

Fast Implementation of Pixel Purity Index Algorithm

Antonio Plaza^{a,b} and Chein-I Chang^b

^aComputer Science Department, University of Extremadura,
Avda. de la Universidad s/n, 10.071 Cáceres, Spain;

^bRemote Sensing Signal and Image Processing Laboratory,
Department of Computer Science and Electrical Engineering,
University of Maryland Baltimore County, Baltimore, MD 21250

ABSTRACT

Pixel purity index (PPI) algorithm has been widely used in hyperspectral image analysis for endmember extraction because of its publicity and availability in the Research Systems ENVI software. In this paper, we develop a fast algorithm to implement the PPI, which provides several significant advantages over the PPI. First, it uses a newly developed concept, virtual dimensionality (VD) to estimate the number of endmembers required to be generated by the algorithm. Second, it uses an endmember initialization algorithm (EIA) to generate an appropriate set of initial endmembers that can reduce a significant number of runs required for the PPI. Third, it provides a new iterative rule and a stopping rule to terminate the algorithm, a feature that is not available in the original PPI which is not an iterative algorithm. Most importantly, unlike the PPI which requires a visualization tool to manually select a final set of endmembers, the FPPI is completely automatic and unsupervised. Since the original PPI algorithm has never been fully disclosed in the literature due to its propriety, the step-by-step algorithmic implementation of FPPI presented in this paper is considered to be new and may be very beneficial to users who are interested in this algorithm without soliciting help from particular software.

Keywords: Endmember extraction algorithm (EEA). Endmember initialization algorithm (EIA). Pixel purity index (PPI). Virtual dimensionality (VD).

1. INTRODUCTION

Endmember extraction is considered to be an important and crucial step in hyperspectral data exploitation. An endmember can be defined as an idealized, pure signature for a class¹. Therefore, an endmember is generally not a pixel. Instead, it is a spectral signature that is completely specified by the spectrum of a single material substance. Accordingly, a pixel is pure if its spectral signature is an endmember. To the contrary, a pixel is mixed if its spectral signature is not an endmember.

Over the last decade, several endmember extraction algorithms (EEAs) have been developed². One of the most popular ones has been Boardman's Pixel Purity Index (PPI), designed to search for a set of vertices of a convex hull^{3,4}. The concept of the original PPI is rather simple. The algorithm proceeds by generating a large number, k , of L -dimensional random vectors called "skewers." All pixel vectors in the input data are then projected onto each of the skewers, and the pixel vectors resulting in the maximum projection scores increase their pixel purity score. After many repeated projections, those pixels with score above a certain cut-off threshold, t , are declared "pure." These potential endmember spectra are then loaded into an interactive L -dimensional visualization tool to manually select pixel vectors that correspond to pure pixel vectors. The PPI algorithm has been widely used in the remote sensing community due to its publicity and availability, provided by Research Systems's ENVI software package, which also provides a tool called " L -dimensional visualizer" that allows a trained user to select endmembers by visual inspection. It should be noted that the PPI algorithm was originally conceived not as a solution to the endmember extraction problem, but as a guide. In fact, the author proposed comparing the pure pixels with library spectra and successively projecting the data toward lower dimensional spaces while endmembers were identified³. In other words, the original PPI requires the intervention of a trained user to select a final set of endmembers.

Although PPI has been successfully applied to many problems related with hyperspectral data exploitation, our experience with the algorithm revealed several important issues resulting from its implementation. First and foremost, a detailed implementation of PPI has never been fully disclosed in the literature due to the algorithm's propriety. Resultingly, most potential users of PPI need to appeal for the ENVI software in order to run PPI. A second issue is the algorithm's sensitivity to parameters k and t , which can be highly application-dependent and subjective. Another shortcoming is randomness in the generation of skewers. The original implementation of PPI suggests the generation of randomly generated skewers. It is well known that the PPI algorithm requires a high number of skewers to provide a reliable identification of image endmembers, a fact that results in very high computational complexity. However, intelligent selection of existing vectors to skew the data set can reduce the number of required skewers. Most importantly, the PPI is not an iterative process and does not guarantee its convergence in finite runs, in spite of that it may converge asymptotically as claimed by the authors.

In order to address the important issues above, which have been recently studied and analyzed in detail^{5,6}, this paper develops a fast algorithm for implementation of PPI, referred to as fast PPI (FPPI). It has several significant advantages over the original PPI. First, it makes use of a newly developed concept, virtual dimensionality (VD)^{7,8} to estimate the number of endmembers, p , required to be generated. This allows us to replace the two parameters, k and t used in the PPI so that the issue of their sensitivity to number of runs and cut-off threshold value is resolved. Second, the FPPI takes advantage of an automatic target generation process (ATGP) to generate an appropriate set of initial endmembers that can reduce a significant number of runs required for the PPI. Here, we make a clear distinction between "run" and "iteration" because ENVI's PPI is not iterative algorithm as explained above, while the FPPI is an iterative algorithm that provides a new iterative rule and a stopping criterion to ease tremendous computation involved in the PPI. Most importantly, unlike the PPI which requires an " L -dimensional visualizer" to manually select a final set of endmembers, the FPPI is completely automatic and unsupervised. In other words, while different users of PPI may select different sets of endmembers for the same data set, the final set of FPPI-generated endmembers is always the same, regardless of who is a user of the FPPI. This, along with the significant savings in computational time, are considered to be the most significant advantages of the FPPI over the PPI.

The remainder of this paper is organized as follows. Section 2 introduces the concept of the virtual dimensionality (VD) used to estimate the number of endmembers, and develops a simple method to find the VD. Section 3 presents an automatic target generation process (ATGP) to produce a set of initial endmembers. Section 4 describes a new fast iterative algorithm for implementation of the PPI. Section 5 conducts extensive computer simulations and real data experiments to develop a comparative study between the PPI and the FPPI. Finally, section 6 concludes with some remarks.

2. ESTIMATION OF THE NUMBER OF ENDMEMBERS

The virtual dimensionality (VD) is a new concept defined and coined by Chang^{7,8} which has shown to provide a good estimate of the number of image endmember signatures. Here, we describe the method developed by Harsanyi et al., referred to as Harsanyi-Farrand-Chang's (HFC) method⁹ which has been shown to be effective in determining the number of endmembers. Let $\mathbf{R}_{L \times L}$ and $\mathbf{K}_{L \times L}$ denote the sample correlation matrix and the sample covariance matrix respectively. Let us also assume that L is the number of spectral channels. Further, let $\{\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \dots \geq \hat{\lambda}_L\}$ and $\{\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_L\}$ be two sets of eigenvalues generated by $\mathbf{R}_{L \times L}$ and $\mathbf{K}_{L \times L}$, called correlation eigenvalues and covariance eigenvalues, respectively. By assuming that signal sources are nonrandom unknown positive constants and noise is white with zero mean, we can expect that

$$\hat{\lambda}_l > \lambda_l \text{ for } l = 1, 2, \dots, \text{VD}, \quad (1)$$

and

$$\hat{\lambda}_l = \lambda_l \text{ for } l = \text{VD} + 1, \dots, L. \quad (2)$$

Using Eqs. (1-2), the eigenvalues in the l -th spectral channel can be related by

$$\hat{\lambda}_l > \lambda_l > \sigma_{n_l}^2 \text{ for } l = 1, \dots, \text{VD}$$

and

$$\hat{\lambda}_l = \lambda_l = \sigma_{n_l}^2 \text{ for } l = \text{VD} + 1, \dots, L \quad (3)$$

where $\sigma_{n_l}^2$ is the noise variance in the l -th spectral channel. In order to determine the VD, Harsanyi et al.⁹ formulated the VD determination problem as a binary hypothesis problem as follows.

$$\begin{aligned} H_0 : \quad z_l = \hat{\lambda}_l - \lambda_l = 0 \\ \text{versus} \quad \quad \quad \text{for } l = 1, 2, \dots, L \\ H_1 : \quad z_l = \hat{\lambda}_l - \lambda_l > 0 \end{aligned} \quad (4)$$

where the null hypothesis H_0 and the alternative hypothesis H_1 represent the case that the correlation-eigenvalue is equal to its corresponding covariance eigenvalue, and the case that the correlation-eigenvalue is greater than its corresponding covariance eigenvalue, respectively. In other words, when H_1 is true (i.e., H_0 fails), it implies that there is an endmember contributing to the correlation-eigenvalue in addition to noise, since the noise energy represented by the eigenvalue of $\mathbf{R}_{L \times L}$ in that particular component is the same as the one represented by the eigenvalue of $\mathbf{K}_{L \times L}$ in its corresponding component.

Despite the fact that the $\hat{\lambda}_l$ and λ_l in Eqs. (1-4) are unknown constants, we can model each pair of eigenvalues, $\hat{\lambda}_l$ and λ_l , under hypotheses H_0 and H_1 as random variables by the asymptotic conditional probability densities given by

$$p_0(z_l) = p(z_l | H_0) \cong N(0, \sigma_{z_l}^2) \text{ for } l = 1, 2, \dots, L \quad (5)$$

and

$$p_1(z_l) = p(z_l | H_1) \cong N(\mu_l, \sigma_{z_l}^2) \text{ for } l = 1, 2, \dots, L \quad (6)$$

respectively, where μ_l is an unknown constant and the variance $\sigma_{z_l}^2$ is given by

$$\sigma_{z_l}^2 = \text{var}[\hat{\lambda}_l - \lambda_l] = \text{var}[\hat{\lambda}_l] + \text{var}[\lambda_l] - 2\text{Cov}(\hat{\lambda}_l, \lambda_l) \text{ for } l = 1, 2, \dots, L. \quad (7)$$

It can be demonstrated that, when the total number of samples, N , is sufficiently large, then $\text{var}[\hat{\lambda}_l] \cong 2\hat{\lambda}_l^2 / N$ and $\text{var}[\lambda_l] \cong 2\lambda_l^2 / N$. Therefore, the noise variance $\sigma_{z_l}^2$ in Eq. (6)-(7) can be estimated and approximated using Eq. (7). Now, we use Schwarz's inequality to bound $\text{Cov}(\hat{\lambda}_l, \lambda_l)$ in Eq. (7) as follows

$$\text{Cov}(\hat{\lambda}_l, \lambda_l) \leq \sqrt{\text{var}[\hat{\lambda}_l] \text{var}[\lambda_l]} \cong \frac{2}{N} (\hat{\lambda}_l \cdot \lambda_l). \quad (8)$$

If we further assume that the estimators $\hat{\lambda}_l$ and λ_l are consistent in mean square, the variances of $\hat{\lambda}_l$ and λ_l are asymptotically zero. In other words, $\text{var}[\hat{\lambda}_l] \cong 2\hat{\lambda}_l^2 / N$ and $\text{var}[\lambda_l] \cong 2\lambda_l^2 / N$ converge to zero as $N \rightarrow \infty$. This further implies that $\text{Cov}(\hat{\lambda}_l, \lambda_l) \rightarrow 0$ as well as

$$\sigma_{z_l}^2 = \text{var}[\hat{\lambda}_l] + \text{var}[\lambda_l] \cong (2\hat{\lambda}_l^2 / N) + (2\lambda_l^2 / N) \rightarrow 0 \text{ for } l = 1, 2, \dots, L \text{ as } N \rightarrow \infty. \quad (9)$$

From Eqs. (6) and (9), we define the false alarm probability and detection power (i.e., detection probability) as follows:

$$P_F = \int_{\tau_l}^{\infty} p_0(z) dz \quad (10)$$

$$P_D = \int_{\tau_l}^{\infty} p_1(z) dz. \quad (11)$$

A Neyman-Pearson detector for $\hat{\lambda}_l - \lambda_l$, denoted by $\delta_{NP}(\hat{\lambda}_l - \lambda_l)$ for the binary composite hypothesis testing problem specified by Eq. (4) can be obtained by maximizing the detection power P_D in Eq. (11), while the false alarm probability P_F in Eq. (10) is fixed at a specific given value, α , which determines the threshold value τ_l in Eqs. (10-11). So, a case of $\hat{\lambda}_l - \lambda_l > \tau_l$ indicating that $\delta_{NP}(\hat{\lambda}_l - \lambda_l)$ fails the test, in which case there is signal energy assumed to contribute to the eigenvalue, $\hat{\lambda}_l$ in the l -th data dimension. It should be noted that the test for Eq. (4) must be performed for each of L spectral dimensions. Therefore, for each pair of $\hat{\lambda}_l - \lambda_l$ the threshold τ is different and should be l -dependent, that is τ_l . Since HFC's method does not require a noise whitening process, weak signal sources may be obscured by noise and cannot be detected. As an alternative, the HFC method can be modified by including a noise whitening process as a preprocessing to remove the second-order statistical correlation so that signal sources can be decorrelated from noise to achieve better signal detection. The resulting HFC method will be referred to as noise-whitened HFC (NWHFC) method⁸.

3. AUTOMATED TARGET GENERATION PROCESS

This section briefly describes a method to intelligently generate a set of skewers for the PPI algorithm. The automatic target generation process (ATGP) was previously developed to find potential target pixels that can be used to generate a set of initial endmembers using an orthogonal subspace projection (OSP) approach¹⁰. It makes use of an orthogonal subspace projector defined in [8] by

$$P_U^\perp = \mathbf{I} - \mathbf{U}(\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T. \quad (12)$$

The ATGP repeatedly makes use of Eq. (12) to find target pixel vectors of interest from the data without prior knowledge regardless of what types pixels of these targets. It can be briefly described as follows. Let us assume that \mathbf{t}_0 is an initial target pixel vector with the maximum length, namely, $\mathbf{t}_0 = \arg\{\max_r \mathbf{r}^T \mathbf{r}\}$, i.e., the brightest pixel vector in the image scene. It is worth noting that this selection may not be necessarily the best selection. However, according to our experiments it was found that the brightest pixel vector was always extracted later on, if it was not used as an initial target pixel vector in the initialization. The ATGP begins with the initial target pixel vector \mathbf{t}_0 by applying an orthogonal subspace projector P_U^\perp specified by Eq. (12) with $\mathbf{U} = \mathbf{t}_0$ to all image pixel vectors. It then finds a target pixel vector, denoted by \mathbf{t}_1 with the maximum orthogonal projection in the orthogonal complement space, denoted by $\langle \mathbf{t}_0 \rangle^\perp$ that is orthogonal to the space $\langle \mathbf{t}_0 \rangle$ linearly spanned by \mathbf{t}_0 . The reason for this selection is that the selected \mathbf{t}_1 generally has the most distinct features from \mathbf{t}_0 in the sense of orthogonal projection because \mathbf{t}_1 has the largest magnitude of the projection in $\langle \mathbf{t}_0 \rangle^\perp$ produced by $P_{\mathbf{t}_0}^\perp$. A second target pixel vector \mathbf{t}_2 can be found by applying an orthogonal subspace projector P_U^\perp with $\mathbf{U} = [\mathbf{t}_0 \ \mathbf{t}_1]$ to the original image, and a target pixel vector that has the maximum orthogonal projection in $\langle \mathbf{t}_0, \mathbf{t}_1 \rangle^\perp$ is selected as \mathbf{t}_2 . The above procedure is repeated over and over again to find a third target pixel vector \mathbf{t}_3 , a fourth target pixel vector \mathbf{t}_4 and so on, until a final target set which comprises p target vectors, $\{\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_{p-1}\} = \{\mathbf{t}_0\} \cup \{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_{p-1}\}$, is generated, where p is estimated by using the VD introduced in Section 2.

4. FAST PIXEL PURITY INDEX ALGORITHM

This section develops a fast algorithm for implementation of PPI which makes use of the VD concept in section 2 and an endmember initialization algorithm in section 3 to address four major shortcomings of the original PPI algorithm. The

first limitation of the original PPI is its computational complexity, which generally requires a dimensionality reduction via the maximum noise fraction (MNF) transform¹¹. Our proposed implementation, however, is a fast algorithm that does not need a previous dimensionality reduction. Second, opposite to PPI, the FPPI is an iterative algorithm where the termination criteria is given by the number of endmembers, p , estimated by the VD in section 2. The parameter p replaces parameters k and t , which are data-dependent and rather subjective. Finally, the FPPI starts with an appropriate set of initial endmembers and iteratively refines this initial selection until a stopping rule is satisfied. These features are missing from the original PPI, which is not an iterative algorithm. As will be shown by experiments, many of the initial endmembers generated by ATGP in section 3 turn out to be final endmembers, a fact that can save tremendous computation time. In the following, we provide a step-by-step implementation of FPPI algorithm.

FPPI Algorithm

1) *Initialization:*

Assume that p , the number of endmembers required to generate is estimated by the HFC method in Section 2. Let $\{\mathbf{e}_j^{(0)}\}_{j=1}^p$ be the initial set of p endmembers generated by ATGP in Section 3.

2) *Iterative rule:*

At iteration $k \geq 0$, for each $\mathbf{skewer}_j^{(k)}$ we project all the data sample vectors onto this particular $\mathbf{skewer}_j^{(k)}$ to find those sample vectors that are at its extreme positions to form an extrema set, denoted by $S_{\text{extrema}}(\mathbf{skewer}_j^{(k)})$.

Despite the fact that a different \mathbf{skewer}_j generates a different extrema set $S_{\text{extrema}}(\mathbf{skewer}_j^{(k)})$, it is very likely that some sample vectors may appear in more than one extrema set. Define an indicator function of a set S , $I_S(\mathbf{r})$ by the following expression:

$$I_S(\mathbf{r}) = \begin{cases} 1; & \text{if } \mathbf{r} \in S \\ 0; & \text{if } \mathbf{r} \notin S \end{cases} \text{ and } N_{PPI}(\mathbf{r}) = \sum_j I_{S_{\text{extrema}}(\mathbf{skewer}_j^{(k)})}(\mathbf{r}). \quad (13)$$

3) *Pure pixel candidate selection:*

Find the sample vectors, denoted by $\{\mathbf{r}_j^{(k)}\}_{j=1}^p$, that produce the first p largest values of $N_{PPI}(\mathbf{r})$.

4) *Stopping rule:*

Find the p largest values of N_{PPI} in the joint set of $\{N_{PPI}(\mathbf{r}_j^{(k)})\}_{j=1}^p \cup \{N_{PPI}(\mathbf{skewer}_j^{(k)})\}_{j=1}^p$ and let $\{N_{PPI}(\mathbf{skewer}_j^{(k+1)})\}_{j=1}^p$ be those vectors corresponding to the obtained p largest values of N_{PPI} , denoted by $\{\mathbf{e}_j^{(k+1)}\}_{j=1}^p$. If $\{N_{PPI}(\mathbf{skewer}_j^{(k+1)})\}_{j=1}^p = \{N_{PPI}(\mathbf{skewer}_j^{(k)})\}_{j=1}^p$, that is, $\{\mathbf{e}_j^{(k+1)}\}_{j=1}^p = \{\mathbf{e}_j^{(k)}\}_{j=1}^p$, then there is no more endmember has been replaced. In this case, the algorithm is terminated. Otherwise, let $k \leftarrow k + 1$ and go to step 2.

5. EXPERIMENTS

This section develops a comparative study between the PPI and the FPPI algorithms. The image data to be used for experiments is a well-known Airborne Visible Infrared Imaging Spectrometer (AVIRIS) image cube in radiance units shown in Fig. 1(a), collected over the Cuprite mining site, Nevada, in 1997. This 224-band scene is well understood mineralogically, and has reliable ground truth in the form of a library of mineral spectra collected at the site by USGS, where several pure pixels made up of five minerals of interest, namely, alunite, buddingtonite, calcite, kaolinite and muscovite are white-circled and labeled by letters A, B, C, K and M in Fig. 1(b), with their corresponding USGS spectra shown in Fig. 1(c). The availability of ground-truth has made this scene a standard test site for comparison of EEAs as shown in previous work¹². The AVIRIS Cuprite data, available online at <http://aviris.jpl.nasa.gov>, was used in computer simulations and real image experiments as follows.

5.1. Synthetic data experiments

The reflectance spectra of mineral spectra in Fig. 1(c) were used to create a synthetic scene. First of all, a background image with size of 20×20 pixels was simulated by mixing 50%-alunite and 50%-kaolinite. Next, three sets, each of which has three 2×2 panels were simulated by three signatures, buddingtonite, calcite and muscovite as tabulated in Table 1, $\{p_{ij}^k\}_{i=1,j=1}^{2,2}$, $\{\tilde{p}_{ij}^k\}_{i=1,j=1}^{2,2}$ and $\{\hat{p}_{ij}^k\}_{i=1,j=1}^{2,2}$ for $k=1,2,3$ and shown in Fig. 2(a). It should be noted that the notations, p , \tilde{p} and \hat{p} denote three sets of the pixels in the 2×2 panels in rows 1, 2 and 3 respectively, while the superscript “ k ” indicates the k -th 2×2 panel in each of the three sets. For example, the first set of the three 2×2 panels $\{p_{ij}^k\}_{i=1,j=1}^{2,2}$ was simulated as follows. The four pixels in the first 2×2 panel, $\{p_{ij}^1\}_{i=1,j=1}^{2,2}$ were all simulated by 100% pure buddingtonite signature. They were considered as pure pixels made by buddingtonite. For the second 2×2 panel, its two top pixels p_{11}^2, p_{12}^2 were simulated by 100% pure buddingtonite, while the two bottom left and right pixels p_{21}^2, p_{22}^2 were mixed pixels simulated by (75%-calcite, 25%-muscovite) and (25%-calcite, 25%-muscovite) respectively. For the third 2×2 panel, its top left pixel, p_{11}^3 was simulated by 100% pure buddingtonite and the other three (i.e., top right, bottom left and bottom right pixels), $p_{12}^3, p_{21}^3, p_{22}^3$ were simulated by (50%-calcite, 50%-muscovite), (50%-buddingtonite, 50%-calcite) and (50%-buddingtonite, 50%-muscovite) respectively. Similarly, the second and third sets of the three 2×2 panels, $\{\tilde{p}_{ij}^k\}_{i=1,j=1}^{2,2}$ and $\{\hat{p}_{ij}^k\}_{i=1,j=1}^{2,2}$ were also simulated by calcite and muscovite respectively. These three sets of three 2×2 panels in Fig. 2(a) were then implanted in order (1st row: buddingtonite $\{p_{ij}^k\}_{i=1,j=1}^{2,2}$; 2nd row: calcite $\{\tilde{p}_{ij}^k\}_{i=1,j=1}^{2,2}$; 3rd row: muscovite $\{\hat{p}_{ij}^k\}_{i=1,j=1}^{2,2}$) in the background image at the center to generate the synthetic image shown in Fig. 2(b).

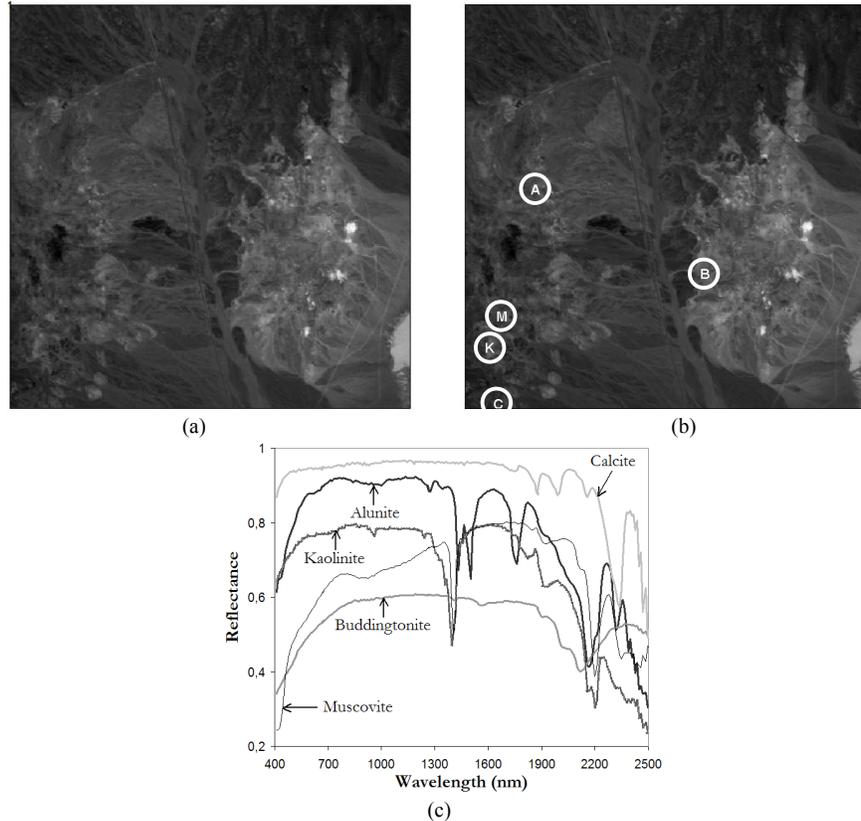


Figure 1. (a) Spectral band at 827 nm of the Cuprite AVIRIS image scene; (b) Spatial positions of pure pixels made up of Alunite (A), Buddingtonite (B), Calcite (C), Kaolinite (K) and Muscovite (M); (c) USGS spectral signatures.

Panel pixel	Buddingtonite	Calcite	Muscovite
$\{p_{ij}^1\}_{i=1,j=1}^{2,2}, p_{11}^2, p_{12}^2, p_{11}^3$	100%	0%	0%
p_{21}^2	0%	75%	25%
p_{22}^2	0%	25%	75%
p_{12}^3	0%	50%	50%
p_{21}^3	50%	50%	0%
p_{22}^3	50%	0%	50%
$\{\tilde{p}_{ij}^1\}_{i=1,j=1}^{2,2}, \tilde{p}_{11}^2, \tilde{p}_{12}^2, \tilde{p}_{11}^3$	0%	100%	0%
\tilde{p}_{21}^2	75%	0%	25%
\tilde{p}_{22}^2	25%	0%	75%
\tilde{p}_{12}^3	50%	0%	50%
\tilde{p}_{21}^3	50%	50%	0%
\tilde{p}_{22}^3	0%	50%	50%
$\{\bar{p}_{ij}^1\}_{i=1,j=1}^{2,2}, \bar{p}_{11}^2, \bar{p}_{12}^2, \bar{p}_{11}^3$	0%	0%	100%
\bar{p}_{21}^2	75%	25%	0%
\bar{p}_{22}^2	25%	75%	0%
\bar{p}_{12}^3	50%	50%	0%
\bar{p}_{21}^3	50%	0%	50%
\bar{p}_{22}^3	0%	50%	50%

Table 1. Three sets of 2×2 panels simulated by three USGS signatures: buddingtonite, calcite and muscovite.

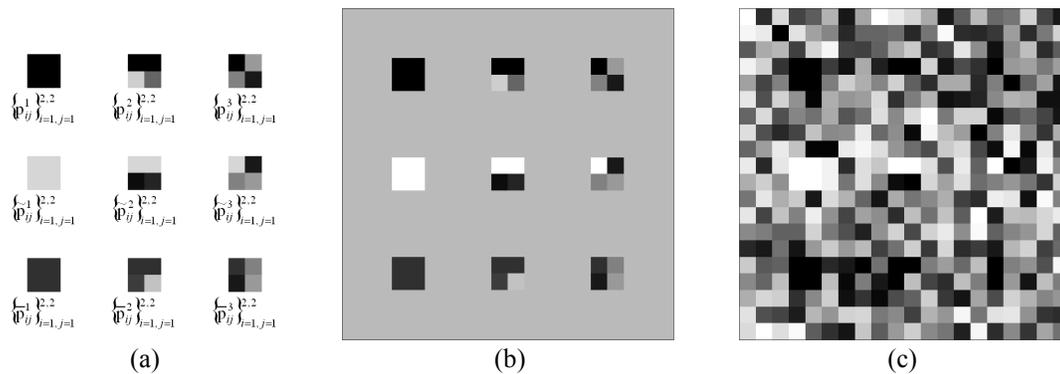


Figure 2. (a) Nine 2×2 panels simulated by buddingtonite, calcite and muscovite; (b) Synthetic image obtained after implanting the panels in a clean background image; (c) Synthetic image obtained after implanting the panels in a background image corrupted by Gaussian noise with SNR of 20:1.

Example 1. Synthetic data with no noise

According to the synthetic image description in Table 1, there were 19 distinct signatures, of which there were only three 100% pure signatures, buddingtonite, calcite, muscovite, and 16 mixed signatures. Therefore, in this synthetic image, $p = 3$. Since there are 19 distinct signatures, we ran the ATGP to generate 19 target pixels. However, due to the fact that only five mineral signatures, alunite, buddingtonite, calcite, kaolinite and muscovite were used to create the synthetic image,

the ATGP could only generate 5 target pixels shown in Fig. 3(a) according to Eq. (12). After the 5 pixels were generated, the ATGP ran into a singular problem for the matrix inverse in Eq. (12). This made sense because all the image pixels in Fig. 2(b) were linearly mixed by the five mineral signatures, and the ATGP is a linear technique which can only unmix five signatures. Now, if we used the ATGP-generated pixels as the initial endmember pixels for FPPI, the algorithm extracted four pixels in Fig. 3(b). Interestingly, despite the fact that the background pixel was not a pure pixel, it was extracted as a pure pixel by the FPPI. This was because the background signature simulated by mixing 50%-alunite and 50%-kaolinite was considered as the pure signature of a mixture of 50%-alunite and 50%-kaolinite. For illustrative purposes, we also ran ENVI's PPI algorithm on the synthetic scene, where Fig. 3(c) shows that this algorithm extracted the same four pixels as the FPPI in different order.

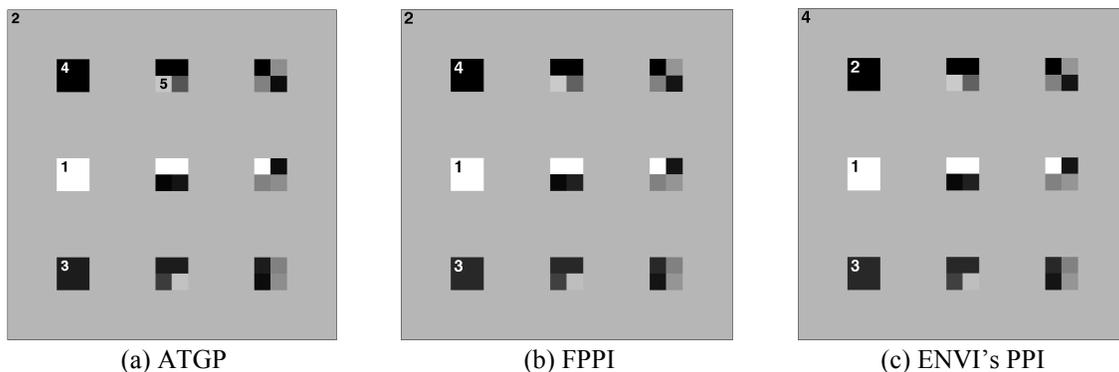


Figure 3. Results for a synthetic data with no simulated noise: (a) Five target pixels generated by ATGP algorithm; (b) Four endmember pixels extracted by FPPI; (c) Four endmember pixels extracted by ENVI's PPI.

Example 2. Noise effect

The same simulated image used in Example 1 was also used in this example, except that the background image was corrupted by Gaussian noise with SNR 20:1 as shown in Fig. 2(c). In this case, we did not have prior knowledge about how many endmembers, p , were present in the image. In order to estimate this number, we used the HFC and NWHFC methods in section 2 to determine the VD. Table 2 tabulates the estimated VD based on various false alarm probabilities. According to the table, an appropriate estimate seemed to be $p = 3$ for $P_F = 10^{-3}$.

Method	$P_F = 10^{-1}$	$P_F = 10^{-2}$	$P_F = 10^{-3}$	$P_F = 10^{-4}$	$P_F = 10^{-5}$
HFC	4	3	3	2	2
NWHFC	4	3	3	2	2

Table 2. VD estimated by the HFC and NWHFC methods for the synthetic image in Fig. 2(c).

In this case, only three target pixels were generated by ATGP as shown by Fig. 4(a). Interestingly, ATGP extracted only two endmembers plus a background pixel, while both FPPI and ENVI's PPI extracted all three pure pixels in exactly the same order. In this regard, the two considered EEAs performed exactly as expected, while ATGP extracted the most significant pixels in terms of distinct signatures, but not necessarily pure signatures. It should be noted that the background signature was considered as a crucial and critical signature by ATGP because it was used to create the entire background, although it is not a pure signature.

Results in Fig. 4 raise an important question about the performance of an EEA, that is, what pixels an EEA would extract if there were no pure pixels in the data? In order to answer this question, Fig. 5 shows what would happen if a pixel was extracted and removed subsequently before a new pixel is processed in Examples 1 and 2. Interestingly, such a remove-before-extract strategy allows both FPPI and PPI to extract all endmember pixels specified by the same pure signature before extracting the mixed signatures in the scene. After exhausting all endmember pixels, the mixed pixels that were most likely to be extracted next by both algorithms were those containing the purest signatures, as shown by Fig. 5. It is remarkable that the two algorithms, PPI and FPPI, produced exactly the same results in the two examples, a fact that

reveals that FPPI can indeed produce the same results as those found by ENVI's PPI, but with significant savings in computation since no manual processing by a trained user is required.

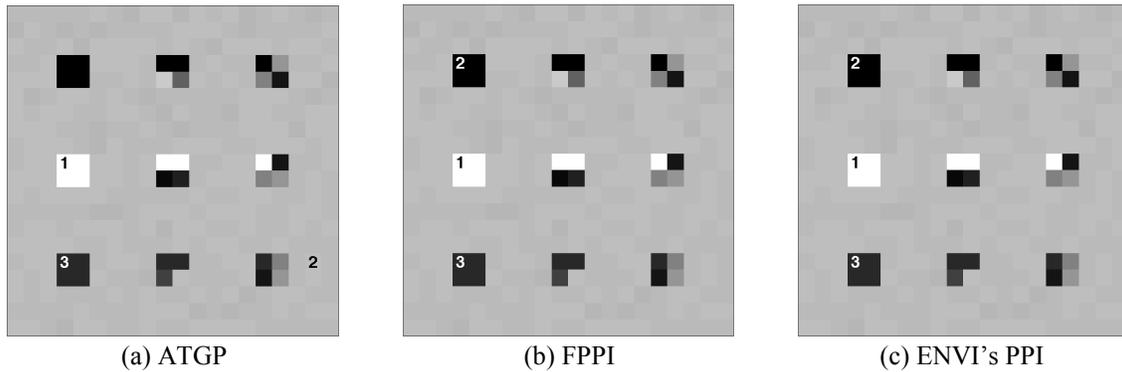


Figure 4. Results for a synthetic data corrupted by noise with SNR of 20:1: (a) Three target pixels generated by ATGP algorithm; (b) Three endmember pixels extracted by FPPI; (c) Three endmember pixels extracted by ENVI's PPI.

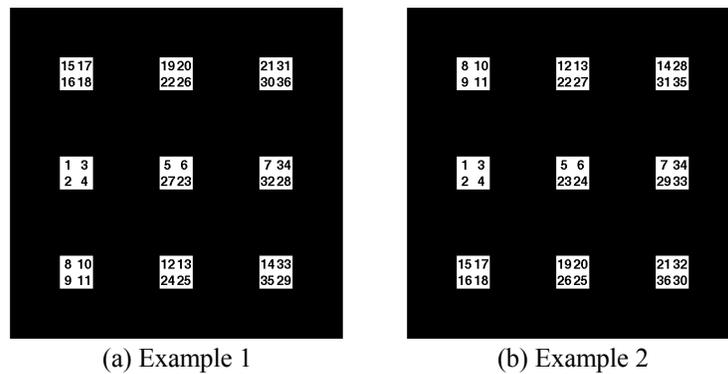


Figure 5. Results produced by the FPPI and ENVI's PPI using the remove-before-extract strategy.

5.2. Real data experiments

The AVIRIS Cuprite data set in Fig. 1(a) is now used in real data experiments. Table 3 shows the number of endmembers, p , estimated by HFC and NWHFC methods with different false alarm probabilities, where a reasonable estimate for VD seemed to be 16 or 19 when P_F was set to 10^{-3} or 10^{-4} . The results obtained for different values of the p listed in Table 3 were very similar, in the sense that the pure pixels representing the five pure mineral signatures were extracted and most of the extracted pixels were overlapped. Therefore, only experiments for the case of p equal to 16 are presented in this study for demonstration.

Method	$P_F=10^{-1}$	$P_F=10^{-2}$	$P_F=10^{-3}$	$P_F=10^{-4}$	$P_F=10^{-5}$
HFC	30	19	16	16	15
NWHFC	27	21	19	19	18

Table 3. VD estimated by the HFC and NWHFC methods for the AVIRIS Cuprite image in Fig. 1(a).

Fig. 6(a) shows the 16 target pixels generated and labeled in order by ATGP. Using these 16 ATGP-generated target pixels as initial endmember pixels, the FPPI algorithm extracted 16 pixels shown in Fig. 6(b). For illustrative purposes, Fig. 6(c) shows the 16 endmember pixels extracted by ENVI's PPI algorithm using random initial endmembers. Fig. 7 further shows the graphical spatial positions of the pixels extracted by the three algorithms above, where Fig. 7(a) shows that most of the pixels initially found by ATGP turned out to be final endmembers after running

FPPI. Also, Fig. 7(b) shows that most of the pixels produced by PPI and FPPI were overlapped, specifically, those associated with pure pixels labeled as A, B, C, K and M in Fig. 1(b). As a result, we can conclude that using ATGP-generated pixels as initial endmembers not only can speed up the convergence of the algorithm, but most of them can be also potential endmember pixels as well.

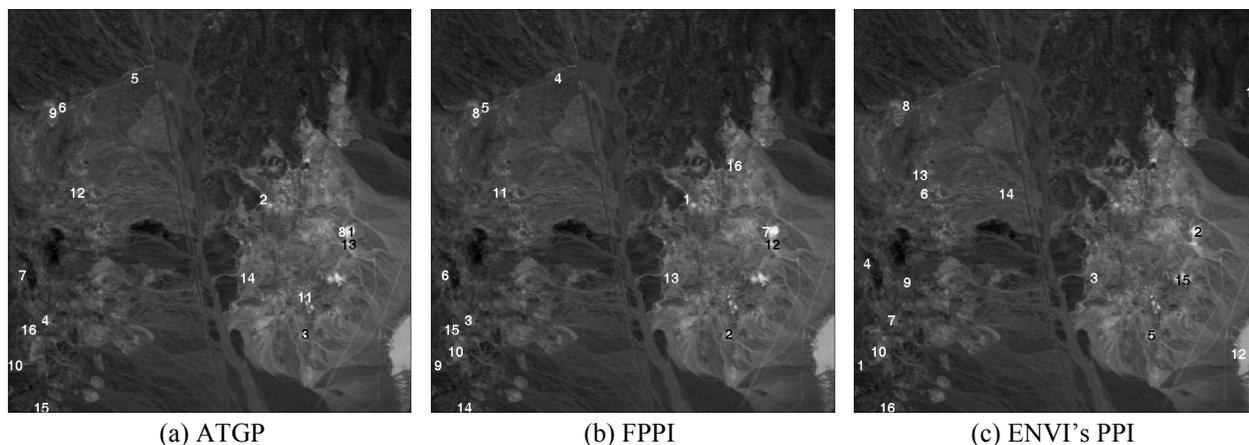


Figure 6. The 16 pixels extracted by (a) ATGP; (b) FPPI algorithm; (c) ENVI's PPI algorithm.

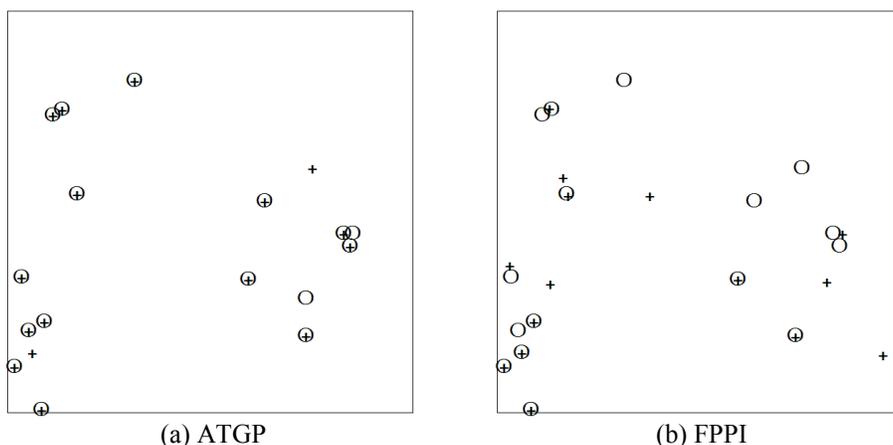


Figure 7. (a) Graphical spatial positions of pixels generated by ATGP, labeled by “o” and pixels extracted by FPPI, labeled by “+”. (b) Graphical spatial positions of pixels generated by FPPI, labeled by “o”, and pixels extracted by ENVI's PPI, labeled by “+”.

Finally, Table 4 tabulates the number of runs or iterations, and computing time for both FPPI and ENVI's PPI, where the numbers in the parentheses in the FPPI column are the numbers of runs or iterations which actually carried out replacements. As we can see from Table 4, the number of iterations were very small for the FPPI. This implied that most of the ATGP-generated target pixels were already pure pixels before the execution of the FPPI. On other hand, ENVI's PPI required thousands of runs to achieve results similar to those found by the FPPI in terms of signature purity. This indicated that using ATGP-generated target pixels as initial endmember pixels not only improves the iterative process for the FPPI, but it also reduces the computational time of the algorithm tremendously (the running times reported on Table 5 were measured in a PC with AMD Athlon 2.6 GHz processor and 512 Mb of RAM). As shown by Table 5, the FPPI algorithm was more than 13 times faster than the ENVI's PPI algorithm in the same computing environment.

FPPI		ENVI's PPI	
Number of iterations	Computation time	Number of runs	Computation time
697 (5)	235	10,000	3068

Table 4. Number of iterations or runs, and computation time in seconds for the FPPI and ENVI's PPI algorithms.

CONCLUSIONS

This paper has described a new fast iterative algorithm for computation of pixel purity index. The algorithm addresses several shortcomings observed in the traditional PPI algorithm, available from Research Systems ENVI software. First, the FPPI is an iterative algorithm while ENVI's PPI is not. Second, ENVI's PPI cannot guarantee that all the extracted endmembers are actually true endmembers since it is initialized with a set of k randomly generated skewers, where the final selection depends on a threshold parameter, t . The FPPI replaces the two parameters, k and t in the PPI with the virtual dimensionality (VD) to eliminate the issue of sensitivity on k and t . Third, the ENVI's PPI relies on users who manually extract final set of endmember pixels. Therefore, different users may produce different sets of endmembers according to their subjectivity. On other hand, the FPPI is completely automatic and does not need any kind of visual inspection or dimensionality reduction as opposed to the original PPI. Finally, a new feature included in the FPPI which is not available in the standard PPI is to include an automatic target generation process (ATGP) to produce a good set of initial endmembers. Such an initial endmember set can reduce the computation time and the number of iterations tremendously, because many of the initial endmembers are also potential final endmembers as demonstrated in experiments. Another major contribution of this paper is the provision of a detailed, step-by-step algorithmic description of the FPPI, which can be very beneficial to users who are interested in implementing this algorithm without soliciting help from any proprietary software.

ACKNOWLEDGEMENT

A. Plaza would like to thank for support received from the Spanish Ministry of Education and Science (PR2003-0360 Fellowship).

REFERENCES

1. R.A. Schowengerdt, *Remote Sensing: Models and Methods for Image Processing*, 2nd ed., Academic Press, 1997.
2. A. Plaza, P. Martínez, R. Pérez and J. Plaza, "A Quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 42, no 2, pp.650-663, March 2004.
3. J.W. Boardman, "Geometric mixture analysis of imaging spectrometry data," *Proc. Intl. Geoscience and Remote Sensing Symp.*, vol. 4, pp. 2369-2371, 1994.
4. J.W. Boardman, F.A. Kruse and R.O. Green, "Mapping target signatures via partial unmixing of AVIRIS data," *Summaries of JPL Airborne Earth Science Workshop*, Pasadena, 1995.
5. F. Chaudhry, S. Chakravarty, A. Plaza and C.-I Chang, "Design of fast algorithms for pixel purity index for endmember extraction in hyperspectral imagery," *Annual Meeting, Proceedings of American Society of Photogrammetry & Remote Sensing*, Baltimore, 2005.
6. C.-I Chang and A. Plaza, "Fast pixel purity index (FPPI): A fast iterative algorithm for implementation of pixel purity index (PPI)," *IEEE Trans. on Geoscience and Remote Sensing* (under review).
7. C.-I Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*, Kluwer Academic/Plenum Publishers, 2003.
8. C.-I Chang and Q. Du, "Estimation of number of spectrally distinct signal sources in hyperspectral imagery," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 42, no. 3, pp. 608-619, March 2004.
9. J.C. Harsanyi, W. Farrand and C.-I Chang, "Detection of subpixel spectral signatures in hyperspectral image sequences," *Annual Meeting, Proceedings of American Society of Photogrammetry & Remote Sensing*, Reno, pp. 236-247, 1994.
10. H. Ren and C.-I Chang, "Automatic spectral target recognition in hyperspectral imagery," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1232-1249, October 2003.
11. A.A. Green, M. Berman, P. Switzer and M.D. Craig, "A transformation for ordering multispectral data in terms of image quality with implications for noise removal," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 26, pp. 65-74, 1988.
12. A. Plaza, P. Martínez, R. Pérez and J. Plaza, "Spatial/spectral endmember extraction by multidimensional morphological operations," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 40, no. 9, pp. 2025-2041, Sept. 2002.