

# Spectral/Spatial Hyperspectral Image Compression

Bharath Ramakrishna<sup>1</sup> Antonio J. Plaza<sup>1,2</sup> Chein-I Chang<sup>1</sup> Hsuan Ren<sup>3</sup>  
Qian Du<sup>4</sup> Chein-Chi Chang<sup>5</sup>

<sup>1</sup>Remote Sensing Signal and Image Processing Laboratory  
Department of Computer Science and Electrical Engineering  
University of Maryland, Baltimore County, Baltimore, MD 21250

<sup>2</sup>Computer Science Department, University of Extremadura  
Avda. de la Universidad s/n,10.071 Caceres, SPAIN

<sup>3</sup>Center for Space and Remote Sensing Research  
Graduate Institute of Space Science  
Department of Computer Science and Information Engineering

National Central University, Chungli, Taiwan, ROC

<sup>4</sup>Department of Electrical and Computer Engineering  
Mississippi State University, Mississippi State, MS 39762

<sup>5</sup>Department of Civil and Environmental Engineering  
University of Maryland, Baltimore County, Baltimore, MD 21250

## 1. INTRODUCTION

Hyperspectral image compression has received considerable interest in recent years due to enormous data volumes collected by imaging spectrometers which consists of hundreds of contiguous spectral bands with very high between-band spectral correlation. Due to such significantly improved spatial and spectral resolution provided by a hyperspectral imaging sensor, hyperspectral imagery expands the capability of multispectral imagery in many ways, such as subpixel target detection, object discrimination, mixed pixel classification, material quantification, etc. It also presents new challenges to image analysts, particularly, how to effectively deal with its enormous data volume so as to achieve their desired goals. One common practice is to compress data prior to image analysis. Two types of data compression can be performed, lossless and lossy in accordance with redundancy removal. More specifically, lossless data compression is generally considered as data compaction which eliminates *unnecessary* redundancy without loss of information. By contrast, lossy data compression removes *unwanted* information or *insignificant* information which results in entropy reduction. Which compression should be used depends heavily upon

various applications. For example, in medical imaging, lossless compression is preferred to lossy compression in order to avoid potential lawsuits against doctors. However, in this case, only small compression ratios can be achieved, generally less than 3:1. On the other hand, video processing such as HDTV (High Definition TV) can benefit from lossy compression. For remotely sensed imagery, both types of compression can be beneficial and have been studied and investigated extensively in the past [1-8]. Since we are interested in exploitation-based applications, data analysis is generally determined by features of objects in the image data rather than the image itself. As a result, lossless compression may not offer significant advantages over lossy compression in the sense of feature extraction. So, in this chapter the main interest will be focused on lossy hyperspectral image compression.

The success of a lossy compression technique is generally measured by whether or not its effectiveness meets a preset desired goal which in turn determines which criterion should be used for compression. As an example, Principal Components Analysis (PCA) is a compression technique that represents data in a few principal components determined by data variances [9-10]. Its underlying assumption is based on the fact that the data are well-represented and structured in terms of variance, where most of data points are clustered and can be packed in a low dimensional space. Unfortunately, it was recently shown in [11-13] that Signal-to-Noise Ratio (SNR) was a better measure than data variance to measure image quality in multispectral imagery. Similarly, the Mean Squared Error (MSE) has been also widely used as a criterion for optimality in communications and signal processing such as quantization. However, it is also known that it may not be appropriate to be used as a measure of image interpretation. This is particularly true for hyperspectral imagery which can uncover many unknown signal sources, some of which may be very important in data analysis such as anomalies, small targets which generally contribute very little to SNR or MSE. In the PCA these targets may only be retained in minor components instead of principal components. So, preserving only the first few principal components may lose these targets. In SNR or MSE, such targets may very likely be suppressed by lossy compression if no extra care is taken since missing these targets may only cause inappreciable loss of signal energy or small error.

By realizing the importance of hyperspectral data compression, many efforts have been devoted to design and development of compression algorithms for hyperspectral imagery. Two major approaches have been studied. One is a direct extension of 2D image compression to 3D image

compression where many 2D image compression algorithms that have proven to be efficient and effective in 2D images are extended to 3D algorithms. Another is spectral/spatial compression which deals with spectral and spatial compression separately. While the former considers a hyperspectral image as an image cube as a whole, the latter performs spectral/spatial compression on a hyperspectral image with 1D compression on spectral information and 2D compression on spatial information. Despite a hyperspectral image can be considered as an image cube, a direct application of 3-D image compression to such a 1-D spectral/2-D spatial image cube may not be applicable in some cases as shown by examples. This is largely due to the fact that the spectral correlation of a hyperspectral image cube provides more crucial information than the spatial information in many exploitation-based applications. Therefore, an effective hyperspectral image compression technique must be able to explore and retain critical spectral information while the images are compressed spatially. This paper investigates these two approaches and provides evidence that 3D compression does not necessarily perform better than spectral/spatial compression in hyperspectral image compression from an exploitation point of view. In particular, using MSE or SNR as a compression criterion may result in significant loss of spectral information in data analysis. Additionally, in many cases, separating spectral and spatial compression may achieve better results in terms of preserving spectral information that is crucial in hyperspectral data exploitation. In order to demonstrate that it is indeed the case, this chapter studies various scenarios via a synthetic image simulated by a real HYperspectral Digital Image Collection Experiment (HYDICE) image to show that a simple spectral/spatial compression technique may perform as well as or even better than 3D lossy compression. Finally, we further develop several PCA-based spectral/spatial hyperspectral image compression techniques for hyperspectral image compression which are easy to implement, but yet achieve at least same results than a 3D lossy compression technique that is extended from its 2D compression counterpart. Experiments show that the proposed spectral/spatial hyperspectral image compression generally performs better than standard 3D hyperspectral image compression.

The remainder of this chapter is organized as follows. Section 2 reviews two well-known 2D image compression techniques, wavelet-based JPEG 2000 and set partitioning in hierarchical tree (SPIHT), and their extensions to 3D compression. Section 3 develops two principal components analysis (PCA)-based spectral/spatial compression techniques for hyperspectral image

compression, referred to as Inverse PCA (IPCA)/spatial compression and PCA/spatial compression. Section 4 demonstrates that 3D lossy compression does not necessarily perform better than the PCA-based spectral/spatial compression techniques in terms of mixed pixel classification via experiments. Section 5 conducts real image experiments for comparative analysis. Finally, Section 6 concludes with some remarks.

## 2. 3D COMPRESSION TECHNIQUES

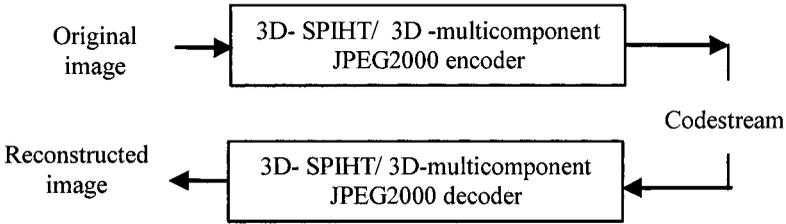
3D compression techniques are generally extended from their 2-D counterparts. Two techniques of particular interest that will be used in our investigation are the JPEG2000 Multicomponent which is an extension of the wavelet-based 2D-JPEG2000 and the 3D-SPIHT [14] which is extended by 2D-SPIHT developed by Said and Pearlman [15].

The JPEG2000 is a new still image compression standard that has replaced the commonly used DCT-based JPEG. It is a wavelet-based compression technique that adds/improves features such as coding of regions of interest, progressive coding, scalability etc. The entire coding can be divided into three stages: discrete wavelets transform (DWT), quantization and block coding. The 5/3 integer wavelet transform is used for reversible transformation and 9/7 floating point transform is used for irreversible transformation. After the 2D DWT, all wavelet coefficients are subjected to uniform scalar quantization employing a fixed dead-zone about the origin. One quantization step size is allowed for each subband. The quantized subbands are split further into partitions. A partition is formed by collections of blocks from different sub-bands. Each subband contributes one block to the partition. Partitions are split further into code-blocks. Block coding is then performed independently on each of the code-blocks. The used block coding is based on the principles of Embedded Block Coding with Optimized Truncation (EBCOT) [16-17]. For the case of hyperspectral imagery the Part II of JPEG2000 [18-19] is implemented to allow multi-component image compression which involves grouping of arbitrary subsets of components into component collections and applying point transforms along the spectral direction like wavelet transform.

Recently, an approach developed by Said and Pearlman [15], Set Partitioning in Hierarchical Trees (SPIHT) has become very popular. Two main features are introduced in the SPIHT algorithm. First, it utilizes a partial ordering of coefficients by magnitude and transmits the most

significant bits first. Second, the ordering data are not explicitly transmitted. The decoder running the same algorithm can trace the ordering information from the transmitted information. Xiong et al [14] later extended the 2D-SPIHT to 3D-SPIHT for video compression in a relatively straightforward manner. There is no constraint imposed on the SPIHT algorithm regarding the dimensionality of the data. If all pixels are lined up in decreasing order of magnitude, 3D-SPIHT performs exactly the same as does 2D-SPIHT. In the case of 3D subband structure, one can use a wavelet packet transform to allow a different number of decompositions between the spatial and spectral dimensions.

In summary, a flowchart to describe an application of 3D compression to a hyperspectral image can be depicted in block diagram 1.



**Block diagram 1:3D compression**

### **3. SPECTRAL/SPATIAL HYPERSPECTRAL IMAGE COMPRESSION**

Despite the fact that a hyperspectral image can be viewed as a 3D image cube, there are some major unique features that distinguish a hyperspectral image from commonly used 3D images such as videos. Unlike pure voxels considered in traditional 3D images, a hyperspectral image pixel vector is characterized by spectral properties such as subpixels and mixed pixels which are generally not encountered in pure voxels. Additionally, many material substances of interest that are present in hyperspectral imagery can be only explored by their spectral properties, not spatial properties such as combat vehicles embedded in single pixels and decoys in surveillance applications. Most importantly, many objects that are considered to be relatively small but yet provide significant information generally cannot be identified by prior knowledge, but can only be uncovered and revealed by their spectral properties such as chemical plumes or biological agents.

Therefore, whether or not a hyperspectral image compression technique is effective may not be necessarily determined by its spatial compression, but rather its spectral compression. Accordingly, performing spectral and spatial compression separately may be more desirable than compressing spectral and spatial information simultaneously. In this section, we develop several methods which implement the PCA to perform spectral compression in conjunction with 3D or 2D compression techniques to perform spatial compression.

### 3.1. Determination of Number of PCs to be Preserved

One of primary obstacles to implement PCA is to determine how many Principal Components (PCs) are significant for information preservation. In the past, the number of PCs is determined by the amount of signal energy calculated from data variances that correspond to eigenvalues. Unfortunately, it was shown in [20-21] that using the sum of eigenvalues as criterion to determine the number of PCs was not reliable and also incorrect in most cases in hyperspectral imagery. This is because subtle objects that contribute little eigenvalues may not be retained in the PCs. In order to mitigate this dilemma, the concept of Virtual Dimensionality (VD) was developed in [20-21] and further used to determine the number of spectrally distinct signatures. If we assume that each spectrally distinct signature is accommodated by a single PC, then the total number of PCs required to accommodate all the spectrally distinct signatures will be determined by the VD. One such method is the one developed by Harsanyi, Farrand and Chang in [22] (HFC) method.

The HFC method first calculates the sample correlation matrix,  $\mathbf{R}$ , and sample covariance matrix,  $\mathbf{K}$ , then finds the difference between their corresponding eigenvalues where  $L$  is the number of spectral channels. Let  $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \dots \geq \hat{\lambda}_L$  and  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_L$  be two sets of eigenvalues generated by  $\mathbf{R}$  and  $\mathbf{K}$ , called correlation eigenvalues and covariance eigenvalues, respectively. By assuming that signal sources are nonrandom unknown positive constants and noise is white with zero mean, we can expect that

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

and

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

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

$$\hat{\lambda}_l > \lambda_l > \sigma_{n_l}^2 \text{ for } l = 1, \dots, \text{VD}, \tag{3}$$

and

$$\hat{\lambda}_l = \lambda_l = \sigma_{n_l}^2 \text{ for } l = \text{VD} + 1, \dots, L. \tag{4}$$

where  $\sigma_{n_l}^2$  is the noise variance in the  $l$ -th spectral channel.

In order to determine the VD, Harsanyi et al. [22] formulated the VD determination problem as a binary hypothesis problem as follows.

$$\begin{aligned} H_0 : z_l &= \hat{\lambda}_l - \lambda_l = 0 \\ \text{versus} & & \text{for } l = 1, \dots, L \\ H_1 : z_l &= \hat{\lambda}_l - \lambda_l > 0 \end{aligned} \tag{5}$$

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

Despite the fact that the  $\hat{\lambda}_l$  and  $\lambda_l$  in Eqs. (1)-(4) are unknown constants, according to [23], we can model each pair of eigenvalues,  $\hat{\lambda}_l$  and  $\lambda_l$ , under hypotheses  $H_0$  and  $H_1$  as random variables by the asymptotic conditional probability densities given by

$$p_0(z_l) = p(z_l | H_0) \cong N(0; \sigma_{z_l}^2) \text{ for } l = 1, \dots, L \tag{6}$$

and

$$p_1(z_l) = p(z_l | H_1) \cong N(\mu_l; \sigma_{z_l}^2) \text{ for } l = 1, \dots, L \tag{7}$$

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

$$\sigma_{z_l}^2 = \text{var}[\hat{\lambda}_l - \lambda_l] = \text{var}[\hat{\lambda}_l] + \text{var}[\lambda_l] - 2 \text{cov}(\hat{\lambda}_l, \lambda_l) \text{ for } l = 1, \dots, L. \quad (8)$$

It is shown in [23] that when the total number of samples,  $N$  is sufficiently large,  $\text{var}[\hat{\lambda}_l] \cong \frac{2\hat{\lambda}_l^2}{N}$  and  $\text{var}[\lambda_l] \cong \frac{2\lambda_l^2}{N}$ . Therefore, the noise variance  $\sigma_{z_l}^2$  in Eq. (7) can be estimated and approximated using Eq. (8).

Now, we use Schwarz's inequality to bound  $\text{cov}(\hat{\lambda}_l, \lambda_l)$  in Eq. (8) as follows

$$\text{cov}(\hat{\lambda}_l, \lambda_l) \leq \sqrt{\text{var}[\hat{\lambda}_l] + \text{var}[\lambda_l]} \cong \frac{2}{N} (\hat{\lambda}_l \lambda_l) \quad (9)$$

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

$$\sigma_{z_l}^2 = \text{var}[\hat{\lambda}_l] + \text{var}[\lambda_l] \approx \frac{2\hat{\lambda}_l^2}{N} + \frac{2\lambda_l^2}{N} \rightarrow 0 \text{ for } l = 1, \dots, L \text{ as } N \rightarrow \infty. \quad (10)$$

From Eqs. (6), (7) and (10), we define the false alarm probability and detection power (i.e., detection probability) as follows:

$$P_F = \int_{\tau_l}^{\infty} p_0(z) dz \quad (11)$$

$$P_D = \int_{\tau_l}^{\infty} p_1(z) dz. \quad (12)$$

A Neyman-Pearson detector for  $\hat{\lambda}_l - \lambda_l$ , denoted by  $\delta_{NP}(\hat{\lambda}_l - \lambda_l)$  for the binary composite hypothesis testing problem specified by Eq. (5) can be obtained by maximizing the detection power  $P_D$  in Eq. (12), while the false

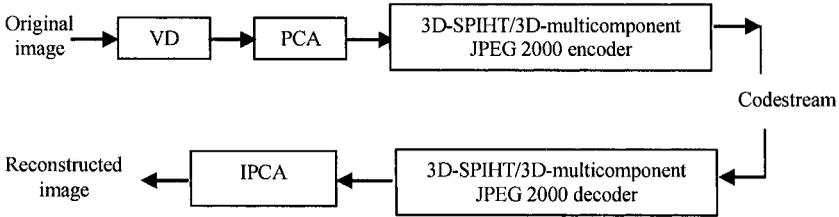
alarm probability  $P_F$  in Eq. (11) is fixed at a specific given value, which determines the threshold value  $\tau_l$  in Eqs. (11)-(12). So a case of  $\hat{\lambda}_l - \lambda_l > \tau_l$  indicating that  $\delta_{NP}(\hat{\lambda}_l - \lambda_l)$  fails the test, in which case there is signal energy assumed to contribute to the eigenvalue,  $\hat{\lambda}_l$ , in the  $l$ -th data dimension. It should be noted that the test for Eq. (5) must be performed for each of  $L$  spectral dimensions. Therefore, for each pair of  $\hat{\lambda}_l - \lambda_l$ , the threshold  $\tau$  is different and should be  $l$ -dependent, that is  $\tau_l$ .

### 3.2. PCA-Spectral/Spatial Compression

Using the PCA to de-correlate spectral information between bands has been a common practice in data compression [24]. However, it has been a long standing problem with how to determine the number of principal components (PCs) for information preservation. This section proposes a new idea to resolve this issue. It uses the concept of the VD to determine the number of PCs required for compression. Since the PCs resulting from the PCA are spectrally de-correlated, we can compress these VD-determined PCs in three different ways described in the following subsections.

#### 3.2.1. Inverse PCA (IPCA)/3D Compression

One straightforward approach is to use PCA in conjunction with a 3D compression technique. More specifically, the PCA is first used to de-correlate a hyperspectral image for spectral compression. Then the VD determines how many PCs must be retained for compression. Then a 3D lossy compression technique is applied to an image cube formed by VD-determined PC images for further spatial compression. Finally, an inverse PCA is implemented to reconstruct a 3D image with the same number of spectral bands as the original image. This approach is referred to as Inverse PCA (IPCA)/3D compression. A flow chart of the IPCA/3D compression is depicted in Block diagram 2.



**Block diagram 2: IPCA/3D Compression**

The details of its implementation can be summarized as follows.

#### *IPCA/3D Compression Algorithm*

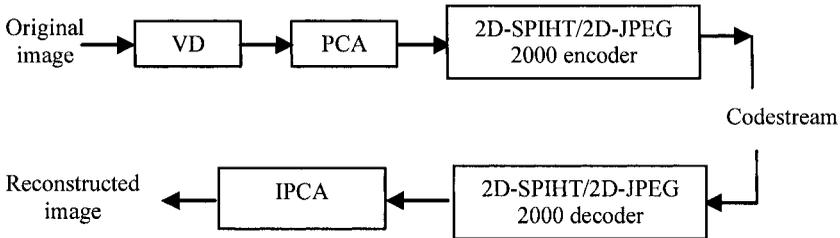
1. Determine the VD of an  $L$ -band hyperspectral image,  $p$ .
2. Form the first  $p$  PCA-generated PC images as a 3D image cube, referred to as 3D  $p$ -PC image cube.
3. Use a 3D compression technique to the 3D  $p$ -PC image cube.
4. Apply IPCA to the 3D image obtained in step 3 to reconstruct a 3D  $L$ -band image cube.
5. Exploit the resulting compressed 3D  $L$ -band image cube for various applications.

Depending upon which 3D compression is used, two IPCA/3D compression algorithms can be implemented, IPCA/3D-SPIHT and IPCA/3D-multicomponent JPEG2000. However, since 3D SPIHT requires dimensions to be multiples of  $2^{n+1}$  with  $n$  being the number of levels in wavelet decomposition, the IPCA/3D SPIHT may not be applicable when the number of spectral bands does not meet this constraint.

#### 3.2.2. Inverse PCA (IPCA)/2D Compression

In addition to PCA/3D compression as described above, a second approach is to implement the PCA for spectral de-correlation in conjunction with 2D spatial compression [25-26]. More precisely, instead of applying 3D compression directly to a PCs-formed image cube as is done in IPCA/3D compression, the proposed PCA/2D compression applies a 2D compression technique to each of the VD-determined PCs for spatial compression. Then an inverse PCA is applied to an image cube formed by the VD-determined 2D compressed images to reconstruct a 3D image with the same number of

spectral bands as the original image. Such an approach is referred to as Inverse PCA (IPCA)/2D compression. A flowchart of the IPCA/2D compression is delineated in block diagram 3.



**Block diagram 3: IPCA/2D Compression**

The detailed implementation of the IPCA/2D compression is described as follows.

#### *IPCA/2D Compression Algorithm*

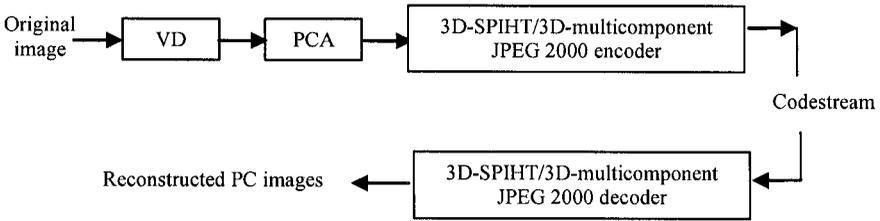
1. Determine the VD of an  $L$ -band multispectral/hyperspectral image.
2. Use a 2D image compression technique to each of the first  $p$  PC images obtained in step 1.
3. Form a 3D image cube by the  $p$  compressed PC image.
4. Apply IPCA to the 3D image obtained in step 3 to reconstruct a 3D  $L$ -band image cube.
5. Exploit the resulting compressed 3D  $L$ -band image cube for various applications.

In analogy with the IPCA/3D compression, two IPCA/2D compression algorithms can be also implemented, referred to as IPCA/2D-SPIHT and IPCA/2D-JPEG2000.

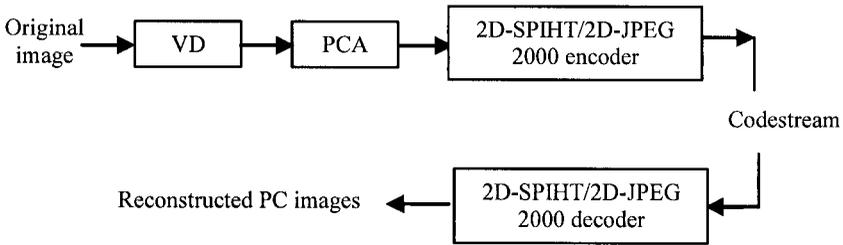
#### 3.2.3. PCA/2D Compression

The IPCA/3D compression and IPCA/2D compression described previously require an inverse PCA to reconstruct a 3D image from the compressed PC images so that the resulting 3D image has the same number of spectral bands as the original image does. In fact, it may not be necessary to do so in many

applications. As an alternative, we can simply de-compress the compressed PC images for image analysis. In this case, the last step of processing IPCA, i.e., step 4 in the IPCA/3D compression and the IPCA/2D compression can be skipped. As a result, the IPCA/3D compression and the IPCA/2D compression are reduced to PCA/3D compression and the PCA/2D compression respectively where block diagrams 2 and 3 become block diagrams 4 and 5 as depicted below.



**Block diagram 4: PCA/3D Compression**



**Block diagram 5: PCA/2D Compression**

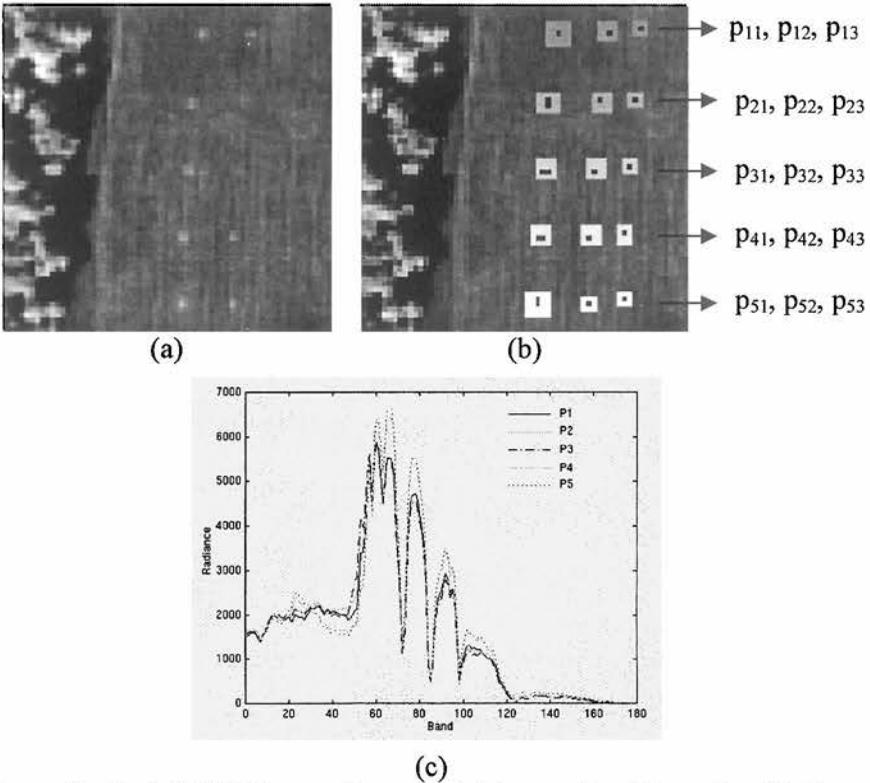
In analogy with IPCA/3D compression and IPCA/2D compression, two PCA/3D compression and two PCA/2D compression algorithms can be also implemented, referred to as PCA/3D-SPIHT, PCA/3D-multicomponent JPEG2000, PCA/2D-SPHIT and PCA/2D-JPEG2000 respectively.

#### 4. COMPARATIVE ANALYSIS VIA SYNTHETIC IMAGES

In this section, computer simulations via a synthetic image are presented to demonstrate that a direct application of 3-D data compression to a hyperspectral image without extra care may result in significant loss of information. Furthermore, it may not perform as well as compression of spectral and spatial separately. Additionally, the same example will also

show that the commonly used variance-based measure, i.e., sum of eigenvalues to determine the number of PCs required to be preserved for spectral compression is generally not an adequate criterion. Instead, the proposed VD is more appropriate and effective.

The experiments were based on a synthetic image that was simulated from a HYperspectral Digital Image Collection Experiment (HYDICE) image scene in [21]. It is an image scene shown in Fig. 1(a), which has a size of  $64 \times 64$  pixel vectors with 15 panels in the scene and the ground truth map in Fig. 1(b).



**Figure 1:** (a) A HYDICE panel scene which contains 15 panels; (b) Ground truth map of spatial locations of the 15 panels; (c) five panel signatures  $\{P_i\}_{i=1}^5$

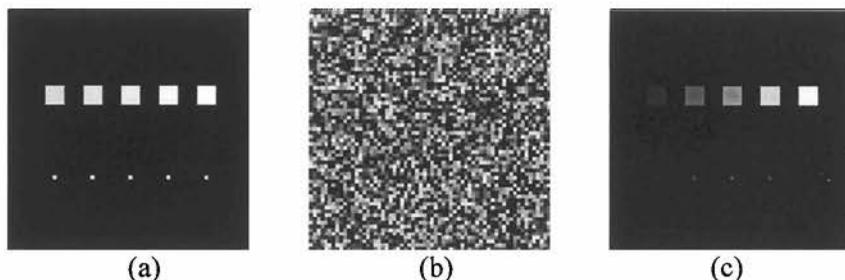
It was acquired by 210 spectral bands with a spectral coverage from  $0.4\mu\text{m}$  to  $2.5\mu\text{m}$ . Low signal/high noise bands: bands 1-3 and bands 202-

210; and water vapor absorption bands: bands 101-112 and bands 137-153 were removed. So, a total of 169 bands were used. The spatial resolution is 1.56m and the spectral resolution is 10nm. Within the scene in Fig. 1(a) there is a large grass field background, and a forest on the left edge. Each element in this matrix is a square panel and denoted by  $p_{ij}$  with rows indexed by  $i$  and columns indexed by  $j$ . For each row  $i = 1, 2, \dots, 5$ , there are three panels  $p_{i1}$ ,  $p_{i2}$ ,  $p_{i3}$ , painted by the same material but with three different sizes. For each column  $j = 1, 2, 3$ , the 5 panels  $p_{1j}$ ,  $p_{2j}$ ,  $p_{3j}$ ,  $p_{4j}$ ,  $p_{5j}$  have the same size but with five different materials. Nevertheless, they were still considered as different materials. The sizes of the panels in the first, second and third columns are  $3m \times 3m$ ,  $2m \times 2m$  and  $1m \times 1m$  respectively. Since the size of the panels in the third column is  $1m \times 1m$ , they cannot be seen visually from Fig. 1(a) due to the fact that its size is less than the 1.56m spatial resolution. Fig. 1(b) shows the precise spatial locations of these 15 panels where red pixels (R pixels) are the panel center pixels and the pixels in yellow (Y pixels) are panel pixels mixed with the background. The 1.56m-spatial resolution of the image scene suggests that most of the 15 panels are one pixel in size except that  $p_{21}$ ,  $p_{31}$ ,  $p_{41}$ ,  $p_{51}$  which are two-pixel panels. Fig. 1(c) plots the 5 panel spectral signatures  $\{P_i\}_{i=1}^5$  with  $P_i$  obtained by averaging R pixels in the  $3m \times 3m$  and  $2m \times 2m$  panels in row  $i$  in Fig. 1(b).

It should be noted the R pixels in the  $1m \times 1m$  panels are included because they are not pure pixels due to that fact that the spatial resolution of the R pixels in the  $1m \times 1m$  panels is 1 m smaller than the pixel resolution is 1.56 m. These panel signatures along with the R pixels in the  $3m \times 3m$  and  $2m \times 2m$  panels were used as required prior target knowledge for the following comparative studies.

The synthetic image to be used for our experiments was similar to the real scene and has size of  $64 \times 64$  pixel vectors. There are 10 panels located at its center and arranged in five rows with five panels in each row. The five panels in the 1<sup>st</sup> row have the same size of  $5 \times 5$  pixel vectors and simulated by  $P_1, P_2, P_3, P_4, P_5$  in Fig. 1(c) respectively where pixel vectors in the same panel were simulated the same panel signatures. The five panels in the 2<sup>nd</sup> row are all single-pixel panels, denoted by  $p_{21}, p_{22}, p_{23}, p_{24}$ , and  $p_{25}$  with abundance fractions 100%, 80%, 60%, 40% and 20% by  $P_1, P_2, P_3, P_4, P_5$  respectively. In other words, the panels in the  $j$ -th column were all simulated by the  $j$ -th panel signature  $P_j$ . The background in the synthetic image was simulated by a grass field signature  $\mathbf{b}$  in the image scene in Fig. 1 with an added Gaussian noise to achieve signal-to-noise ratio (SNR) 30:1 as defined

in [29]. Fig. 2(a-b) shows the 10 simulated panels and the background image. Fig. 2(c) is a synthetic image obtained by implanting the 10 simulated panels in Fig. 2(a) in the background image in Fig. 2(b) where their corresponding background pixels were removed to accommodate the implanted panel pixels. It should be noted that the noise background in Fig. 2(c) has been visually suppressed because of high intensity gray level values of panel pixels.



**Figure 2:** A 10-panel synthetic image

Here two major compression techniques 3D-SPIHT and 3D-multicomponent JPEG2000 will be implemented and compared to our proposed PCA based techniques. For the 1D wavelet transform used in 3D-multicomponent JPEG2000 the QccPack [27] was implemented. It should be noted that the 3D-SPIHT algorithm requires all dimensions to be multiples of  $2^{n+1}$ , where  $n$  is the number of decomposition levels. Since the HYDICE image has 169 bands (after removal of water absorption and noisy bands), it does not meet this requirement. Therefore, the last 9 bands of the HYDICE image were removed to accommodate its implementation. For 3D-SPIHT, 4-level spectral and 5-level spatial wavelet packet decomposition was performed. In particular, the wavelet packet decomposition instead of the dyadic decomposition was used because it could be better tailored to the data and the wavelet packet decomposition is considered to be more suitable for hyperspectral images.

The JPEG2000 is an embedded wavelet-based coder that supports the coding of hyperspectral images in Part II of the standard. This standard specifies arbitrary decorrelating transforms in the spectral direction, thus permitting implementation of wavelet transforms. The currently available implementations of the JPEG2000 standard do not yet implement the Part II of the standard. The JPEG2000 implementation used for our experiments was the Kakadu version 4.2.1 [28]. Since the Kakadu software implements only

Part I of the JPEG2000 standard, a 1-D spectral transform was performed separately on the data before feeding the data to the Kakadu coder. Additionally, the popular Cohen-Daubechies-Feauveau 9-7 wavelet with symmetric extension was used. In our experiments a 4-level wavelet decomposition in the spectral domain was first performed and followed by the Kakadu coder.

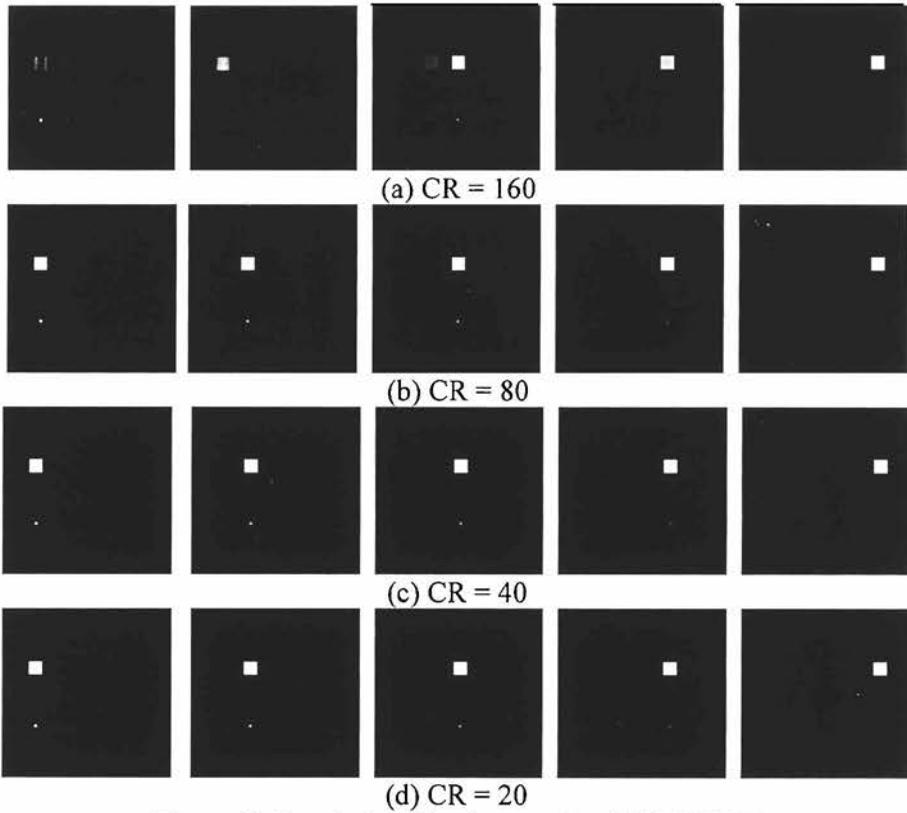
The compression obtained from the PCA-based methods was through two stages, PCA/VD spectral compression and the 2D-JPEG2000 or 3D-multicomponent JPEG2000 spatial compression. Since the compression obtained by the PCA is fixed for a given hyperspectral image, a variable bit-rate lossy compression technique was applied in the 2D-JPEG2000/3D-multicomponent JPEG2000 stage.

Four scenarios were conducted to evaluate performance of 3D compression and our proposed PCA/spatial compression algorithms.

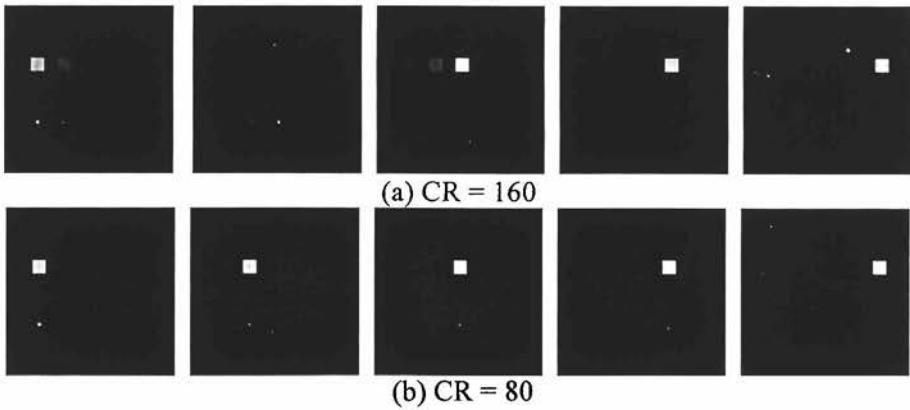
### **Scenario 1:** Classification based on 3D de-compressed image cubes

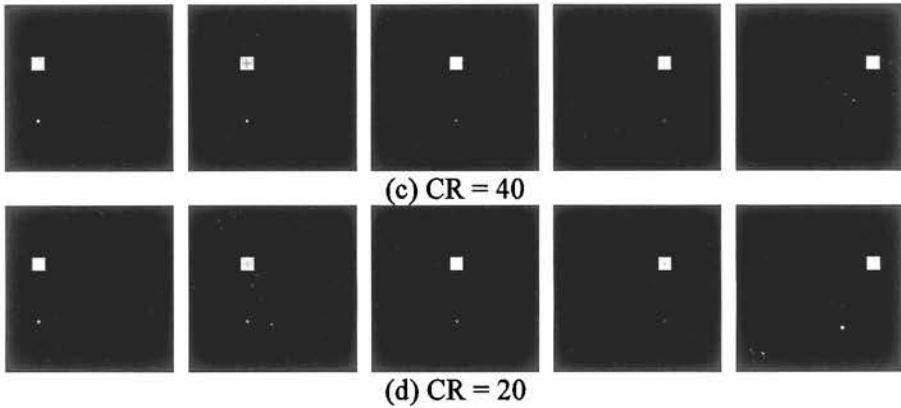
In this scenario, two 3D compression techniques, 3D-SPIHT and 3D multicomponent JPEG2000 described in block diagram 1 were directly applied to the 10-panel image in Fig. 2(c) with compression ratio (CR) equal to 20, 40, 80 and 160. Then such compressed images were de-compressed to reconstruct the original image and used for unsupervised mixed pixel classification where the unsupervised fully constrained least squares (UFCLS) mixed pixel classification method in [21,29,30] was used to classify the 10 panels where among the first six signatures generated by the UFCLS, five of them were found to correspond to the five panel signatures. It should be noted that the UFCLS must be unsupervised since there is no prior knowledge available during compression. It was performed on de-compressed images. Figs. 3 and 4 show their respective classification results with compression ratio (CR) equal to 20, 40, 80 and 160.

As shown in Figs. 3-4, both 3D-SPIHT and 3D-multicomponent JPEG2000 performed very similarly for all cases where there was no visible difference among CR = 20, 40 and 80. Nevertheless, the two 3D algorithms performed very poorly for the case of CR = 160.



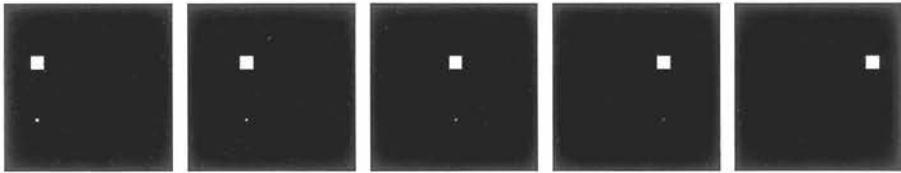
**Figure 3:** Panel classification results of 3D-SPIHT





**Figure 4:** Panel classification results of JPEG2000 Multicomponent

In order to make comparison, Fig. 5 shows the classification results produced by the UFCLS based on the original un-compressed image in Fig. 2(c).



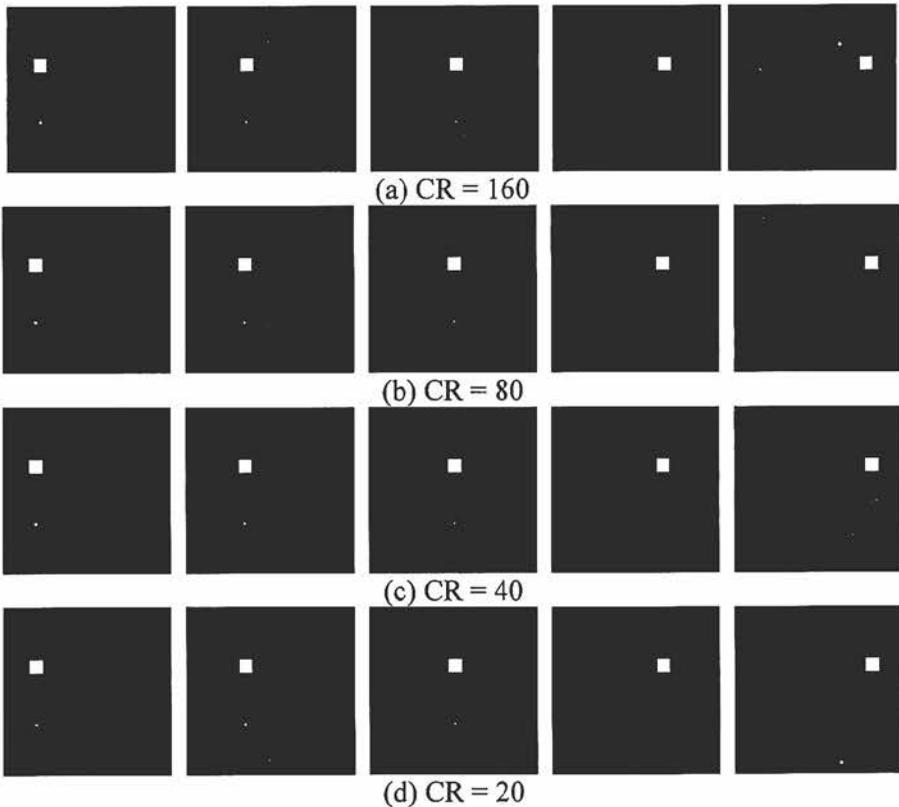
**Figure 5:** Panel classification results produced by the UFCLS on un-compressed synthetic image

Comparing Fig. 5 to Figs. 3-4, the results in Fig. 3(b-d) and Fig. 4(b-d) produced by CR = 80, 40 and 20 was very comparable to those in Fig. 5.

**Scenario 2:** Classification based on 3D images reconstructed by applying inverse PCA to 3D compressed image cubes

Another scenario is to apply the IPCA/3D lossy compression to the image in Fig. 2(c) where the VD was used to determine how many principal components (PCs) needed to be preserved for compression. For the synthetic image in Fig. 2(c), the VD was estimated to be 6. So, the first six PCs were preserved and formed as a 3D image cube so that a lossy 3D compression technique was applied to compress the 6-PC formed image cube to achieve further compression. Since the spectral dimensions are smaller than 8, the 3D-SPIHT is not applicable to this scenario. Therefore, only 3D

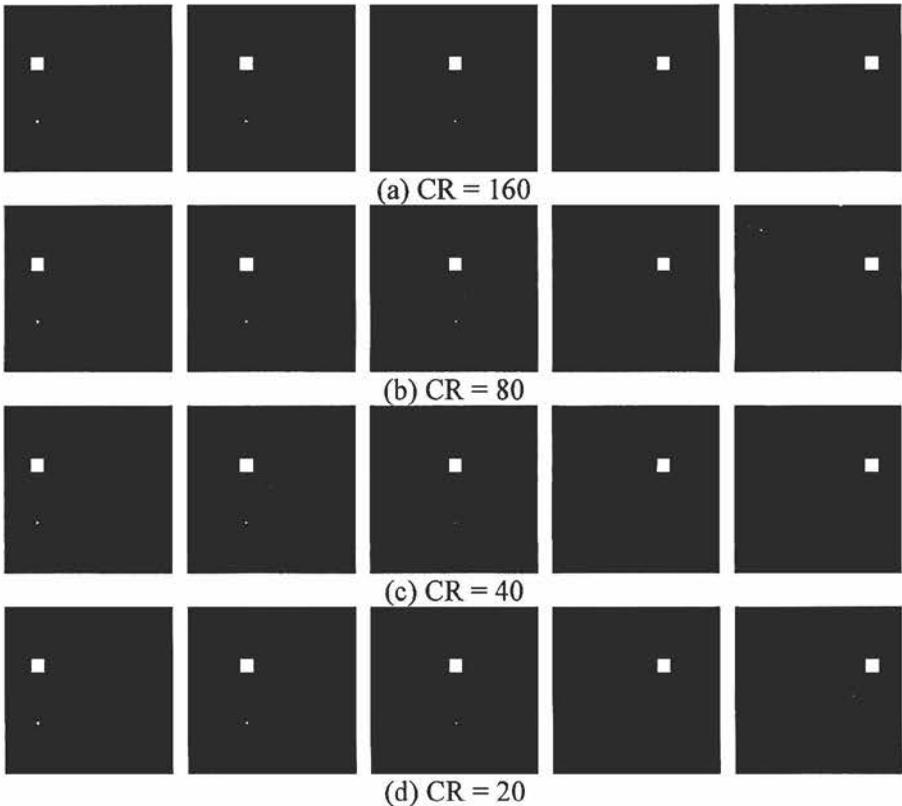
multicomponent JPEG2000 was applied to this scenario. Finally, an inverse PCA (IPCA) was then applied to reconstruct an image with 160 bands that is a compressed image of the original image in Fig. 2(c). This approach is called IPCA/3D-multicomponent JPEG2000 in Block diagram 2. The resulting IPCA/3D-multicomponent JPEG2000 compressed image was then used for the UFCLS classification. Since only five out of the first six PCs contained panels, only the five classification images are provided to show the classification of panels. Fig. 6 shows the classification results produced by IPCA/3D-multicomponent JPEG2000 were very close to those in Fig. 5 produced by the UFCLS operating on the original un-compressed image. Additionally, the IPCA/3D-multicomponent JPEG2000 also significantly improved those results obtained for CR = 160 in Figs. 3-4.



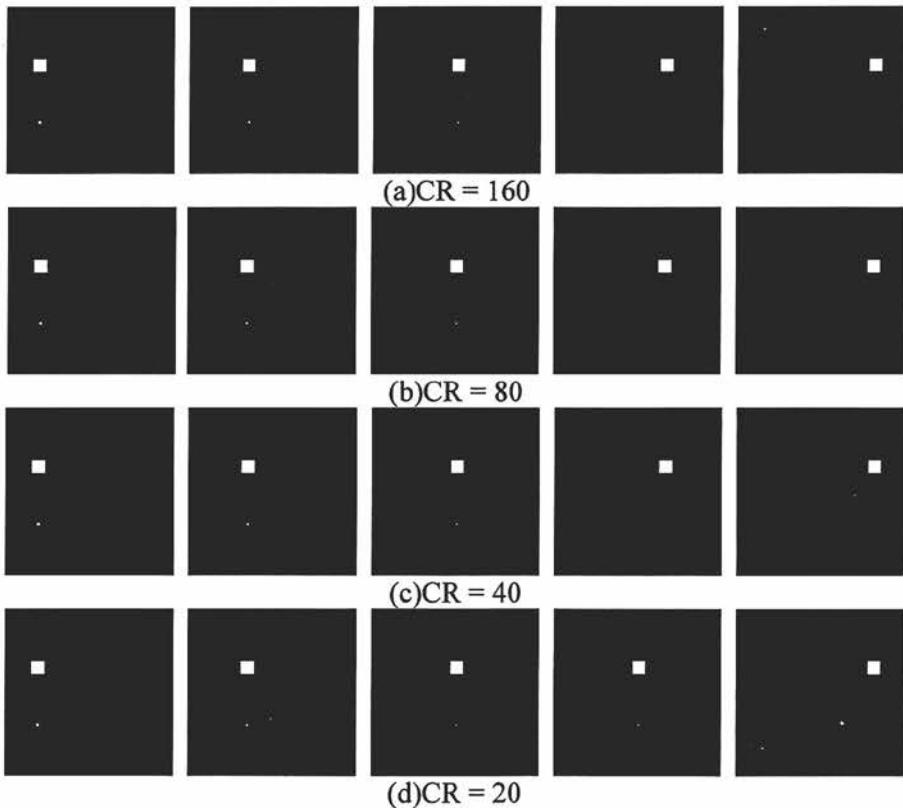
**Figure 6:** Panel classification results of IPCA/3D multicomponent-JPEG2000

**Scenario 3:** Classification based 3D images reconstructed by applying IPCA to 3D image cubes formed by a set of 2D compressed PC images

A third scenario was to apply the IPCA/2D-SPIHT and IPCA/2D-JPEG2000 lossy compression techniques to compress individually and separately the six PCs. Then an inverse PCA was applied to an image cube that was formed by the six resulting 2D compressed PCs to reconstruct a de-compressed image with 160 bands for unsupervised mixed pixel classification where the UFCLS method used for Figs. 3-6 was also applied here. The classification results of the IPCA/2D-SPIHT and the IPCA/2D-JPEG2000 for panels are shown in Figs. 7 and 8 respectively.



**Figure 7:** Panel classification results of IPCA/2D-SPIHT



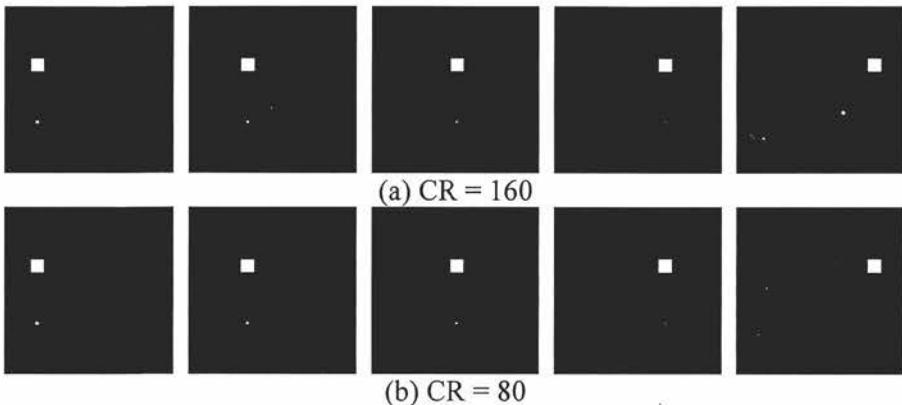
**Figure 8:** Panel classification results of IPCA/2D-JPEG2000

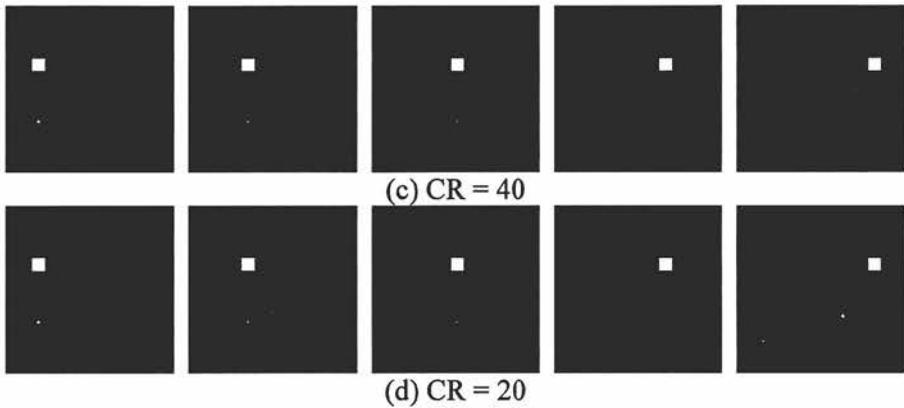
Similar to Fig. 6, both the IPCA/2D-SPIHT and the IPCA/2D-JPEG2000 performed well on the case of CR = 160 and their results were very comparable to the IPCA/3D multicomponent-JPEG2000 in Fig. 6. Comparing Figs. 7-8 to Figs. 3-4, the most visible and significant improvement was the case of CR = 160 where the PCA/2D-JPEG2000 and PCA/3D lossy compression performed as well as the cases of CR = 20, 40 and 80.

**Scenario 4:** Classification based image cubes formed by a set of 2D compressed PC images

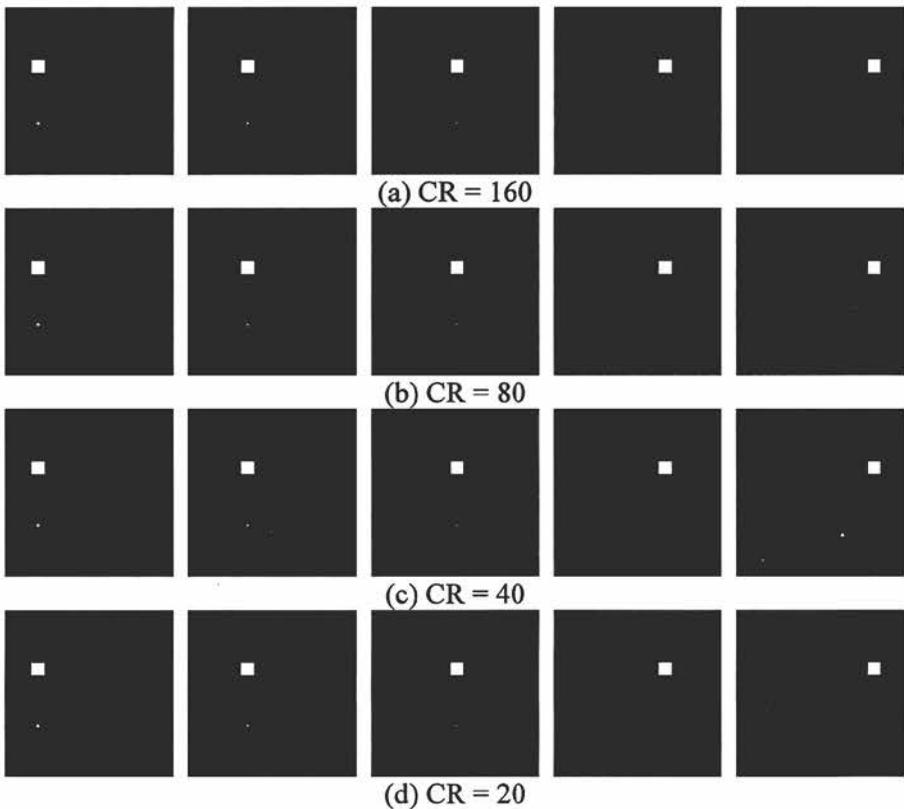
This scenario is different from all three scenarios described above. It applied the PCA/3D-multicomponent JPEG2000 in Block diagram 4 where

the 3D-multicomponent JPEG2000 was performed in the same way that it was performed in Scenario 1. The resulting compressed image cube was then used for the UFCLS classification. The only difference is that the 3D-multicomponent JPEG2000 in Scenario 4 was applied to a 6-PC image cube while the 3D multicomponent JPEG2000 in Scenario 1 was applied to the original 160-band image. Also note that there was no PCA/3D-SPIHT because the number of PCs, 6 is smaller than 8. Another was to use the PCA/2D-SPIHT and PCA/2D-JPEG2000 described in Block diagram 5 to compress PC images to form a 3D image cube with 6 spectral dimensions. The resulting 6-PC image cube was then used for UFCLS classification. In this case, the 2D-SPHIT and 2D-JPEG2000 were applied to compress the six PC images in the same way that they were performed in Scenario 3. A major advantage of this approach is reduction of computation complexity by skipping the last process in all the above three scenarios, which is to reconstruct an image cube with the same number of bands as the original image. Figs. 9-11 show the panel classification results obtained by the PCA/3D-multicomponent JPEG2000, the PCA/2D-SPHIT and the PCA/2D-JPEG2000 respectively where it is very difficult to visually compare the classification results among Figs. 9-11. In particular, the five subpixel panels are single-pixel panels and their detection in abundance fractions could be easily suppressed by the detected abundance fractions of their corresponding  $5 \times 5$  large panels.

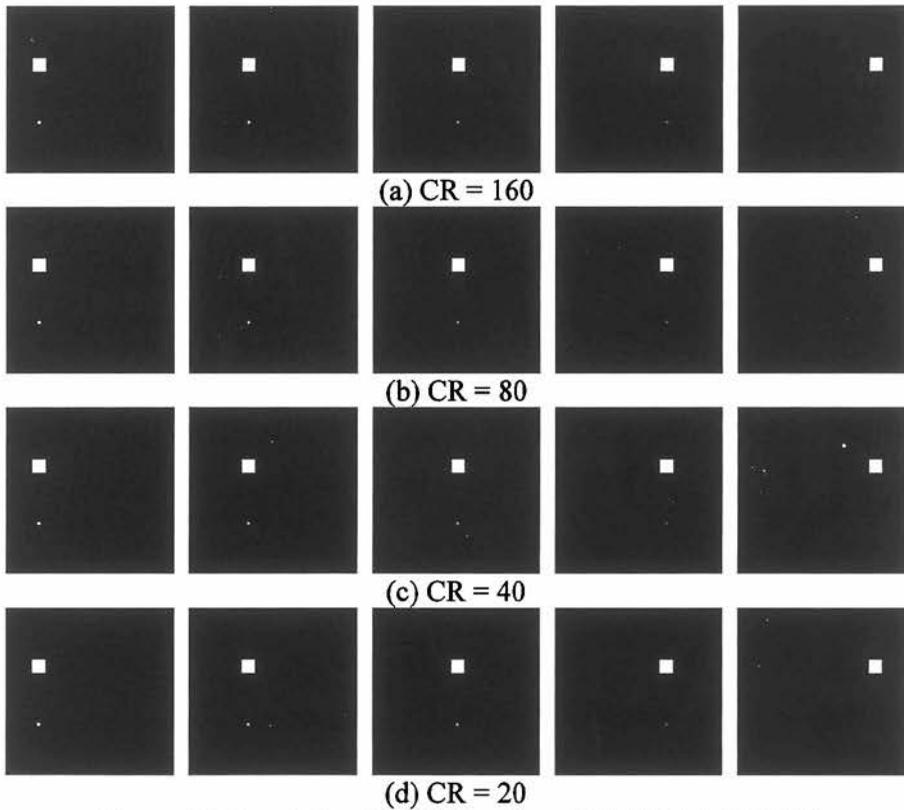




**Figure 9:** Panel classification results of PCA/ 3D multicomponent-JPEG2000



**Figure 10:** Panel classification results of PCA/2D-SPIHT



**Figure 11:** Panel classification results of PCA/2D-JPEG2000

In order to further demonstrate how much abundance fraction of the five subpixel panels in the 2<sup>nd</sup> row in the scene were detected in Figs. 3-11 by all the eight lossy compression techniques evaluated in this chapter, 3D-SPIHT, 3D-multicomponent JPEG2000, IPCA/3D-multicomponent JPEG2000, IPCA/2D-SPHIT, IPCA/2D-JPEG2000, PCA/3D-multicomponent JPEG2000, PCA/2D-SPIHT and PCA/2D-JPEG2000, Table 1 tabulates their quantitative results along with the quantification results obtained by applying the UFCLS to original uncompressed image for comparison. For simplicity, we have used numerals 1-8 to represent the results produced by (1) 3D-SPIHT, (2) 3D-multicomponent JPEG2000, (3) IPCA/3D multicomponent-JPEG2000, (4) IPCA/2D-SPHIT, (5) IPCA/2D-JPEG2000, (6) PCA/3D multicomponent JPEG2000, (7) PCA/2D-SPIHT, (8) PCA/2D-JPEG2000

and numeral 9 to represent the results produced by the UFCLS on the original uncompressed image respectively.

CR	Subpixel panels	1	2	3	4	5	6	7	8	9
160	p <sub>21</sub>	100	100	98	98	97	98	98	97	100
	p <sub>22</sub>	0	38	71	74	75	71	74	75	75
	p <sub>23</sub>	49	0	59	58	59	59	59	59	59
	p <sub>24</sub>	22	29	39	39	40	39	39	40	39
	p <sub>25</sub>	17	0	10	15	13	11	15	12	14
80	p <sub>21</sub>	92	100	100	100	100	100	100	100	100
	p <sub>22</sub>	73	62	75	75	74	74	75	74	75
	p <sub>23</sub>	56	52	59	59	59	59	59	59	59
	p <sub>24</sub>	34	30	39	39	39	34	34	34	39
	p <sub>25</sub>	19	18	15	15	15	15	15	15	14
40	p <sub>21</sub>	94	100	100	100	100	100	100	100	100
	p <sub>22</sub>	73	69	74	74	74	74	74	74	75
	p <sub>23</sub>	57	56	59	59	59	59	59	59	59
	p <sub>24</sub>	37	38	39	39	39	34	34	34	39
	p <sub>25</sub>	18	19	15	15	15	15	15	15	14
20	p <sub>21</sub>	98	100	100	100	100	100	100	100	100
	p <sub>22</sub>	74	70	74	74	74	74	74	74	75
	p <sub>23</sub>	59	58	59	59	59	59	59	59	59
	p <sub>24</sub>	38	36	39	39	39	34	34	34	39
	p <sub>25</sub>	19	19	15	15	15	15	15	15	14

**Table 1:** Abundance fractions of the 5 subpixel panels in the 2<sup>nd</sup> row

Finally, we tabulate SNR and MSE for each CR in Table 2 for Figs. 3-4 and Figs. 6-11.

CR	160		80		40		20	
	SNR	MSE	SNR	MSE	SNR	MSE	SNR	MSE
1	42.50	37017	43.16	31809	43.76	27734	44.91	21255
2	41.66	44967	42.96	33288	43.53	29235	44.61	22787
3	43.38	30240	43.46	29675	43.47	29659	43.47	29659
4	43.38	30215	43.46	29675	43.47	29654	43.47	29656
5	43.33	30619	43.46	29706	43.46	29690	43.46	29690
6	33.67	587	47.36	25	54.14	5	54.14	5
7	33.85	563	47.61	24	56.67	3	56.94	3
8	31.49	969	43.91	56	45.22	41	45.22	41

**Table 2:** SNR and MSE for Figs. 3-4 and Figs. 6-11

In Table 2, the same numerals used in Table 1 were used for consistency where (1) 3D-SPIHT, (2) 3D multicomponent-JPEG2000, (3) IPCA/3D multicomponent-JPEG2000, (4) IPCA/2D-SPHIT, (5) IPCA/2D-JPEG2000, (6) PCA/3D multicomponent JPEG2000, (7) PCA/2D-SPIHT and (8) PCA/2D-JPEG2000. Apparently, (6) the PCA/3D-multicomponent JPEG2000, (7) PCA/2D-SPIHT and (8) PCA/2D-JPEG2000 produced much smaller MSEs. Additionally, since (6) the PCA/3D-multicomponent JPEG2000, (7) PCA/2D-SPIHT and (8) PCA/2D-JPEG2000 operated on only PC images not on de-compressed images with the original number of bands, their produced SNRs were generally lower than expected.

In the previous experiments, the number of PCs was determined by the VD which was 6. In order to demonstrate the advantage of using the VD, a general and widely used approach to determination of the number of PCs was considered, which calculates the sum of the largest eigenvalues to measure how much percentage of energy is required to be preserved during compression. Table 3 tabulates the energy contributed by the first 33 PCs where the energy was calculated by summing all eigenvalues of the PCs up to the PC being considered and normalizing the sum to 100%.

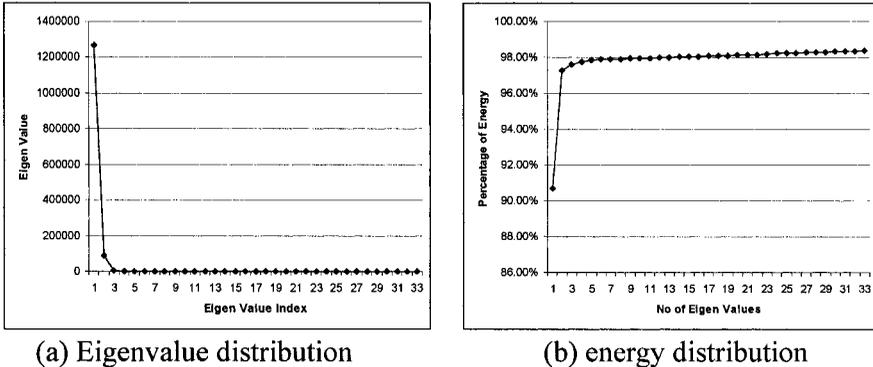
PC	Eigenvalue	energy	PC	eigenvalue	percent	PC	eigenvalue	energy
1	1267555.526	90.70%	12	260.5764	97.99%	23	243.1267	98.21%
2	91517.509	97.25%	13	259.0156	98.01%	24	241.7741	98.22%
3	5201.5112	97.63%	14	257.0252	98.03%	25	240.2581	98.24%
4	2044.3714	97.77%	15	256.3777	98.05%	26	239.7718	98.26%
5	1147.3887	97.86%	16	255.104	98.06%	27	238.8506	98.28%
6	273.5699	97.88%	17	253.4133	98.08%	28	237.9477	98.29%
7	271.5065	97.89%	18	252.4769	98.10%	29	237.5739	98.31%
8	269.7012	97.91%	19	251.3845	98.12%	30	236.3263	98.33%
9	268.3106	97.93%	20	249.2878	98.14%	31	234.688	98.34%
10	265.0195	97.95%	21	247.9757	98.15%	32	233.8727	98.36%
11	262.3911	97.97%	22	245.5467	98.17%	33	232.3503	98.38%

**Table 3:** Energy contributed by eigenvalues in the first 33 PCs

According to Table 3, Fig. 12(a-b) also plots the eigenvalue distribution of the first 33 PCs and their energy distribution respectively.

As shown in Table 3 and Fig. 12, there was a sudden and drastic drop between 2 and 3. Therefore, an obvious choice for the number of PCs was 2. Of course, it was not correct because there were at least five distinct panel signatures in the synthetic image scene. This fact implied that using sum of eigenvalues to calculate signature energies as a criterion to determine the

number of PCs was not a good measure. Similar observations were also witnessed in [20-21].



**Figure 12:** Eigenvalue distribution and energy distribution of the first 33 PCs

In order to see whether there was a change in the VD estimate for lossy compressed images, Table 4(a-d) also tabulates the results of the VD estimated with for various false alarm probabilities by the HFC method based on the de-compressed images with CR = 160, 80, 40, and 20 produced by (1)3D-SPIHT, (2)3D-multicomponent-JPEG2000, (3)IPCA/3D multicomponent-JPEG2000, (4)IPCA/2D-SPHIT, (5)IPCA/2D-JPEG2000.

$P_F$	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$
1	26	17	12	12	11
2	25	16	14	13	11
3	10	8	8	8	7
4	10	8	8	7	7
5	10	9	7	7	7

**Table 4(a):** Estimated VD for compressed image with CR = 160 by the HFC method with various false alarm probabilities

$P_F$	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$
1	13	8	8	7	7
2	21	16	11	8	8
3	9	8	7	7	7
4	10	8	7	7	7
5	10	7	7	7	7

**Table 4(b):** Estimated VD for compressed image with CR = 80 by the HFC method with various false alarm probabilities

$P_F$	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$
1	8	7	7	7	7
2	11	9	9	9	8
3	11	8	8	7	7
4	9	8	7	7	7
5	10	9	7	7	7

**Table 4(c):** Estimated VD for compressed image with CR = 40 by the HFC method with various false alarm probabilities

$P_F$	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$
1	7	7	7	7	7
2	7	7	7	7	7
3	11	8	8	7	7
4	11	8	7	7	7
5	11	9	7	7	7

**Table 4(d):** Estimated VD for compressed image with CR = 20 by the HFC method with various false alarm probabilities

It is clearly shown in Table 4(a-d) that the VD estimates were more consistent when the CR was low. It should be also noted that the PCA/3D compression and PCA/2D compression were not applicable since they did not apply IPCA to recover the PCA-compressed image in the original image space with 160 spectral bands.

### 5. REAL HYPERSPECTRAL IMAGE EXPERIMENTS

In this section, the 15-panel HYDICE image scene in Fig. 1 was used for experiments. The VD was estimated by the HFC method is tabulated in Table 5 where the VD was empirically selected to be 9.

$P_F$	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$
HYDICE	14	11	9	9	7

**Table 5:** VD estimates for the HYDICE image by the HFC method

Fig. 13 shows the panel classification results of the UFCLS based on the original un-compressed image where the UFCLS generated 9 signatures for unsupervised mixed pixel classification. For simplicity of demonstration,

only those abundance fraction maps that detected panels are shown in figures.

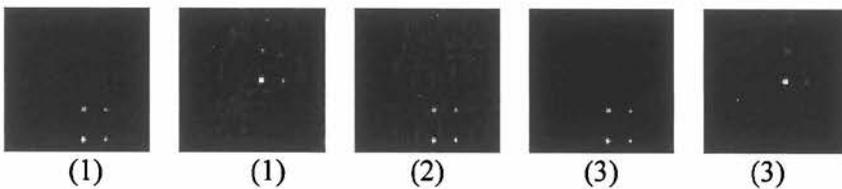


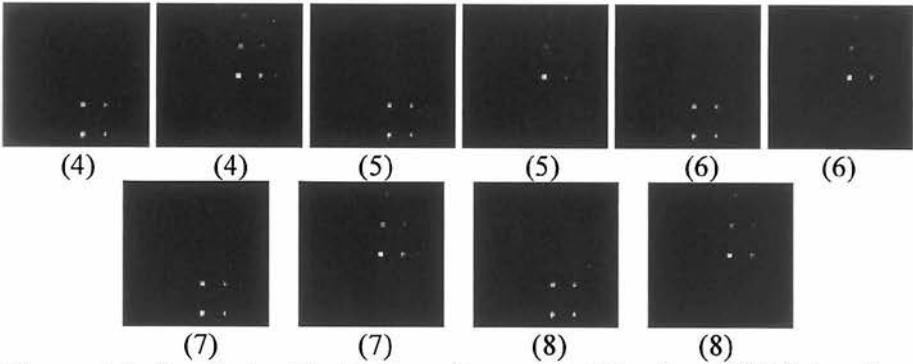
(a) panels in rows 4-5      (b) panels in rows 1-3

**Figure 13:** Panel classification results produced by the UFCLS using the original un-compressed image

As shown in Fig. 13, the UFCLS could neither separate those panels in rows 1-3 nor the panels in rows 4-5 due to the fact that the panel signatures  $P_1$ - $P_3$  were very similar and so are  $P_4$  and  $P_5$  [21]. In order to see the performance of the UFCLS on images compressed and de-compressed by a compression algorithm, the same experiments done for Fig. 13 were also conducted for the images compressed and de-compressed by the eight algorithms, (1) 3D-SPIHT, (2) 3D-multicomponent-JPEG2000, (3) IPCA/3D-multicomponent JPEG2000, (4) IPCA/2D-SPHIT, (5) IPCA/2D-JPEG2000, (6) PCA/3D-multicomponent JPEG2000, (7) PCA/2D-SPIHT and (8) PCA/2D-JPEG2000. For simplicity of demonstration, Figs. 14-17 only show those abundance fraction maps that detected panels with CR = 160, 80, 40 and 20 respectively. The numerals under figures indicate which algorithm was used to compress and decompress the images.

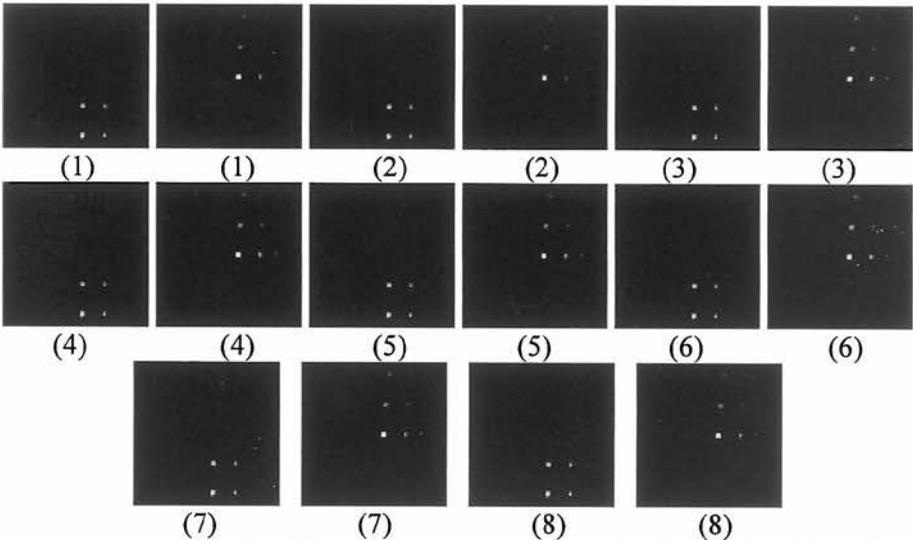
It is interesting to see that in Fig. 14 with CR = 160 the second algorithm, 3D-multicomponent JPEG2000 could only produced one panel classification map that classified panels in rows 4-5 and missed all panels in rows 1-3. Other than the 3D-multicomponent JPEG2000, all the other 7 algorithms produced very similar results which were close to the two panel classification maps in Fig. 13 that were produced by the UFCLS operating on the un-compressed HYDICE image.



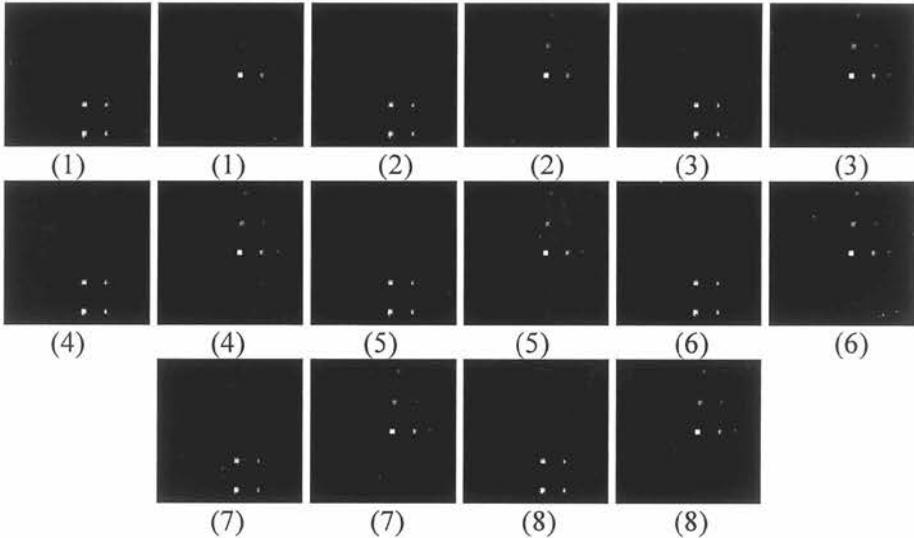


**Figure 14:** Panel classification results produced by the UFCLS based on images by compressed and de-compressed by the 8 algorithms with CR = 160

For images compressed by the eight algorithms with CR = 80 and 40, the panel classification results in Figs. 15-16 produced by the UFCLS were nearly the same as the two classification maps in Fig. 13. This implied that there was no much difference for the UFCLS operating on the un-compressed image or compressed images as long as CR was lower than 80:1.

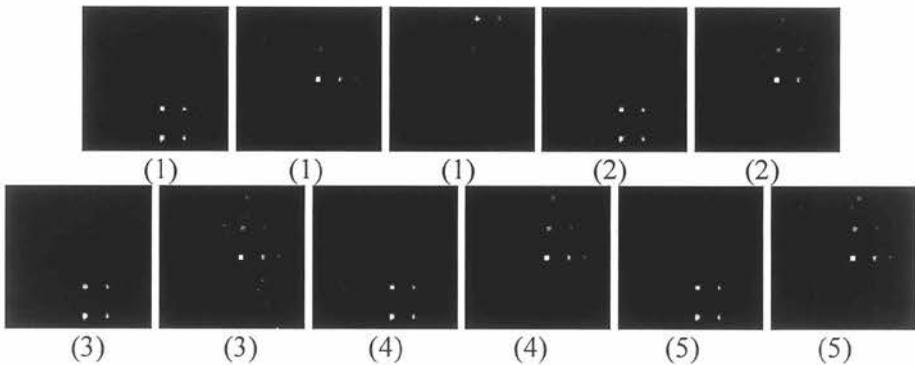


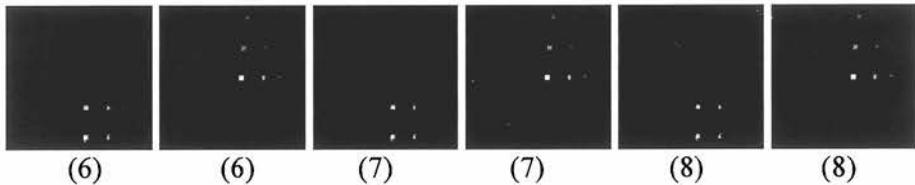
**Figure 15:** Panel classification results produced by the UFCLS based on images by compressed and de-compressed by the 8 algorithms with CR = 80



**Figure 16:** Panel classification results produced by the UFCLS based on images by compressed and de-compressed by the 8 algorithms with  $CR = 40$

When the  $CR = 20$ , Fig. 17 shows an interesting result produced by the UFCLS based on the compressed image by the first algorithm, the 3D-SPIHT where three panel classification maps were generated by the UFCLS compared to only two maps generated by the UFCLS based on the un-compressed image. These three classification maps were able to classify the panels in row 1 from the panels in rows 2-3, a task that could not be accomplished in Fig. 13. This implied that the UFCLS actually performed better on the 3D-SPIHT-compressed image than on the un-compressed image.





**Figure 17:** Panel classification results produced by the UFCLS based on images by compressed and de-compressed by the 8 algorithms with CR = 20

According to Figs. 13-17, the PCA-based spectral/spatial compression algorithms generally performed at least as well as 3D compression algorithms. One comment is noteworthy. Since the UFCLS only generated 9 signatures according to  $VD = 9$ , its mixed pixel classification was not able to separate all the panels in the five rows. This was due to the fact that there were many other unknown interfering signatures present in the scene. The 9 signatures were not sufficient to well represent the image scene. As shown in [21], if we allowed the UFCLS to extract 34 signatures for mixed pixel classification, then the classification images in Figs. 13-17 could have been able to separate all the 15 panels in the five rows in separate images. This was because the additional generated signatures would have been used in the UFCLS for interference annihilation to significantly improve classification results. A detailed study can be found in [31].

To conclude our experiments, we further calculated the SNR and MSE for performance of the eight algorithms with CR = 160, 80, 40 and 20. Table 6 tabulates their respective results which showed that PCA-based spectral/spatial compression always produced smaller MSE than 3D compression.

CR	160		80		40		20	
	SNR	MSE	SNR	MSE	SNR	MSE	SNR	MSE
1	28.85	807374	33.28	290854	38.19	94022	43.45	28023
2	27.08	1214986	30.47	555363	34.92	199507	40.21	58973
3	35.44	177005	41.62	42683	42.79	32613	42.79	32559
4	27.82	1023242	36.66	133739	42.69	33380	42.80	32525
5	24.56	2166596	34.46	221737	42.56	34397	42.79	32564
6	24.91	144479	36.44	10162	56.84	93	61.23	34
7	16.55	990819	26.46	101220	47.16	860	69.94	5
8	13.22	2134410	23.74	189222	43.78	1873	60.38	41

**Table 6:** SNR and MSE for the HYDICE image

As also shown in Table 6, for most of cases except CR =160, PCA-based spectral/spatial compression performed better than 3D compression in the sense that the former produced higher SNR than did the latter.

Like Table 2, since (6) the PCA/3D-multicomponent JPEG2000, (7) PCA/2D-SPIHT and (8) PCA/2D-JPEG2000 operated on only PC images not on de-compressed images with the original number of bands, their produced SNRs in Table 6 were generally lower than expected.

In order to see whether or not there was a change in the estimation of the VD for de-compressed images, Table 7 tabulates the VD estimated by the HFC method with various false alarm probabilities based on the decompressed images by (1) 3D-SPIHT, (2) 3D-multicomponent JPEG2000, (3) IPCA/3D multicomponent-JPEG2000, (4) IPCA/2D-SPHIT, (5) IPCA/2D-JPEG2000. Similar to Table 4(a-d), Table 7(a-d) also demonstrates that the VD estimates were more consistent when the CR was low.

$P_F$	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$
1	53	42	36	34	30
2	54	41	32	25	22
3	12	12	12	12	12
4	17	15	14	13	13
5	20	18	18	18	18

**Table 7(a):** Estimated VD for compressed image with CR = 160 by the HFC method with various false alarm probabilities

$P_F$	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$
1	47	31	27	23	22
2	47	34	28	24	20
3	10	10	10	9	9
4	10	9	9	9	8
5	11	9	9	9	8

**Table 7(b):** Estimated VD for compressed image with CR = 80 by the HFC method with various false alarm probabilities

$P_F$	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$
1	29	26	19	16	14
2	32	20	15	13	13
3	9	9	9	9	8
4	10	9	9	9	8
5	10	9	9	9	8

**Table 7(c):** Estimated VD for compressed image with CR = 40 by the HFC method with various false alarm probabilities

$P_F$	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$
1	10	10	8	8	7
2	10	9	9	9	8
3	11	9	9	9	8
4	10	9	9	9	8
5	10	9	9	9	8

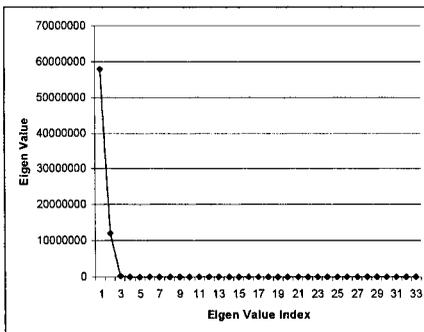
**Table 7(d):** Estimated VD for compressed image with CR = 20 by the HFC method with various false alarm probabilities

It is also noted that the PCA/3D compression and PCA/2D compression were not applicable since they did not apply IPCA to recover the PCA-compressed image in the original image space with 160 spectral bands.

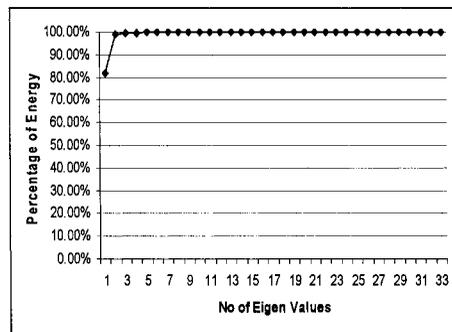
Finally, we also calculated the eigenvalue distribution and energy distribution of the first 33 PCs for the HYDICE image in Fig. 1 and the results are tabulated in Table 8 and plotted in Fig. 18.

PC	Eigenvalue	Percent	PC	Eigenvalue	Percent	PC	Eigenvalue	Percent
1	57995933.33	81.98%	12	2828.34	99.95%	23	617.0386	99.98%
2	12121150.07	99.14%	13	1986.268	99.95%	24	552.206	99.98%
3	173987.8	99.53%	14	1580.478	99.96%	25	508.911	99.98%
4	72008.77	99.69%	15	1377.929	99.96%	26	478.4526	99.98%
5	46640.85	99.80%	16	1139.912	99.96%	27	467.6903	99.98%
6	24729.75	99.85%	17	1054.234	99.97%	28	413.1114	99.98%
7	15673.41	99.89%	18	963.4904	99.97%	29	397.1537	99.98%
8	12623.25	99.92%	19	890.6664	99.97%	30	388.2148	99.98%
9	5230.008	99.93%	20	803.1869	99.97%	31	363.9604	99.98%
10	4051.45	99.94%	21	695.9851	99.97%	32	329.4372	99.98%
11	2975.395	99.94%	22	660.3904	99.97%	33	323.8451	99.99%

**Table 8:** Eigenvalues and energy percentages of the first 33 PCs



(a) Eigenvalue distribution



(b) energy distribution

**Figure 18:** Eigenvalue distribution and energy distribution of the first 33 PCs

From Table 8 and Fig. 18, there was a sudden drop between 2 and 3. So, an obvious choice for the number of PCs was 2, which is certainly not accurate. A detailed analysis and study on this issue was conducted by Ramakrishna in [31].

## 6. CONCLUSIONS

This chapter investigates the applicability of direct application of 3D compression techniques to hyperspectral imagery and develops PCA-based spectral/spatial compression techniques in conjunction with the virtual dimensionality (VD) for hyperspectral image compression where the VD is used to estimate number of principal components required to be preserved. In particular, we conduct computer simulations based on a synthetic image and real image experiments to demonstrate that simple PCA-based spectral/spatial lossy compression techniques can perform at least as well as 3D lossy compression techniques in applications such as mixed pixel classification and quantification. This interesting finding provides evidence that PCA-based spectral/spatial compression can be as competitive as the 3D compression for hyperspectral image compression. Additionally, this chapter also further demonstrates that the number of PCs required to be preserved by lossy compression is crucial and the proposed VD provides a much better estimate than the commonly used criterion determined by the sum of largest eigenvalues. For more details we refer to [31].

## ACKNOWLEDGEMENT

The authors would like to acknowledge the use of the QccPack developed by Dr. J.E. Fowler with Mississippi State University for the experiments conducted in this work. A. Plaza would like to thank for support received from the Spanish Ministry of Education and Science (PR2003-0360 Fellowship).

## REFERENCES

- [1] V.D. Vaughn and T.S. Wilkinson, "System considerations for multispectral image compression designs," *IEEE Signal Processing Magazine*, pp. 19-31, January 1995.

- [2] J.A. Saghri, A.G. Tescher and J.T. Reagan, "Practical transform coding of multispectral imagery," *IEEE Signal Processing Magazine*, pp. 32-43, January 1995.
- [3] J.A. Saghri and A.G. Tescher, "Near-lossless bandwidth compression for radiometric data," *Optical Engineering*, vol. 30, no. 7, pp. 934-939, July 1991.
- [4] J.A. Saghri, A.G. Tescher and A. Boujarwah, "Spectral-signature-preserving compression of multispectral data," *Optical Engineering*, vol. 38, no. 12, pp. 2081-2088, December 1999.
- [5] G.P. Abousleman, E. Gifford and B.R. Hunt, "Enhancement and compression techniques for hyperspectral data," *Optical Engineering*, vol. 33, no. 8, pp. 2562-2571, August 1994.
- [6] G.P. Abousleman, M.W. Marcellin and B.R. Hunt, "Hyperspectral image compression using entropy-constrained predictive trellis coded quantization," *IEEE Trans. on Image Processing*, vol.6, no. 4, pp. 566-573, April 1994.
- [7] G.P. Abousleman, M.W. Marcellin and B.R. Hunt, "Compression of hyperspectral imagery using the 3-D DCT and hybrid DPCM/DCT," *IEEE Trans. on Geoscience and Remote Sensing*, vol.33, no. 1, pp. 26-34, April 1994.
- [8] S.-E. Qian, A.B. Hollinger, D. Williams and D. Manak, "Fast three-dimensional data compression of hyperspectral imagery using vector quantization with spectral-feature-based binary coding," *Optical Engineering*, vol. 35, no. 7, pp. 3242-3249, November 1996.
- [9] R.O. Duda and R.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, 1973.
- [10] J.A. Richards, *Remote Sensing Digital Image Analysis*, 2nd ed. Springer-Verlag. 1993.
- [11] A.A. Green, M. Berman, P. Switzer and M.D. Craig, "A transformation for ordering multispectral data in terms of image quality with implications for noise removal," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 26, no. 1, pp. 65-74, January 1988.
- [12] J.B. Lee, A.S. Woodyatt and M. Berman, "Enhancement of high spectral resolution remote sensing data by a noise-adjusted principal components transform," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 28, no. 3, pp. 295-304, May 1990.
- [13] C.-I Chang, Q. Du, T.S. Sun and M.L.G. Althouse, "A joint band prioritization and band decorrelation approach to band selection for

- hyperspectral image classification," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 37, no. 6, pp. 2631-2641, November 1999.
- [14] B.-J. Kim, Z. Xiong, and W.A. Pearlman, "Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 8, pp.1374-1387, December 2000.
- [15] A. Said and W.A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and systems for Video Technology*, vol. 6, no. 3, pp. 243-350, June 1996.
- [16] D. Taubman, "High performance scalable image compression with EBCOT", *IEEE Trans. Image Proc.*, 9, 1158-1170
- [17] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standard and Practice*. Boston, MA: Kluwer, 2002.
- [18] ISO, Information Technology—JPEG 2000 Image Coding System - Part 1: Core Coding System, ISO, Geneva, Switzerland, 2000.
- [19] ISO, Information Technology—JPEG 2000 Image Coding System - Part 2: Extensions; Final Committee Draft, ISO, Geneva, Switzerland, Dec. 2000.
- [20] C.-I Chang and Q. Du, "Estimation of number of spectrally distinct signal sources in hyperspectral imagery," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 42, no. 3, pp. 608-619, March 2004.
- [21] C.-I Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*, Kluwer Academic/Plenum Publishers, New York, N.Y., 2003
- [22] J.C. Harsanyi, W. Farrand and C.-I Chang, "Detection of subpixel spectral signatures in hyperspectral image sequences," *Annual Meeting, Proceedings of American Society of Photogrammetry & Remote Sensing*, Reno, pp. 236-247, 1994.
- [23] T.W. Anderson, *Multivariate Analysis*, Academic Press, 2<sup>nd</sup> ed., 1984.
- [24] C.-I Chang, Q. Du, T.S. Sun and M.L.G. Althouse, "A joint band prioritization and band decorrelation approach to band selection for hyperspectral image classification," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 37, no. 6, pp. 2631-2641, November 1999.
- [25] S. S. Shen and B. S. Beard, "Effects of hyperspectral compression on non-literal exploitation," *Proceedings of SPIE*, vol. 3438, pp. 191-199, 1998.

- [26] Arto Kaarna, Pekka J. Toivanen, Pekka Keranen, "Compression of multispectral AVIRIS images," *Proceedings of SPIE*, vol. 4725, pp.588-599, 2002
- [27] J.E. Fowler, "QccPack: An open-source software library for quantization, compression, and coding," in *Applications of Digital Image Processing XXIII*, vol. 4115, pp.249-301, San Diego, CA, 2000.
- [28] Kakadu software: A Comprehensive Framework for JPEG2000, [www.Kakadusoftware.com](http://www.Kakadusoftware.com), Implementation of JPEG2000.
- [29] J.C. Harsanyi and C.-I Chang, "Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection," *IEEE Trans on Geoscience and Remote Sensing*, vol. 32, no. 4, pp. 779-785, July 1994.
- [30] D. Heinz and C.-I Chang, "Fully constrained least squares linear mixture analysis for material quantification in hyperspectral imagery," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 39, no. 3, pp. 529-545, March 2001.
- [31] Bharath Ramakrishna, *Principal Components Analysis (PCA)-Based Spectral/Spatial Hyperspectral Image Compression*, MS. Thesis, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD, 2004.