742	0.0742	**0.0375**	0.0692	0.0677	0.0692	0.0692	**0.0146**
3	0.0892	0.0870	0.0890	0.0892	**0.0452**	0.0813	0.0777	0.0811	0.0813	**0.0222**
4	0.0896	0.0868	0.0896	0.0896	**0.0469**	0.0841	0.0822	0.0841	0.0841	**0.0211**
5	0.0835	0.0776	0.0835	0.0835	**0.0329**	0.0813	0.0754	0.0813	0.0813	**0.0143**
6	0.0719	0.0725	0.0719	0.0719	**0.0352**	0.0679	0.0679	0.0679	0.0679	**0.0138**
7	0.0905	0.0883	0.0905	0.0905	**0.0592**	0.0837	0.0821	0.0838	0.0837	**0.0259**
8	0.1093	0.1093	0.1092	0.1093	**0.0551**	0.1023	0.1026	0.1022	0.1023	**0.0243**
9	0.0572	0.0481	0.0571	0.0572	**0.0205**	0.0546	0.0471	0.0545	0.0546	**0.0081**
10	0.0724	0.0710	0.0724	0.0724	**0.0340**	0.0680	0.0668	0.0680	0.0680	**0.0132**
11	0.0671	0.0627	0.0671	0.0671	**0.0354**	0.0640	0.0579	0.0640	0.0640	**0.0153**
12	0.1240	0.1200	0.1240	0.1240	**0.0564**	0.1206	0.1171	0.1206	0.1206	**0.0265**
13	0.1229	0.1195	0.1229	0.1229	**0.0715**	0.1164	0.1151	0.1165	0.1164	**0.0346**
14	0.2949	0.3224	0.2949	0.2949	**0.0770**	0.3042	0.3294	0.3040	0.3042	**0.0431**
15	0.0947	0.0951	0.0947	0.0947	**0.0458**	0.0908	0.0912	0.0909	0.0908	**0.0201**
16	0.0802	0.0804	0.0802	0.0802	**0.0371**	0.0741	0.0735	0.0742	0.0741	**0.0174**
17	0.0663	0.0680	0.0663	0.0663	**0.0279**	0.0633	0.0646	0.0633	0.0633	**0.0116**
18	0.0714	0.0696	0.0714	0.0714	**0.0299**	0.0689	0.0665	0.0689	0.0689	**0.0147**
19	0.0741	0.0748	0.0741	0.0741	**0.0344**	0.0710	0.0706	0.0710	0.0710	**0.0141**
20	0.0811	0.0791	0.0811	0.0811	**0.0369**	0.0763	0.0745	0.0763	0.0763	**0.0146**
mean	0.0946	0.0940	0.0946	0.0946	**0.0430**	0.0908	0.0899	0.0908	0.0908	**0.0194**

TABLE IV
RMSE values (per each EM from Set B), under varying illumination effect, for FCLSU, GBM, SUnSAL, CLSUnSAL, and VPGDU

EM#	SNR=20dB					SNR=30dB				
	FCLSU	**GBM**	**SUnSAL**	**CLSUnSAL**	**VPGDU**	**FCLSU**	**GBM**	**SUnSAL**	**CLSUnSAL**	**VPGDU**
1	0.0851	0.0838	0.0851	0.0851	**0.0596**	0.0790	0.0744	0.0790	0.0790	**0.0271**
2	0.1440	0.1416	0.1439	0.1440	**0.1028**	0.1373	0.1350	0.1373	0.1373	**0.0485**
3	0.0609	0.0605	0.0609	0.0609	**0.0396**	0.0564	0.0561	0.0564	0.0564	**0.0175**
4	0.0864	0.0841	0.0863	0.0864	**0.0542**	0.0801	0.0745	0.0801	0.0801	**0.0240**
5	0.0988	0.0983	0.0988	0.0988	**0.0634**	0.0910	0.0908	0.0910	0.0910	**0.0251**
6	0.0890	0.0881	0.0890	0.0890	**0.0649**	0.0767	0.0757	0.0767	0.0767	**0.0263**
7	0.1113	0.1105	0.1112	0.1113	**0.0775**	0.0980	0.0975	0.0980	0.0980	**0.0332**
8	0.1155	0.1109	0.1155	0.1155	**0.0765**	0.1056	0.0979	0.1056	0.1056	**0.0345**
9	0.1388	0.1402	0.1388	0.1388	**0.0987**	0.1232	0.1213	0.1232	0.1232	**0.0460**
10	0.1085	0.1074	0.1085	0.1085	**0.0713**	0.0997	0.0982	0.0997	0.0997	**0.0282**
11	0.1066	0.1187	0.1066	0.1066	**0.0975**	0.0960	0.0921	0.0960	0.0960	**0.0503**
12	0.1170	0.1162	0.1170	0.1170	**0.0934**	0.0988	0.0985	0.0988	0.0988	**0.0401**
13	0.1428	0.1405	0.1428	0.1428	**0.1065**	0.1352	0.1324	0.1352	0.1352	**0.0495**
14	0.4720	0.4748	0.4730	0.4720	**0.1163**	0.4763	0.4768	0.4772	0.4763	**0.0622**
15	0.1452	0.1379	0.1452	0.1452	**0.0947**	0.1381	0.1291	0.1381	0.1381	**0.0476**
16	0.2535	0.2517	0.2530	0.2535	**0.1362**	0.2473	0.2503	0.2462	0.2472	**0.0798**
17	0.1619	0.1611	0.1618	0.1619	**0.1290**	0.1571	0.1562	0.1570	0.1571	**0.0661**
18	0.0858	0.0898	0.0858	0.0858	**0.0647**	0.0771	0.0721	0.0771	0.0771	**0.0312**
19	0.1745	0.1847	0.1721	0.1745	**0.1445**	0.1622	0.1682	0.1617	0.1621	**0.0894**
20	0.1563	0.1561	0.1562	0.1563	**0.1379**	0.1545	0.1542	0.1546	0.1545	**0.0796**
mean	0.1427	0.1429	0.1426	0.1427	**0.0915**	0.1345	0.1326	0.1345	0.1345	**0.0453**

The RMSE values (per EM) for all the methods seem correlative and appear to increase consistently as the SNR decreases to 20dB. All of the methods perform in a comparable manner; VPGDU is slightly more accurate for SNR=30dB. Also, note that the RMSE values of all the methods increase considerably when using the EMs of Set B. While the RMSE values for VPGDU remain roughly fixed, under the varying illumination, those obtained by FCLSU, GBM, SUnSAL, and CLSUnSAL increase significantly.

3) Experiment 3

We generated 19 synthetic image sets, each containing 10,000 mixed pixels. Each image is associated with a certain number of EMs from which the spectral mixture of each pixel is composed of. Specifically, the first image consists of pixels containing only two EMs, the second image consists of pixels containing three EMs, and so on; finally, the 19th image is composed of pixels containing all 20 EMs. The fractions in each simulated pixel follow a Dirichlet distribution and fulfill the ANC and ASC constraints. An additive white Gaussian noise was added to the mixed spectra with SNRs of 30dB and 20dB. We evaluated the unmixing performance, for each synthetic image, by the following RMSE measure:

$$RMSE = \frac{1}{N}\sum_{j=1}^{N}\left(\sqrt{\frac{1}{L}\sum_{i=1}^{L}\left(f_{i,j}-\hat{f}_{i,j}\right)^2}\right), \qquad (51)$$

where N = 10,000 is the number of pixels in each synthetic image, L = 20 is the number of EMs, and $f_{i,j}$ and $\hat{f}_{i,j}$ are the true and estimated fractions, respectively, of the i^{th} EM in the j^{th} pixel. Note that all 20 EMs are utilized during the unmixing, while taking into account the effect of non-participating EMs [57]. The experiment was applied to Sets A and B and was repeated by decreasing the number of spectral bands from 224/188 to 112/94 to 45/40 (for A/B, respectively), while increasing, accordingly, the bandwidth from 10 nm to 20 nm to 50 nm. Fig. 9 and Fig. 10 show the RMSE obtained for FCLSU, GBM, SUnSAL, CLSUnSAL, and VPGDU as a function of the actual number of participating EMs for Sets A and B.

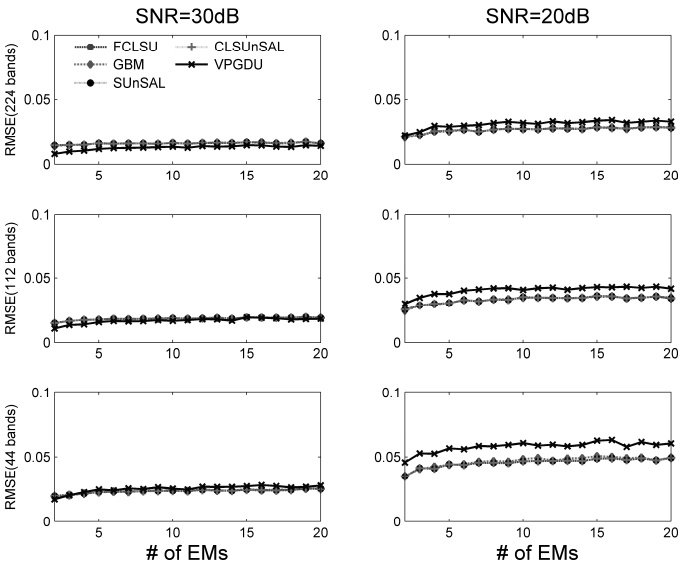

Fig. 9. RMSE (using Set A) for FCLSU, GBM, SUnSAL, CLSUnSAL, and VPGDU vs. actual number of EMs at two SNR levels and three spectral resolutions (bandwidth: 10, 20, and 40 nm).

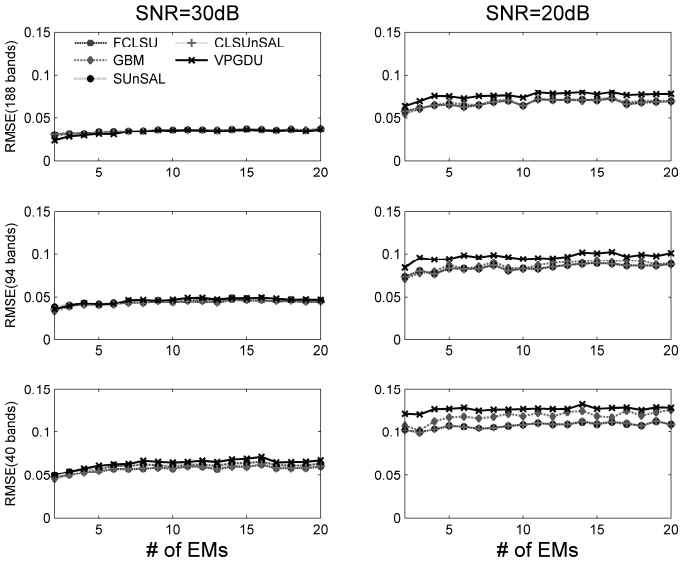

Fig. 10. RMSE (using Set B) for FCLSU, GBM, SUnSAL, CLSUnSAL, and VPGDU vs. actual number of EMs at two SNR levels and three spectral resolutions (bandwidth: 10, 20, and 40 nm).

4) Experiment 4

This experiment is similar to the previous one, except that each generated spectral mixture was multiplied by a random number between 0.7 and 1 to simulate a varying illumination effect. As before, we evaluated the unmixing performance, for each synthetic image, in terms of the RMSE expression in (51). Fig. 11 and Fig. 12 show the RMSE obtained for FCLSU, GBM, SUnSAL, CLSUnSAL, and VPGDU as a function of the actual number of participating EMs for the Sets A and B, respectively.

The RMSE values for the five methods increase consistently as the SNR and the spectral resolution decrease. The results for all the methods are highly correlative. FCLSU, GBM, SUnSAL, and CLSUnSAL are slightly advantageous for SNR=20dB (especially for low spectral resolution). As in the previous experiments, the RMSE values for the five methods are considerably higher when using the spectra of Set B. This could be attributed mainly to the higher collinearity between the spectra of Set B (compared to the one between the spectra of Set A).

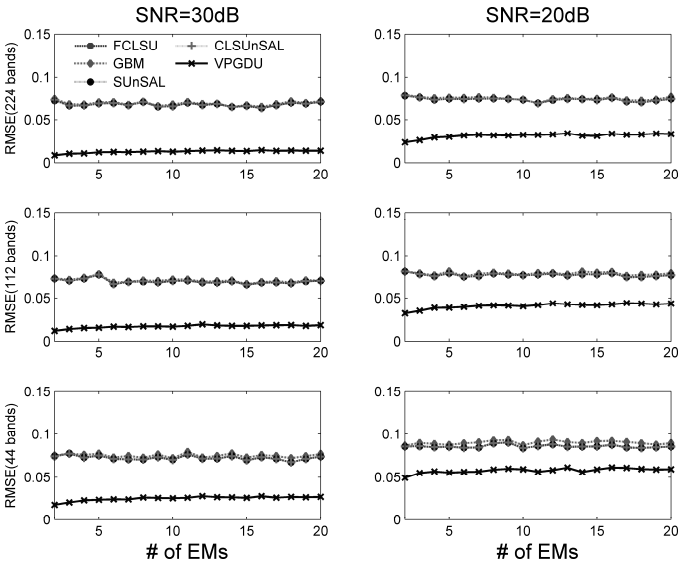

Fig. 11. RMSE (using Set A), under varying illumination, for FCLSU, GBM, SUnSAL, CLSUnSAL, and VPGDU vs. actual number of EMs at two SNR levels and three spectral resolutions (bandwidth: 10, 20, and 40 nm).

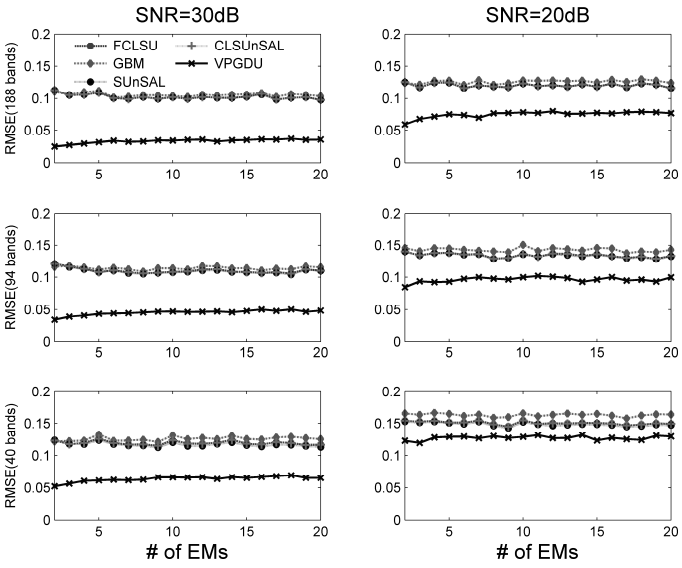

Fig. 12. RMSE (using Set B), under varying illumination, for, FCLSU, GBM, SUnSAL, CLSUnSAL, and VPGDU vs. actual number of EMs at two SNR levels and three spectral resolutions (bandwidth: 10, 20, and 40 nm).

As can be seen from Fig. 11 and, Fig. 12 the robustness of VPGDU to illumination change yields RMSE values that are fairly fixed. Note, on the other hand, the increased RMSE values obtained for FCLSU, GBM, SUnSAL, and CLSUnSAL. In other words, VPGDU significantly outperforms these methods under varying illumination.

5) Experiment 5 (using real data)

FCLSU, GBM, SUnSAL, CLSUnSAL, and VPGDU were applied to the real Cuprite image (Fig. 13), taking into account 12 EMs extracted from it automatically via VCA and to the real AISA image (Fig. 14), taking into consideration 12 of the 14 EMs selected from it manually.

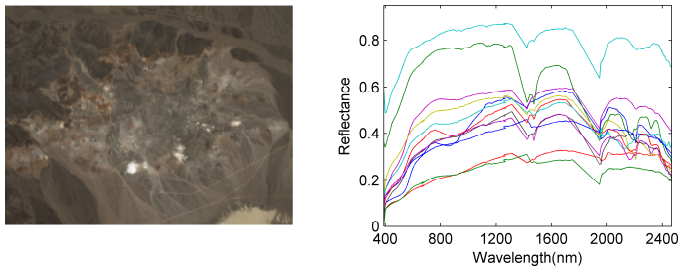

Fig. 13. RGB composite (left) and reflectance spectra (right) of the 12 EMs automatically extracted from the Cuprite image using VCA.

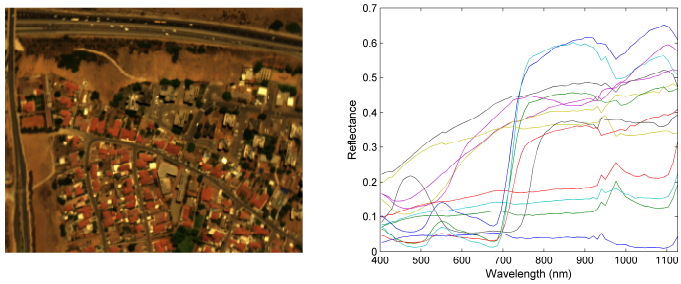

Fig. 14. RGB composite (left) and reflectance spectra (right) of the 14 EMs selected from the 2006 AISA image, containing 4 vegetation types, 4 soil/rock types, 3 kinds of pavement, and 3 roof types.

Fig. 15 and Fig. 16, and Fig. 17 and Fig. 18 below show the estimated fraction maps for all EMs obtained by FCLSU, SUnSAL, and VPGDU for both the AISA and Cuprite images, respectively. (These results are comparable to those obtained by GBM and CLSUnSAL, which are available from our supplementary material.)

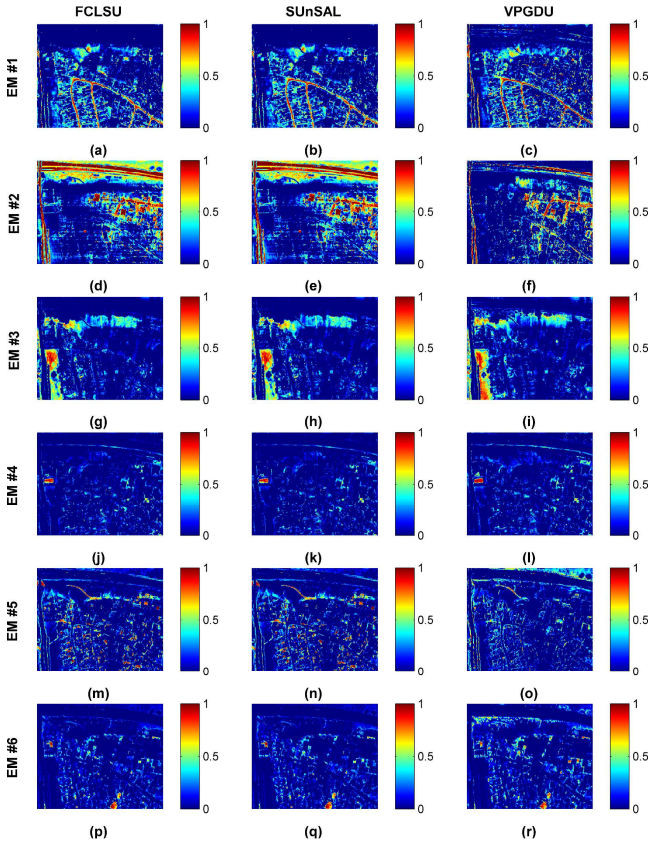

Fig. 15. Abundance fraction images for EMs 1, 2, 3, 4, 5, and 6 as estimated by FCLSU, SUnSAL, and VPGDU applied to AISA image.

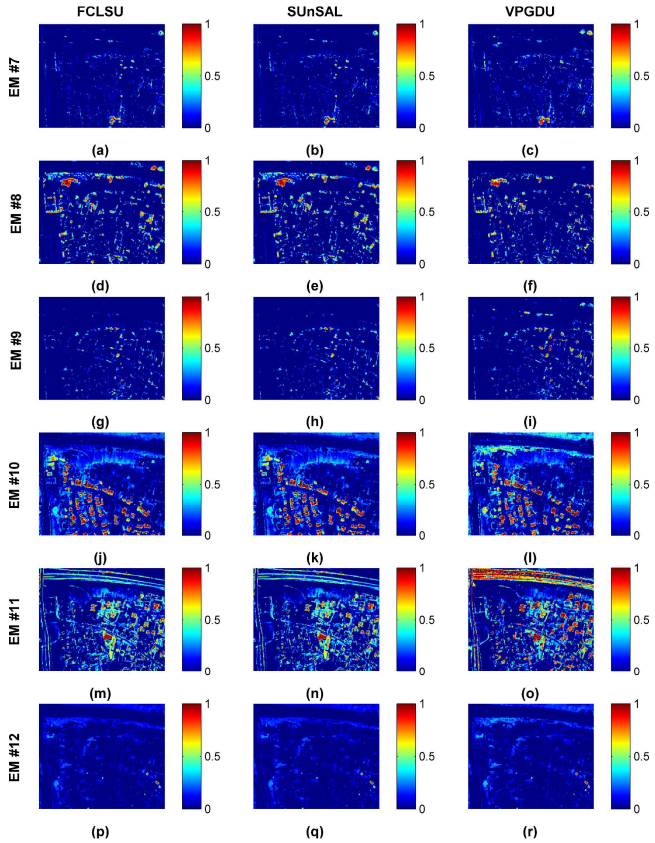

Fig. 16. Abundance fraction images for EMs 7, 8, 9, 10, 11, and 12 as estimated by FCLSU, SUnSAL, and VPGDU applied to AISA image.

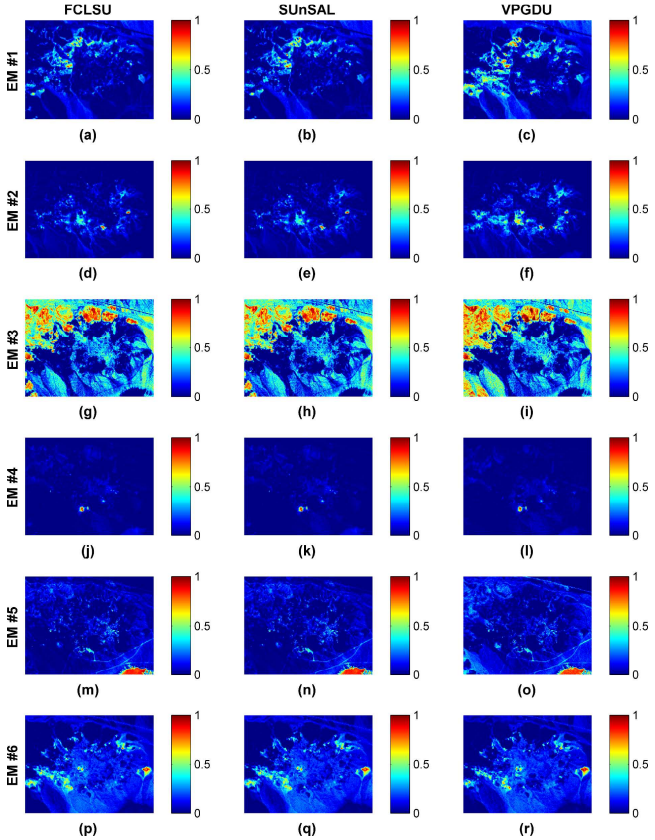

Fig. 17. Abundance fraction images for EMs 1, 2, 3, 4, 5, and 6 as estimated by FCLSU, SUnSAL, and VPGDU applied to Cuprite image

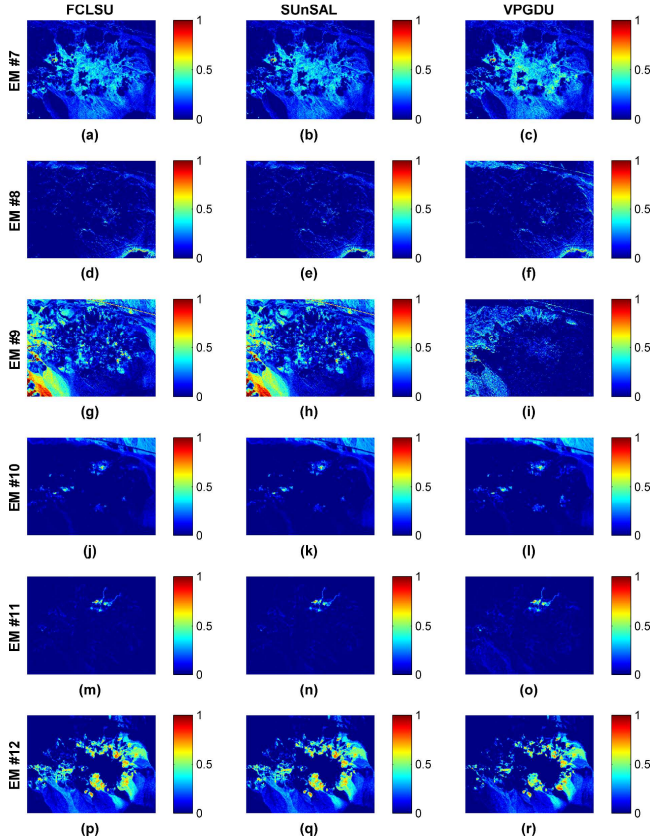

Fig. 18. Abundance fraction images for EMs 6, 7, 8, 9, 10, and 12 as estimated by FCLSU, SUnSAL, and VPGDU applied to Cuprite image.

As can be observed, the results obtained by FCLSU and SUnSAL are fairly close; also, they largely agree with those obtained by VPGDU for both images, modulo some differences in a few EM fractions. For example, in the AISA image, FCLSU and SUnSAL tend to overestimate the fractions of the second Asphalt type in areas of dark soil, as can be noticed in the fraction maps of EM #2 (Fig. 15 (d), (e), and (f)). On the other hand, VPGDU overestimates the fractions of Concrete, especially along the main road; see the fraction maps of EM #11 (Fig. 16 (m), (n), and (o)). The results obtained by all the methods for the Cuprite image are even more correlated, except for minor differences in EM #1 and EM #3 (Fig. 17 (a), (b), and (c) and (g), (h), and (i)) and a difference in EM #9 (Fig. 18 (g), (h), and lower-left corner of (i)).

VI. Conclusion And Future Work

In this study, we presented a novel methodology for a fully constrained spectral unmixing, the vectorized code projected gradient descent unmixing (VPGDU). The newly proposed scheme performs iterative search in EM fraction space, based on analytical projected gradient descent optimization with respect to a variant of the SAM similarity measure as an objective function. The detailed derivation determines also analytically the optimal step size in each iteration and employs a projection onto a simplex to fulfill the required constraints imposed on the fractions. The entire scheme was implemented using code vectorization, which is basically a special form of parallel computing. In particular, we showed how to "fold" the computational process (applied to an entire image) into a single loop of matrix operations. This results in a considerably more efficient performance of the fully constrained spectral unmixing proposed.

The methodology presented is capable of applying the unmixing process to a (relatively) large number of EMs, thereby taking advantage of the numerous amounts of available hyperspectral imagery.

A comprehensive assessment of the proposed scheme was done relatively to FCLSU, GBM, and the fast state-of-the-art SUnSAL and CLSUnSAL methods using simulated and real data, including the well-known Cuprite image, an AISA spectral image, and EM signatures extracted automatically/manually from these real images or selected from a spectral library. The experimental results indicate that the fractions obtained by the new methodology are in good agreement with those obtained by FCLSU, GBM, SUnSAL, and CLSUnSAL for all data sets tested, especially in the case of SNR = 30dB. Moreover, the unmixing performance, under varying illumination, is greatly enhanced due to the inherent advantage of the SAM-based objective function used by our VPGDU scheme.

The suggested framework can be easily adapted to other objective functions, especially if they are differentiable with respect to the fraction vector, so that the gradient and the step size can be analytically formulated. Otherwise, the gradient and step size should be calculated numerically, but the running time is likely to increase, of course. Also, the VPGDU framework could be easily modified to handle any linear constrains that might be imposed on the fractions. For example, we show in [58], [59] how to determine first a subset of EMs that are actually present in each pixel using novel spatial-spectral preprocessing; VPGDU is then applied with predefined fraction upper bounds of 0 to EMs non-present in a given pixel. In summary, the algorithmic framework is rather modular and its components are easy to understand and implement.

As part of future research, it would also be of interest to reduce further the run-time (beyond that obtained by code vectorization). This could be done, for example, by applying VPGDU simultaneously to a number of image sub-regions via parallel multi-core computing. In an attempt to improve the rate of convergence, it would also be of interest to apply, for example, the conjugate gradient optimization to the objective function presented and compare the accuracy and run-time obtained to those of VPGDU. Finally, one could investigate the influence of initialization on the unmixing performance.

Appendix

Convergence Of Projected Gradient Descent Unmixing (PGDU)

We provide here a convergence proof of our PGDU method, which draws on previous works on the convergence of projected gradient descent (PGD). The optimization in question is presented usually as a minimization problem. Thus, to keep this discussion consistent with the relevant literature regarding gradient descent optimization, we will minimize $-\phi(\mathbf{f})$, instead of maximizing $\phi(\mathbf{f})$ according to Eq. (16), as $\max_{\mathbf{f}}\{\phi(\mathbf{f})\} = \min_{\mathbf{f}}\{-\phi(\mathbf{f})\}$. That is, we will consider the minimization problem

$$\min_{\mathbf{f}}\{-\phi(\mathbf{f}) \,|\, \mathbf{f} \in \Omega^L, \tag{52}$$

where $-\phi(\mathbf{f})$ is a generalized convex function [60], which is continuously differentiable on the convex set $\Omega^L \subseteq \mathbb{R}^L$.

The point $\mathbf{f}^* \in \Omega^L$ is called a *stationary point*, i.e., an optimal solution of the problem in (52), if

$$\left\langle \nabla\left(-\phi\left(\mathbf{f}^*\right)\right), \left(\mathbf{f} - \mathbf{f}^*\right) \right\rangle \geq 0, \quad \forall \mathbf{f} \in \Omega^L, \tag{53}$$

where $\langle \cdot, \cdot \rangle$ denotes an inner product, and the gradient is

$$\nabla\left(-\phi\left(\mathbf{f}^*\right)\right) = -\nabla\phi\left(\mathbf{f}^*\right).$$

Recall that the iterative update of the projected gradient

descent to minimize $-\phi(\mathbf{f})$ is

$$\hat{\mathbf{f}}^{k+1} = \mathrm{P}_{\Omega}\left(\hat{\mathbf{f}}^{k} - \gamma_k \nabla\phi\left(\hat{\mathbf{f}}^{k}\right)\right). \tag{54}$$

Thus, we need to show that

$$\lim_{k\to\infty} \hat{\mathbf{f}}^{k} = \mathbf{f}^{*} \in \Omega^{L}. \tag{55}$$

Due to space limitation, we only provide a proof of convexity of the objective function $-\phi(\mathbf{f})$ and of the set Ω^{L}. We then refer to relevant works, which provide a complete convergence proof of PGD under similar conditions to those of our PGDU algorithm.

We first prove the concavity of $\phi(\mathbf{f})$, i.e., the convexity of $-\phi(\mathbf{f})$.

Definition 1:

A function $g(x)$ is *strictly quasi-concave*, if for all $x_1 \neq x_2$ and $0<\theta<1$, the following holds:

$$g\left((1-\theta)x_1 + \theta x_2\right) > \min\{g(x_1), g(x_2)\} \tag{56}$$

Proposition 1:

The function $\phi(\mathbf{f})$ is strictly quasi-concave, i.e., for all $\mathbf{f}_1 \neq \mathbf{f}_2$ and $0<\theta<1$, we have

$$\phi\left((1-\theta)\mathbf{f}_1 + \theta\mathbf{f}_2\right) > \min\{\phi(\mathbf{f}_1), \phi(\mathbf{f}_2)\}. \tag{57}$$

Proof:

The value of $\phi(\mathbf{f}_i)$ is the cosine of the angle between the vectors $\mathbf{m}(\lambda\times 1)$ and $\mathbf{Ef}_i\left((\lambda\times L)(L\times 1)\right)$ in $\mathbb{R}^{\lambda}_{\geq 0}$ Euclidean space (Fig. 19). Let $\hat{\mathbf{v}}$ denote the normalized vector $\mathbf{v}$, i.e., $\hat{\mathbf{v}} = \dfrac{\mathbf{v}}{\|\mathbf{v}\|}$. Thus,

$$\phi(\mathbf{f}_i) = \hat{\mathbf{m}}^{T}\widehat{\mathbf{Ef}}_i \tag{58}$$

where $\hat{\mathbf{m}} = \dfrac{\mathbf{m}}{\|\mathbf{m}\|}$ and $\widehat{\mathbf{Ef}}_i = \dfrac{\mathbf{Ef}_i}{\|\mathbf{Ef}_i\|}$ are the normalized vectors of $\mathbf{m}$ and $\mathbf{Ef}_i$, respectively. Accordingly, $\phi(\mathbf{f}_i)$ is the length of the arc (on a great circle), which connects these two vectors on the surface of the ℓ_2 unit sphere (in cosine units). Assume that $\phi(\mathbf{f}_1) = \hat{\mathbf{m}}^{T}\widehat{\mathbf{Ef}}_1 = \min\{\phi(\mathbf{f}_1), \phi(\mathbf{f}_2)\}$ and let $\mathbf{f}_\theta = (1-\theta)\mathbf{f}_1 + \theta\mathbf{f}_2$. We need to show the following strict inequality:

$$\phi(\mathbf{f}_\theta) = \hat{\mathbf{m}}^{T}\widehat{\mathbf{Ef}}_\theta > \hat{\mathbf{m}}^{T}\widehat{\mathbf{Ef}}_1 = \phi(\mathbf{f}_1). \tag{59}$$

Let $\pi_{1,2}$ (Fig. 19) denote the line segment joining $\mathbf{Ef}_1$ and $\mathbf{Ef}_2$. The point $\mathbf{Ef}_\theta$ must lie on $\pi_{1,2}$, and can thus be expressed as

$$\mathbf{Ef}_\theta = \mathbf{E}\left((1-\theta)\mathbf{f}_1 + \theta\mathbf{f}_2\right) = (1-\theta)\mathbf{Ef}_1 + \theta\mathbf{Ef}_2. \tag{60}$$

Analogously, if we let $\bar{\pi}_{1,2}$ (Fig. 19) denote the line segment joining $\widehat{\mathbf{Ef}}_1$ and $\widehat{\mathbf{Ef}}_2$, then the corresponding point of $\mathbf{Ef}_\theta$, denoted by $\overline{\mathbf{Ef}}_\theta$, must lie on $\bar{\pi}_{1,2}$, i.e., it can be expressed as:

$$\overline{\mathbf{Ef}}_\theta = (1-\delta\theta)\widehat{\mathbf{Ef}}_1 + \delta\theta\widehat{\mathbf{Ef}}_2 = \widehat{\mathbf{Ef}}_1 + \delta\theta\left(\widehat{\mathbf{Ef}}_2 - \widehat{\mathbf{Ef}}_1\right), \tag{61}$$

where the parameter $\delta > 0$ depends on the angle between the lines $\pi_{1,2}$ and $\bar{\pi}_{1,2}$. (According to the Intercept Theorem, if $\pi_{1,2} \parallel \bar{\pi}_{1,2}$, then $\delta = 1$.)

Definition 2:

Let $\omega(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|$ denote the Euclidean distance metric in $\mathbb{R}^{\lambda}_{\geq 0}$. Then the open unit ball $B_1(\mathbf{o})$ and the unit sphere $S_1(\mathbf{o})$ centered at the origin point $\mathbf{o}$ are defined, respectively, as

$$B_1(\mathbf{o}) = \left\{\mathbf{x}\in\mathbb{R}^{\lambda}_{\geq 0} \;\middle|\; \omega(\mathbf{x},\mathbf{o}) < 1\right\}$$

and

$$S_1(\mathbf{o}) = \left\{\mathbf{x}\in\mathbb{R}^{\lambda}_{\geq 0} \;\middle|\; \omega(\mathbf{x},\mathbf{o}) = 1\right\}.$$

In practice, $S_1(\mathbf{o})$ is the convex hull of $B_1(\mathbf{o})$; however, $S_1(\mathbf{o})$ itself is a non-convex set. Accordingly, $\left\{\widehat{\mathbf{Ef}}_1, \widehat{\mathbf{Ef}}_2\right\} \in S_1(\mathbf{o})$ but $\overline{\mathbf{Ef}}_\theta \notin S_1(\mathbf{o})$, and $\overline{\mathbf{Ef}}_\theta \in B_1(\mathbf{o})$ while $\widehat{\mathbf{Ef}}_\theta = \dfrac{\overline{\mathbf{Ef}}_\theta}{\left\|\overline{\mathbf{Ef}}_\theta\right\|} \in S_1(\mathbf{o})$. Therefore, $\left\|\overline{\mathbf{Ef}_\theta}\right\| \leq \left\|\widehat{\mathbf{Ef}_\theta}\right\| = 1$ and thus $\widehat{\mathbf{Ef}}_\theta = \eta\overline{\mathbf{Ef}}_\theta$, where $1<\eta$. Next, we write

$$\begin{aligned}
\phi(\mathbf{f}_\theta) &= \hat{\mathbf{m}}^{T}\widehat{\mathbf{Ef}}_\theta = \eta\hat{\mathbf{m}}^{T}\overline{\mathbf{Ef}}_\theta \\
&= \eta\hat{\mathbf{m}}^{T}\left((1-\delta\theta)\widehat{\mathbf{Ef}}_1 + \delta\theta\widehat{\mathbf{Ef}}_2\right) \\
&= \eta\hat{\mathbf{m}}^{T}\left(\widehat{\mathbf{Ef}}_1 + \delta\theta\left(\widehat{\mathbf{Ef}}_2 - \widehat{\mathbf{Ef}}_1\right)\right) \\
&= \eta\hat{\mathbf{m}}^{T}\widehat{\mathbf{Ef}}_1 + \eta\delta\theta\left(\hat{\mathbf{m}}^{T}\widehat{\mathbf{Ef}}_2 - \hat{\mathbf{m}}^{T}\widehat{\mathbf{Ef}}_1\right)
\end{aligned} \tag{62}$$

Assigning $\phi(\mathbf{f}_i) = \hat{\mathbf{m}}^{T}\widehat{\mathbf{Ef}}_i$, we have

$$\phi(\mathbf{f}_\theta) = \eta\phi(\mathbf{f}_1) + \eta\delta\theta\left(\phi(\mathbf{f}_2) - \phi(\mathbf{f}_1)\right). \tag{63}$$

Recall that $\phi(\mathbf{f}_1) < \phi(\mathbf{f}_2)$ and that η, δ, and θ are all positive. Thus, $0 < \eta\delta\theta\left(\phi(\mathbf{f}_2) - \phi(\mathbf{f}_1)\right)$ and

$$\phi(\mathbf{f}_\theta) > \eta\phi(\mathbf{f}_1). \tag{64}$$

Finally, since $1<\eta$ we have

$$\phi(\mathbf{f}_\theta) = \eta\phi(\mathbf{f}_1) > \phi(\mathbf{f}_1) = \min\{\phi(\mathbf{f}_1), \phi(\mathbf{f}_2)\}, \tag{65}$$

i.e., $\phi(\mathbf{f}_i)$ is strictly quasi-concave and $-\phi(\mathbf{f})$ is strictly quasi-convex. QED.

We now prove that the set Ω^L is convex.

Proposition 2: Given

$$\{\mathbf{f}_1, \mathbf{f}_2\} \in \Omega^L : 0 \le \mathbf{f}_i^j \le 1 \text{ and } \sum_{j=1}^{L} \mathbf{f}_i^j = 1,\ i = 1,2,$$

The following holds:

$$\mathbf{f}_\theta \in \Omega^L : \mathbf{f}_\theta = (1-\theta)\mathbf{f}_1 + \theta\mathbf{f}_2 \text{ with } 0 \le \theta \le 1.$$

***Proof*:**

We may assume, without loss of generality, that $\mathbf{f}_1^j = \min\{\mathbf{f}_1^j, \mathbf{f}_2^j\}$. Thus, we can write $\mathbf{f}_2^j = \mathbf{f}_1^j + \delta$ and $\mathbf{f}_\theta^j = (1-\theta)\mathbf{f}_1^j + \theta(\mathbf{f}_1^j + \delta) = \mathbf{f}_1^j + \theta\delta$ with $0 \le \delta$. Recall that $\mathbf{f}_1 \in \Omega^L$, and accordingly, $0 \le \mathbf{f}_1^j$. Since $0 \le \theta\delta$, it must hold that $0 \le \mathbf{f}_1^j \le \mathbf{f}_\theta^j = \mathbf{f}_1^j + \theta\delta$. Analogously, we can easily show that $\mathbf{f}_1^j \le 1$, i.e., $0 \le \mathbf{f}_\theta^j \le 1$.

To prove the sum-to-one property of $\mathbf{f}_\theta$ we start by writing

$$\sum_{i=1}^{L} \mathbf{f}_\theta^i = \sum_{i=1}^{L} \left[(1-\theta)\mathbf{f}_1^i + \theta\mathbf{f}_2^i \right]. \tag{66}$$

$$= \sum_{i=1}^{L} (1-\theta)\mathbf{f}_1^i + \sum_{i=1}^{L} \theta\mathbf{f}_2^i = (1-\theta)\sum_{i=1}^{L} \mathbf{f}_1^i + \theta\sum_{i=1}^{L} \mathbf{f}_2^i$$

Recalling that $\{\mathbf{f}_1, \mathbf{f}_2\} \in \Omega^L$, i.e., $\sum_{i=1}^{L} \mathbf{f}_1^i = 1$ and $\sum_{i=1}^{L} \mathbf{f}_2^i = 1$, we may write

$$\sum_{i=1}^{L} \mathbf{f}_\theta^i = (1-\theta)\sum_{i=1}^{L} \mathbf{f}_1^i + \theta\sum_{i=1}^{L} \mathbf{f}_2^i = (1-\theta)1 + \theta 1 = 1. \tag{67}$$

We showed that $0 \le \mathbf{f}_\theta^j \le 1$ and $\sum_{i=1}^{L} \mathbf{f}_\theta^i = 1$, and accordingly, $\mathbf{f}_\theta \in \Omega^L : \mathbf{f}_\theta = (1-\theta)\mathbf{f}_1 + \theta\mathbf{f}_2$, i.e., the set Ω^L is convex. QED.

The convergence of gradient descent for unconstrained minimization of a quasi-convex function was addressed in [61]. The fact that $-\phi(\mathbf{f})$ is strictly quasi-convex ensures that every local minimum of the function is also a global minimum [62]. Accordingly, in view of the nature of gradient descent optimization, which is based on first order differentiations only, the difference between a convex and a strictly quasi-convex objective function is negligible.

The convergence of PGD has been addressed in several works, considering both the type of the objective function and the choice of a step size. As can be expected, in general, the convergence is highly affected by the step size. Specifically, to guarantee convergence of the entire process, the chosen step size must satisfy a sufficient reduction at each iteration of the objective function in question. The analytical derivation in our case of an optimal step size (i.e., a step size that guarantees a maximal reduction in the objective function of Eq. (52)), bodes well with the above premise. A complete convergence proof of PGD for a convex objective function and a step size determined by the Armijo rule is provided in [62]. A convergence analysis of PGD for a generalized convex function (i.e., a quasi-convex/pseudo-convex function) is given in [63]—[65].

Finally, a comprehensive discussion of PGD with an exact line search (as in the PGDU case) and a detailed proof of convergence are given in [64], [65].

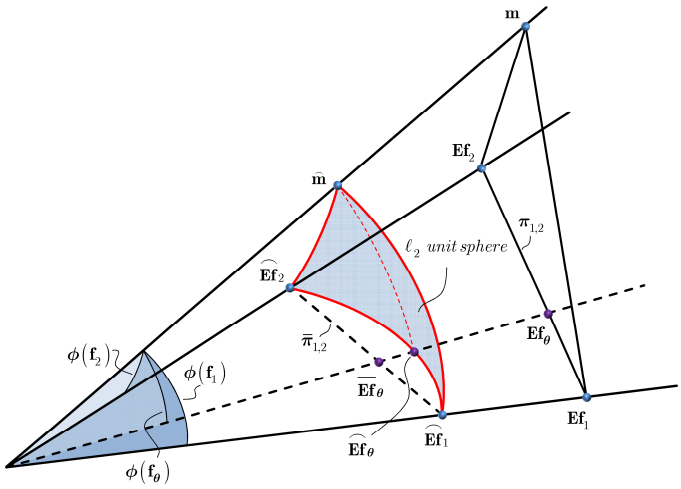

Fig. 19. Geometric interpretation of objective function and relations between different associated vectors.

To gain more insight to the convergence of PGDU, we created 1000 simulated mixtures with the same true simulated fractions using 20 EMs. An additive white noise was added separately to each of the mixtures. The estimation error of the estimated fraction vector, $\hat{\mathbf{f}}$, with respect to the true fraction vector, $\mathbf{f}^*$, defined as $\left\|\hat{\mathbf{f}} - \mathbf{f}^*\right\|$, was computed for each iteration of the process. Fig. 20 depicts the estimation error and the objective function value as a function of the iteration number for the best, worst, and median cases (i.e., simulated pixels with the smallest, largest, and median error, respectively).

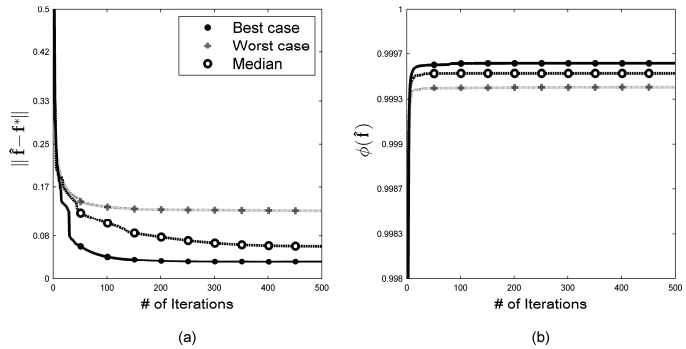

Fig. 20. Illustration of PGDU convergence: (a) Estimation error and (b) objective function vs. iteration number (of best, worst, and median cases), using 1000 simulated mixtures of 20 EMs.

Both the theoretical analysis and the empirical evidence provide a good indication as to the convergence of PGDU.

ACKNOWLEDGMENT

The authors would like to thank the anonymous associate editor and reviewers for their helpful comments and insightful suggestions.

REFERENCES

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

[2] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 44–57, Jan. 2002.

[3] A. Plaza, Q. Du, J. M. Bioucas-Dias, X. Jia, and F. A. Kruse, "Foreword to the special issue on spectral unmixing of remotely sensed data," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 1, pp. 4103–4110, Nov. 2011.

[4] M. Shoshany and T. Svoray, "Multidate adaptive unmixing and its application to analysis of ecosystem transitions along a climatic gradient," *Remote Sens. Environ.*, vol. 82, no. 1, pp. 5–20, Sept. 2002.

[5] C. A. Bateson and B. Curtiss, "A method for manual endmember selection and spectral unmixing," *Remote Sens. Environ.*, vol. 55, no. 3, pp. 229–243, Mar. 1996.

[6] M. Parente and A. Plaza, "Survey of geometric and statistical unmixing algorithms for hyperspectral images," in *2nd IEEE Workshop on Hyperspectral Image and Signal Process.: Evolution in Remote Sens.*, 2010, Reykjavìk, Iceland.

[7] M. E. Winter, "N-FINDR: an algorithm for fast autonomous spectral end-member determination in hyperspectral data," in *Proc. SPIE Int. Symp. Optical Science, Engineering, Instrumentation, Conf. on Imaging Spectorometry V,* vol. 3753, Denver, CO, 1999, pp. 266–275.

[8] C. I. Chang, C. C. Wu, W. M. Liu, and Y. C. Ouyang, "A new growing method for simplex-based endmember extraction algorithm," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 10, pp. 2804–2819, Oct. 2006.

[9] E. M. T. Hendrix, I. Garcia, J. Plaza, G. Martin, and A. Plaza, "A new minimum-volume enclosing algorithm for endmember identification and abundance estimation in hyperspectral data," *IEEE Trans. on Geosci. and Remote Sens.*, vol. 50, no. 7. pp. 2744–2757, July 2012.

[10] J. M. P. Nascimento and J. M. Bioucas-Dias, "Hyperspectral unmixing based on mixtures of Dirichlet components," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 863–878, Mar. 2012.

[11] J. M. Bioucas-Dias and M. A. T. Figueiredo, "Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing," in *2nd IEEE Workshop on Hyperspectral Image and Signal Process.: Evolution in Remote Sensing,*2010, Reykjavìk, Iceland.

[12] M. D. Iordache, J. M. Bioucas-Dias, and A. Plaza, "Collaborative sparse regression for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 341–354, Jan. 2014.

[13] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, Apr. 1995.

[14] A. Plaza, P. Martínez, R. Pérez, and J. Plaza, "A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 3, pp. 650–663, Mar. 2004.

[15] S. Sánchez, G. Martín, and A. Plaza, "Parallel implementation of the N-FINDR endmember extraction algorithm on commodity graphics processing units," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. Honolulu, HI*, 2010, pp. 955–958.

[16] C. González, J. Resano, D. Mozos, A. Plaza, and D. Valencia, "FPGA implementation of the pixel purity index algorithm for remotely sensed hyperspectral image analysis," *EURASIP J. Adv. Signal Process.*, June 2010.

[17] Chein-I Chang, Chao-Cheng Wu, Chien-Shun Lo, and Mann-Li Chang, "Real-time simplex growing algorithms for hyperspectral endmember extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 4, pp. 1834–1850, Apr. 2010.

[18] Y. E. Shimabukuro and J. A. Smith, "The least-squares mixing models to generate fraction images derived from remote sensing multispectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 29, no. 1, pp. 16–20, Jan. 1991.

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

[20] J. W. Boardman, "Inversion of high spectral resolution data," in *Proc. SPIE Imaging Spectroscopy of the Terrestrial Environment*, vol. 1298, Orlando, FL, 1990, pp. 222–233.

[21] Y. E. Shimabukuro, "Shade images derived from linear mixing models of multispectral measurements of forested areas," Ph.D. dissertation, Dept. Forest Wood Sci., Colorado St. Univ., Fort Collins, CO, 1987.

[22] E. A. Ashton and A. Schaum, "Algorithms for the detection of sub-pixel targets in multispectral imagery," *Photogramm. Eng. Remote Sens.*, vol. 64, no. 7, pp. 723–731, July 1998.

[23] C. Theys, N. Dobigeon, J. Y. Tourneret, and H. Lantéri, "Linear unmixing of hyperspectral images using a scaled gradient method," in *Proc. IEEE Workshop on Stat. Signal Process.*, Cardiff, United Kingdom, 2009, pp. 729–732.

[24] J. Chen, C. Richard, H. Lanteri, C. Theys, and P. Honeine, "A gradient based method for fully constrained least-squares unmixing of hyperspectral images," in *Proc. IEEE Workshop on Stat. Signal Process.*, Nice, France, 2011, pp. 301–304.

[25] A. Halimi, Y. Altmann, N. Dobigeon, and J. Y. Tourneret, "Unmixing hyperspectral images using the generalized bilinear model," in *Proc. Int. Geosci. and Remote Sens. Symp.*, Colorado Springs, CO, 2011, pp. 1886–1889.

[26] A. Halimi, Y. Altmann, N. Dobigeon, and J.-Y. Tourneret, "Nonlinear unmixing of hyperspectral images using a generalized bilinear model," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4153–4162, Nov. 2011.

[27] Y. Altmann, A. Halimi, N. Dobigeon, and J. Y. Tourneret, "Supervised nonlinear spectral unmixing using a postnonlinear mixing model for hyperspectral imagery," *IEEE Trans. Image Process.*, vol. 21, no. 6, pp. 3017–3025, June 2012.

[28] 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, July 2009.

[29] N. Keshava, "A Survey of spectral unmixing algorithms," *Lincoln Lab. J.*, vol. 14, no. 1, pp. 55–78, 2003.

[30] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2. pp. 354–379, Apr. 2012.

[31] H. K. Aggarwal and A. Majumdar, "Hyperspectral unmixing in the presence of mixed noise using joint-sparsity and total variation,"

IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 9, no. 9, pp. 4257–4266, Sep. 2016.

[32] W. He, H. Zhang, and L. Zhang, "Sparsity-regularized robust non-negative matrix factorization for hyperspectral unmixing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4267–4279, Sep. 2016.

[33] C. Li, Y. Ma, X. Mei, C. Liu, and J. Ma, "Hyperspectral unmixing with robust collaborative sparse regression," *Remote Sens.*, vol. 8, no. 7, p. 588, July 2016.

[34] Y. Zhou, A. Rangarajan, and P. D. Gader, "A spatial compositional model for linear unmixing and endmember uncertainty estimation," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5987–6002, Dec. 2016.

[35] T. Uezato, R. J. Murphy, A. Melkumyan, and A. Chlingaryan, "Incorporating spatial information and endmember variability into unmixing analyses to improve abundance estimates," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5563–5575, Dec. 2016.

[36] R. Feng, Y. Zhong, and L. Zhang, "Adaptive spatial regularization sparse unmixing strategy based on joint MAP for hyperspectral remote sensing imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 12, pp. 5791–5805, Dec. 2016.

[37] J. Sigurdsson, M. O. Ulfarsson, and J. R. Sveinsson, "Blind hyperspectral unmixing using total variation and l_q sparse regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 11, pp. 6371–6384, Nov. 2016.

[38] R. Heylen, M. Parente, and P. Gader, "A review of nonlinear hyperspectral unmixing methods," *IEEE J. Sel. Topics in Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 1844–1868, June 2014.

[39] N. Dobigeon, J. Y. Tourneret, C. Richard, J. C. M. Bermudez, S. McLaughlin, and A. O. Hero, "Nonlinear unmixing of hyperspectral images: Models and algorithms," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 82–94, Jan. 2014.

[40] F. Tsai and W. Philpot, "Derivative analysis of hyperspectral data," *Remote Sens. Environ.*, vol. 66, no. 1, pp. 41–51, Oct. 1998.

[41] Z. Ben Rabah, I. R. Farah, G. Mercier, and B. Solaiman, "A new method to change illumination effect reduction based on spectral angle constraint for hyperspectral image unmixing," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 6, pp. 1110–1114, Nov. 2011.

[42] A. A. Goldstein, "Convex programming in Hilbert space," *Bull. Amer. Math. Soc.*, vol. 70, no. 5, pp. 709–710, Sept. 1964.

[43] E. S. Levitin and B. T. Polyak, "Constrained minimization methods," *USSR Comput. Math. Math. Phys.*, vol. 6, no. 5, pp. 1–50, 1966.

[44] S. Becker and M. J. Fadili, "A quasi-Newton proximal splitting method," in *Proc. ACM 25th Int. Conf. Neural Inform. Process. Syst.*, Lake Tahoe, NV., 2012, pp. 2618–2626.

[45] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, ch. 10, H. H. Bauschke, R. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, Eds. New York: Springer, 2011, pp. 185–212.

[46] D. P. Bertsekas, "Incremental gradient, subgradient, and proximal methods for convex optimization: A survey," Cornell Univ. Lib., Ithaca, NY, arXiv:1507.01030, July 2015.

[47] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, United Kingdom: Cambridge University Press, 2004.

[48] J. Nocedal and S. J. Wright. (2006). *Numerical Optimization* [Online]. Available: http://www.springer.com/gp/book/9780387303031

[49] Y. Chen and X. Ye, "Projection onto a simplex," Cornell Univ. Lib., Ithaca, NY, arXiv1101.6081v2, Feb. 2011.

[50] S. Boyd, N. Parikh, E. Chu, and B. Peleato, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends® Mach. Learn.*, vol. 3, no. 1, p. 1–122, Nov. 2010.

[51] M. Shoshany, F. Kizel, N. S. Netanyahu, N. Goldshlager, T. Jarmer, and G. Even-Tzur, "An iterative search in end-member fraction space for spectral unmixing," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 4, pp. 706–709, July 2011.

[52] R. Heylen, D. Burazerović, and P. Scheunders, "Fully constrained least squares spectral unmixing by simplex projection," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4112–4122, Nov. 2011.

[53] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l_1 -ball for learning in high dimensions," *Proc. 25th ACM Int. Conf. Mach. Learn.* - Helsinki, Finland, 2008, pp. 272–279.

[54] G. A. Shaw and H. K. Burke, "Spectral imaging for remote sensing," *Lincoln Lab. J.*, vol. 14, no. 1, pp. 3–28, 2003.

[55] J. M. P. Nascimento and J. M. B. Dias, "Vertex component analysis: A fast algorithm to unmix hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 898–910, Apr. 2005.

[56] F. J. García-Haro, S. Sommer, and T. Kemper, "A new tool for variable multiple endmember spectral mixture analysis (VMESMA)," *Int. J. Remote Sens.*, vol. 26, no. 10, pp. 2135–2162, May 2005.

[57] D. M. Rogge, B. Rivard, Z. Jinkai, and F. Jilu, "Iterative spectral unmixing for optimizing per-pixel endmember sets," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 12, pp. 3725–3735, Dec. 2006.

[58] F. Kizel, "Novel methods for stepwise analytical and spatially adaptive hyperspectral unmixing," Ph.D. thesis, Dept. Geoinform. Map. Eng., Technion, Haifa, Israel, 2014.

[59] F. Kizel, M. Shoshany, and N. S. Netanyahu, "Spatially adaptive hyperspectral unmixing based on sums of 2D Gaussians for modelling endmember fraction surfaces," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.,* Milano, Italy, 2015, pp. 4440–4443.

[60] S. K. Mishra, S.-Y. Wang, and K. K. Lai, *Generalized Convexity and Vector Optimization*, Berlin, Heidelberg: Springer-Verlag, 2009, pp. 7–24.

[61] K. C. Kiwiel and K. Murty, "Convergence of the steepest descent method for minimizing quasiconvex functions," *J. Opt. Theory Appl.*, vol. 89, no. 1, pp. 221–226, Apr. 1996.

[62] I. Barrodale, "Best rational approximation and strict quasi-convexity," *SIAM J. Numer. Anal.*, vol. 10, no. 1, pp. 8–12, Mar. 1973.

[63] C. Wang and N. Xiu, "Convergence of the gradient projection method for generalized convex minimization," *Comput. Opt. Appl.*, vol. 16, no. 2, pp. 111–120, July 2000.

[64] W. W. Hager and S. Park, "The gradient projection method with exact line search," *J. Glob. Opt.*, vol. 30, no. 1, pp. 103–118, Sep. 2004.

[65] N. Xiu, C. Wang, and L. Kong, "A note on the gradient projection method with exact stepsize rule," *J. Comput. Math.*, vol. 25, no. 2, pp. 221–230, Mar. 2007.