

Structured covariance principal component analysis for real-time onsite feature extraction and dimensionality reduction in hyperspectral imaging

Jaime Zabalza,¹ Jinchang Ren,^{1,*} Jie Ren,² Zhe Liu,³ and Stephen Marshall¹

¹Centre for Excellence in Signal and Image Processing, Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, UK

²College of Electronics and Information, Xi'an Polytechnic University, Xi'an, China

³Department of Applied Mathematics, Northwestern Polytechnical University, Xi'an, China

*Corresponding author: jinchang.ren@strath.ac.uk

Received 13 March 2014; revised 27 May 2014; accepted 28 May 2014;
posted 3 June 2014 (Doc. ID 208226); published 4 July 2014

Presented in a three-dimensional structure called a hypercube, hyperspectral imaging suffers from a large volume of data and high computational cost for data analysis. To overcome such drawbacks, principal component analysis (PCA) has been widely applied for feature extraction and dimensionality reduction. However, a severe bottleneck is how to compute the PCA covariance matrix efficiently and avoid computational difficulties, especially when the spatial dimension of the hypercube is large. In this paper, structured covariance PCA (SC-PCA) is proposed for fast computation of the covariance matrix. In line with how spectral data is acquired in either the push-broom or tunable filter method, different implementation schemes of SC-PCA are presented. As the proposed SC-PCA can determine the covariance matrix from partial covariance matrices in parallel even without prior deduction of the mean vector, it facilitates real-time data analysis while the hypercube is acquired. This has significantly reduced the scale of required memory and also allows efficient onsite feature extraction and data reduction to benefit subsequent tasks in coding and compression, transmission, and analytics of hyperspectral data. © 2014 Optical Society of America

OCIS codes: (100.4145) Motion, hyperspectral image processing; (100.2960) Image analysis.
<http://dx.doi.org/10.1364/AO.53.004440>

1. Introduction

Hyperspectral imaging (HSI), through capturing data from numerous and contiguous spectral bands, has provided a unique and invaluable solution for a number of application areas. Due to the large spectral range covered including not only visible light, like normal imaging devices, but also (near) infrared and even ultraviolet, HSI is able to detect and identify the minute differences of objects and even their changes in terms of temperature and moisture. Consequently,

HSI is able to address traditional applications in remote sensing, mining, agriculture, geology, and military surveillance as well as many newly emerging lab-based data analyses. These new applications can be easily found in security, food quality analysis, and medical and pharmaceutical fields as well as counterfeit goods and documents detection [1–6].

In HSI, as shown in Fig. 1, the captured data forms a three-dimensional (3D) structure, namely a hypercube, including a 2D spatial measurement and a spectral dimension. As a result, the total data contained can be indexed as *HWB*, where each of the symbols refers to the height, width, and bands of the hypercube, respectively. With the narrow band in

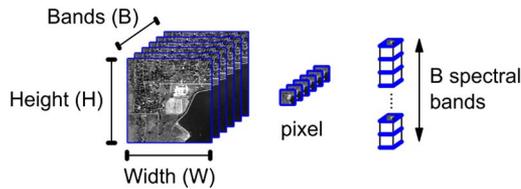


Fig. 1. 3D hypercube in HSI.

nanometers for spectral sampling, HSI enables high discrimination ability in data analysis at the cost of extremely large data sets and computational complexity.

Processing and analysis of a hypercube can be a hard task. When the number of spectral bands exceeds 100, data processing on the hypercube appears like video analytics, a really time-consuming and memory-intensive problem. Moreover, large dimensional data results in the curse-of-dimensionality issue, also known as the Hughes effect [7]. As a result, it is crucial to effectively reduce the data while maintaining the analysis performance in this context. Considering the high spectral resolution used, considerable redundancy exists between neighboring spectral bands, which enables the potential for data reduction.

For dimension reduction in HSI, principal component analysis (PCA) [8] has been widely applied [9,10], becoming a reference technique in such a context. However, for the 3D hypercube in Fig. 1, conventional PCA analysis needs to convert the hypercube into a $B \times HW$ matrix to determine the relevant covariance matrix. As the second dimension of the matrix becomes extreme large, usually over 100 kilobytes, it introduces a fundamental difficulty in calculating the covariance matrix. As a result, the implementation often crashes due to memory and computing problems.

In most cases, only a 2D data slice can be captured as a subspace, in a sequential manner, to form a 3D data cube. The method of data slicing and subspace partition is actually determined by the process of data acquisition, where the time gap between two sequential acquisitions can be potentially used for efficient data analysis. Taking into account the HSI acquisition technologies, the covariance matrix can be more efficiently computed thanks to nature of the hypercube. In this paper, four novel implementation schemes are proposed to calculate the overall covariance matrix by accumulating a group of partial covariance matrices, which can be obtained by much smaller matrices in a real-time manner simultaneous to the acquisition process. Theoretical analysis and experimental results have fully validated the effectiveness of the proposed approaches, where the prior mean adjustment of data is not required to facilitate the real-time implementation.

The remaining part of this paper is organized as follows. Section 2 introduces some background of the acquisition techniques used in HSI. In Section 3, structured covariance PCA (SC-PCA) for efficient

computation of the covariance matrix in PCA is proposed. Experiments and results are presented and discussed in Section 4. Finally, some concluding remarks are summarized in Section 5.

2. Data Acquisition and Conventional PCA in HSI

The 3D hypercube is acquired by means of two most representative HSI acquisition technologies, namely sequential scanning and a tunable filter. Once the hypercube is acquired, conventional PCA can be applied for dimensionality reduction. Relevant techniques are briefed as follows.

A. Data Acquisition Techniques

There are several methods and proposals for HSI acquisition. They can be generally classified as sequential, simultaneous, and pseudo-simultaneous [11,12] depending on the way the acquisition is carried out. Typical methods for acquiring a hypercube are sequential and can be divided into two main groups: scanning and filter-based methods, the optic schemes of which are shown in Fig. 2.

On one hand, scanning techniques are those where partial scenes in the spatial domain along with their corresponding spectra are obtained. In this case, the spatially discrete acquisition is repeated several times until the whole hypercube is generated. In that sense, the scanning methods include pixel scanning, where only one pixel is acquired in every step, and line scanning, also known as push-broom, where several pixels forming a row or a column in the spatial scene are acquired simultaneously per step.

On the other hand, filter-based methods are able to collect the entire spatial domain at a step but only for a given wavelength of the total spectrum. Sequential steps complete the hypercube by adding the spatial scenes collected for every wavelength. Two different versions are available in this kind of acquisition, depending on whether the filter used to collect a specific wavelength is passive or dynamic. While a passive filter selects particular wavelengths, the dynamic or tunable filter is ready for a definite range and is much preferred.

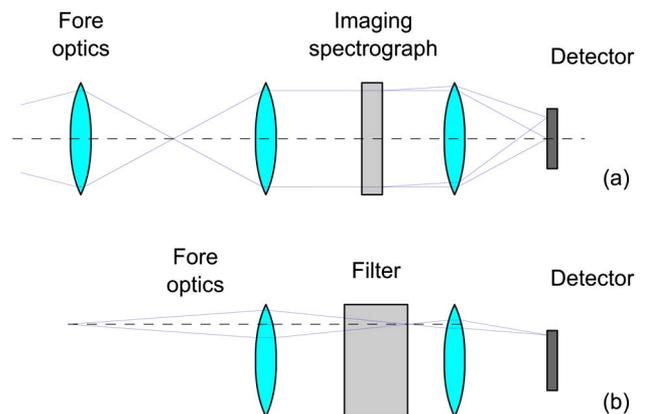


Fig. 2. Optic scheme for (a) scanning methods and (b) tunable filter.

Therefore, Fig. 3 shows the different acquisition methods in HSI, where the collection is sequential and made in terms of pixel, row, column, or band steps. In following sections, we introduce approaches that take advantage of the hypercube structure and its acquisition for efficiently constructing the covariance matrix required in feature extraction and data reduction techniques such as PCA.

B. Conventional PCA in HSI

PCA, also known as the Karhunen–Loeve transform, has been widely used in dimensionality reduction and feature extraction in HSI. Based on the covariance matrix gained from the input data, a large number of correlated features are transformed into a smaller set of uncorrelated ones, using orthogonal projection and truncation in PCA. Accordingly, efficient computation of the covariance matrix in HSI is one key component for dimensionality reduction and data analysis.

In conventional PCA, the hypercube is represented in a matrix form, an image \mathbf{X} called the data matrix (see Fig. 4), where the columns correspond to the samples ($H \times W$ pixels) and the rows contain the different features (B spectral bands) from each sample. PCA computation consists of three stages, where the covariance matrix computation comes in first place and is the stage on which the present work is focused.

Let us denote $\mathbf{x}_n = [x_{n1}, x_{n2}, \dots, x_{nB}]^T$ as the spectral vector of a pixel in the hypercube, where $n \in [1, HW]$ and B is the number of spectral bands. Then, the mean spectral vector of all these pixels can be obtained as

$$\bar{\mathbf{x}} = \frac{1}{HW} \sum_{n=1}^{HW} \mathbf{x}_n. \quad (1)$$

Accordingly, all the spectral vectors can be mean-adjusted by subtracting from this mean spectral vector as

$$\mathbf{p}_n = \mathbf{x}_n - \bar{\mathbf{x}} \quad n \in [1, HW], \quad (2)$$

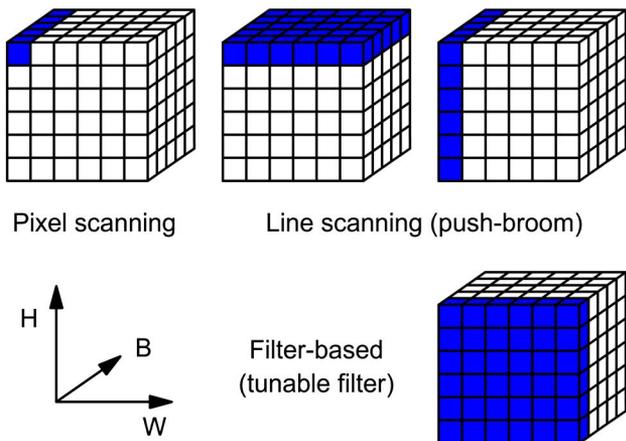


Fig. 3. Different techniques in acquiring a 3D hypercube.

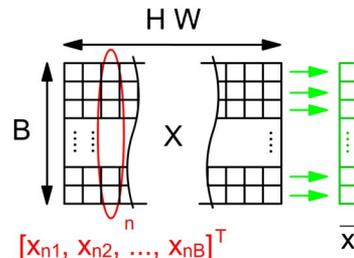


Fig. 4. Data matrix in conventional PCA.

where p_n is the obtained mean-adjusted spectral vector (pixel) used for the covariance matrix calculation.

Let $\mathbf{P} = [\mathbf{p}_1 \ \mathbf{p}_2 \ \dots \ \mathbf{p}_{HW}]$ be a matrix formed by the mean-adjusted spectral vectors, the covariance matrix of \mathbf{X} , \mathbf{C} , can be further obtained as

$$\mathbf{C} = E\{(\mathbf{x}_n - E\{\mathbf{x}_n\})(\mathbf{x}_n - E\{\mathbf{x}_n\})^T\} = E\{\mathbf{p}_n \mathbf{p}_n^T\} = \mathbf{P} \mathbf{P}^T, \quad (3)$$

where $\mathbf{P} \in \mathcal{R}^{B \times HW}$ and $\mathbf{C} \in \mathcal{R}^{B \times B}$, with the dividing term omitted for simplicity.

Conventional PCA suffers two main drawbacks when applied to HSI. First, it requires the completed hypercube to obtain the mean vector and mean-adjusted data matrix \mathbf{P} thus it cannot be implemented in real time while the data is acquired. Second, due to the extremely large dimension of the matrix \mathbf{P} , the calculation of the covariance matrix usually causes memory and computation problems. To solve such drawbacks, several SC-PCA approaches are proposed in the next section.

3. Proposed SC-PCA

For efficient computation of the covariance matrix, four SC-PCA schemes are proposed. By taking advantage of the 3D hypercube structure and the HSI acquisition process, sliced data partitions are sequentially acquired and accessed to compute partial covariance matrices even in real time. Since the matrix sizes by which we obtain the partial covariance matrices are much smaller, this has significantly reduced the overall memory requirement in determining the covariance matrix.

In the real-time case, when the partial covariance matrices are separately computed without adjusting by the mean vector, a mean correction stage is required, where the way the correction matrix can also be determined in real-time is derived as well. In addition, the proposed approaches can be easily implemented in parallel with multiple threads for further improved efficiency.

In Fig. 5, conventional PCA is compared with our proposed strategies for SC-PCA implementation. In total, four different schemes are used in SC-PCA, including pixel-based, row-based, column-based, and band-based approaches. For clarity, symbols, notations, and main equations used in the conventional PCA and our proposed SC-PCA are summarized in Table 1 for comparisons. Detailed discussions and

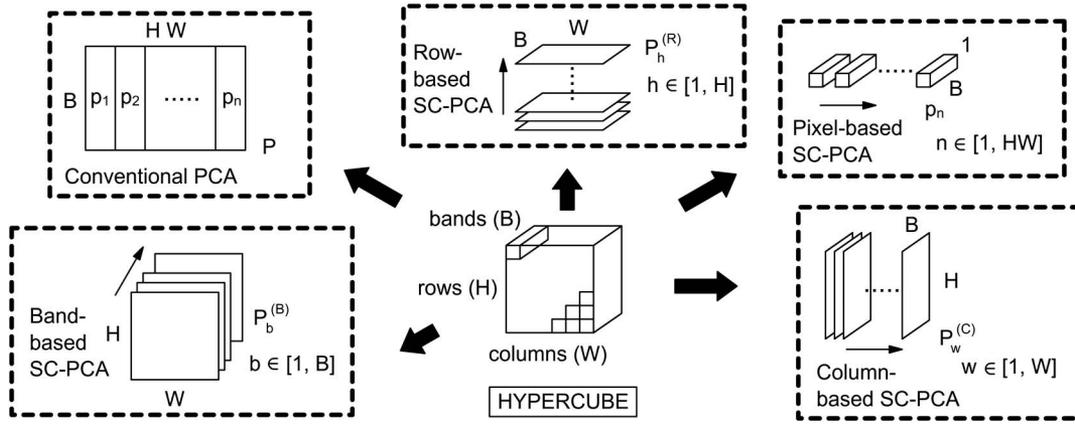


Fig. 5. Comparison between the conventional PCA and the proposed approaches with SC-PCA from the HSI hypercube in determining the covariance matrix.

comparisons of these approaches are presented as follows.

A. Pixel-Based SC-PCA Scheme

In the pixel-based scheme, a single spectral pixel \mathbf{p}_n coming from the pixel scanning method is employed to obtain the partial covariance matrix, where $n \in [1, HW]$ and $\mathbf{p}_n \in \mathfrak{R}^{B \times 1}$,

$$\mathbf{C}^{(\text{pixel})} = \sum_{n=1}^{HW} \mathbf{C}_n^{(\text{pixel})}, \quad \mathbf{C}_n^{(\text{pixel})} = \mathbf{p}_n \mathbf{p}_n^T. \quad (4)$$

As can be seen, the covariance matrix $\mathbf{C}^{(\text{pixel})}$ is obtained by accumulating a group of partial covariance matrices $\mathbf{C}_n^{(\text{pixel})}$, where the matrix to compute the partial covariance is actually the vector \mathbf{p}_n whose dimension is $B \times 1$ in comparison to $B \times HW$ in conventional PCA. However, the number of partial covariance matrices has increased to HW .

B. Row-Based SC-PCA Scheme

In the row-based scheme, 2D planes are defined along the row direction of the hypercube, as a particular case of push-broom or line scanning. Let $\mathbf{P}_h^{(R)} \in \mathfrak{R}^{B \times W}$ (partition by row) denote one of these 2D planes, where $h \in [1, H]$. In fact, each 2D image $\mathbf{P}_h^{(R)}$ is extracted as

$$\mathbf{P}_h^{(R)} = [\mathbf{p}_h \quad \mathbf{p}_{h+H} \quad \cdots \quad \mathbf{p}_{h+(W-1)H}]_{B \times W}. \quad (5)$$

Consequently, the covariance matrix $\mathbf{C}^{(\text{Row})}$ can be derived as the summation of H partial covariance matrices obtained from $\mathbf{P}_h^{(R)}$ as follows:

$$\mathbf{C}^{(\text{Row})} = \sum_{h=1}^H \mathbf{C}_h^{(\text{Row})}, \quad \mathbf{C}_h^{(\text{Row})} = \mathbf{P}_h^{(R)} [\mathbf{P}_h^{(R)}]^T, \quad (6)$$

where the covariance matrix $\mathbf{C}^{(\text{Row})} \in \mathfrak{R}^{B \times B}$ is obtained by accumulating a group of partial covariance matrices $\mathbf{C}_h^{(\text{Row})}$. Note that the partial covariance is defined on $\mathbf{P}_h^{(R)}$, which has a much reduced dimension of $B \times W$ rather than $B \times HW$ in conventional PCA.

C. Column-Based SC-PCA Scheme

In the column-based scheme, 2D planes are defined along the column direction of the hypercube, as the alternative to the row-based approach in the push-broom scanning method. Let $\mathbf{P}_w^{(C)} \in \mathfrak{R}^{B \times H}$ (partition by column) denote one of these 2D planes, where $w \in [1, W]$. The 2D image $\mathbf{P}_w^{(C)}$ is extracted as

$$\mathbf{P}_w^{(C)} = [\mathbf{p}_{1+H(w-1)} \quad \mathbf{p}_{2+H(w-1)} \quad \cdots \quad \mathbf{p}_{H+H(w-1)}]_{B \times H}. \quad (7)$$

Accordingly, the covariance matrix $\mathbf{C}^{(\text{Col})}$ can be obtained via summation of W partial covariance matrices obtained from $\mathbf{P}_w^{(C)}$ as follows:

$$\mathbf{C}^{(\text{Col})} = \sum_{w=1}^W \mathbf{C}_w^{(\text{Col})}, \quad \mathbf{C}_w^{(\text{Col})} = \mathbf{P}_w^{(C)} [\mathbf{P}_w^{(C)}]^T, \quad (8)$$

Table 1. Symbols, Notations and Major Equations Used in Conventional PCA and the Proposed SC-PCA

Method	Data Matrix/Vector Used	Covariance Matrix (Partial)	Covariance Matrix (Full)
Conventional PCA	$\mathbf{P} = [\mathbf{p}_1 \quad \mathbf{p}_2 \quad \cdots \quad \mathbf{p}_{HW}] \in \mathfrak{R}^{B \times HW}$	$\mathbf{C} = \mathbf{P}\mathbf{P}^T \in \mathfrak{R}^{B \times B}$	
Pixel-based SC-PCA	$\mathbf{p}_n \in \mathfrak{R}^{B \times 1}$	$\mathbf{C}_n^{(\text{pixel})} = \mathbf{p}_n \mathbf{p}_n^T, \quad n \in [1, HW]$	$\mathbf{C}^{(\text{pixel})} = \sum_{n=1}^{HW} \mathbf{C}_n^{(\text{pixel})}$
Row-based SC-PCA	$\mathbf{P}_h^{(R)} \in \mathfrak{R}^{B \times W}$	$\mathbf{C}_h^{(\text{Row})} = \mathbf{P}_h^{(R)} [\mathbf{P}_h^{(R)}]^T, \quad h \in [1, H]$	$\mathbf{C}^{(\text{Row})} = \sum_{h=1}^H \mathbf{C}_h^{(\text{Row})}$
Column-based SC-PCA	$\mathbf{P}_w^{(C)} \in \mathfrak{R}^{B \times H}$	$\mathbf{C}_w^{(\text{Col})} = \mathbf{P}_w^{(C)} [\mathbf{P}_w^{(C)}]^T, \quad w \in [1, W]$	$\mathbf{C}^{(\text{Col})} = \sum_{w=1}^W \mathbf{C}_w^{(\text{Col})}$
Band-based SC-PCA	$\mathbf{P}_b^{(B)} \in \mathfrak{R}^{H \times W}$	$\mathbf{C}^{(\text{Band})}(i, j) = \text{vec}(\mathbf{P}_{b=i}^{(B)}) [\text{vec}(\mathbf{P}_{b=j}^{(B)})]^T, \quad i, j \in [1, B]$	

where the partial covariance matrix is sized of $B \times B$. Again the covariance matrix is obtained by accumulating a group of partial covariance matrices $\mathbf{C}_w^{(Col)}$, where the matrix to compute the partial covariance has a much reduced dimension of $B \times H$ rather than $B \times HW$ in conventional PCA.

D. Band-Based SC-PCA Scheme

Different from the abovementioned three SC-PCA approaches, the band-based SC-PCA scheme is derived from a tunable-filter-based data acquisition process, where all the spatial information is captured for a selected wavelength by tuning the optic part of the system. Let us define 2D planes along the band direction of a hypercube as $\mathbf{P}_b^{(B)} \in \mathfrak{R}^{H \times W}$ (partition by spectral band), where $b \in [1, B]$. The 2D image $\mathbf{P}_b^{(B)}$ can be represented as

$$\mathbf{P}_b^{(B)} = \begin{bmatrix} \mathbf{p}_1(b) & \cdots & \mathbf{p}_{H(W-1)+1}(b) \\ \vdots & \ddots & \vdots \\ \mathbf{p}_H(b) & \cdots & \mathbf{p}_{HW}(b) \end{bmatrix}. \quad (9)$$

Note that the selected data partition $\mathbf{P}_b^{(B)}$ contains all pixels in only one specified spectral band rather than the whole spectrum. This fundamental difference has led to the computation of partial covariance matrices, an impossible task as implemented in other SC-PCA approaches. As a result, element-based covariance matrix computation is employed as explained below.

For an element in position (i, j) of the final covariance matrix, we have that

$$\mathbf{C}^{(Band)}(i, j) = \text{vec}(\mathbf{P}_{b=i}^{(B)})[\text{vec}(\mathbf{P}_{b=j}^{(B)})]^T, \quad (10)$$

where $\text{vec}()$ transforms the 2D plane $\mathbf{P}_b^{(B)}$ into a vector in $\mathfrak{R}^{1 \times HW}$, resulting in a scalar value from the multiplication in Eq. (10). The overall covariance matrix is obtained by progressive inclusion of elements (i, j) derived from bands i and j , respectively, where the vectors to compute these elements have a much reduced dimension of $1 \times HW$ rather than $B \times HW$ in conventional PCA.

E. Mean Correction for Real-Time Computation

Note that the proposed schemes can be directly applied on the whole hypercube when the data acquisition process is completed, which results in a much reduced memory requirement in determining the covariance matrix. However, the main idea here is to apply these approaches during the acquisition stage for fast computation of the covariance matrix even without parallel implementation, as suggested in [13,14].

For simultaneous data acquisition and PCA processing, sequentially obtained partitions of data are not mean-adjusted. As a result, instead of using the partitions \mathbf{p}_n , $\mathbf{P}_h^{(R)}$, and $\mathbf{P}_w^{(C)}$ for pixel-, row-, and

column-based schemes, respectively, only the non-mean-adjusted equivalent partitions, namely \mathbf{x}_n , $\mathbf{X}_h^{(R)}$, and $\mathbf{X}_w^{(C)}$, can be used in these three real-time implementation schemes. Note that in the band-based SC-PCA scheme the initial mean adjustment in Eq. (2) is feasible as all pixel values from the same wavelength are available when sequentially collected band by band. As a result, the mean correction is not needed for band-based SC-PCA.

Taking the pixel-based approach for example, sequentially obtained pixels \mathbf{x}_n are not mean-adjusted as in the conventional PCA (1,2). Hence a correction factor \mathbf{M} from the mean spectral vector $\bar{\mathbf{x}}$ in Eq. (1) must be applied:

$$\mathbf{p}_n \mathbf{p}_n^T = \mathbf{x}_n \mathbf{x}_n^T + \mathbf{M}_n^{(\text{pixel})} \mathbf{M}_n^{(\text{pixel})} = \bar{\mathbf{x}} \bar{\mathbf{x}}^T - \mathbf{x}_n \bar{\mathbf{x}}^T - \bar{\mathbf{x}} \mathbf{x}_n^T, \quad (11)$$

where $\mathbf{M}_n^{(\text{pixel})} \in \mathfrak{R}^{B \times B}$ is made by the corresponding pixel \mathbf{x}_n and $\bar{\mathbf{x}}$, and its construction can be easily understood by thinking in terms of the product of subtracted values,

$$\begin{aligned} \mathbf{p}_n(i) \mathbf{p}_n(j) &= (\mathbf{x}_n(i) - \bar{\mathbf{x}}(i))(\mathbf{x}_n(j) - \bar{\mathbf{x}}(j)) \\ &= \mathbf{x}_n(i) \mathbf{x}_n(j) + \bar{\mathbf{x}}(i) \bar{\mathbf{x}}(j) - \mathbf{x}_n(i) \bar{\mathbf{x}}(j) - \bar{\mathbf{x}}(i) \mathbf{x}_n(j). \end{aligned} \quad (12)$$

Therefore, for real-time processing, the covariance matrix obtained by the pixel-based SC-PCA now is

$$\mathbf{C}^{(\text{pixel})} = \sum_{n=1}^{HW} \mathbf{x}_n \mathbf{x}_n^T + \sum_{n=1}^{HW} \mathbf{M}_n^{(\text{pixel})}. \quad (13)$$

In the row-based approach, a correction factor \mathbf{M} must be applied again, similar to the pixel-based case:

$$\begin{aligned} \mathbf{P}_h^{(R)} [\mathbf{P}_h^{(R)}]^T &= \mathbf{X}_h^{(R)} [\mathbf{X}_h^{(R)}]^T + \mathbf{M}_h^{(R)} \mathbf{M}_h^{(R)} \\ &= \bar{\mathbf{x}} \bar{\mathbf{x}}^T - \mathbf{X}_h^{(R)} [\bar{\mathbf{x}} \cdots \bar{\mathbf{x}}]_{B \times W}^T - [\bar{\mathbf{x}} \cdots \bar{\mathbf{x}}]_{B \times W} [\mathbf{X}_h^{(R)}]^T, \end{aligned} \quad (14)$$

where $\mathbf{M}_h^{(R)} \in \mathfrak{R}^{B \times B}$ is made by the corresponding partition or subspace $\mathbf{X}_h^{(R)}$ and $\bar{\mathbf{x}}$. Now the covariance matrix is obtained, again adding the correction factor at the end of the acquisition:

$$\mathbf{C}^{(\text{Row})} = \sum_{h=1}^H \mathbf{X}_h^{(R)} [\mathbf{X}_h^{(R)}]^T + \sum_{h=1}^H \mathbf{M}_h^{(R)}. \quad (15)$$

Analogously to the row-based approach, a correction is added in the column-based case:

$$\begin{aligned} \mathbf{P}_w^{(C)} [\mathbf{P}_w^{(C)}]^T &= \mathbf{X}_w^{(C)} [\mathbf{X}_w^{(C)}]^T + \mathbf{M}_w^{(C)} \mathbf{M}_w^{(C)} \\ &= \bar{\mathbf{x}} \bar{\mathbf{x}}^T - \mathbf{X}_w^{(C)} [\bar{\mathbf{x}} \cdots \bar{\mathbf{x}}]_{B \times H}^T - [\bar{\mathbf{x}} \cdots \bar{\mathbf{x}}]_{B \times H} [\mathbf{X}_w^{(C)}]^T, \end{aligned} \quad (16)$$

where $\mathbf{M}_w^{(C)} \in \mathfrak{R}^{B \times B}$ is made by the corresponding partition or subspace $\mathbf{X}_w^{(C)}$ and $\bar{\mathbf{x}}$. Finally, a similar expression to the other cases but now using column-based partitions is achieved:

$$\mathbf{C}^{(\text{Col})} = \sum_{w=1}^W \mathbf{X}_w^{(C)} [\mathbf{X}_w^{(C)}]^T + \sum_{w=1}^W \mathbf{M}_w^{(C)}. \quad (17)$$

The correction elements in Eqs. (13), (15), and (17) are in fact equivalent to each other and can be computed in the same way for these three approaches as a correction matrix $\mathbf{CM} \in \mathfrak{R}^{B \times B}$:

$$\begin{aligned} \mathbf{CM} &= \sum_{n=1}^{HW} \mathbf{M}_n^{(\text{pixel})} = \sum_{h=1}^H \mathbf{M}_h^{(R)} = \sum_{w=1}^W \mathbf{M}_w^{(C)} \\ &= \sum_{n=1}^{HW} [\bar{\mathbf{x}} \bar{\mathbf{x}}^T - \mathbf{x}_n \bar{\mathbf{x}}^T - \bar{\mathbf{x}} \mathbf{x}_n^T]. \end{aligned} \quad (18)$$

In addition, it is unnecessary to wait until the end of the acquisition process in order to start determining the correction matrix. Taking the pixel-based SC-PCA for example; an element (i, j) in the correction matrix \mathbf{CM} can be expressed as

$$\begin{aligned} \mathbf{CM}(i, j) &= \sum_{n=1}^{HW} \bar{\mathbf{x}}(i) \bar{\mathbf{x}}(j) - \sum_{n=1}^{HW} \mathbf{x}_n(i) \bar{\mathbf{x}}(j) - \sum_{n=1}^{HW} \bar{\mathbf{x}}(i) \mathbf{x}_n(j) \\ &= HW(\bar{\mathbf{x}}(i) \bar{\mathbf{x}}(j)) - \bar{\mathbf{x}}(j) \sum_{n=1}^{HW} \mathbf{x}_n(i) - \bar{\mathbf{x}}(i) \sum_{n=1}^{HW} \mathbf{x}_n(j), \end{aligned} \quad (19)$$

where the second and third element, multiplying and dividing by the same factor, can be obtained as

$$\begin{aligned} \bar{\mathbf{x}}(j) HW \frac{1}{HW} \sum_{n=1}^{HW} \mathbf{x}_n(i) &= \bar{\mathbf{x}}(j) HW \bar{\mathbf{x}}(i) \\ \bar{\mathbf{x}}(i) HW \frac{1}{HW} \sum_{n=1}^{HW} \mathbf{x}_n(j) &= \bar{\mathbf{x}}(i) HW \bar{\mathbf{x}}(j). \end{aligned} \quad (20)$$

Accordingly, we can further define correction matrix \mathbf{CM} as

$$\mathbf{CM}(i, j) = -HW \bar{\mathbf{x}}(i) \bar{\mathbf{x}}(j). \quad (21)$$

And finally, \mathbf{CM} can be obtained by

$$\mathbf{CM}(i, j) = -\frac{1}{HW} \sum_{n=1}^{HW} \mathbf{x}_n(i) \sum_{n=1}^{HW} \mathbf{x}_n(j). \quad (22)$$

As a result, elements in the correction matrix can be obtained by accumulating the values in line with the process when the data is acquired. Finally, the covariance matrices in pixel-, row-, and column-based SC-PCA approaches can be, respectively, obtained by correction using \mathbf{CM} as follows:

$$\begin{aligned} \mathbf{C}^{(\text{pixel})} &= \sum_{n=1}^{HW} \mathbf{x}_n \mathbf{x}_n^T + \mathbf{CM} \\ \mathbf{C}^{(\text{Row})} &= \sum_{h=1}^H \mathbf{X}_h^{(R)} [\mathbf{X}_h^{(R)}]^T + \mathbf{CM} \\ \mathbf{C}^{(\text{Col})} &= \sum_{w=1}^W \mathbf{X}_w^{(C)} [\mathbf{X}_w^{(C)}]^T + \mathbf{CM}. \end{aligned} \quad (23)$$

F. Equivalency of the Approaches

It is worth noting that these four proposed strategies are in fact equivalent to the conventional PCA. Below we will show how the pixel-based SC-PCA scheme is actually equal to the conventional PCA.

In conventional PCA, the covariance matrix \mathbf{C} is defined by $\mathbf{C} = \mathbf{PP}^T \in \mathfrak{R}^{B \times B}$ from Eq. (3). For each element (i, j) , it can be expressed as

$$\mathbf{C}(i, j) = \sum_{n=1}^{HW} \mathbf{p}_n(i) \mathbf{p}_n(j). \quad (24)$$

On the other hand, in the pixel-based approach, the partial covariance matrix is determined by

$$\begin{aligned} \mathbf{C}_n^{(\text{pixel})} &= \mathbf{p}_n \mathbf{p}_n^T \\ &= \begin{bmatrix} \mathbf{p}_n(1) \mathbf{p}_n(1) & \cdots & \mathbf{p}_n(1) \mathbf{p}_n(B) \\ \vdots & \ddots & \vdots \\ \mathbf{p}_n(B) \mathbf{p}_n(1) & \cdots & \mathbf{p}_n(B) \mathbf{p}_n(B) \end{bmatrix}_{B \times B}. \end{aligned} \quad (25)$$

According to Eq. (4), the covariance matrix can be obtained by accumulating these partial covariance matrices as

$$\begin{aligned} \mathbf{C}^{(\text{pixel})} &= \sum_{n=1}^{HW} \mathbf{C}_n^{(\text{pixel})} \\ &= \begin{bmatrix} \sum_{n=1}^{HW} \mathbf{p}_n(1) \mathbf{p}_n(1) & \cdots & \sum_{n=1}^{HW} \mathbf{p}_n(1) \mathbf{p}_n(B) \\ \vdots & \ddots & \vdots \\ \sum_{n=1}^{HW} \mathbf{p}_n(B) \mathbf{p}_n(1) & \cdots & \sum_{n=1}^{HW} \mathbf{p}_n(B) \mathbf{p}_n(B) \end{bmatrix}_{B \times B}. \end{aligned} \quad (26)$$

If we compare Eqs. (24) and (26), it is apparent that the pixel-based SC-PCA approach generates the same covariance matrix as conventional PCA does; hence the two approaches are equivalent to each other. Similar mechanisms can be also used to prove

the equivalency of the conventional PCA to other proposed SC-PCA approaches.

4. Experimental Results

The proposed approaches have been implemented using Matlab, along with the conventional PCA, yet the achieved results are independent of any specific programming language. With the extracted principal components from various PCA schemes, a support vector machine (SVM) takes them as input vectors for data classification. Corresponding results are then compared to validate the effectiveness of these SC-PCA schemes. Four stages of the experimental setup, including data description, data conditioning, feature extraction, and data classification, are discussed in detail as follows, along with the results.

A. Data Description

The Airborne Visible/InfraRed Imaging Spectrometer (AVIRIS) is a sensor instrument widely used by HSI researchers for data acquisition, delivering calibrated images in 224 contiguous bands with spectral wavelengths ranging from 400 to 2500 nm [15]. Another well-known remote sensing instrument is the Reflective Optics System Imaging Spectrometer (ROSIS), providing 114 bands with a spectral range between 430 and 860 nm [16]. Finally HYPERION, aboard the EO-1 spacecraft, is able to provide scenes with more than 200 spectral bands in the range 400–2500 nm from on-orbit missions [17]. Using the above three sensors, three publicly available data sets with defined ground truth [18] are used in our experiments for quantitative performance evaluation.

First, the AVIRIS Indian Pines data set, as shown in Fig. 6, was collected over an agricultural study site in northwest Indiana in United States. The image is made of 145×145 pixels with 220 spectral reflectance bands in the wavelength range 400–2500 nm. This data set is for land usage evaluation purpose, where sixteen land cover classes are labeled in the image, presenting mostly agriculture, forest, and perennial vegetation.

Second, the ROSIS Pavia University A data set (Pavia UA, shown in Fig. 7), corresponding to

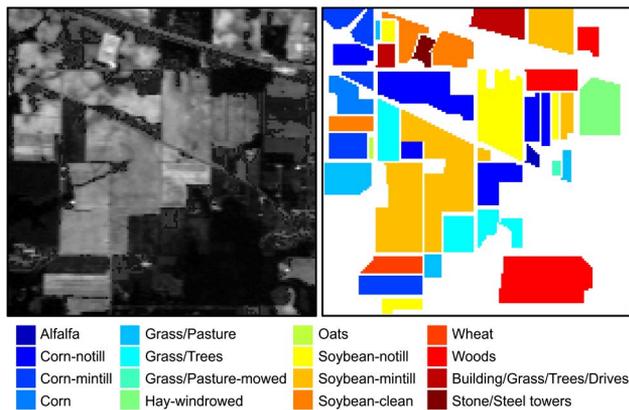


Fig. 6. One band image at a wavelength of 667 nm (left) and the ground truth maps (right) for the Indian Pines data set.

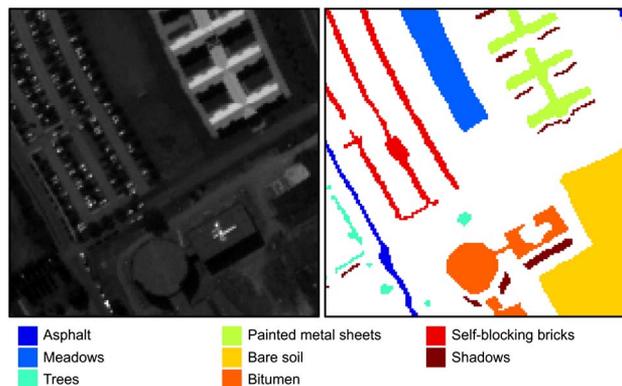


Fig. 7. One band image at a wavelength of 521 nm (left) and the ground truth maps (right) for the Pavia UA data set.

northern Italy, is a subscene made of 150×150 pixels, with 114 spectral bands at a geometric resolution of 1.3 m. A total number of eight classes can be differentiated in its ground truth, corresponding to meadows, asphalt, bare soil, and trees, among others.

Finally, the HYPERION Botswana A data set (see Fig. 8), a 75×300 pixels subscene extracted from the image taken over the Okavango Delta in Botswana, is used, providing 242 spectral bands in the range 400–2500 nm. With a 30 m resolution, the ground truth contains five classes corresponding to different acacia and mopane land cover types.

B. Data Conditioning

No specific data conditioning has been applied, except as recommended by others [18] to remove certain bands affected by severe noise or water absorption. After this preprocessing, the number of bands for AVIRIS Indian Pines data set is reduced from 220 to 200. For ROSIS Pavia UA and HYPERION Botswana A data sets, their number of bands is reduced from 114 to 103 and from 242 to 145, respectively. These data are then taken as input to the PCA for feature extraction.

C. Feature Extraction

Taking the spectral data as input, PCA is applied on such data to extract features for the following

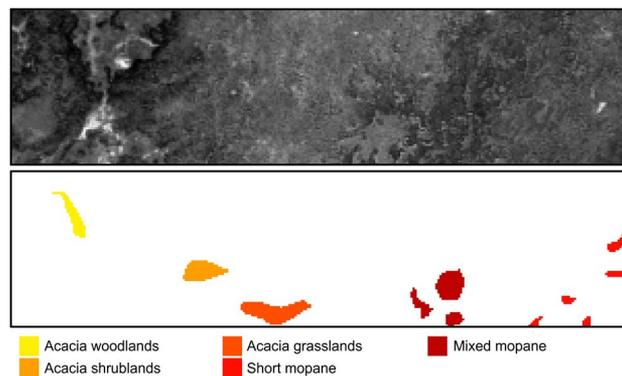


Fig. 8. One band image at a wavelength of 671 nm (top) and the ground truth maps (bottom) for the Botswana A data set.

classification tasks. Implemented in Matlab, the conventional PCA and the proposed SC-PCA strategies are employed for feature extraction, respectively. Therefore, in total five approaches are compared, which include

- Conventional PCA (PCA)
- Pixel-based SC-PCA (SC-PCA/P)
- Row-based SC-PCA (SC-PCA/R)
- Column-based SC-PCA (SC-PCA/C)
- Band-based SC-PCA (SC-PCA/B).

D. Data Classification

The SVM classifier is implemented using LIBSVM [19], a publicly available library with interface to Matlab. Although the classifier supports several kernels including linear, polynomial, and Gaussian radial basis function (RBF), it is found that the Gaussian RBF kernel produces particular good results. Consequently, the RBF kernel is selected in all our experiments, and this is consistent with findings from many other researchers [20,21]. To learn the SVM model, the training ratio is always fixed at a relatively low value of 30%. The number of PCA reduced components is tested on the interval 1–10.

The data sets are randomly split for training and testing by stratified sampling inside each class in the ground truth. This is repeated 10 times providing a total of 10 possible experiments. Two parameters for the RBF kernel, the penalty C and the gamma γ , are optimized for every experiment in the training process through a grid search. For the five PCA approaches, the mean classification rate in terms of overall accuracy and the standard deviation over the 10 experiments are obtained for performance evaluation. These are reported in the following.

E. Results and Discussion

First of all, classification rates using features from the different PCA approaches is compared in Table 2, where the numbers of principal components used for classification is 10. As can be seen, the results from these five PCA approaches are exactly the same.

In Fig. 9, the classification rate under different numbers of PCA components, 1–10, is compared for the three data sets. The general behavior is an increment along with more components, yet it shows a fast increase for the Pavia UA and the Botswana A data sets when the number of PCA components changes from 1 to 2. For the Indian Pines data set, the increment is modest for the first 3 or 4 PCA components. When the number of PCA components is 5

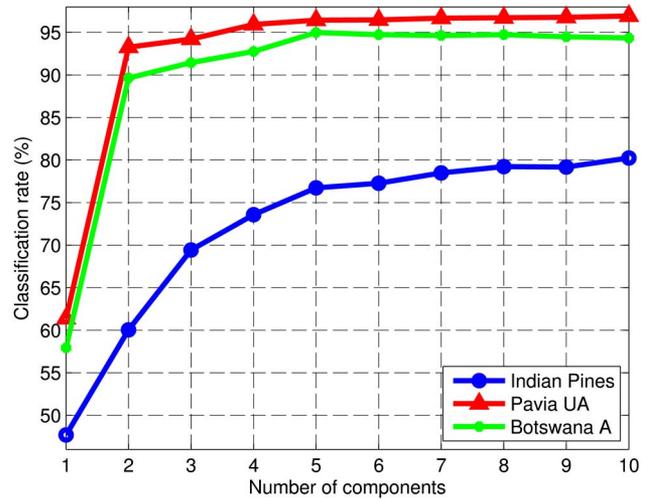


Fig. 9. Classification rate (%) using 1–10 components extracted for the three data sets.

Table 3. Matrix Sizes and Memory Requirements (kB) in the Covariance Matrix Computation for the Three Data Sets

Method	Matrices Size	Indian Pines	Pavia UA	Botswana A
PCA	$B \times HW$	33,640	18,540	26,100
SC-PCA/P	$B \times 1$	1.6	0.83	1.2
SC-PCA/R	$B \times W$	232	124	348
SC-PCA/C	$B \times H$	232	124	87
SC-PCA/B	$1 \times HW$	168	180	180

and over, the increment is small and becomes quite stable when the number reaches 10. All the classification rates shown can be achieved by any of our proposed schemes with no difference, proving therefore their mathematical equivalency.

In Table 3, memory requirements for the five approaches over the three different data sets are compared, where the data format used in Matlab is double (float), that is, 8 bytes for each data value. Conventional PCA in a Matlab environment needs massive contiguous memory in calculating the covariance matrix, yet the proposed approaches have significantly reduced this requirement. In fact, as shown in Table 3, the ratio of memory reduction in the pixel-, row-, column-, and band-based schemes is HW , H , W , and B , respectively.

Regarding the computational complexity, the number of multiplications and additions needed for the five approaches are given in Table 4 for comparison.

Table 2. Means and Standard Deviations of the Classification Rate (%) for the Three Data Sets with 10 Principal Components

Method	Indian Pines	Pavia UA	Botswana A
PCA	80.24 ± 0.31	96.93 ± 0.20	94.34 ± 1.67
SC-PCA/P	80.24 ± 0.31	96.93 ± 0.20	94.34 ± 1.67
SC-PCA/R	80.24 ± 0.31	96.93 ± 0.20	94.34 ± 1.67
SC-PCA/C	80.24 ± 0.31	96.93 ± 0.20	94.34 ± 1.67
SC-PCA/B	80.24 ± 0.31	96.93 ± 0.20	94.34 ± 1.67

Table 4. Number of Multiplications and Additions in the Covariance Matrix Computation for Conventional PCA and the SC-PCA Schemes When Applied on the Whole Hypercube

Method	Multiplications	Additions
PCA	$B + B^2HW$	$B(2HW - 1) + B^2(HW - 1) + 0$
SC-PCA/P	$B + B^2HW$	$B(2HW - 1) + 0 + B^2(HW - 1)$
SC-PCA/R	$B + B^2HW$	$B(2HW - 1) + B^2H(W - 1) + B^2(H - 1)$
SC-PCA/C	$B + B^2HW$	$B(2HW - 1) + B^2(H - 1)W + B^2(W - 1)$
SC-PCA/B	$B + B^2HW$	$B(2HW - 1) + B^2(HW - 1) + 0$

Table 5. Number of Multiplications and Additions in the Covariance Matrix Computation for the Proposed SC-PCA Schemes Distributed in the Acquisition Process

Method	Loop (k) Complexity		Loop (k)	After Acquisition	
	Mult.	Add.		Mult.	Add.
SC-PCA/P	B^2	$3B^2$	HW	$2B^2$	B^2
SC-PCA/R	B^2W	$3B^2W$	H	$2B^2$	B^2
SC-PCA/C	B^2H	$3B^2H$	W	$2B^2$	B^2
SC-PCA/B	$HW(2k - 1) + 1$	$HW(2k + 1) - 2k$	B	0	0

The multiplications contain two parts: the first is for the mean-adjustment implementation, while the second is for the covariance matrix construction. On the other hand, the additions needed contain three parts: the first for the mean adjustment, the second for calculating the partial covariance matrices, and the third for their summation. Not surprisingly, the number of required multiplications and additions is exactly the same, yet these operations are differently distributed.

In the following, the numbers of multiplications and additions needed for real-time covariance computation in the four SC-PCA approaches are compared. In Table 5, the computational complexity in each loop is shown. As can be seen, SC-PCA/P has the minimum required computations in each loop, yet it takes HW sequential scans to complete the acquisition process. In contrast, the other three SC-PCA approaches have more computations in each loop, yet the overall number of sequential scans is much reduced. The total computational cost in computing the covariance matrix for all the SC-PCA approaches is almost the same.

Also a comparison based on the running time is employed to assess the efficiency of different schemes used. As shown in Fig. 10, though the band-based approach is the most efficient, among the group of four approaches, pixel-, row-, and column-based

schemes only need a few extra milliseconds to complete the covariance matrix computation, dramatically reducing the time consumed by the conventional PCA. This has clearly proven that all the proposed SC-PCA approaches outperform the conventional PCA in terms of significantly improved efficiency.

Finally, it is worth noting that based on the symmetry of the covariance matrix, its computational cost can be further reduced approximately from B^2 to $B(B + 1)/2$ as explained in [13]. However, for simplicity this point as well as the potential for parallel implementation or calibration error is not considered in this paper.

5. Conclusions

No matter how the HSI data is acquired, conventional PCA can only be implemented offline as it requires the mean vector and the covariance matrix to be first obtained from the fully completed hypercube. As a result, it also suffers from a large amount of memory needed in computing the covariance matrix. In line with the way the hypercube is acquired, four different SC-PCA schemes are proposed to determine the covariance matrix in a real-time manner while the data is acquired. Theoretical analysis and experimental results have verified the equivalency of the SC-PCA approaches to the conventional PCA. Consequently, the proposed approaches have great potential to facilitate the subsequent data processing and analytics in terms of much improved efficiency and significantly reduced memory requirement. Nevertheless, a much reduced memory requirement can also be achieved in efficient computation of the covariance matrix when implementing the proposed schemes on the completed hypercube. In addition, as the partial covariance matrices can be separately computed, parallel implementation can be introduced to further improve the computational efficiency. Future work will focus on optimization of other feature extraction algorithms, such as singular spectrum analysis, folded PCA, and non-PCA-based approaches [22–24].

References

1. Z. Liu, J. Yan, D. Zhang, and Q.-L. Li, "Automated tongue segmentation in hyperspectral images for medicine," *Appl. Opt.* **46**, 8328–8334 (2007).
2. K. Gill, J. Ren, S. Marshall, S. Karthick, and J. Gilchrist, "Quality-assured fingerprint image enhancement and extraction using hyperspectral imaging," in *4th International*

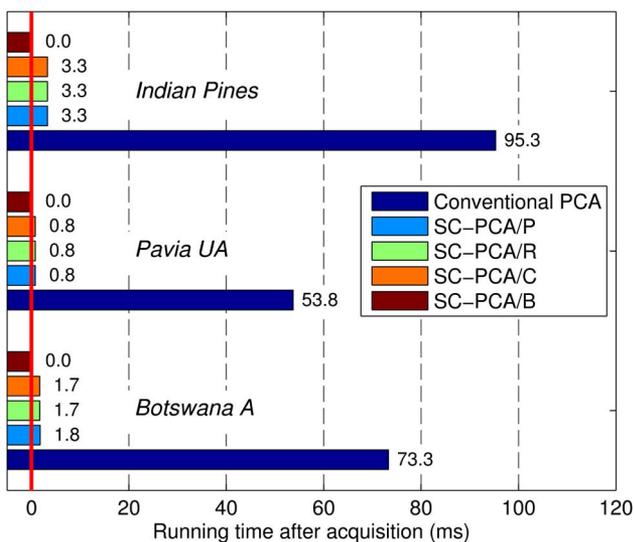


Fig. 10. Approximated running time (ms) in the covariance matrix computation for the three data sets after acquisition is completed (red line).

- Conference on Imaging for Crime Detection and Prevention*, London (2011).
3. S. Sumriddetchkajorn and Y. Intaravanne, "Hyperspectral imaging-based credit card verifier structure with adaptive learning," *Appl. Opt.* **47**, 6594–6600 (2008).
 4. T. Kelman, J. Ren, and S. Marshall, "Effective classification of Chinese tea samples in hyperspectral imaging," *Artific. Intell. Res.* **2**, 87–96 (2013).
 5. C. Zhao, X. Li, J. Ren, and S. Marshall, "Improved sparse representation using adaptive spatial support for effective target detection in hyperspectral imagery," *Int. J. Remote Sens.* **34**, 8669–8684 (2013).
 6. S. E. Craig, S. E. Lohrenz, Z. Lee, K. L. Mahoney, G. J. Kirkpatrick, O. M. Schofield, and R. G. Steward, "Use of hyperspectral remote sensing reflectance for detection and assessment of the harmful alga, *Karenia brevis*," *Appl. Opt.* **45**, 5414–5425 (2006).
 7. G. F. Hughes, "On the mean accuracy of statistical pattern recognition," *IEEE Trans. Inf. Theory* **14**, 55–63 (1968).
 8. H. Abdi and L. J. Williams, *Principal Component Analysis* (WIREs Comp Stat, 2010).
 9. R. Dianat and S. Kasaei, "Dimension reduction of optical remote sensing images via minimum change rate deviation method," *IEEE Trans. Geosci. Remote Sens.* **48**, 198–206 (2010).
 10. T. W. Du Bosq, J. M. Lopez-Alonso, and G. D. Boreman, "Millimeter wave imaging system for land mine detection," *Appl. Opt.* **45**, 5686–5692 (2006).
 11. F. Ndi, F. Adar, and S. H. Atzeni, "Spectral imaging," *Readout* **38**, 68–73 (2011).
 12. F. Vagni, "Survey of hyperspectral and multispectral imaging technologies," NATO Tech. Rep., 2007.
 13. R. Jošth, J. Antikainen, J. Havel, A. Herout, P. Zemčík, and M. Hauta-Kasari, "Real-time PCA calculation for spectral imaging (using SIMD and GP-GPU)," *J. Real Time Image Proc.* **7**, 1–9 (2012).
 14. M.-Z. Wang, D.-M. Wang, W.-X. Xu, B.-Y. Chen, and K. Guo, "Parallel computing of covariance matrix and its application on hyperspectral data process," in *Geoscience and Remote Sensing Symposium (IGARSS)*, July 22–27, 2012, pp. 4058–4061.
 15. R. O. Green, M. L. Eastwood, C. M. Sarture, T. G. Chrien, M. Aronsson, B. J. Chippendale, J. A. Faust, B. E. Pavri, C. J. Chovit, M. Solis, M. R. Olah, and O. Williams, "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sens. Environ.* **65**, 227–248 (1998).
 16. S. Holzwarth, A. Müller, M. Habermeyer, R. Richter, A. Hausold, S. Thiemann, and P. Strohl, "HySens-DAIS 7915/ROSIS imaging spectrometers at DLR," in *Proceedings of the 3rd Earsel Workshop on Imaging Spectroscopy*, Herrsching, Germany, May 13–16, 2003, pp. 3–14.
 17. J. S. Pearlman, P. S. Barry, C. C. Segal, J. Shepanski, D. Beiso, and S. L. Carman, "Hyperion, a space-based imaging spectrometer," *IEEE Trans. Geosci. Remote Sens.* **41**, 1160–1173, (2003).
 18. "Hyperspectral remote sensing scenes," 2014, http://www.ehu.es/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.
 19. C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," *ACM Trans. Intell. Syst. Technol.* **2**, 1–27 (2013).
 20. F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.* **42**, 1778–1790 (2004).
 21. M. Rojas, I. Dópido, A. Plaza, and P. Gamba, "Comparison of support vector machine-based processing chains for hyperspectral image classification," *Proc. SPIE* **7810**, 78100B (2010).
 22. J. Zabalza, J. Ren, Z. Wang, S. Marshall, and J. Wang, "Singular spectrum analysis for effective feature extraction in hyperspectral imaging," *IEEE Geosci. Remote Sens. Lett.* **11**, 1886–1890 (2014).
 23. J. Zabalza, J. Ren, M. Yang, Y. Zhang, J. Wang, S. Marshall, and J. Han, "Novel folded-PCA for improved feature extraction and data reduction with hyperspectral imaging and SAR in remote sensing," *ISPRS J. Photogr. Remote Sens.* **93**, 112–122 (2014).
 24. J. Ren, J. Zabalza, S. Marshall, and J. Zheng, "Effective feature extraction and data reduction with hyperspectral imaging in remote sensing," *IEEE Signal Process. Mag.* **31**(4), 149–154 (2014).