Earth Science Research; Vol. 1, No. 2; 2012
ISSN 1927-0542 E-ISSN 1927-0550
Published by Canadian Center of Science and Education

Unmixing and Target Recognition in Airborne Hyper-Spectral Images

Amir Averbuch[1], Michael Zheludev[1] & Valery Zheludev[1]

[1] School of Computer Science, Tel Aviv University, Tel Aviv, Israel

Correspondence: Amir Averbuch, School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel. Tel: 972-54-569-4455. E-mail: amir@math.tau.ac.il

Received: May 25, 2012 Accepted: June 16, 2012 Online Published: July 6, 2012

doi:10.5539/esr.v1n2p200 URL: http://dx.doi.org/10.5539/esr.v1n2p200

Abstract

We present two new linear algorithms that perform unmixing in hyper-spectral images and then recognize their targets whose spectral signatures are given. The first algorithm is based on the ordered topology of spectral signatures. The second algorithm is based on a linear decomposition of each pixel's neighborhood. The sought after target can occupy sub- or above pixel. These algorithms combine ideas from algebra and probability theories as well as statistical data mining. Experimental results demonstrate their robustness. This paper is a complementary extension to Averbuch & Zheludev (2012).

Keywords: hyper-spectral processing, target recognition, sub- and above pixel, unmixing, dimensionality reduction, diffusion maps

1. Introduction

1.1 Data Representation and Extraction of Spectral Information

We assume that a hyper-spectral signature of a sought after material is given. In many applications according to Winter (1999), a fundamental processing task is to automatically identify pixels whose spectra coincide with the given spectral shape (signature). This problem raises the following issues: How the measured spectrum of a ground material is related to a given "pure" spectrum and how to compare between them to determine if they are the same? Spatial and spectral sampling produce a 3D data structure referred to as a data cube. A data cube can be visualized as a stack of images where each plane on the stack represents a single spectral channel (wavelength). The observed spectral radiance data, or the derived surface reflectance data, can be viewed as a scattering of points in a K -dimensional Euclidean space $\mathbb{R}^K$ where K is the number of spectral bands (wavelengths). Each spectral band is assigned to one axis. All the axes are mutually orthogonal. Therefore, the spectrum of each pixel can be viewed as a vector $x=(x_1,\dots,x_K)$ where its Cartesian coordinates x_i are either radiance or reflectance values at each spectral band. Since $x_i \geq 0, i=1,\dots,K$, then the spectral vectors lie inside a positive cone in $\mathbb{R}^K$. Changes in the illumination level can change the length of the spectral vector but not its, which is related to the shape of the spectrum. When targets are too small to be resolved spatially or when they are partially obscured or of an unknown shape, as shown in Winter (1999), then the detection has to rely on the available spectral information. Unfortunately, a perfect fixed spectrum for any given material does not exist.

In agreement with Winter (1999), spectra of the same material are probably never identical even in laboratory experiments. This is due to variations in the material surface. The variability amount is even more profound in remote sensing applications because of the variations in atmospheric conditions, sensor noise, material composition, location, surrounding materials and other contributing factors. As a result, the measured spectra, which correspond to pixels with the same surface type, exhibit an inherent spectral variability that prevents the characterization of homogeneous surface materials by unique spectral signatures.

Another significant complication arises from the interplay between the spatial resolution of the sensor and the spatial variability present in the observed ground scene. According to Winter (1999), a sensor integrates the radiance from all the materials within the ground surface that are "seen" by the sensor as a single image pixel. Therefore, depending on the spatial resolution of the sensor and the distribution of surface materials within each ground resolution cell, the result is a hyper-spectral data cube comprised of "pure" and "mixed" pixels, where a pure pixel contains a single surface material and a mixed pixel contains multiple (superposition of) materials.

A linear mixing model is the most widely used spectral mixing model. It assumes that the observed reflectance spectrum of a given pixel is generated by a linear combination of a small number of unique

constituent known as endmembers. This model is defined with constraints in the following way (Harsanyi & Chang, 1994):

$$x=\sum_{k=1}^{M}a_k s_k+w=Sa+w,\quad \sum_{k=1}^{M}a_k=1,\ \text{additivity constraint},\ a_k\geq 0\ \ \text{positivity constraint} \tag{1}$$

where $s_1,\ldots,s_M$ are the M endmember spectra that assumed to be linearly independent, $a_1,\ldots,a_M$ are the corresponding abundances (cover material fractions) and w is an additive-noise vector.

1.2 Outline of the Algorithms to Identify Target with Known Spectra

The new methods in this paper achieve targets identification with known spectra. Target identification in hyper-spectral has the following consecutive steps:

1) Finding suspicious points: there are points whose spectra are different in any norm from the spectra of the points in its neighborhood. This is also called anomaly detection;

2) Extracting from the suspicious points the spectra of the independent components (unmixing) where one of them is the target that its spectrum fits the given (sought after) spectrum.

We assume that spectra of different materials are statistically dependent and the difference between them occurs from the behavior of the first and second derivatives in some sections in the spectrum. If they are statistically independent, then all the related work such as Maximum Likelihood (ML) and Geometrical (MVT, PPI and N-FINDR) work well.

The experiments in this paper were performed on three real hyper-spectral datasets, which were measured as reflectance, titled: “desert”, “city” and “field” which were acquired by the Specim camera SPECIM camera (2006) located on a plane. Their properties with a display of one waveband per dataset are given in Figures 1-3.

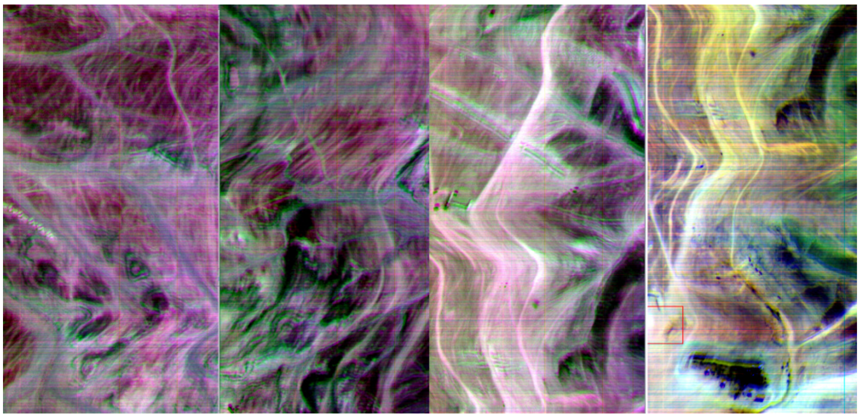


Figure 1. The dataset “desert” is a hyper-spectral image of a desert place taken from an airplane flying 10,000 feet above sea level. The resolution is 1.3 meter/pixel, 286×2640 pixels per waveband with 168 wavebands

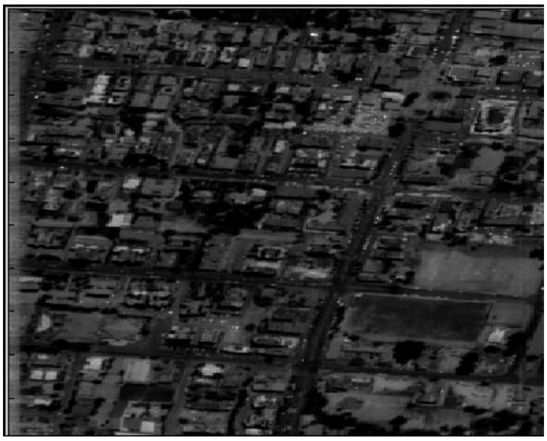


Figure 2. The dataset “city” is a hyper-spectral image of a city taken from an airplane flying 10,000 feet above sea level. The resolution is 1.5 meter/pixel, 294×501 pixels per waveband with 28 wavebands

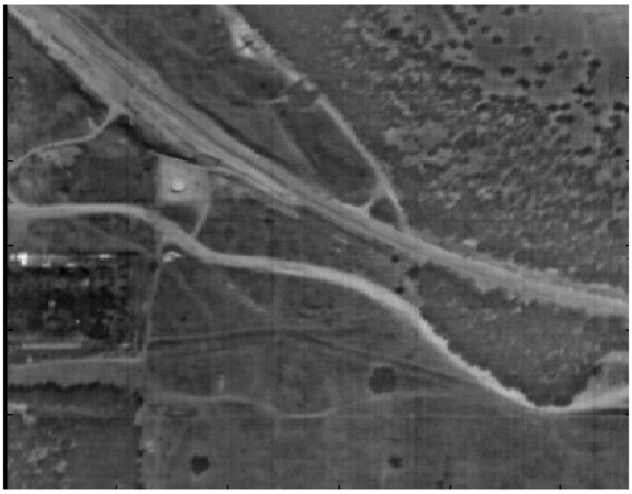


Figure 3. The dataset “field” is a hyper-spectral image of a field taken from an airplane flying 9,500 feet above sea level. The resolution is 1.2 meter/pixel, 286×300 pixels per waveband with 50 wavebands

The paper has the following structure: Section 2 describes the related work. The two algorithms, which are described in this paper, are compared with the performance of the orthogonal subspace projection (OSP) algorithm. Section 3 presents an algorithm that identifies the target's spectrum where the target occupies at least a whole pixel. This method assumes that the target's spectrum is distorted by atmospheric conditions and noised. Section 4 presents an unmixing method that is based on neighborhood analysis of each pixel. This method can also be used for detecting a subpixel target. This algorithm contains two parts. In the first part, suspicious points are discovered. The algorithm is based on the properties of neighborhood morphology and on the properties of the Diffusion Maps (DM) algorithm Coifman & Lafon (2006). The second part unmixes the suspicious point. It is based on the application of DM to the linear span of the neighboring background spectra. The appendix describes the Diffusion Maps algorithm for dimensionality reduction.

2. Related Work

Up-to-date overview on hyper-spectral unmixing is given in Bioucas-Dias & Plaza (2010; 2011). The challenges related to target detection, which is the main focus of this paper, are described in the survey papers Manolakis, Marden, & Shaw (2001), Manolakis & Shaw (2002). They provide tutorial review on state-of-the-art target detection algorithms for hyper-spectral imaging (HIS) applications. The main obstacles in having effective detection algorithms are the inherent variability target and background spectra. Adaptive algorithms are effective to solve some of these problems. The solution provided in this paper meets some of the challenges mentioned in Manolakis & Shaw (2002).

In the rest of this section, we divided the many existing algorithms into several groups. We wish to show some trends but do not attempt to cover the avalanche of related work on unmixing and target detection.

Linear approach: Under the linear mixing model, where the number of endmembers and their spectral signatures are known, hyper-spectral unmixing is a linear problem, which can be addressed, for example, by the ML setup Settle (1996) and by the constrained least squares approach Chang (2003). These methods do not supply sufficiently accurate estimates and do not reflect the physical behavior. Distinction between different material's spectra is conditioned generally by the distinction in the behavior of the first and second derivatives and not by a trend.

Independent component analysis (ICA) is an unsupervised source separation process that finds a linear decomposition of the observed data yielding statistically independent components Common (1994), Hyvarinen, Karhunen, & Oja (2001). It has been applied successfully to blind source separation, to feature extraction and to unsupervised recognition such as in Bayliss, Gualtieri, & Cromp (1997), where the endmember signatures are treated as sources and the mixing matrix is composed by the abundance fractions. Numerous works including Nascimento & Bioucas-Dias (2005) show that ICA cannot be used to unmix hyper-spectral data.

Geometric approach: Assume a linear mixing scenario where each observed spectral vector is given by $r = x + n = M\gamma a + n,\ \gamma a = s,$ where r is an L vector (L is the number of bands), $M = \left[m_1,\dots,m_p\right]$ is the mixing matrix (m_i denotes the ith endmember signature and p is the number of endmembers present in the sensed area), $s = \gamma a$ (γ is a scale factor that models illumination variability due to a surface topography), $a = \left[a_1,\dots,a_p\right]^T$ is the abundance vector that contains the fractions of each endmember (T denotes a transposed vector) and n is the system's additive noise. Owing to physical constraints, abundance fractions are nonnegative and satisfy the so-called positivity constraint $\sum_{k=1}^{p} a_k = 1$. Each pixel can be viewed as a vector in a L -dimensional Euclidean space, where each channel is assigned to one axis. Since the set $\left\{a \in \mathbb{R}^p : \sum_{k=1}^{p} a_k = 1, a_k > 0, \forall k\right\}$ is a simplex, then the set $S_x = \left\{x \in \mathbb{R}^L : x = Ma, \sum_{k=1}^{p} a_k = 1, a_k > 0, \forall k\right\}$ is also a simplex whose vertices correspond to endmembers.

Several approaches Ifarraguerri & Chang (1999), Boardman (1993), Craig (1994) exploited this geometric feature of hyper-spectral mixtures. The minimum volume transform (MVT) algorithm Craig (1994) determines the simplex of a minimal volume that contains the data. The method presented in Bateson, Asner, & Wessman (2000) is also of MVT type, but by introducing the notion of bundles, it takes into account the endmember variability that is usually present in hyper-spectral mixtures.

The MVT type approaches are complex from computational point of view. Usually, these algorithms first find the convex hull defined by the observed data and then fit a minimum volume simplex to it. Aiming at a lower computational complexity, some algorithms such as the pixel purity index (PPI) Boardman (1993) and the N-FINDR Winter (1999) still find the maximum volume simplex that contains the data cloud. They assume the presence of at least one pure pixel of each endmember in the data. This is a strong assumption that may not be true in general. In any case, these algorithms find the set of most of the pure pixels in the data.

Extending subspace approach: A fast unmixing algorithm, termed *vertex component analysis* (VCA), is described in Nascimento & Bioucas-Dias (2005). The algorithm is unsupervised and utilizes two facts: 1) The endmembers are the vertices of a simplex; 2) The affine transformation of a simplex is also a simplex. It works with projected and unprojected data. As PPI and N-FINDR algorithms, VCA also assumes the presence of pure pixels in the data. The algorithm iteratively projects data onto a direction orthogonal to the subspace spanned by the endmembers already detected. The new endmember's signature corresponds to the extreme projection. The algorithm iterates until all the endmembers are exhausted. VCA performs much better than PPI and better than or comparable to N-FINDR. Yet, its computational complexity is between one and two orders of magnitude lower than N-FINDR.

If the image is of size approximately 300×2000 pixels, then this method, which builds linear span in each step, is too computationally expensive. In addition, it relies on “pure” spectra which are not available all the time.

Statistical methods: In the statistical framework, spectral unmixing is formulated as a statistical inference problem by adopting a Bayesian methodology where the inference engine is the posterior density of the random objects to be estimated as described for example in Dobigeon, Moussaoui, Coulon, Tourneret, & Hero (2009), Moussaoui, Carteretb, Briea, & Mohammad-Djafaric (2006), Arngren, Schmidt, & Larsen (2009).

2.1 Orthogonal Subspace Projection (OSP)

The method of orthogonal subspace projection (OSP) for unmixing and target detection is described in Ahmad & Ul Haq (2011), Ahmad, Ul Haq, & Mushtaq (2011), Ren & Chang (2003). We will compare between our method and the method in Ahmad & Ul Haq (2011) that is currently considered to be very effective. According to the notation in Ahmad & Ul Haq (2011), we are given the dataset $X_i = SA + W$ where S is the set of pure signatures, A is the corresponding abundance fractions and W is a white noise matrix. According to the OSP method in Ahmad & Ul Haq (2011), the mixing matrix is found as $A = \left(I - U\left(\widehat{U^T U}\right)U^T\right)\left(U\Sigma U^T\right)$ where U, Σ are a singular matrix and an eigenvalues-matrix, respectively, of the projection matrix to the subspace L of the pure signatures and $\widehat{U^T U}$ is the pseudo inverse of U . The creation of the subspace L is described in Ren, H., & Chang, C. I. (2003), pp. 1236.

We present the results from target detection by the application of the OSP method with a given target signature s and compare them to our method. The targets in the scene are detected via the application of the OSP method on multipixels, which contain the dominant coefficient from the matrix A , corresponding to target signature s .

2.2 Linear Classification for Threshold Optimization

According to Cristianini & Shawe-Taylor (2000), a binary classification is frequently performed by using a real-valued function $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$ in the following way: the input $x = \left(x_1, \ldots, x_n\right)^T$ is assigned to a positive class if $f\left(x\right) \geq 0,$ otherwise, to a negative class. We consider the case where $f\left(x\right)$ is a linear function of x with the parameters w and b such that

$$f\left(x\right) = \langle w \cdot x\rangle + b = \sum_{i=1}^{n} w_i x_i + b \tag{2}$$

where $\left(w, b\right) \in \mathbb{R}^n \times \mathbb{R}$ are the parameters that control the function. The decision rule is given by $\operatorname{sgn}\left(f\left(x\right)\right).$ w is assumed to be the weight vector and b is the threshold.

Definition 2.1. (Cristianini & Shawe-Taylor, 2000)) *A training set is a collection of training examples (data)*

$$S = \left(\left(x_1, y_1\right), \ldots, \left(x_l, y_l\right)\right) \subseteq \left(X \times Y\right)^l \tag{3}$$

where l is the number of examples, $X \subseteq \mathbb{R}^n, Y = \left\{-1, 1\right\}$ is the output domain.

The Rosenblatt's Perceptron algorithm (Cristianini & Shawe-Taylor, 2000; Burges, 1998; pages 12 and 8, respectively) creates an hyperplane $\langle w \cdot x\rangle + b = 0$ with respect to a training set S . It creates the best linear separation between positive and negative examples via minimization of measurement function of “margin” distribution $\lambda_i = y_i\left(\langle w, x_i\rangle + b\right).$ $\lambda_i > 0$ that implies the correct classification for $\left(x_i, y_i\right).$

The perceptron algorithm is guaranteed to converge only if the training data are linearly separable. A procedure that does not suffer from this limitation is the Linear Discriminant Analysis (LDA) via Fisher's discriminant functional Cristianini & Shawe-Taylor (2000). The aim is to find the hyperplane $\left(w, b\right)$ on which the projection of the data is maximally separated. The cost function (the Fisher's function) to be optimized is:

$$F = \frac{m_1 - m_{-1}}{\sigma_1^2 - \sigma_{-1}^2} \tag{4}$$

where m_i and σ_i are the mean and the standard deviation, respectively, of the function output values $P_i = \left\{\langle w \cdot x_j\rangle + b : y_j = i\right\}$ for the two classes $P_i, i = 1, -1.$

Definition 2.2. (Cristianini & Shawe-Taylor, 2000) *The dataset S from Equation 3 is linearly separable if the hyperplane $\langle w \cdot x\rangle + b = 0,$ which is* obtained *via the LDA algorithm (*Cristianini & Shawe-Taylor, 2000)*, correctly classifies the training data. It means that $\lambda_i = y_i\left(\langle w, x_i\rangle + b\right) > 0, i = 1, \ldots, l.$ In this case, b is the separation threshold. If $\lambda_i < 0$ then the dataset is linearly inseparable.*

Definition 2.3. *The vector $x \in \mathbb{R}^n$ is isolated from the set $P = \left\{p_1, \ldots, p_k\right\} \subseteq \mathbb{R}^n$ if the training set $S = \left(\left(x, 1\right), \left(p_1, -1\right), \ldots \left(p_k, -1\right)\right)$ is linearly separable according to definition 2.2. In this case, the absolute value of b is the separation threshold.*

Suppose that we have a set $S=\{x_1,\ldots,x_n\}$ of n samples. First, we want to partition the data into exactly two disjoint subsets S_1 and S_{-1}. Each subset represents a cluster. The solution is based on the K-means algorithm (Duda, Hart, & Stork, 2001). K-means maximizes the function $J(e)$ where e is a partition. The value of $J(e)$ depends on how the samples are grouped into clusters and on the number of clusters (see Duda, Hart, & Stork, 2001)

$$J(e)=tr\left(S_W^{-1}S_B\right) \tag{5}$$

where $S_W=\sum_{i=1}^{l}\sum_{x\in S_i}(x-m_i)(x-m_i)^T$ is an "within-cluster scatter matrix" (Duda, Hart, & Stork, 2001), l is the classes, S_i are the classes and m_i are the center of each class. S_B is called "between-cluster scatter matrix" (Duda, Hart, & Stork, 2001), where $S_B=\sum_{i=1}^{l}n_i(m_i-m)(m_i-m)^T$, n_i is the cardinality of a class and m is the center for all the dataset.

Definition 2.4. *Let (w,b) be the best separation for the set $S=\{x_1,\ldots,x_n\}\subseteq\mathbb{R}^n$ via K-means and Fisher's discriminant analyzes* Cristianini and Shawe-Taylor (2000), Burges (1998). *(w,b) is called the Fisher's separation and b the Fisher's threshold for the data P.*

When a dataset is separable? One criterion is when $m_1-m_{-1}>\max\left(diam(P_1),diam(P_{-1})\right)$, where the notation in Equation 4 is used. $diam$ is defined as $diam(P)=\max\left\{\|x-y\|_{L_2}\ :x,y\in P\right\}$.

Another criterion is:

Definition 2.5. (Duda, Hart, & Stork, 2001) *A dataset is separable if from Equation 5 $J(e_1)<J(e_2)$ where e_1 is the partition and the number of classes is 1 and e_2 is the best partition into two classes. If $J(e_1)\geq J(e_2)$ then the dataset is inseparable and Fisher's separation is incorrect.*

3. Method I: Weak Dependency Recognition (WDR) of Targets That Occupy One or More Pixels

We assume that a target occupies one or more pixels. The process, which determines whether a given target's spectrum and the spectrum of the current pixel are dependent, is described next.

Definition 3.1. *Two discrete functions Y_1 and Y_2 are weakly dependent if there exists a permutation σ of the coordinates that provides monotonic order for the values of $\sigma(Y_1)$ and $\sigma(Y_2)$.*

Let T be a given target's spectrum and P is the pixel's spectrum. We assume that the spectra of T and P are discrete vectors. In general, we assume that T and P are normalized and centralized. The following hypotheses are assumed:

H_0: T and P are weakly dependent.

H_1: T and P are not weakly dependent.

3.1 Hypotheses Check

We find an orthogonal transformation Π that permutes the coordinates of T into a decreasing order. This permutation Π is applied to P and T. We get that $P_1=\Pi(P),T_1=\Pi(T)$ where T_1 is monotonic. If H_0 holds, which means that T and P are weakly dependent, then the values of P_1 are either monotonic decreasing or increasing and the first and second derivatives of P_1 are close to zero - see Figure 4 (left). Otherwise, H_1 holds and P_1 has an oscillatory behavior - see Figure 4 (right). In addition, P_1 has a subset of coordinates whose first and second derivatives have an oscillatory behavior - see Figure 4 (right).

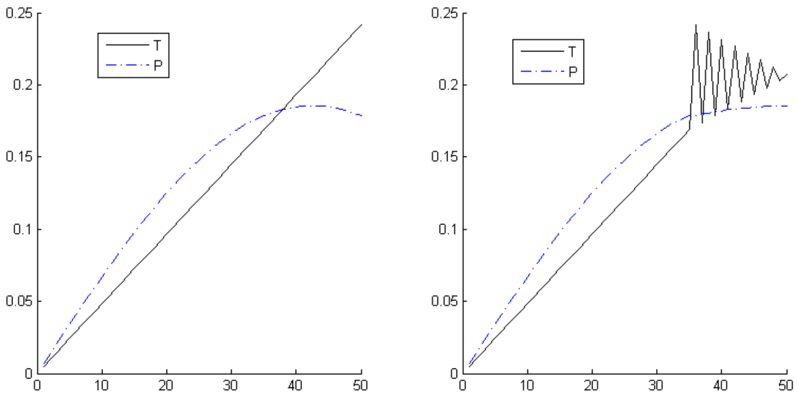


Figure 4. The x - and the y -axes are the wavebands and their reflectance values, respectively. The spectra are represented after the application of the permutation to the coordinates, which permutes T into a monotonic deceasing order. Left: Weak dependency between T and P , Right: No weak dependency between T and P

If the permutation of the coordinates of P provides that their values are either decreasing or increasing monotonically, then the first and second derivatives of P have a minimal norm. This is another criterion for deciding who has weak dependency.

Let $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$. The norm is defined as $\|x\|_\infty = \max_i (|x_i|)$.

Definition 3.2. *Let π be an orthogonal transformation that permutes the coordinates of T into a decreasing order. Denote the second derivative of a vector X by X_2 . Define the mapping $\Omega : \mathbb{R}^n \to \mathbb{R}^+$ such that* $\Omega(x) = \|(\pi(X))_2\|_\infty$.

Let $\{X_1, \ldots, X_\omega\}$ be a dataset of spectra from all the pixels in the scene. Denote $Y_i = \Omega(X_i)$. The dataset $\{Y_1, \ldots, Y_\omega\}$ can be classified as:

1) The set $\{Y_1, \ldots, Y_\omega\}$ is separable according to definition2.5.

2) The set $\{Y_1, \ldots, Y_\omega\}$ is inseparable according to definition 2.5.

In the first case, (w, b) is the best separation for the set $S = \{Y_1, \ldots, Y_\omega\} \subseteq \mathbb{R}^+$ according to definition 2.4 and b is the Fisher's threshold for this separation. Then, the set $\{i : Y_i < b\}$ is the set of targets. In the other case, there are no targets in the scene.

The flow of the WDR algorithm is given in Figure 5.

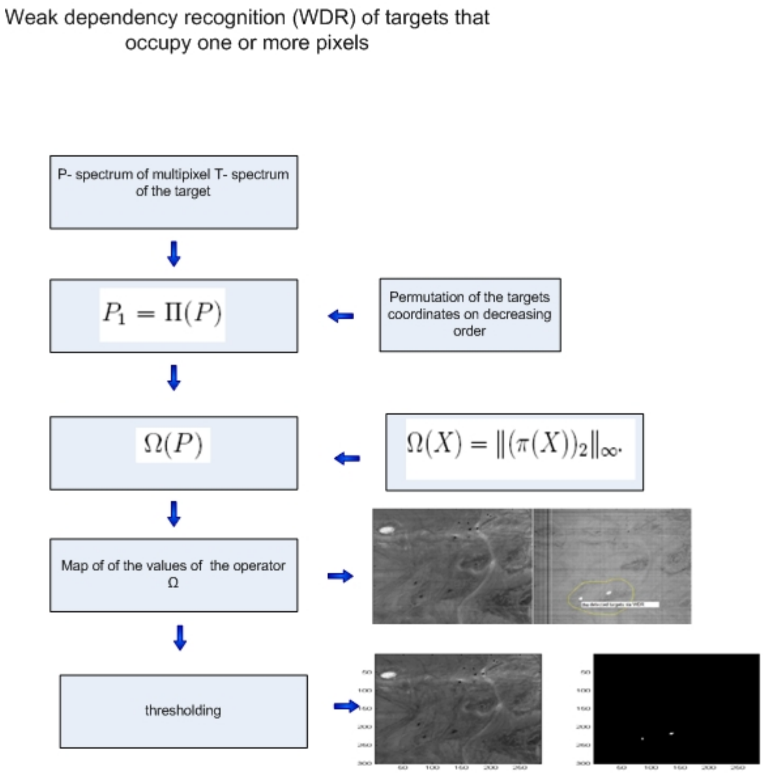

Weak dependency recognition (WDR) of targets that occupy one or more pixels
P- spectrum of multipixel T- spectrum of the target
$P_1 = \Pi(P)$
Permutation of the targets coordinates on decreasing order
$\Omega(P)$
$\Omega(X) = \|(\pi(X))_2\|_\infty.$
Map of of the values of the operator Ω
thresholding

Figure 5. The flow of the WDR algorithm

3.2 Experimental Results

Figures 6-8 display the results after the application of the algorithm in section 3.1 to the "desert" image (Figure 1). The yellow lines mark the neighborhood of the detected targets.

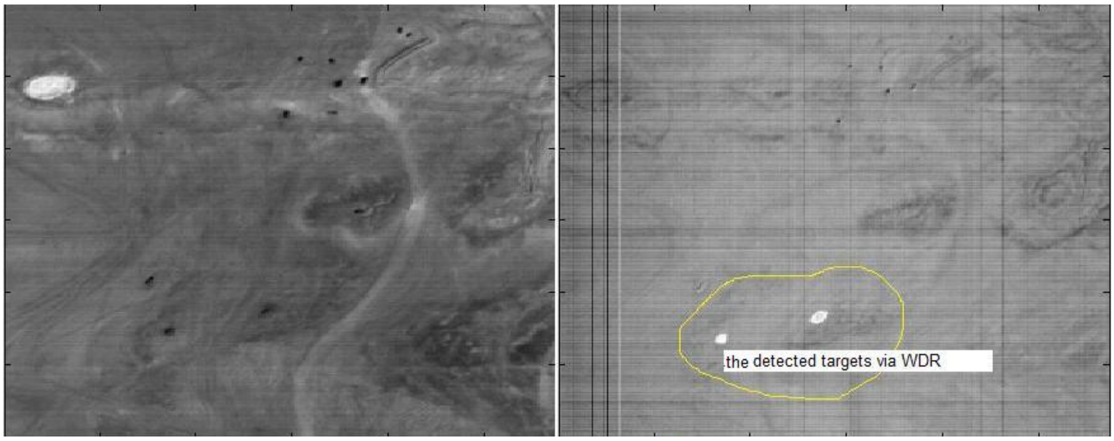


Figure 6. Left: One wavelength part from the original "desert" image (Figure 1). Right: The white points mark the detected targets. The intensity of each pixel in the right side corresponds to the value $-\Omega(X)$ where X the spectrum in the current pixel

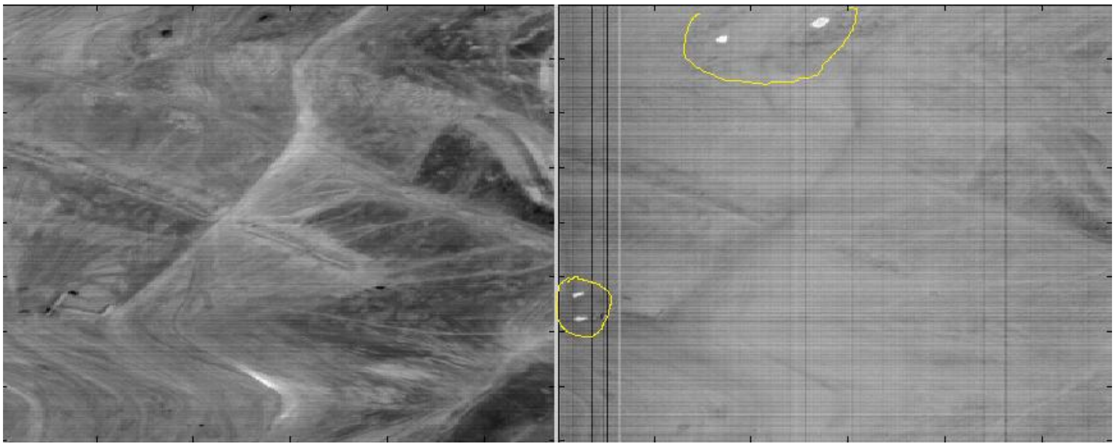


Figure 7. Left: One wavelength part from the original “desert” image (Figure 1). Right: The white points mark the detected targets. The intensity of each pixel in the right side corresponds to the value $-\Omega(X)$ where X is the spectrum in the current pixel

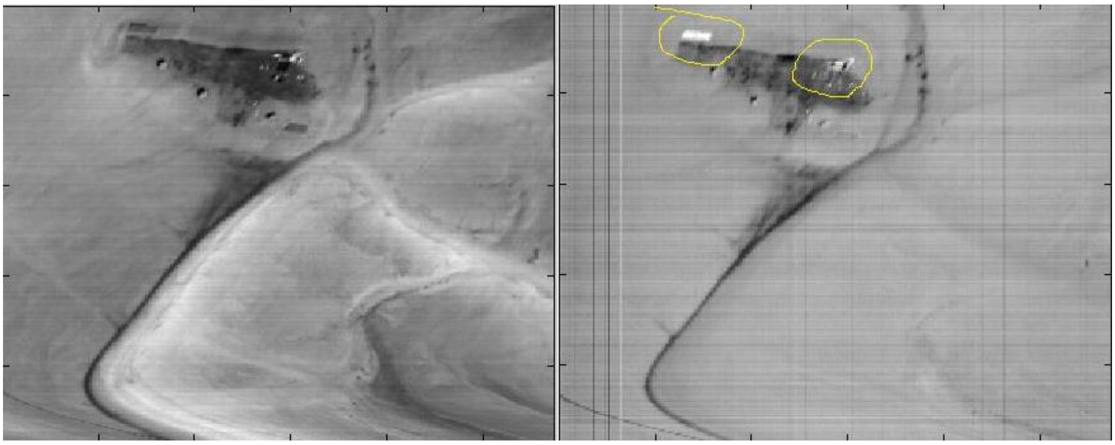


Figure 8. Left: One wavelength part from the original “desert” image (Figure 1). Right: The white points mark the detected targets. The intensity of each pixel in the right side corresponds to the value $-\Omega(X)$ where X is the spectrum in the current pixel

The desert image contains documented targets. The detection of the suspicious points in Figures 6-8 match exactly the known targets.

The point P_1 in Figure 8 is the pattern of the known target's material. Its spectrum is displayed in Figure 4 as a plot of the “target”. Other spectra plots, which were detected by the WDR algorithm in the scenes of Figures 6-8, are classified as “spectra of suspicious points”.

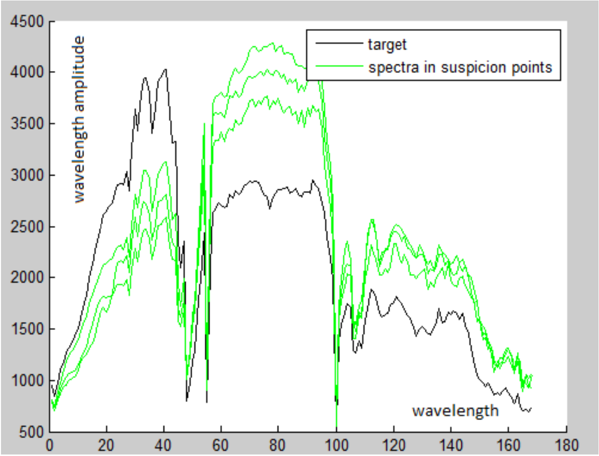


Figure 9. Comparison between the target's spectrum and the suspicious points' spectra in Figures 6-8. The x - and the y -axes are the wavebands and their reflectance values, respectively

The above results were compared next with the OSP algorithm.

3.2.1 Comparison between WDR and OSP Algorithms

In this section, we compare between the performance of WDR and OSP (Ahmad & Ul Haq, 2011) algorithms. The false alarm generated by the OSP algorithm is shown.

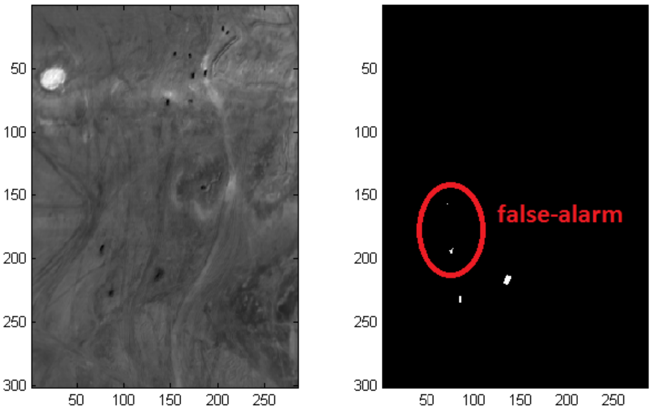


Figure 10. Left: Portion of one wavelength (multipixel) from the original "desert" image (Figure 1). Right: The white points mark the detected targets by the OSP algorithm Ahmad & Ul Haq (2011). The Red circle marks "false-alarm"

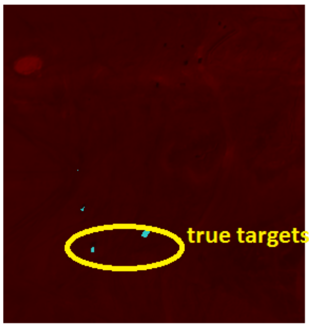


Figure 11. The detected targets by the OSP algorithm Ahmad & Ul Haq (2011) are marked on portion of one wavelength from the "desert" image. OSP generates false-alarm where WDR does not

Figure 12 shows the ROC-curves while comparing two methods for continuously varying the threshold.

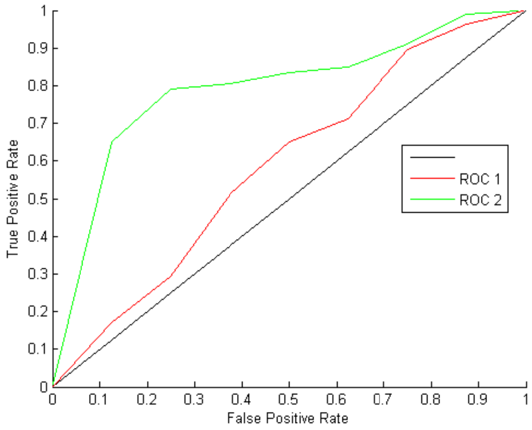


Figure 12. The "ROC-curve" for scene in Figure 11. The red line corresponds to OSP Ahmad & Ul Haq (2011) method. The green line corresponds to WDR method

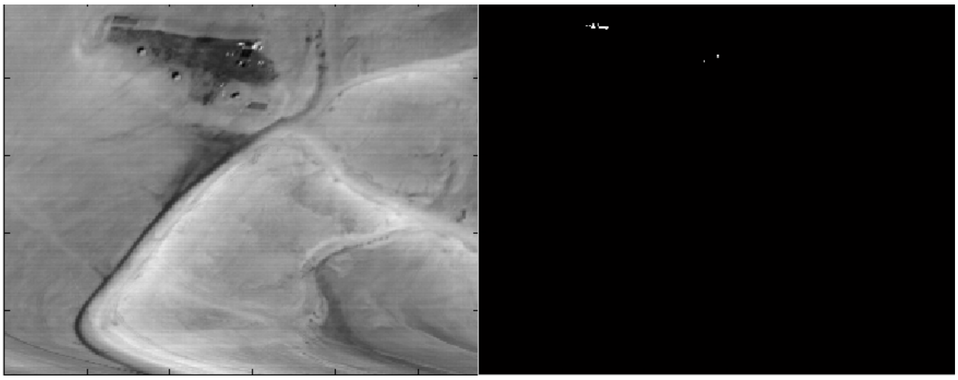


Figure 13. Left: Portion of one wavelength (multipixel) from the original "desert" image (Figure 1). Right: The white points mark the detected targets by the WDR algorithm. The intensity of each pixel in the right side corresponds to the value $-\Omega(X)$ where X is the spectrum in the current pixel

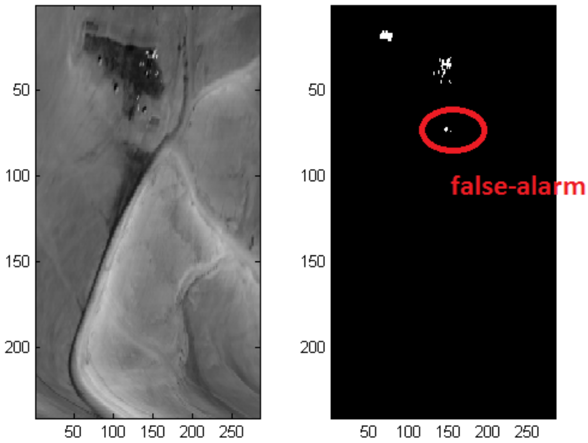


Figure 14. Left: Portion of one wavelength (multipixel) from the original "desert" image (Figure 1). Right: The white points mark the detected targets by the OSP algorithm Ahmad & Ul Haq (2011). The Red circle marks the "false-alarm"

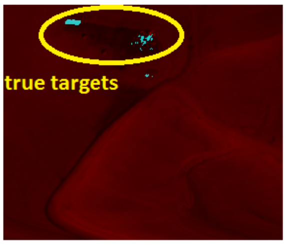


Figure 15. The detected targets by the OSP Ahmad & Ul Haq (2011) algorithm are marked on portion of one wavelength from the “desert” image. OSP produces more false-alarm than WDR

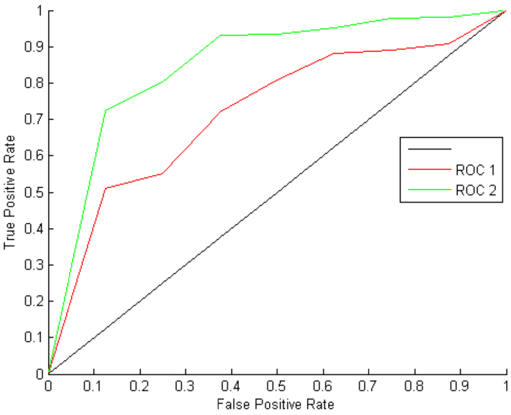


Figure 16. The “ROC-curve” for scene in Figure 15. The red line corresponds to OSP Ahmad & Ul Haq (2011) method. The green line corresponds to WDR method

4. Method II: Unmixing by Examining the Neighborhood of a Suspicious Point (UNSP)

In this section, we provide an algorithm that detects subpixel targets. The idea of the algorithm is close to Chang, Zhao, Althouse, & Pan (1998), Harsanyi & Chang, (1994). But in contrast to them, we construct a different projection. They project the data into linear subspace of the known targets. Our mapping projects the dataset into an orthogonal complement of the background of each pixel. The local model of the background is based on the morphological structure of the pixel's neighborhood. This yields better anomaly (suspicious point) detection.

The UNSP algorithm introduces the parameter m which is the neighborhood's size that we use in the processing. For ease of notation, a square of $m = 2m_1 + 1$ pixels on each side with a center at the pixel X is called the m -neighborhood of the pixel X, denoted by $\Omega_m(X)$, where m_1 is the radius of this neighborhood as displayed in Figure 17.

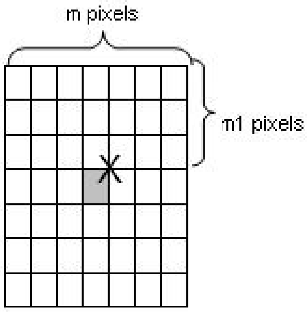


Figure 17. The m -neighborhood of the pixel X denoted by $\Omega_m(X)$

A connected component is a set of pixels in which any two pixels are connected to each other. This connection means that there exists a path between two pixels. The path is a sequence of pixels such that for each of its pixels the next pixel is adjacent to it either horizontally or vertically as we see in Figure 18.

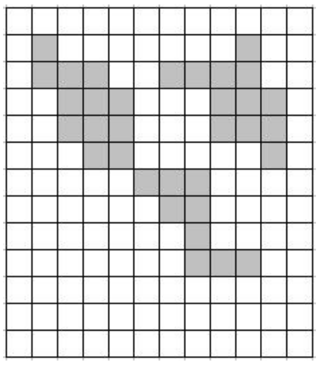


Figure 18. The morphological structure of the neighborhood is represented by three connected components

Consider the spectra from different materials which are present in a hyper-spectral image. Usually there is high correlation between these spectra in real situations. For example, Figure 19 displays the spectra from three different materials.

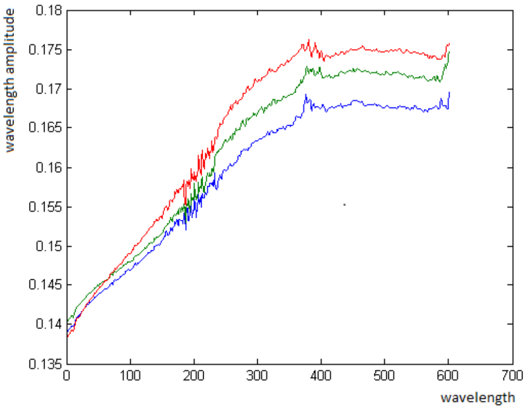


Figure 19. Spectra of three different materials

To reduce the correlation between spectra, we use the first derivative of the spectra. We denote the first derivative of the spectrum of the pixel X by $d(X)$ and it is called the d -spectrum of the pixel X.

We assume that the pixels, which contain a target (as a subpixel or as a whole pixel), represent one connected component that occupies less than half of the m -neighborhood of some pixel. T is the given target's spectrum.

The UNSP algorithm has two steps: Detection of suspicious points and extraction of the target spectrum from the suspicious point (unmixing).

4.1 The Morphology-Filter: Detection of Suspicious Points via Neighborhood Morphology

The following hypotheses are assumed:

H_0: Y is a suspicious point.

H_1: Y is not a suspicious point.

Hypotheses check: The indices of $\Omega_m(Y)$ are constructed in the following way. A pixel located in row i and column j is denoted by $p_{ij}, i, j = 1, \ldots, m$ - see Figure 20. For example, the central pixel is $Y = p_{m_1+1, m_1+1}$ where m_1 is the of the neighborhood's radius.

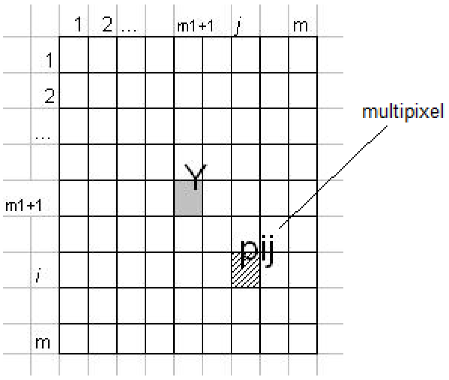


Figure 20. The indices of a pixel

Denote by $S=\{p_{ij}\}_{i,j=1,\dots,m}$ the set of multipixels (multipixel means all the wavelengths that belong to this pixel) in the current neighborhood. Consider the mapping $\Omega:S\to\mathbb{R}$ such that $\Omega(p_{ij})=corr\left(d\left(p_{ij}\right),d(Y)\right)$ where $corr\left(d\left(p_{ij}\right),d(Y)\right)$ is the correlation coefficient between the vectors $d\left(p_{ij}\right)$ and $d(Y)$. Denote $\hat{S}=\Omega(S)=\left\{\Omega\left(p_{ij}\right)\right\}_{i,j=1,\dots,m}\subseteq[-1,1]$.

The set $\hat{S}$ can be in one of two cases:

1) $\hat{S}$ is inseparable according to definition 2.5. This means that the pixels, which are correlated with the target, are inseparable from the other pixels;

2) $\hat{S}$ is separable according to definition 2.5. This means that the pixels, which are correlated with the target, are separated from the other pixels.

If we are in case 1, then Y is not a suspicious point. If we are in case 2, assume that ψ is the first cluster closest to 1. According to definition 2.4, (w,b) provides the best separation. It separates the set Ψ from the other points where b is the Fisher's threshold for this separation. Then, ψ can be represented as $\Psi=\left\{p_{ij}:corr\left(d\left(p_{ij}\right),d(Y)\right)>b\right\}$.

If the set Ψ represents two or more connected components, then Y is also not a suspicious point. If $Y\notin\Psi$ then Y is also not a suspicious point. Therefore, H_1 holds. In other words, if Y is a suspicious point, then Ψ is a set of pixels that intersects with the target and this set of correlated points is concentrated around the central point Y. Here and below, we assume that a correlated point is a pixel whose d -spectrum and $d(Y)$ are correlated with the correlated coefficient that is greater than Fisher's threshold b.

Let N_1 be the neighborhood $\Omega_{m-2}(Y)$. N_1 is called the internal square. Let $N_2=\Omega_m(Y)\setminus N_1$. N_2 is called the external square. They are visualized in Figure 21.

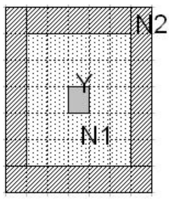


Figure 21. N_1 is the internal square and N_2 is the external square

Assume that Δ is the set of all pixels p_{ij}, which are bounded by the external square with correlation coefficients $corr\left(d\left(p_{ij}\right),d(Y)\right)$, which are associated with the current neighborhood that are less than the Fisher's threshold b. Each pixel in Δ is treated as a vector (multipixel) where its entries are spread all over the wavelengths. The d -spectra of this vector is denoted by v_s where s is one of the $(i,j)\in\Delta$. The set of all these vectors is denoted by V. This is the set of all the d -spectra that belong to Δ. If $|\Delta|=s$ then $V=\{v_1,\dots v_s\}$.

In order to derive the d -spectrum of some material in the central pixel, the background around the central pixel has to be removed. For that, we construct an orthogonal projection ρ, which projects all the d -spectra onto the orthocomplement of the linear span where the background of the d -spectra is located. If the d -spectrum of the central pixel $d(Y)$ does not belong to this linear span, then this projection extracts an orthogonal component of $d(Y)$ which does not mix with the background of the d -spectrum. For example, if $d(Y) = d_1 + d_2$ where d_1 belongs to the linear span generated by the background of the d -spectrum and d_2 belongs to the orthocomplement of this span. Then, after projection we obtain $\rho(d(Y)) = \rho(d_2)$ which does not correlate with the background of the d -spectrum. Hence, the background influence is removed by this projection.

Now, we formalize the above. Assume the matrix E is associated with the vectors $v_1,\ldots,v_s$ where $E(i,j) = v_i \cdot v_j \in \Delta$. Assume that T_e is the Fisher's threshold, which separates between the big and small absolute values of the eigenvalues of the matrix E . In some cases, T_e can separate between zero and nonzero eigenvalues. The eigenvectors associated with the eigenvalues, which are smaller than T_e , generate the eigensubspace, which is the orthocomplement of the linear span of the principal directions of the set V . Denote this orthocomplement by C .

Throughout this paper, we assume that in our model the spectrum of any pixel X consists of three components:

1) The spectrum of the material M is different from its background;

2) The spectrum of the background was generated from a linear combination of spectra of pixels from the X -neighborhood;

3) Random noise is present.

The same model is true for the d-spectra $P^{'} = \tau M^{'} + L(v_1,\ldots,v_s) + N$ where $P^{'} = d(Y)$, $M^{'}$ is the d -spectrum of the material $M, \tau \in [0,1]$ is the portion of the material M in Y , N is a random noise and $L(v_1,\ldots v_s)$ is a linear combination of the vectors $v_1,\ldots v_s$.

If the correlated points concentrate around Y , then these points consist of the same material as Y . If the uncorrelated points do not contain this material then they belong to the background. Consider the orthogonal projection operator ρ . This operator projects vectors onto the orthocomplement C . The vector $\rho(L(v_1,\ldots,v_s))$ is approximated to be a zero vector. Thus, this orthogonal projection removes from the d -spectrum of $d(Y)$ the influence of the background.

Let $T^{'}$ be the given d -spectrum of the target. If the correlation coefficient of $\rho(P^{'})$ and $\rho(T^{'})$ is greater than the correlation coefficient of $P^{'}$ and $T^{'}$, then Y is a suspicious point, M is the target, $T^{'} = M^{'}$ and H_0 holds.

4.2 Detection of Outliers *within a Single Testing Cube*

In section 4.1, we presented how to detect suspicious points. There is another way to do it. An alternative detection method uses dimensionality reduction by the application of the Diffusion Maps (DM) algorithm Coifman & Lafon (2006) and a nearest-neighbor scheme. The DM is a non-linear algorithm for dimensionality reduction.

Assume, we are given a data cube **D** of size $X \times Y \times Z$ where X and Y are the spatial dimensions and Z is the wavebands. We define a small testing cube **d** of size $v \times h \times Z, v \ll X, h \ll Y$ which is included in the hyper-spectral data cube **D**.

4.2.1 Dimensionality Reduction by DM Application

Assume that a sliding testing cube **d**, pointed by the arrows in Figure 22, is moving by ironing each time a different fragment in the data cube **D** described in Figure 2. Section 4.3 describes in details how the testing cube **d** moves.

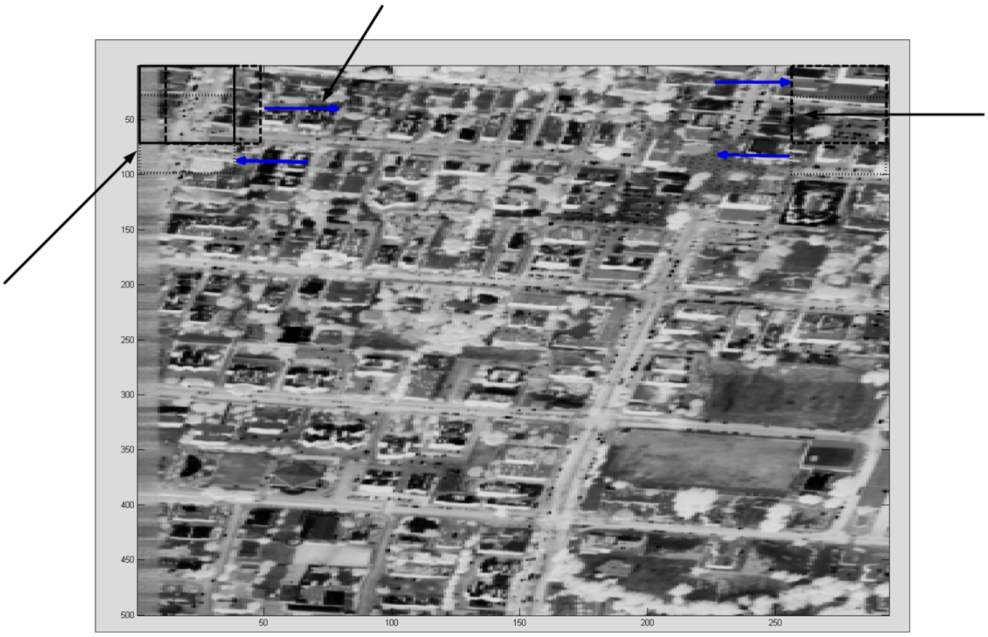
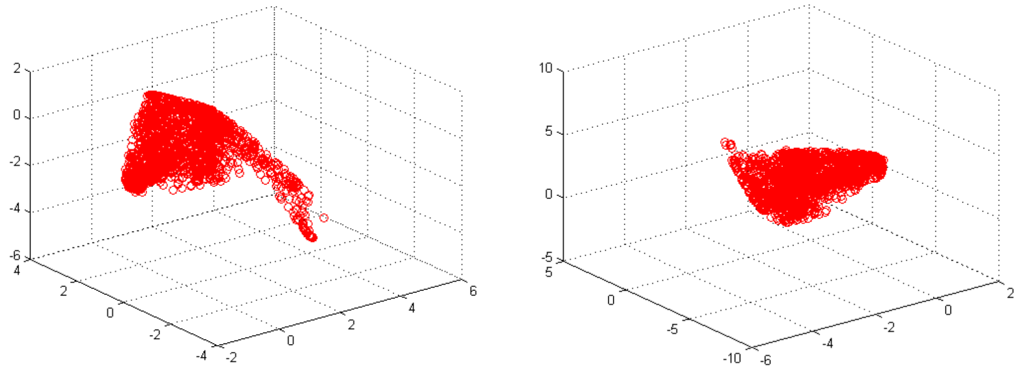


Figure 22. An urban scene of size 294 × 501 (from the "city" in Figure 2) with different locations of the sliding testing cube **d.** The arrows point to these locations

The sliding testing cube **d** contains $N = vh$ multipixels each of which comprises Z wavebands. Typically, v and h are in the range 30-50, Z is in the range 30-100, $Y \approx 290$. Thus, each of the N data points is a vector $m_i, i = 1, \ldots, N,$ of length Z. We arrange these data points into a matrix **M** of size $N \times Z$.

The next step applies the DM (see the appendix for its description) algorithm to the matrix **M**. It reduces the dimensionality of the data vectors by embedding them into the main eigenvectors of the covariance matrix of the data **M**. This projection reveals the geometrical structure of the data and facilitates a search for singular (abnormal) data points. The data matrix **M** of size $N \times Z$ is mapped onto the eigenvectors of the matrix **P** of size $N \times R, R \ll Z$. Typically, R is in the range 3-5, which is determined by the magnitudes of the corresponding eigenvalues. R is the number of essential eigenvalues of the covariance matrix and it is determined as explained in Coifman & Lafon (2006). Figure 23 displays the embedding on three major eigenvectors of the data from four positions of the sliding testing cube. These are the embeddings onto three major eigenvectors of the covariance matrices.



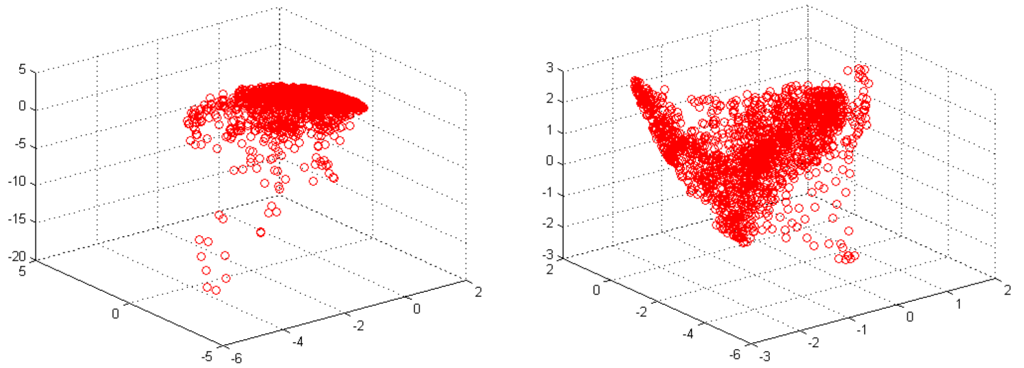


Figure 23. Embedding of the data from different positions of the sliding testing cube on the image in Figure 2 onto three major eigenvectors of the diffusion matrix

We observe that the overwhelming majority of the embedded data points form a dense cloud while a few outliers present. It can be a single point, which lies far away from the rest of points or, more frequently, there exists a small group of points, which are located close to each other but far away from the majority of the cloud. This reflects the situation when an optional target can occupy the area of size from one to several pixels (or even a subpixel). These single or grouped outliers are detected as explained in the next section on outliers detection.

4.2.2 Detection of Grouped Outliers

Assume we are looking for groups of outliers that consist of no more than K members. It is done by the following steps:

1) For each row $r_i, i=1,\dots,N$, of the DM matrix $\tilde{P}$ (see Appendix), calculate its Euclidean distances $d_{i,j}=\|r_i-r_j\|$ to all other rows and sort them in ascending order $d_{i,j}\mapsto s_{i,j}$. Thus, $s_{i,1}=d_{i,j_{i,1}}\le s_{i,2}=d_{i,j_{i,2}}\le\ldots\le s_{i,N-1}=d_{i,j_{i,N-1}}$.

2) Form the matrix $S=\{s_{i,j}\}, i=1,\dots,N, j=1,\dots,N-1$, of the sorted distances and the matrix $J=\{j_{i,k}\}, i=1,\dots,N, k=1,\dots,N-1$, of the corresponding indices.

3) For each row $r_i, i=1,\dots,N$, determine its K nearest neighbors. For this, take the first K columns $\{j_{i,1},\dots,j_{i,K}\}, i=1,\dots,N$, of the index matrix **J**. The corresponding distances are presented in the first K columns $\{s_{i,1},\dots s_{i,K}\}, i=1,\dots,N$, of the matrix **S**. Thus, we have the nearest neighbor index J_K and the distances matrices S_K where both are of size $N\times K$. First, the simplest case $K=2$, which means that we are looking for groups of outliers consisting of no more than two points, is handled.

4) Assume that $\mu_2=\max_i s_{i,2}$ is achieved by $i=i_2$. It means that the distance to the second in order for the nearest neighbor of the i_2-th data point p_{i_2} is the largest among the distances to their second nearest neighbors of all the data points. Restore the coordinates x_2 and y_2 of the data point p_{i_2} (multipixel m_{i_2}) in the data cube **D**. Store the point $P_2(x_2,y_2)$.

5) Find $\max_i s_{i,1}$. Two alternatives are possible:

   a) P_2 is an isolated outlier. It takes place when the maximum $\mu_1=\max_i s_{i,1}$ is achieved by $i=i_2$. It means that the distances from the point P_2 to its first two nearest neighbors is greater than the respective distances of all the other points.

   b) However, it may happen that some point lies close to P_2 while all the others are far apart. It can be interpreted as a pairwise outlier. An indicator of this situation is the fact that the maximum $\mu_1=\max_i s_{i,1}$ is achieved by $i=\tilde{i}_1\ne i_2$. In this case, we add the point $P_1(x_1,y_1)$ closest to the point $P_2(x_2,y_2)$ and regard $\{P_1,P_2\}$ as a pairwise outlier. The index of the point $P_1(x_1,y_1)$ is $i_1=j_{i_2,1}$.

6) While looking for grouped outliers that may contain up to $K>2$ members, we find $i=i_K$, such that $\mu_K=\max_i s_{i,K}$ is achieved by $i=i_K$. Restore the coordinates x_K and y_K of the data point p_{i_K} (multipixel m_{i_K}) in the data cube **D**. Store the point $P_K(X_K,Y_K)$.

7) Find the maximal values in the first $K-1$ columns $\mu_k=\max_i s_{i,k}, k=1,\dots,K-1$, of the distance matrix **S**. The following alternatives are possible:

a) P_K is an isolated outlier. It takes place when all the maxima $\mu_k, k = 1,\ldots,K-1,$ are achieved $i = i_K$. It means that the distances from the point P_K to its first nearest neighbors are greater than the respective distances for all the other points.

b) Grouped outliers arrive when all the maxima $\mu_{k,} k = 2,\ldots,K-1$, except μ_1 are achieved by $i = i_K$. In this case, we add the point $P_1(x_1, y_1)$ that is the closest to the point $P_2(x_2, y_2)$ and regard $\{P_1, P_2\}$ as a pairwise outlier. The index of the point $P_1(x_1, y_1)$ is $i_1 = j_{i_2,1}$.

c) If the maxima in the columns $\mu_{k,} k = L+1,\ldots,K-1, L > 1,$ are achieved by $i = i_K$, while μ_L is achieved by some other $\tilde{i} \neq i_K$, then we have grouped outliers. These outliers $\{P_1,\ldots,P_L,P_K\}$ consist of the point P_K and of the L points closest to P_K. The indices of the points $\{P_1,\ldots,P_L\}$ are $i_1 = j_{i_K,1,\ldots,}i_1 = j_{i_K,L}$, respectively.

We emphasize that, once the upper limit K is given, the number $L+1$ of group members is determined automatically depending on the data within the sliding testing cube **d**. Figure 24 illustrates the grouped detected outliers in the 3-dimensional space of eigenvectors of the data from four positions of the sliding testing cube.

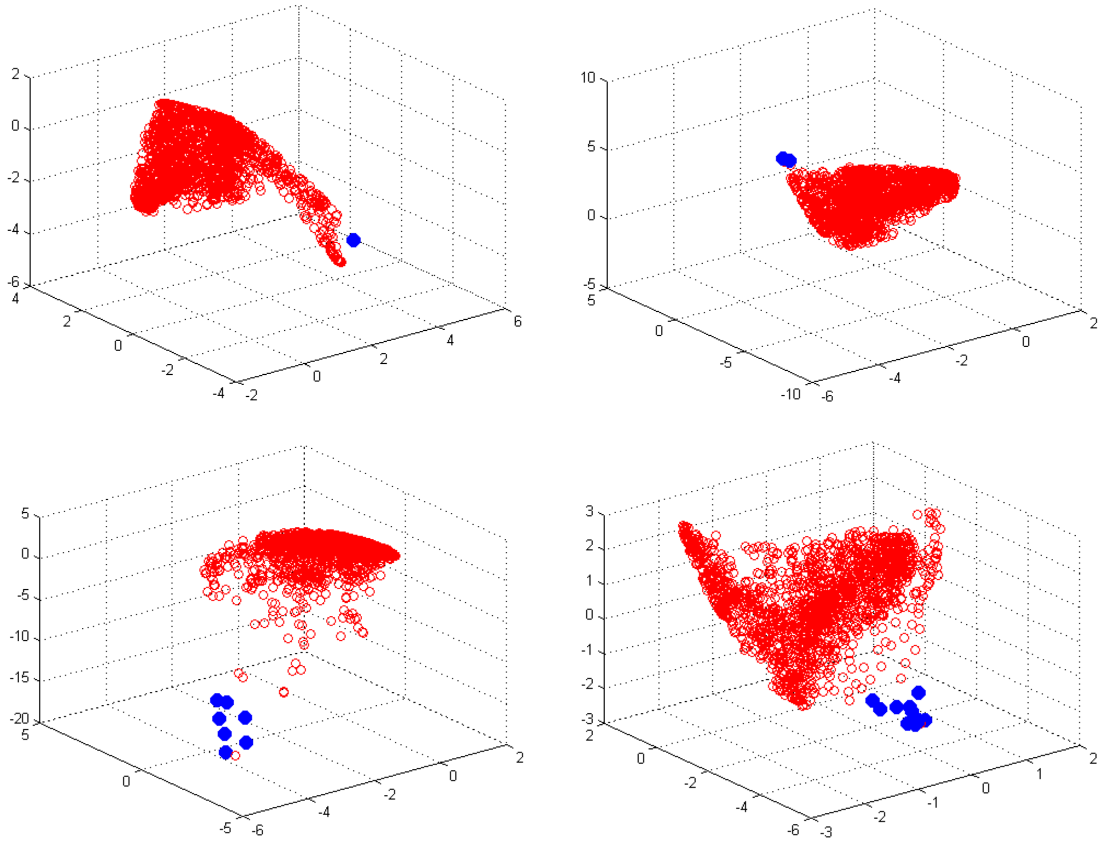


Figure 24. Detection of grouped outliers in data from different positions of the sliding testing cube embedded in the diffusion space

4.3 Detection of Singular Points within the Whole Data Cube

In the section on outliers detection, we described how to find a group of data points (multipixels) within one sliding testing cube, whose geometry differs from the geometry of the majority of the data points. Let $\Lambda^1 = \{P_1^1,\ldots,P_{L^1}^1\}$ be the list of such data points in the sliding testing cube d^1 of size $v \times h \times Z$ located in the upper left corner of the sliding data cube **D** as illustrated by the arrow in Figure 22. The next testing cube d^2 is obtained by a right shift by $\eta \approx h/4$ of d^1. Let $\Lambda^2 = \{P_1^2,\ldots,P_{L^2}^2\}$ be the list of outliers in the cube d^2. Append the list Λ^2 to Λ^1. Because of the vast overlap between the cubes d^2 and d^1, some outliers data points can be common for the lists Λ^2 and

Λ^1. In the united list, these points gain the weight 2. The next right shift produces the sliding testing cube d^3 outliers list Λ^3 s appended to the combined list $\Lambda^1 \cup \Lambda^2$. Again, the common gain weights. We proceed with the right shifts till the right edge of the data cube **D**. Then, the sliding testing cube slides down by $\nu = \nu/4$ and starts η-shifts to the left and so on. As a result, we get a combined list $\Lambda = \bigcup_1^R \Lambda^i$ of outliers, where R is the number of jumps of the testing cube **d** within the sliding data cube **D**. Figure 22 illustrates a route of the cube **d** on the data cube **D**.

It is important that each point P_i in the list Λ is supplied with the weight w_i, which can range from 1 to more than 40. The weight w_i can serve as a singularity measure for the point P_i. A large weight w_i reflects the fact that the point P_i is singular for a big number of overlaps between sliding testing cubes. Thus, it can be regarded as a strong singular point in the sliding data cube **D** and vice versa. Figure 25 illustrates the distribution of the weighted singular points around the data cube D_U of size $500 \times 294 \times\ 64$ from the urban scene displayed in Figure 22 whose source is Figure 2.

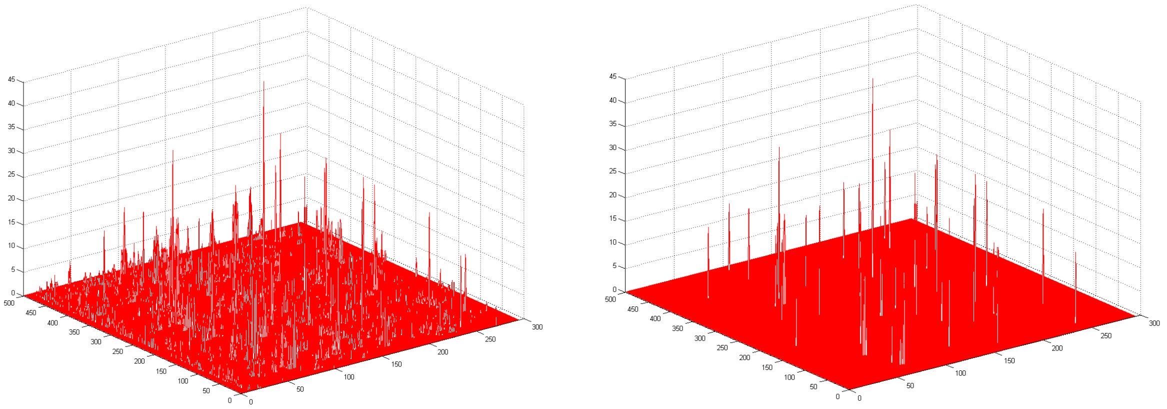


Figure 25. Distribution of the weighted singular points around the data cube D_U. Left: All the singular points. Right: Singular points whose weights exceed 12

4.3.1 Examples of Detected Singular Points

We applied the above algorithm to find singular points in different data cubes. The following figures display a few singular points detected in the data cube D_U.

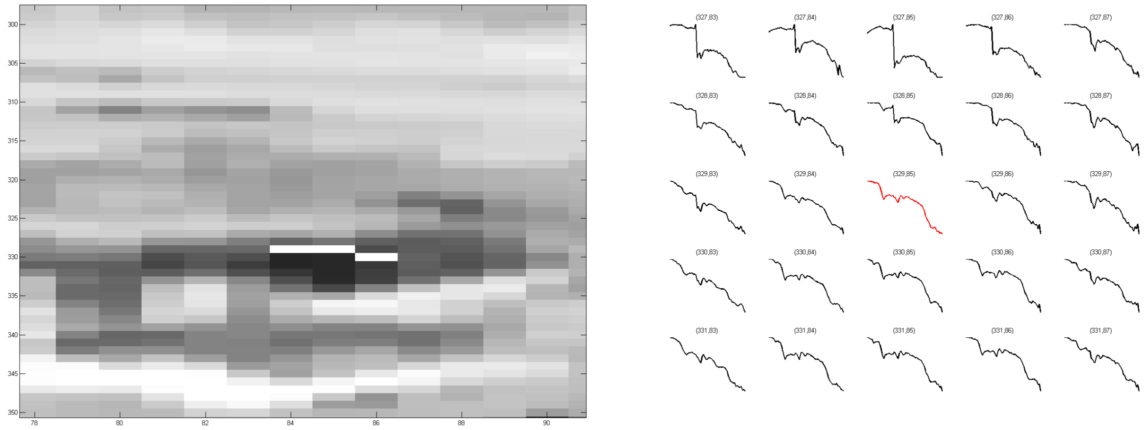


Figure 26. A group of singular points centered around the point $P(329,85)$. Left: Vicinity of the point **P**. Right: Multipixel spectra at the point $P(399,85)$ and the surrounding points. The weight of the data point **P** is 19

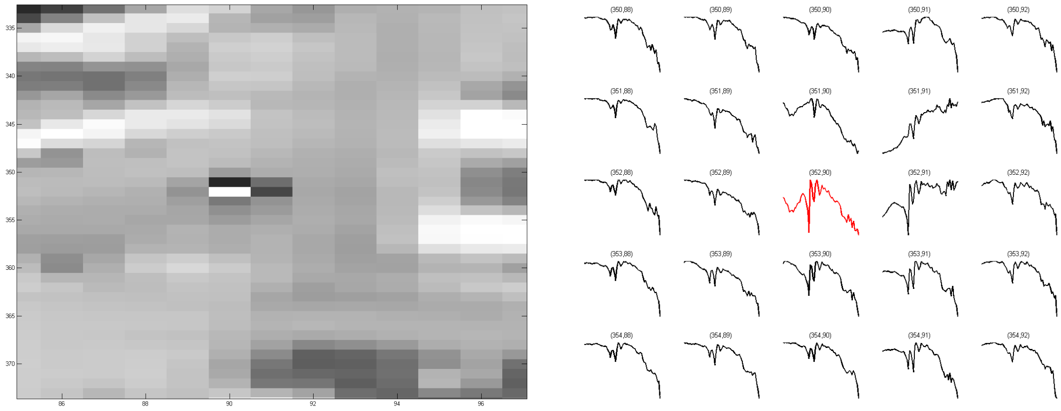


Figure 27. A strong singular point $P(352,90)$. Left: Vicinity of the point **P**. Right: Multipixel spectra of the point $P(352,90)$ and the surrounding points. The weight of the data point **P** is 32

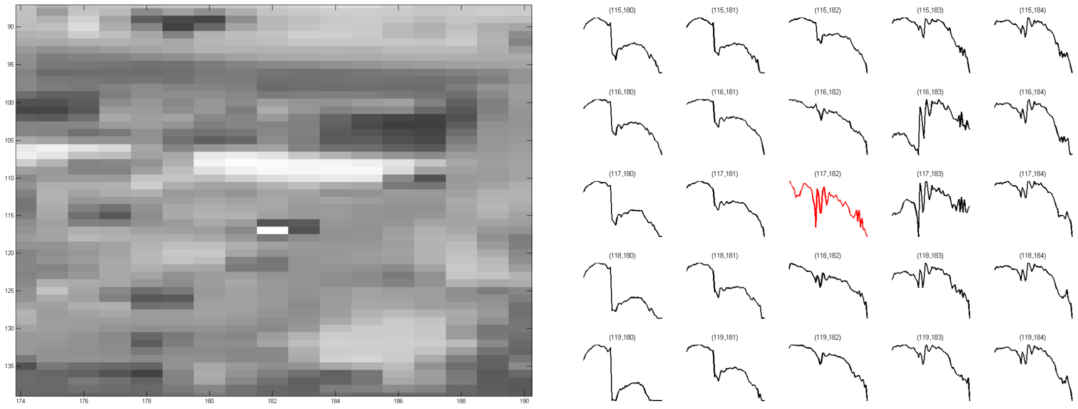


Figure 28. A strong singular point $P(117,182)$. Left: Vicinity of the point **P**. Right: Multipixel spectra at the point $P(117,182)$ and the surrounding points. The weight of the data point **P** is 32

By comparing between Figures 28 and 27 we observe that spectra of singular multipixels located at points $P(117,182)$ and $P(352,90)$ are similar to each other. Supposedly, they correspond to the same material. A different singular multipixel is displayed in Figure 29.

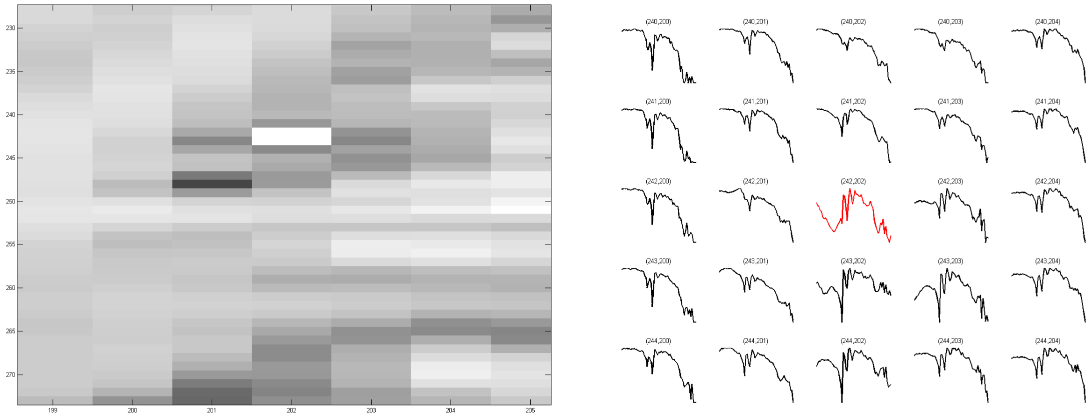


Figure 29. A singular point $P(242,202)$. Left: Vicinity of the point **P**. Right: Multipixel spectra at the point $P(242,202)$ and the surrounding points. The weight of the data point **P** is 32

4.4 Extraction of the *Target's Spectrum from a Suspicious Point*

Let Y be a suspicious point and let T be the given target's spectrum. What portion of the target is contained in Y?

We consider a simplified version of Equation 1 via the definition of a simple mixing model that describes the relation between a target and its background. Assume P is a pixel of mixed spectrum (a spectrum that contains background influence and the target) and T is the given target's spectrum. Consider three spectra: an average background spectrum $B=\sum_{k=1}^{M} a_k B_k$, a mixed pixel spectrum (spectrum of a suspicious point) P and the target's spectrum T. They are related by the following model

$$P = tB + (1-t)T = t\sum_{k=1}^{M} c_k B_k + (1-t)T \tag{6}$$

which is a modified version of Eq. 1, where $a_1 = t$ and $s_1 = T, t \in \mathbb{R}, t \in (0,1)$. $B_k, k = 1,\ldots,M$, was taken from the neighborhood pixel. Therefore, all of them are close to each other and have a similar feature.

We are given the target's spectrum T and the mixed pixel spectrum P. Our goal is to estimate t denoted by $\hat{t}$, which will satisfy Equation 6 provided that B and T have some independent features. Once $\hat{t}$ is found, the estimate of the unknown background spectrum B, denoted by $\hat{B}$, is calculated by $\hat{B} = \left(P - \hat{t}T\right)/\left(1-\hat{t}\right)$. Estimating the parameter t in Equation 6 is called linear unmixing.

In Step 2 from Section 4.1, we calculated the following: V is the d-spectra set, which is uncorrelated with $d(Y)$ pixels from the m-neighborhood of Y and ρ, is the projection operator onto the orthocomplement of the linear span of V. Let $P_2 = \rho(d(Y)), T_2 = \rho(d(T))$, then $P_2 = t^{'}T_2 + N$ where $t^{'}$ is an unknown parameter, N is a random noise that is independent of T_2. The parameter $t^{'} \in [0,1]$ is estimated as the maximum of the independency between the two d spectra T_2 and $P_2 - t^{'}T_2$.

The fact that two vectors X_1 and X_2 are independent is equivalent to $corr(\varphi(X_1),\varphi(X_2)) = 0$ for any analytical function φ (Hyvarinen, Karhunen, & Oja, 2001). An analytical function can be represented by a Taylor expansion of its argument's degrees. Then, the condition $corr(\varphi(X_1),\varphi(X_2)) = 0$ equals to $corr\left((X_1)^n,(X_2)^n\right) = 0$ for any positive integer n where n denotes a power. In our algorithm, we limit our self to $n = 1,2,3,4$. From the independency criterion between the two vectors X_1 and X_2 we can have

$$f = \left|corr(X_1,X_2)\right| + \left|corr\left((X_1)^2,(X_2)^2\right)\right| + \left|corr\left((X_1)^3,(X_2)^3\right)\right| + \left|corr\left((X_1)^4,(X_2)^4\right)\right| \tag{7}$$

which equals to zero in case X_1 and X_2 are independent. If $t^{'}$ is estimated, then $P = t^{'}T + B$ where P is the spectrum of the suspicious point and B is a mix of the background's spectrum from the neighborhood that is affected by noise.

The flow of the UNSP algorithm is given in Figure 30.

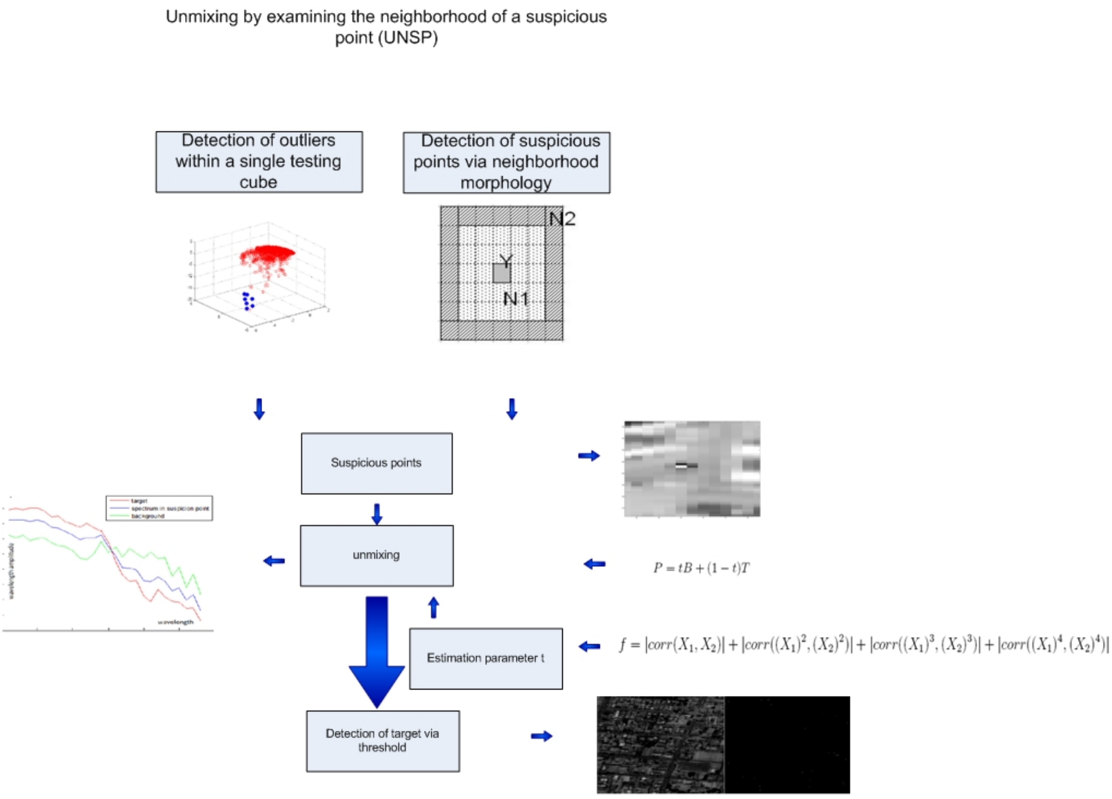

Unmixing by examining the neighborhood of a suspicious point (UNSP)
Detection of outliers within a single testing cube
Detection of suspicious points via neighborhood morphology
N2
Y
N1
Suspicious points
target
spectrum in suspicion point
background
wavelength
unmixing
$P = tB + (1-t)T$
Estimation parameter t
$f = |corr(X_1, X_2)| + |corr((X_1)^2, (X_2)^2)| + |corr((X_1)^3, (X_2)^3)| + |corr((X_1)^4, (X_2)^4)|$
Detection of target via threshold

Figure 30. The flow of the UNSP algorithm

4.5 Experimental Results

In this section, we consider two scenes “field” (Figure 3) and “city” (Figure 2) that contain the subpixel's targets. As a first step, we find all the suspicious points via the application of anomaly detection process (section 4.2). The next step checks the anomaly by the “morphological-filter” which was described in section 4.1. If the pixel is passed via the application of the “morphological-filter” then the target is present in it.

Figures 31 and 32 present the outputs from the application of the “morphological-filter” algorithm to two different hyper-spectral scenarios.

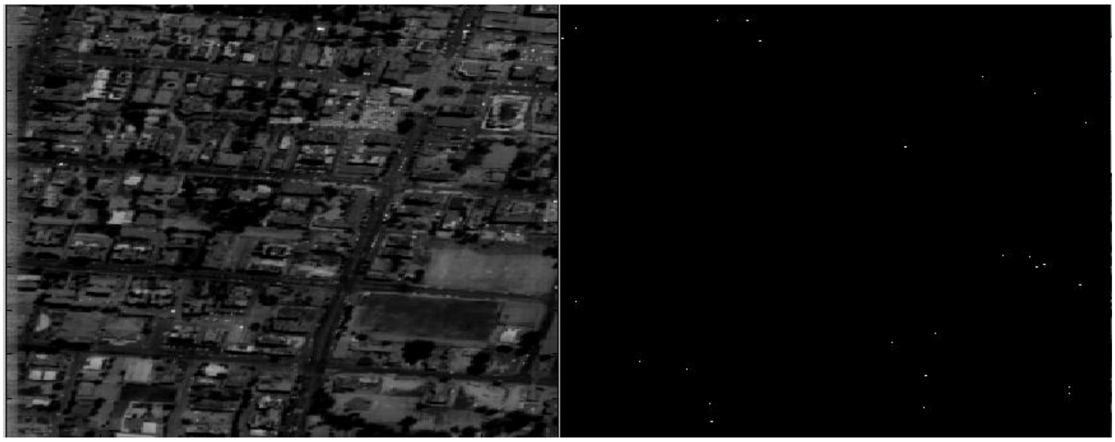


Figure 31. Left: The source image (Figure 2). Right: The white points are the suspicious points in the neighborhood of diameter $m = 10$

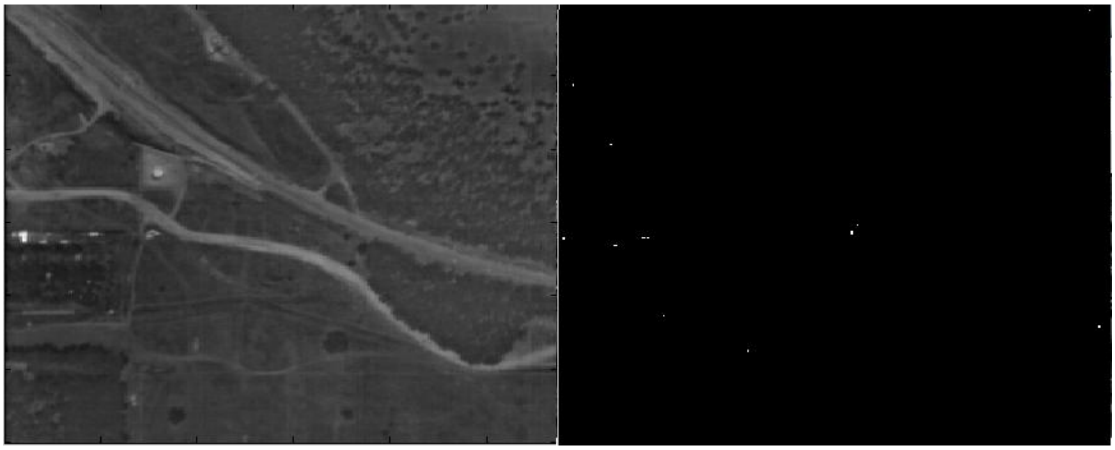


Figure 32. Left: The source image (Figure 2). Right: The white points are the suspicious points in the neighborhood of diameter $m = 10$

In Figures 33 and 34, the x - and y - axes are the wavebands and their values, respectively. The white points in Figures 31 and 32 are the anomalies that contain the known target. The next step performs the unmixing procedure. The unmixing estimates the parameter t from Equation 6 which means that this is a portion from the target that is present in the current suspicious pixel. The estimation of t is done via minimization of the functional f in Equation 7 for a pair of vectors T_2 and $P_2 - t^{'}T_2$ using the notations in section 4.4. Now we present an unmixing example in a pair of singular points detected as anomalies by the application of the “morphological-filter” (section 4.1).

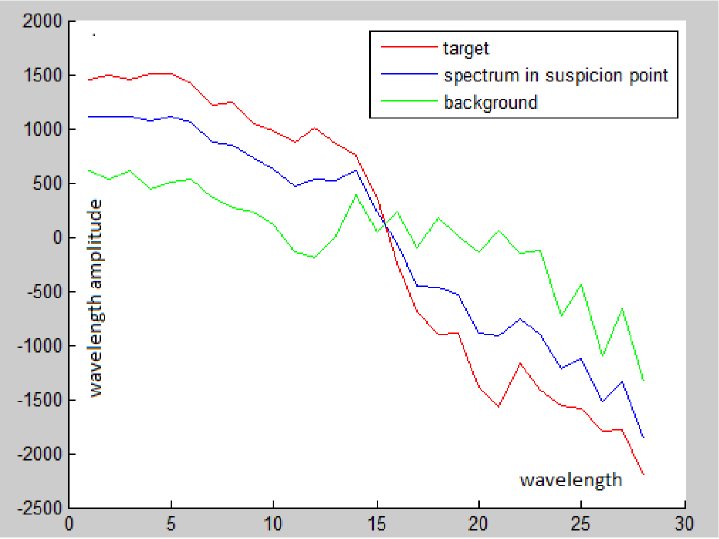


Figure 33. The output from the application of the UNSP unmixing algorithm to the suspicious point in Figure 31. This suspicious point is decomposed into a target and a background spectral portions

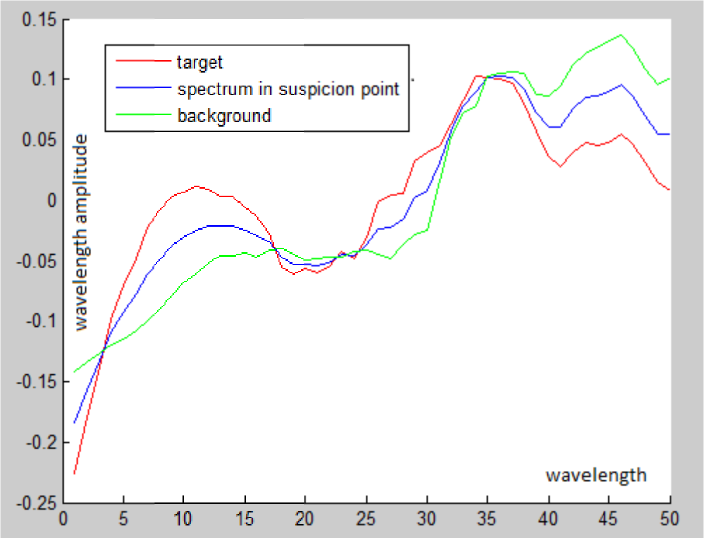


Figure 34. The output from the application of the UNSP unmixing algorithm to the suspicious point in Figure 32. This suspicious point is decomposed into a target and a background spectral portions

4.5.1 Comparison between UNSP and OSP Algorithms

In this section, we compare between the performances of UNSP and OSP (Ahmad & Ul Haq, 2011) algorithms. They are described in Figures 35-41.

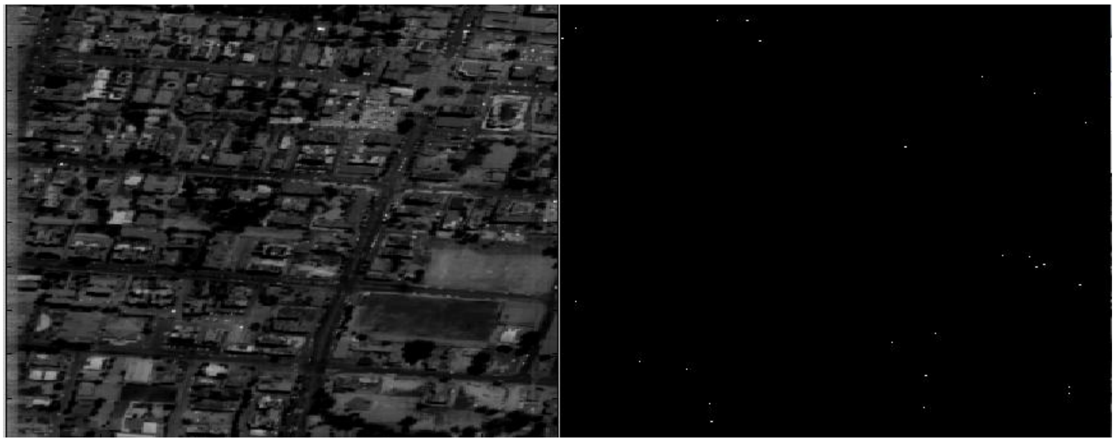


Figure 35. Left: The original “city” image. Right: The white points mark the detected targets by the UNSP algorithm

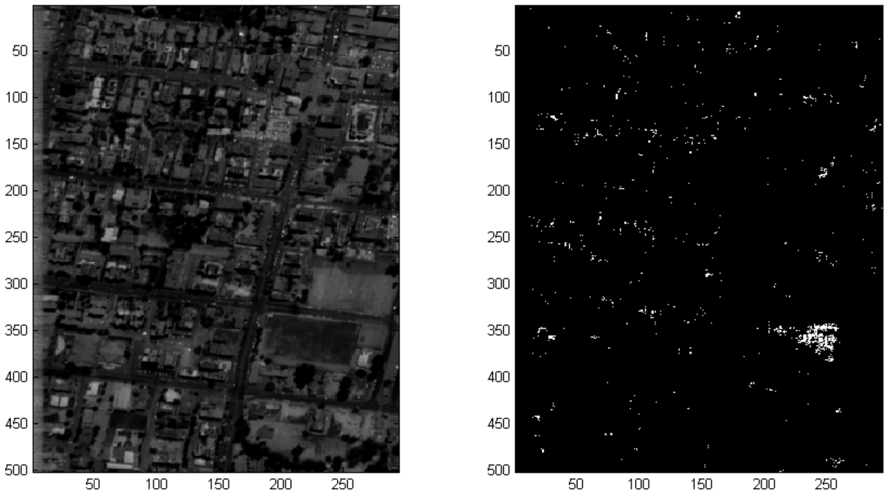


Figure 36. Left: The original “city” image. Right: The white points mark the detected targets by the OSP Ahmad, & Ul Haq (2011) algorithm

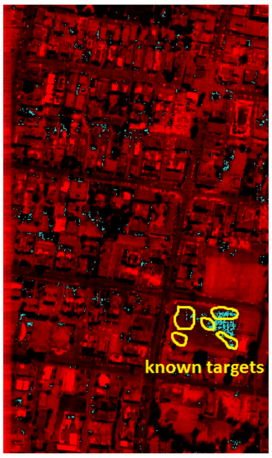


Figure 37. The detected targets by the OSP algorithm are marked over the original “city” image. The yellow circles mark known places that contain the targets. The other points are “false-alarm””. The OSP algorithm produces more “false-alarm” than our method

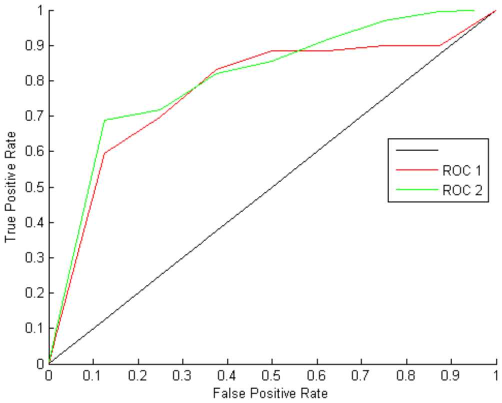


Figure 38. The “ROC-curve” for scene in Figure 37. The red line corresponds to OSP Ahmad & Ul Haq (2011) method. The green line corresponds to the WDR method

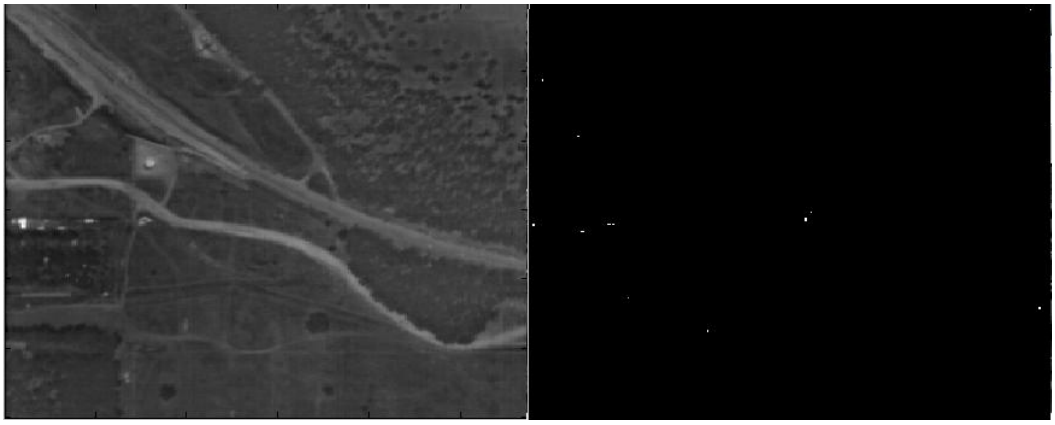


Figure 39. Left: The original "field" image. Right: The white points mark the detected targets by the UNSP algorithm

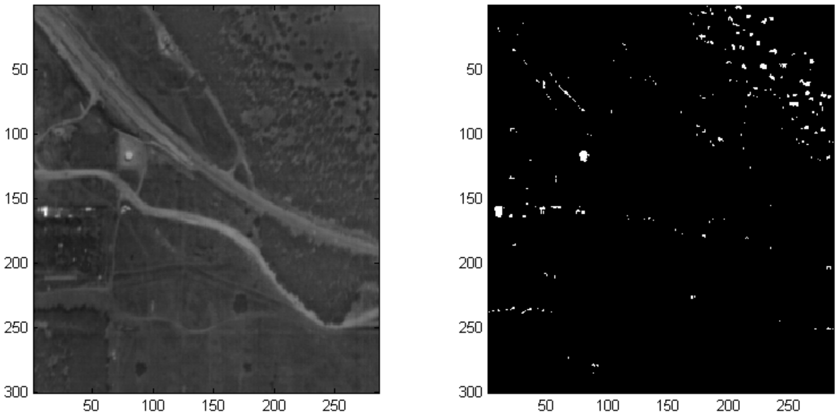


Figure 40. Left: The original "field" image. Right: The white points mark the detected targets found by the OSP Ahmad & Ul Haq (2011) algorithm

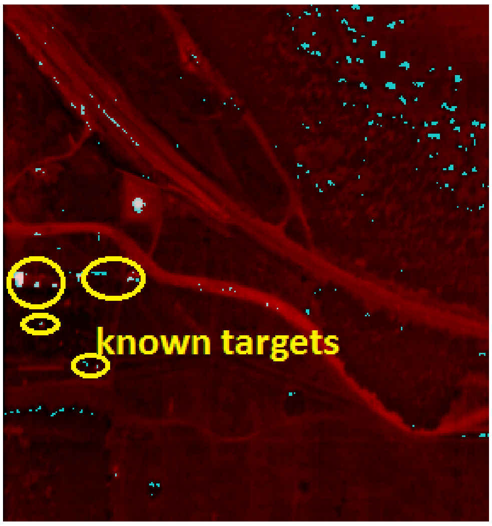


Figure 41. The detected targets by the OSP algorithm are marked over the original "field" image

The yellow circles mark known places containing the targets. Other points are "false-alarm". The OSP Ahmad & Ul Haq (2011) algorithm produces more "false-alarm" than our method.

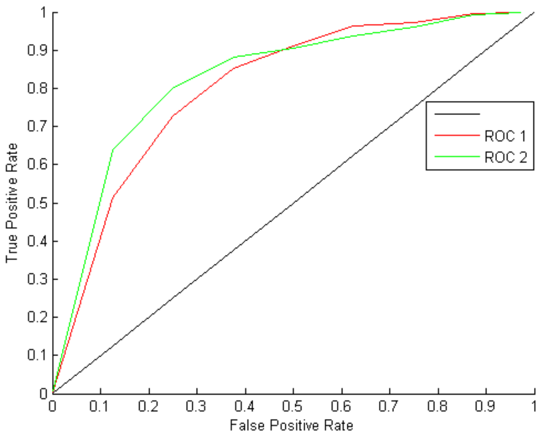


Figure 42. The "ROC-curve" for scene in Figure 41. The red line corresponds to OSP Ahmad, & Ul Haq (2011) method. The green line corresponds to the WDR method

5. Conclusions

We presented two algorithms for linear unmixing. The WDR algorithm detects well targets that occupy at least one pixel but fails to detect sub-pixel targets. The UNSP algorithm detects well sub-pixels targets but it is computational expensive due to the need to search for the spectral decomposition in each pixel's neighborhood by sliding the "morphology-filer". In the future, we plan to add to these algorithms a classification method with machine learning methodologies.

References

Ahmad, M., & Ul Haq, I. (2011). Linear Unmixing and Target Detection of Hyper-spectral Imagery. *2011 International Conference on Modeling, Simulation and Control IPCSIT, IACSIT Press, 10*, 179-183.

Ahmad, M., Ul Haq, I., & Mushtaq, Q. (2011). AIK Method for Band Clustering Using Statistics of Correlation and Dispersion Matrix. *2011 International Conference on Information Communication and Management, IACSIT Press, 10*, 114-118.

Arngren, M., Schmidt, M. N., & Larsen, J. (2009). Bayesian nonnegative matrix factorization with volume prior for unmixing of hyper-spectral images. *IEEE Workshop on Machine Learning for Signal Processing (MLSP), Grenoble*, 1-6. http://dx.doi.org/10.1109/MLSP.2009.5306262

Averbuch, A. Z., & Zheludev, M. V. (2012). Two Linear Unmixing Algorithms to Recognize Targets using Supervised Classification and Orthogonal Rotation in Airborne Hyper-spectral Images. *Remote Sensing, 4*, 532-560. http://dx.doi.org/10.3390/rs4020532

Bateson, C., Asner, G., & Wessman, C. (2000). Endmember bundles: A new approach to incorporating endmember variability into spectral mixture analysis. *IEEE Trans. Geoscience Remote Sensing, 38*, 1083-1094. http://dx.doi.org/10.1109/36.841987

Bayliss, J. D., Gualtieri, J. A., & Cromp, R. F. (1997). Analysing hyper-spectral data with independent component analysis. *Proceeding of the SPIE conference, 3240*, 133-143.

Bioucas-Dias, J., & Plaza, A. (2010). Hyper-spectral unmixing: Geometrical, statistical and sparse regression-based approaches. SPIE Remote Sensing Europe, Image and Signal Processing for Remote Sensing Conference, SPIE.

Bioucas-Dias, J., & Plaza, A. (2011). An overview on hyper-spectral unmixing: geometrical, statistical, and sparse regression based approaches. *Proceeding IEEE Int. Conf. Geosci. and Remote Sensing (IGARSS), IEEE International*, 1135-1138.

Boardman, J. (1993). Automating spectral unmixing of AVIRIS data using convex geometry concepts. *Summaries of the Fourth Annunl Airborne Gcoscicnce Workshop, TIMS Workshop, Jet Propulsion Laboratory, Pasadena, CA, 2*, 11-14.

Burges, J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery, 2*(2), 121-167. http://dx.doi.org/10.1023/A:1009715923555

Chang, C. I. (2003). *Hyper-spectral Imaging: Techniques for spectral detection and classification*. Kluwer Academic New York.

Chang, C. I., Zhao, X., Althouse, M. L. G., & Pan, J. J. (1998). Least squares subspace projection approach to mixed pixel classification for hyper-spectral images. *IEEE Trans. Geoscience Remote Sensing, 36*(3), 898-912. http://dx.doi.org/10.1109/36.673681

Coifman, R. R., &. Lafon, S. (2006). Diffusion Maps. *Applied and Computational Harmonic Analysis, 21*(1), 5-30. http://dx.doi.org/10.1016/j.acha.2006.04.006

Common, P. (1994). Independent component analysis: A new concept. *Signal Processing, 36*, 287-314. http://dx.doi.org/10.1016/0165-1684(94)90029-9

Craig, M. D. (1994). Minimum-volume transforms for remotely sensed data. *IEEE Trans. Geoscience Remote Sensing*, 99-109.

Cristianini, N., & Shawe-Taylor, J. (2000). *Support Vector Machines and other kernel-based learning methods*. Cambridge University Press.

Dobigeon, N., Moussaoui, S., Coulon, M., Tourneret, J. Y., & Hero, A. O. (2009). Joint Bayesian endmember extraction and linear unmixing for hyper-spectral imagery. *IEEE Trans. Signal Processing, 57*(11), 4355-4368. http://dx.doi.org/10.1109/TSP.2009.2025797

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification*. John Wiley & Sons Inc.

Harsanyi, J. C., & Chang, C. I. (1994). Hyper-spectral image classification and dimensionality reduction: an orthogonal subspace projection approach. *IEEE Trans. Geosciencce Remote Sensing, 32*(4), 779-785. http://dx.doi.org/10.1109/36.298007

Hyvarinen, A., Karhunen, J., & Oja, E. (2001). *Independent Component Analysis*. John Wiley & Sons Inc1.

Ifarraguerri, A., & Chang, C. I. (1999). Multispectral and hyper-spectral image analysis with convex cones. IEEE Trans. *Geoscience Remote Sensing, 37*, 756-770. http://dx.doi.org/10.1109/36.752192

Manolakis, D., & Shaw, G. (2002). Detection algorithms for hyper-spectral imaging applications. *IEEE Signal Processing Magazine, 19*(1), 29-43. http://dx.doi.org/10.1109/79.974724

Manolakis, D., Marden, D., & Shaw, G. (2001). Hyper-spectral image processing for automatic target detection applications. *Lincoln Lab Journal, 14*(1), 79-114.

Moussaoui, S., Carteretb, C., Briea, D., & Mohammad-Djafaric, A. (2006). Bayesian analysis of spectral mixture data using markov chain monte carlo methods. *Chemometrics and Intelligent Laboratory Systems, 81*(2), 137-148. http://dx.doi.org/10.1016/j.chemolab.2005.11.004

Nascimento, M. P., & Bioucas-Dias, M. (2005a). Does independent component analysis play a role in unmixing hyper-spectral data? *IEEE Trans. Geoscience Remote Sensing, 43*(1), 175-187. http://dx.doi.org/10.1109/TGRS.2004.839806

Nascimento, M. P., & Bioucas-Dias, M. (2005b). Vertex component analysis: A fast algorithm to unmix hyper-spectral data. *IEEE Trans. Geoscience Remote Sensing, 43*(4), 898-910. http://dx.doi.org/10.1109/TGRS.2005.844293

Ren, H., & Chang, C. I. (2003). Automatic Spectral Target Recognition in Hyper-spectral imagery, *IEEE Trans, on Aerospace and electronic Systems, 39*(4), 1232-1249. http://dx.doi.org/10.1109/TAES.2003.1261124

Settle, J. J. (1996). On the relationship between spectral unmixing and subspace projection. *IEEE Trans. Geoscience Remote Sensing, 34*, 1045-1046. http://dx.doi.org/10.1109/36.508422

SPECIM camera. (2006). Retrieved from http://www.specim.fi/

Winter, M. E. (1999). N-findr: An algorithm for fast autonomous spectral end-member determination in hyper-spectral data. *Proceeding SPIE Conf. Imaging Spectrometry V, SPIE, 266-275.*

Appendix: Diffusion Maps

Diffusion Maps (DM) Coifman, R. R., &. Lafon, S. (2006) analyzes a dataset M by exploring the geometry of the manifold $\boldsymbol{M}$ from which it is sampled. It is based on defining an isotropic kernel $K \in \mathbb{R}^{n \times n}$ whose elements are defined by $k(x,y)=e^{-\frac{\|x-y\|}{\varepsilon}}, x,y\in M$, ε is a meta-parameter of the algorithm. This kernel represents the affinities between data points in the manifold. The kernel can be viewed as a construction of a weighted graph over the dataset M. The data points in M are the vertices and the weights of the edges are defined by the kernel K. The degree of each data point (i.e., vertex) $x\in M$ in this graph is $q(x)=\sum_{y\in M}k(x,y)$. Normalization of the kernel by this degree produces an $n\times n$ row stochastic transition matrix P whose elements are $p(x,y)=k(x,y)/q(x), x,y\in M$, which defines a Markov process (i.e., a diffusion process) over the data points in M. A symmetric conjugate $\overline{P}$ of the transition operator P defines the diffusion affinities between data points by

$$\overline{p}(x,y)=\frac{k(x,y)}{\sqrt{q(x)q(y)}}=\sqrt{q(x)}p(x,y)\frac{1}{\sqrt{q(y)}},\quad x,y\in M\ .$$

DM embeds the manifold into an Euclidean space whose dimensionality is usually significantly lower than the original dimensionality. This embedding is a result from the spectral analysis of the diffusion affinity kernel $\overline{P}$. The eigenvalues $1\geq\sigma_0\geq\sigma_1\geq\ldots$ of $\overline{P}$ and their corresponding eigenvectors $\overline{\phi}_0,\overline{\phi}_1,\ldots$ are used to construct the desired map, which embeds each data point $x\in M$ into the data point $\overline{\Phi}(x)=\left(\sigma_i\overline{\phi}_i(x)\right)_{i=0}^{\delta}$ for a sufficiently small δ, which is the dimension of the embedded space. δ depends on the decay of the spectrum $\overline{P}$.