A Fast Iterative Algorithm for Implementation of Pixel Purity Index

Chein-I Chang, *Senior Member, IEEE,* and Antonio Plaza, *Member, IEEE*

***Abstract*—The pixel purity index (PPI) has been widely used in hyperspectral image analysis for endmember extraction due to its publicity and availability in the Environment for Visualizing Images (ENVI) software. Unfortunately, its detailed implementation has never been made available in the literature. This paper investigates the PPI based on limited published results and proposes a fast iterative algorithm to implement the PPI, referred to as fast iterative PPI (FIPPI). It improves the PPI in several aspects. Instead of using randomly generated vectors as initial endmembers, the FIPPI produces an appropriate initial set of endmembers to speed up its process. Additionally, it estimates the number of endmembers required to be generated by a recently developed concept, virtual dimensionality (VD) which is one of the most crucial issues in the implementation of PPI. Furthermore, it is an iterative algorithm, where an iterative rule is developed to improve each of the iterations until it reaches a final set of endmembers. Most importantly, it is an unsupervised algorithm as opposed to the PPI, which requires human intervention to manually select a final set of endmembers. The experiments show that both the FIPPI and the PPI produce very close results, but the FIPPI converges very rapidly with significant savings in computation.**

***Index Terms*—Automatic target generation process (ATGP), endmember extraction algorithm (EEA), endmember pixel, fast iterative pixel purity index (FIPPI), pixel purity index (PPI), virtual dimensionality (VD).**

I. Introduction

According to the definition in [1], an endmember is an idealized pure signature for a class. Finding pure signatures in hyperspectral imagery is considered to be an important and crucial task in hyperspectral data exploitation. Many endmember extraction algorithms (EEAs) have been developed for this purpose [2]. One of the most popular EEAs has been the pixel purity index (PPI), developed by Boardman *et al.* in [3]. It searches for a set of vertices of a convex hull in a given dataset, which are supposed to be pure signatures present in the data. It has been widely used because it is available in the Environment for Visualizing Images (ENVI) software system originally developed by Analytical Imaging and Geophysics (AIG) [4] and has found multiple applications in different areas [5]. Due to its propriety and limited published results, its detailed implementation has never been made public. Therefore, most of the people who use the PPI for endmember extraction either appeal for ENVI software or implement their versions of the PPI based on whatever available in the literature. This paper presents our experience with the PPI and investigates several issues arising in the practical implementation of the PPI. One of major issues is the sensitivity to input parameters, namely k (number of so-called skewers) and t (cutoff threshold value) used in the PPI. Another important issue is the random procedure employed by the PPI to generate skewers, which are then used to compute endmember candidates. A third issue is its computational complexity. A fourth issue is the requirement of human intervention to manually select a final set of endmembers by visual inspection. Most importantly, the PPI is not an iterative process and does not guarantee its convergence in finite runs, despite that it may converge asymptotically as it is claimed.

In order to address these issues, this paper develops a fast iterative algorithm for implementing the PPI, referred to as fast iterative pixel purity index (FIPPI). It has several significant advantages over the PPI. First of all, it makes use of a recently developed concept, virtual dimensionality (VD), in [6] to estimate the number of endmembers p required to be generated. The VD allows us to replace the two parameters k and t used in the PPI so that the algorithm's sensitivity to these parameters can be resolved. Second, the FIPPI takes advantage of the automatic target generation process (ATGP) in [6] and [7] to generate an appropriate set of initial endmembers that can thus speed up the algorithm considerably. Third, the FIPPI is an iterative algorithm that converges very rapidly with tremendous savings in computation time. Most importantly, the PPI requires a visualization tool to manually select a final set of endmembers. Such a problem is avoided by the FIPPI because the FIPPI is automatic and the final set of FIPPI-generated endmembers is always the same, regardless of who is a user of the FIPPI. This is considered to be one of the most significant advantages of the FIPPI over the PPI. In order to evaluate the performance and effectiveness of the FIPPI, the PPI was implemented via the Research Systems ENVI 3.6 software to make sure that the version of our implemented PPI is exactly the same one developed in [3]. Further, a well-known Airborne Visible Infrared Imaging Spectrometer (AVIRIS) dataset collected over the Cuprite mining district in Nevada in 1997 is used for experiments. Since the image scene is available online at website, people who are interested in the proposed FIPPI can reproduce our results and conduct their experiments to exploit various applications. The experimental results show that many of the final endmembers produced by both the FIPPI and the PPI are identical. Those which are not identical are very close in the sense of spectral similarity. Additionally, the FIPPI only requires a few iterations to generate its final

Manuscript received March 16, 2005; revised May 1, 2005. The work of A. Plaza was supported by Spanish Ministry of Education and Science under PR2003-0360 Fellowship.

C.-I Chang is with the Remote Sensing Signal and Image Processing Laboratory, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD 21250 USA; e-mail: cchang@umbc.edu.

A. Plaza is with the Department of Computer Science, University of Extremadura, E-10071 Cáceres, Spain; e-mail: aplaza@unex.es.

Digital Object Identifier 10.1109/LGRS.2005.856701

set of endmembers as compared to the PPI which requires the parameter k to be a very high number, e.g., 10^4 or greater. Most significantly, the FIPPI substantially reduces the computational complexity via an iterative process.

II. Pixel Purity Index

Since the details of the specific steps to implement ENVIs PPI are not available in the literature, the PPI algorithm described below is only based on the limited published results and our own interpretation. Nevertheless, except a step, step 5), which is subject to human manipulations, both our algorithm and the PPI in ENVI 3.6 produce the same results.

1) Initialization: Apply a maximum noise fraction (MNF) transform [8] to reduce the dimensionality of the dataset, and generate a set of k unit vectors called "skewers," $\{\mathbf{skewer}_j\}_{j=1}^{k}$ randomly, where k is a sufficiently large positive integer.

2) PPI Calculation: For each $\mathbf{skewer}_j$, all the data sample vectors are projected onto $\mathbf{skewer}_j$ to find sample vectors at its extreme positions and form an extrema set for this particular $\mathbf{skewer}_j$, denoted by $S_{\text{extrema}}(\mathbf{skewer}_j)$. Despite the fact that a different $\mathbf{skewer}_j$ generates a different extrema set $S_{\text{extrema}}(\mathbf{skewer}_j)$, 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

$$I_S(\mathbf{r}) = \begin{cases} 1, & \text{if } \mathbf{r} \in S \\ 0, & \text{if } \mathbf{r} \notin S \end{cases}$$

and

$$N_{\text{PPI}}(\mathbf{r}) = \sum_j I_{S_{\text{extrema}}(\mathbf{skewer}_j^{(k)})}(\mathbf{r}) \tag{1}$$

where $N_{\text{PPI}}(\mathbf{r})$ is defined as the PPI score of sample vector $\mathbf{r}$.

3) Candidate Selection: Find the PPI scores $N_{\text{PPI}}(\mathbf{r})$ for all the sample vectors defined by (1).

4) Endmember Extraction: Let t be a threshold value set for the PPI score. Extract all the sample vectors with $N_{\text{PPI}}(\mathbf{r}) \geq t$.

The pixels resulting from the algorithm above are usually input to ENVI's "L-dimensional visualization tool" which is a supplementary tool of the PPI algorithm that allows manual selection of a final set of endmembers [4].

As we can see from the above PPI algorithm, it is not an iterative process. Its process is determined by the parameter k and carried out in step 2). It is performed for a given set of k skewers which are randomly generated initially. Once the data samples are projected on the set of k skewers, the process is terminated. Several drawbacks resulting from the above PPI algorithm are observed as follows.

1) Since the PPI algorithm is not an iterative process, it does not guarantee that all the PPI-generated endmembers are actually true endmembers due to the fact that the k skewers are randomly generated. A different set of k skewers may produce a different set of endmembers.
2) When the MNF transform is implemented for dimensionality reduction, there is no provided guideline to help users select how many dimensions need to be retained.
3) The PPI algorithm is very sensitive to noise. So, the noise estimation used in the MNF transform is crucial.
4) No criteria are provided for how to select the parameter k and the threshold t, which determines the number of final endmembers.
5) It requires human intervention to manually select a final set of endmembers.

III. Fast Iterative Pixel Purity Index

One of major drawbacks from which the PPI suffers is its computational complexity. For instance, the algorithm took more than 50 min to project every data sample vector of a 350×350-pixel subset of the 224-band Cuprite AVIRIS image scene into 10^4 skewers in a PC with AMD Athlon 2.6-GHz processor and 512 MB of RAM. In order to reduce computational complexity, most ENVI users preprocess the data by MNF transformation so that the original data dimensionality can be reduced to ease computation. Additionally, the PPI is not iterative and can only guarantee to produce optimal results asymptotically. So, it is recommended that the algorithm be implemented using as many skewers as possible in order to obtain optimal results. Also, according to our experiments the PPI is also very sensitive to the initial values of parameters k and particularly, t. Furthermore, the ENVI's PPI makes use of a built-in algorithm to randomly generate a large set of so-called skewers and the users of the ENVI's PPI do not have choice to select their own initial sets of skewers. Finally, the PPI needs a supervised procedure to manually select a final set of endmembers which largely depends upon human interpretation. This section addresses these issues by developing a fast iterative algorithm for PPI, referred to as fast iterative PPI (FIPPI) which is described in detail as follows.

1) Initialization: Find the VD using the Harsanyi–Farrand–Chang (HFC) method in [6], and let it be p, the number of endmembers required to generate.

2) Dimensionality Reduction: Apply the MNF transform for dimensionality reduction and retain the first p components. Let $\{\mathbf{skewer}_j^{(0)}\}_{j=1}^{p}$ be an initial set of p skewers generated by selecting those pixels that correspond to target pixels generated by ATGP in [7].

3) Iterative Rule: At iteration $k \geq 0$, for each $\mathbf{skewer}_j^{(k)}$ all the sample vectors are projected onto this particular $\mathbf{skewer}_j^{(k)}$ to find those which are at its extreme positions to form an extrema set, denoted by $S_{\text{extrema}}(\mathbf{skewer}_j^{(k)})$. Find the sample vectors that produce the largest $N_{\text{PPI}}(\mathbf{r}_j^{(k)})$ defined by (1), and let them be denoted by $\{\mathbf{r}_j^{(k)}\}$.

4) Stopping Rule: Form the joint set, $\{\mathbf{skewer}_j^{(k+1)}\} = \{\mathbf{r}_j^{(k)}\}_{N_{\text{PPI}}(\mathbf{r}_j^{(k)})>0} \cup \{\mathbf{skewer}_j^{(k)}\}$. If $\{\mathbf{skewer}_j^{(k+1)}\} = \{\mathbf{skewer}_j^{(k)}\}$, then no new endmembers are added to the skewer set. In this case, the algorithm is terminated. Otherwise, let $k \leftarrow k+1$ and go to step 3).

It should be noted that when the algorithm is terminated in step 4), the vectors corresponding to the pixels with $N_{\text{PPI}}(\mathbf{r}_j^{(k+1)}) > 0$ are the desired endmembers, denoted by $\{\bar{\mathbf{e}}_j^{(k+1)}\}$. Additionally, in step 1) of initialization, the ATGP in [7] was used to generate the initial set of p initial skewers, $\{\mathbf{skewer}_j^{(0)}\}_{j=1}^{p}$. The ATGP can be replaced by any method,

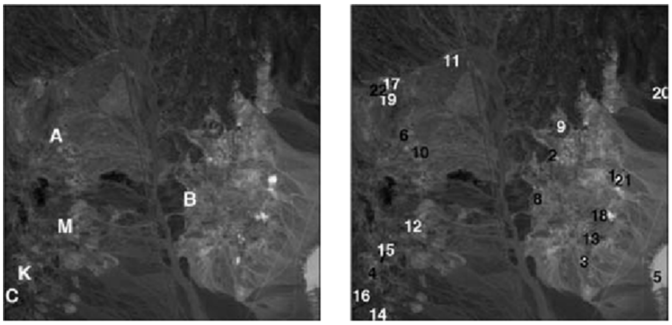

Fig. 1. Spectral band at 827 nm of the Cuprite AVIRIS scene, with (left) spatial positions of various pure pixels labeled by A, B, C, K, M. (Right) The 22 target pixels extracted by ATGP.

including the one used in the PPI that randomly generates so-called “skewers,” $\{\mathbf{skewer}_j^{(0)}\}$ as long as the initialization method provides a good estimate of initial skewers. But, as will be shown in experiments, such randomly generated initial skewers can only slow down the algorithm.

Several major advantages can be obtained from the proposed FIPPI. First, it improves computational efficiency significantly compared to the original PPI since the FIPPI is an iterative algorithm and the PPI is not iterative. Second, the parameters k and t are not required. So, there is no need for users to input these values to avoid human subjectivity. Third, the number of endmembers required to generate, p, can be estimated by VD as opposed to the PPI which must be carried out on a trial-and-error basis. Fourth, the FIPPI implements a replacement rule to iteratively refine each iteration and is terminated when a set of final endmembers $\{\bar{\mathbf{e}}_j\}$ is identified via an implemented stopping rule. Fifth and most importantly, the FIPPI is fully automated, and does not require any human supervision as does the PPI.

IV. Experiments

The image data to be used for experiments is the Cuprite AVIRIS image scene shown in Fig. 1 (left), which was collected over the Cuprite mining district, Nevada, in 1997. The data are available in reflectance units.[1] This 224-band scene is well understood mineralogically, where the ground truth provides the precise spatial locations of pure pixels that correspond to the five minerals, alunite, buddingtonite, calcite, kaolinite, and muscovite labeled by A, B, C, K, and M in Fig. 1 (left). It should be noted that bands 1–3, 105–115, and 150–170 have been removed prior to the analysis due to water absorption and low SNR in those bands. According to our experiments, the parameter k was empirically set to $k = 10^4$ for the PPI. This is because we have found that the PPI algorithm produced essentially the same set of endmembers for $k \geq 10^4$ ($k = 10^5$ and $k = 10^6$ were also tested), while a smaller $k < 10^4$ generally resulted in the loss of important endmembers. The value of t was set to the mean of the values of the $N_{\mathrm{PPI}}(\mathbf{r})$ throughout the dataset (these parameter values are in agreement with those used before in the literature [9]). Finally, the value of p was determined by the HFC method for estimation of virtual dimensionality (VD) where Table I tabulates the various values of the VD produced by different false alarm probabilities $P_{\mathrm{F}} = 10^{-i}$ for $i = 1, 2, 3, 4, 5$.

TABLE I
VD Estimates With Various False Alarm Probabilities

P_{F}	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
p	37	27	23	22	21

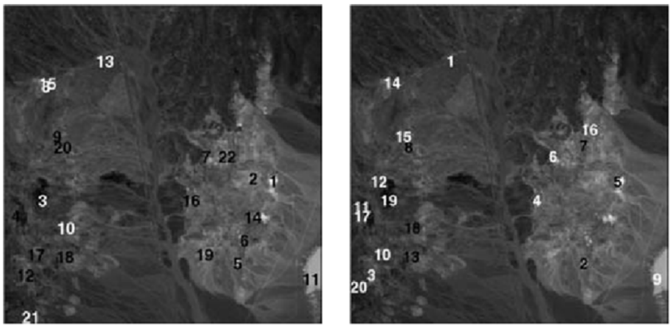

Fig. 2. Endmember pixels extracted by (left) PPI and (right) FIPPI using random initial skewers.

According to our extensive experiments, $p = 22$ seems a reasonable estimate for the Cuprite image scene. Therefore, only experiments based on $p = 22$ will be presented in this paper for illustration and demonstration. Nevertheless, all the arguments made for the case of $p = 22$ can be also applied to other values of p. In order to provide a fair comparison, we avoided generating the final PPI-found endmembers as mean spectra of clusters of extreme pixels in the L-dimensional space. Instead, we selected individual image pixels falling in the corners of the data cloud, and used those pixel signatures as the final endmember set. In this case, an endmember is actually a pixel which represents a pure pixel, referred to as an endmember pixel.

Three experiments were conducted to evaluate the performance of the PPI and FIPPI. Experiment 1 was designed to see how much improvement could be gained for the FIPPI upon the PPI with random initial endmembers. Experiment 2 was designed to see how much improvement would be gained for the PPI if the set of initial endmembers were generated by the ATGP. Experiment 3 compares the results in Experiments 1 and 2.

In order to distinguish the final set of endmember pixels extracted by the PPI and FIPPI, the notations of $\tilde{\mathbf{e}}_j^{(\mathrm{random})}, \tilde{\mathbf{e}}_j^{(\mathrm{ATGP})}$ and $\bar{\mathbf{e}}_j^{(\mathrm{random})}, \bar{\mathbf{e}}_j^{(\mathrm{ATGP})}$ will be used to indicate the final set of endmember pixels extracted by the PPI and the FIPPI, where “bar” and “tilde” will be designated for endmember pixels extracted by the PPI and the FIPPI, respectively, and superscripts “(random)” and “(ATGP)” indicate that the final endmember pixels are generated by using a set of random endmember pixels and the ATGP-generated target pixels as the initial set of endmember pixels.

A. *Experiment 1: Randomly Generated Initial Skewers*

In this experiment, the FIPPI and the PPI were implemented to find endmembers for the image scene in Fig. 1 (left) using the same set of randomly generated initial skewers with $p = 22$. Fig. 2 shows the endmember pixels extracted by the PPI and the FIPPI, respectively. Using the Spectral Angle Mapper (SAM)

[1] http://aviris.jpl.nasa.gov/html/aviris.freedata.html

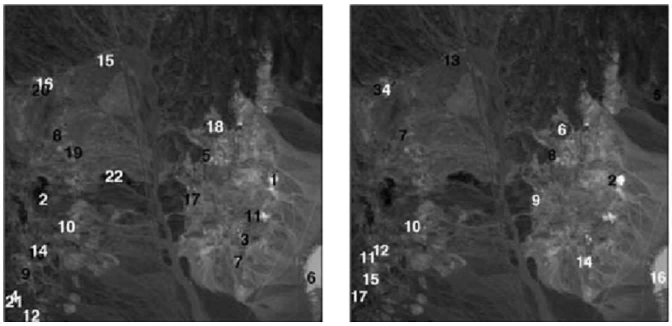

Fig. 3. Endmember pixels extracted by (left) PPI and (right) FIPPI using initial skewers generated by ATGP.

[6] as a spectral measure, we found that among the 22 extracted endmember pixels by PPI, there were many endmember pixels overlapped with the set of 20 endmember pixels produced by FIPPI. Specifically, only five pixels extracted by the FIPPI, i.e., $\bar{\mathbf{e}}_7^{(\text{random})}$, $\bar{\mathbf{e}}_{11}^{(\text{random})}$, $\bar{\mathbf{e}}_{12}^{(\text{random})}$, $\bar{\mathbf{e}}_{16}^{(\text{random})}$, and $\bar{\mathbf{e}}_{20}^{(\text{random})}$, were not included in the set of pixels extracted by PPI. This implied that 15 out of 20 FIPPI-generated pixels were overlapped with the final set of pixels produced by PPI (75% overlap). It should be noted that, when random skewers are used to initialize both the PPI and the FIPPI, there is no guarantee that the results are repeatable between different runs. In this case, we observed that most of the pixels that were not overlapped between the two sets were indeed extracted by the two algorithms in some of the runs.

B. *Experiment 2: Initial Skewers Generated by ATGP*

In this experiment, a comparative study is conducted to see how much improvement can be gained for both the PPI and the FIPPI if their initial sets of endmembers are appropriately selected by the ATGP. Fig. 1 (right) shows the 22 target pixels, $\{\mathbf{t}_j^{(\text{ATGP})}\}_{j=1}^{22}$ generated by the ATGP which are used as the initial set of endmember pixels for both the PPI and FIPPI. Fig. 3 shows the final 22 endmember pixels extracted by the PPI and the 17 endmember pixels extracted by the FIPPI where "$\mathbf{t}$" rather than "$\mathbf{e}$" was used to indicate that the ATGP-generated pixels were target pixels, not necessarily endmember pixels. Once again, the SAM was used to measure similarity between two sets of extracted pixels. It was found that only three out of 22 pixels in the final endmember set $\{\tilde{\mathbf{e}}_j^{(\text{ATGP})}\}_{j=1}^{22}$ were not in the initial set of ATGP-generated pixels, with 86.4% overlap. Similarly, only two out of 20 pixels in the final endmember set $\{\bar{\mathbf{e}}_j^{(\text{ATGP})}\}_{j=1}^{20}$ were not in the initial set of ATGP-generated pixels, with 90% overlap.

C. *Experiment 3: Comparative Study Between Experiments One and Two*

Now we make further comparisons between the final sets of endmember pixels generated by the PPI and the FIPPI using the initial random sets of endmember pixels in Experiment 1 and the initial set of ATGP-generated target pixels in Experiment 2. Since only the spatial locations of these pixels are of major interest, Fig. 4 only shows the graphical positions of the three sets $\{\mathbf{t}_j^{(\text{ATGP})}\}_{j=1}^{22}$ labeled by "×," $\{\tilde{\mathbf{e}}_j^{(\text{random})}\}_{j=1}^{22}$ labeled by "o" and $\{\tilde{\mathbf{e}}_j^{(\text{ATGP})}\}_{j=1}^{22}$ labeled by "+" for PPI (left), and the graphical spatial positions of the three sets $\{\mathbf{t}_j^{(\text{ATGP})}\}_{j=1}^{22}$ labeled by "×," $\{\bar{\mathbf{e}}_j^{(\text{random})}\}_{j=1}^{20}$ labeled by "o" and $\{\bar{\mathbf{e}}_j^{(\text{ATGP})}\}_{j=1}^{17}$ labeled by "+" for FIPPI (right), respectively.

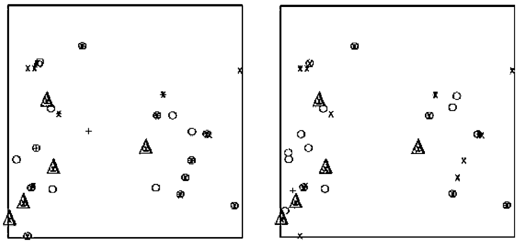

Fig. 4. Graphical spatial locations of ATGP-generated pixels and endmember pixels extracted by (left) PPI and (right) FIPPI.

TABLE II
COMPARATIVE ANALYSIS BETWEEN PPI AND FIPPI WITH DIFFERENT INITIAL ENDMEMBER SETS ON THE AVIRIS CUPRITE SCENE

PPI (random)		PPI (ATGP)		FIPPI (random)		FIPPI (ATGP)	
k	Time	k	Time	Iterations	Time	Iterations	Time
10^4	3052	3146	963	6 (72)	528	4 (32)	217

Using the SAM to measure similarity among all the pixels, it was found that more than 50% of endmember pixels were overlapped among the three sets of pixels, $\{\mathbf{t}_j^{(\text{ATGP})}\}_{j=1}^{22}$, $\{\tilde{\mathbf{e}}_j^{(\text{ATGP})}\}_{j=1}^{22}$, and $\{\bar{\mathbf{e}}_j^{(\mathbf{ATGP})}\}_{j=1}^{17}$. It was particularly true when the ATGP-generated target pixels were used as initial endmember pixels to produce the final set of endmember pixels, 19 out of the 22 pixels were overlapped for the PPI and 18 out of the 20 pixels were overlapped for the FIPPI. According to ground truth in Fig. 1 (left), five signatures associated to {alunite (A), buddingtonite (B), calcite (C), kaolinite (K), muscovite (M)} minerals were identified by triangles in Fig. 4.

It is also interesting to note that when $\{\mathbf{t}_j^{(\text{ATGP})}\}_{j=1}^{22}$ were used as initial endmember pixels, the discrepancy among the set of the ATGP-generated target pixels and two sets of endmember pixels, $\{\bar{\mathbf{e}}_j^{(\text{random})}\}_{j=1}^{20}$ and $\{\bar{\mathbf{e}}_j^{(\text{ATGP})}\}_{j=1}^{17}$ only occurred at the very end of the process of endmember extraction by the FIPPI. A similar observation can be made for the PPI. Most importantly, the computational complexity was significantly reduced if ATGP-generated pixels were used to initialize both the PPI and the FIPPI. This was because less endmember pixel replacements were required. For example, the proposed FIPPI took less than 4 min to converge in the same computing environment described above, and required approximately four additional minutes for the calculation of $\{\mathbf{t}_j^{(\text{ATGP})}\}_{j=1}^{22}$. As a result, we can conclude that ATGP-generated pixels not only can speed up the convergence of the PPI, but most of them can be also potential endmember pixels as well. Table II tabulates the number of runs or iterations, and computing time for both PPI and FIPPI using random sets of initial endmember pixels and the ATGP-generated target pixels as initial endmember pixels, where the numbers in the parentheses under the columns of FIPPI are the total number of skewers generated before the implemented stopping rule was met.

As we can see from Table II, the number of skewers used by the FIPPI was very small in comparison with the number of skewers required by the PPI, in particular, when ATGP-generated pixels were used for initialization. This implied that most of the ATGP-generated target pixels might have already been pure pixels before the execution of the PPI and the FIPPI. Also, from evidence provided by Table II, we can see that using the ATGP-generated target pixels as initial endmember pixels considerably improves the iterative process for the FIPPI, and also reduces the computing time tremendously. In addition, the FIPPI also performed significantly better than the PPI regardless of what initial set of skewers was used. While our main goal in this paper has been to compare the FIPPI with the original PPI algorithm, a comparison of the algorithm to other approaches such as the N-FINDR algorithm is presented in [10].

V. Conclusion

Pixel purity index is a popular method for endmember extraction. This paper presents a new fast iterative algorithm to implement the PPI, called fast iterative pixel purity index which differs from the PPI originally developed in [3] in the following aspects. First of all, FIPPI replaces the two parameters, k and t in the PPI with the virtual dimensionality (VD) developed in [6] to eliminate the sensitivity issue caused by parameters k and t. Most distinctly, the FIPPI is an iterative algorithm which implements an iterative rule to improve each of iterations, and a stopping rule to terminate the algorithm. In addition, the FIPPI takes advantage of an automatic target generation process to produce an appropriate set of initial skewers to reduce a significant number of iterations. These ATGP-generated initial skewers can further reduce a substantial number of replacements executed by the FIPPI. As a result, a tremendous saving in computing time can be accomplished. Despite the fact that the ATGP was used in the FIPPI to produce a set of initial endmembers, the FIPPI does not exclude any algorithm to replace the ATGP as long as it can produce an appropriate set of initial endmembers. Finally, the FIPPI can be implemented automatically with no human intervention as opposed to the PPI, which requires users to perform visual inspection. Such a significant advantage makes the FIPPI more practical than the PPI because different PPI users who manually extract final set of endmember pixels may produce inconsistent results with different sets of endmembers in the end.

Acknowledgment

A. Plaza would like to thank the Spanish Ministry of Education and Science for support.

References

[1] R. A. Schowengerdt, *Remote Sensing: Models and Methods for Image Processing*, 2nd ed. Orlando, FL: Academic, 1997, p. 447.

[2] E. M. Winter and M. E. Winter, "Autonomous hyperspectral end-member determination methods," in *Proc. SPIE EUROPTO Conf.*, vol. 3870, 1999.

[3] J. W. Boardman, F. A. Kruse, and R. O. Green, "Mapping target signatures via partial unmixing of AVIRIS data," in *Summaries of JPL Airborne Earth Science Workshop*, Pasadena, CA, 1995.

[4] Research Systems, *ENVI User's Guide*. Boulder, CO: Research Systems, Inc., 2001.

[5] F. A. Kruse, L. L. Richardson, and V. G. Ambrosia, "Techniques developed for geologic analysis of hyperspectral data applied to near-shore hyperspectral ocean data," in *Proc. ERIM Conf.*, vol. 1, 1997, pp. 233–246.

[6] C.-I Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*. Amsterdam, The Netherlands: Kluwer, 2003.

[7] H. Ren and C.-I Chang, "Automatic spectral target recognition in hyperspectral imagery," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1232–1249, Oct. 2003.

[8] 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. Geosci. Remote Sens.*, vol. 26, no. 1, pp. 65–74, Jan. 1988.

[9] A. Plaza, P. Martínez, R. Pérez, and J. Plaza, "A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 2, pp. 650–663, Mar. 2004.

[10] C.-I Chang and A. Plaza, "On initialization of endmember extraction algorithms," *IEEE Trans. Geosci. Remote Sens.*, 2005, submitted for publication.