PRE-PROCESSING AND CLASSIFICATION OF HYPERSPECTRAL IMAGERY VIA SELECTIVE INPAINTING

Victoria Chayes[⋆] *Kevin Miller*[⋆] *Rasika Bhalerao*[⋆] *Jiajie Luo*[⋆] *Wei Zhu*[†]

Andrea L. Bertozzi[†] *Wenzhi Liao*[†⋆] *Stanley Osher*[†]

[⋆] California Research Training Program in Computational and Applied Mathematics
[†] University of California, Los Angeles, Department of Mathematics
[†⋆]Ghent University, Department of Telecommunications and Information Processing

ABSTRACT

We propose a semi-supervised algorithm for processing and classification of hyperspectral imagery. For initialization, we keep 20% of the data intact, and use Principal Component Analysis to discard voxels from noisier bands and pixels. Then, we use either an Accelerated Proximal Gradient algorithm (APGL), or a modified APGL algorithm with a penalty term for distance between inpainted pixels and endmembers (APGL_Hyp), on the initialized datacube to inpaint the missing data. APGL and APGL_Hyp are distinguished by performance on datasets with full pixels removed or extreme noise. This inpainting technique results in band-by-band datacube sharpening and removal of noise from individual spectral signatures. We can also classify the inpainted cube by assigning each pixel to its nearest endmember via Euclidean distance. We demonstrate improved accuracy in classification over data-mining techniques like k-means, unmixing techniques like Hierarchical Non-Negative Matrix Factorization, and graph-based methods like Non-Local Total Variation.

***Index Terms*—** Hyperspectral imagery (HSI), data preprocessing, image inpainting, image classification, image enhancement

1. INTRODUCTION

Hyperspectral imagery (HSI), wherein sensors capture data at hundreds of different wavelengths, has numerous applications in agriculture, environmental science, mineralogy, medical imaging, and surveillance, because of its fundamental ability to allow the identification of separate objects or materials that cannot be differentiated on sight [1, 2]. However, this branch of study presents particular difficulties: the sheer amount of data in an image can offer processing challenges, and data often are rife with noise or trade spatial for spectral resolution [3, 4].

To address these issues, pre-processing techniques are used to remove noisy voxels, dead pixels, and water bands, as well as to sharpen the image. Wavelet based methods and statistical methods are commonly used for HSI denoising [5, 6, 7]. [8] presents a multihypothesis prediction technique for HSI data preprocessing to achieve a denoised image with less intraclass variability and greater spatial smoothness. In [9], a fusion technique is introduced to combine the hyperspectral and LiDAR data to remove a large cloud shadow present during the acquisition of the HSI. Graph-regularized low-rank representation method is utilized in [10] to alleviate striping noise which is a ubiquitous phenomenon in HSI. However, pre-processing and classification are generally considered separate phases of hyperspectral analysis.

We propose a novel algorithm that can be used for both pre-processing and classification. Our algorithm resides in the relatively new field of hyperspectral impaiting [11, 12]. The heart of our algorithm lies in an accelerated proximal gradient scheme (APGL) for matrix completion to inpaint an initialized incomplete datacube [13]. This matrix completion method minimizes the rank of the inpainted matrix, which is especially suited for hyperspectral imagery under the assumption that most pixels should be linear combinations of the underlying pure materials. The result is a versatile and robust algorithm for hyperspectral noise reduction, pixel smoothing, and classification. We demonstrate a band-by-band sharpening in our inpainted datacube which corresponds to a lower signal-to-noise ratio (SNR) per cluster, as well as the denoising of individual pixel spectral signatures. The inpainted hyperspectral cube is also clean enough that we can classify each pixel according to the nearest endmember in Euclidean distance, which is not possible on the original, noisier cube; this final thresholding step turns our pre-processing algorithm into a classification scheme. In comparing APGL and our modified APGL method (APGL_Hyp) to other existing classification methods, we found that our algorithms performed strongly over a breadth of datasets. Furthermore, because of the inherent sharpening effect, our algorithms displayed a robustness to images with full pixel corruption. On the datasets worked with in this paper, our algorithm ran in under five minutes in its entirety, and often took under one minute for smaller datasets.

2. THE ALGORITHM

The algorithm we present has three main segments. First, the datacube is initialized through discarding noisy or unwanted voxels. This datacube is fed to the APGL inpainting scheme proposed in [13], or a modified version of this algorithm which takes into account distance between inpainted pixels and endmembers. APGL has been used in the past for Netflix user rating problems, but our application to hyperspectral inpainting is a novel application of the algorithm. Furthermore, our initialization step ensures that the algorithm inpaints over noise and not useful data. Second, we create a modified version of the APGL algorithm that takes prior knowledge of endmembers into account; this runs nearly identically to APGL in

This work was supported by NSF grants DMS-1045536, DMS-1118971, DMS-1417674, ONR grant N00014-16-1-2119, DOE grant DE-SC0013838, and Fund for Scientific Research in Flanders (FWO, Belgium) project G037115N

Fig. 1. Pixel Index Matrix (training set in red, discarded voxels in blue)

standard circumstances, but outperforms APGL on datasets with a significant amount of the original pixels missing. Our final step is to classify the inpainted hyperspectral image, and we can now do this via assigning each pixel to the cluster of its nearest endmember.

We assume pre-known endmembers. To discard the data in a systematic manner, we construct an $mn \times B$ pixel index matrix for the original $m \times n \times B$ hyperspectral datacube. Pixels are ordered in the index by distance to the closest endmember, and the top 20% are kept as a training set. We use Principle Component Analysis (PCA) to identify the bands of greatest variance and structure, introduce twenty [1] orderings of bands from the PCA coefficients , and assign one of these orderings randomly to each pixel to ensure no bands are deleted entirely. We then catalog the right bottom corner of this matrix for removal of voxels in the original hyperspectral image. In especially noisy cases, we discard entire pixels from the bottom of the index matrix. Generally 25-35% of the data is discarded in the initialization. For the inpainting stage, to capture some of the spatial properties and relations in the data, we represent pixels as patches of the full data cube. The initialization takes less than 5 seconds on all datasets in this paper.

The core stage of our algorithm uses an accelerated proximal gradient with line search technique for matrix completion to inpaint the hyperspectral datacube. The minimization problem APGL strives to solve is the following:

$$\arg\min_X ||\mathcal{A}(X) - b||_2^2 + \mu||X||_*. \tag{1}$$

Here X is the reconstruction of the hyperspectral image, $\mathcal{A}$ is a linear map which functions as a projection onto the index set of the known pixels, b is the observed partial datacube, and $||X||_*$ is the nuclear norm of X. In this case minimizing the nuclear norm is equivalent to minimizing the rank of X, [14] [15], which corresponds to the small number of endmembers in the hyperspectral image.

Equation (1) is a special case of the minimization problem of the form:

$$\min_{X \in \mathbb{R}^{m \times n}} F(X) := f(X) + P(X) \tag{2}$$

where $P : \mathbb{R}^{m \times n} \to (-\infty, \infty]$ is a proper, convex, lower semicontinuous function and f is convex smooth on an open subset of $\mathbb{R}^{m \times n}$ containing dom$P = \{X | P(X) < \infty\}$. In the Equation (1) case,

$$f(X) = \frac{1}{2}||\mathcal{A}(X) - b||_2^2 \ , \ \ P(X) = \mu||X||_*$$

and dom$P = \mathbb{R}^{m \times n}$. The basic numerical technique to solve this problem is to iteratively minimize with respect to $f(X)$ by gradient descent, then with respect to $P(X)$ by taking the proximal via an SVD thresholding shrink operator [16]. APGL accelerates convergence with a line search to solve for the optimal step τ_k at each iteration.

[1] twenty chosen to accelerate this initialization

We further modify the APGL algorithm to take knowledge of endmembers into account. APGL_Hyp solves:

$$\arg\min_{X \in \mathbb{R}^{m \times n}} \quad \frac{1}{2}||\mathcal{A}(X) - b||_2^2 + \mu||X||_* + \frac{\lambda}{2}||X - CX||_F^2. \tag{3}$$

where CX is a projection of each pixel onto the nearest endmember, with λ as a weighing parameter; the ideal value was determined to be $\lambda = 10^{-2}$ after testing. To retain convexity, we calculate CX_{old} for each iteration, and thus CX can be treated as a constant. The modified proximal gradient minimization is then:

$$f(X) = \frac{1}{2}||\mathcal{A}(X) - b||_2^2 + \frac{\lambda}{2}||X - CX||_F^2 \ \ , \ \ P(X) = \mu||X||_*,$$

and the same numerical methods can be used. We employed a k-d tree and used an approximate nearest neighbor search to calculate the projection step (CX) for the $O(\log n)$ complexity [17] [18], and further shortened the max comparisons to $e^4/6$, where e is the number of endmembers, to decrease computational time.

Initialization

1. Create pixel index matrix and training set, and remove desired percentage of voxels.
2. Create patch matrix of stripped datacube and k-d tree of endmember permutations.

Inpainting *The following algorithm is taken directly from [13] for the reader's convenience. Changes to the algorithm for APGL_Hyp are in boldface.*

3. Let $\mu > 0$ be a fixed regularization parameter, let $\eta \in (0, 1)$ be a given constant. Let $X^0 = X^1 = 0 \in \mathbb{R}^{m \times n}$, let $t^0 = t^{-1} = 1$ and let $\tau^0 = 1 + \boldsymbol{\lambda}$.
4. Repeat the following loop until convergence: for $k = 0, 1, 2, \ldots$, generate X^{k+1} according to the following iteration:
   (a) Set $Y^k = X^k + \frac{t^{k-1}-1}{t^k}(X^k - X^{k-1})$
   (b) **Calculate $\mathbf{CX^{k-1}}$.**
   (c) Set $\widehat{\tau_0} = \eta\tau^{k-1}$
   (d) For j=0, 1, 2, ...
   Set $G = Y^k - (\widehat{\tau_j})^{-1}[\mathcal{A}^*(\mathcal{A}(Y^k) - b) + \boldsymbol{\lambda}(\mathbf{X^k - CX^{k-1}})]$.
   Compute $S_{\widehat{\tau_j}}(G) = U\text{Diag}(\sigma - \mu/\widehat{\tau_j})_+ V^T$
   If $F(S_{\widehat{\tau_j}}(G)) \leq Q_{\widehat{\tau_j}}(S_{\widehat{\tau_j}}(G))$,
   Set $\tau^k = \widehat{\tau_j}$, break
   Else,
   Set $\widehat{\tau}_{j+1} = \min\{\eta^{-1}\widehat{\tau_j}, \tau^0\}$
   end
   end
   (e) Set $X^{k+1} = S_{\tau^k}(G)$
   (f) Set $t^{k+1} = \frac{1+\sqrt{1+4(t^k)^2}}{2}$.
5. Recreate the hyperspectral image from the patches.

Classification

6. Assign each pixel to the cluster of its nearest endmember.

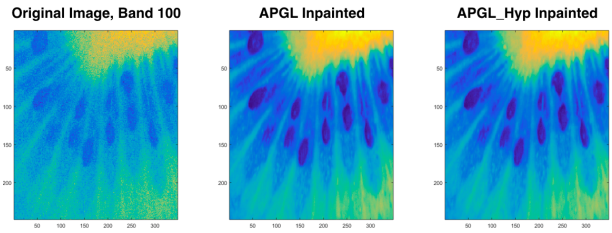

Fig. 2. Kiwi at 100th Band

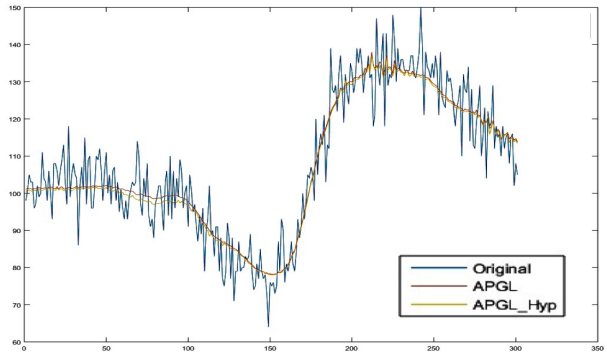

Fig. 3. Random Pixel from Kiwi

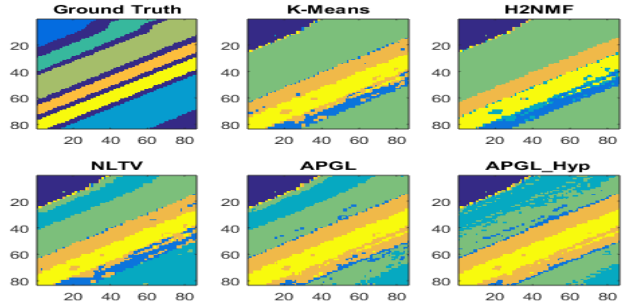

Fig. 4. Algorithms run on Original Salinas-A Dataset

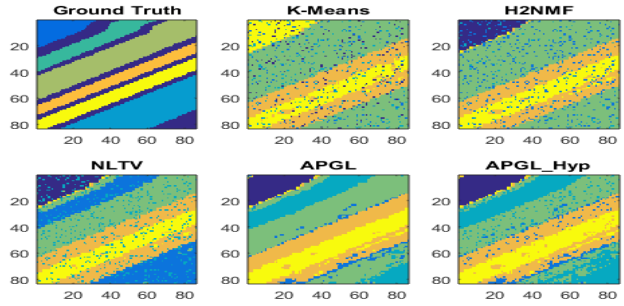

Fig. 5. Algorithms run on Salinas-A Dataset with 10% Pixels Replaced by Noise

3. PRE-PROCESSING RESULTS

3.1. Datacube Sharpening

The inpainting stage of our algorithm outputs cleaner band-by-band images. Visually, this is evident in Figure 1. To quantify this, we take the Signal-to-Noise Ratio (SNR) of the datacube. For each band, we take the weighted average of the mean over the standard deviation (μ/σ) of each cluster as determined by the ground truth, and then take the average of this ratio over all of the bands. For the Kiwi dataset, the SNR of the original datacube is 10.6185. The SNR of the datacube inpainted using APGL is 24.1549, and the SNR of the datacube inpainted using APGL_Hyp is 24.6162.

3.2. Signature Smoothing

Similar to band-by-band sharpening, our inpainted datacube displayed a smoothing of the pixel signatures while retaining fidelity to the pixel shape (Figure 2). We examined the total variation of each pixel divided by the mean of the pixel; the average of all of these values in the original kiwi datacube is 0.0774, but for APGL and APGL_Hyp it is 0.0080, nearly an order of magnitude lower. This indicates that our algorithm could be used as pre-processing on a noisy datacube before endmember extraction algorithms are employed, to ensure that the endmembers extracted are as clean as possible.

4. CLASSIFICATION RESULTS

For classification, we threshold each pixel into a cluster by assigning it to its nearest endmember. We compare APGL and APGL_Hyp to three algorithms chosen to represent a broad range of hyperspectral classification techniques. K-Means is a common data-mining technique used in machine learning, Hierarchical Non-Negative Matrix Factorization (H2NMF) [19] is a linear unmixing model relying on matrix factorization, and Non-Local Total Variation (NLTV) is a graph-based method which minimizes an energy functional [20]. Experiments unless otherwise noted were run on Dell Optiplex 9020 with Intel(R) Core i7 3.2 GHz running Windows 7 Professional with Service Pack 1, on MATLAB R2015a.

4.1. Urban Dataset

The first dataset tested on is the Urban dataset from HYDICE, which contains 307 $\times$ 307 pixels and 162 clean spectral bands. This dataset has the advantage of only having six classes of material: road, dirt, house, metal, tree, grass. We compared classifications to a ground truth originally taken from a structured sparse algorithm [21], and corrected pixel-by-pixel by eye with both the RGB and spectral signatures for reference. To preserve space, we only present quantitative results.

Table 1. Comparison of Numerical Results on the Urban Dataset

	Original Dataset		**10% Replaced by Noise**	
	Time	Accuracy	Time	Accuracy
K-Means	3.75 s	65.30%	8.17 s	69.34%
H2NMF	7.83 s	73.21%	14.17 s	63.27%
NLTV	260.33 s	81.60%	315.72 s	45.67%
APGL	149.09 s	82.95%	408.08 s	81.68%
APGL_Hyp	275.24 s	82.97%	880.20 s	82.04%

4.2. Salinas-A Dataset

Salinas-A scene was a small subscene of Salinas image, which was acquired by the AVIRIS sensor over Salinas Valley. It contains 86 $\times$ 83 pixels and 204 bands with six ground truth clusters. This was run on on Lenovo Yoga with Intel(R) Core i7 running Windows 10. The

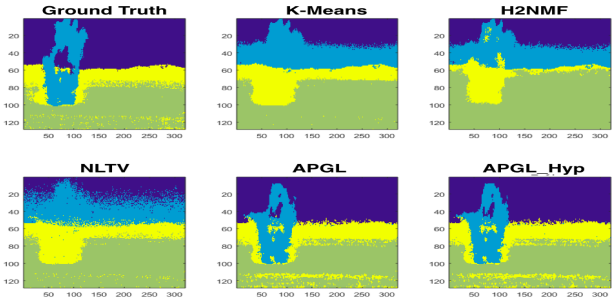

Fig. 6. Algorithms run on Original Chemical Plume Dataset

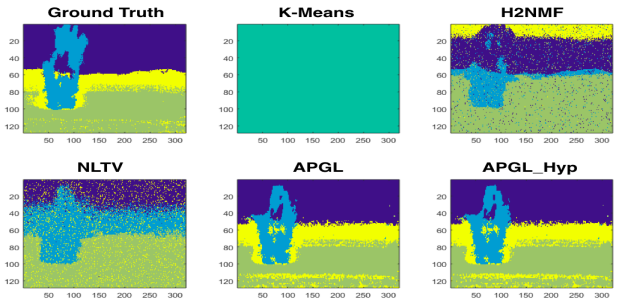

Fig. 7. Algorithms run on Chemical Plume Dataset with 10% Pixels Replaced by Noise

overall accuracies and visual clustering results are shown in Table 2 and Fig. 4.

Table 2. Comparison of Numerical Results on the Salinas-A Dataset

	Original Dataset		**10% Replaced by Noise**	
	Time	Accuracy	Time	Accuracy
K-Means	1.04 s	69.52%	4.60s	50.90%
H2NMF	2.41 s	70.08%	1.75 s	58.36%
NLTV	53.83 s	80.42%	54.23 s	71.02%
APGL	29.98 s	76.93%	33.57 s	76.78%
APGL_Hyp	65.95 s	69.60%	77.73 s	72.57%

4.3. Chemical Plume Dataset

The next dataset tested on is a chemical plume dataset, taken from frames of a hyperspectral video provided by the John Hopkins University Applied Physics Laboratory. These images were taken by long wave infrared spectrometers 2km from the release of the plume at an elevation of approximately 1300 feet, with dimensions 128×320 pixels, and 129 clean spectral bands [22]. It presents a unique challenge in terms of classification: as the image was taken in the infrared range, there is interference from heat vortexes in the desert; the plume itself is diffuse and invisible to the naked eye, and the spectral signatures of the atmosphere and the mountains filter through the plume; the dataset itself is noisy and no form of pre-processing has been run. We compared our results to a hand-made ground truth using [23, 24] as our guide and the first five principle bands to pick out clusters.

Table 3. Comparison of Numerical Results on the Chemical Plume Dataset

	Original Dataset		**10% Replaced by Noise**	
	Time	Accuracy	Time	Accuracy
K-Means	1.15 s	70.90%	N/A	N/A
H2NMF	2.24 s	63.42%	2.04 s	64.02%
NLTV	91.33 s	66.21%	102.57 s	61.56%
APGL	29.79 s	86.37%	29.59 s	85.76%
APGL_Hyp	47.66 s	86.40%	48.16 s	85.60%

5. ROBUSTNESS TO FULL PIXEL REMOVAL

APGL and APGL_Hyp are distinguished by their performance on datasets with full pixel removal. Both algorithms are resilient to this treatment, retaining nearly identical percentage accuracies on the

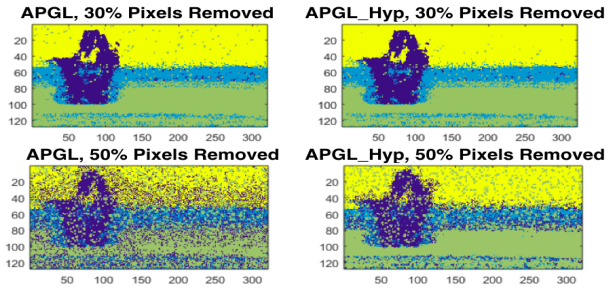

Fig. 8. APGL and APGL_Hyp run on Chemical Plume Dataset with Full Pixel Removal

chemical plume dataset all the way to 20% pixel removal. Greater removal of pixels resulted in the first major deviation between APGL and APGL_Hyp inpainting: consistently, APGL_Hyp classifies with a higher percent accuracy, as well as maintaining structure of the image. In examining pixel-by-pixel the signatures of the fully zeroed pictures, APGL leaves a number of pixels fully zero, while APGL_Hyp can partially reconstruct the original signature. Figures 8 shows results from both APGL and APGL_Hyp for 30% and 50% removal.

Table 4. Accuracy with Percent Full Pixel Removal (PFPR)

PFPR	APGL	APGL_Hyp
10%	85.99%	85.83%
20%	85.31%	85.36%
30%	81.85%	83.45%
50%	60.66%	69.80%
70%	48.61%	51.95%

6. CONCLUSION

In this paper we present the framework for a new type of dual hyperspectral sharpening and classification scheme. Using APGL or modified APGL inpainting on a specially initialized datacube, we can produce band-by-band sharpening, pixel smoothing, as well as classification that is highly robust to noise. While APGL and APGL_Hyp have slightly longer run-times than k-means and H2NMF, this is made up for by accuracy and resistance to noise, as well as the breadth of datasets it performs proficiently on. Finally, APGL_Hyp demonstrates the ability to reconstruct and classify images with significant percentages of the pixels missing entirely.

7. REFERENCES

[1] Chein-I Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*, Plenum Publishing Co., 2003.

[2] Hans Grahn and Paul Geladi, *Techniques and Applications of Hyperspectral Image Analysis*, John Wiley & Sons, Ltd, 2007.

[3] 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 Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 354–379, April 2012.

[4] Alex Chen, "The inpainting of hyperspectral images: a survey and adaptation to hyperspectral data," *Proc. SPIE*, vol. 8537, pp. 85371K–85371K–8, 2012.

[5] Hao Yang, Dongyan Zhang, Wenjiang Huang, Zhongling Gao, Xiaodong Yang, Cunjun Li, and Jihua Wang, *Application and Evaluation of Wavelet-Based Denoising Method in Hyperspectral Imagery Data*, pp. 461–469, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[6] G. Chen and S. E. Qian, "Denoising of hyperspectral imagery using principal component analysis and wavelet shrinkage," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 3, pp. 973–980, March 2011.

[7] B. Rasti, J. R. Sveinsson, M. O. Ulfarsson, and J. A. Benediktsson, "Hyperspectral image denoising using 3d wavelets," in *2012 IEEE International Geoscience and Remote Sensing Symposium*, July 2012, pp. 1349–1352.

[8] C. Chen, W. Li, E. W. Tramel, M. Cui, S. Prasad, and J. E. Fowler, "Spectral-spatial preprocessing using multihypothesis prediction for noise-robust hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 4, pp. 1047–1059, April 2014.

[9] C. Debes, A. Merentitis, R. Heremans, J. Hahn, N. Frangiadakis, T. van Kasteren, W. Liao, R. Bellens, A. Piurica, S. Gautama, W. Philips, S. Prasad, Q. Du, and F. Pacifici, "Hyperspectral and lidar data fusion: Outcome of the 2013 grss data fusion contest," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2405–2418, June 2014.

[10] X. Lu, Y. Wang, and Y. Yuan, "Graph-regularized low-rank representation for destriping of hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 7, pp. 4009–4018, July 2013.

[11] Alex Chen, "The inpainting of hyperspectral images: a survey and adaptation to hyperspectral data," in *SPIE Remote Sensing*. International Society for Optics and Photonics, 2012, pp. 85371K–85371K.

[12] Zuoqiang Shi, Wei Zhu, and Stanley Osher, "Low dimensional manifold model in hyperspectral image reconstruction," *arXiv preprint arXiv:1605.05652*, 2016.

[13] Kim-Chuan Toh and Sangwoon Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems," *Pacific Journal of Optimization*, vol. 6, no. 615-640, pp. 15, 2010.

[14] Emmanuel J. Candès and Benjamin Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[15] M. Fazel, H. Hindi, and S. Boyd, "Rank minimization and applications in system theory," in *American Control Conference, 2004. Proceedings of the 2004*, June 2004, vol. 4, pp. 3273–3278 vol.4.

[16] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[17] Jerome H. Friedman, Jon Louis Bentley, and Raphael Ari Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 209–226, Sept. 1977.

[18] Russell A Brown, "Building a balanced kd tree in o (kn log n) time," *arXiv preprint arXiv:1410.5420*, 2014.

[19] N. Gillis, D. Kuang, and H. Park, "Hierarchical clustering of hyperspectral images using rank-two nonnegative matrix factorization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 4, pp. 2066–2078, April 2015.

[20] Wei Zhu, Victoria Chayes, Alexandre Tiard, Stephanie Sanchez, Devin Dahlberg, Da Kuang, Andrea L Bertozzi, Stanley Osher, and Dominique Zosso, "Unsupervised classification in hyperspectral imagery with nonlocal total variation and primal-dual hybrid gradient algorithm," *accepted in IEEE Trans. Geo. Remote Sens.*, 2016.

[21] Feiyun Zhu, Ying Wang, Shiming Xiang, Bin Fan, and Chunhong Pan, "Structured sparse method for hyperspectral unmixing," {*ISPRS*} *Journal of Photogrammetry and Remote Sensing*, vol. 88, pp. 101 – 118, 2014.

[22] Torin Gerhart, Justin Sunu, Lauren Lieu, Ekaterina Merkurjev, Jen-Mei Chang, Jrme Gilles, and Andrea L. Bertozzi, "Detection and tracking of gas plumes in lwir hyperspectral video sequence data," 2013, vol. 8743, pp. 87430J–87430J–14.

[23] E. Merkurjev, J. Sunu, and A. L. Bertozzi, "Graph mbo method for multiclass segmentation of hyperspectral stand-off detection video," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 689–693.

[24] Huiyi Hu, Justin Sunu, and Andrea L. Bertozzi, *Multi-class Graph Mumford-Shah Model for Plume Detection Using the MBO scheme*, pp. 209–222, Springer International Publishing, Cham, 2015.