Principal Skewness Analysis: Algorithm and Its Application for Multispectral/Hyperspectral Images Indexing

Xiurui Geng, Luyan Ji, and Kang Sun

Abstract—**In this letter, we present a new feature extraction approach based on third-order statistics (coskewness tensor) called principal skewness analysis (PSA). PSA is the natural extension of principal components analysis from second-order statistics to third-order statistics. The result of PSA is equivalent to that of FastICA when skewness is considered as a non-Gaussian index. Similar to FastICA, PSA also applies the fixed-point method to search the skewness extreme directions. However, when calculating the new projected direction in each iteration, PSA only requires a coskewness tensor, whereas FastICA requires all the pixels to be involved. Therefore, PSA has an advantage over FastICA in speed.**

Index Terms—**Coskewness tensor, independent components analysis (ICA), multispectral/hyperspectral data.**

I. Introduction

SINCE a multispectral/hyperspectral image usually consists of tens or even hundreds of bands, the corresponding feature space is also high dimensional. However, the data structure usually lies in a low-dimensional subspace. This is to say that most part of the high-dimensional feature space is actually quite sparse [1]. Therefore, feature extraction is crucial for multispectral/hyperspectral application.

Principal components analysis (PCA) [2] is a very popular linear transformation technique that uses variance as a measure index. Due to the fact that most information of an image is distributed in the subspace spanned by the first few eigenvectors (corresponding to the maximum eigenvalues), PCA transforms the problem of feature extraction into the calculation of the eigenvalues and eigenvectors of an image's covariance matrix. The maximum noise fraction (MNF) [3] is another popular feature extraction algorithm in hyperspectral remote sensing. It takes the image quality into consideration and seeks for the projection that optimizes the ratio of signal power to noise power [3]. In [4], a new feature extraction method based on wavelet transform is proposed and is shown to significantly increase the overall classification accuracy. Recently, a nearest feature line embedding (NFLE) transformation that simultaneously takes the class separability, neighborhood structure preservation, and nearest feature line measurement into account is presented [5]. It is demonstrated that NFLE is effective for dimensionality reduction in terms of classification.

In fact, both PCA and MNF are special cases of the projection pursuit (PP) approach, which was developed by Friedman and Tukey [6], [7]. For decades, the PP technique has received considerable interest in hyperspectral data analysis [8]–[10] because it can be used to explore interesting projections from a high-dimensional data set by a projection index (PI). Different types of PIs have been developed for various applications. Jimenez and Landgrebe [9] used a PI based on Bhattacharyya's distance to reduce the dimensionality of hyperspectral data, and Ifarragaerri and Chang [8] used information divergence as a PI to search interesting projections that deviate from Gaussian distributions. Based on the third and fourth moments of a hyperspectral image, four PIs (i.e., skewness, kurtosis, with the remaining two being mixed indices of skewness and kurtosis) are suggested to capture targets in [10]. The optimization algorithm used in [10] is the evolutionary algorithm (EA). Due to the nature of EA, the computational complexity of the algorithm is generally very high [10]. References [11] and [12] are based on the high-order statistics for anomaly detection in hyperspectral imagery. In fact, [11] and [12] can be regarded as the application of independent component analysis (ICA) in hyperspectral remote sensing. ICA also plays an important role in multispectral data analysis [13]. ICA aims to find a linear representation of non-Gaussian data so that the components are statistically independent or as independent as possible. JADE [14] and FastICA [15] are the two most popular ICA methods. JADE exploits the fourth-order independence only. FastICA is based on the fixed-point iteration scheme and uses skewness or negentropy (or other indices) as a measure of non-Gaussianity. Because of the computational efficiency, FastICA has been widely applied in the areas of feature extraction, mixing analysis, etc. [16]–[19].

Although FastICA has a fast convergence rate, some problems still exist. It has been found that all the pixels are involved in the process of searching for the optimal projection direction. This becomes quite time consuming when the total number of iterations is large. In this letter, we introduce an equivalent algebraic method for FastICA when using skewness as a non-Gaussian index. It is based on the coskewness tensor of an image, and we call it principal skewness analysis (PSA). In PSA, the fixed-point method is applied for the calculation of the optimal projection directions. Different from FastICA, PSA does not need all the pixels to calculate the projection direction each time. Therefore, we can expect PSA to have an obvious advantage over FastICA in computing time.

Manuscript received October 31, 2013; revised December 31, 2013 and February 10, 2014; accepted March 7, 2014. Date of publication April 3, 2014; date of current version May 12, 2014.

The authors are with the Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2014.2311168

II. Method

FastICA [14] is a widely used method for blind source separation, which indeed is also a PP method. It tries to find the orthogonal directions in which the skewness (or negentropy or other non-Gaussian indices) is maximized. Using a fixed-point method to find the desired direction, FastICA has a cubic convergence rate [14]. However, these directions can be also obtained by performing "eigen-analysis" to the third-order statistics. The method is referred as PSA in this letter. Like PCA, PSA also can be considered as a PP method using data skewness as the PI. The main purpose of PSA is to *find the directions in which the skewness is locally maximized*.

In PCA, the directions that maximize the variance can be conveniently computed by performing eigen-analysis to covariance matrix. In the same way, we can also calculate the skewness by introducing third-order statistics, i.e., the *coskewness tensor.*

A. Coskewness Tensor

In order to calculate the coskewness tensor, the image needs to be centralized and whitened first by

$$\begin{aligned}\hat{\mathbf{B}} &= \{\hat{\mathbf{r}}_1\hat{\mathbf{r}}_2,\ldots,\hat{\mathbf{r}}_M\}\\ &= \left\{\mathbf{F}^T(\mathbf{r}_1-\boldsymbol{\mu})\mathbf{F}^T(\mathbf{r}_2-\boldsymbol{\mu}),\ldots,\mathbf{F}^T(\mathbf{r}_M-\boldsymbol{\mu})\right\} \quad (1)\end{aligned}$$

where $\hat{\mathbf{B}}$ is the whitened data set, $\mathbf{F} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^{\mathbf{T}}$ is the whitening matrix (i.e., $\mathbf{E}$ is the eigenvector matrix of the covariance matrix of the observations $\mathbf{r}_i$, and $\mathbf{D} = \mathrm{diag}\{\lambda_1\lambda_2\ldots\lambda_L\}$ is the corresponding eigenvalue matrix (L is the number of bands)), and $\boldsymbol{\mu}$ is the mean vector of the image.

The corresponding *coskewness tensor* of an image, denoted as $\underline{\mathbf{S}}$, is defined as

$$\underline{\mathbf{S}} = \frac{1}{M}\sum_{i=1}^{M}\hat{\mathbf{r}}_i^3 = \frac{1}{M}\sum_{i=1}^{M}\hat{\mathbf{r}}_i\circ\hat{\mathbf{r}}_i\circ\hat{\mathbf{r}}_i \quad (2)$$

where the three-way outer product $\hat{\mathbf{r}}_i^3 = \hat{\mathbf{r}}_i\circ\hat{\mathbf{r}}_i\circ\hat{\mathbf{r}}_i$ is a three-order tensor with L dimensions (where L is the number of the bands in the image).

Equation (2) provides one way to calculate the coskewness tensor, but it is too time consuming in practice. For reasons of computational simplicity, we introduce the following formula:

$$\underline{\mathbf{S}} = \frac{1}{M}ten\left(\hat{\mathbf{B}}\left(\hat{\mathbf{B}}*\hat{\mathbf{B}}\right)^T\right) \quad (3)$$

where $ten()$ is defined as the tensorization operator here. $ten()$ is designed to reform a $l\times l^2$ matrix into a $l\times l\times l$ tensor according to

$$\underline{\mathbf{A}}(:,:,i) = \mathbf{A}\left(:,(i-1)l+1:il\right), i = 1\ldots l \quad (4)$$

where $\mathbf{A}$ is a $l\times l^2$ matrix, and $\underline{\mathbf{A}}$ is a $l\times l\times l$ tensor. In addition, in (3), $\hat{\mathbf{B}}*\hat{\mathbf{B}}$ represents the Khatri–Rao product of $\hat{\mathbf{B}}$ and $\hat{\mathbf{B}}$ with

$$\hat{\mathbf{B}}*\hat{\mathbf{B}} = \{\hat{\mathbf{r}}_1\otimes\hat{\mathbf{r}}_1\hat{\mathbf{r}}_2\otimes\hat{\mathbf{r}}_2\ldots\hat{\mathbf{r}}_M\otimes\hat{\mathbf{r}}_M\} \quad (5)$$

where $\otimes$ represents the Kronecker product.

B. PSA Algorithm

Similar to PCA, the skewness of the image in any direction $\mathbf{u}$ can be calculated by the following formula:

$$g(\mathbf{u}) = \underline{\mathbf{S}}\mathbf{u}^3, \text{with } \|\mathbf{u}\|_2 = 1. \quad (6)$$

According to [20], (6) can be rewritten as

$$g(\mathbf{u}) = \underline{\mathbf{S}}\times_1\mathbf{u}\times_2\mathbf{u}\times_3\mathbf{u} = \sum_{i=1}^{L}\sum_{j=1}^{L}\sum_{k=1}^{L}s_{i,j,k}u_iu_ju_k \quad (7)$$

where $s_{i,j,k}$ is the (i,j,k) element of $\underline{\mathbf{S}}$, and u_i is the ith element of $\mathbf{u}$.

In order to search the direction of the maximum skewness, we can form a Lagrangian function such as PCA

$$G(\mathbf{u}) = \frac{1}{3}\underline{\mathbf{S}}\mathbf{u}^3 + \frac{\lambda}{2}(1-\mathbf{u}^T\mathbf{u}). \quad (8)$$

Let $G'_{\mathbf{u}} = \mathbf{0}$, we can have $G'_{\mathbf{u}} = \underline{\mathbf{S}}\mathbf{u}^2 - \lambda\mathbf{u} = \mathbf{0}$, which results in the following equation:

$$\underline{\mathbf{S}}\mathbf{u}^2 = \lambda\mathbf{u} \quad (9)$$

where $\underline{\mathbf{S}}\mathbf{u}^2 = \underline{\mathbf{S}}\times_1\mathbf{u}\times_3\mathbf{u}$ is a vector. Equation (9) can be rewritten as follows:

$$\underline{\mathbf{S}}\times_1\mathbf{u}\times_3\mathbf{u} = \lambda\mathbf{u}. \quad (10)$$

We can regard λ and $\mathbf{u}$ as the "eigenvalue" and "eigenvector" of $\underline{\mathbf{S}}$ just as those of the matrix. However, unlike the covariance matrix, the problem of calculating the eigenvalues and eigenvectors of the coskewness tensor is not solved yet. Here, the fixed-point scheme is derived as follows:

$$\begin{aligned}\mathbf{u} &= \underline{\mathbf{S}}\times_1\mathbf{u}\times_3\mathbf{u}\\ \mathbf{u} &= \mathbf{u}/\|\mathbf{u}\|_2. \quad (11)\end{aligned}$$

If (11) does have a fixed-point $\mathbf{u} = \mathbf{u}_1$ (i.e., $\mathbf{u}_1$ is a unit vector), $\lambda_1 = \underline{\mathbf{S}}\mathbf{u}_1^3$ is the corresponding skewness in the direction $\mathbf{u}_1$. Here, $\mathbf{u}_1$ is called the first *principal skewness direction*. However, it does not guarantee that λ_1 corresponds to the global maximum skewness.

Another challenging problem of (11) is that the principal skewness directions are not always orthogonal to each other, whereas in the case of PCA, the eigenvector matrix is an orthogonal matrix. To prevent the second *principal skewness direction* from converging to the same maxima as the first one, we project the data into the orthogonal complement space of $\mathbf{u}_1$

$$\hat{\mathbf{B}}_1 = \mathbf{P}_1\hat{\mathbf{B}} = \{\mathbf{P}_1\hat{\mathbf{r}}_1\mathbf{P}_1\hat{\mathbf{r}}_2\ldots\mathbf{P}_1\hat{\mathbf{r}}_M\} \quad (12)$$

where $\mathbf{P}_1 = \mathbf{I} - \mathbf{u}_1\mathbf{u}_1^{\#}$ is the corresponding orthogonal complement projection operator (i.e., $\mathbf{I}$ is the $L\times L$ unit matrix, and $\mathbf{u}_1^{\#} = (\mathbf{u}_1^T\mathbf{u}_1)^{-1}\mathbf{u}_1^T$ is the pseudo-inverse of $\mathbf{u}_1$).

Then, according to (6), the coskewness tensor corresponding to $\hat{\mathbf{B}}_1$, noted as $\underline{\mathbf{S}}_1$, can be expressed as

$$\underline{\mathbf{S}}_1 = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{P}_1\hat{\mathbf{r}}_i)^3 = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{P}_1\hat{\mathbf{r}}_i)\circ(\mathbf{P}_1\hat{\mathbf{r}}_i)\circ(\mathbf{P}_1\hat{\mathbf{r}}_i). \quad (13)$$

Comparing (13) with (2), it seems that directly calculating the coskewness tensor $\underline{\mathbf{S}}_1$ by (13) is very repetitive work. Fortunately, $\underline{\mathbf{S}}_1$ can be conveniently calculated by the following formula:

$$\underline{\mathbf{S}}_1 = \underline{\mathbf{S}} \times_1 \mathbf{P}_1 \times_2 \mathbf{P}_1 \times_3 \mathbf{P}_1 \tag{14}$$

which is the tensor–matrix product between $\underline{\mathbf{S}}$ and $\mathbf{P}_1$. Clearly, it will greatly reduce the computing time of $\underline{S}_1$ and moreover the algorithm. It should be pointed out that (14) is the most important factor that makes PSA faster than FastICA.

After we have the new coskewness tensor $\underline{\mathbf{S}}_1$, the calculation of skewness and the corresponding Lagrangian function can be rewritten as

$$g_1(\mathbf{u}) = \underline{\mathbf{S}}_1\mathbf{u}^3 = \underline{\mathbf{S}}(\mathbf{P}_1\mathbf{u})^3 \tag{15}$$

$$G_1(\mathbf{u}) = \frac{1}{3}\underline{\mathbf{S}}_1\mathbf{u}^3 + \frac{\lambda}{2}(1 - \mathbf{u}^T\mathbf{u}). \tag{16}$$

Let $G_1'(\mathbf{u}) = \mathbf{0}$, we can have

$$\underline{\mathbf{S}} \times_1 (\mathbf{P}_1\mathbf{u}) \times_2 \mathbf{P}_1 \times_3 (\mathbf{P}_1\mathbf{u}) = \lambda\mathbf{u}. \tag{17}$$

Similarly, we can derive the new fixed-point iteration as

$$\begin{aligned} \mathbf{u} &= \underline{\mathbf{S}} \times_1 (\mathbf{P}_1\mathbf{u}) \times_2 \mathbf{P}_1 \times_3 (\mathbf{P}_1\mathbf{u}) \\ &= \mathbf{P}_1 (\underline{\mathbf{S}} \times_1 (\mathbf{P}_1\mathbf{u}) \times_3 (\mathbf{P}_1\mathbf{u})) \\ \mathbf{u} &= \mathbf{u}/\|\mathbf{u}\|_2. \end{aligned} \tag{18}$$

If (18) has a fixed-point $\mathbf{u} = \mathbf{u}_2$ (i.e., $\mathbf{u}_2$ is a unit vector), $\lambda_2 = \underline{\mathbf{S}}(\mathbf{P}_1\mathbf{u}_2)^3 = \underline{\mathbf{S}}_1\mathbf{u}_2^3$ is the corresponding skewness of the projected image $\hat{\mathbf{B}}_1$ in the direction $\mathbf{u}_2$. Here, $\mathbf{u}_2$ is called to be the second *principal skewness direction*.

Then, generate the orthogonal complement projection operator of $[\mathbf{u}_1\mathbf{u}_2]$, $\mathbf{P}_2 = \mathbf{I} - [\mathbf{u}_1\mathbf{u}_2][\mathbf{u}_1\mathbf{u}_2]^{\#}$, and compute the third *principal skewness direction*, etc. The general iterative expression is

$$\begin{aligned} \mathbf{u} &= \underline{\mathbf{S}} \times_1 (\mathbf{P}_k\mathbf{u}) \times_2 \mathbf{P}_k \times_3 (\mathbf{P}_k\mathbf{u}) \\ &= \mathbf{P}_k (\underline{\mathbf{S}} \times_1 (\mathbf{P}_k\mathbf{u}) \times_3 (\mathbf{P}_k\mathbf{u})) \\ \mathbf{u} &= \frac{\mathbf{u}}{\|\mathbf{u}\|_2} \end{aligned} \tag{19}$$

where $\mathbf{P}_k = \mathbf{I} - [\mathbf{u}_1 \ldots \mathbf{u}_k][\mathbf{u}_1 \ldots \mathbf{u}_k]^{\#}$ (i.e., p is the total number of principal skewness directions). Clearly, PSA is equivalent to FastICA (when using a skewness index); hence, the principal skewness directions extracted by PSA are equal to the independent components extracted by FastICA (when using a skewness index).

The pseudocode for the PSA algorithm is described below.

Algorithm 1: PSA

```
1) Input the data set $\mathbf{B}$, the number of the principal skewness
   directions, $p$;
   %% data whitening%%
2) Calculate the mean vector of the image $\mu = \mathrm{mean}(\mathbf{B})$;
3) Let $\mathbf{F}_p = \mathbf{E}_p\mathbf{D}_p^{-1/2}\mathbf{E}_p^T$;
4) Let $\hat{\mathbf{B}} = \mathbf{F}_p^{\mathrm{T}}(\mathbf{B} - \boldsymbol{\mu})$;
   %% Calculate coskewness tensor%%
5) Calculate the coskewness tensor of image, $\underline{\mathbf{S}}$ according
   to (3)
   %% Initialize the projection matrix%%
6) Let $\mathbf{P}_0 = \mathbf{I}_{p\times p}$;
   %% main loop search for projected directions%%
7) For $i = 1$ to $p$
8)     $k = 0$;
9)     Initialize $\mathbf{u}_i^{(k)}$ with random unit vector;
10)    While stop conditions are not met do; inner iteration
11)        $\mathbf{u}_i^{(k+1)} = \underline{\mathbf{S}} \times_1 (\mathbf{P}_{i-1}\mathbf{u}_i^{(k)}) \times_2 \mathbf{P}_{i-1} \times_3 (\mathbf{P}_{i-1}\mathbf{u}_i^{(k)})$;
12)        $\mathbf{u}_i^{(k+1)} = \mathbf{u}_i^{(k+1)}/norm(\mathbf{u}_i^{(k+1)})$;
13)        $k = k + 1$;
14)    End
15)    Let $\mathbf{u}_i = \mathbf{u}_i^{(k)}$ and $\mathbf{U} = [\mathbf{u}_1 \ldots \mathbf{u}_i]$;
16)    Let $\mathbf{P}_i = \mathbf{I} - \mathbf{U}\mathbf{U}^{\#}$;
17) End
   %% output%%
18) $\mathbf{Y} = \mathbf{U}^T\hat{\mathbf{B}}$; //$\mathbf{U}$ is the final principal skewness transfor-
   mation matrix, and $\mathbf{Y}$ is the transformed image.
```

Some illustration is required here for PSA. In Step 3), $\mathbf{F}_p$ is the whitening matrix for the first p principal components. $\mathbf{E}_p$ is the first p columns of the eigenvector matrix, whereas $\mathbf{D}_p$ is the submatrix of the eigenvalue matrix with the first p columns and rows. In Step 4), the data set is centralized and whitened. In Step 10), we use *error tolerance* as the stop criterion in this letter, i.e., the iteration stops when $\|\mathbf{u}_i^{(k+1)} - \mathbf{u}_i^{(k)}\| < tol$ (threshold). For comparison, the threshold is set to $tol = 0.0001$ (the same as that in FastICA [21]). In Step 18), $\mathbf{U} = [\mathbf{u}_1 \ldots \mathbf{u}_p]$ is a $p \times p$ principal skewness transformation matrix. $\mathbf{Y} = \mathbf{U}^T\hat{\mathbf{B}}$ is a $p \times M$ matrix, and the p rows of $\mathbf{Y}$ are called the p principal skewness components of $\mathbf{B}$.

III. Computational Complexity Analysis

The main advantage of PSA over FastICA lies in the fact that each projected direction is determined by the same coskewness tensor, which is calculated only once before the "iteration" process. However, for FastICA, the expectation value [14], which involves the whole image for computation, needs to be repetitiously calculated for each projection direction. The detailed theoretical analysis and simulation experiment are as follows.

In this section, the computational complexity of FastICA (using skewness as the criterion) and PSA are taken into theoretical comparison. Suppose we are extracting p independent components from an L-band and M-pixel hyperspectral image. Since both methods involve data whitening, only the computational complexity of finding projected directions is considered. For FastICA, it takes about $2pM$ float-point operations (flops) when searching one projected direction during each iteration. If converging in k iterations, it takes $2kpM$ flops for finding one projected direction. Therefore, the total flops of FastICA to find p projected directions are $2kp^2M$, as shown in Table I.

To implement PSA, the coskewness tensor must be calculated first, which needs $p^3M/3$ flops. The following process, i.e., "eigen-analysis" of coskewness tensor, which is similar to that of FastICA, takes kp^4 flops. Therefore, the total computational complexity of PSA is $p^3M/3 + kp^4$. Generally, $M \gg kp$; hence, the computational complexity can be reduced to $p^3M/3$. Therefore, PSA is about 6 k/p times faster than

TABLE I
COMPUTATIONAL COMPLEXITY COMPARISON BETWEEN FASTICA AND PSA

Algorithm	FastICA	PSA
Computational complexity	$2kp^2M$	$p^3M/3+kp^4$

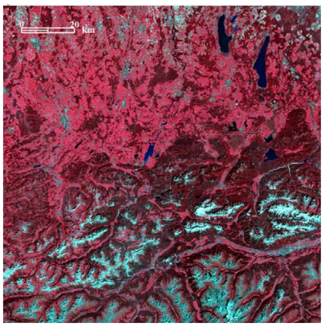

Fig. 1. False color of TM image (Location: center latitude 47° 34′ 12.23″N, center longitude 10° 47′ 57.28″E).

FastICA, where k is the average number of iterations and p is the number of ICs. Since k is usually several times p, PSA is much faster than FastICA in most cases.

In fact, the computational complexity of PSA is mainly dependent on calculation of the coskewness tensor, which accounts for $(1/3)p^3M/((1/3)p^3M + kp^4) = M/(M + 3kp)$ of the total computational time. Let $M = 200 * 200$, $p = 20$, and $k = 80$, then the proportion is 0.8929. Therefore, if the coskewness tensor can be calculated in a parallel manner, the computational performance of PSA can be further improved.

IV. Experiments With Real Data

In this section, real multispectral and hyperspectral data sets are used to evaluate the performance of FastICA and PSA, particularly the capability of computational speed. The code of FastICA is downloaded at [21]. All the tests are conducted on the same PC with Intel Core i5-2400 at 3.10-GHz CPU and 2-GB RAM, and the system is Windows XP Professional Service Pack 3. All the algorithms are programmed in MATLAB 7.0.

A. Experiments Based on TM Image

In this section, both PSA and FastICA (with different non-Gaussian measures) were applied to a real multispectral image to evaluate their performance in terms of computation speed. A 4000×4000 size TM image with six bands is selected for this study (see Fig. 1). The image is derived from a Landsat-5 TM reflectance scene, which is obtained on August 31, 2009. Radiometric processing, including atmospheric correction and topographic correction, is done automatically using the Global Mapper (GM) software package developed by Gong *et al.* [22]. The image is located at the frontier between Austria and Germany.

Since all methods involve whitening data, we focused on the time of finding projected directions in this experiment. The number of ICs is fixed to six since the number of total bands of the data is six. Considering the effect of random initialization for the projected direction, we take the average results of 50 runs for speed comparison. The corresponding computational time is plotted in Fig. 2.

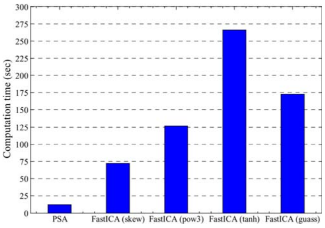

Fig. 2. Average computing time for PSA and FastICAs.

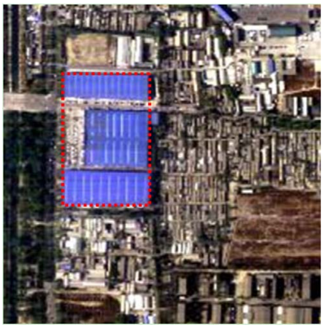

Fig. 3. True color of the image (the area in red rectangle is a blue roof).

It can be found from the results that PSA does have an advantage over FastICA in terms of speed, which is consistent with the theoretical analysis. More specifically, PSA is about six times faster than the fastest version of FastICA (skew). However, PSA is a little slower than theoretical expectation. This is because the calculation of the coskewness tensor involves complicated array structure management, which takes a lot of CPU time but has no float-point operations.

B. Experiment on Hyperspectral Data

The hyperspectral data of 200 × 200 pixels from Operational Modular Imaging Spectrometer-II are used to test the method. The data, which were acquired by the Aerial Photogrammetry and Remote Sensing Bureau in Beijing, China, in 2000, include 64 bands from visible to thermal infrared with about 3-m spatial resolution and 10-nm spectral resolution. Since the signal-to-noise ratio is low in bands 55–60, only the remaining 58 bands are selected for experiment. The true color image is shown in Fig. 3. It is noticeable that there exists a large area of roof painted with blue lacquer (framed with the red rectangle).

The results of PSA are shown in Fig. 4. The distributions of the three different parts of roof are clearly presented in Fig. 4(b), (d), and (i) (see the lighter part). During the field survey, we were told that, although the whole area of the roof was painted with the same blue lacquer, the roof was made from three kinds of tin plate manufactured by different factories [23], which is why the homogenous blue roof in the visible bands exhibits differently in ICs. This is to say that the three parts can be extracted separately as independent components from the image. Furthermore, we can also differentiate vegetation from artificial buildings from Fig. 4(f).

We plot the curves of eigenvalue (from PCA) and skewness (from PSA) in Fig. 5. Unlike eigenvalue, skewness does not strictly follow a descending order. This means that when searching the optimal principal skewness directions, PSA may be trapped in local maxima and the same applied for FastICA.

The time efficiency of PSA and FastICA is also compared here (see Table II). As for FastICA, different non-Gaussian

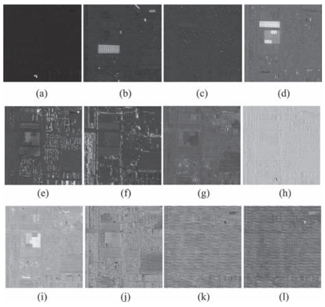

Fig. 4. Results of PSA (12 principal skewness directions).

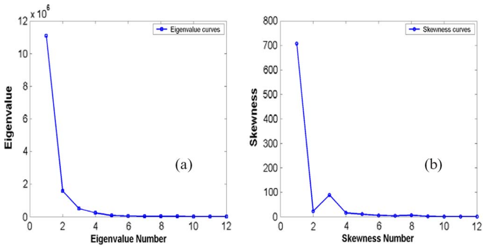

Fig. 5. Plots of (A) eigenvalue and (B) skewness.

TABLE II
AVERAGE COMPUTING TIME OF PSA AND FASTICA

Algorithm	Computing time (Sec)
PSA	0.1226
FastICA (skew)	0.4416
FastICA (pow3)	1.0812
FastICA (tanh)	1.7184
FastICA (gauss)	1.1584

indices are used, including skewness, pow3, tanh, and gauss. Since both PSA and FastICA cannot provide consistent results, the average computing time of 50 runs is used for comparison. It can be seen that PSA is faster than all FastICAs. During the experiment, we also find that when the number of principal skewness directions is increased to 16, both FastICA (using skewness index) and PSA fail to converge. In this case, PSA is much faster than FastICA since the computing time of FastICA is very sensitive to iterations.

V. CONCLUSION

In this letter, we have presented a new algebraic method for ICA, named the PSA, which can be regarded as an extension of PCA. PSA adopts the skewness as the projected index and transfers the ICA problem into the problem of calculating the eigenvalues and eigenvectors of the image's coskewness tensor. Unlike FastICA, PSA only needs the coskewness tensor instead of all pixels to compute the principal skewness directions. Therefore, PSA has a great advantage in computing efficiency. In addition, the calculation of coskewness tensor accounts for most of the computing time of PSA; hence, we will be able to focus on how to estimate the coskewness tensor quickly and accurately in the future.

ACKNOWLEDGMENT

The authors would like to thank Prof. A. P. Cracknell from the University of Dundee, Dundee, U.K., for his great help in language.

REFERENCES

[1] L. Jimenez and D. A. Landgrebe, "Supervised classification in high-dimensional space: Geometrical, statistical, and asymptotical properties of multivariate data," *IEEE Trans. Syst., Man, Cybern. C*, vol. 28, no. 1, pp. 39–54, Feb. 1998.

[2] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Educ. Psych.*, vol. 24, no. 6, pp. 417–441, Sep. 1933.

[3] 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.

[4] L. M. Bruce, C. H. Koger, and L. Jiang, "Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 10, pp. 2331–2338, Oct. 2002.

[5] Y.-L. Chang, J.-N. Liu, C.-C. Han, and Y.-N. Chen, "Hyperspectral image classification using nearest feature line embedding approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 278–287, Jan. 2014.

[6] J. H. Friedman and J. W. Tukey, "A projection pursuit algorithm for exploratory data analysis," *IEEE Trans. Comput.*, vol. C-23, no. 9, pp. 881–889, Sep. 1974.

[7] J. H. Friedman, "Exploratory projection pursuit," *J. Amer. Statist. Assoc.*, vol. 82, no. 397, pp. 249–266, Mar. 1987.

[8] A. Ifarragaerri and C. I. Chang, "Multispectral and hyperspectral image analysis with projection pursuit," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, pp. 2529–2538, 2000.

[9] L. Jimenez and D. A. Landgrebe, "Hyperspectral data analysis and supervised feature reduction via projection pursuit," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 6, pp. 2653–2667, Nov. 1999.

[10] S.-S. Chiang and C. I. Chang, "Unsupervised target detection in hyperspectral images using projection pursuit," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 7, pp. 1380–1391, Jul. 2001.

[11] L. Xun and Y. Fang, "Anomaly detection based on high-order statistics in hyperspectral imagery," in *Proc. 6th World Congr. Intell. Control Autom.*, 2006, pp. 10 416–10 419.

[12] H. Ren and Y.-L. Chang, "A parallel approach for initialization of high-order statistics anomaly detection in hyperspectral imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2008, pp. II-1017–II-1020.

[13] S. Marchesi and L. Bruzzone, "ICA and kernel ICA for change detection in multispectral remote sensing images," in *Proc. IEEE IGARSS*, 2009, pp. II-980–II-983.

[14] J. F. Cardoso and A. Souloumiac, "Blind beamforming for non Gaussian signals," *IEE Proc.-F*, vol. 140, no. 6, pp. 362–370, Dec. 1993.

[15] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Trans. Neural Netw.*, vol. 10, no. 3, pp. 626–634, May 1999.

[16] J. Wang and I. C. Chein, "Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 6, pp. 1586–1600, Jun. 2006.

[17] M. R. Stites, "Assessing and enabling independent component analysis as a hyperspectral unmixing approach," Ph.D. dissertation, Utah State Univ., Logan, UT, USA, 2012.

[18] J. M. P. Nascimento and J. M. Bioucas-Dias, "Does independent component analysis play a role in unmixing hyperspectral data?" *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 1, pp. 175–187, Jan. 2005.

[19] S. Chu, H. Ren, and I. C. Chein, "High-order statistics-based approaches to endmember extraction for hyperspectral imagery," in *Proc. SPIE*, 2008, vol. 6966, p. 69661F.

[20] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.

[21] [Online]. Available: http://research.ics.aalto.fi/ica/fastica/

[22] P. Gong, J. Wang, L. Yu, Y. Zhao, Y. Zhao, L. Liang, Z. Niu, X. Huang, H. Fu, S. Liu, C. Li, X. Li, W. Fu, C. Liu, Y. Xu, X. Wang, Q. Cheng, L. Hu, W. Yao, H. Zhang, P. Zhu, Z. Zhao, H. Zhang, Y. Zheng, L. Ji, Y. Zhang, H. Chen, A. Yan, J. Guo, L. Yu, L. Wang, X. Liu, T. Shi, M. Zhu, Y. Chen, G. Yang, P. Tang, B. Xu, C. Giri, N. Clinton, Z. Zhu, J. Chen, and J. Chen, "Finer resolution observation and monitoring of global land cover: First mapping results with Landsat TM and ETM + data," *Int. J. Remote Sens.*, vol. 34, no. 7, pp. 2607–2654, 2013.

[23] Y. Zhao, "Spectral feature analysis and information extraction models for some typical terrestrial objects in hyperspectral remote sensing—A report for some key questions," Chinese Academy of Sciences (CAS), Beijing, China, 2001, Report of postdoc.