AFRL-OSR-VA-TR-2013-0129

Compressive Hyperspectral Imaging and Anomaly Detection

Pradeep Thiyanarantnam, Stanley Osher, Susan Chen, Wotao Yin and Kevin Kelly

Level Set Systems

March 2013
Final Report

DISTRIBUTION A: Approved for public release.

AIR FORCE RESEARCH LABORATORY
AF OFFICE OF SCIENTIFIC RESEARCH (AFOSR)
ARLINGTON, VIRGINIA 22203
AIR FORCE MATERIEL COMMAND

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to the Department of Defense, Executive Services and Communications Directorate (0704-0188). Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
17/07/2013	FINAL	01-05-2010 - 30-04-2012

4. TITLE AND SUBTITLE

Compressive Hyperspectral Imaging and Anomaly Detection

5a. CONTRACT NUMBER

FA9550-10-C-0108

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Pradeep Thiyanarantnam, Stanley Osher, Susan Chen, Wotao Yin and Kevin Kelly

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Level Set Systems
1058 Embury St
Pacific Palisades, CA 90272

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFOSR
875 N Randolph St
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

AFRL-OSR-VA-TR-2013-0129

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution A: Approved for public release

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We have developed and applied successfully new algorithms for hyperspectral imagery. These include compressive sensing, anomaly detection, target detection, endmember detection, unmixing and change detection. These were tested on data provided by AFRL with good results, including change detection under different lightning conditions. Ideas involved Bregman iteration applied to L1 and total variation based optimizations were used and also successfully applied to subsampled data. A nonnegative matrix factorization and completion algorithm was introduced which allows the reconstruction of partially observed or corrupted hyperspectral data. A surprising spinoff is sparse reconstruction of offshore oil spills based on multispectral measurements.

15. SUBJECT TERMS

HYPERSPECTRAL IMAGERY, COMPRESSIVE, ANOMALY, TARGET, ENDIMEMBER, UNMIXING, CHANGE DETECTION, L1, TOTAL VARIATION, MATRIX FACTORIZATION

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Stanley Osher
U	U	U	U		19b. TELEPHONE NUMBER *(Include area code)* 310-573-9339

Reset

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

STTR Report #9 (Final Report)
STTR Topic # AF08-B724
Proposal #F08B-724-0062
Contract #FA9550-10-C-0108
Compressive Hyperspectral Imaging and Anomaly Detection

Level Set Systems, Rice CAAM Group, Rice Kelly Lab

May 1, 2010 to April 30, 2012

1 Summary

In this final report, we summarize the algorithms developed and some results for hyperspectral imagery over the period of the contract. It includes target detection, endmember detection, and change detection. The change detection algorithm has been tested on a newer data set provided by the AFRL, with good results. A new total variation(TV) based method for the unmixing of hyperspectral data through compressed sensing is introduced. It allows unmixing directly from the compressed sensing data without first reconstructing the entire hyperspectral cube. A nonnegative matrix factorization and completion algorithm is presented which allows the reconstruction of partially observed or corrupted hyperspectral data. As a spinoff of our hyperspectral effort we propose a novel video rate IR multispectral imaging system which achieves higher accuracy at lower cost for offshore oil spill sensing.

2 Target Detection

We summarize our target detection algorithm as follows. Given a hyperspectral image (HSI) with NxN pixels and M spectral bands, we wish to locate the positions of pixels that correspond to a given spectral signature f, which also has M spectral bands. We rearrange A as an $M \times N^2$ matrix, where generally $M < N^2$. The signals a_i are the columns of this spectral matrix A and correspond to each pixel in the image.

Our goal is to find $u \in R^{N^2}$ by solving the constrained minimization problem

$$\begin{aligned} &u = argmin\, |u|_1 \;\; s.t. \;\; \|Au - f\| < \delta, \\ &u \geq 0, \end{aligned} \tag{1}$$

where δ is a measure of the noise in the system. This is an extremely underdetermined system, but we seek solutions u whose components u_i ideally vanish when the i^{th} column a_i is not a match for f, and $u_i > 0$ when a_i is a match.

To solve this, we apply Bregman iteration [1, 2], by approximately solving a sequence of unconstrained minimization problems.

$$u^{n+1} = argmin \left(\mu |u|_1 + \frac{\lambda}{2} \|Au - f^n\|^2 \right) \tag{2a}$$

$$f^n = f^{n-1} + f - Au^{n-1} \tag{2b}$$

for $n = 1, 2, ...$, with $u^0 = 0$. The constant λ is usually chosen around $\lambda = \frac{100}{\|A^T A\|}$. $\|Au - f^n\|^2$ monotonically decreases to zero and u^n converges very quickly to a solution of (1) with $\delta = 0$, see [2, 1]. The sparsity of the solution, which is equivalent to the sensitivity of the matching, is controlled by the parameter μ.

It now becomes a matter of solving (2a) and (1) efficiently. We propose the recently developed Split Bregman[3] algorithm. The idea behind this is quite simple. There are two simple minimization problems to be solved. To solve

$$argmin\left(\mu\left|u\right|_1+\frac{1}{2}\left\|u-f\right\|^2\right) \tag{3}$$

we have the following well known shrinkage formula

$$u_i=shrink(f_i,\mu)=\begin{cases} f_i-\mu & \text{if } f_i>\mu \\ 0 & \text{if } |f_i|\le\mu \\ f_i+\mu & \text{if } f_i<-\mu \end{cases} \tag{4}$$

Moreover, if we add the constraint that $u_i\ge 0$, then

$$u_i=shrink^+(f_i,\mu)=\begin{cases} f_i-\mu & \text{if } f_i>\mu \\ 0 & \text{if } f_i\le\mu \end{cases} \tag{5}$$

To solve

$$argmin\left(\frac{\lambda}{2}\left\|Au-f\right\|^2+\frac{1}{2}\left\|d-u\right\|^2\right) \tag{6}$$

for a fixed vector d, we have

$$u=\left(\lambda A^TA+I\right)^{-1}\left(\lambda A^Tf+d\right) \tag{7}$$

The idea behind split Bregman is as follows: We replace the problem (2a) by a sequence of approximations generated by Bregman iteration:

$$\begin{cases} \left(d^{k+1},U^{k+1}\right)=argmin\mu\left|d\right|_1+\frac{\lambda}{2}\left\|Au-f\right\|^2+\frac{1}{2}\left\|d-U-b^k\right\|^2 \\ b^k=b^{k-1}+U^k-d^{k-1} \end{cases} \tag{8}$$

The steps used in the solution for (8a) and (8b) involve splitting

$$\begin{cases} U^{k+1}=\left(\lambda A^TA+I\right)^{-1}\left(\lambda A^Tf^n-b^k+d^k\right) \\ d^{k+1}=shrink\left(U^{k+1}+b^{k+1},\mu\right) \end{cases} \tag{9}$$

U^k approaches u^{n+1} monotonically, $\left\|U^k-u^n\right\|\searrow 0$, and of course, $\left\|d-U^k\right\|\searrow 0$. Thus we use an inner iteration to obtain the sequence U^k,d^k, which approximates the updated u. We then update using (2) to get f^{n+1} and repeat the inner iteration to get u^{n+2}. This procedure is very efficient. The number of of inner iterations needed is problem dependent, but usually between 5 and 10.

2.1 Incorporating Spatial Information in the Target Detection Process

Our L1 target detection method described above, as well as most other methods, does not directly take into account the spatial relationship between pixels. Once the hyperspectral data cube has been converted to a matrix, the relationship between the spectral pixels of the HSI is ignored. A more sophisticated model would take this spatial information into account. We will describe a simple, yet effective method of using the spatial information to increase the accuracy of target detection. The idea is to apply TV denoising [4] to the output of the L1 target detection algorithm. It is reasonable to assume that in a natural scene, a target is composed of several neighboring pixels. Therefore, it seems likely that a single pixel identified as a target, but surrounded by non-target pixels is a false alarm. Conversely, a pixel identified as a non-target(or with a small return value) surrounded by target pixels is likely missed target pixel. If spatial information is used, such errors can be reduced or eliminated. This is especially true for the L1 target detection output. Due to the sparsity of the method, most non-targets have a zero value, and isolated false alarm pixels are usually eliminated by the TV denoising algorithm.

2.1.1 TV Denoising

Here we briefly describe the total variation denoising model[4] we use in the above. Given an image $I \in R^2$, we solve the following L1 minimization problem to denoise the image:

$$min_v \left|\nabla v\right|_1 + \frac{\gamma}{2}\left\|v - I\right\|_2^2,$$

where γ is an adjustable fidelity parameter. The Split Bregman method as described in [3] is used to solve this minimization problem very efficiently.

2.2 Experimental Results

To test our target detection algorithm, we used data that was provided by the AFRL. This data set has 300x600 spatial pixels, 124 spectral bands and consists of 4 panels of different material or paint, with grass in the foreground and trees in the background. In order to compute ROC curves, the ground truth must be known. We chose the panels as the 4 targets, but since the ground truth was not provided, we slightly modified the data set. The boundaries of a panel were replaced with random background pixels, leaving the center, which we could say for certain was the target and the ground truth. Figure 1 shows modified RGB of the data. We tested the L1, L1 with TV, Matched Filter, and ACE on the modified data set to obtain ROC curves for each of the 4 targets. Figures 2-5 show the results. All of the methods were able to detect the last 3 panels, with no false alarms(using the proper thresholding), however on the first panel, L1 with TV performed the best. We scaled the output of the L1 target detection to [0,255] before applying the TV denoising. This was done so a consistent fidelity parameter could be used. All of the target detection results below used the value $\gamma = 0.05$.

To test robustness to noise, we also added some random noise to the data and performed the same tests with results shown in Figures 6-9. On the first panel, the L1 and L1 with TV methods performed much better than Matched Filter and ACE. However, on the second panel, Matched Filter and ACE had better performance than either of the L1 methods. The L1 with TV edged out the others for the 3rd panel in terms of detecting all of the target pixels with the fewest false alarms. L1 with TV was able to detect all of the target pixels for the 4th panel with no false alarms.

3 Endmember Detection

Our endmember detection algorithm is based on the work done in [5], and we summarize it as follows. Let I be an HSI with NxN pixels and M spectral bands. We set A as an $M \times N^2$ matrix whose columns are the spectral vectors of I. We want to find $U \in R^{N^2 \times N^2}$, referred to as the abundance matrix, such that $AU \approx A$. In order to minimize the number of endmembers, we need to minimize the number of nonzero rows of U. Additionally, to make each pixel a sparse nonnegative linear combination of the endmembers, we need to minimize the number of nonzero entries in each row, i.e. make the matrix sparse. In practical terms, the matrix U is extremely large and solving for it would nearly impossible. In order to make the problem manageable, we need to reduce the number of candidate endmembers by choosing a subset of the columns of A to make a much smaller matrix A_s of size $M \times P$, where $P << N^2$. We then try find $U_s \in R^{P \times P}$ such that $A_s U_s \approx A_s$. One method of obtaining A_s is to cluster the pixels of A and choose a representative subset from each cluster. The problem is then stated as

$$\min_{U_s \geq 0} \zeta \sum_i \max_j (U_{s\ i,j}) + \langle \sigma c_w, U_s \rangle + \frac{\beta}{2}\left\|(A_s U_s - U_s) C_w\right\|_F^2. \tag{10}$$

The data fidelity term is $\frac{\beta}{2}\left\|(A_s U_s - U_s) C_w\right\|_F^2$, $\left\|\cdot\right\|_F$ is the Frobenius norm, and C_w is a diagonal matrix used to weight the columns of $(A_s U_s - U_s)$. Since A_s is composed of representative members of clusters, we want give those from larger clusters more weight. The term $\zeta \sum_i \max_j (U_{s\ i,j})$ promotes the rows of U_s to be zero, and $\langle \sigma c_w, U_s \rangle$ encourages the sparsity of U_s. σ is a $P \times P$ matrix of weights and the weights are chosen as $\sigma_{i,j} = \nu\left(1 - e^{\frac{-[1-(A^T X_s)_{i,j}]^2}{2h^2}}\right)$. This choice of weights promotes the sparsity of U_s. We solve (10) by using split Bregman algorithm which is related to the alternating direction method of multipiers(ADMM[6]).

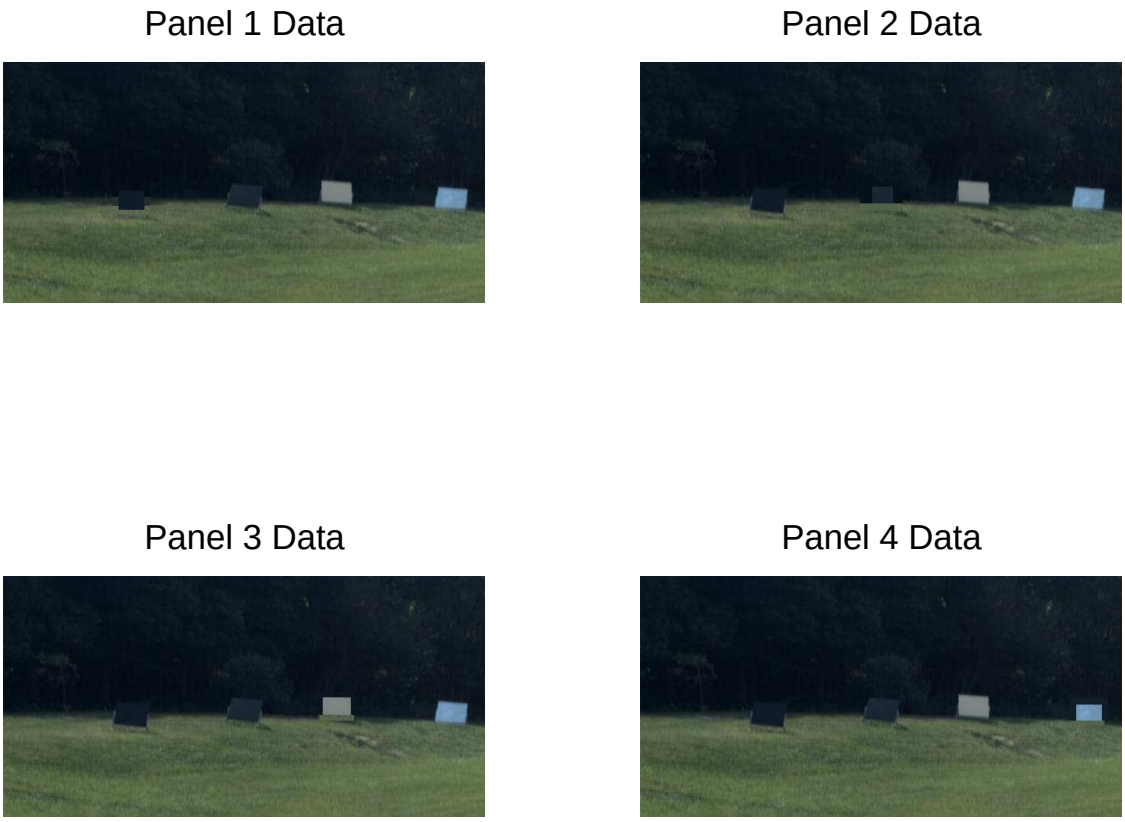
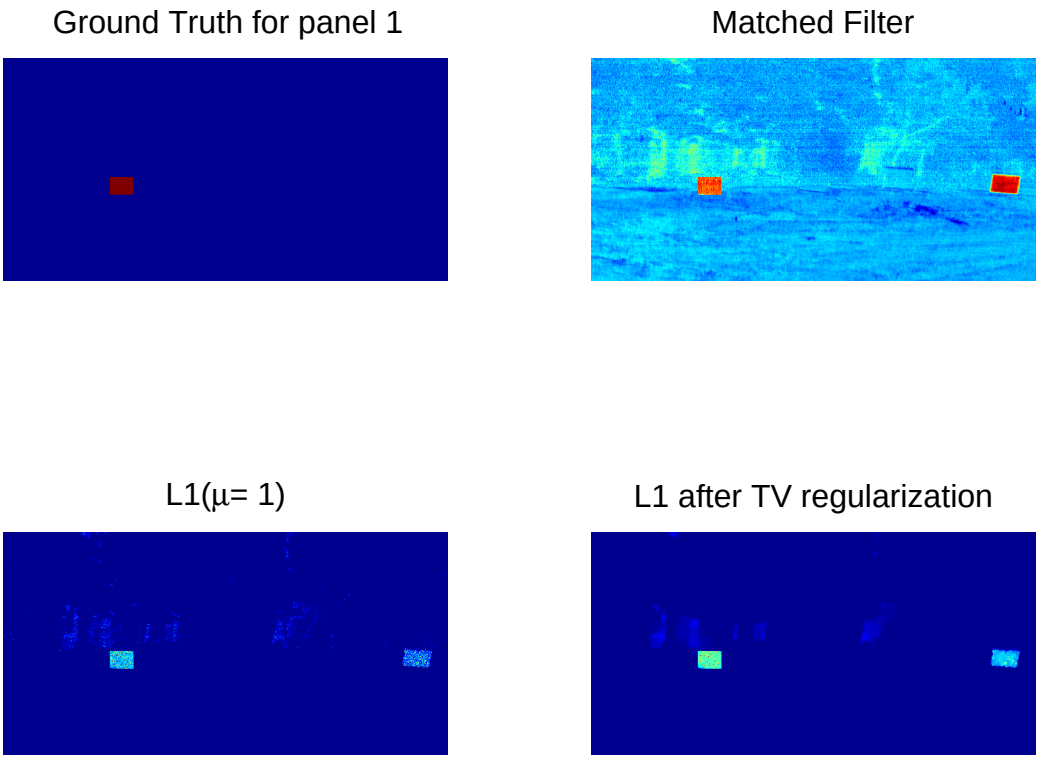


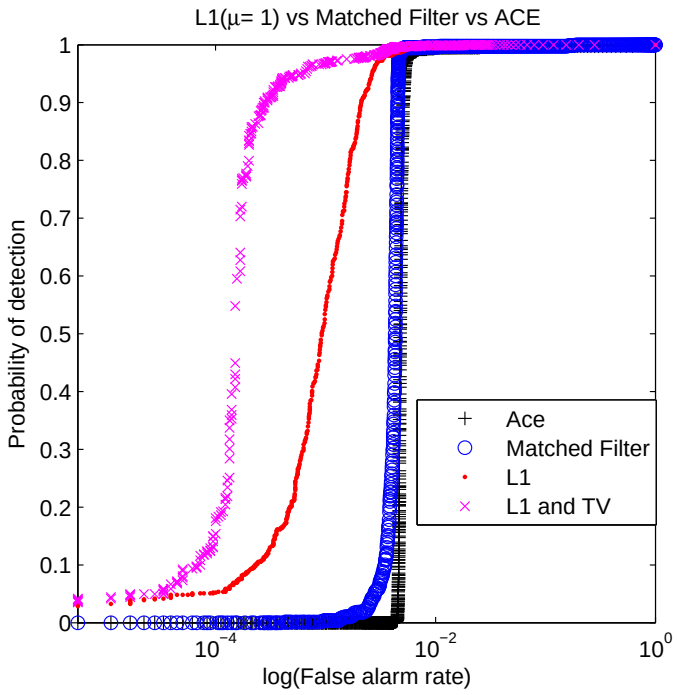
Figure 1: RGB of the modified scenes for the 4 panels

3.1 Experimental Results

The test HSI is the URBAN data, which has 307x307 spatial pixels and 163 spectral bands. Figure 10 shows a color composite of the scene. Six endmembers were computed, using parameter values $\zeta = 2$ and $\beta = 250$. Once the endmembers have been computed, we unmix the HSI to get abundance values for material. We can "cluster" the data by setting the material with the largest abundance value as the material for the pixel. Figure 11 shows a composite of the 6 clusters and Figure 12 shows the clusters plotted individually. By comparing to the color image of the scene, the detected clusters appear to represent asphalt, trees, grass, roofing material, dirt and some other type of vegetation.

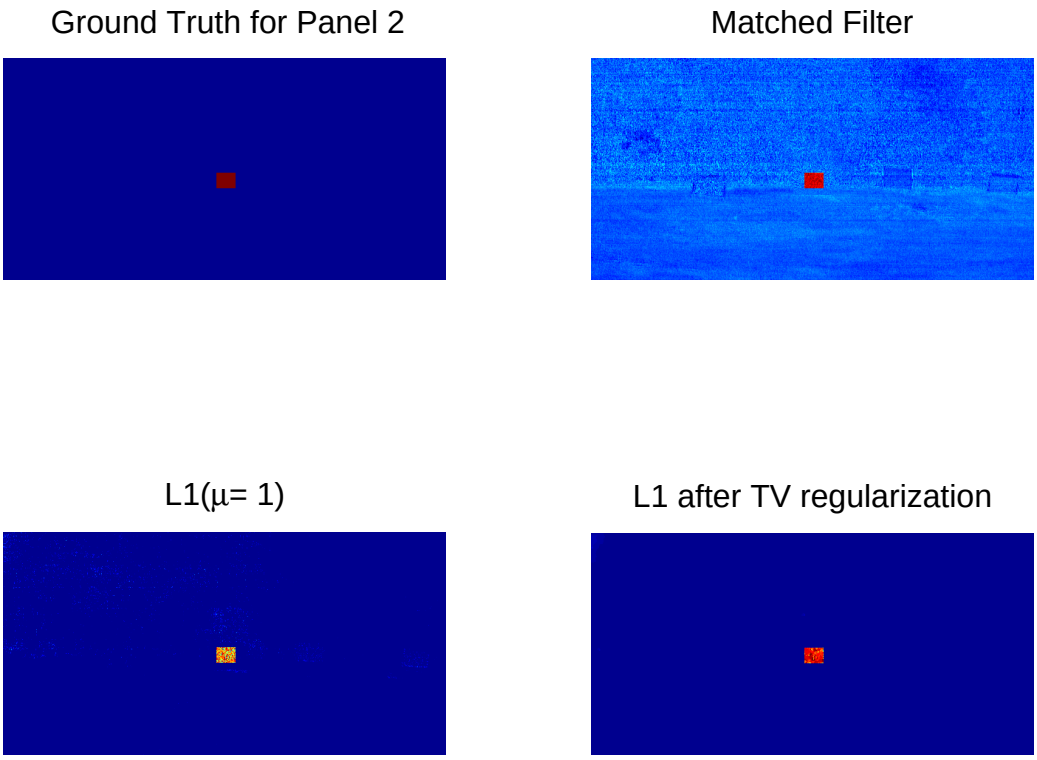


(a) The ground truth and outputs of Matched Filter, L1, and L1 after TV.

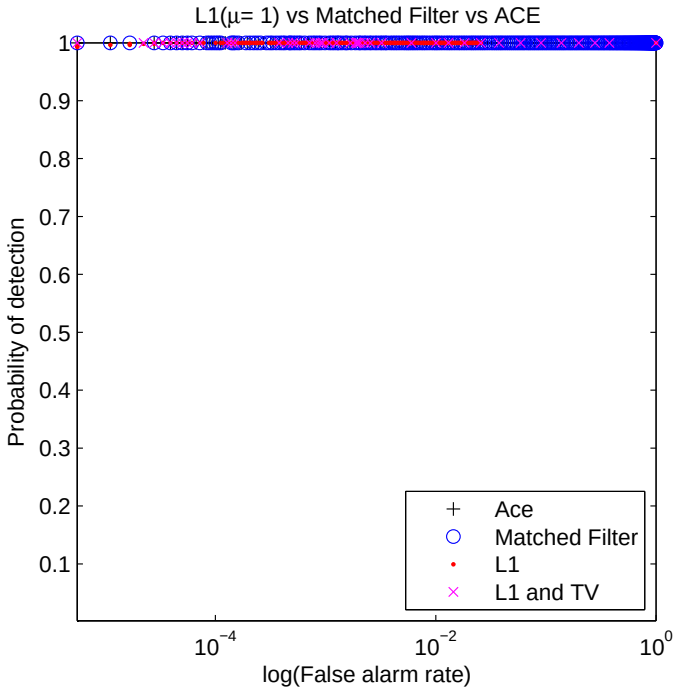


(b) ROC curves

Figure 2: Detecting the 1st panel.

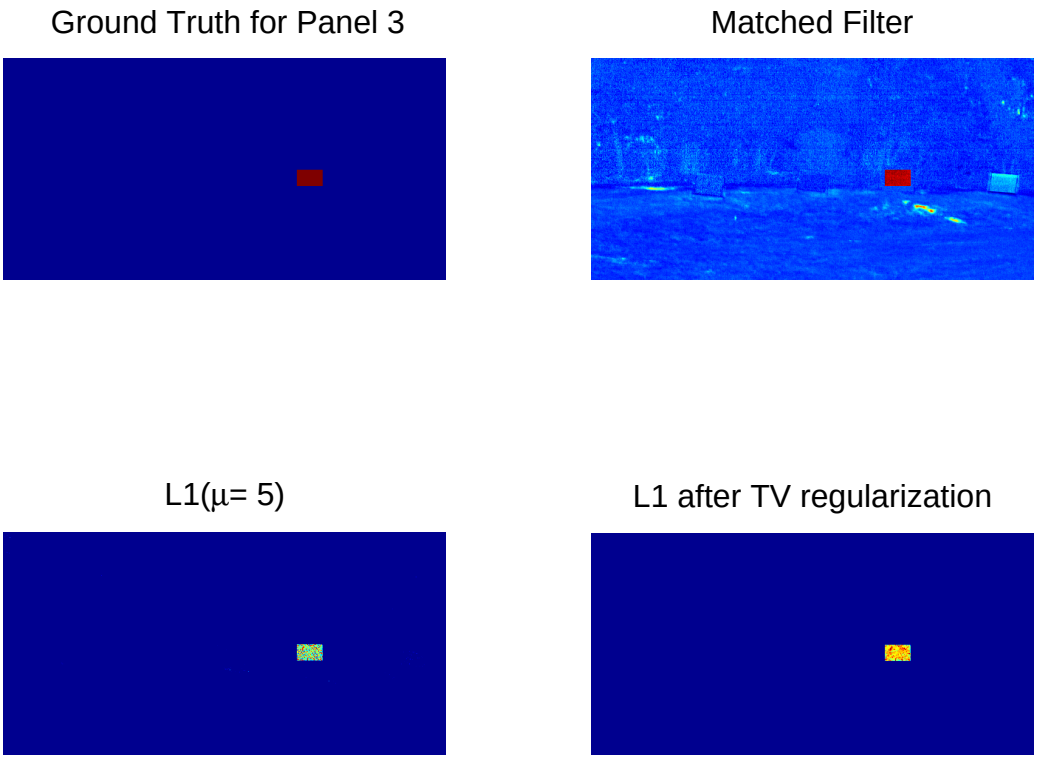


(a) The ground truth and outputs of Matched Filter, L1, and L1 after TV.

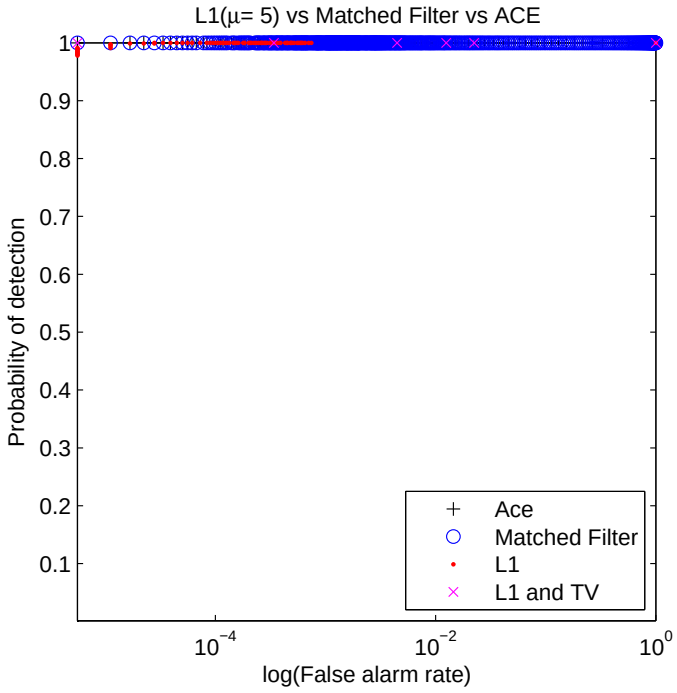


(b) ROC curves

Figure 3: Detecting the 2nd panel.

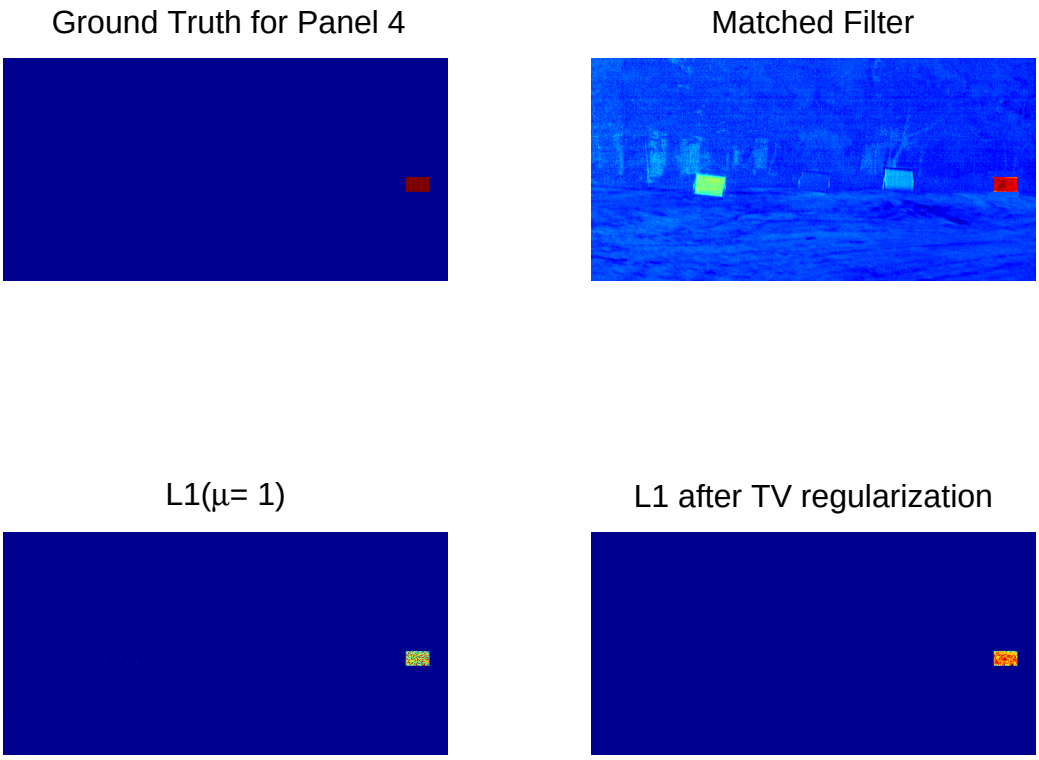


(a) The ground truth and outputs of Matched Filter, L1, and L1 after TV.

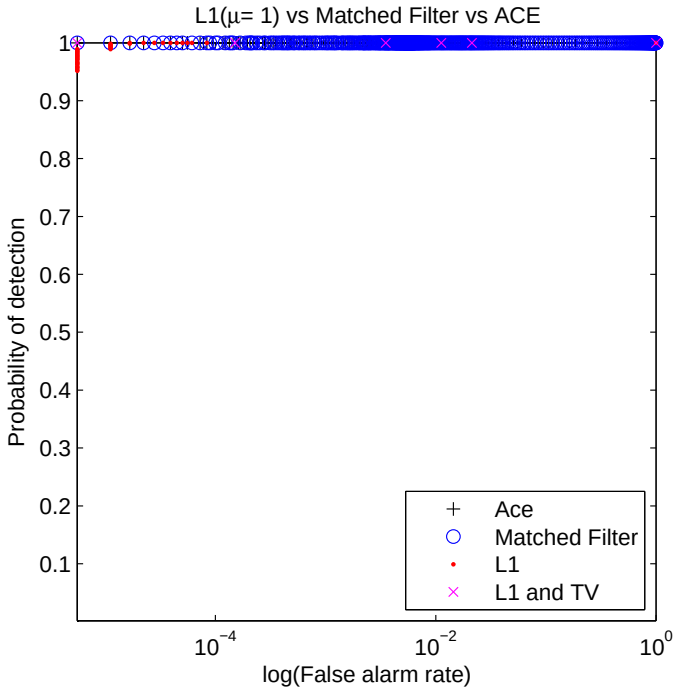


(b) ROC curves

Figure 4: Detecting the 3rd.

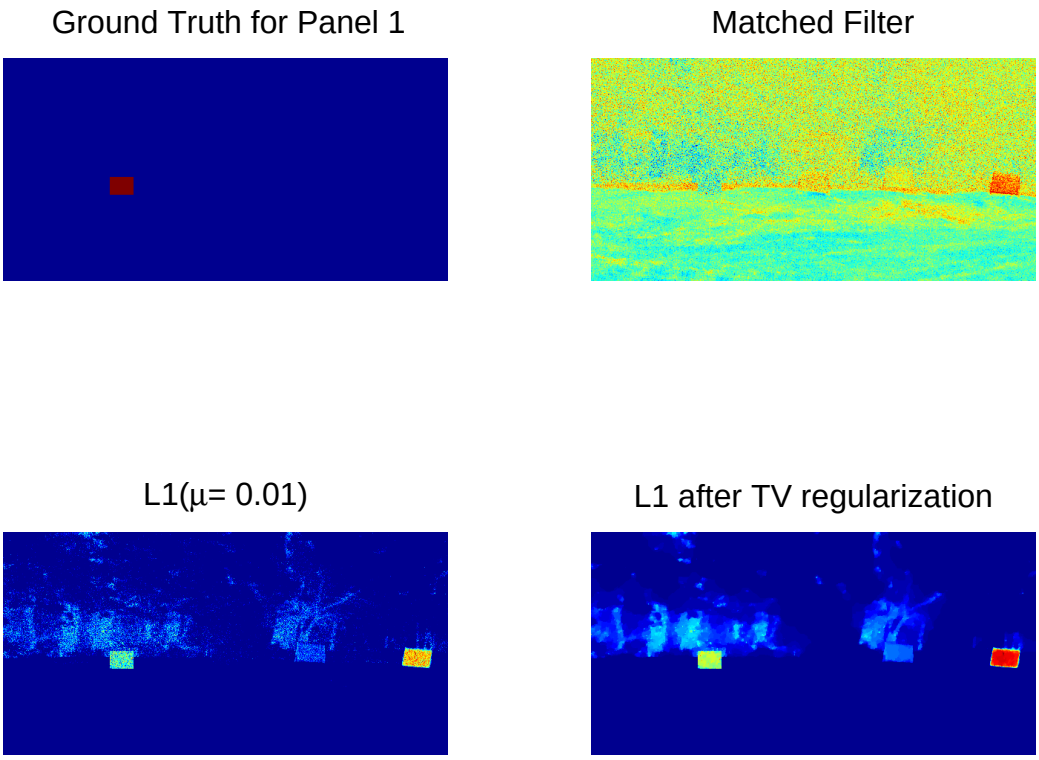


(a) The ground truth and outputs of Matched Filter, L1, and L1 after TV.

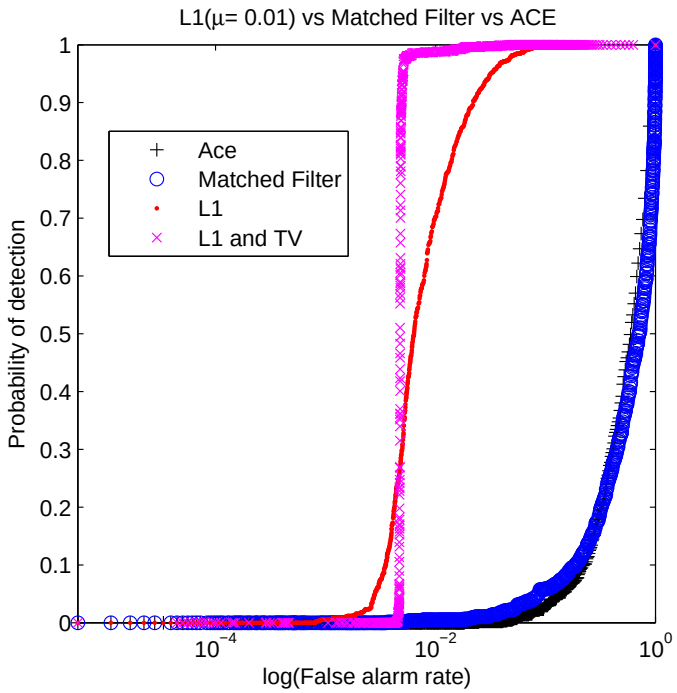


(b) ROC curves

Figure 5: Detecting the 4th panel.

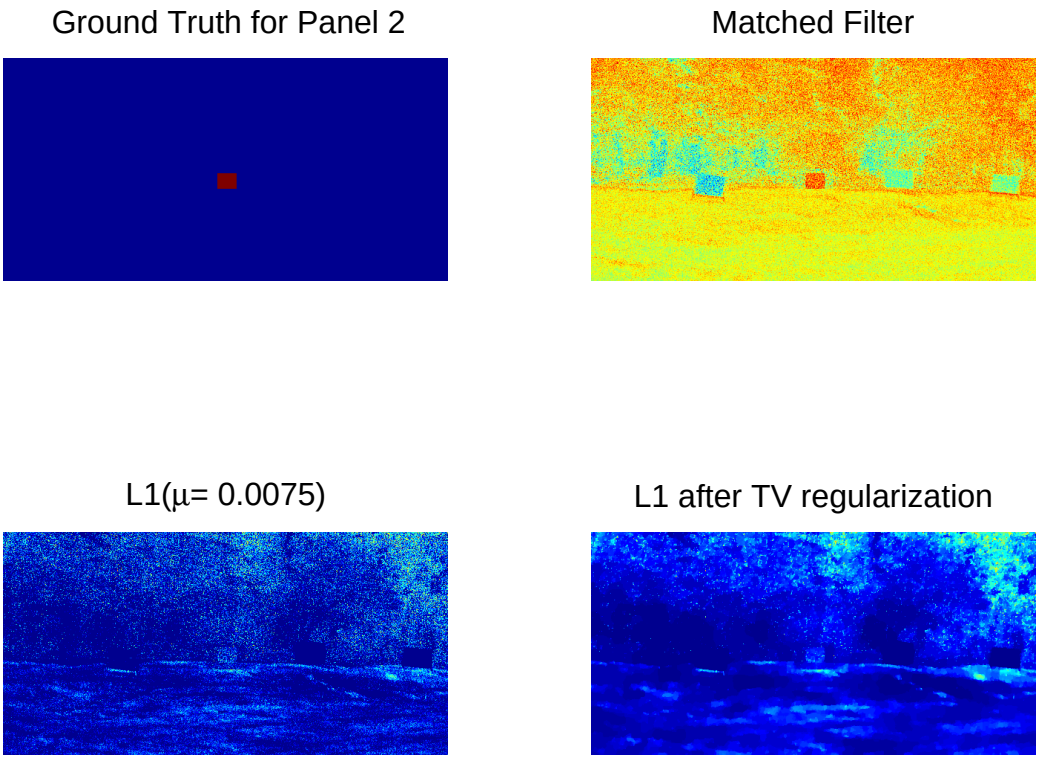


(a) The output of the 4 methods

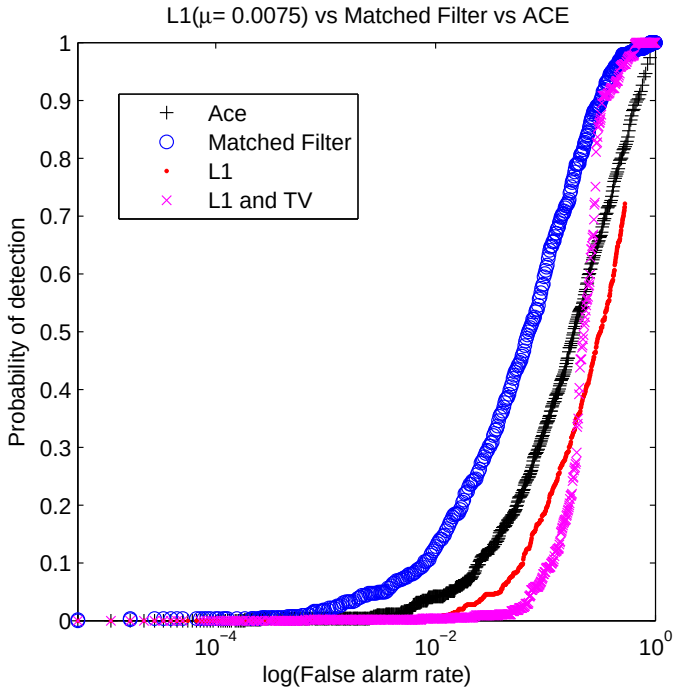


(b) ROC curves

Figure 6: Detecting the 1st panel on data with noise.

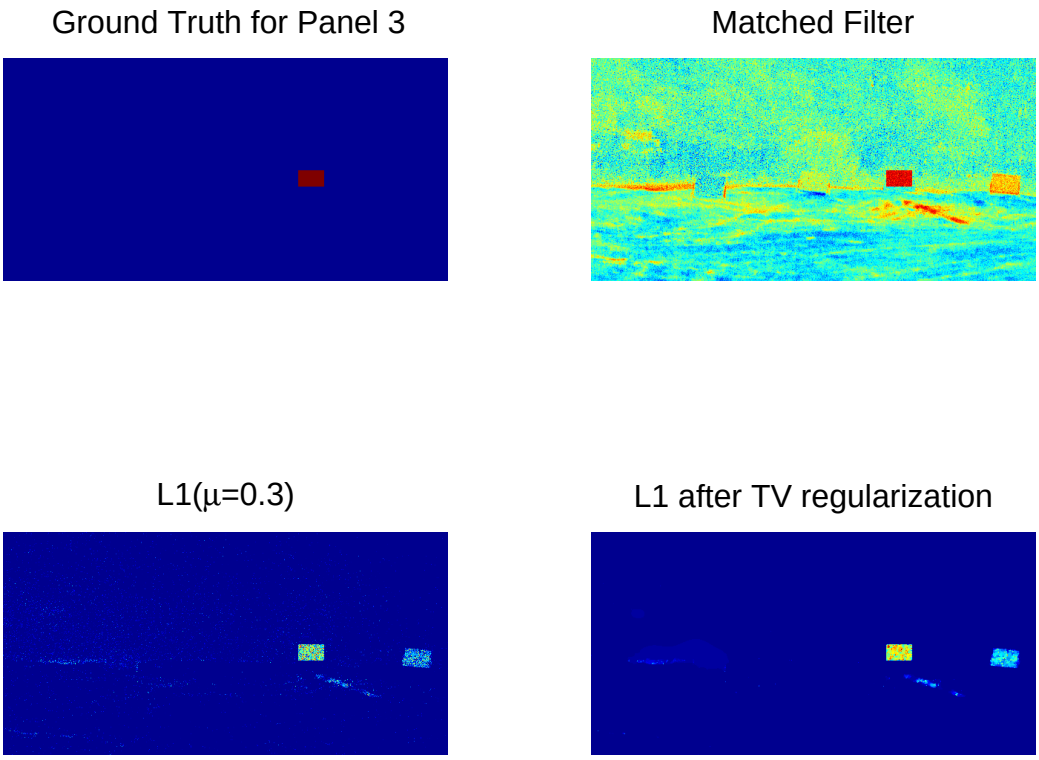


(a) The output of the 4 methods

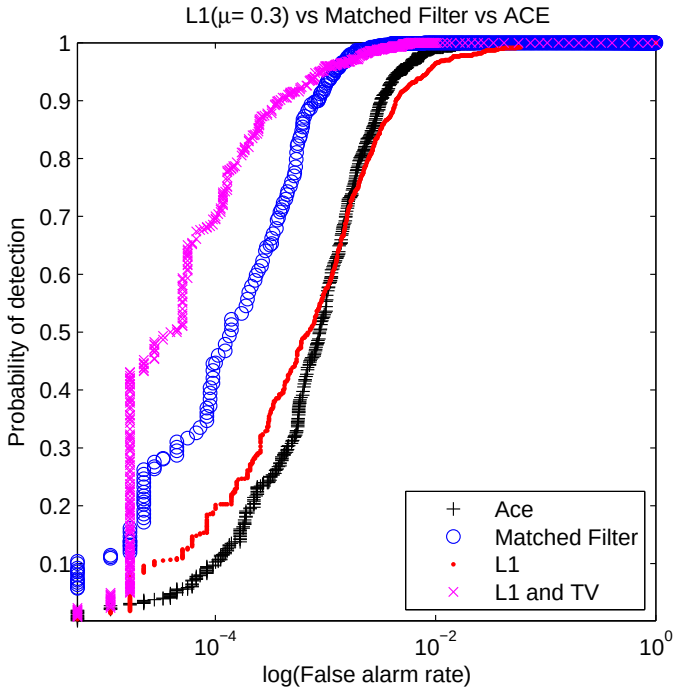


(b) ROC curves

Figure 7: Detecting the 2nd panel on data with noise.

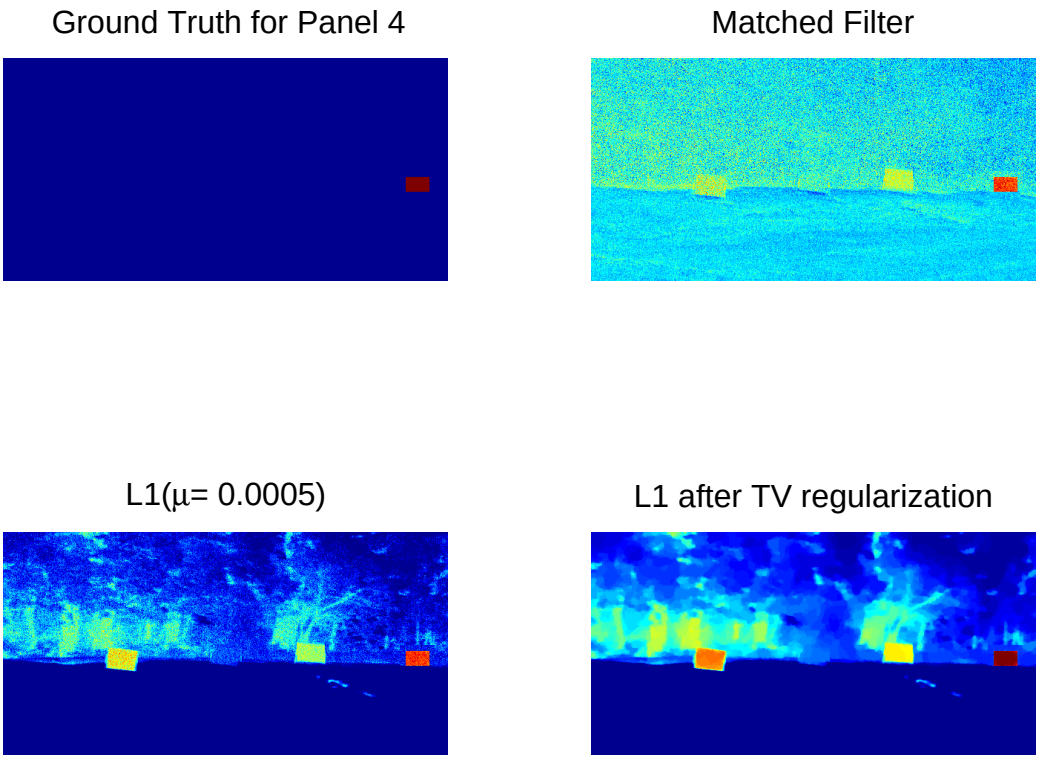


(a) The output of the 4 methods

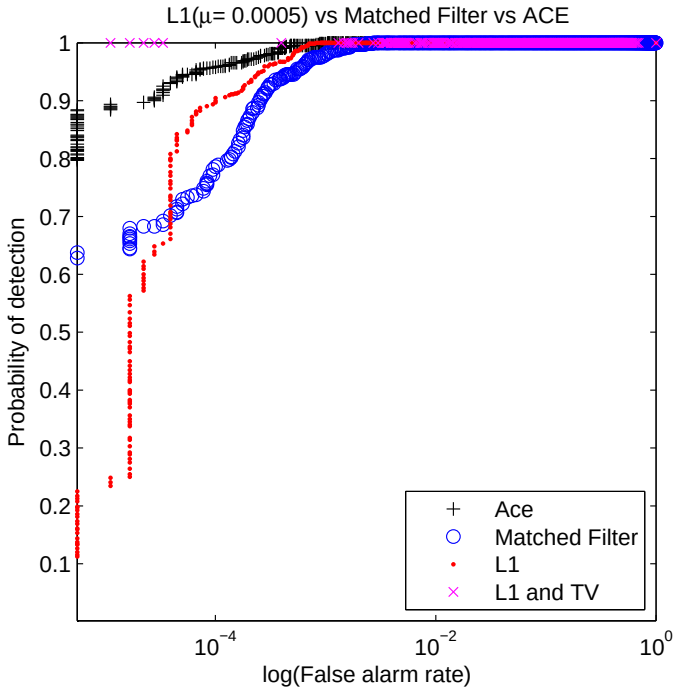


(b) ROC curves

Figure 8: Detecting the 3rd panel on data with noise.



(a) The output of the 4 methods



(b) ROC curves

Figure 9: Detecting the 4th panel on data with noise.

Figure 10: Color composite of the scene for the URBAN data.

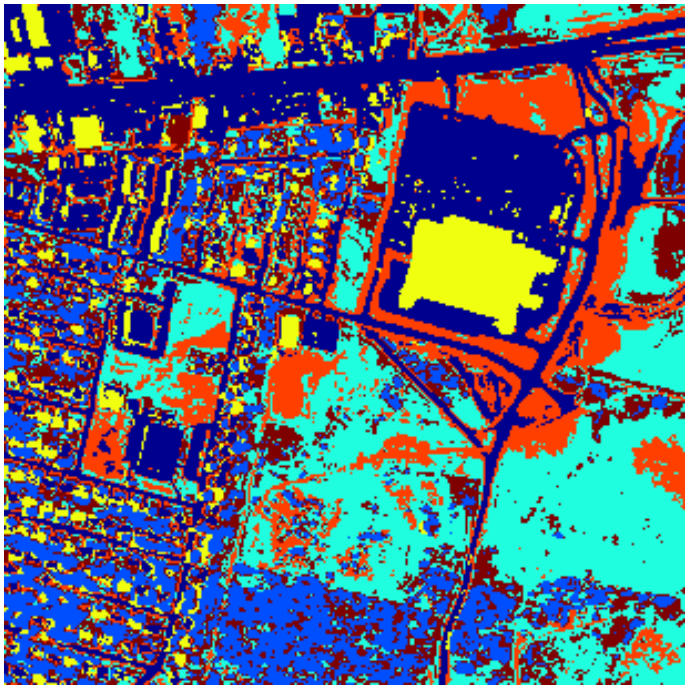


Figure 11: Clustering of the 6 materials.

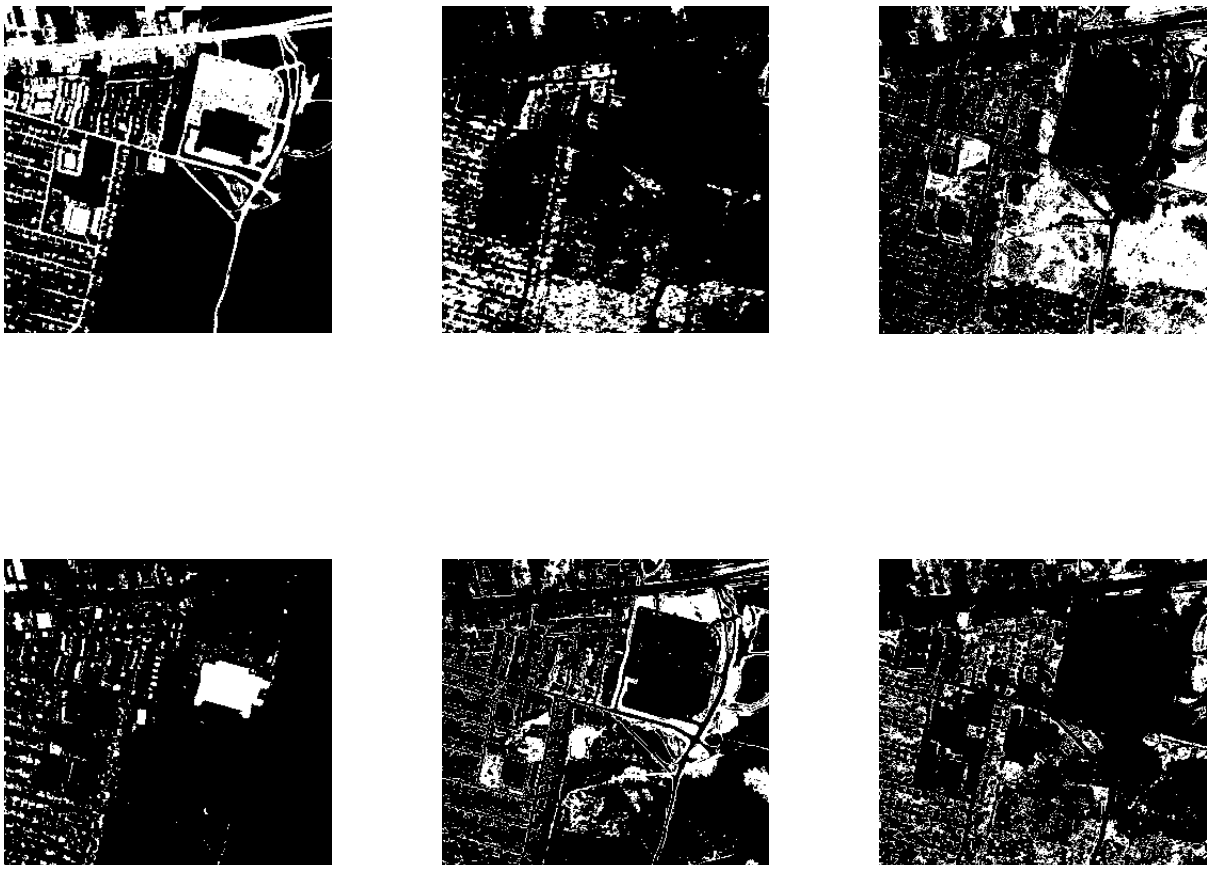


Figure 12: The 6 clusters that appear to be(starting from top, left to right) asphalt, trees, grass, roofing material, dirt and some other type of vegetation.

4 Change Detection

In this report we apply our change detection algorithm to a new data set. First, we summarize the problem and steps of the method. Given two hyperspectral images I_1 and I_2 of the same scene taken at different times, we want to determine any changes or differences between them. The first step is to obtain a set of endmembers for both HSIs simultaneously. This is done using our endmember detection algorithm presented in previous reports, and it chooses pixels of the HSIs as endmembers. By using both HSIs in the endmember detection, we will get some endmembers from the I_1 and the rest from I_2. Let $\left\{e_1^1, e_2^1, \ldots e_{N_1}^1\right\}$ be the N_1 endmembers chosen from I_1 and $\left\{e_1^2, e_2^2, \ldots e_{N_2}^2\right\}$ be the N_2 endmembers chosen from I_2. Using the combined set of endmembers and the unmixing algorithm we developed in previous reports, we unmix each pixel of each HSI obtaining an abundance vector for each pixel. The components of the abundance vectors correspond to the endmembers. One measure of change is to compute the angle between the abundance vectors of I_1 and I_2. If there was a change in the materials of a given pixel, it is expected that the difference between abundance vectors would indicate that change.

Although this method can work well for certain data sets, we have found that shadows present in one image but not present in the other, are often detected as a change. We observed that the shadowed regions are usually detected as an endmember by the endmember detection algorithm. Additionally, a similar material that is not under a shadow is often chosen as an endmember as well. We observed that if the components of the abundance vectors corresponding to these two endmembers representing similar materials under different lighting conditions are combined, we can drastically reduce the detection of shadows as change. Through numerical experimentation, we have developed a method that appears to help to determine which components of the abundance vectors to combine. Essentially, we look at the angle between pairs of endmembers, one from each HSI. The smaller the angle, the more similar we consider the materials to be. We have found that taking an average of the pixels around the chosen endmembers produced better results. This may be because the endmember detection algorithm tries to pick pixels that are different from each other. Therefore if a pixel of a given material is chosen, it is likely to be fairly different from the average signature of that material. The averaging process makes it easier to determine the similarity of materials, and thus is more useful in determining which components of the abundance vector to combine. In previous reports, we did experiments to determine the effect of the patch size that is used to get the average endmember signature, and we found that the size is not that critical. Of course, this patch size may depend on the spatial resolution of the image, but a patch size of 5x5 has worked well for all of the data sets we have tried. This is a simple approach, and has produced good results, but other methods of determining which endmembers are similar may work better. One example would be to use a library of spectral signatures of materials under different lighting and atmospheric conditions.

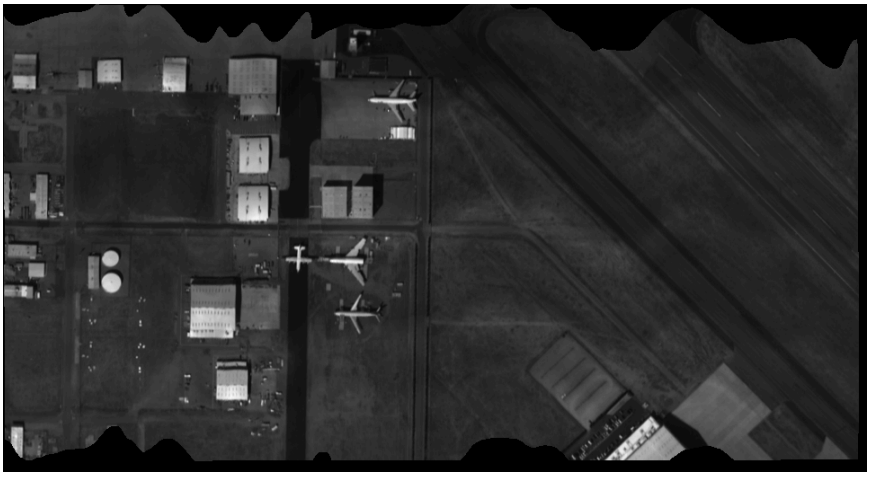
Once we have a list of the "closest" endmembers, we need to determine whether or not to combine the corresponding components of the abundance vectors. Again through numerical experiments of the data we have, we found that combining averaged endmembers that have an angle less than 0.1-0.15 gave us good results. Further experimentation on several data sets should give a better idea of what a good threshold value is, and if that value is dependent on the data itself.

4.1 Results on Airport Data

The data set consists of two hyperspectral images of the same scene taken at different times. They have 961x521 spatial pixels and 40 spectral bands each. Figure 13 shows the first band of each HSI. We see that almost half of the scene is runways and grass where there does not appear to be any change in the object present, so we will work on two subsets of the scene where more change has taken place. Figure 14 is a 350x480 pixel subset of the image that includes planes, hangers, and other small objects. In these figures, we can see the different shadows produced by taking the images at different times of day. Such dark and pronounced shadows make change detection difficult, as they are often detected as change when it is simply a change of lighting instead of an actual change in material.

The first step in our change detection algorithm is to compute endmembers, and Figure 15 shows the locations of the pixels computed as endmembers. Figure 16 shows the spectral signatures of the endmembers. Once the endmembers are obtained, each HSI is unmixed and an abundance vector is computed for each pixel. The angle between corresponding abundance vectors are then computed. Figure 17 shows the angle

between the abundance vectors from each HSI. We see that the areas of dark shadows are indicated as areas of change. Following the next step of our method, we compute an average spectral signature for each endmember by taking an average of a 5x5 patch centered on the endmember pixel. Figure 18 shows the spectral signatures of the averaged endmembers. Then the angle between these averaged endmember signatures was computed, and the components of the abundance vectors corresponding to the closest averaged endmembers are combined. Table 1 lists the 3 closest endmember pairs and the angles between them. We see that the 5th and 6th are the closest endmembers, and they appear to be shady concrete and sunny concrete, respectively. Combining the 5th and 6th components of the abundance vectors and computing the angle between the reduced abundance vectors gives us the result in Figure 19. A significant amount of the shadows produced by the buildings and some of the planes has been reduced. The final step we used was to apply total variation denoising to incorporate spatial information, and that result is shown in Figure 20.



(a) First image

(b) Second image

Figure 13: Plots of the first band for both images

	Endmember from first HSI	Endmember from second HSI	angle
1	Endmember 5	Endmember 6	0.0983
2	Endmember 3	Endmember 7	0.2386
3	Endmember 1	Endmember 4	0.3060

Table 1: Sorted table of closest averaged 7 endmembers.

5 TV-based Hyperspectral Compressed Sensing and Unmixing

Hyperspectral unmixing is to decompose each pixel spectrum to identify and quantify the relative abundance of each endmember (material), which typically demands enormous computational resources in terms of storage, computation and I/O throughput, especially when real-time processing is desired. Therefore, it is particularly difficult to directly process and analyze hyperspectral data cubes in real time or near real time. On the other hand, hyperspectral data are highly compressible with two-fold compressibility: 1) each spatial image is compressible, and 2) the entire cube, when treated as a matrix, is of low rank. To fully exploit such rich compressibility, we propose a scheme that never requires us to explicitly store or process any hyperspectral cube itself. In this CSU (compressive sensing and unmixing) scheme, data are acquired by means of CS and directly unmixed without reconstructing the underlying data cube.

Mathematically and ideally, the hyperspectral data model has the form

$$X = HW, \quad H\mathbf{1}_{n_e} = \mathbf{1}_{n_p}, \quad \text{and} \quad H \geq 0, \tag{11}$$

where X, H and W are the matrix representations of hyperspectral cube, abundance fraction and signatures of endmembers. $\mathbf{1}$ represents a vector of all ones, and n_e, n_b and n_p denote the number of endmembers, bands and pixels, respectively. Similar to other CS models, we consider the data acquisition model $AX = F$ where $A \in \mathbb{R}^{m \times n_p}$ is a random-like sensing matrix with $m < n_p$. To directly unmix the compressed observation F, we proposed to solve a compressed unmixing model (or its variants)

$$\min_H \sum_{j=1}^{n_e} \mathrm{TV}(He_j) \ \text{ s.t. } \ AHW = F, \ H\mathbf{1}_{n_e} = \mathbf{1}_{n_p}, \tag{12}$$

by TVAL3's variation specially tailored to this model. When the endmember spectral signatures W is given, we also propose a preprocessing procedure based on the singular value decomposition of F to substantially reduce the problem sizes, and enable near-real-time processing speeds. The feasibility of the proposed approach has been demonstrated by a test using compressed hyperspectral data collected by hardware similar to the single-pixel camera [7], as shown in Figure 21. Our target image is the color wheel on the left, composed of various intensity levels of three colors: yellow, cyan and magenta. The abundance fractions corresponding to the three endmembers were computed from 10% measured data, and are shown on the right side of Figure 21. The computational time to process the compressed unmixing was about 26 seconds. As we can see, our model and algorithm have accurately detected the areas corresponding to each color at various levels of brightness.

Four slices of the computed hyperspectral cube (corresponding to 4 different spectral bands), obtained by multiplying the estimated abundance matrix H with W, are given on the left side of Figure 22. For a comparison, we applied the 2D TV solver TwIST [8] to the same 10% measured data set and the results are given on the right side of Figure 22. Clearly the results of CSU are much better than the TwIST algorithm. Additionally, the computational time required for CSU is at least an order of magnitude less than that required by state-of-the-art 2D TV solvers.

This work for the first time has proved the concept of unmixing a hyperspectral data cube without the cube itself. We have also obtained results on "blind" unmixing where information of endmember signatures is either only partially known or severely corrupted. These results may have potential impact in the field of hyperspectral data processing.

6 Nonnegative matrix factorization and completion

6.1 The model and an ADM-based algorithm

Our paper [9] focuses on the optimization problem

$$\begin{aligned} \min_{X,Y} \quad & \|\mathcal{P}_\Omega(XY - M)\|_F^2 \\ \text{s.t.} \quad & X \in \mathbb{R}^{m\times q}, Y \in \mathbb{R}^{q\times n}, \\ & X_{ij} \geq 0, Y_{ij} \geq 0, \forall\ i, j, \end{aligned} \tag{13}$$

where $\Omega \subset \{1, \cdots, m\} \times \{1, \cdots, n\}$ indexes the known entries of M and $\mathcal{P}_\Omega(A)$ returns a copy of A that zeros out entries not in Ω. We call this problem *nonnegative matrix factorization/completion* (NMFC) since it is a combination of *nonnegative matrix factorization* (NMF)−which finds nonnegative factors of a nonnegative matrix given all of its entries−and *low-rank matrix completion* (LRMC)−which recovers M from an incomplete set of its entries without assuming nonnegativity.

We are interested in NMFC since it complements NMF and LRMC. NMF has been widely used in data mining such as text mining, dimension reduction and clustering, as well as spectral data analysis. Unlike NMF, NMFC assumes that the underlying matrix is incompletely sampled; hence, it leads to saving of sampling time and storage (for data such as images) and has broader applicability. On the other hand, LRMC has recently found a large number of applications including collaborative filtering, global positioning, system identification and order reduction, as well as the background subtraction and structure-from-motion problems in computer vision.

A rank-q matrix M can be written as $M = XY$ for matrices X with q columns and Y with q rows. When X and Y are known to be nonnegative *a priori*, empirical evidence shows that imposing nonnegativity on the factors improves the recovery quality. In particular, in certain applications such as hyperspectral unmixing, the factors are nonnegative due to their physical nature, so these applications will benefit from NMFC. To summarize, NMFC combines NMF and LRMC, and NMFC is useful when the underlying matrix has both low rank and nonnegative factors.

Since problem (13) is nonconvex, finding its solution becomes quite difficult. Although there is no theoretical guarantee of a solution, we observe very good numerical results by applying the alternating direction method (ADM) to NMF, which is also nonconvex.

6.2 Numerical results

We compared our algorithm in [9] to the algorithm proposed in [10], which takes complete samples of M and performs similar ADM-based iterations on random matrices with varying number of sampled entries. The recovery qualities and speeds are illustrated in Figure 23.

We compared our algorithm in [9] with LMaFit [11] and FPCA [12] on recovering three-dimensional hyperspectral images from their incomplete observations. Our test hyperspectral datacube has 163 slices, and the size of each slice is 80×80. Three selected slices are shown in Figure 24. The three algorithms were compared on recovering M from incomplete observations of SR = 30%, 40%, 50%, and their results were compared in terms of peak signal-to-noise ratio (PSNR), mean squared error (MSE), as well as relative nonnegativity feasibility (FA). The results are given in table 2, and the three slices of the recovered data cube that correspond to those in Figure 24 are given in Figure 25.

6.3 A Fast Algorithm with Convergence Guarantee

The method proposed in our recent paper [13] closes the gap between the above ADM-based algorithm, which is very fast but lacks convergence guarantees, and ANLS-type (alternating nonnegative least-squares) algorithms [14, 15, 16], which are widely used and have convergence guarantees but are comparatively slow. The new method is based on applying the alternating proximal gradient method to

$$\min_{X,Y} F(X, Y) \equiv \frac{1}{2}\|XY - M\|_F^2, \text{ s.t. } X \in \mathbb{R}_+^{m\times q}, Y \in \mathbb{R}_+^{q\times n}, \tag{14}$$

Table 2: Recovered slices by Algorithm [9], LMaFit, and FPCA. The rank estimate for Algorithm [9] and LMaFit is 30.

problem	Alg [9]				LMaFit				FPCA			
seed	CPU	PSNR	MSE	FA	CPU	PSNR	MSE	FA	CPU	PSNR	MSE	FA
SR: 30%												
3445	35.04	47.52	1.85e+1	0	40.39	42.10	6.45e+1	8.57e-3	38.49	44.53	3.69e+1	1.12e-3
31710	34.77	47.32	1.94e+1	0	23.43	43.46	4.71e+1	5.08e-3	38.40	44.42	3.79e+1	1.24e-3
43875	34.31	47.42	1.89e+1	0	38.27	42.54	5.83e+1	7.70e-3	38.64	44.71	3.54e+1	1.21e-3
69483	34.19	47.36	1.92e+1	0	34.53	42.74	5.57e+1	6.83e-3	38.75	44.62	3.61e+1	1.04e-3
95023	33.69	47.42	1.90e+1	0	27.88	42.98	5.27e+1	6.19e-3	38.78	44.51	3.70e+1	1.19e-3
SR: 40%												
3445	36.84	48.83	1.37e+1	0	35.05	43.90	4.26e+1	7.55e-3	42.76	44.72	3.53e+1	1.09e-3
31710	36.45	48.66	1.42e+1	0	24.45	44.71	3.54e+1	5.48e-3	42.57	44.53	3.69e+1	1.06e-3
43875	36.61	48.92	1.34e+1	0	17.67	46.07	2.59e+1	4.65e-3	42.85	44.66	3.58e+1	1.31e-3
69483	38.37	48.68	1.42e+1	0	19.13	45.59	2.89e+1	5.52e-3	42.88	44.52	3.70e+1	1.06e-3
95023	38.65	48.50	1.48e+1	0	22.16	45.30	3.09e+1	5.02e-3	43.22	44.56	3.66e+1	1.09e-3
SR: 50%												
3445	39.87	49.74	1.11e+1	0	34.31	44.72	3.53e+1	9.62e-3	47.12	44.24	3.94e+1	8.55e-4
31710	39.77	49.75	1.11e+1	0	29.90	45.23	3.14e+1	4.78e-3	46.64	45.25	3.12e+1	1.12e-3
43875	37.78	49.78	1.10e+1	0	25.53	45.92	2.68e+1	6.12e-3	46.63	44.60	3.63e+1	1.44e-3
69483	38.24	49.65	1.14e+1	0	31.14	44.92	3.37e+1	7.77e-3	47.07	44.39	3.81e+1	1.08e-3
95023	40.09	49.64	1.14e+1	0	29.92	45.42	3.00e+1	5.64e-3	47.90	43.93	4.24e+1	1.16e-3

where for presentation brevity we have let $P_\Omega = I$ in (13) and $\mathbb{R}^{m\times n}_+ = \{X \in \mathbb{R}^{m\times n} : X_{ij} \geq 0, 1 \leq i \leq m, 1 \leq j \leq n\}$. Specifically, proximal gradient steps are applied alternatively to the X-subproblem (15a) and Y-subproblem (15b):

$$X^k = \arg\min_{X\geq 0} \frac{1}{2}\|XY^{k-1} - M\|_F^2, \tag{15a}$$

$$Y^k = \arg\min_{Y\geq 0} \frac{1}{2}\|X^kY - M\|_F^2, \tag{15b}$$

respectively. Our method uses simple updates at each iteration and has fast convergence like the ADM-based algorithm. In addition, it decreases the objective monotonically and has provable convergence like the ANLS-type algorithms. Under some mild assumptions, we prove convergence and estimate the convergence rate by applying the Kurdyka-Łojasiewicz inequality [17, 18].

6.3.1 Proximal gradient method

Consider the convex optimization problem

$$\min_{x\in\mathcal{X}} f(x), \tag{16}$$

where $\mathcal{X}$ is a convex set and f is a differentiable convex function. Assume f has Lipschitz continuous gradient, i.e., there is a constant $L_f > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L_f\|x - y\|_2, \quad \text{for all } x, y \in \mathcal{X}.$$

The proximal gradient method for solving (16) is

$$x^{k+1} = \mathcal{P}_\mathcal{X}(\hat{x}^k - \nabla f(\hat{x}^k)/L_f), \text{ for } k = 0, 1, \cdots \tag{17}$$

where $\mathcal{P}_\mathcal{X}$ denotes the projection to $\mathcal{X}$. In (17), x^{k+1} is the solution of

$$\min_{x\in\mathcal{X}} f(\hat{x}^k) + \langle\nabla f(\hat{x}^k), x - \hat{x}^k\rangle + \frac{L_f}{2}\|x - \hat{x}^k\|_2^2. \tag{18}$$

With an appropriate choice of $\hat{x}^k$, this method is guaranteed to converge to a solution of (16).

Applying this method to our subproblem

$$\min_{W\geq 0} h(W) \equiv \frac{1}{2}\|AW - B\|_F^2, \tag{19}$$

Algorithm 1 APG: Alternating proximal gradient method for NMF

Input: $m \times n$ nonnegative matrix M and pre-determined dimension q
Initialization: choose a positive number $\delta_\omega < 1$ and randomize matrices $X^{-1} = X^0 \geq 0$ and $Y^{-1} = Y^0 \geq 0$
for $k = 0, 1, 2, \ldots$ **do**
  Set ω_X^k, ω_Y^k according to (22) and let $\hat{X}^k = X^k + \omega_X^k(X^k - X^{k-1}), \hat{Y}^k = Y^k + \omega_Y^k(Y^k - Y^{k-1})$.
  Update (X^{k+1}, Y^{k+1}) according to (21).
  if $F(X^{k+1}, Y^{k+1}) \geq F(X^k, Y^k)$ **then**
    Update (X^{k+1}, Y^{k+1}) according to (21) with $\hat{X}^k = X^k, \hat{Y}^k = Y^k$
  end if
  if some stopping criterion is met **then**
    Stop and output (X^{k+1}, Y^{k+1})
  end if
end for

we obtain

$$W^{k+1} = \mathcal{P}_+\left(\hat{W}^k - \nabla h(\hat{W}^k)/L_h\right), \tag{20}$$

where L_h is a Lipschitz constant of $\nabla h(W)$. Although we can iterate (20) many times to obtain (15a) and do the same to obtain (15b), we propose to run only one iteration of (20) toward (15a) and do the same toward (15b). This way, we alternatively update X, Y by

$$X^{k+1} = \mathcal{P}_+\left(\hat{X}^k - \nabla_X F(\hat{X}^k, Y^k)/L_X^k\right), \tag{21a}$$

$$Y^{k+1} = \mathcal{P}_+\left(\hat{Y}^k - \nabla_Y F(X^{k+1}, \hat{Y}^k)/L_Y^k\right), \tag{21b}$$

where L_X^k and L_Y^k are Lipschitz constants of $\nabla_X F(X, Y^k)$ and $\nabla_Y F(X^{k+1}, Y)$ with respect to X and Y, respectively.

In our algorithm, we take L_X^k and L_Y^k as the spectral norm of Y^k and X^{k+1}, respectively. In addition, we set $\hat{X}^k = X^k + \omega_X^k(X^k - X^{k-1})$ and $\hat{Y}^k = Y^k + \omega_Y^k(Y^k - Y^{k-1})$, where ω_X^k, ω_Y^k are chosen based on the Nesterov-type extrapolation weight.[1] For ease of convergence analysis, we impose $\omega_X^k \leq \delta_\omega\sqrt{\frac{L_X^{k-1}}{L_X^k}}$ and $\omega_Y^k \leq \delta_\omega\sqrt{\frac{L_Y^{k-1}}{L_Y^k}}$ with some nonnegative constant $\delta_\omega < 1$. Specifically, we set

$$\omega_X^k = \min\left(\hat{\omega}_k, \delta_\omega\sqrt{\frac{L_X^{k-1}}{L_X^k}}\right), \qquad \omega_Y^k = \min\left(\hat{\omega}_k, \delta_\omega\sqrt{\frac{L_Y^{k-1}}{L_Y^k}}\right), \tag{22}$$

where $\hat{\omega}_k = (t_{k-1} - 1)/t_k$ and $t_k = \frac{1}{2}\left(1 + \sqrt{1 + 4t_{k-1}^2}\right)$ with initial value $t_{-1} = 1$. To ensure monotonic objective decrease, whenever $F(X^{k+1}, Y^{k+1}) \geq F(X^k, Y^k)$, we redo the kth iteration to update (X^{k+1}, Y^{k+1}) with $\hat{X}^k = X^k, \hat{Y}^k = Y^k$. The method is summarized in Algorithm 1, and we call it APG.

6.3.2 Convergence Guarantee

Below we present the theorems that summarize the convergence theory given in [13].

Theorem 6.1 (Global convergence)**.** *Assume that the sequence $\{Z^k\}$ generated by the algorithm APG is uniformly away from* zero *and bounded, i.e., there exist positive constants ℓ, L such that $\ell \leq L_X^k, L_Y^k \leq L$ for any $k \geq 0$. In addition, assume $\delta_\omega \leq \delta\sqrt{\frac{\ell}{L}}$ for some nonnegative $\delta < 1$. Then, $\{Z^k\}$ converges to a critical point of* (14) *from any starting point Z^0.*

[1] The dynamically updated extrapolation weight ω_k used in [19, 20] also works well for our algorithm.

	APG		ADM		**blockpivot**	
q	relerr	time	relerr	time	relerr	time
20	1.18e-2	34.2	2.34e-2	87.5	1.38e-2	62.5
30	9.07e-3	63.2	2.02e-2	116	1.10e-2	143
40	7.56e-3	86.2	1.78e-2	140	9.59e-3	194
50	6.45e-3	120	1.58e-2	182	8.e-3	277

Table 3: Comparison results of the solutions obtained by APG, ADM and **blockpivot** on hyperspectral cube of size $150 \times 150 \times 163$ with dimension q varying among $\{20, 30, 40, 50\}$

$T = 50$	APG-MC		ADM-MC		$T = 100$	APG-MC		ADM-MC	
SR	PSNR	MSE	PSNR	MSE	SR	PSNR	MSE	PSNR	MSE
0.20	32.30	5.89e-4	28.72	1.35e-3	0.20	32.57	5.54e-4	28.80	1.33e-3
0.30	40.65	8.62e-5	33.58	4.64e-4	0.30	41.19	7.61e-5	33.69	4.52e-4
0.40	45.77	2.66e-5	38.52	1.46e-4	0.40	46.03	2.50e-5	38.69	1.41e-4

Table 4: Comparison results of the solutions obtained by APG-MC and ADM-MC on a hyperspectral cube of size $150 \times 150 \times 163$ for sample ratio SR $= \{0.20, 0.30, 0.40\}$ and maximum running time (sec) $T = 50, 100$.

Theorem 6.2 (Convergence rate)**.** *Under the assumptions of Theorem 6.1, suppose that Z^k converges to some critical point $\bar{Z}$. Then there exists some $\theta \in (0, 1)$ such that the following estimations hold:*

1. *If $\theta \in (0, \frac{1}{2}]$, then $\|Z^k - \bar{Z}\|_F \leq C\tau^k$ for some constants $C > 0$ and $\tau \in [0, 1)$;*
2. *If $\theta \in (\frac{1}{2}, 1)$, then $\|Z^k - \bar{Z}\|_F \leq Ck^{-(1-\theta)/(2\theta-1)}$ for some constant $C > 0$.*

6.4 Hyperspectral Simulation

It has been shown in [21] that NMF can be applied to spectral data analysis. In [21], a regularized NMF model is also considered with penalty terms $\alpha\|X\|_F^2$ and $\beta\|F\|_F^2$ added in the objective of (14). The parameters α and β can be tuned for a specific purpose in practice. Here, we focus on the original NMF model (14) to show the effectiveness of our algorithm. However, our method can be easily extended to solve the regularized NMF model. In this test, we use a $150 \times 150 \times 163$ hyperspectral cube to test the compared algorithms. Each slice of the cube is reshaped as a column vector, and a 22500×163 matrix M is obtained. In addition, the cube is scaled to have the unit maximum element. Four selected slices are shown in the first row of Figure 26 corresponding to the 1st, 50th, 100th and 150th columns of M. The dimension q is chosen from $\{20, 30, 40, 50\}$, and Table 3 lists the average results of 10 independent trials. We can see from the table that APG is superior to ADM and **blockpivot** in both speed and solution quality.

We also compare the algorithm APG-MC and the ADM-based algorithm (ADM-MC) in [9] on the hyperspectral data used in last test. It is demonstrated in [9] that ADM-MC outperforms matrix completion solvers APGL and LMaFit for nonnegative matrix completion problems, because ADM-MC utilizes the nonnegativity property of the data while the latter two do not. We fix the dimension $q = 40$ in (13) and choose sample ratio: SR $\triangleq \frac{|\Omega|}{mn}$ from $\{0.20, 0.30, 0.40\}$. The parameter δ_ω for APG-MC is set to 1, and all the parameters for ADM-MC are set to their default values. To make the comparison consistent and fair, we let both of the two algorithms run to a maximum time (sec) $T = 50, 100$, and we employ peak-signal-to-noise-ratio (PSNR) and mean squared error (MSE) to measure the performance of the two algorithms. Table 4 lists the average results of 10 independent trials, and Figure 26 plots the 4 corresponding slices of each algorithm for SR $= 0.20$ and $T = 50$ in the second and third row. From the table, we can see that APG-MC is significantly better than ADM-MC in all cases.

7 Multispectral Oil Spill Detection

The paper [22] presents a novel application of sparse reconstruction for offshore oil spill sensing based on multispectral measurements. Early detection of the onset of oil spills provides rapid first response and more time to develop a post-accident strategy. It is the key to prevent the snowball effect where multiple catastrophic events lead to a major accident such as the recent disastrous accident of British Petroleum. The current solution employs routine helicopter fly-bys that rely on assisted human vision, which apparently leaves out much useful information imperceptible to human eyes. Also, these fly-by surveys can only be performed during daytime and when weather permits. In [22], we propose a novel video-rate infrared (IR) multispectral imaging system, which aims to replace the traditional fly-by survey and achieve higher accuracy at lower cost.

The proposed imaging system is shown in Figure 27, which is based on a similar system operating in the visible wavelengths that was developed in [23] with a fixed random coded mask. First, the thermal emission from a region of interest, represented by $f_s(x, y, \lambda)$, where (x, y) is the spatial coordinate and λ is spectrum. It is first demagnified and imaged to the object plane of a telescopic 4f system $f_0(x, y, \lambda)$, and then passed through a reflective pixelated spatial light modulator with a binary or greyscale reflection function $T(x, y)$. The resulting field, Tf_0, is then imaged by the dispersive 4f system with a grating placed at its Fourier plane. The field at the detector plane can be written as the convolution of the point spread function of the spectrograph and Tf_0 as

$$f(x, y, \lambda) = T(x, y)f_0(x, y, \lambda). \tag{23}$$

Then f is convolved with the dispersive 4f system. Since the detector array sensitivity covers the wavelength range of 7–14μm, the measured quantity received is integrated over the wavelengths:

$$g(x, y) = \int f(x + \alpha(\lambda - \lambda_c), y, \lambda)d\lambda, \tag{24}$$

where α is the linear dispersion of the grating and λ_c is the center wavelength. The above equation serves as the general imaging model in the continuous spatial domain. By recognizing that both the mask and the detector array are in fact pixelated and measurement noise is always present, (24) can be rewritten in a matrix form as

$$g = Hf + w. \tag{25}$$

Our goal is to recover f_0 from g and H with unknown noise w. The recovery process has two steps: first, recover f from g; then recover f_0 by inpainting f.

The recovery of f takes advantages of the fact that oil spills are approximately sparsely composed of a small number of spectrum signatures (shown in Fig. 28), which form a dictionary $\Phi = (\phi_1, \phi_2, \ldots, \phi_J)$. If we define a new matrix $H_2 := H\Phi$ and let u be the coefficients of the combination, then we have $H_2u = Hf = g$ and u is sparse. To find a sparse u such that $H_2u = g$ we use the following ℓ_1 minimization model:

$$\min |u|_1, \text{s.t. } H_2u = g \text{ and } 0 \le u \le 1. \tag{26}$$

The next task is essentially an image inpainting problem, i.e., to fill in the unknown region in the image according to the known region. We skip the details of this part.

The simulated data in Fig. 29 is a lab picture. The left figure shows a 2D representative of the 3D image ($f_0(x, y, \lambda)$), where every pixel is classified as one of three different oil film thickness values (10μm, 50μm, 100μm) and surrounding water. Then, the mask $T(x, y)$ is chosen with 30% random nonzero binary elements. The middle subfigure shows the resulting measurement g, which contains Gaussian random noise (SNR:30). The right subfigure displays the recovered result, demonstrating the efficacy of the proposed method. Fig. 30 shows the performance with different noise levels in the measurements, and with randomly selected 25% nonzeros in T. Overall, 90% pixels are correctly classified with noise less than 5%.

Learning Circulant Sensing Kernels

In the paper [24], we are motivated by [25] and propose numerical methods to minimize $\|(\Phi\Psi)^*(\Phi\Psi) - I\|_F$ to improve the recoverability of the basis pursuit problem

$$\min_\theta \|\theta\|_1, \text{ s.t. } \Phi\Psi\theta = b, \tag{27}$$

or the denoising model

$$\min_{\theta} \|\theta\|_1 + \frac{1}{\rho}\|\Phi\Psi\theta - b - \sigma\eta\|_2^2, \tag{28}$$

where Φ is either a full or partial circulant matrix, Ψ is either an analytic or learned dictionary, b is the measurement vector and η is noise with noise level σ. Like [25], we also learn Φ and Ψ together (namely, coupled learning of Φ and Ψ), rather than just learning Φ for a given Ψ.

Let $\Phi = PC$, where P is a downsampling matrix selecting m out of n rows from C and C is a square circulant matrix. We first optimize C by minimizing $\|(C\Psi)^*(C\Psi) - I\|_F$ for a given Ψ and then P by minimizing $\|(PC\Psi)^*(PC\Psi) - I\|_F$ for given C and Ψ. Let F be the normalized discrete Fourier transformation matrix. Then C can be written as $C = F\mathrm{diag}(d)F^*$ for some d. Using this fact, the problem of minimizing $\|(C\Psi)^*(C\Psi) - I\|_F$ can be equivalently transformed to

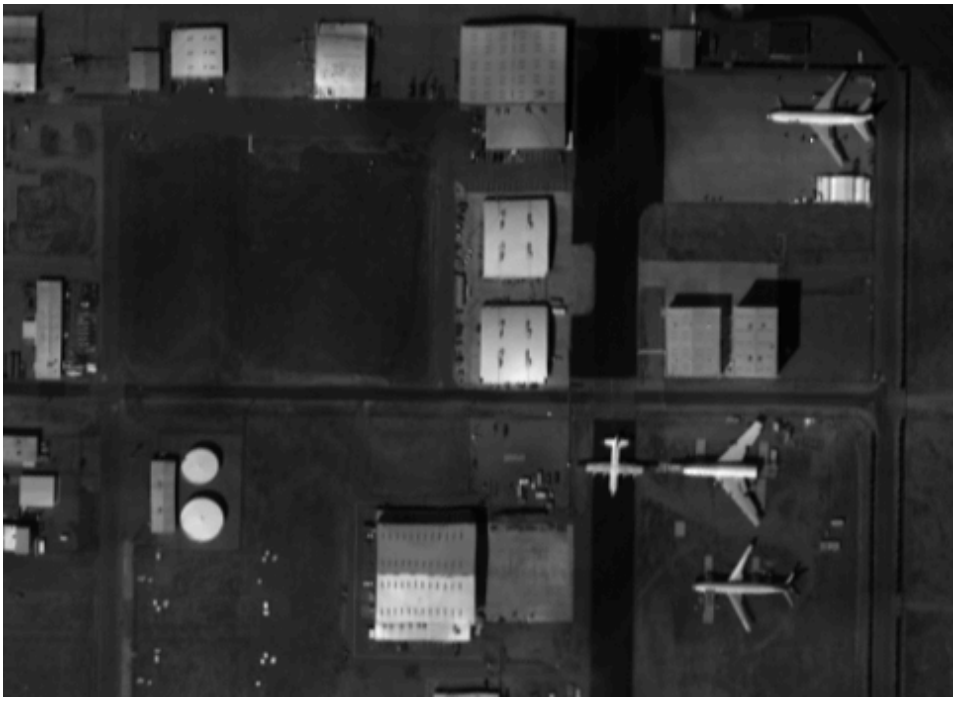
$$\min_{x\geq 0} \frac{1}{2}x^\top \bar{B}x - x^\top \mathrm{diag}(B), \tag{29}$$

where $x = [|d_1|^2, \ldots, |d_n|^2]^\top$, $B = F^*\Psi\Psi^*F$ and $\bar{B} = [|B_{ij}|^2]$. In addition, the problem of minimizing $\|(PC\Psi)^*(PC\Psi) - I\|_F$ can be tranformed to

$$\min_{p} p^\top Hp - 2f^\top p, \text{ s.t. } \sum_i p_i = m \text{ and } p_i \in \{0, 1\}, \ \forall i, \tag{30}$$

where $p_i = 1$ if P selects row i and *zero* otherwise, $H = [|G_{ij}|^2]$, and $f = \mathrm{diag}(G)$ with $G = C\Psi\Psi^*C^*$. To get a real valued C, we can make d conjugate symmetric by confining $d_i = \mathrm{conj}(d_{i'})$ for every i and $i' = \mathrm{mod}(n - i + 1, n) + 1$ in the problem (29).

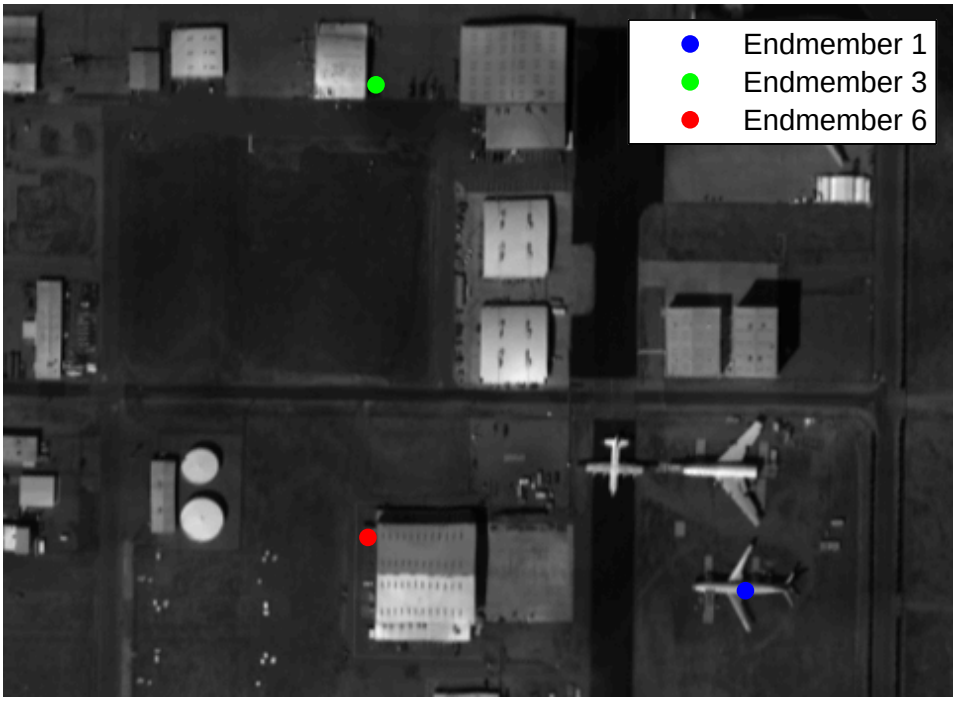
In synthetic tests, we use YALL1 [26] to solve (27) with Gaussian random basis and Fourier basis to compare four different sensing matrices: *rand-circ* (randomly generated complex partial circulant matrix Φ), *Gaussian* (complex Gaussian random matrix Φ), *opt-circ* ($\Phi = PC$ with optimized C and uniformly random P) and *opt-circ-and-P* ($\Phi = PC$ with optimized C and P). Figure 31 depicts the success rate out of 50 independent trials (a recovery is called *successful* if the relative error is less than 10^{-4}). Both the two subfigures of Figure 31 reveal that optimized circulant sensing matrices lead to equally good performance as Gaussian random matrices. For the random basis, random circulant matrices achieve similar recovery success rate, while they perform extremly bad for the Fourier basis. On the other hand, optimizing the selection matrix P hardly makes further improvement. We believe that unless the underlying signal has dominant frequencies, optimizing the selection matrix P will not lead to consistent improvement.



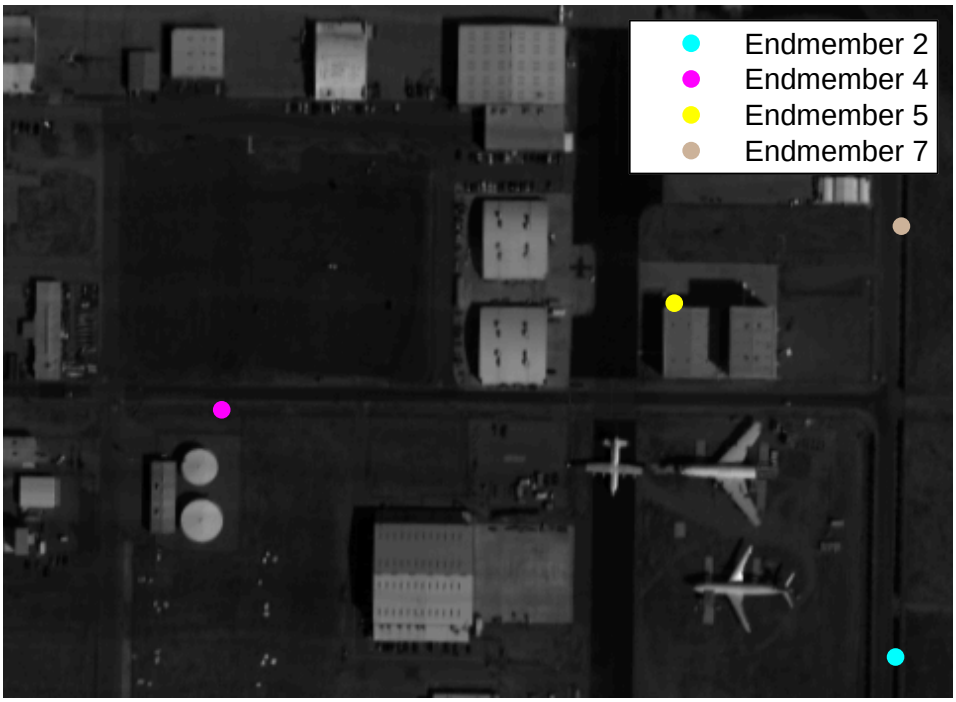
(a) First image

(b) Second image

Figure 14: Plots of the first band for both images



(a) 3 endmembers from first scene.



(b) 4 endmembers from second scene.

Figure 15: Locations of the computed endmembers.

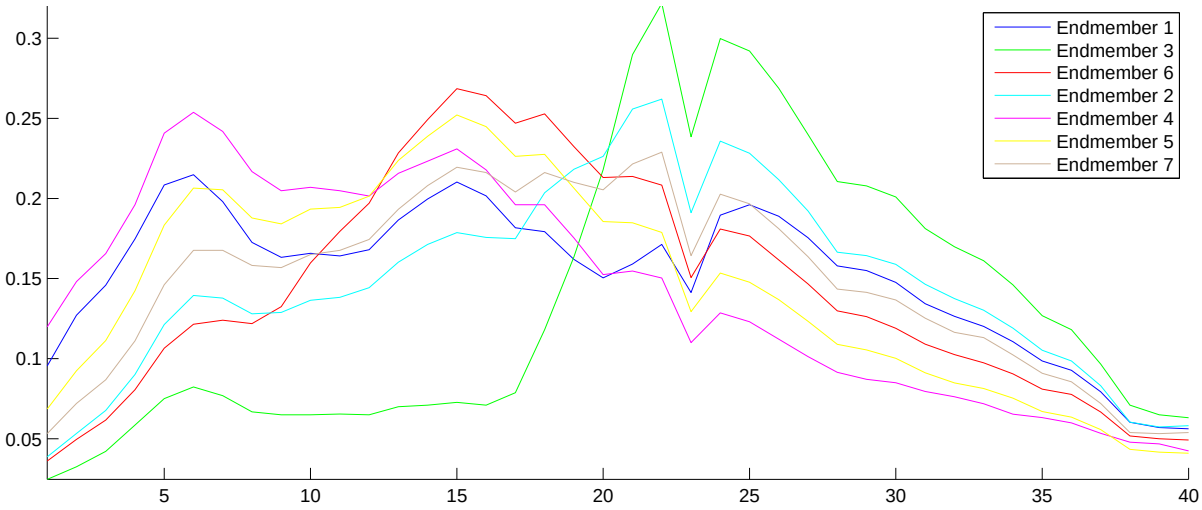


Figure 16: Spectral signatures of the 7 endmembers

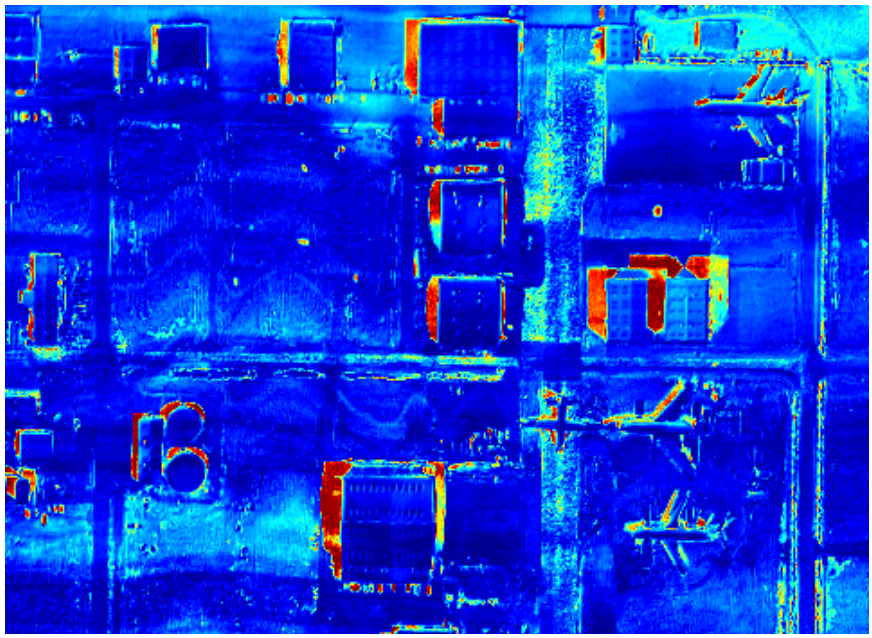


Figure 17: Angle between abundance vectors computed using 7 endmembers.

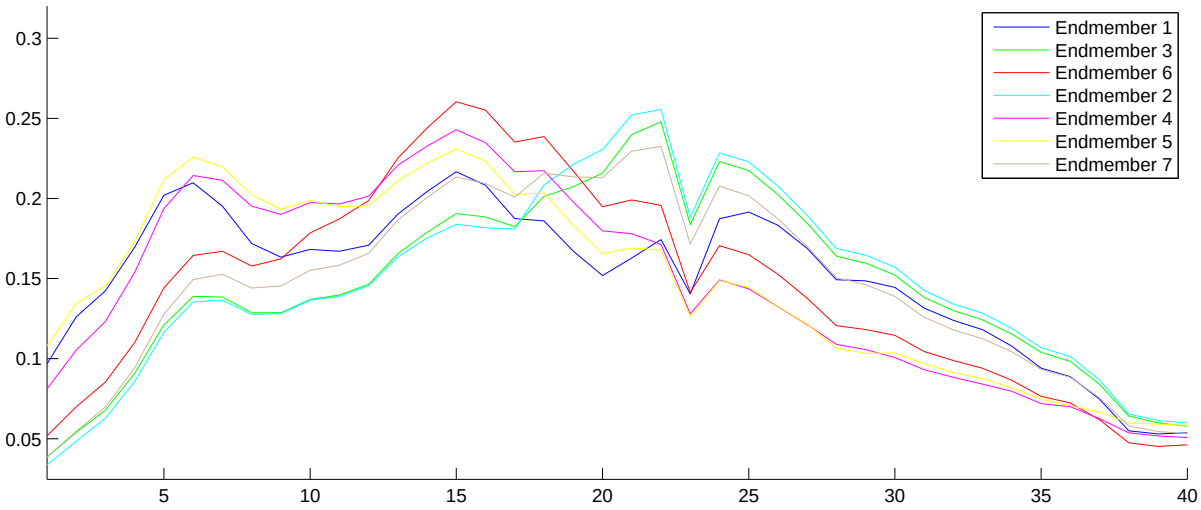


Figure 18: Averaged spectral signatures of the 7 endmembers

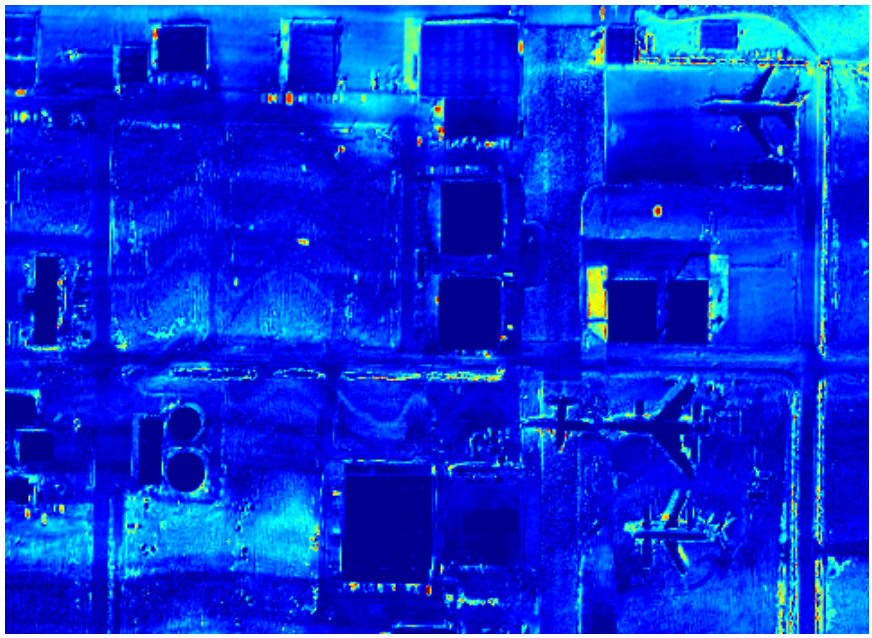


Figure 19: Angle between reduced abundance vectors combining the 5th and 6th components.

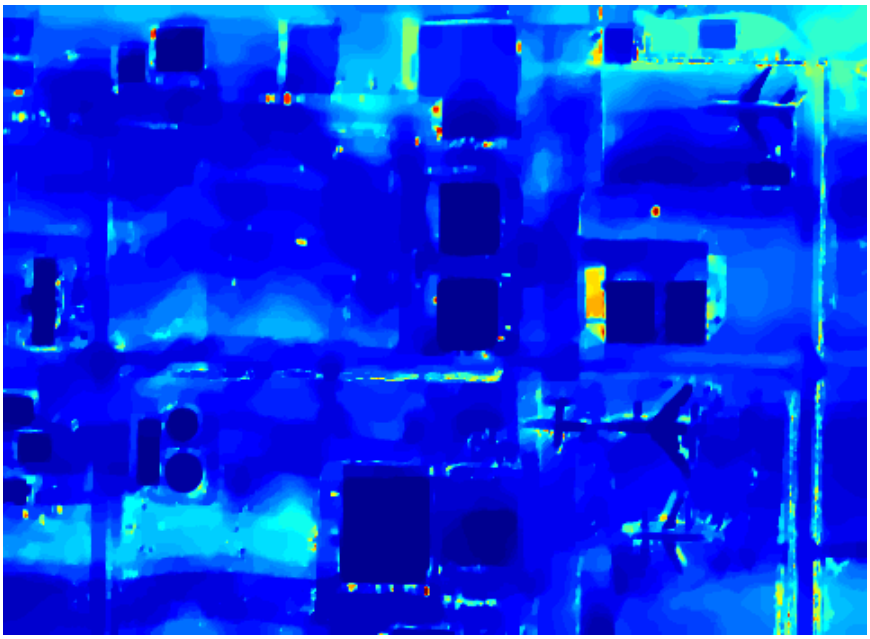


Figure 20: Change detection result after applying TV regularization.

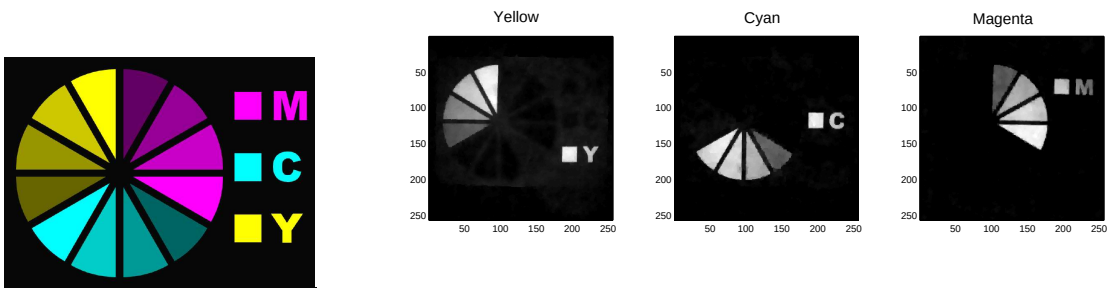


Figure 21: Target image "Color wheel" (left) and estimated abundance from 10% measurements.

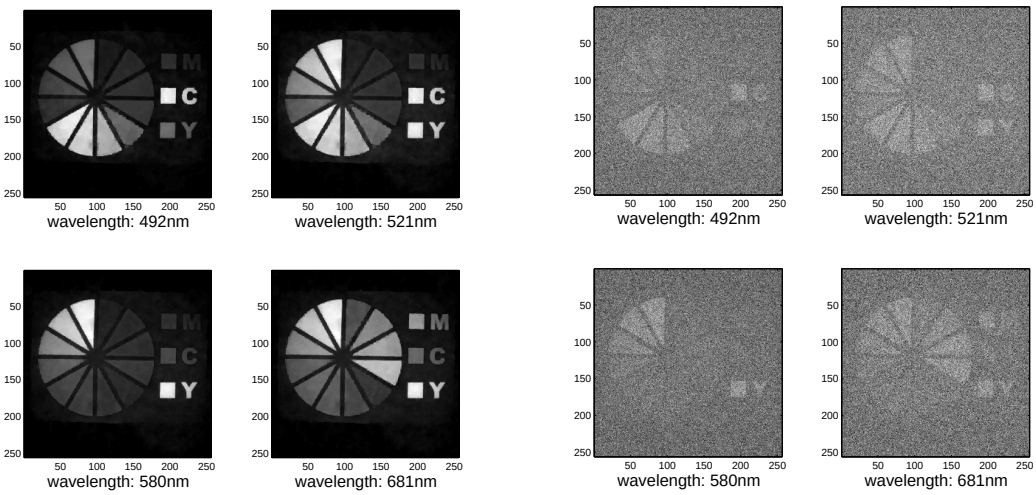


Figure 22: Four slices computed by the proposed approach (left) and by TwIST (right)

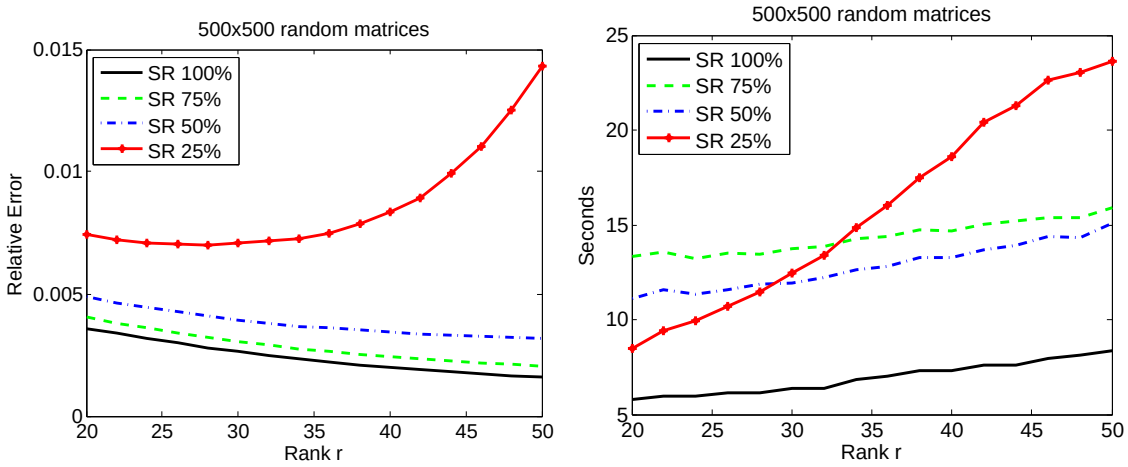


Figure 23: Matrix completion with different sample rates (SRs). Left: relative error in Frobenious norm; Right: cpu time in seconds. The algorithm in [10] was used for SR=100%. Algorithm [9] was used for SR=70%, 50%, 25%. All tests used the same parameters and stopping tolerances, and results are the averages over 50 independent trials

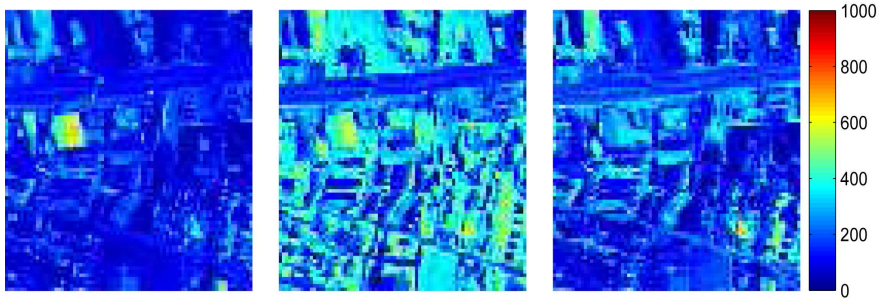
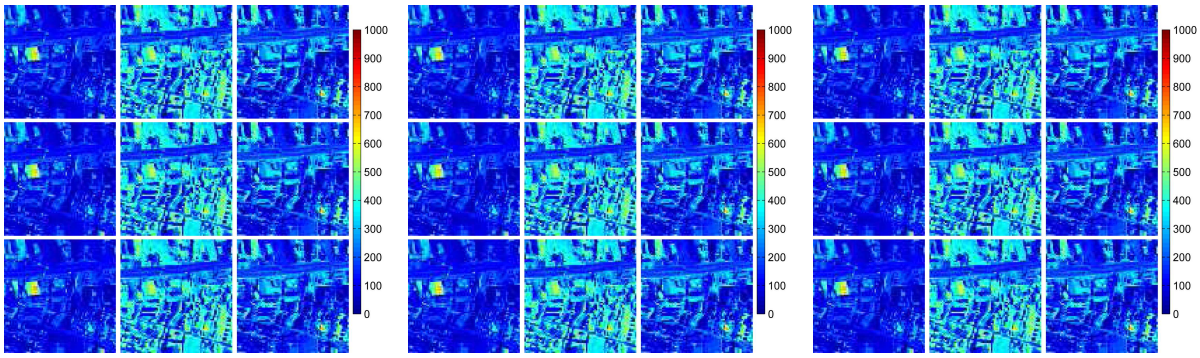


Figure 24: Three selected slices for hyperspectral data cube



(a) q=30, SR = 30% (b) q=30, SR = 40% (c) q=30, SR = 50%

Figure 25: Real data: the first, second and third row of each subfigure are recovered images by Algorithm [9], LMaFit, and FPCA, respectively

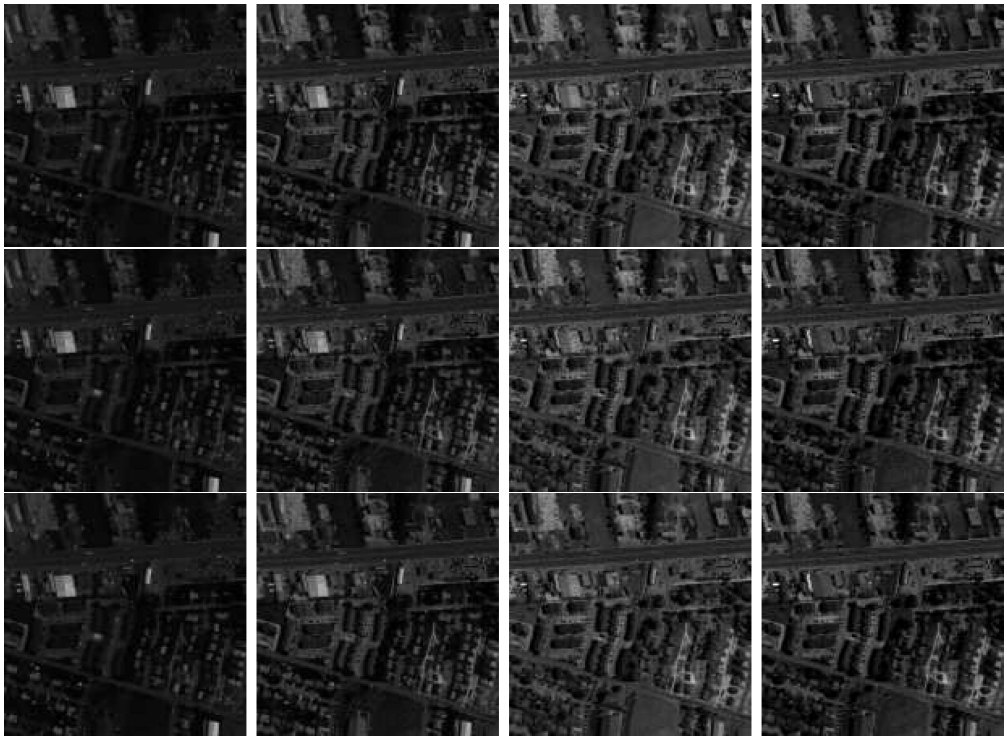


Figure 26: Hyperspectral data: four selected slices (first row) and the corresponding recovered slices by APG-MC (second row) and ADM-MC (third row) for sample ratio SR = 0.20 and maximum running time $T = 50$.

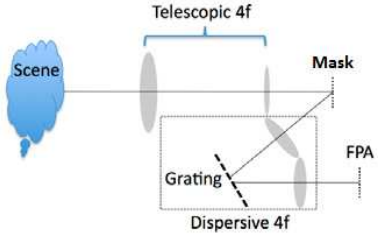


Figure 27: Proposed imaging system.

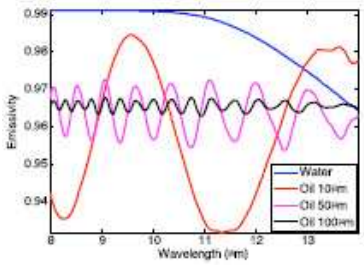


Figure 28: Spectral response of different oil film thickness and water.

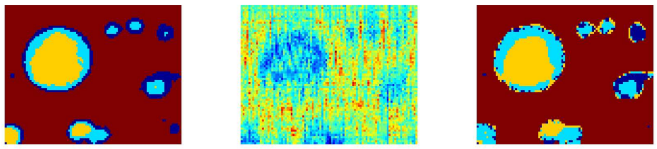


Figure 29: Simulated results (92% recovered correctly) on a 67×92 image with 30% nonzeros in T and noise (SNR=30) added, and the running time is 5 seconds.

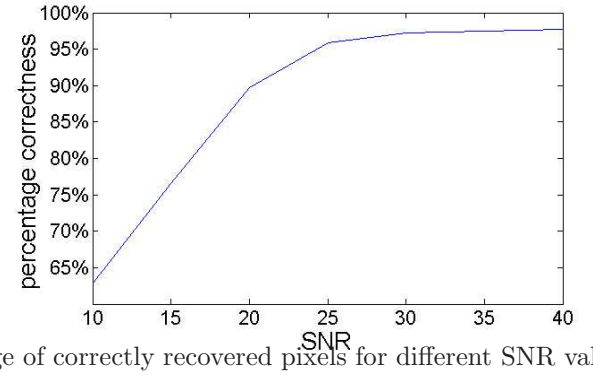


Figure 30: The percentage of correctly recovered pixels for different SNR values with 25% nonzeros in T.

Figure 31: Comparison results of four different sensing matrices: *rand-circ*, *Gaussian*, *opt-circ* and *opt-circ-and-P* with Gaussian random basis (left) and Fourier basis (right).

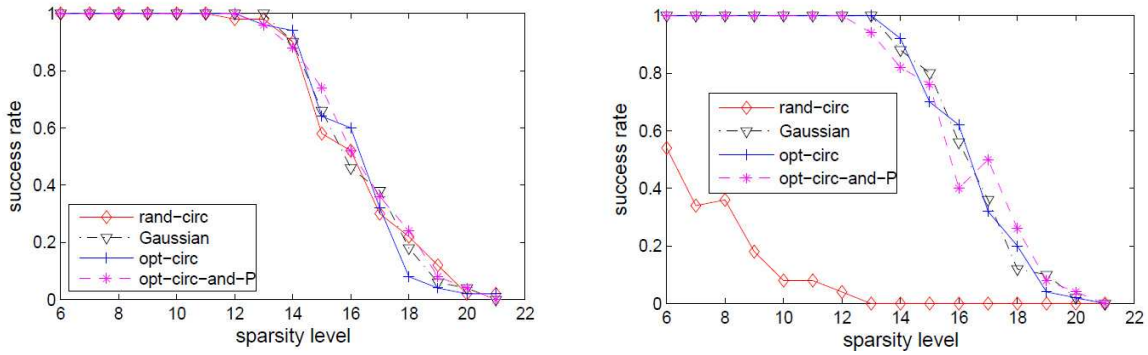
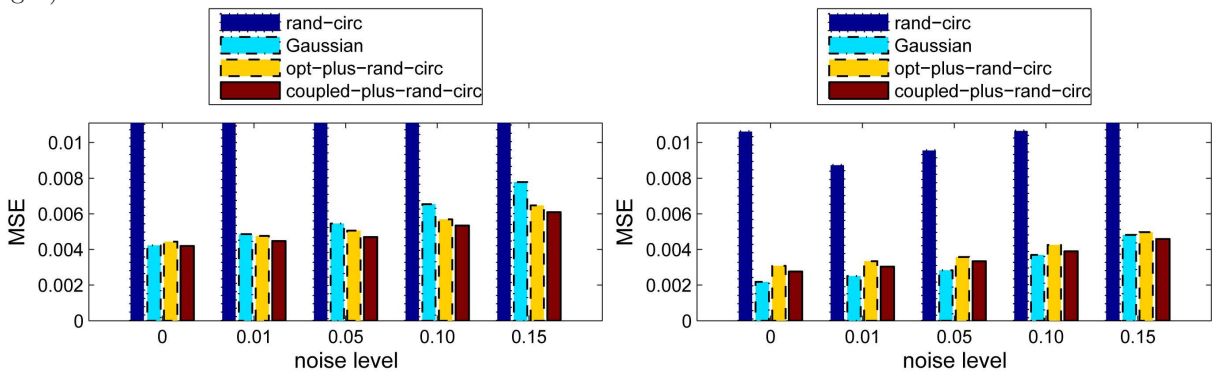


Figure 32: Comparison results of four different sensing matrices: *rand-circ*, *Gaussian*, *opt-plus-rand-circ* and *coupled-plus-rand-circ* on Berkeley segmentation dataset for sampled row numbers $m = 16$ (left) and $m = 24$ (right).



For real image, we did two sets of tests. In the first set, we use YALL1 to solve (28) with learned dictionaries Ψ by KSVD method [27] using 20,000 randomly extracted 8×8 patches from the 200 training images in the Berkeley segmentation dataset [28]. We test *rand-circ*, *Gaussian* and *opt-plus-rand-circ* (*opt-circ* with C added by a real random circulant), as well as *coupled-plus-rand-circ* ($\Phi = PC$ with uniformly random P and C the sum of a real random circulant and that learned simultaniously with its corresponding dictionary Ψ) on 600 uniformly randomly chosen 8×8 patches from the 100 testing images in the Berkeley segmentation dataset. Figure 32 depicts the mean squared error: $\text{MSE} = \sum_{i=1}^{\ell} \|x^i - \Psi\theta^i\|_2^2/(64\ell)$, where x^i is the ith selected patch, ℓ is the number of tested patches, and θ^i was the solution of (28) output by YALL1. All results are averages of 20 independent trials. Both the two pictures in Figure 32 reveal that uncoupled and coupled learning approaches achieve significantly better recovery performance over random circulant matrices, and they are even better than Gaussian random matrices when $m = 16$. The coupled learning approach makes a slightly better performance over the uncoupled one. We believe that coupled learned circulant matrices will not make a large improvement over uncoupled learned ones unless the underlying signal is sparser under the learned dictionary.

In the second set of tests, we use a $150 \times 150 \times 163$ hyperspectral cube and learn a dictionary from all the 21025 overlapping 6×6 patches extracted from the first slice of the hyperspectral cube, which is shown in the left side of Figure 33. The four different sensing matrices *rand-circ*, *Gaussian*, *opt-plus-rand-circ* and *coupled-plus-rand-circ* are compared on 500 uniformly randomly selected patches from the remaining 162 slices. Figure 34 depicts the average MSEs of 20 independent trials, and Figure 35 plots one set of recovered slices corresponding to the 150th slice shown in the right side of Figure 33. Again, the learned circulant sensing matrices make greater improvement over the random circulant one and even better than the Gaussian random one.

Figure 33: Two selected slices of the hyperspectral data

Figure 34: Comparison results of four different sensing matrices: *rand-circ*, *Gaussian*, *opt-plus-rand-circ* and *coupled-plus-rand-circ* on hyperspectral data for sampled row numbers $m = 8$ (left) and $m = 12$ (right).

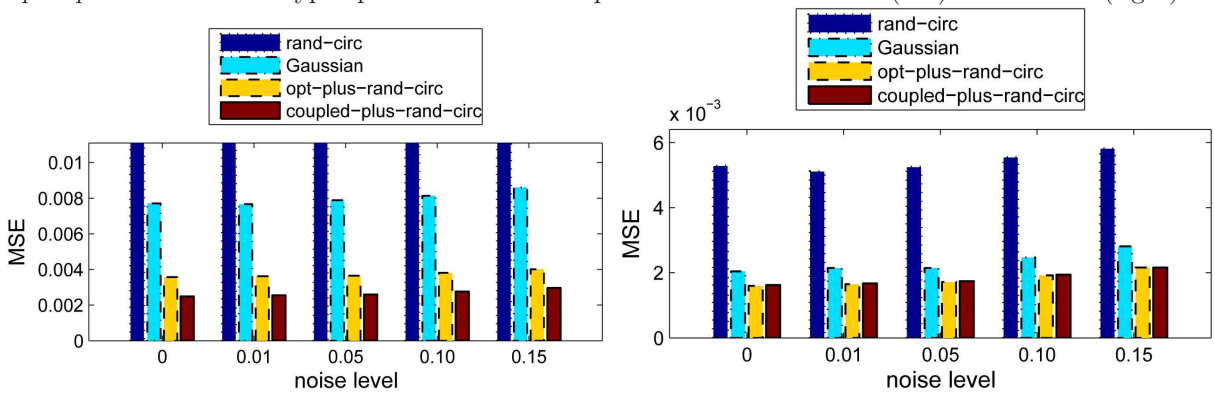
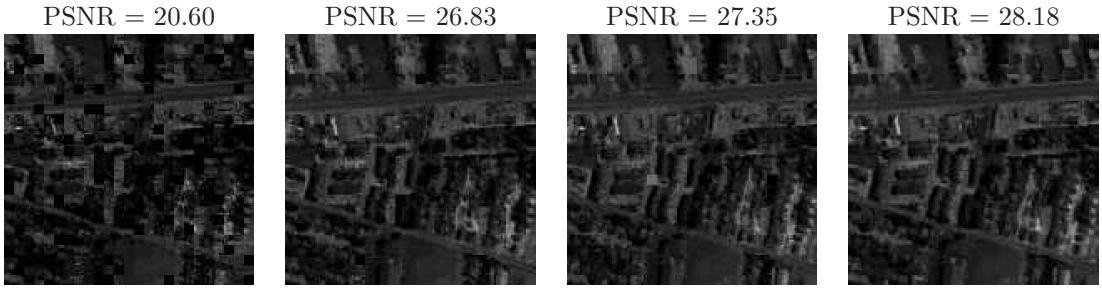


Figure 35: One set of recovered slices by YALL1 with *rand-circ*, *Gaussian*, *opt-plus-rand-circ* and *coupled-plus-rand-circ* for $m = 12$



References

[1] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *SIAM Journal on Multiscale Modeling and Simulation* **4**(2), pp. 460–489, 2005.

[2] Y. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for l1-minimization with applications to compressed sensing," *SIAM J. Imagins Sciences* **1(1)**, pp. 143–168, 2008.

[3] T. Goldstein and S. Osher, "The split Bregman algorithm for L1 regularized problems," *UCLA CAM Report 08-29* , 2008.

[4] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removing algorithms," *Physica, D.,* **60:**, pp. 259–268, 1992.

[5] M. Moeller, E. Esser, S. Osher, G. Sapiro, and J. Xin, "A convex model for blind hyperspectral unmixing," *UCLA CAM Report* **10-71**, 2010.

[6] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite-element approximations," *Comp. Math. Appl.* **vol. 2**, pp. 17–40, 1976.

[7] D. Takhar, J. Laska, M. Wakin, M. Duarte, D. Baron, S. Sarvotham, K. Kelly, and R. Baraniuk, "A new compressive imaging camera architecture using optical-domain compression," *IS&T/SPIE Computational Imaging IV* **6065**, 2006.

[8] J. Bioucas-Dias and M. Figueiredo, "Two-step algorithms for linear inverse problems with non-quadratic regularization," in *IEEE International Conference on Image Processing – ICIP*, **1**, IEEE.

[9] Y. Xu, W. Yin, Z. Wen, and Y. Zhang, "An alternating direction algorithm for matrix completion with nonnegative factors," *Rice University CAAM Technical Report TR11-03. Invited for submission to Special Issue on Computational Mathematics, Journal of Frontiers of Mathematics in China (Springer)* , 2011.

[10] Y. Zhang, "An alternating direction algorithm for nonnegative matrix factorization," *Rice Technical Report* , 2010.

[11] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Rice University CAAM Technical Report TR10-07* , 2010.

[12] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and Bregman iterative methods for matrix rank minimization," *Mathematical Programming* , pp. 1–33, 2009.

[13] Y. Xu and W. Yin, "Alternating proximal gradient method for nonnegative matrix factorization," *Rice University CAAM Report TR12-10* , 2012.

[14] C. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Computation* **19**(10), pp. 2756–2779, 2007.

[15] H. Kim and H. Park, "Non-negative matrix factorization based on alternating non-negativity constrained least squares and active set method," *SIAM J. Matrix Anal. Appl* **30**(2), pp. 713–730, 2008.

[16] J. Kim and H. Park, "Toward faster nonnegative matrix factorization: A new algorithm and comparisons," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pp. 353–362, IEEE, 2008.

[17] J. Bolte, A. Daniilidis, and A. Lewis, "The Lojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems," *SIAM Journal on Optimization* **17**(4), pp. 1205–1223, 2007.

[18] J. Bolte, A. Daniilidis, A. Lewis, and M. Shiota, "Clarke subgradients of stratifiable functions," *SIAM Journal on Optimization* **18**(2), pp. 556–572, 2007.

[19] Z. Wen, W. Yin, and Y. Zhang, " Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Rice Technical Report* , 2010.

[20] Q. Ling, Y. Xu, W. Yin, and Z. Wen, "Decentralized low-rank matrix completion," *Rice Technical Report* , 2011.

[21] V. Pauca, J. Piper, and R. Plemmons, "Nonnegative matrix factorization for spectral data analysis," *Linear Algebra and its Applications* **416**(1), pp. 29–47, 2006.

[22] Y. Li, W.-C. Shih, Z. Han, and W. Yin, "Oil spill sensor using multispectral infrared imaging via l1 minimization," *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* , 2011.

[23] A. Wagadarihar, R. John, R. Willett, and D. Brady, "Single disperser design for coded aperture snapshot spectral imaging," *Applied Optics* **47**, pp. B44–B51, 1986.

[24] Y. Xu, W. Yin, and S. Osher, "Learning circulant sensing kernels.,"

[25] J. Duarte-Carvajalino and G. Sapiro, "Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization.," *IEEE Transactions on Image Processing* **18**(7), pp. 1395–1408, 2009.

[26] J. Yang and Y. Zhang, "Alternating direction algorithms for ℓ_1-problems in compressive sensing," *Rice University CAAM Technical Report TR09-37* , 2009.

[27] M. Aharon, M. Elad, and A. Bruckstein, "k-svd: An algorithm for designing overcomplete dictionaries for sparse representation.," *IEEE Transactions on Signal Processing* **54**(11), pp. 4311–4322, 2006.

[28] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics.," *Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001. Proceedings.* **2**, pp. 416–423, 2001.