Unsupervised Classification in Hyperspectral Imagery With Nonlocal Total Variation and Primal-Dual Hybrid Gradient Algorithm

Wei Zhu, Victoria Chayes, Alexandre Tiard, Stephanie Sanchez, Devin Dahlberg, Andrea L. Bertozzi, Stanley Osher, Dominique Zosso, *Member, IEEE*, and Da Kuang

***Abstract*— In this paper, a graph-based nonlocal total variation method is proposed for unsupervised classification of hyperspectral images (HSI). The variational problem is solved by the primal-dual hybrid gradient algorithm. By squaring the labeling function and using a stable simplex clustering routine, an unsupervised clustering method with random initialization can be implemented. The effectiveness of this proposed algorithm is illustrated on both synthetic and real-world HSI, and numerical results show that the proposed algorithm outperforms other standard unsupervised clustering methods, such as spherical K-means, nonnegative matrix factorization, and the graph-based Merriman–Bence–Osher scheme.**

***Index Terms*— Hyperspectral images (HSI), nonlocal total variation (NLTV), primal-dual hybrid gradient (PDHG) algorithm, stable simplex clustering, unsupervised classification.**

I. Introduction

HYPERSPECTRAL image (HSI) is an important domain in the field of remote sensing with numerous applications in agriculture, environmental science, mineralogy, and surveillance [1]. Hyperspectral sensors capture information of intensity of reflection at different wavelengths, from the infrared to ultraviolet. They take measurements 10–30 nm apart, and up to 200 layers for a single image. Each pixel has a unique spectral signature, which can be used to differentiate objects that cannot be distinguished based on visible spectra, for example: invisible gas plumes, oil or chemical spills over water, or healthy from unhealthy crops.

The majority of HSI classification methods are either *unmixing* methods or *clustering* methods. Unmixing methods extract the information of the constitutive materials (the *endmembers*) and the abundance map [2]–[5]. Clustering methods do not extract endmembers; instead, they return the spectral signatures of the centroids of the clusters. Each centroid is the mean of the signatures of all the pixels in a cluster. However, when it is assumed that most of the pixels are dominated mostly by one endmember, i.e., in the absence of partial volume effects [6], which is usually the case for high-resolution HSI, these two types of methods are expected to give similar results [5]. The proposed nonlocal total variation (NLTV) method for HSI classification in this paper is a clustering method.

Much work has been carried out in the literature in both the unmixing and the clustering categories. HSI unmixing models can be characterized as linear or nonlinear. In a linear unmixing model (LUM), each pixel is approximated by a linear combination of the endmembers. When the linear coefficients are constrained to be nonnegative, it is equivalent to nonnegative matrix factorization (NMF), and good unsupervised classification results have been achieved in [3]–[5] using either NMF or hierarchical rank-2 NMF (H2NMF). Despite the simplicity of LUM, the assumption of a linear mixture of materials has been shown to be physically inaccurate in certain situations [7]. Researchers are starting to expand aggressively into the much more complicated nonlinear unmixing realm [8], where nonlinear effects, such as atmospheric scattering, are explicitly modeled. However, most of the work that has been done for nonlinear unmixing so far is supervised in the sense that prior knowledge of the endmember signatures is required [2]. Discriminative machine learning methods, such as support vector machine-based [9]–[11] and relevance vector machine-based [12]–[14] approaches, have also been applied to hyperspectral images, but they are also supervised methods, since a training set is needed to learn the classifiers.

On the contrary, graph-based clustering methods implicitly model the nonlinear mixture of the endmembers. This type of method is built upon a weight matrix that encodes the similarity between the pixels, which is typically a sparse matrix constructed using the distances between the spectral signatures. Graph-cut problems for graph segmentation have been well studied in the literature [15]–[18]. Bertozzi and Flenner [19] proposed a diffuse interface

Manuscript received April 28, 2016; revised August 18, 2016 and December 11, 2016; accepted December 13, 2016. Date of publication February 13, 2017; date of current version March 17, 2017. This work was supported in part by the National Science Foundation under Grant DMS-1118971, Grant DMS-1045536, and Grant DMS-1417674, and in part by the Office of Naval Research under Grant N00014-16-1-2119.

W. Zhu, A. L. Bertozzi, S. Osher, and D. Kuang are with the Department of Mathematics, University of California at Los Angeles, Los Angeles, CA 90095 USA (e-mail: weizhu731@math.ucla.edu; bertozzi@math.ucla.edu; sjo@math.ucla.edu; dakuang@math.ucla.edu;).

V. Chayes is with Bard College, Annandale-On-Hudson, NY 12504 USA (e-mail: vminervachayes@gmail.com).

A. Tiard is with UCLA Vision Lab, School of Engineering, University of California at Los Angeles, Los Angeles, CA 90095 USA (e-mail: alexandretiard@gmail.com).

S. Sanchez is with the Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: ssanche2@stanford.edu).

D. Dahlberg is with the University of California at San Diego, La Jolla, CA 92093 USA (e-mail: dahlbergdevin@gmail.com).

D. Zosso is with the Department of Mathematical Sciences, Montana State University, Bozeman, MT 59717 USA (e-mail: dominique.zosso@montana.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2017.2654486

model on graphs with applications to classification of high-dimensional data. This idea has been combined with the Merriman–Bence–Osher (MBO) scheme [20] and applied to multiclass graph segmentation [21], [22] and HSI classification [23], [24]. The method in [19] minimizes a graph version of the Ginzburg–Landau (GL) functional, which consists of the Dirichlet energy of the labeling function and a double-well potential, and uses Nyström extension to speed up the calculation of the eigenvectors for inverting the graph Laplacian. This graph-based method performed well compared with other algorithms in the detection of chemical plumes in hyperspectral video sequences [23], [24]. However, the GL functional is nonconvex due to its double-well term, which may cause the algorithm to get stuck in local minima. This issue can be circumvented by running the algorithm multiple times with different initial conditions and hand picking the best result.

The two methods proposed in this paper are unsupervised graph-based clustering techniques. Instead of minimizing the GL functional, which has been proved to converge to the TV seminorm, this paper proposes to minimize the NLTV seminorm of the labeling functions $\|\nabla_w u_l\|_{L^1}$ directly. A detailed explanation of the nonlocal operator ∇_w and the labeling function u_l will be provided in Sections II and III. The L^1 regularized convex optimization problem is solved by the primal-dual hybrid gradient (PDHG) algorithm, which avoids the need to invert the graph Laplacian. We also introduce the novel idea of the quadratic model and a stable simplex clustering technique, which ensures that anomalies converge to their own clusters and makes random endmember initialization possible in the proposed algorithm. The direct usage of the NLTV seminorm makes the proposed clustering methods more accurate than other methods when evaluated quantitatively on HSI with ground-truth labels, and the quadratic model with stable simplex clustering is a completely new addition to the field of HSI classification.

This paper is organized as follows. In Section II, background is provided on TV and nonlocal operators. Two NLTV models (linear and quadratic) and a stable simplex clustering method are presented in Section III. Section IV provides a detailed explanation on the application of the PDHG algorithm to solving the convex optimization problems in the linear and quadratic models. Section V presents the numerical results and a sensitivity analysis on the key model parameters. Section VI presents the conclusions.

II. Total Variation and Nonlocal Operators

TV method was introduced in [25] and has been applied to various image processing tasks [26]. Its advantage is that one can preserve the edges in the image when minimizing $\|\nabla u\|_{L^1}$ (TV seminorm). The TV model is

$$\min_u E(u) = \|\nabla u\|_{L^1} + \lambda S(u).$$

The parameter λ can be adjusted to give higher priority to the TV-regularizing term, or the data fidelity term $S(u)$.

Despite its huge success in image processing, the TV method is still a local method. More specifically, the gradient of a pixel is calculated using its immediate adjacent pixels. It is known that local image processing techniques fail to produce satisfactory results when the image has repetitive structures, or intrinsically related objects in the image are not spatially connected. To address this problem, Buades *et al.* [27] proposed a nonlocal means method based on patch distances for image denoising. Gilboa and Osher [28] later formalized a systematic framework for nonlocal image processing. Nonlocal image processing produces much better results, because theoretically any pixel in the image can interact with any other, which better preserves texture and fine details.

In HSI classification, clusters can have elements that are not spatially connected. Thus, it is necessary to develop a nonlocal method of gradient calculation. We provide a review of nonlocal operators in the rest of this section. Note that the model is continuous, and the weights are not necessarily symmetric [29].

Let Ω be a region in $\mathbb{R}^n$, and $u : \Omega \to \mathbb{R}$ be a real function. In the model for HSI classification, Ω is the domain of the pixels, and $u : \Omega \to [0, 1]$ is the labeling function of a cluster. The larger the value of $u(x)$, the more likely that pixel x would be classified in that cluster. The nonlocal derivative is

$$\frac{\partial u}{\partial y}(x) := \frac{u(y) - u(x)}{d(x, y)}, \quad \text{for all } x, y \in \Omega$$

where d is a positive distance between x and y. In the context of hyperspectral images, $d(x, y)$ provides a way to measure the similarity between pixels x and y. Smaller $d(x, y)$ implies more resemblance between these two pixels. The nonlocal weight is defined as $w(x, y) = d^{-2}(x, y)$.

The nonlocal gradient $\nabla_w u$ for $u \in L^2(\Omega)$ can be defined as the collection of all partial derivatives, which is a function from Ω to $L^2(\Omega)$, i.e., $\nabla_w u \in L^2(\Omega, L^2(\Omega))$

$$\nabla_w u(x)(y) = \frac{\partial u}{\partial y}(x) = \sqrt{w(x, y)}(u(y) - u(x)).$$

The standard L^2 inner products on Hilbert spaces $L^2(\Omega)$ and $L^2(\Omega, L^2(\Omega))$ are used in the definition. More specifically, for $u_1, u_2 \in L^2(\Omega)$ and $v_1, v_2 \in L^2(\Omega, L^2(\Omega))$

$$\langle u_1, u_2 \rangle := \int_\Omega u_1(x) u_2(x) dx$$
$$\langle v_1, v_2 \rangle := \int_\Omega \int_\Omega v_1(x)(y) v_2(x)(y) dy dx.$$

The nonlocal divergence div_w is defined as the negative adjoint of the nonlocal gradient

$$\mathrm{div}_w v(x) := \int_\Omega \sqrt{w(x, y)} v(x)(y) - \sqrt{w(y, x)} v(y)(x) dy.$$

At last, a standard L^1 and L^∞ norm is defined on the space $L^2(\Omega, L^2(\Omega))$

$$\|v\|_{L^1} := \int_\Omega \|v(x)\|_{L^2} dx = \int_\Omega \left| \int_\Omega \left| v(x)(y) \right|^2 dy \right|^{\frac{1}{2}} dx$$
$$\|v\|_{L^\infty} := \sup_x \|v(x)\|_{L^2}.$$

III. Two NLTV Models for Unsupervised HSI Classification

In this section, two NLTV models are explained for unsupervised classification of HSI. The linear model runs faster

in each iteration, but it requires a more accurate centroid initialization. The quadratic model runs slower in each iteration, but it is more robust with respect to the centroid initialization. Moreover, the quadratic model converges faster if the initialization is not ideal.

A. Linear Model

We extend the idea from [30] to formulate a linear model for classification on HSI. The linear model seeks to minimize

$$E_1(u) = \|\nabla_w u\|_{L^1} + \langle u, f\rangle = \sum_{l=1}^{k} \|\nabla_w u_l\|_{L^1} + \sum_{l=1}^{k} \int u_l(x) f_l(x) dx \quad (1)$$

where $u = (u_1, u_2, \ldots, u_k) : \Omega \to \mathbb{K}^k$ is the labeling function, k is the number of clusters, $\mathbb{K}^k = \{(x_1, x_2, \ldots, x_k) | \sum_{i=1}^{k} x_i = 1, x_i \geq 0\}$ is the unit simplex in $\mathbb{R}^k$, and $\nabla_w u = (\nabla_w u_1, \ldots, \nabla_w u_k)$, such that $\|\nabla_w u\|_{L^1} = \sum_{l=1}^{k} \|\nabla_w u_l\|_{L^1}$. $f_l(x)$ is the error function defined as $f_l(x) = \frac{\lambda}{2}|g(x) - c_l|_\mu^2$, where $g(x)$ and c_l are the spectral signatures of pixel x and the lth centroid, which is initially either picked randomly from the HSI or generated by any fast unsupervised centroid extraction algorithm (e.g., H2NMF and K-means). The distance in the definition of $f_l(x)$ is a linear combination of cosine distance and Euclidean distance

$$|g(x) - c_l|_\mu = 1 - \frac{\langle g(x), c_l\rangle}{\|g(x)\|_2 \|c_l\|_2} + \mu \|g(x) - c_l\|_2, \quad \mu \geq 0.$$

In HSI processing, the cosine distance is generally used, because it is more robust to atmospheric interference and topographical features [31]. The reason why the Euclidean distance is also used is that sometimes different classes have very similar spectral angles, but vastly different spectral amplitudes (e.g., "dirt" and "road" in the Urban data set, which is illustrated in Section V). This is called the linear model, since the power of the labeling function u_l in (1) is one.

The intuition of the model is as follows. In order to minimize the fidelity term $\sum_{l=1}^{k} \int u_l(x) f_l(x)$, a small $u_l(x)$ is required if $f_l(x)$ is large, while no such requirement is needed if $f_l(x)$ is relatively small. This combined with the fact that $(u_1(x), \ldots, u_l(x))$ lies on a unit simplex implies that $u_l(x)$ would be the largest term if pixel x is mostly similar to the lth centroid c_l. Meanwhile, the NLTV-regularizing term $\sum_{l=1}^{k} \|\nabla_w u_l\|_{L^1}$ ensures that pixels similar to each other tend to have analogous values of u. Therefore, a classification of pixel x can be obtained by choosing the index l that has the largest value $u_l(x)$.

Now, we discuss how to discretize (1) for numerical implementation.

1) Weight Matrix: Following the idea from [28], the patch distance is defined as:

$$d_\sigma(x, y) = \int_\Omega G_\sigma(t) |g(x + t) - g(y + t)|^2 dt$$

where G_σ is a Gaussian of standard deviation σ. To build a sparse weight matrix, we take a patch P_i around every pixel i, and truncate the weight matrix by constructing a k-d tree [32] and searching the m nearest neighbors of P_i. k-d tree is a space-partitioning data structure that can significantly reduce the time cost of nearest neighbor search [33]. We employ a randomized and approximate version of this algorithm [34] implemented in the open source VLFeat package.[1] The weight is binarized by setting all nonzero entries to one. In the experiments, patches of size 3×3 are used, and m is set to 10. Note that unlike RGB image processing, the patch size for HSI does not have to be very large. The reason is that while low-dimensional RGB images require spatial context to identify pixels, high-dimensional hyperspectral images already encode enough information for each pixel in the spectral dimension. Of course, a larger patch size that is consistent with the spatial resolution of the HSI will still be preferable when significant noise is present.

2) Labeling Function and the Nonlocal Operators: The labeling function, $u = (u_1, u_2, \ldots, u_k)$, is discretized as a matrix of size $r \times k$, where r is the number of pixels in the hyperspectral image, and $(u_l)_j$ is the lth labeling function at jth pixel; $(\nabla_w u_l)_{i,j} = (w_{i,j})^{1/2}((u_l)_j - (u_l)_i)$ is the nonlocal gradient of u_l; $(\mathrm{div}_w v)_i = \sum_j (w_{i,j})^{1/2} v_{i,j} - (w_{j,i})^{1/2} v_{j,i}$ is the divergence of v at the ith pixel; and the discrete L^1 and L^∞ norm of $\nabla_w u_l$ are defined as: $\|\nabla_w u_l\|_{L^1} = \sum_i (\sum_j (\nabla_w u_l)_{i,j}^2)^{(1/2)}$, and $\|\nabla_w u_l\|_{L^\infty} = \max_i (\sum_j (\nabla_w u_l)_{i,j}^2)^{(1/2)}$.

The next issue to address is how to minimize (1) efficiently. The convexity of the energy functional E_1 allows us to consider using convex optimization methods. The first-order primal-dual algorithms have been successfully used in image processing with L^1 type regularizers [30], [35]–[37]. We use the PDHG algorithm. The main advantage is that no matrix inversion is involved in the iterations, as opposed to general graph Laplacian methods. The most expensive part of the computation comes from sparse matrix multiplications, which are still inexpensive due to the fact that only $m = 10$ nonzero elements are kept in each row of the nonlocal weight matrix.

We then address centroid updates and stopping criteria for the linear model. The concept of centroid updates is not uncommon; in fact, the standard K-means algorithm consists of two steps: first, it assigns each point to a cluster whose mean yields the least within-cluster sum of squares, and then, it recalculates the means from the centroids, and terminates when assignments no longer change [38]. Especially for data-based methods, recalculating the centroid is essential for making the algorithm less sensitive to initial conditions and more likely to find the "true" clusters.

After solving (1) using the PDHG algorithm, the output u will be thresholded to u_{hard}. More specifically, for every $i \in \{1, 2, \ldots, r\}$, the largest element among $((u_1)_i, (u_2)_i, \ldots, (u_k)_i)$ is set to 1, while the others are set to 0, and we claim the ith pixel belongs to that particular cluster. Then, the lth centroid is updated by taking the mean of all the pixels in that cluster. The process is repeated until the difference between two consecutive u_{hard} drops below a certain threshold. The pseudocode for the proposed linear model on HSI is listed in Algorithm 1.

Before ending the discussion of the proposed linear model, we point out its connection to the piecewise constant

[1] http://www.vlfeat.org

Algorithm 1 Linear Model

1: Initialization of centroids: Choose $(c_l)_{l=1}^k$ (randomized or generated by unsupervised centroid extraction algorithms).
2: Initialization of parameters: Choose $\tau, \sigma > 0$ satisfying $\sigma\tau\|\nabla_w\|^2 \leq 1$, $\theta = 1$
3: Initial iterate: Set $u^0 \in \mathbb{R}^{r\times k}$ and $p^0 \in \mathbb{R}^{(r\times r)\times k}$ randomly, set $\bar{u}^0 = u^0$, $u_{hard} = threshold(u^0)$
4: **while** not converge **do**
5: Minimize energy E_1 using PDHG algorithm
6: $u_{hard} = threshold(u)$
7: Update $(c_l)_{l=1}^k$
8: **end while**

Mumford–Shah model for multiclass graph segmentation [39]. Assume that the domain Ω of the HSI is segmented by a contour Φ into k disjoint regions, $\Omega = \cup_{l=1}^k \Omega_l$. The piecewise constant Mumford–Shah energy is defined as

$$E_{MS}\left(\Phi, \{c_l\}_{l=1}^k\right) = |\Phi| + \lambda \sum_{l=1}^{k} \int_{\Omega_l} |g(x) - c_l|^2 dx \quad (2)$$

where $|\Phi|$ is the length of the contour. To illustrate the connection between (1) and (2), consider the "local" version of (1), which essentially replaces the NLTV regularizer $\|\nabla_w u_l\|_{L^1}$ with its local counterpart

$$E_1^{\text{loc}}(u) = \sum_{l=1}^{k} \|\nabla u_l\|_{L^1} + \sum_{l=1}^{k} \int u_l(x) f_l(x) dx. \quad (3)$$

Assume that the labeling function u_l is the characteristic function of Ω_l. Then, $\int u_l(x) f_l(x) dx$ is equal to $\int_{\Omega_l} |g(x) - c_l|^2 dx$ up to a multiplicative constant. Moreover, the TV of a characteristic function of a region equals the length of its boundary, and hence, $|\Phi| = \sum_{l=1}^k \|\nabla u_l\|_{L^1}$. So the linear model (1) can be viewed as a nonlocal convex-relaxed version of Mumford–Shah model. We also note that the linear energy (1) has been studied in [23]. But in their work, the authors used a graph-based MBO method to minimize (1) instead of the PDHG algorithm, and the difference of the numerical performances can be seen in Section V.

B. Quadratic Model

1) Intuition: The aforementioned linear model performs very well when the centroids are initialized by accurate centroid extraction algorithms. As shown in Section V, the linear model can have a significant boost to the accuracy of other algorithms if the centroid extraction algorithm is reasonable, without sacrificing speed. However, if centroids are not extracted accurately, or if random initialization is used, the segmenting results are no longer reliable, and the algorithm takes far more iterations to converge to a stable classification.

To reduce the times of centroid updates and merge similar clusters automatically and simultaneously, the following quadratic model is proposed:

$$E_2(u) = \sum_{l=1}^{k} \|\nabla_w u_l\|_{L^1} + \sum_{l=1}^{k} \int u_l^2(x) f_l(x) dx. \quad (4)$$

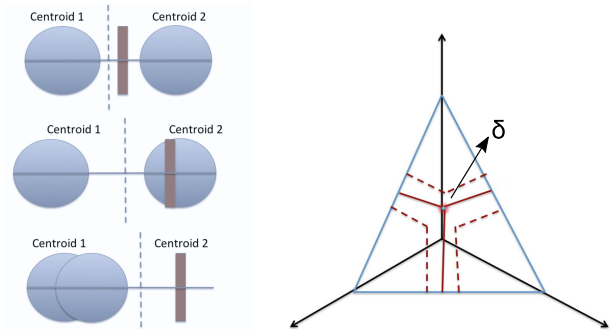


Fig. 1. First figure shows the "pushing" mechanism of the quadratic model. The horizontal line represents the unit simplex in $\mathbb{R}^2$. Signatures from cluster A_1 are colored blue, and signatures from cluster A_2 are colored brown. The vertical dashed bar is generated by a stable simplex clustering method, and it thresholds the points on the simplex into two categories. The second figure shows the stable simplex clustering. Every grid point δ on the simplex generates a simplex clustering. We want to choose a δ, such that there are very few data points falling into the "Y-shaped region."

Similar as before, $u = (u_1, u_2, \ldots, u_k) : \Omega \to \mathbb{K}^k$ is the labeling function, k is the number of clusters, $\mathbb{K}^k$ is the unit simplex in $\mathbb{R}^k$, and $f_l(x)$ is the error function.

Note that the only difference between (1) and (4) is that the power of the labeling function u_l here is two. The intuition for this is as follows.

Consider, for simplicity, a hyperspectral image with a ground truth of only two clusters, A_1 and A_2. Suppose the randomized initial centroids are chosen, such that $c_1 \approx c_2 \in A_1$; or, that the two random initial pixels are of very similar spectral signatures and belong to the same ground-truth cluster.

Let x be a pixel from A_2. Then, $0 \ll |g(x) - c_1|^2 \approx |g(x) - c_2|^2$. When (1) is applied, the fidelity term $\langle u, f\rangle$ does not change when $u(x)$ moves on the simplex in $\mathbb{R}^2$, and thus, pixels of A_2 will be scattered randomly on the simplex. After thresholding, an approximately equal number of pixels from cluster A_2 will belong to clusters C_1 and C_2, so the new centroids $\tilde{c}_1$ and $\tilde{c}_2$ that are the means of the spectral signatures of the current clusters will once again be approximately equal.

This situation changes dramatically when (4) is minimized.

1) Observe that the fidelity term in E_2 is minimized for a pixel $x \in A_2$ when $u_1(x) \approx u_2(x) \approx (1/2)$. Therefore, the pixels of cluster A_2 will be "pushed" toward the center of the simplex once E_2 is minimized.
2) With a stable simplex clustering method (explained in Section III-B2), the clusters are divided, such that all of these pixels in the center belong to either C_1 or C_2; without loss of generality, suppose they belong to C_2. Then, the updated centroid $\tilde{c}_1$ is essentially c_1, while the updated centroid $\tilde{c}_2$ is a linear combination of the spectral signature of members belonging to A_1 and A_2, and thus quite different from the original c_2.
3) After minimizing the energy E_2 again, pixels from A_1 will be clustered in C_1, and pixels from A_2 will be pushed to C_2. Therefore, the clustering will be finished in just two steps in theory. See Fig. 1 for a graphical illustration.

The quadratic model not only reduces the number of iterations needed to find the "true" clustering because of its

Algorithm 2 Quadratic Model with Stable Simplex Clustering

1: Initialization of centroids: Choose $(c_l)_{l=1}^k$ (randomized or generated by unsupervised centroid extraction algorithms).
2: Initialization of parameters: Choose $\tau, \sigma > 0$ satisfying $\sigma\tau\|\nabla_w\|^2 \leq 1$, $\theta = 1$
3: Initial iterate: Set $u^0 \in \mathbb{R}^{r\times k}$ and $p^0 \in \mathbb{R}^{(r\times r)\times k}$ randomly, set $\bar{u}^0 = u^0$,
4: **while** not converge **do**
5: Minimize energy E_2 using PDHG algorithm
6: $u_{hard} = threshold(u)$ with stable simplex clustering
7: Update $(c_l)_{l=1}^k$
8: **end while**

capability of anomaly detection, but it allows for random initialization as well, making it a more robust technique.

2) Stable Simplex Clustering: As mentioned earlier, the quadratic model pushes anomalies into the middle of the unit simplex. Therefore, it would be ill-conceived to simply classify the pixels based on the largest component of the labeling function $u(x) = (u_1(x), u_2(x), \ldots, u_k(x))$. Instead, a stable simplex clustering method has to be used.

The concept behind the stable simplex clustering is to choose a division that puts all the data points in the "middle" of the unit simplex into a single cluster. Fig. 1 shows this in the simple two-cluster case. Also refer to Section III-B1 for explanation of the "pushing" process. The idea to accomplish this goal is inspired by [5]. We first create a grid on a k-dimensional simplex, where k is the number of clusters, and each grid point δ generates a simplex clustering. Then, a δ is searched to minimize the energy $g(\delta)$

$$g(\delta) = -\log\left(\prod_{l=1}^{k} F_l(\delta)\right) + \eta\exp(G(\delta))$$

where $F_l(\delta)$ is the percentage of data points in cluster l, and $G(\delta)$ is the percentage of data points on the edges near the division, i.e., the "Y-shaped region" in Fig. 1. The first term in $g(\delta)$ rewards keeping clusters approximately of the same size, ensuring no skewed data from clusters far too small. And the second term rewards sparsity of points in the intermediate region. The constant η is chosen to be large enough, such that stability has a bigger weight in the energy.

Algorithm 2 shows the quadratic model using stable simplex clustering. Fig. 2 shows how this detected the chemical plumes in a frame with background centroids precalculated and random initialization for the final centroid. Notice that no plume is detected in the first iteration. But by the 12th iteration, the gas plume is nearly perfectly segmented.

Finally, we present the comparison between the results of the linear model and the quadratic model on the Urban data set with identical random pixel initialization in Fig. 3. The linear model took about 50 iterations to converge, and the quadratic model only took four iterations.

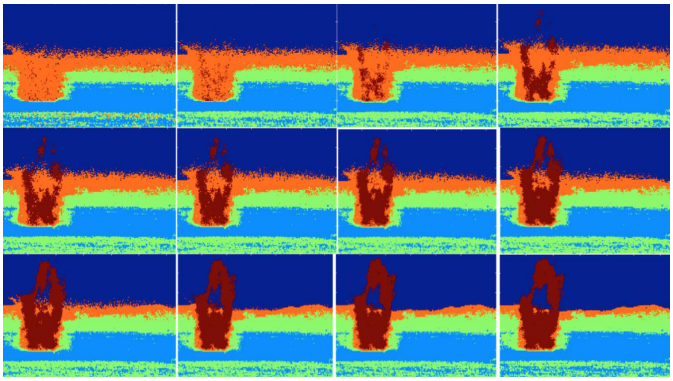


Fig. 2. Quadratic model and stable simplex clustering on the plume data set. The chemical plume (brown) is perfectly detected in 12 iterations.

Algorithm 3 Primal-Dual Hybrid Gradient Algorithm

1: Initialization: Choose $\tau, \sigma > 0$, $\theta \in [0, 1]$, $(x^0, y^0) \in X \times Y$, and set $\bar{x}^0 = x^0$
2: **while** not converge **do**
3: $y^{n+1} = (I + \sigma\partial F^*)^{-1}(y^n + \sigma K\bar{x}^n)$
4: $x^{n+1} = (I + \tau\partial G)^{-1}(x^n - \tau K^* y^{n+1})$
5: $\bar{x}^{n+1} = x^{n+1} + \theta(x^{n+1} - x^n)$
6: $n = n + 1$
7: **end while**

IV. Primal-Dual Hybrid Gradient Algorithm

In this section, a detailed explanation is provided on the application of the PDHG algorithm [30], [35]–[37] to minimizing E_1 (1) and E_2 (4) in Section III-B2. A review of the algorithm is provided in a more general setting to contextualize the extension to nonlocal model for hyperspectral imagery.

A. Review of PDHG Algorithm

Consider the following convex optimization problem:

$$\min_{x\in X}\{F(Kx) + G(x)\} \tag{5}$$

where X and Y are finite-dimensional real vector spaces, F and G are proper convex lower semicontinuous functions $F : Y \to [0, \infty]$, $G : X \to [0, \infty]$, and $K : X \to Y$ is a continuous linear operator with the operator norm $\|K\| = \sup\{\|Kx\| : x \in X, \|x\| \leq 1\}$. The primal-dual formulation of (5) is the saddle-point problem

$$\min_{x\in X}\max_{y\in Y}\{\langle Kx, y\rangle - F^*(y) + G(x)\} \tag{6}$$

where F^* is the convex conjugate of F defined as $F^*(y) = \sup_x < x, y > -F(x)$.

The saddle-point problem (6) is then solved using the iterations of [30, Algorithm 3].

In Algorithm 3, $(I + \lambda\partial f)^{-1}(x)$ is the proximal operator of f, which is defined as

$$(I + \lambda\partial f)^{-1}(x) = \text{prox}_{\lambda f}(x) = \arg\min_y f(y) + \frac{1}{2\lambda}\|y - x\|_2^2.$$

It has been shown in [30] that $O(1/N)$ (where N is the number of iterations) convergence can be achieved as long as σ, τ satisfy $\sigma\tau\|K\|^2 \leq 1$.

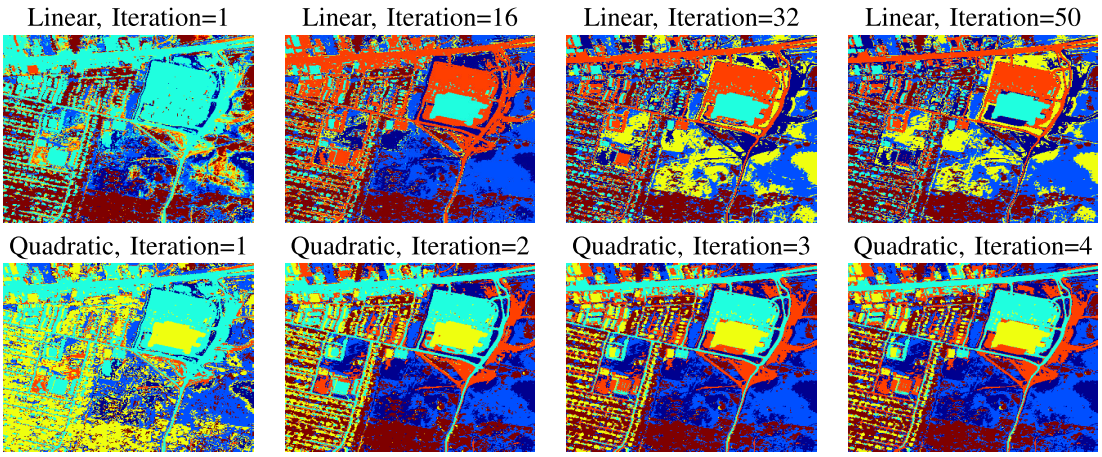


Fig. 3. Linear versus quadratic model on the Urban data set with the same centroid initialization. To produce essentially identical results, the linear model (first row) took 50 iterations of centroid updates, and the quadratic model (second row) took just 4 iterations.

B. Primal-Dual Iterations to Minimize E_1 and E_2

Recall from Section III that the discretized linear and quadratic energy E_1 and E_2 are

$$\begin{aligned} E_1(u) &= \sum_{l=1}^{k} \|\nabla_w u_l\|_{L^1} + \sum_{l=1}^{k}\sum_{i=1}^{r} (u_l)_i (f_l)_i \\ &= \|\nabla_w u\|_{L^1} + \langle u, f\rangle \\ E_2(u) &= \sum_{l=1}^{k} \|\nabla_w u_l\|_{L^1} + \sum_{l=1}^{k}\sum_{i=1}^{r} (u_l)_i^2 (f_l)_i \\ &= \|\nabla_w u\|_{L^1} + \langle u, f \odot u\rangle \end{aligned}$$

where $u = (u_1, u_2, \ldots, u_k)$ is a nonnegative matrix of size $r \times k$, with each row of matrix u summing to one, and $f \odot u$ denotes the pointwise product between two matrices f and u. After adding an indicator function δ_U, minimizing E_1 and E_2 are equivalent to solving

$$\min_u \|\nabla_w u\|_{L^1} + \langle u, f\rangle + \delta_U(u) \tag{7}$$

$$\min_u \|\nabla_w u\|_{L^1} + \langle u, f \odot u\rangle + \delta_U(u) \tag{8}$$

where $U = \{u = (u_1, u_2, \ldots, u_k) \in \mathbb{R}^{r\times k} : \sum_{l=1}^{k} (u_l)_i = 1, \forall i = 1, \ldots, r, (u_l)_i \geq 0\}$, and δ_U is the indicator function on U. More specifically

$$\delta_U(u) = \begin{cases} 0 & \text{if } u \in U \\ \infty & \text{otherwise.} \end{cases} \tag{9}$$

By comparing (5), (7), and (8), we can set $K_1 = K_2 = \nabla_w$, $F_1(q) = F_2(q) = \|q\|_{L^1}$, $G_1(u) = \langle u, f\rangle + \delta_U(u)$, and $G_2(u) = \langle u, f \odot u\rangle + \delta_U(u)$. The convex conjugate of F_1 (and F_2) is $F_1^*(p) = F_2^*(p) = \delta_P(p)$, where the set $P = \{p \in \mathbb{R}^{(r\times r)\times k} : \|p_l\|_\infty \leq 1\}$.

Next, we derive the closed forms of the proximal operators $(I + \sigma\partial F_{1,2}^*)^{-1}$ and $(I + \tau\partial G_{1,2})^{-1}$, so that Algorithm 3 can be implemented efficiently to minimize E_1 and E_2

$$\begin{aligned} (I + \sigma\partial F_{1,2}^*)^{-1}(\tilde{p}) &= (I + \sigma\partial\delta_P)^{-1}(\tilde{p}) \\ &= \arg\min_p \delta_P(p) + \frac{1}{2\sigma}\|p - \tilde{p}\|_2^2 = \text{proj}_P(\tilde{p}) \end{aligned} \tag{10}$$

Algorithm 4 Primal-Dual Iterations for the Linear Model

1: **while** not converge **do**
2: $p^{n+1} = \text{proj}_P(p^n + \sigma\nabla_w \bar{u}^n)$
3: $u^{n+1} = \text{proj}_U(u^n + \tau \text{div}_w p^{n+1} - \tau f)$
4: $\bar{u}^{n+1} = u^{n+1} + \theta(u^{n+1} - u^n)$
5: $n = n + 1$
6: **end while**

where $\text{proj}_P(\tilde{p})$ is the projection of $\tilde{p}$ onto the closed convex set P

$$\begin{aligned} &(I+\tau\partial G_1)^{-1}(\tilde{u}) = \arg\min_u < u, f > +\delta_U(u) + \frac{1}{2\tau}\|u - \tilde{u}\|_2^2 \\ &= \arg\min_{u\in U} \|u - \tilde{u} + \tau f\|_2^2 = \text{proj}_U(\tilde{u} - \tau f) \\ &\quad \cdot (I + \tau\partial G_2)^{-1}(\tilde{u}) = \arg\min_u \left\langle u, \frac{\tau}{2}\mathcal{A}u\right\rangle \\ &\quad + \tau\delta_U(u) + \frac{1}{2}\|u - \tilde{u}\|_2^2 \end{aligned} \tag{11}$$

$$= \arg\min_{u\in U} \frac{1}{2}\langle u, (I+\tau\mathcal{A})u\rangle - \langle u, \tilde{u}\rangle + \frac{1}{2}\left\langle \tilde{u}, (I + \tau\mathcal{A})^{-1}\tilde{u}\right\rangle$$

$$= \arg\min_{u\in U} \frac{1}{2}\|(I + \tau\mathcal{A})^{\frac{1}{2}}u - (I + \tau\mathcal{A})^{-\frac{1}{2}}\tilde{u}\|_2^2 \tag{12}$$

where $\mathcal{A} : \mathbb{R}^{r\times k} \to \mathbb{R}^{r\times k}$ is a linear operator defined as $1/2\mathcal{A}u = f \odot u$. Therefore, $\mathcal{A}$ is a positive semidefinite diagonal matrix of size $rk \times rk$. It is worth mentioning that the matrix $(I + \tau\mathcal{A})$ is diagonal and positive definite, and hence, it is trivial to compute its inverse and square root. Problem (12) can be solved as a preconditioned projection onto the unit simplex $\mathbb{K}^k$, and the solution will be explained in Section IV-C.

Combining (10)–(12) and Algorithm 3, we have the primal-dual iterations for minimizing E_1 (Algorithm 4) and E_2 (Algorithm 5).

Before moving on to explaining how to solve (12), we specify the two orthogonal projections proj_P and proj_U in Algorithm 4. Let $\tilde{p} = \text{proj}_P(p)$, where $p = (p_l)_{l=1}^k \in \mathbb{R}^{(r\times r)\times k}$. Then, for every $i \in \{1, 2, \ldots, r\}$ and every $l \in \{1, 2, \ldots, k\}$, the ith row of $\tilde{p}_l$ is the projection of the ith row of p_l on to the unit ball in $\mathbb{R}^r$. Similarly, if $\tilde{u} = \text{proj}_U(u)$,

Algorithm 5 Primal-Dual Iterations for the Quadratic Model

while not converge **do**
  $p^{n+1} = \text{proj}_P(p^n + \sigma \nabla_w \bar{u}^n)$
  Update u^{n+1} as in (12), where $\tilde{u} = u^n + \tau \text{div}_w p^{n+1}$
  $\bar{u}^{n+1} = u^{n+1} + \theta(u^{n+1} - u^n)$
  $n = n + 1$
end while

then for every $i \in \{1, 2, \ldots, r\}$, $((\tilde{u}_1)_i, (\tilde{u}_2)_i, \ldots, (\tilde{u}_k)_i)$ is the projection of $((u_1)_i, (u_2)_i, \ldots, (u_k)_i)$ onto the unit simplex $\mathbb{K}^k$ in $\mathbb{R}^k$.

C. Preconditioned Projection Onto the Unit Simplex

This section is dedicated to solving (12). It is easy to see that the rows of u in (12) are decoupled, and the only problem that needs to be solved is

$$\min_{u \in \mathbb{R}^k} \delta_{\mathbb{K}^k}(u) + \frac{1}{2}\|Au - y\|^2 \tag{13}$$

where $A = \text{diag}(a_1, a_2, \ldots, a_k)$ is a positive definite diagonal matrix of size $k \times k$, $\mathbb{K}^k$ is the unit simplex in $\mathbb{R}^k$, and $y \in \mathbb{R}^k$ is a given vector.

Theorem 1: The solution $u = (u_1, u_2, \ldots, u_k)$ of (13) is

$$u_i = \max\left(\frac{a_i y_i - \lambda}{a_i^2}, 0\right) \tag{14}$$

where λ is the unique number satisfying

$$\sum_{i=1}^{k} \max\left(\frac{a_i y_i - \lambda}{a_i^2}, 0\right) = 1 \tag{15}$$

The proof of Theorem 1 is shown in the Appendix. The most computationally expensive part of solving (15) is sorting the sequence $(a_i y_i)_{1 \le i \le k}$ of length k, which is trivial, since k, the number of clusters, is typically a small number.

V. Numerical Results

A. Comparison Methods and Experimental Setup

All experiments were run on a Linux machine with Intel core i5, 3.3 Hz with 2 GB of DDR3 RAM. The following unsupervised algorithms have been tested.

1) *(Spherical) K-Means:* Built in MatLab Code.
2) *NMF:* Nonnegative Matrix Factorization [40].
3) *H2NMF:* Hierarchical Rank-2 NMF [5].
4) *MBO:* Graph MBO scheme [23], [24]. The code is run for ten times on each data set, and the best result is chosen.
5) *NLTV2:* NLTV, quadratic model with random pixel initialization.
6) *NLTV1(H2NMF/K-Means):* NLTV, linear model with endmembers/centroids extracted from H2NMF/K-means.

Every algorithm can be initialized via the same procedure as that in "K-means++" [41], and the name "Algorithm++" is used if the algorithm is initialized in such a way. For example, "NLTV2++" means NLTV, quadratic model with "K-means++" initialization procedure.

The algorithms are compared on the following data sets.

1) *Synthetic Data Set:* This data set[2] contains five endmembers and 162 spectral bands. The 40 000 abundance vectors were generated as a sum of Gaussian fields. The data set was generated using a generalized bilinear mixing model

$$y = \sum_{i=1}^{p} a_i e_i + \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \gamma_{ij} a_i a_j e_i \odot e_j + n$$

where γ_{ij} values are chosen uniformly and randomly in the interval [0, 1], n is the Gaussian noise, with an SNR of 30 dB, and a_i satisfies: $a_i \ge 0$, and $\sum_{i=1}^{p} a_i = 1$.
2) *Salinas-A Data Set:* Salinas-A scene[3] was a small subscene of Salinas image, which was acquired by the AVIRIS sensor over Salinas Valley. It contains 86×83 pixels and 204 bands. The ground truth includes six classes: broccoli, corn, and four types of lettuce.
3) *Urban Data Set:* The Urban data set[4] is from the Hyperspectral Digital Imagery Collection Experiment (HYDICE), which has 307×307 pixels and contains 162 clean spectral bands. This data set only has six classes of material: road, dirt, house, metal, tree, and grass.
4) *San Diego Airport Data Set:* The San Diego Airport (SDA) data set[5] is provided by the HYDICE sensor. It comprises 400×400 pixels and contains 158 clean spectral bands. There are seven types of material: trees, grass, three types of road surfaces, and two types of rooftops [5]. The RGB image with cluster labels is shown in Fig. 7.
5) *Chemical Plume Data Set:* The chemical plume data set[6] consists of frames taken from a hyperspectral video of the release of chemical plumes provided by the Applied Physics Laboratory, John Hopkins University. The image has 128×320 pixels, with 129 clean spectral bands. There was no ground truth provided for this data, so a segmentation of four classes is assumed: chemical plume, sky, foreground, and mountain. A fifth cluster is added, so that the noise pixels would not interfere with the segmentation [23].
6) *Pavia University Data Set:* The Pavia University data set is collected by the ROSIS sensor. It contains 103 clean spectral bands and 610×340 pixels, and comprises nine classes of material.
7) *Indian Pines Data Set:* The Indian Pines data set was acquired by AVIRIS sensor and consists of 145×145 pixels, with 200 clean spectral bands. The available ground truth is labeled into 16 classes.
8) *Kennedy Space Center Data Set:* This data set was gathered by the NASA AVIRIS sensor over the Kennedy Space Center (KSC), Florida. A subscene of the western shore of the center is used in the numerical experiment.

[2] Available at http://www.math.ucla.edu/~weizhu731/

[3] Available at http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes

[4] Available at http://www.agc.army.mil/

[5] Available at http://www.math.ucla.edu/~weizhu731/

[6] Available at http://www.math.ucla.edu/~weizhu731/

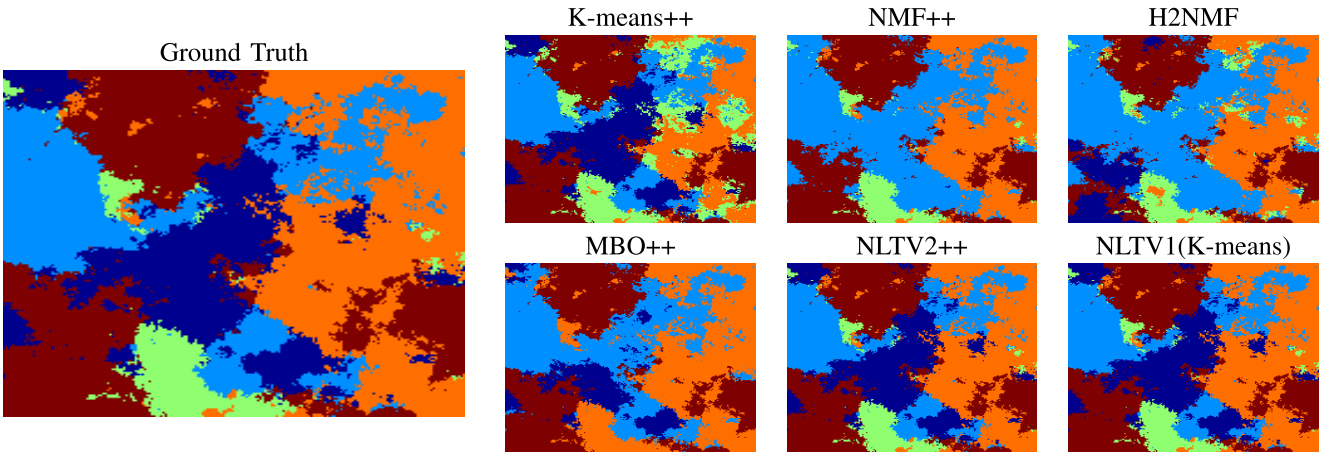


Fig. 4. Clustering results for the synthetic data set generated by five endmembers. The first image on the left is the ground truth, and the remaining six images are the clustering results of the corresponding algorithms.

TABLE I
KEY PARAMETERS USED FOR DIFFERENT DATA SETS

Datasets	λ	μ	Datasets	λ	μ
Synthetic	10^{-1}	10^{-4}	**Plume**	10^{7}	10^{-2}
Urban	10^{6}	10^{-5}	**Pavia**	10^{6}	10^{-8}
Salinas-A	10^{4}	10^{-4}	**Pines**	10^{6}	10^{-9}
SDA	10^{6}	10^{-7}	**KSC**	10^{6}	10^{-8}

Twelve classes of different materials are reported in the datacube of size $512 \times 365 \times 176$.

K-means and NMF are nonparametric, and the parameter setups of H2NMF and the MBO scheme are described in [5], [23], and [24]. The key parameters λ and μ in the NLTV models are determined in the following way.

1) λ is chosen, such that the data fidelity term is around ten times larger than the NLTV-regularizing term $\|\nabla_w u\|_{L^1}$.
2) μ is chosen, such that the Euclidean distances between different endmembers are roughly ten times smaller than the cosine distances.

Table I displays the parameters chosen for the numerical experiments. The large variance of the parameter scales results from the variety of image sizes and scales. A sensitivity analysis over the parameters is presented in Section V-G.

B. Synthetic Data Set and Salinas-A Data Set

All the algorithms are first tested on the synthetic data set. The classification results are shown in Table II and Fig. 4. Both NLTV algorithms have better overall accuracy than all of the other methods, although they took a longer time to converge. The qudratic model classified the image almost perfectly.

The visual classification results and overall accuracies of the Salinas-A data set are shown in Fig. 5 and Table II. Both NLTV methods performed at higher accuracy compared with other methods. The linear model improved the result of K-means by incorporating spatial information of the data set, and the quadratic model only took four iterations to converge.

TABLE II
COMPARISON OF NUMERICAL RESULTS ON THE SYNTHETIC AND SALINAS-A DATA SETS

Algorithm	Synthetic		Salinas-A	
	Run-Time	Accuracy	Run-Time	Accuracy
K-means++	2s	90.98%	0.9s	79.92%
NMF++	9s	80.99%	1.0s	64.47%
H2NMF	2s	72.02%	1.5s	70.08%
MBO++	21s	84.49%	7.8s	68.62%
NLTV2++	29s	99.93%	1.6s	83.69%
NLTV1(K-means)	29s	95.96%	3.4s	83.75%

C. Urban Data Set

There was no ground truth provided for the Urban HSI. A structured sparse algorithm [42] (which is different from all of the testing algorithms) has been used to initialize a ground truth, which is then corrected pixel by pixel to provide a framework for numerical analysis of accuracy. As this "ground truth" was hand-corrected, it does not necessarily represent the most accurate segmentation of the image; however, it provides a basis for quantitative comparison.

After running all the algorithms that are compared to create six clusters, we noticed that they all split "grass" into two different clusters (one of them corresponds to a mixture of grass and dirt), while treating "road" and "metal" as the same. To obtain a reliable overall accuracy of the classification results, the two "grass" clusters are combined in every algorithm, hence obtaining the classification results for five clusters, which are "grass," "dirt," "road + metal," "roof," and "tree."

The overall classification accuracies and run times are displayed in Table III. As can be seen, the proposed NLTV algorithms performed consistently better with comparable run time. It is easier to see visually in Fig. 6 that the NLTV algorithm performed best of the five algorithms tested; specifically, the NLTV algorithm alone distinguished all of the dirt beneath the parking lot and the intricacies of the road around the parking lot. The TV regularizer also gives the segmented image smoother and more distinct edges, allowing easier human identification of the clusters.

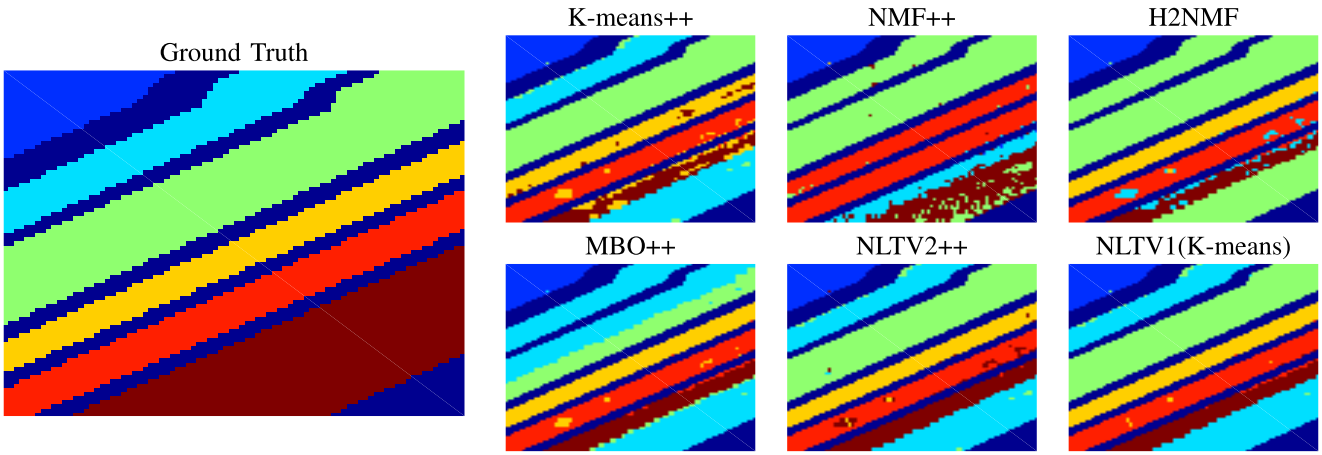


Fig. 5. Clustering results for the Salina-A data set. The first image on the left is the ground truth, and the remaining six images are the clustering results of the corresponding algorithms.

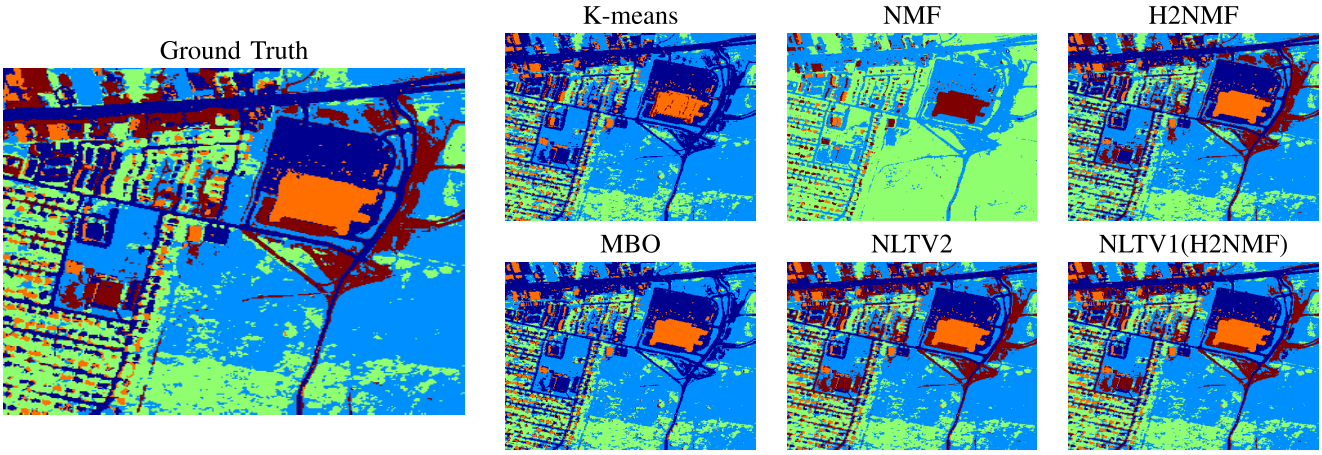


Fig. 6. Clustering results for the Urban data set. Five clusters, including rooftops, grass, trees, dirt, and "road+metal," are generated by the algorithms.

TABLE III
COMPARISON OF NUMERICAL RESULTS ON THE URBAN DATA SET

Algorithm	Run-Time	Accuracy
K-means	7s	75.20%
NMF	87s	55.70%
H2NMF	7s	85.96%
MBO	92s	78.86%
NLTV2	96s	92.14%
NLTV1(H2NMF)	17s	91.56%

TABLE IV
RUN TIMES FOR THE SDA, CHEMICAL PLUME (PLUME), PAVIA UNIVERSITY (PAVIA), INDIAN PINES (PINES), AND KSC DATA SETS

Algorithm	SDA	Plume	Pavia	Pines	KSC
K-means	9s	2s	26s	10s	47s
NMF	4s	2s	120s	19s	135s
H2NMF	13s	2s	12s	4s	24s
MBO	329s	18s	1020s	198s	754s
NLTV2	43s	23s	299s	64s	561s
NLTV1(H2NMF)	17s	18s	132s	21s	188s

D. San Diego Airport Data Set

The classification results and computational run times are shown in Fig. 7 and Table IV. No ground-truth classification is available for this HSI, but after examining the spectral signatures of various pixels in the scene, we managed to pinpoint some errors that were common for each algorithm. We will not go into detail about the NMF and H2NMF algorithms, which clearly do not perform well on this data set. K-means obtained some decent results, but splitted the rooftops of the four buildings on the bottom-right of the image into two distinct clusters, and failed to separate two different road types (clusters 5 and 6). The MBO scheme failed on two accounts: it did not properly segment two different road surfaces (clusters 6 and 7), and did not account for the different rooftop types (clusters 3 and 4). The linear NLTV model with H2NMF initialization is significantly more accurate than H2NMF and MBO. It successfully picked out two different types of roof (clusters 3 and 4) and two different types of road (clusters 6 and 7), although the other type of road (cluster 5) is mixed with one type of roof (cluster 3). The best result was obtained by using the NLTV quadratic model with random initialization, with the only problem that tree and grass (clusters 1 and 2) are mixed together. However, the mixing of grass and tree is actually the case for all the other algorithms. This means that NLTV quadratic model alone was able to identify six of the seven clusters correctly.

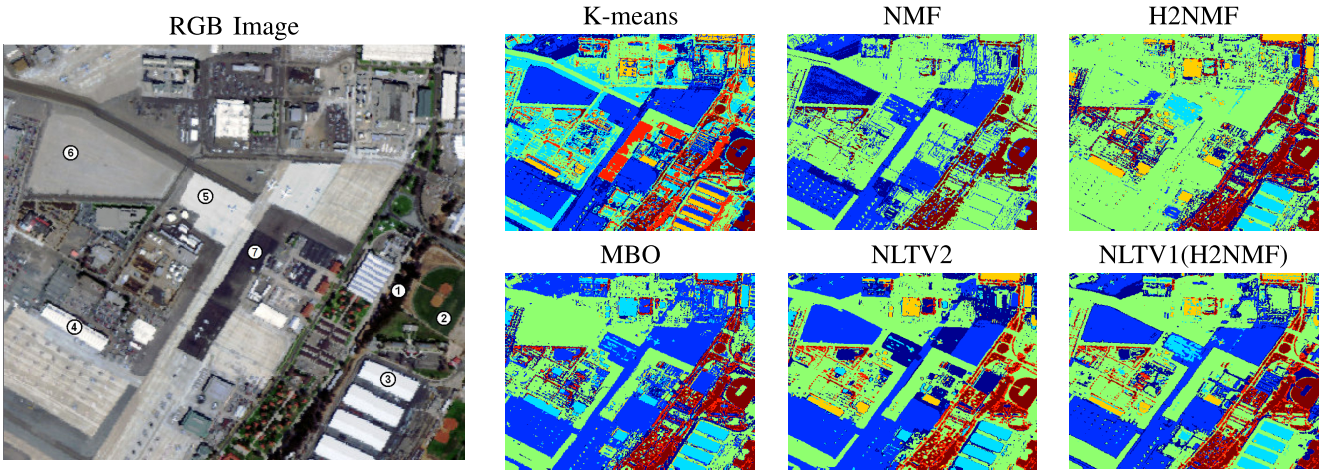


Fig. 7. Clustering results for the SDA data set. The first image on the left is the RGB image, and the remaining six images are the clustering results of the corresponding algorithms.

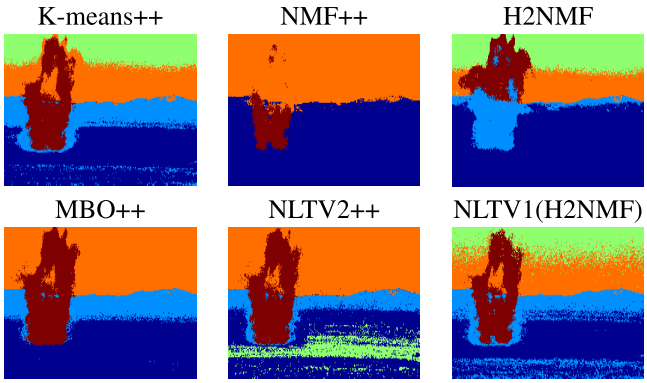


Fig. 8. Clustering results for the Chemical Plume data set.

E. Chemical Plume Data Set

Analyzing images for chemical plumes is more difficult because of its diffusive nature. All the algorithms are run on the image before it was denoised and the results are shown in Fig. 8. The unmixing methods, such as NMF and H2NMF, do not perform satisfactorily on this data set. MBO++, K-means++, and NLTV2++ can all properly identify the chemical plume. Note that NLTV with H2NMF as centroid initialization outperforms H2NMF as a classification method. We have to point out that the NLTV quadratic model is not so robust with respect to the centroid initialization even with a "K-means++" type procedure on this data set. But, this is also the case for all the other testing algorithms. The MBO scheme, which was specifically designed for this data set [23], does seem to have the highest robustness among all the algorithms.

F. Pavia University, Indian Pines, and Kennedy Space Center Data Set

The Pavia University (9 clusters), Indian Pines (16 clusters), and KSC (12 clusters) data sets are frequently used to test supervised classification algorithms. To save space, we only report the numerical overall accuracies in Table V. As can be seen, all the competing unsupervised algorithms performed poorly on these three data sets. Different clusters were merged, and the same clusters were splitted in various fashions by all the algorithms, which rendered the numerical accuracies no longer reliable.

TABLE V

COMPARISON OF OVERALL ACCURACIES ON THE PAVIA UNIVERSITY, INDIAN PINES, AND KSC DATA SETS

Algorithm	Pavia	Pines	KSC
K-means++	42.31%	38.99%	41.73%
NMF++	54.97%	38.84%	37.07%
H2NMF	43.75%	36.78%	37.07%
MBO++	50.04%	36.49%	41.85%
NLTV, H2NMF init	42.83%	36.22%	41.41%
NLTV++	44.01%	42.35%	41.48%

The computational run times of these three data sets are listed in Table IV. Unfortunately, when the number of clusters is increasing, the computational complexity of the quadratic model grows exponentially. The reason is that the number of grid points (δ in Fig. 1) on the unit simplex grows exponentially as the dimension of the simplex increases. Therefore, when the number of clusters is large enough (greater than 10), the stable simplex clustering will become the most time-consuming part of the quadratic model. On these three data sets, we sacrificed the accuracy of the quadratic model by creating a coarser mesh on the unit simplex.

The reason why NLTV, as well as all the other competing unsupervised algorithms, performed poorly on these three data sets is twofold. First, when the number of classes is too large in an HSI covering a large geographic location, the variation of spectral signatures within the same class cannot be neglected when compared with the difference between the constitutive materials, especially when the endmembers themselves are similar. As a result, the unsupervised algorithms tend to split a ground-truth cluster with large variation in spectral signatures and merge clusters with similar centroids or endmembers. Second, there might exist more distinct materials in the image than reported in the ground truth. Therefore, the algorithms might detect those unreported materials, because no labeling has been used in these unsupervised algorithms. Thus, we can conclude that NLTV, as well as other unsupervised methods reported in this paper, is not suitable for such images at current

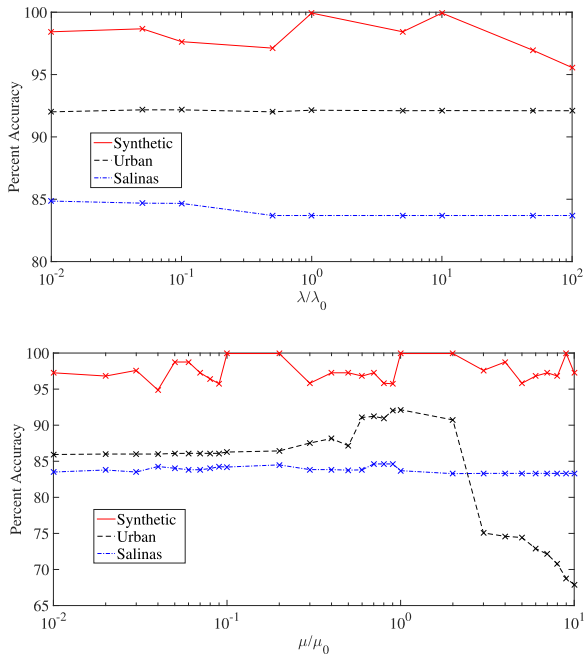


Fig. 9. This figure shows the robustness of the NLTV algorithm with respect to λ and μ. Centroid initialization remains identical as λ and μ are changing. λ_0 and μ_0 are the optimal values specified in Section V-A. The overall accuracies of the Synthetic, Urban, and Salinas-A data sets are displayed.

stage. Modifying the NLTV algorithm to work for such data sets would be the direction of future work.

G. Sensitivity Analysis Over Key Model Parameters

At last, a sensitivity analysis is provided over the parameters λ and μ in the NLTV models. As mentioned in Section V-A, λ and μ are chosen to balance the scale of the regularizing and fidelity terms or the cosine and Euclidean distances. Fig. 9 shows the robustness of the NLTV algorithm on the Synthetic, Urban, and Salinas-A data sets with respect to λ and μ within the variance of two magnitudes. Centroid initialization remains identical as λ and μ are changing. It is clear that the NLTV algorithm is fairly robust with respect to λ on all three data sets. The algorithm is also relatively robust with respect to μ on the Synthetic and Salinas-A data sets. As for the Urban data set, a significant decay in accuracy can be observed as μ increases. This phenomenon is due to the fact that larger μ causes Euclidean distance to be the dominant one, which is not ideal with the presence of atmospheric interference in the Urban data set. Smaller μ also leads to lower accuracy in the Urban data set, which results from the similarity of "road" and "dirt" clusters measured in cosine distance. Overall, a reasonable robustness with respect to the key parameters λ and μ can be concluded on these three tests.

Similar robustness can be observed on other data sets except for the Chemical Plume. Fig. 10 shows the sensitivity of the result with respect to μ. All the centroids are initialized using H2NMF, and vastly different results occurred as μ changes. This could be due to the presence of significant noise.

VI. Conclusion

In this paper, we present the framework for an NLTV method for unsupervised HSI classification, which is solved with the PDHG algorithm. A linear and a quadratic version of this model are developed; the linear version updates more quickly and can refine results produced by a centroid extraction algorithm, and the quadratic model with stable simplex clustering method provides a robust means of classifying HSI with randomized pixel initialization.

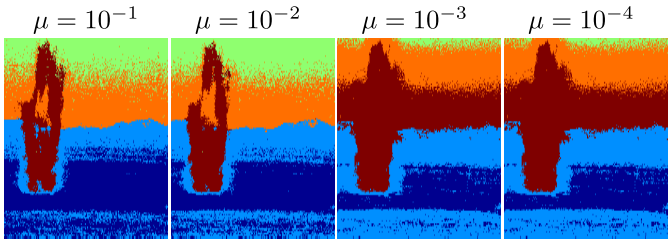


Fig. 10. Sensitivity of the NLTV algorithm with respect to μ in the plume data set. All the tests used the same centroid initialization (H2NMF).

The algorithm is tested on both synthetic and seven real-world data sets, with promising results. The proposed NLTV algorithm consistently performed with highest accuracy on synthetic and urbanized data sets such as Urban, Salinas-A, and the SDA, both producing smoother results with easier visual identification of segmentation, and distinguishing classes of material that other algorithms failed to differentiate. The NLTV algorithm also performed well on anomaly detection scenarios, such as the Chemical Plume data sets; with proper initialization, it performed on par with the MBO scheme developed specifically for this data set. However, NLTV, as well as other unsupervised algorithms, failed to achieve satisfactory results on data sets with a relatively large number of clusters. The run times of the NLTV algorithms are generally comparable to the other methods, and the consistent higher accuracy on different types of data sets suggests that this technique is a more robust and precise means of classifying hyperspectral images with a moderate number of clusters.

Appendix
Proof of Theorem 1

Problem (13) is equivalent to

$$\min_{\sum_{i=1}^{k} u_i = 1} \delta_{\mathbb{R}_+^k}(u) + \frac{1}{2}\|Au - y\|_2^2 \tag{16}$$

where $\mathbb{R}_+^k = \{u \in \mathbb{R}^k : u_i \geq 0\}$ is the nonnegative quadrant of $\mathbb{R}^k$. The Lagrangian of (16) is

$$\mathcal{L}(u, \lambda) = \sum_{i=1}^{k} \left(\frac{1}{2}|a_i u_i - y_i|^2 + \delta_{\mathbb{R}_+}(u_i) + \lambda u_i \right) - \lambda.$$

If u^* is a soluton of (16), KKT conditions [43] imply that there exists a λ, such that

$$u^* = \arg\min_u \mathcal{L}(u, \lambda) = \arg\min_{u_i \geq 0} \sum_{i=1}^{k} \frac{1}{2} a_i^2 \left(u_i + \frac{\lambda - a_i y_i}{a_i^2} \right)^2.$$

Therefore, $u_i^* = \max(a_i y_i - \lambda/a_i^2, 0)$. Meanwhile, the primal feasibility requires

$$\sum_{i=1}^{k} u_i^* = \sum_{i=1}^{k} \max\left(\frac{a_i y_i - \lambda}{a_i^2}, 0 \right) = 1.$$

And this proves Theorem 1.

ACKNOWLEDGMENT

The authors would like to thank Z. Meng and J. Sunu for providing and helping with the Merriman–Bence–Osher code.

REFERENCES

[1] C.-I. Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*, vol. 1. New York, NY, USA: Kluwer, 2003.

[2] J. M. Bioucas-Dias *et al.*, "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.

[3] N. Gillis and S. A. Vavasis, "Fast and robust recursive algorithms for separable nonnegative matrix factorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 698–714, Apr. 2014.

[4] S. Jia and Y. Qian, "Constrained nonnegative matrix factorization for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 1, pp. 161–173, Jan. 2009.

[5] N. Gillis, D. Kuang, and H. Park, "Hierarchical clustering of hyperspectral images using rank-two nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2066–2078, Apr. 2015.

[6] M. Soret, S. L. Bacharach, and I. Buvat, "Partial-volume effect in PET tumor imaging," *J. Nucl. Med.*, vol. 48, no. 6, pp. 932–945, 2007.

[7] 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.

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

[9] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[10] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1351–1362, Jun. 2005.

[11] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.

[12] B. Demir and S. Erturk, "Hyperspectral image classification using relevance vector machines," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 4, pp. 586–590, Oct. 2007.

[13] G. M. Foody, "RVM–based multi-class classification of remotely sensed data," *Int. J. Remote Sens.*, vol. 29, no. 6, pp. 1817–1823, 2008.

[14] F. A. Mianji and Y. Zhang, "Robust hyperspectral classification using relevance vector machine," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2100–2112, Jun. 2011.

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

[16] M. Stoer and F. Wagner, "A simple min-cut algorithm," *J. ACM*, vol. 44, no. 4, pp. 585–591, Jul. 1997. [Online]. Available: http://doi.acm.org/10.1145/263867.263872

[17] A. Szlam and X. Bresson, "A total variation-based graph clustering algorithm for Cheeger ratio cuts," Univ. California, Los Angeles, Los Angeles, CA, USA, Tech. Rep. CAM09–68, 2009.

[18] X. Bresson and A. D. Szlam, "Total variation, Cheeger cuts," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 1039–1046.

[19] A. L. Bertozzi and A. Flenner, "Diffuse interface models on graphs for classification of high dimensional data," *Multiscale Model. Simul.*, vol. 10, no. 3, pp. 1090–1118, 2012.

[20] B. Merriman, J. K. Bence, and S. J. Osher, "Motion of multiple junctions: A level set approach," *J. Comput. Phys.*, vol. 112, no. 2, pp. 334–363, 1994.

[21] C. Garcia-Cardona, E. Merkurjev, A. Bertozzi, A. Flenner, and A. G. Percus, "Multiclass data segmentation using diffuse interface methods on graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1600–1613, Aug. 2014.

[22] E. Merkurjev, C. Garcia-Cardona, A. L. Bertozzi, A. Flenner, and A. G. Percus, "Diffuse interface methods for multiclass segmentation of high-dimensional data," *Appl. Math. Lett.*, vol. 33, pp. 29–34, Jul. 2014.

[23] H. Hu, J. Sunu, and A. L. Bertozzi, "Energy minimization methods in computer vision and pattern recognition," in *Proc. 10th Int. Conf. (EMMCVPR)*, 2015, pp. 13–16.

[24] E. Merkurjev, J. Sunu, and A. L. Bertozzi, "Graph MBO method for multiclass segmentation of hyperspectral stand-off detection video," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 689–693.

[25] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D, Nonlinear Phenomena*, vol. 60, nos. 1–4, pp. 259–268, 1992.

[26] T. Chan, S. Esedoglu, F. Park, and A. Yip, "Recent developments in total variation image restoration," *Math. Models Comput. Vis.*, vol. 17, p. 2, Dec. 2005.

[27] A. Buades, B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.

[28] G. Gilboa and S. Osher, "Nonlocal operators with applications to image processing," *Multiscale Model. Simul.*, vol. 7, no. 3, pp. 1005–1028, 2008.

[29] D. Zosso, G. Tran, and S. J. Osher, "Non-local Retinex—A unifying framework and beyond," *SIAM J. Imag. Sci.*, vol. 8, no. 2, pp. 787–826, 2015.

[30] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imag. Vis.*, vol. 40, no. 1, pp. 120–145, 2010.

[31] J. Zhang, W. Zhu, L. Wang, and N. Jiang, "Evaluation of similarity measure methods for hyperspectral remote sensing data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2012, pp. 4138–4141.

[32] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 209–226, Sep. 1977.

[33] R. A. Brown. (2014). "Building a balanced k-d tree in o (kn log n) time." [Online]. Available: https://arxiv.org/abs/1410.5420

[34] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proc. VISAPP*, vol. 2. Feb. 2009, pp. 331–340.

[35] M. Zhu and T. Chan, "An efficient primal-dual hybrid gradient algorithm for total variation image restoration," Univ. California, Los Angeles, Los Angeles, CA, USA, Tech. Rep. CAM08–34, 2008.

[36] M. Zhu, S. J. Wright, and T. F. Chan, "Duality-based algorithms for total-variation-regularized image restoration," *Comput. Optim. Appl.*, vol. 47, no. 3, pp. 377–400, 2008.

[37] E. Esser, X. Zhang, and T. F. Chan, "A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science," *SIAM J. Imag. Sci.*, vol. 3, no. 4, pp. 1015–1046, 2010.

[38] D. MacKay, "An example inference task: Clustering," in *Information Theory, Inference and Learning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2003, ch. 20, pp. 284–292.

[39] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Commun. Pure Appl. Math.*, vol. 42, no. 5, pp. 577–685, 1989.

[40] J. Kim and H. Park, "Fast nonnegative matrix factorization: An active-set-like method and comparisons," *SIAM J. Sci. Comput.*, vol. 33, no. 6, pp. 3261–3281, 2011.

[41] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA)*, Philadelphia, PA, USA, 2007, pp. 1027–1035. [Online]. Available: http://dl.acm.org/citation.cfm?id=1283383.1283494

[42] F. Zhu, Y. Wang, S. Xiang, B. Fan, and C. Pan, "Structured sparse method for hyperspectral unmixing," *ISPRS J. Photogramm. Remote Sens.*, vol. 88, pp. 101–118, Feb. 2014.

[43] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

Wei Zhu received the bachelor's degree in mathematics from Tsinghua University, Beijing, China, in 2012. He is now a fifth year graduate student at the Department of Mathematics, University of California at Los Angeles, Los Angeles, CA, USA. He is currently pursuing the Ph.D. degree under the supervision of Prof. S. Osher.

His research interests include image processing, optimization, topological data analysis, and machine learning.

Victoria Chayes is currently pursuing the bachelor's degree with Bard College, Annandale-On-Hudson, NY, USA, with a joint major in math and physics. She intends to pursue the Ph.D. degree in applied math immediately after graduating in 2017.

Her research interests include exoplanets, dynamical systems, complex analysis, and number theory.

Alexandre Tiard received the M.Sc. degree in engineering sciences from the Grenoble Institute of Technology, Grenoble, France, in 2016.

He is currently a graduate student at UCLA Vision Lab, University of California at Los Angeles, Los Angeles, CA, USA. His research interests are hyperspectral imagery and computer vision.

Stephanie Sanchez received the B.Sc. degree in applied mathematics with a specialization in computing from the University of California at Los Angeles, Los Angeles, CA, USA, in 2015. She is currently pursuing the master's degree with the Institute of Computational and Mathematical Engineering, Stanford University, Stanford, CA.

Her research interests are data mining, computer vision, and computer-generated imagery.

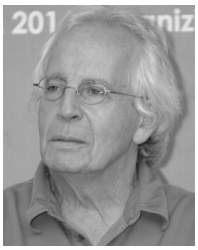
Devin Dahlberg received the B.S. degree in mathematics from the University of California at San Diego, La Jolla, CA, USA.

The aspect of research he enjoys the most is teamwork. His research interests include signal processing, financial mathematics, and education.

Andrea L. Bertozzi received the B.A., M.A., and Ph.D. degrees from Princeton University, Princeton, NJ, USA, in 1987, 1988, and 1991, respectively, all in mathematics.

She was on the Faculty of the University of Chicago at Illinois, Chicago, IL, USA, from 1991 to 1995 and Duke University, Durham, NC, USA, from 1995 to 2004. From 1995 to 1996, she was the Maria Goeppert-Mayer Distinguished Scholar with Argonne National Laboratory, Lemont, IL. Since 2003, she has been with the University of California at Los Angeles, Los Angeles, CA, USA, as a Professor of Mathematics, where she is currently the Director of Applied Mathematics. In 2012, she was appointed the Betsy Wood Knapp Chair for Innovation and Creativity. Her research interests include image inpainting, image segmentation, cooperative control of robotic vehicles, swarming, fluid interfaces, and crime modeling.

Dr. Bertozzi is currently a fellow of the Society for Industrial and Applied Mathematics and the American Mathematical Society. She is a Member of the American Physical Society. She was a recipient of the Sloan Foundation Research Fellowship, the Presidential Career Award for Scientists and Engineers, and the SIAM Kovalevsky Prize in 2009.



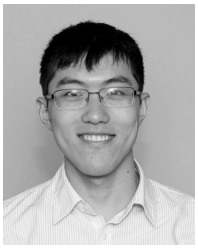
Stanley Osher received the M.S. and Ph.D. degrees in mathematics from the Courant Institute of Mathematical Sciences, New York University, New York, NY, USA, in 1964 and 1966, respectively.

He taught at SUNY Stony Brook, Stony Brook, NY, where he became a Professor in 1975. He was with the Faculty of the University of California Los Angeles (UCLA), Los Angeles, CA, USA, in 1977. He is currently a Professor of mathematics, computer science, chemical engineering, and electrical engineering with UCLA. He is also an Associate Director of the National Science Foundation-funded Institute for Pure and Applied Mathematics with UCLA. He co-founded three successful companies, each-based largely on his own (joint) research. His research interests include information science, which includes optimization, image processing, compressed sensing and machine learning, and applications of these techniques to the equations of physics, engineering, and elsewhere.

Dr. Osher is a Fellow of the Society for Industrial and Applied Mathematics (SIAM) and the American Mathematical Society. He was a recipient of the Carl Friedrich Gauss Prize from the International Mathematics Union in 2014, which is regarded as the highest prize in applied mathematics. He also gave the John von Neumann Lecture at the SIAM 2013 annual meeting. He has received numerous academic honors. He has been elected to the US National Academy of Science and the American Academy of Arts and Sciences. He gave a one hour plenary address at the 2010 International Conference of Mathematicians. He is a Thomson-Reuters highly cited researcher-among the top 1% from 2002 to 2012 in mathematics and computer science with an h-index of 100.

Dominique Zosso (S'06–M'11) received the M.Sc. degree in electrical and electronics engineering and the Ph.D. degree from the École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 2006 and 2011, respectively.

He was a Researcher with the Structural Bioinformatics Group, Swiss Institute of Bioinformatics and Biozentrum, University of Basel, Basel, Switzerland, from 2006 to 2007. He was a Research and Teaching Assistant with the Signal Processing Laboratory, EPFL, from 2007 to 2012. He was a Post-Doctoral Fellow and a CAM Assistant Adjunct Professor with the Department of Mathematics, University of California at Los Angeles, Los Angeles, CA, USA. He is currently an Assistant Professor in Applied Mathematics with Montana State University, Bozeman, MT, USA. His research interests include efficient algorithms, scientific computing, and numerical PDE, with a particular emphasis on variation and PDE problems stemming from inverse problems in image and data science.



Da Kuang received the bachelor's degree in computer science from Tsinghua University, Beijing, China, in 2009, and the Ph.D. degree in computational science and engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2014.

He is currently an Assistant Adjunct Professor with the Department of Mathematics, University of California at Los Angeles (UCLA), Los Angeles, CA, USA. His research interests are numerical methods for large-scale machine learning.