Unmixing and target recognition in hyper-spectral images

Amir Z. Averbuch Valery Zheludev Michael V. Zheludev
School of Computer Science
Tel Aviv University
Tel Aviv 69978, Israel

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 in 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.

1 Introduction

1.1 Data representation and extraction of spectral information

We assume that an 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. Figure 1.1, taken from (Winter, 1999), shows an example of an hyper-spectral data cube.

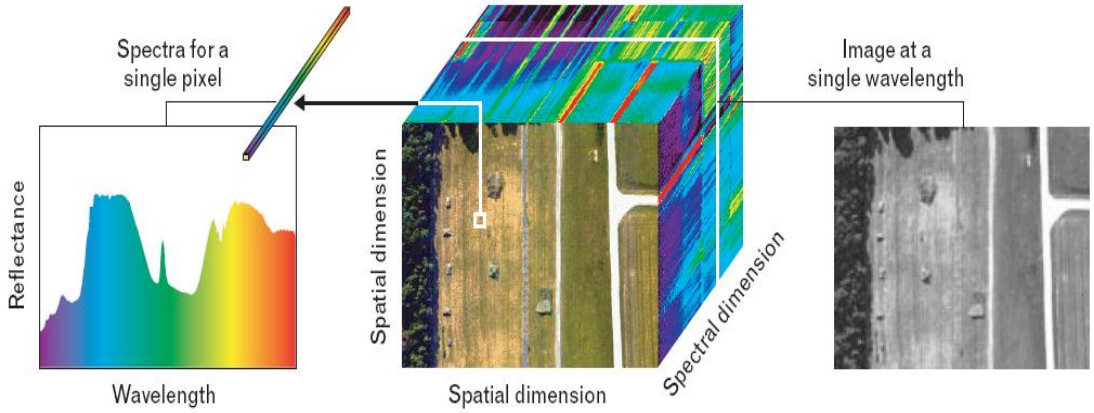

Figure 1: Center: Data cube structure of an hyperspectral image. Left: A single pixel with all its wavelengths also called multipixel. Right: The data cube is visualized as a stack of images in a single spectral channel (wavelength).

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 $x = (x_1, x_2, \ldots, x_K)$ where its Cartesian coordinates x_i are either radiance or reflectance values at each spectral band. Since $x_i \geq 0,\ i = 1, \ldots, K$, 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 orientation, 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, 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.

The most widely used spectral mixing model is the linear mixing model, which 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 ((Winter, 1999)):

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

where $s_1, s_2, \ldots, s_M$ are the M endmember spectra that assumed to be linearly independent, $a_1, a_2, \ldots, a_M$, 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;

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), which are mentioned in section 2, work well.

The experiments in this papier were performed on three real hyper-spatial datasets titled: "desert", "city" and "field" which were acquired by the Specim camera (http://www.specim.fi/,) on a plane. Their properties with a display of one waveband per dataset is given in Figs. 2-4

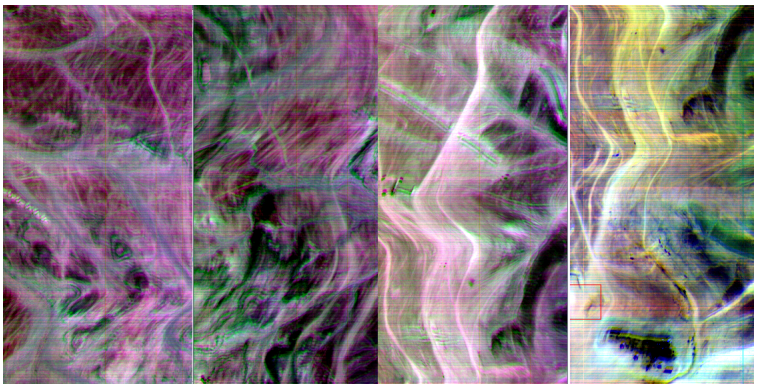

Figure 2: The dataset "desert" is an hyper-spectral image of a desert place taken by airplane flying 10,000 feet above sea level. The resolution is 1.3 meter/pixel, 286 $\times$ 2640 pixels per waveband with 168 wavebands

Figure 3: The dataset "city" is an hyper-spectral image of a city taken by airplane flying 10,000 feet above sea level. The resolution is 1.5 meter/pixel, 294 $\times$ 501 pixels per waveband with 28 wavebands

Figure 4: The dataset "field" is an hyper-spectral image of a field taken by airplane flying 9,500 feet above sea level. The resolution is 1.2 meter/pixel, 286 $\times$ 300 pixels per waveband with 50 wavebands

The paper has the following structure: Section 2 describes some of the main known unmixing methods. 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 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 Diffusion Maps (DM). The second part unmixes the suspicious point. It is based on a PCA of the linear span of the neighboring background spectra.

2 Related work

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 .

If the mixture of hyper-spectral data is linear, then ICA is a possible tool to perform unmixing. For example, the application of ICA to hyper-spectral data was proposed in (Bayliss, Gualtieri, & Cromp, 1997), where the endmember signatures are treated as sources and the mixing matrix is composed by the abundance fractions - see (Botchko, Berina, Korotkaya, Parkkinen, & Jaaskelainen, 2003; Hyvarinen et al., 2001; Chiang, Chang, & Ginsberg, 2000; Keshava, Kerekes, Manolakis, & Shaw, 2000; Parra, Mueller, Spence, Ziehe, & Sajda, 2000; Tu, 2000), where the sources are the abundance fractions of each endmember. The first approach has two difficulties: 1. The number of samples is limited to the number of channels. 2. The process of pixel selection, which plays the role of mixed sources, is not straightforward. The second approach also faces difficulties since the sum of the abundance fractions is constant. This implies statistical dependency among abundances (i.e. among sources). This dependency violates a key assumption of ICA of having statistical source independency. The applicability of ICA to hyper-spectral images is thus compromised. In addition, hyper-spectral data are immersed in noise, which degrades the ICA performance. Independent factor analysis (IFA) (Attias, 1999; Moulines, Cardoso, & Gassiat, 1997) was introduced as a method for recovering independent hidden sources from their observed noisy mixtures. IFA implements two steps: 1. Source densities and noise covariance are estimated from the observed data by ML; 2. Sources are reconstructed by an optimal nonlinear estimator. Although IFA is a well suited technique to unmix independent sources under noisy observations, the dependency among abundance fractions in hyper-spectral imagery compromises, as in the ICA case, the IFA performance.

The impact of source dependency on unmixing hyper-spectral data with ICA and IFA algorithms is investigated in (Nascimento & Bioucas-Dias, 2005a). It shows that these algorithms do not correctly unmix hyper-spectral data such that the unmixing matrix, which minimizes the mutual information, can be far from being true.

Geometric approach: Assuming a linear mixing scenario where each observed spectral vector is given by $\mathbf{r} = \mathbf{x}+\mathbf{n}=\mathbf{M}\gamma\mathbf{a}+\mathbf{n}$, $\gamma\mathbf{a} = \mathbf{s}$, where $\mathbf{r}$ is an L-vector (L is the number of bands), $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \dots \mathbf{m}_p]$ is the mixing matrix ($\mathbf{m}_i$ denotes the ith endmember signature and p is the number of endmembers present in the sensed area), $\mathbf{s} \triangleq \gamma\mathbf{a}$ (γ is a scale factor that models illumination variability due to a surface topography), $\mathbf{a} = [a_1, a_2, \dots a_p]^T$ is the abundance vector that contains the fractions of each endmember ($(\cdot)^T$ denotes a transposed vector) and $\mathbf{n}$ 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 $\{\mathbf{a} \in \mathbb{R}^p : \quad \sum_{k=1}^{p} a_k = 1,\ a_k > 0 \text{ for all } k\}$ is a simplex, then the set $S_x \triangleq \left\{\mathbf{x} \in \mathbb{R}^L : \quad \mathbf{x} = \mathbf{M}\mathbf{a}, \quad \sum_{k=1}^{p} a_k = 1,\ a_k > 0 \text{ for all } k\right\}$ is also a simplex whose vertices correspond to endmembers.

Several approaches (Ifarraguerri & Chang, 1999),(Boardman, 1993) and (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 minimum 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, 2005b). 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 computational expansive. In addition, it relies on "pure" spectra which are not available all the time.

2.1 Linear classification for threshold optimization

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

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

where $(w, b) \in \mathbb{R}^n \times \mathbb{R}$ are the parameters that control the function. The decision rule is given by $sgn(f(x))$. w as 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)*

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

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

The Rosenblatt's Perseptron algorithm ((Cristianini & Shawe-Taylor, 2000; Burges,), 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 $\lambda_i = y_i(\langle w, x_i \rangle + b)$. $\lambda_i > 0$ that implies the correct classification for (x_i, y_i).

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

Definition 2.2. *((Cristianini & Shawe-Taylor, 2000)) The dataset S from Eq. 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 $\lambda_i = y_i(\langle w, x_i \rangle + b) > 0$, $i = 1, \ldots, l$. 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 $P = \{p_1, ..., p_k\} \subseteq \mathbb{R}^n$ if the training set $S = ((x, 1), (p_1, -1), ..., (p_k, -1))$ 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 $S = \{x_1, .., x_n\}$ 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, 2000)). 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 et al., 2000)),

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

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 et al., 2000)), l is the number of classes, S_i are the classes and m_i are the center of each class. S_B is called "between-cluster scatter matrix"((Duda et al., 2000)), 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 $S = \{x_1, ..., x_n\} \subseteq \mathbb{R}^n$ via K-means and Fisher's discriminant analyzes (Cristianini & Shawe-Taylor, 2000; Burges,). (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 $m_1 - m_{-1} > max(diam(P_1), diam(P_{-1}))$, where the notation in Eq. 4 is used. Another criterion is:

Definition 2.5. *((Duda et al., 2000)) A dataset is separable if from Eq. 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 $J(e_1) \geq (e_2)$, 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

When a target occupies one or more pixels, the target's spectrum can be recognized by comparing it with patterns from a given database of spectra. 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 Fig. 5 (left). Otherwise, H_1 holds and P_1 has an oscillatory behavior - see Fig. 5 (right). In addition, P_1 has a subset of coordinates whose first and second derivatives have an oscillatory behavior - see Fig. 5 (right).

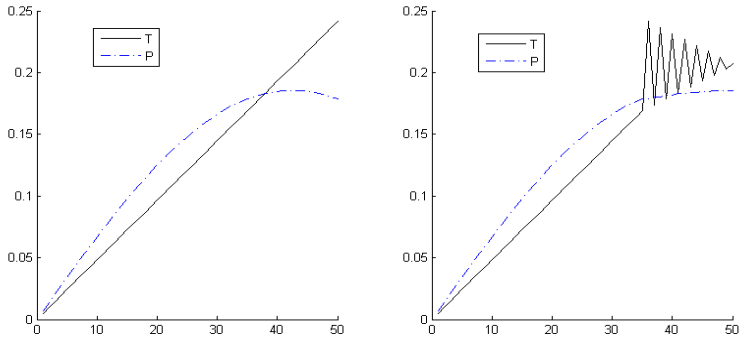

Figure 5: The x- and the y-axes are the wavebands and their values, respectively. The spectra are represented after the application of the permutation to the coordinates, which permutes T into a monotonic deceasing order. Left: Weak dependency between T and P. Right: No weak dependency between T and 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 $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$. 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 definition 2.5.

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.

3.2 Experimental results

Figures 6-8 display the results after the application of the algorithm in section 3.1 to the "desert" image (Fig. 2).

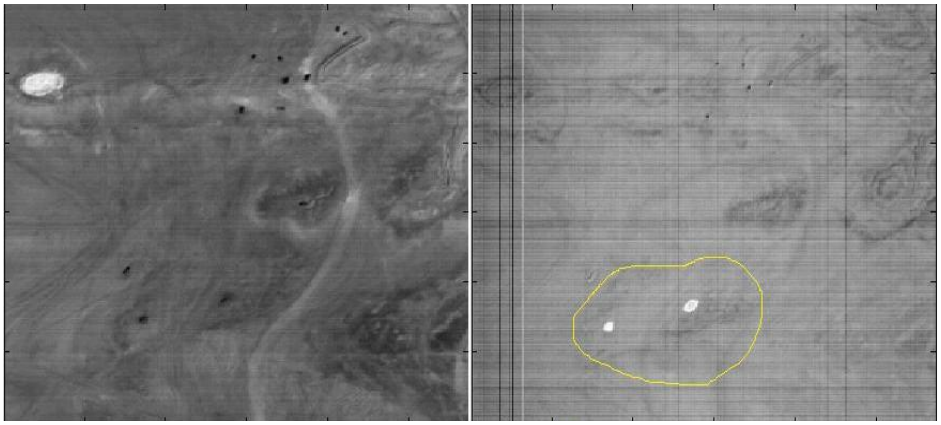

Figure 6: Left: One wavelength part from the original "desert" image (Fig. 2). Right: The white points mark the detected targets. The intensity of each pixel in the right side corresponds to the value $-\Omega(X)$ where X is the spectrum in the current pixel.

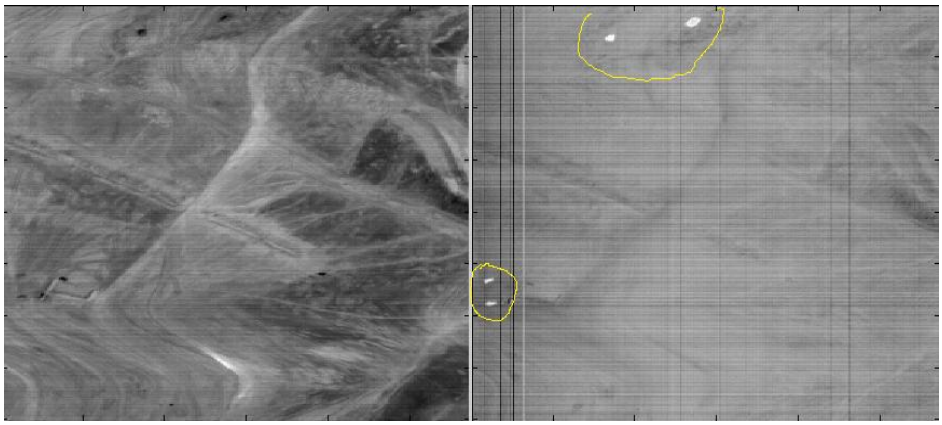

Figure 7: Left: One wavelength part from the original "desert" image (Fig. 2). Right: The white points mark the detected targets. The intensity of each pixel in the right side corresponds to the value $-\Omega(X)$ where X is spectrum in the current pixel.

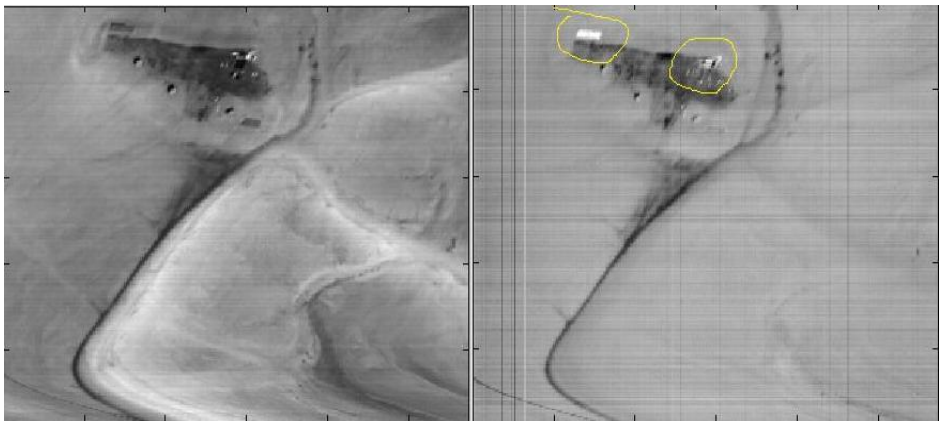

Figure 8: Left: One wavelength part from the original "desert" image (Fig. 2). Right: The white points mark the detected targets. The intensity of each pixel in the right side corresponds to the value $-\Omega(X)$ where X is the spectrum in the current pixel.

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

The point P_1 In Fig. 8 is the pattern of the known target's material. Its spectrum is displayed in Fig. 9 as a plot of the "target". Other spectra plots, which were detected by the WDR algorithm in the scenes of Figs. 6-8, are classified as "spectra of suspicious points".

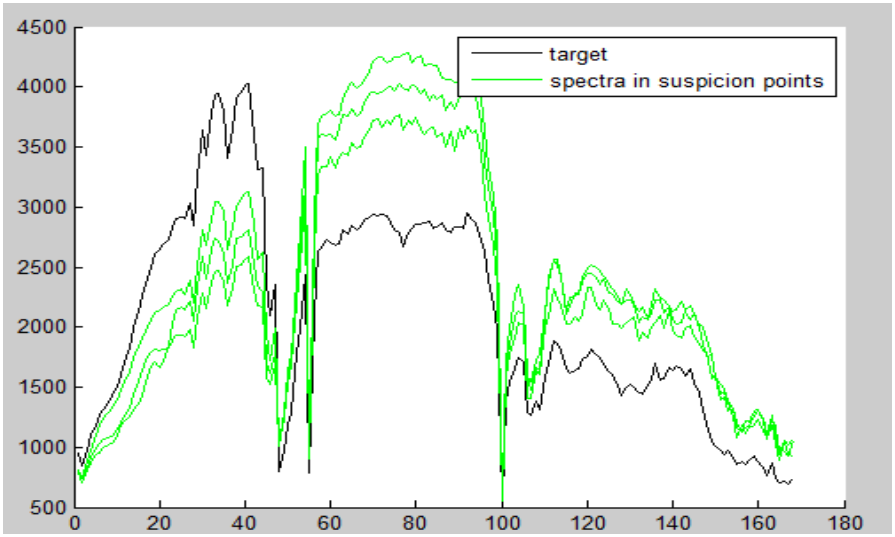

Figure 9: Comparison between the target's spectrum and the suspicious points' spectra in Figs. 6-8. The x- and the y-axes are the wavebands and their values, respectively.

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 size that we use for 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 Fig. 10.

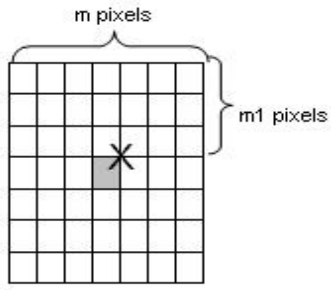

Figure 10: The m-neighborhood of the pixel X denoted by $\Omega_m(X)$

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 Fig. 11.

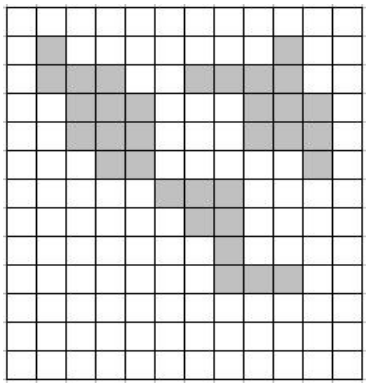

Figure 11: The morphological structure of the neighborhood is represented by three connected components

Consider the spectra from different materials which are present in a hyper-spectral image. In real situations, usually there is high correlation between these spectra. For example, Fig. 12 displays spectra from three different materials.

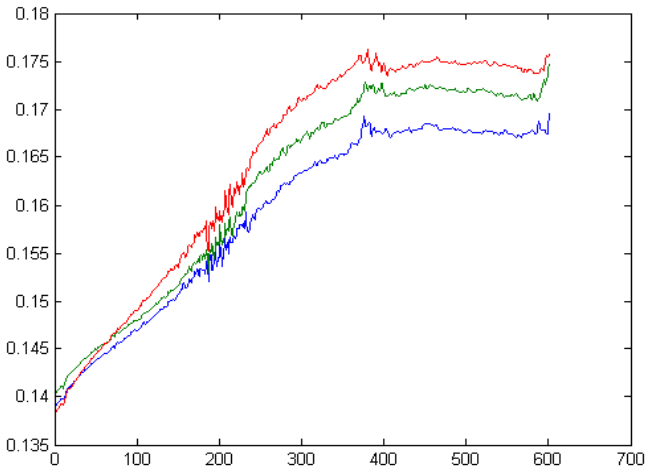

Figure 12: Spectra of three different materials

To reduce the correlation between spectra, we use the first derivatives of the spectra. We denote the first derivative of a 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 Fig. 13. For example, the central pixel is $Y = p_{m_1+1 m_1+1}$ where m_1 is the radius of the neighborhood.

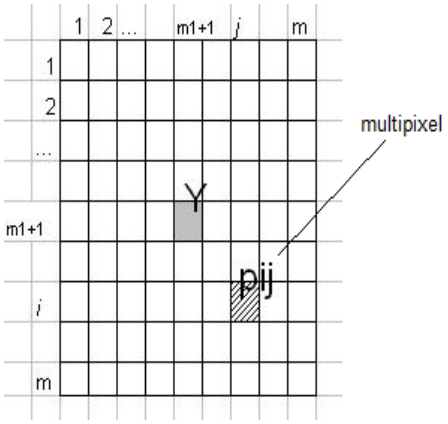

Figure 13: The indices of a pixel

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

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 not a suspicious point also. 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 Fig. 14.

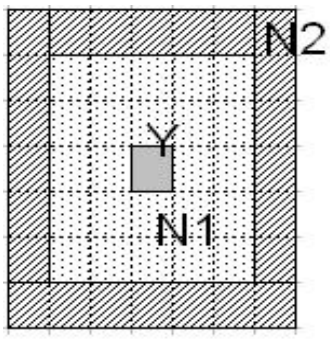

Figure 14: N_1 is the internal square and N_2 is the external square

Assume that Δ is the set of all pixels p_{ij}, which are bounded by the external square with correlation coefficients $corr(d(p_{ij}), d(Y))$, which associated with the current neighborhood, that are less than the Fisher's threshold b. Each pixel in Δ is treated as a vector where its entries are spread all over the wavelengths. This pixel is also called multipixel. 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 $V = \{v_1, \ldots, v_s\}$.

In order to derive the d-spectrum of some material in a 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 is not mixed 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 $E(i,j) = v_i \cdot v_j$, $(i,j) \in \Delta$. Assume that T_e is Fisher's threshold, which separates between the big and small absolute values of 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 $P' = \tau M' + L(v_1, ..., v_s) + N$, 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 $L(v_1, ..., v_s)$ is a linear combination of the vectors $v_1, ..., v_s$.

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 $\rho(L(v_1, ..., v_s))$ 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 a 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 Principal Component Analysis (PCA) and a nearest-neighbor scheme. Assume, we are given a data cube **D** of size $X \times Y \times Z$, where X and Y are 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

Assume that a moving testing cube **d**, pointed by the arrows in Fig. 15, is moving by ironing each time a different fragment in the data cube **D**, described in Fig. 3. Section 4.3 describes in details how the testing cube **d** moves.

Figure 15: An urban scene of size 294 × 501 (from the "city" in Fig. 3) with different locations of the moving testing cube **d**. The arrows point to these locations

The testing cube **d** contains $N = v\,h$ multipixels each of which comprises Z wavebands. Typically, v and h are in the range $30 \div 50$, Z is in the range $30 \div 100$, $Y \approx 290$. Thus, each of the N data points is a vector $\mathbf{m}_i,\ i = 1, ..., N$, of length Z. We arrange these data points

into a matrix $\mathbf{M}$ of size $N \times Z$.

The next step applies the PCA algorithm to the matrix $\mathbf{M}$. It reduces the dimensionality of the data vectors by projecting them into the main eigenvectors of the covariance matrix of the data $\mathbf{M}$. This projection reveals the geometrical structure of the data and facilitates a search for singular (abnormal) data points. The data matrix $\mathbf{M}$ of size $N \times Z$ is mapped onto the principal components (PC) of the matrix $\mathbf{P}$ of size $N \times R$, $R \ll Z$. Typically, R in the range $3 \div 5$, which is determined by the magnitudes of the corresponding eigenvalues. Figure 16 displays three principal components of the data from four positions of the moving testing cube. These are the projections onto three major eigenvectors of the covariance matrices.

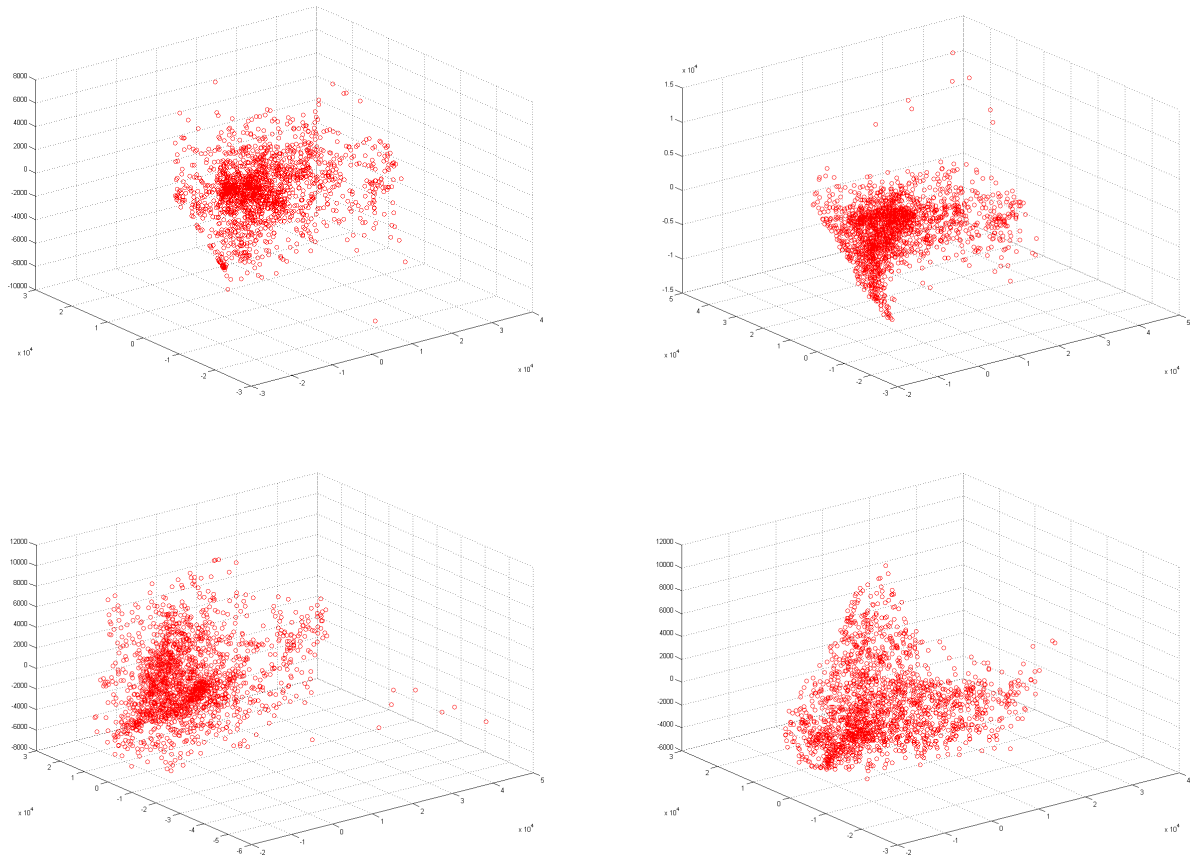

Figure 16: Three principal components of the data from different positions of the testing cube on the image in Fig. 3

We observe that the overwhelming majority of the projected 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 major 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 section 4.2.2.

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 $r_i,\ i = 1, ..., N$, of the PC matrix $\mathbf{P}$, 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,

$$s_{i,1} = d_{i,j_{i,1}} \leq s_{i,2} = d_{i,j_{i,2}} \leq ... \leq s_{i,N-1} = d_{i,j_{i,N-1}}.$$

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

3. For each row $r_i,\ i = 1, ..., N$, determine its K nearest neighbors. For this, take the first K columns $\{j_{i,1}, j_{i,2}, ..., j_{i,K}\},\ i = 1, ..., N$, of the index matrix $\mathbf{J}$. The corresponding distances are presented in the K first columns $\{s_{i,1}, s_{i,2}, ..., s_{i,K}\},\ i = 1, ..., N$, of the matrix $\mathbf{S}$. Thus, we have the nearest neighbor index $\mathbf{J}_K$ and the distances matrices $\mathbf{S}_K$ 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 $\mathbf{m}_{i_2}$) in the data cube $\mathbf{D}$. 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 $i = \tilde{i}_1 \neq i_2$. 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 $\mathbf{m}_{i_K}$) in the data cube $\mathbf{D}$. Store the point $P_K(x_K, y_K)$.

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

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

   (b) Grouped outliers arrive when all the maxima $\mu_k,\ k = 2, ..., K-1$, 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 $\mu_k,\ k = L+1, ..., K-1,\ L > 1$, 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 $\{P_1, ..., P_L, P_K\}$ consist of the point P_K and of the L points closest to P_K. The indices of the points $\{P_1, ..., P_L\}$ are $i_1 = j_{i_K,1}, ..., i_1 = j_{i_K,L}$, 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 moving testing cube $\mathbf{d}$. Figure 17 illustrates the grouped detected outliers in the 3-dimensional space of principal components of the data from four positions of the moving testing cube.

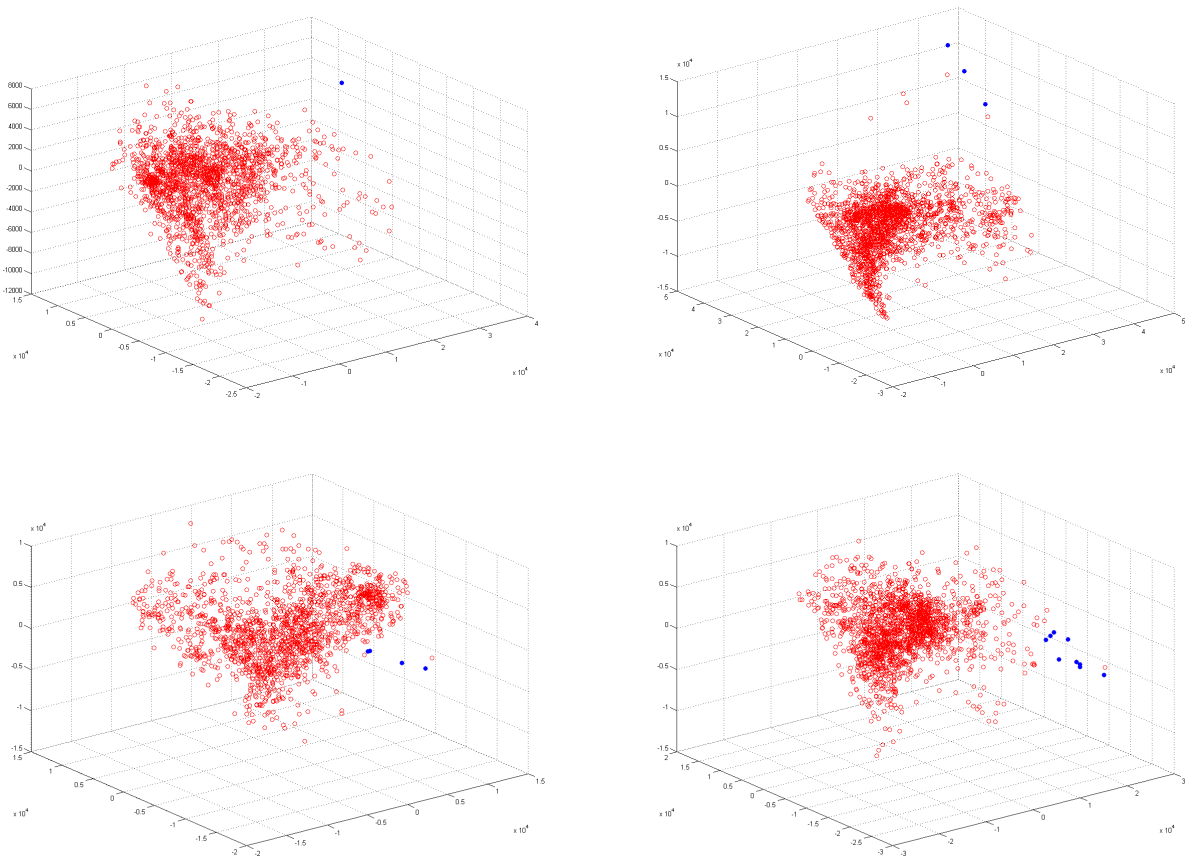

Figure 17: Detection of the grouped outliers in the principal components of the data from different positions of the moving testing cube.

4.3 Detection of singular points within the whole data cube

In section 4.2.2, we described how to find a group of data points (multipixels) within one moving testing cube, whose geometry differs from the geometry of the majority of the data points. Let $\Lambda^1 = \left\{P_1^1, ..., P_{L^1}^1\right\}$ be the list of such data points in the moving testing cube $\mathbf{d}^1$ of size $v \times h \times Z$ located in the upper left corner of the data cube $\mathbf{D}$ as illustrated by the arrow in Fig. 15. The next testing cube $\mathbf{d}^2$ is obtained by a right shift by $\eta \approx h/4$ of $\mathbf{d}^1$. Let $\Lambda^2 = \left\{P_1^2, ..., P_{L^2}^2\right\}$ be the list of outliers in the cube $\mathbf{d}^2$. Append the list Λ^2 to Λ^1. Because of the vast overlap between the cubes $\mathbf{d}^2$ and $\mathbf{d}^1$, 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 moving testing cube $\mathbf{d}^3$ whose outliers list Λ^3 is appended to the combined list $\Lambda^1 \bigcup \Lambda^2$. Again, the common gain weights. We proceed with the right shifts

till the right edge of the data cube $\mathbf{D}$. Then, the testing cube slides down by $\nu \approx v/4$ and starts η-shifts to the left and so on. As a result, we get a combined list $\Lambda = \Lambda^1 \bigcup \Lambda^2 ... \Lambda^R$ of outliers, where R is the number of jumps of the testing cube $\mathbf{d}$ within the data cube $\mathbf{D}$. Figure 15 illustrates a route of the cube $\mathbf{d}$ on the data cube $\mathbf{D}$.

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 measure of singularity 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 overlapping testing cubes. Thus, it can be regarded as a strong singular point for the data cube $\mathbf{D}$ and vice versa. Figure 18 illustrates the distribution of the weighted singular points around the data cube $\mathbf{D}_U$ of size $500 \times 294 \times 64$ from the urban scene displayed in Fig. 15 whose source is Fig. 3.

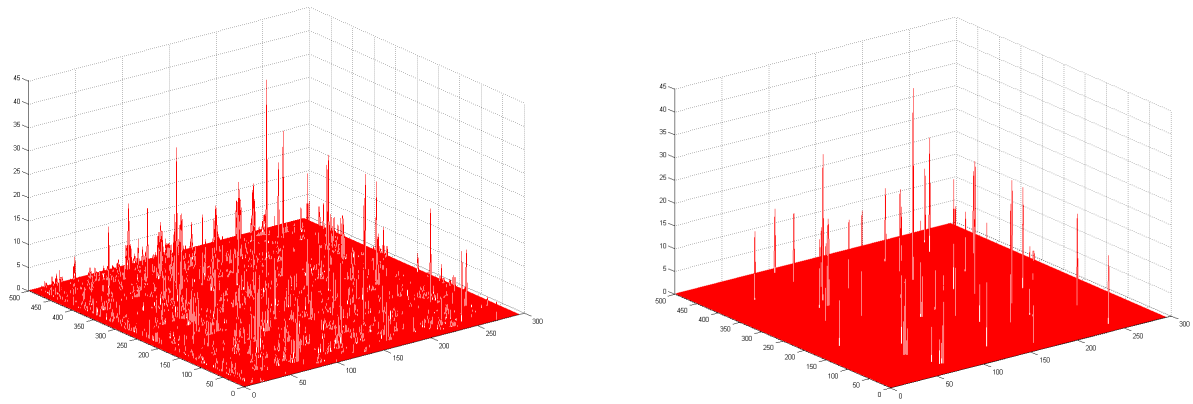

Figure 18: Distribution of weighted singular points around the data cube $\mathbf{D}_U$. Left: all singular points. Right: singular points whose weight exceed 12.

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 $\mathbf{D}_U$.

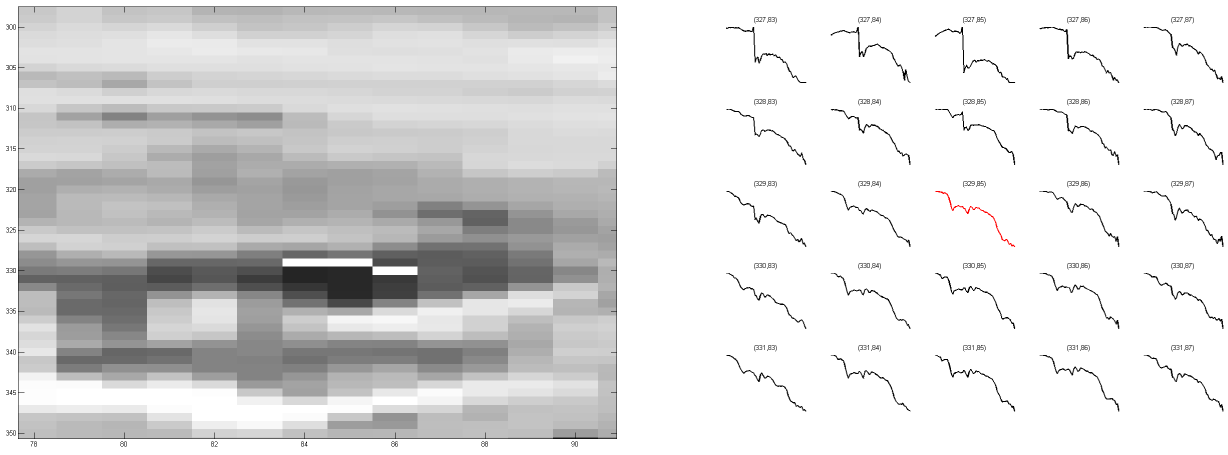

Figure 19: A group of singular points centered around the point $P(329, 85)$. Left: Vicinity of the point P. Right: Spectra of the multipixels at the point $P(399, 85)$ and the surrounding points. Weight of the data point P is 19.

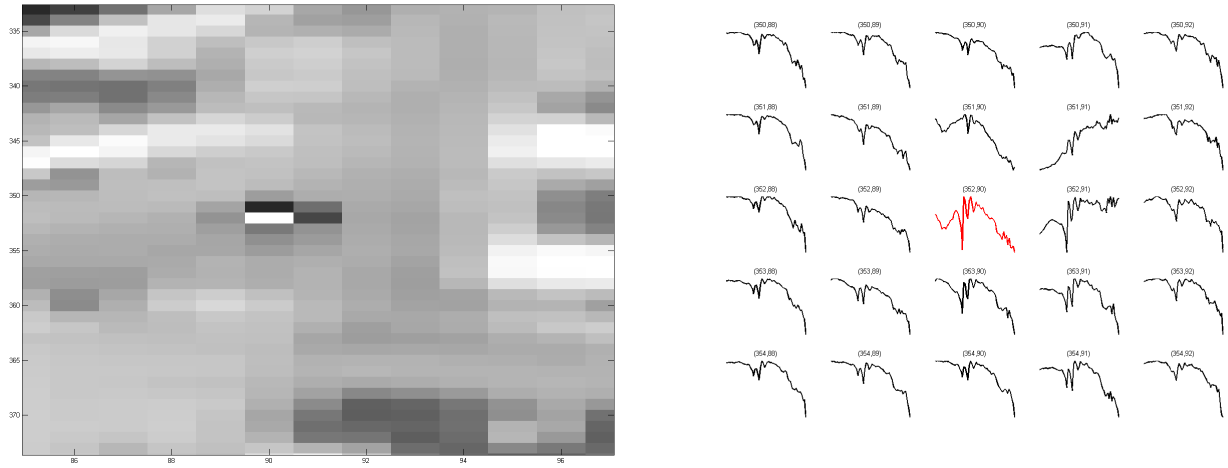

Figure 20: A strong singular point $P(352, 90)$. Left: Vicinity of the point P. Right: Spectra of the multipixels at the point $P(352, 90)$ and the surrounding points. Weight of the data point P is 32.

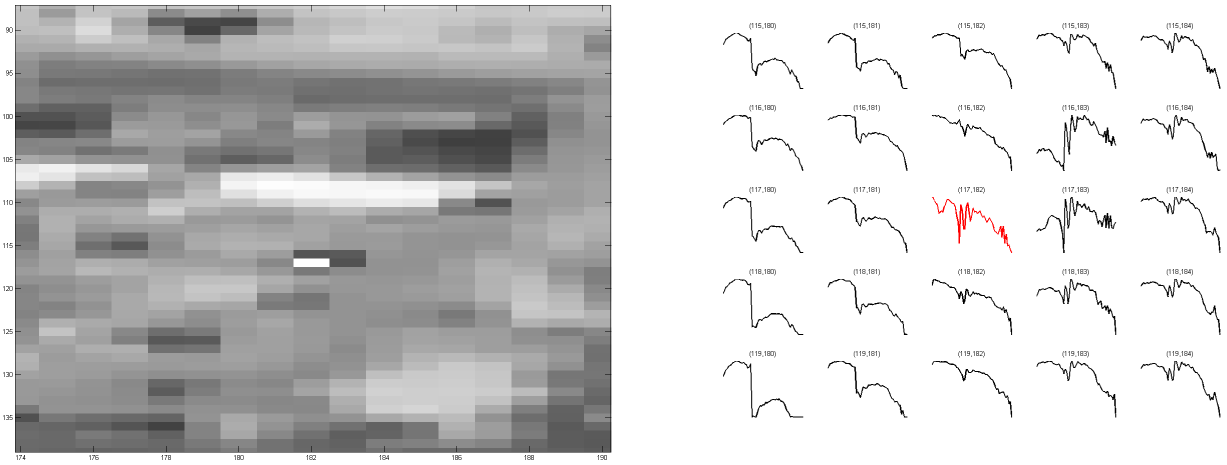

Figure 21: A strong singular point $P(117, 182)$. Left: Vicinity of the point P. Right: Spectra of the multipixels at the point $P(117_1 82)$ and the surrounding points. Weight of the data point P is 32.

461

462 By comparing between Figs. 21 and 20 we observe that spectra of singular multipixels
463 located at points $P(117_1 82)$ and $P(352, 90)$ are similar to each other. Supposedly, they
464 correspond to the same material. A different singular multipixel is displayed in Fig. 22.

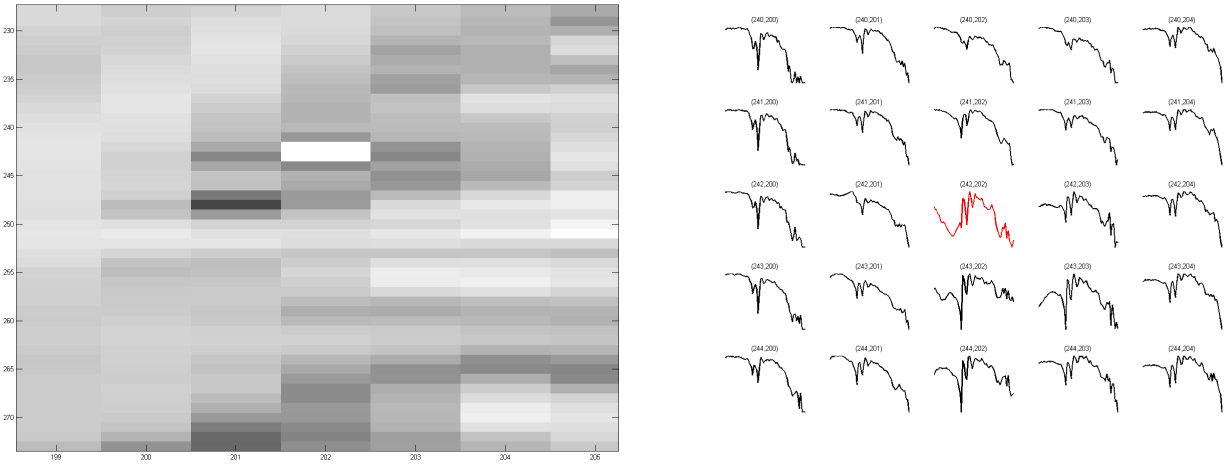

Figure 22: A singular point $P(242, 202)$. Left: Vicinity of the point P. Right: Spectra of the multipixels at the point $P(242, 202)$ and the surrounding points. Weight of the data point P is 32.

465

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 Eq. 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: a background average spectrum $B = \sum_{k=1}^{M} a_k B_k$, a mixed pixel spectrum (spectrum of a suspicious point) P and a target 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 $s_1 = T, t \in \mathbb{R}, t \in (0,1)$. All of $\{B_k\}$ 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 Eq. 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 $\hat{B}$, can be calculated as $\hat{B} = (P - \hat{t}T)/(1-\hat{t})$. Estimating the parameter t in Eq. 6 is called the linear unmixing.

In Step 2 in section 4.1, we calculated the following: V is the d-spectra set that 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 et al., 2001)). An analytical function can be represented as the Taylor expansion of its argument's degrees. Then, the condition $corr(\varphi(X_1), \varphi(X_2)) = 0$ equals to $corr((X_1)^n, (X_2)^n) = 0$ for any positive integer n where n denotes a power. In our algorithm, we limit ourself to $n = \{1,2,3,4\}$. From the independency criterion bteween two vectors X_1 and X_2, we can have

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

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.

4.5 Experimental results

In this section, we consider two scenes "field" and "city" 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 "morphology-filter" which was described in section 4.1. If the pixel is passed via the application of the "morphology-filter", then the target is present in it.

Figures. 23 and 24 present the outputs from the application of the "morphology-filter" algorithm to two different hyper-spectral scenarios.

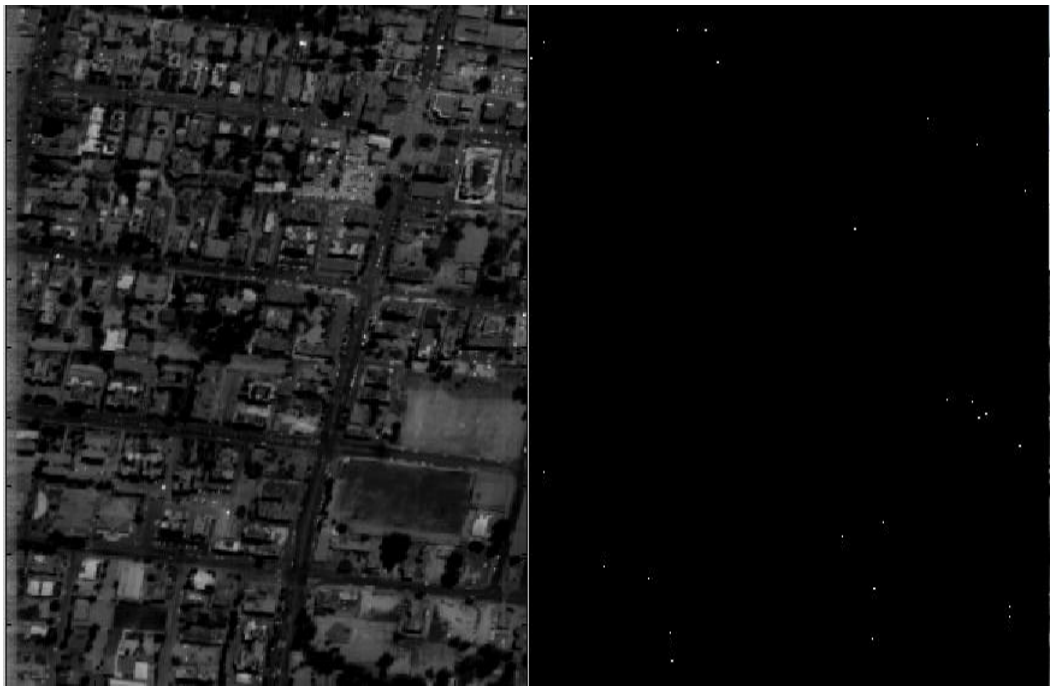

Figure 23: Left: The source image (Fig. 3). Right: The white points are the suspicious points in the neighborhood with diameter $m = 10$

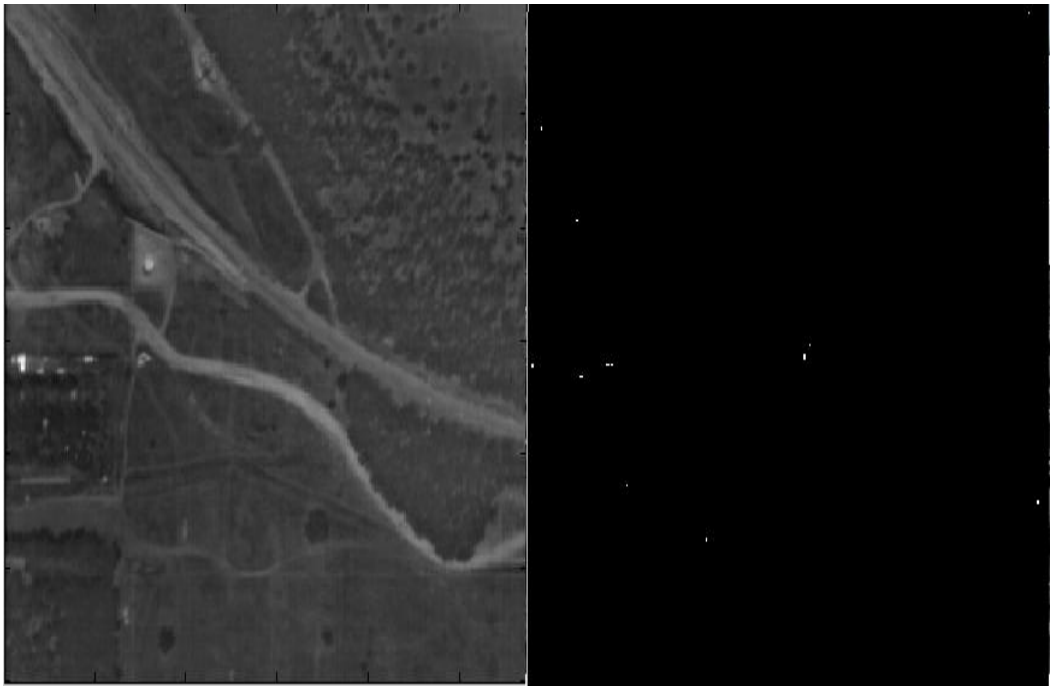

Figure 24: Left: The source image (Fig. 4). Right: The white points are the suspicious points in the neighborhood with diameter $m = 10$

In Figs. 25 and 26, the x- and y- axes are the wavebands and their values, respectively. The white points in Figs. 23 and 24 are the anomalies that contain the known target. The next step performs the unmixing procedure. The unmixing estimates the parameter t from Eq. 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 Eq. 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 "morphology-filter"(section 4.1).

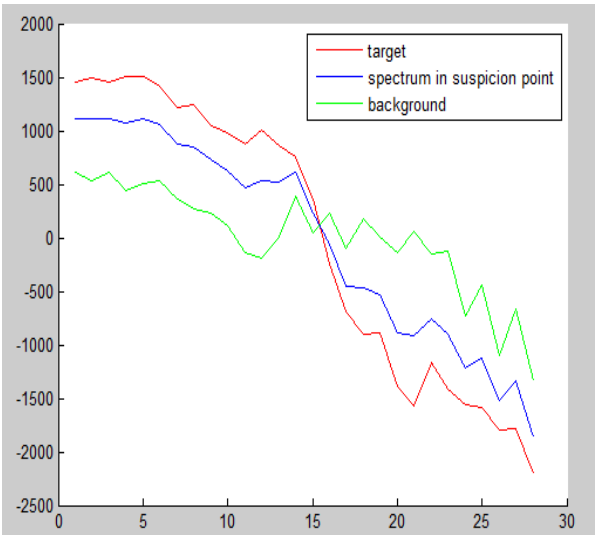

Figure 25: The result from the application of the UNSP unmixing algorithm to the suspicious point in Fig. 23. This suspicious point is decomposed into target and background spectral portions.

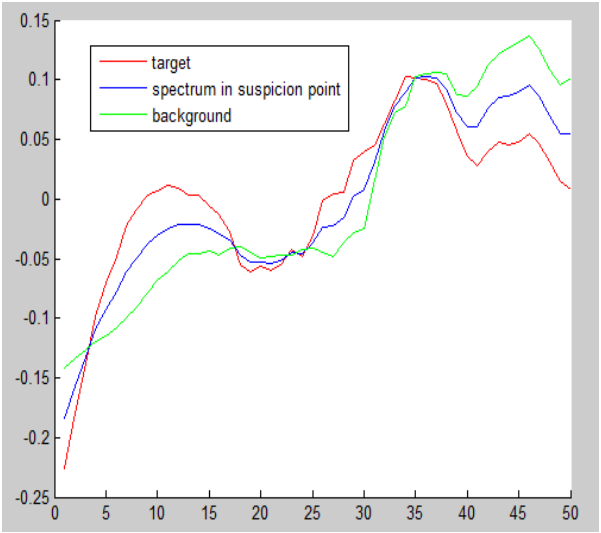

Figure 26: The result from the application of the UNSP unmixing algorithm to the suspicious point in Fig. 24. This suspicious point is decomposed into target and background spectral portions.

Conclusions

We presented two algorithms for linear unmixing. The first algorithm (titled WDR) works well but does not detect sub-pixel targets. The second algorithm (titled UNSP) works well for

detection of subpixels targets but it is computational expensive due to the need to search for the spectral decomposition in each pixel's neighborhood by moving the "morphology-filer". In a future research, we plan to add to these algorithms a classification method with machine learning methodologies to separate between the background and the targets' spectra.

References

Attias, H. (1999). Independent factor analysis. *Neural Computation*, *4*(11), 803–851.

Bateson, C., Asner, G., & Wessman, C. (2000). Endmember bundles: A new approach to incorporating endmember variability into spectral mixture analysis. *IEEE Trans Geosci Remote Sens*, *38*, 1083–1094.

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

Boardman, J. (1993). Automating spectral unmixing of AVIRIS data using convex geometry concepts. *in Summaries*, *1*, 1114.

Botchko, V., Berina, E., Korotkaya, Z., Parkkinen, J., & Jaaskelainen, T. (2003). Independent compo- nent analisys in spectral images. *In Proc of the*, *4th*, 203–207.

Burges, J. C. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, *2*.

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 Geosci Remote Sensing*, *3*(36), 898–912.

Chiang, S.-S., Chang, C., & Ginsberg, I. W. (2000). Unsupervised hyper-spectral image analysis using independent component analysis. *In Proc of the IEEE Int Geosci and Remote Sensing Symp*.

Common, P. (1994). Independent component analysis: A new concept. *Signal Processing*, pp. 287–314.

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

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

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). Pattern Classification. *optimized DJVU file with searchable text*, *2*.

Harsanyi, J. C., & Chang, C. (1994). Hyper-spectral image classification and dimensionality reduction: an orthogonal subspace projection approach. *IEEE Trans Geosci Remote Sensing*, *4*(32), 779–785.

http://www.specim.fi/. Specim camera..

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

Ifarraguerri, A., & Chang, C. I. (1999). Multispectral and hyper-spectral image analysis with convex cones. *IEEE Trans Geosci Remote Sens*, *37*, 756–770.

Keshava, N., Kerekes, J., Manolakis, D., & Shaw, G. (2000). An algorithm taxonomy for hyper-spectral unmixing. *In Proc of the SPIE AeroSense Conference on Algorithms for Multispectral and hyper-spectral Imagery VI*, *4049*, 42–63.

Moulines, E., Cardoso, J., & Gassiat, E. (1997). Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models. *In Proc of the IEEE Int Conf on Acoustics Speech and Signal Processing*, *5*, 3617–3620.

Nascimento, M. P., & Bioucas-Dias, M. (2005a). Does independent component analysis play a role in unmixing hyper-spectral data?. *IEEE Trans Geosci Remote Sensing*, *1*(43), 175–187.

Nascimento, M. P., & Bioucas-Dias, M. (2005b). Vertex component analysis: A fast algorithm to unmix hyper-spectral data. *IEEE Trans Geosci Remote Sensing*, *4*(43), 898–910.

Parra, L., Mueller, K., Spence, C., Ziehe, A., & Sajda, P. (2000). Unmixing hyper-spectral data. *Advances in Neural Information Processing Systems*, pp. 942–948.

572 Settle, J. J. (1996). On the relationship between spectral unmixing and subspace projection.
573 *IEEE Trans Geosci Remote Sensing*, pp. 1045–1046.

574 Tu, T. M. (2000). Unsupervised signature extraction and separation in hyper-spectral images:
575 A noise-adjusted fast independent component analysis approach. *Optical Engineering*
576 *of SPIE*, *4*(39), 897–906.

577 Winter, M. E. (1999). N-findr: An algorithm for fast autonomous spectral end-member
578 determination in hyper-spectral data. *in Proc SPIE Conf Imaging Spectrometry V*,
579 266–275.