Zabalza, Jaime and Ren, Jinchang and Marshall, Stephen (2015) 'On the fly' dimensionality reduction for hyperspectral image acquisition. In: 23rd European Signal Processing Conference, 2015 (EUSIPCO 2015), 2015-08-31 - 2015-09-04. , http://dx.doi.org/10.1109/EUSIPCO.2015.7362483

‘ON THE FLY’ DIMENSIONALITY REDUCTION FOR HYPERSPECTRAL IMAGE ACQUISITION

Jaime Zabalza, Jinchang Ren, and Stephen Marshall

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

ABSTRACT

Hyperspectral imaging (HSI) devices produce 3-D hypercubes of a spatial scene in hundreds of different spectral bands, generating large data sets which allow accurate data processing to be implemented. However, the large dimensionality of hypercubes leads to subsequent implementation of dimensionality reduction techniques such as principal component analysis (PCA), where the covariance matrix is constructed in order to perform such analysis. In this paper, we describe how the covariance matrix of an HSI hypercube can be computed in real time ‘on the fly’ during the data acquisition process. This offers great potential for HSI embedded devices to provide not only conventional HSI data but also preprocessed information.

Index Terms— Covariance matrix, data reduction, hypercube, hyperspectral cameras, principal component analysis (PCA)

1. INTRODUCTION

A large number of applications and developments have been proposed in recent years with relation to the use of hyperspectral imaging (HSI) data for signal and image processing. HSI sensors and cameras provide what is called hypercube, a 3-D structure where the pixels in a spatial scene are formed by a vector array with each of the vector elements corresponding to a given wavelength in the spectrum. Therefore, this large amount of data allows in-depth data processing to be applied in many diverse areas such as food analysis and security [1-2].

However, such large volumes of data require complex analysis. For that reason, HSI hypercubes and related data are usually subject to a feature extraction process, where different techniques are used to extract salient features [3-6]. This also includes dimensionality reduction, where the high correlation between adjacent spectral bands is addressed by classical and well-known techniques such as principal component analysis (PCA), independent component analysis (ICA), and maximum noise fraction (MNF) [3, 6]. In particular PCA with a number of its variants [7-9] is one of the most widely used methods in HSI.

Implementation of many of these algorithms usually requires the computation of spectral covariance matrices as a way of capturing information across the whole data cube. This computation can be quite complex in HSI applications, when the corresponding hypercube is of extremely large dimension in both spectral (hundreds of wavelengths) and spatial (thousands of pixels) domains, so not surprisingly, the literature already documents a number of parallel implementations [10-11]. However, bearing in mind the acquisition process by which many HSI cameras operate, i.e. sequentially acquiring sub-partitions of data, a novel innovation proposed here is to carry out the real time ‘on the fly’ computation of the spectral covariance matrix within the image capture device simultaneously with the acquisition procedure [12], as we explain in this paper.

2. ‘ON THE FLY’ COVARIANCE COMPUTATION

Our proposal involves including dedicated signal processing within the HSI devices so their sequential acquisition of data (see Section 2.2) can be exploited for alternative covariance construction (Fig. 1), relieving memory requirements and allowing real time computation of the covariance matrix.

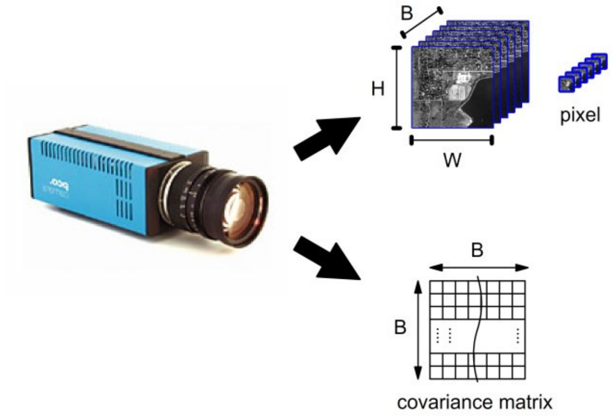

Fig. 1. ‘On the fly’ covariance for HSI devices.

2.1. Conventional covariance matrix computation

Given a pixel vector $\mathbf{x}_n = [x_{n_1}, x_{n_2}, \cdots, x_{n_B}]^T$ in a hypercube of dimensions $H \times W \times B$, where B is the number of spectral bands, the procedure here consists of partitioning the 3-D hypercube into a 2-D data matrix namely $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_{HW}] \in \Re^{B \times HW}$, where the initial pixels $\mathbf{x}_n$ are subtracted the mean value $\bar{\mathbf{x}} = \sum_{n=1}^{HW} \mathbf{x}_n / HW$, resulting in $\mathbf{p}_n = [p_{n_1}, p_{n_2}, \cdots, p_{n_B}]^T$ so the covariance matrix is obtained by the following multiplication, where the dividing term is omitted for simplicity.

$$\mathbf{C} \ = \mathbf{P}\,\mathbf{P}^T \in \Re^{B \times B} . \tag{1}$$

Even though this calculation is straightforward for modern computers, this conventional procedure still has some drawbacks, as large dimensions of HSI hypercubes can lead to memory and computation problems, making implementation in portable or embedded systems unfeasible. For that reason, we propose a simple approach for obtaining the covariance matrix ‘on the fly’ simultaneously with the acquisition process.

2.2. HSI acquisition procedures

The data capture present in current HSI devices and cameras can be divided into two groups: scanning and filter-based methods. On one hand, scanning techniques consist of a sequential procedure where the final image cube is obtained from partial spatial scenes, these may involve pixel scanning when individual pixels are acquired in every step, and line scanning (also known as push-broom technique) when spatial lines of pixels are captured to build the final image. Conversely filter-based methods collect the whole spatial scene at once but only for a particular spectral wavelength. Consequently, each spectral band is captured sequentially in every step. A schematic illustration of the different HSI acquisition techniques is shown in Fig. 2.

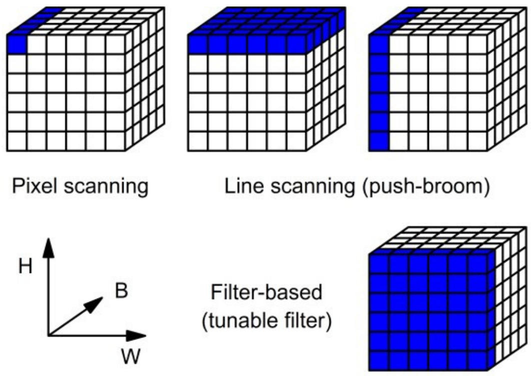

Fig. 2. HSI acquisition procedures [12].

2.3. ‘On the fly’ covariance matrix computation

Whichever acquisition technique is applied, it is clear that the procedure is sequential such that a series of small subspaces of the original data $\mathbf{P}$ can be used to gradually calculate the covariance matrix. This can result in faster implementations even up to real time.

To this end, the partition by pixel $\mathbf{p}_n$ (2), partition by row $\mathbf{P}_h^{(R)}$ (3), partition by column $\mathbf{P}_w^{(C)}$ (4), and partition by band $\mathbf{P}_b^{(B)}$ (5) can be used as subspaces of data for constructing the covariance matrix sequentially. These are defined as in Table 1.

Partition	Subspace defined	
Pixel	$\mathbf{p}_n \in \Re^{B \times 1}$	(2)
Row	$\mathbf{P}_h^{(R)} = [\mathbf{p}_h, \mathbf{p}_{h+H}, \cdots, \mathbf{p}_{h+(W-1)H}]_{B \times W}$	(3)
Column	$\mathbf{P}_w^{(C)} = [\mathbf{p}_{1+H(w-1)}, \mathbf{p}_{2+H(w-1)}, \cdots \qquad {}_{-1)}]_{B \times H}$	(4)
Band	$\mathbf{P}_b^{(B)} = \begin{bmatrix} \mathbf{p}_1(b) & \cdots & \mathbf{p}_{H(W-1)+1}(b) \\ \vdots & \ddots & \vdots \\ \mathbf{p}_H(b) & \cdots & \mathbf{p}_{HW}(b) \end{bmatrix}_{H \times W}$	(5)

Table 1. Proposed partitions and their defined subspace.

where the final covariance matrix is formed by the addition of the partial covariance matrices shown in Table 2.

Partition	Covariance matrix accumulation	
Pixel	$\mathbf{C}^{(pixel)} = \sum_{n=1}^{HW} \mathbf{p}_n \mathbf{p}_n^T$	(6)
Row	$\mathbf{C}^{(Row)} = \sum_{h=1}^{H} \mathbf{P}_h^{(R)} [\mathbf{P}_h^{(R)}]^T$	(7)
Column	$\mathbf{C}^{(Col)} = \sum_{w=1}^{W} \mathbf{P}_w^{(C)} [\mathbf{P}_w^{(C)}]^T$	(8)
Band	$\mathbf{C}^{(Band)}(i,j) = vec(\mathbf{P}_{b=i}^{(B)})[vec(\mathbf{P}_{b=j}^{(B)})]^T \ i,j \in [1,B]$	(9)

Table 2. Covariance matrix accumulation by the partitions.

2.4. Modification for real-time calculation

In cases where the covariance matrix computation is required not only ‘on the fly’ within the HSI device but also in real-time, some modifications are required to include the mean subtraction procedure usually performed before multiplication in (1).

Formulation of partitions $\mathbf{p}_n$, $\mathbf{P}_h^{(R)}$, $\mathbf{P}_w^{(C)}$, and $\mathbf{P}_b^{(B)}$ given in this manuscript already includes the mean subtraction procedure, as these partitions can be taken once the whole hypercube has been acquired to produce the covariance matrix ‘on the fly’. However, covariance can also be computed in real time by the corresponding and original non-subtracted partitions $\mathbf{x}_n$, $\mathbf{X}_h^{(R)}$, and $\mathbf{X}_w^{(C)}$. As the mean subtraction procedure requires all pixel values within a spectral band, this leads to a correction for all our approaches except the one using band partitions $\mathbf{P}_b^{(B)}$.

Revisiting, the simple pixel partition problem, its corresponding multiplication can be expressed as

$$\begin{aligned} \mathbf{p}_n\mathbf{p}_n^T &= \mathbf{x}_n\mathbf{x}_n^T + \mathbf{M}_n \\ \mathbf{M}_n &= \bar{\mathbf{x}}\bar{\mathbf{x}}^T - \mathbf{x}_n\bar{\mathbf{x}}^T - \bar{\mathbf{x}}\mathbf{x}_n^T \end{aligned}, \tag{10}$$

where $\mathbf{M}_n \in \Re^{B\times B}$ is a correction matrix calculated from the $\mathbf{x}_n$ and the average pixel $\bar{\mathbf{x}}$, an expression simply derived from the product of subtracted values

$$\begin{aligned} &\mathbf{p}_n(i)\mathbf{p}_n(j) = \left(\mathbf{x}_n(i) - \bar{\mathbf{x}}(i)\right)\left(\mathbf{x}_n(j) - \bar{\mathbf{x}}(j)\right) \\ &= \mathbf{x}_n(i)\mathbf{x}_n(j) + \bar{\mathbf{x}}(i)\bar{\mathbf{x}}(j) - \mathbf{x}_n(i)\bar{\mathbf{x}}(j) - \bar{\mathbf{x}}(i)\mathbf{x}_n(j) \end{aligned}. \tag{11}$$

Updating the covariance equation (6), now we have

$$\mathbf{C}^{(pixel)} = \sum_{n=1}^{HW}\mathbf{x}_n\mathbf{x}_n^T + \mathbf{CM}, \tag{12}$$

where the correction matrix $\mathbf{CM} \in \Re^{B\times B}$, which is equivalent for the rest of partitions, is expressed as

$$\mathbf{CM} = \sum_{n=1}^{HW}\mathbf{M}_n = \sum_{n=1}^{HW}\left[\bar{\mathbf{x}}\bar{\mathbf{x}}^T - \mathbf{x}_n\bar{\mathbf{x}}^T - \bar{\mathbf{x}}\mathbf{x}_n^T\right]. \tag{13}$$

Simply developing this correction matrix, we have

$$\begin{aligned} &\mathbf{CM}(i,j) = \sum_{n=1}^{HW}[\bar{\mathbf{x}}(i)\bar{\mathbf{x}}(j) - \mathbf{x}_n(i)\bar{\mathbf{x}}(j) - \bar{\mathbf{x}}(i)\mathbf{x}_n(j)] \\ &= HW\left(\bar{\mathbf{x}}(i)\bar{\mathbf{x}}(j)\right) - \bar{\mathbf{x}}(j)\sum_{n=1}^{HW}\mathbf{x}_n(i) - \bar{\mathbf{x}}(i)\sum_{n=1}^{HW}\mathbf{x}_n(j) \end{aligned}, \tag{14}$$

where the second and third elements can be expressed as

$$\begin{aligned} \bar{\mathbf{x}}(j)HW\frac{1}{HW}\sum_{n=1}^{HW}\mathbf{x}_n(i) &= \bar{\mathbf{x}}(j)HW\bar{\mathbf{x}}(i) \\ \bar{\mathbf{x}}(i)HW\frac{1}{HW}\sum_{n=1}^{HW}\mathbf{x}_n(j) &= \bar{\mathbf{x}}(i)HW\bar{\mathbf{x}}(j) \end{aligned}. \tag{15}$$

Finally, $\mathbf{CM}$ is formulated as (16), which means it can be calculated in real time, at each iteration

$$\mathbf{CM}(i,j) = -HW\bar{\mathbf{x}}(i)\bar{\mathbf{x}}(j) = -\frac{1}{HW}\sum_{n=1}^{HW}\mathbf{x}_n(i)\sum_{n=1}^{HW}\mathbf{x}_n(j). \tag{16}$$

3. EXPERIMENTS AND RESULTS

A complete evaluation of the different approaches is performed in order to show a comparison between them and in relation to the conventional case. Initially, the HSI data sets employed in the experiments are introduced and described. Then, mathematical equivalency among the approaches is demonstrated. Finally, a comparison in terms of memory requirements, complexity and simulated timing demonstrates the benefits of our proposed approach.

3.1. HSI datasets for evaluations

Two data sets from different HSI sensors are employed to validate our proposal. First, Pavia University A (Pavia UA) is a subscene extracted from the original urban image taken in Italy with the Reflective Optics System Imaging Spectrometer (ROSIS) camera. Shown in Fig. 3, it includes 8 labeled classes in the ground truth related to meadows, asphalt and others, presenting 150×150 pixels with 1.3 m resolution, and 103 available spectral bands in the range 430-860 nm.

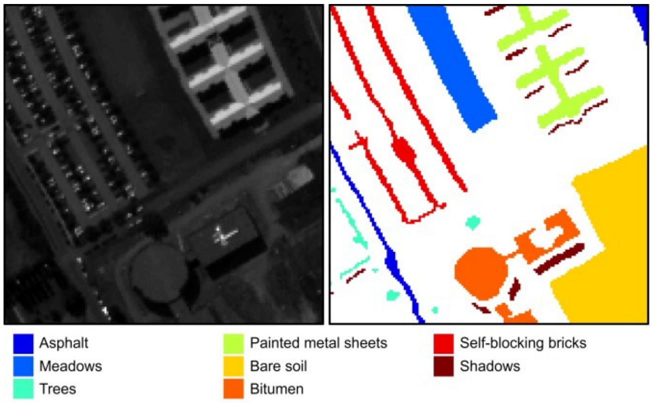

Fig. 3. Pavia UA data set description.

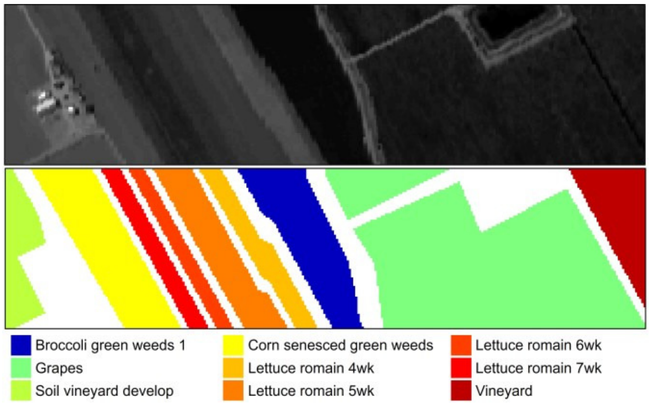

Fig. 4. Salinas B data set description.

Second, Salinas B (shown in Fig. 4) is a 75×300 pixels subscene from a natural image taken over Salinas Valley, in

California, providing spatial resolution of 3.7 m in 204 valid spectral bands. It was acquired by the Airborne Visible/InfraRed Imaging Spectrometer (AVIRIS) instrument. The ground truth here contains 9 classes corresponding to broccoli, lettuce, grapes and others.

3.2. Mathematical equivalencies

The fundamental assumption behind alternative implementations of the covariance matrix is that the resulting final matrix obtained is exactly the same in all cases. Therefore, a simple example demonstrating mathematical equivalency is presented here.

In the conventional implementation in (1), the covariance matrix $\mathbf{C}$ is obtained from a large multiplication based on the original partition of data $\mathbf{P}$. Accordingly, it is straightforward to express each element in matrix $\mathbf{C}$ as

$$\mathbf{C}(i,j)=\sum_{n=1}^{HW}\mathbf{p}_n(i)\mathbf{p}_n(j)\,. \tag{17}$$

Alternatively, for a simple pixel partition the subspaces multiplication leads to

$$\mathbf{p}_n\mathbf{p}_n^T=\begin{bmatrix}\mathbf{p}_n(1)\mathbf{p}_n(1) & \cdots & \mathbf{p}_n(1)\mathbf{p}_n(B)\\ \vdots & \ddots & \vdots\\ \mathbf{p}_n(B)\mathbf{p}_n(1) & \cdots & \mathbf{p}_n(B)\mathbf{p}_n(B)\end{bmatrix}_{B\times B}\,. \tag{18}$$

Simply by accumulating (18) for all pixels in the hypercube, the final covariance matrix is found as

$$\mathbf{C}^{(pixel)}=\begin{bmatrix}\sum_{n=1}^{HW}\mathbf{p}_n(1)\mathbf{p}_n(1) & \cdots & \sum_{n=1}^{HW}\mathbf{p}_n(1)\mathbf{p}_n(B)\\ \vdots & \ddots & \vdots\\ \sum_{n=1}^{HW}\mathbf{p}_n(B)\mathbf{p}_n(1) & \cdots & \sum_{n=1}^{HW}\mathbf{p}_n(B)\mathbf{p}_n(B)\end{bmatrix}_{B\times B} \tag{19}$$

which provides the same values as the conventional case (17) in the $B\times B$ elements of the matrix. The equivalency of the rest of partitions can be proved in a similar way.

This equivalency can be also demonstrated in practical terms. Actually, the implementation by the different partitions leads to negligible differences in the covariance matrix elements, clearly below 0.001%.

Additionally, the application of PCA by means of the different covariance matrices (Table 3) proves again the equivalency, as regardless of the partition employed to compute the covariance matrix in the PCA method, exactly the same classification results in land-cover analysis are achieved, in this case by the Support Vector Machine (SVM) classifier with a rate of 30% for training the model.

Partition used	Pavia UA	Salinas B
Conventional	96.45 ± 0.27	94.50 ± 0.16
Row	96.45 ± 0.27	94.50 ± 0.16
Column	96.45 ± 0.27	94.50 ± 0.16
Pixel	96.45 ± 0.27	94.50 ± 0.16
Band	96.45 ± 0.27	94.50 ± 0.16

Table 3. Classification accuracy (%) using PCA (5 features).

3.3. Memory requirements

By using smaller partitions of data, which can be accessed during the sequential acquisition in HSI devices, the first clear advantage is the reduced size of the multiplying matrices for covariance calculation. The diverse sizes and contiguous memory requirements for the different partitions are compared to the original case in Table 4.

Each partition sees its size reduced by a factor related to its dimensionality in the hypercube. Therefore, the pixel partition gives the highest reduction for the current data sets. Memory is expressed in kB, where data format considered is 8 bytes per value (double).

Partition	Matrices size	Pavia UA	Salinas B
Original	B×HW	18540	36720
Pixel	B×1	0.83	1.63
Row	B×W	124	490
Column	B×H	124	122
Band	1×HW	180	180

Table 4. Size of matrices and memory requirements (kB).

3.4. Number of multiplications and additions

The global number of multiplications and additions for all the cases is just the same, due to the equivalent implementations. However, there is a small difference when the implementation is performed in real time.

In the real time case, partitions require different complexity in each iteration (loop), with additional operations to be undertaken once the iteration process is completed, as can be seen in Table 5.

A trade-off between loop complexity and number of iterations is easily recognizable when using different partitions. However, only the band partition is free of further calculations after the sequential acquisition. This is simply because the real-time correction is not necessary, as the average procedure is already included in the iterations.

Partition	Single loop		Loop	Afterwards	
	Mult.	Add.	(k)	Mult.	Add.
Pixel	B^2	$3B^2$	HW	$2B^2$	B^2
Row	B^2W	$3B^2W$	H	$2B^2$	B^2
Column	B^2H	$3B^2H$	W	$2B^2$	B^2
Band	HW(2k-1)+1	HW(2k+1)-2k	B	0	0

Table 5. Multiplications and additions during acquisition.

3.5. Simulated timing comparison

Finally, to give an idea of the efficiency resulting from our proposal, a simulation of execution time required after acquisition in real time covariance computation is presented. These experiments are performed using MATLAB 8.0 on a PC with 3.1 GHz CPU and 8 GB Memory.

Fig. 5 shows the total execution time required just after the sequential acquisition process is finalised by the corresponding partition. Our approaches take advantage of the time gap between sequential acquisitions of partitions; therefore, this gap can be employed for real-time computation, which includes partial covariance accumulation and related issues such as memory access and data transfer.

Once the acquisition process is completed (red line in Fig. 5), band partitions require no further operations, while the remaining row, column and pixel partitions still need a few milliseconds to complete the covariance calculation. Nevertheless, these times are very small in comparison to the timing needed if the original partition is selected, which clearly shows the efficiency involved by the 'on the fly' real time calculation.

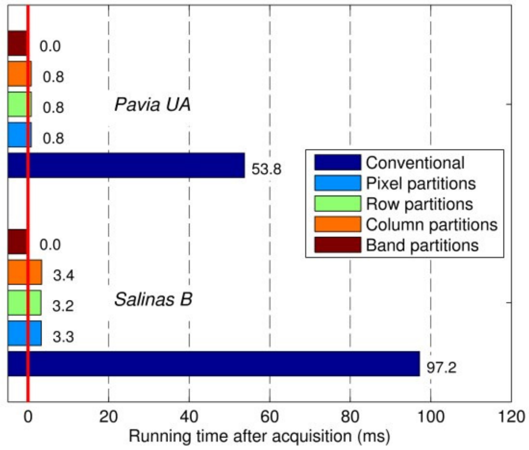

Fig. 5. Simulated timing for different partitions.

4. CONCLUSIONS

The fast growth and remarkable development experienced by HSI technology in recent years has dramatically increased its use as it becomes a tool with enormous potential and a promising future. Due to the high dimensionality of the hypercubes, large volumes of data must be processed. Common preprocessing techniques such as PCA and others, require access to the covariance matrix. To this end, its efficient computation is of huge value in many HSI applications.

However, covariance acquisition from high dimensional HSI data sets suffers from huge drawbacks, especially in portable and embedded systems, due to restricted capacities in terms of memory and computation power. For this reason, we propose an 'on the fly' procedure, which makes the computation of the covariance matrix feasible in real time directly from HSI devices with embedded processing.

REFERENCES

[1] S. Sumriddetchkajorn, and Y. Intaravanne, "Hyperspectral imaging-based credit card verifier structure with adaptive learning", *Applied Optics* vol. 47, no. 35, pp. 6594-6600, 2008.

[2] T. Kelman, J. Ren, and S. Marshall, "Effective classification of Chinese tea samples in hyperspectral imaging", *Artificial Intelligence Research*, vol. 2, no. 4, pp. 87-96, 2013.

[3] X. Jia, B-C. Kuo, and M. M. Crawford, "Feature mining for hyperspectral image classification", *Proceedings of the IEEE*, vol. 101, no. 3, pp. 676-697, March 2013.

[4] J. Zabalza, J. Ren, Z. Wang, S. Marshall, and J. Wang, "Singular spectrum analysis for effective feature extraction in hyperspectral imaging," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 11, pp. 1886–1890, November 2014.

[5] J. Zabalza, J. Ren, J. Zheng, J. Han, H. Zhao, S. Li, and S. Marshall, "Novel two-dimensional singular spectrum analysis for effective feature extraction and data classification in hyperspectral imaging," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 8, pp. 4418–4433, August 2015.

[6] I. Dopido, A. Villa, A. Plaza, and P. Gamba, "A comparative assessment of several processing chains for hyperspectral image classification: What features to use?", in *Proc. WHISPERS*, 2011.

[7] X. Jia and J. A. Richards, "Segmented principal components transformation for efficient hyperspectral remote-sensing image display and classification", *IEEE Trans. Geoscience and Remote Sensing*, vol. 37, no. 1, pp. 538-542, January 1999.

[8] 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 Journal of Photogrammetry and Remote Sensing*, vol. 93, pp.112-122, July 2014.

[9] J. Ren, J. Zabalza, S. Marshall, and J. Zheng, "Effective feature extraction and data reduction in remote sensing using hyperspectral imaging [Applications Corner]", *IEEE Signal Processing Magazine*, vol. 31, no. 4, pp. 149-154, July 2014.

[10] 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)", *Journal of Real-Time Image Processing*, vol. 7, no. 2, pp. 95-103, April 2012.

[11] 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 *Proc. Geoscience and Remote Sensing Symposium (IGARSS)*, 2012.

[12] J. Zabalza, J. Ren, J. Ren, Z. Liu, and S. Marshall, "Structured covariance principal component analysis for real-time onsite feature extraction and dimensionality reduction in hyperspectral imaging", *Applied Optics*, vol. 53, no. 20, pp. 4440-4449, July 2014.