Low-Complexity Principal Component Analysis for Hyperspectral Image Compression

Qian Du and James E. Fowler

Abstract— **Principal component analysis (PCA) is an effective tool for spectral decorrelation of hyperspectral imagery, and PCA-based spectral transforms have been employed successfully in conjunction with JPEG2000 for hyperspectral-image compression. However, the computational cost of determining the data-dependent PCA transform is high due to its traditional eigendecomposition implementation which requires calculation of a covariance matrix across the data. Several strategies for reducing the computation burden of PCA are explored, including both spatial and spectral subsampling in the covariance calculation as well as an iterative algorithm that circumvents determination of the covariance matrix entirely. Experimental results investigate the impacts of such low-complexity PCA on JPEG2000 compression of hyperspectral images, focusing on rate-distortion performance as well as data-analysis performance at an anomaly-detection task.**

Index Terms— **principal component analysis, hyperspectral image compression, JPEG2000, spectral decorrelation, anomaly detection**

I. Introduction

The vast data volumes contained in typical hyperspectral imagery have entailed an increasing interest in wavelet-based lossy compression techniques which have exhibited excellent rate-distortion performance in many image-processing application settings. Prominent wavelet-based techniques include SPIHT (Said and Pearlman 1996) and SPECK (Pearlman, Islam, Nagaraj, and Said 2004), both of which having been extended to 3D for the compression of hyperspectral imagery (Kim, Xiong, and Pearlman 2000; Tang and Pearlman 2005). Yet, it is perhaps the recent JPEG2000 standard (Taubman and Marcellin 2002), that has garnered the most attention and has been the most widely deployed in hyperspectral applications (e.g., (Rucker, Fowler, and Younan 2005; Kulkarni, Bilgin, Marcellin, Dagher, Kasner, Flohr, and Rountree 2006; Penna, Tillo, Magli, and Olmo 2006b; Fowler and Rucker 2007; Du and Fowler 2007)). However, direct extension to 3D of algorithms such as these that were, in essence, originally designed for 2D imagery may be problematic without considerations of characteristics specific to hyperspectral imagery (Pickering and Ryan 2006).

In general, JPEG2000 is increasing in prominence because it is a widely-used standard and because its performance tends to be superior to that of other wavelet-based techniques—according to our experience in hyperspectral-image compression, JPEG2000 usually provides a signal-to-noise (SNR)[1] that is about 0.1–0.5 dB higher than that of SPIHT or SPECK operating at the same bitrate[2]. For example, Table I compares the rate-distortion performance 3D-SPIHT (Kim, Xiong, and Pearlman 2000), 3D-SPECK (Tang and Pearlman 2005), and JPEG2000 for three popular AVIRIS radiance datasets. For these results, each of the coders uses a wavelet packet transform consisting of a 1D discrete wavelet transform (DWT) deployed spectrally followed by a dyadic 2D DWT deployed spatially—this particular 3D transform structure has been employed widely (Fowler and Rucker 2007) and has been shown to offer near-optimal performance over all possible packet transforms (Penna, Tillo, Magli, and Olmo 2006b).

Although a 3D DWT such as used in Table I is perhaps the most obvious transform choice, other approaches are possible. In particular, principal component analysis (PCA) is a widely-used technique that has demonstrated excellent performance at spectral decorrelation. PCA can be deployed in hyperspectral compression by replacing the 1D DWT in the transform used in Table I with a 1D spectral PCA. The resulting hybrid transform consisting of spectral PCA coupled with spatial DWT is compliant with the multicomponent-transformation extension of Part 2 of the JPEG2000 standard (ISO/IEC 15444-2 2004), and it has been been demonstrated that JPEG2000 using this transform (which we call “PCA+JPEG2000”) yields superior rate-distortion performance (Du and Fowler 2007). For example, Fig. 1 compares the rate-distortion performance of PCA+JPEG2000 with that of DWT+JPEG2000, the corresponding coder in which the spectral transform is a DWT. We observe that, although the two coders differ only in the spectral transform employed, there is up to a 5-dB gap between the two techniques.

Since the number of spectrally distinct signal sources composing a given hyperspectral scene is limited, it has been argued that the spectral dimensionality intrinsic to a hyperspectral image is typically much less than the number of spectral bands. As a consequence, we expect that many principal components (PCs) contain essentially noise only. For example, in (Chang and Du 2004), a virtual dimensionality (VD) was estimated by counting the number of PCs containing signals via hypothesis testing to determine the minimum number of spectrally distinct signals present in the dataset. These observations argue that, for data compression, one should encode only a subset of PCs rather than the full complement. Indeed, in (Du and Fowler 2007), such an action resulted in an improvement in rate-distortion performance for a PCA-based spectral transform. In this sense, PCA is exploited not only for spectral decorrelation, but also for explicit spectral dimensionality reduction. As shown in Fig. 2, choosing the ideal number of PCs to retain can result in a significant improvement in SNR, particular at low bitrates. In (Du and Fowler 2007), it was also determined that data-analysis performance, such as

[1] We define SNR as the log ratio between the variance of the original image and that of the difference image.

[2] We measure rate in terms of bits per pixel per band (bpppb).

International Journal of High Performance Computing Applications, vol. 22, pp. 438-448, November 2008.

anomaly detection, is also similarly maximized by coding a subset of PCs rather than the full number, and the number of PCs that yields maximum SNR for a given dataset is usually close to the number of PCs that yields maximum detection performance. A simple linear model was proposed in (Du and Fowler 2007) to estimate the optimal number of PCs for different bitrates.

These prior investigations have revealed that a PCA-based spectral transform can be quite advantageous in the compression of hyperspectral imagery. Yet, PCA is a data-dependent transform, meaning that the transform itself must be calculated for each individual dataset before it can be applied. Unfortunately, the computational complexity of this transform-design process is quite substantial, a fact that has hindered greater adoption of PCA-based transforms in practice. In this paper, we explore the reasons for this high computational cost, as well as several ways to circumvent it. Specifically, we see that the calculation of a covariance matrix underlies the majority of the computation, and that effectively subsampling the dataset can ease the computational burden. Additionally, we find that avoiding calculation of the covariance matrix in the first place by directly producing the transform via neural-network-based iterative PCA (IPCA) (Baldi and Hornik 1995) is feasible in certain instances of low-rate coding. Such an approach differs from others that employ direct matrix operations to conduct PCA (e.g., (Dremmel 1997)) since the covariance matrix must be explicitly calculated in this latter case.

Hardware implementation of PCA based on both matrix operations as well as neural networks has been studied. For PCA based on matrix operations, systolic arrays have been used to compute eigenvectors via singular value decomposition (SVD) (Schreiber 1986). Additionally, parallel processing (Subramanian, Gat, Ratcliff, and Eismann 2000; Fleury, Self, and Downton 2004; Kumar, Kamakoti, and Das 2007) has been employed; in such schemes, the covariance-matrix computation and PC transform permit fine-grain parallelism but the eigenvector generation does not, leading to an offline computation of the latter in many cases. PCA based on neural networks is generally more suitable to VLSI implementation because of topology regularity. For example, a systolic-array mapping was discussed in (Diamantaras and Kung 1996). As for hardware design and implementation in field-programmable gate arrays (FPGAs), pulsed-circuit architectures based on pulse-density modulation and delta-sigma modulation can be found in (Hirai and Nishizawa 2000) and (Murahashi, Hotta, Doki, and Okuma 2004), respectively, and a fully digital design was presented in (Prasanna, Sudha, and Kamakoti 2005). Each of these schemes implement a well-known neural-network PCA called Generalized Hebbian learning (Sanger 1989). In the work here, we also use PCA based on Hebbian learning, but with a slightly different algorithm. Consequently, even though our focus in this paper is on algorithmic issues surrounding spectral PCA for hyperspectral imagery, (Diamantaras and Kung 1996; Hirai and Nishizawa 2000; Murahashi, Hotta, Doki, and Okuma 2004; Prasanna, Sudha, and Kamakoti 2005) suggest a number of paths toward a hardware implementation of our approach.

The remainder of the paper is organized as follows. In Sec. II, we describe the computational complexity of PCA and delineate the components with the largest computational load. In Sec. III, we explore various alternatives for low-complexity PCA, including subsampling both spatially and spectrally in additional to IPCA implemented with a neural-network architecture. Then, in Sec. IV, we investigate the impact that the proposed low-complexity PCA strategies have on not only rate-distortion performance, but also data-analysis performance at an anomaly-detection task. Finally, we make some concluding remarks in Sec. V. We note that a number of the investigations presented here originated in the preliminary work and results of (Zhu 2007).

II. Computational Complexity of PCA

Suppose we have a dataset consisting of M vectors $\mathbf{x}_i$, $1 \leq i \leq M$, wherein each vector has dimension N; i.e., $\mathbf{x}_i \in \Re^N$. A linear transform T maps from N-dimensional space to P-dimensional space ($P \leq N$),

$$\mathbf{y}_i = T(\mathbf{x}_i) \in \Re^P. \tag{1}$$

For finite dimensions, $T(\cdot)$ is a vector-matrix multiplication involving a $P \times N$ matrix $\mathbf{T}$,

$$\mathbf{y}_i = \mathbf{T}\mathbf{x}_i, \tag{2}$$

an operation that requires PN multiplications, i.e., an operation of complexity $O(PN)$.

Assume a hyperspectral image consists of a total of M pixels in N spectral bands. Like other linear transforms, PCA that retains P PCs can be applied via (2) as a spectral transform on the hyperspectral image with complexity, $\mathcal{C}_{\mathbf{T}}$,

$$\mathcal{C}_{\mathbf{T}} = O(MPN). \tag{3}$$

However, unlike other linear transforms, PCA is data dependent, and the transform matrix $\mathbf{T}$ itself must be determined before (2) can be applied. This tends to be a costly process, and herein lies the primary drawback of PCA in compression applications.

The straightforward procedure for determining $\mathbf{T}$ for PCA consists of two steps:

Step 1: calculate the covariance matrix $\mathbf{\Sigma}$ of the set of pixel vectors;

Step 2: decompose $\mathbf{\Sigma}$ to find its eigenvectors,

with the resulting eigenvectors placed row-wise to assemble $\mathbf{T}$. The determination of $\mathbf{\Sigma}$ in Step 1 needs MN^2 multiplications and thus is of complexity $\mathcal{C}_{\mathbf{\Sigma}}$,

$$\mathcal{C}_{\mathbf{\Sigma}} = O(MN^2). \tag{4}$$

On the other hand, the eigendecomposition of Step 2 can be accomplished by any of a number of matrix-diagonalization methods of varying complexity. For instance, singular value decomposition (SVD) is widely used for this purpose due to its numerical stability. SVD decomposition finds all N eigenvectors such that its computational complexity, $\mathcal{C}_{\mathbf{\Lambda}}$, is

$$\mathcal{C}_{\mathbf{\Lambda}} = O(N^3). \tag{5}$$

However, we need only the first P eigenvectors to assemble $\mathbf{T}$, assuming that the eigenvectors are sorted in order of descending eigenvector magnitude. Fig. 2 suggests that the ideal P is much smaller than N ($P \ll N$) for typical hyperspectral imagery; thus, Hotelling's power method (Hotelling 1933), an iterative procedure that finds only the largest eigenvalues and their corresponding eigenvectors, is more reasonable, since its complexity is $O(PN^2)$.

Because the number of pixels M in a remotely sensed image is typically much larger than the number of bands N, $\mathcal{C}_{\mathbf{\Sigma}}$ usually dominates both $\mathcal{C}_{\mathbf{\Lambda}}$ well as $\mathcal{C}_{\mathbf{T}}$. That is, the complexity, $\mathcal{C}_{\text{PCA}}$, of the traditional eigendecomposition approach to PCA is

$$\begin{aligned}\mathcal{C}_{\text{PCA}} &= \mathcal{C}_{\mathbf{\Sigma}} + \mathcal{C}_{\mathbf{\Lambda}} + \mathcal{C}_{\mathbf{T}} \\ &= O(MN^2). \end{aligned} \tag{6}$$

For example, consider a hyperspectral image of spatial size 512×512 with 224 spectral bands. The calculation of $\mathbf{\Sigma}$ requires $\mathcal{C}_{\mathbf{\Sigma}} = 512 \times 512 \times 224^2 \approx 1.3 \times 10^{10}$ multiplications, while only an additional $\mathcal{C}_{\mathbf{\Lambda}} = 224^3 \approx 1.1 \times 10^7$ multiplications are required to find the eigenvectors via SVD. If, say, 40 PCs were to be retained for the spectral transform, $\mathcal{C}_{\mathbf{T}} = 512 \times 40 \times 224 \approx 2.3 \times 10^9$ multiplications would be devoted to performing the spectral transform itself, still an order of magnitude less complex than the covariance computation.

III. Approaches to Reduction of PCA Complexity

Since $\mathcal{C}_{\text{PCA}}$ is dominated by $\mathcal{C}_{\mathbf{\Sigma}}$, the most straightforward approach to reducing PCA complexity focuses on reducing the two components of $\mathcal{C}_{\mathbf{\Sigma}}$, namely, M or N (or both). This is accomplished by subsampling the dataset either spatially or spectrally in advance of the covariance-matrix calculation; these approaches are discussed below in Sec. III-A. However, in certain situations, it is possible to determine the PCA eigendecomposition without first explicitly determining the covariance matrix. We outline one such method based on a neural network and determine when the use of such a technique is warranted below in Sec. III-B.

A. Subsampling for Complexity Reduction

A straightforward approach to reducing the computational complexity of PCA is to decrease M in (6). That is, instead of using all M pixels, only a small subset of pixels of size M', $M' \ll M$, is used to estimate $\mathbf{\Sigma}$. The transform matrix $\mathbf{T}$ is then determined from the estimated covariance matrix, $\widehat{\mathbf{\Sigma}}$, following the usual eigendecomposition procedure, which now has complexity

$$\mathcal{C}_{\text{SS+PCA}} = O(M'N^2). \tag{7}$$

Here, "SS+PCA" denotes PCA coupled with spatial subsampling for the calculation of the covariance matrix. We assume here that we ignore $\mathcal{C}_{\mathbf{T}}$ as a fixed cost of applying any linear transform (PCA or other) to focus solely on the transform-design costs.

While M' needs to be relatively small in order to reduce computational complexity to the greatest extent, we must choose M' to be sufficiently large so as to adequately capture the stastical variation in the dataset. This will, in turn, depend on how many PCs we want to obtain, since a greater number of PCs will require a larger M'. We adopt the general heuristic that we need at least an order of magnitude more samples than the number of items we wish to obtain through training—a similar heuristic was often employed to determine the size of a training set for vector quantizers; e.g., see observations in (Makhoul, Roucos, and Gish 1985) surrounding the design of vector quantizers for speech signals. Since we need to determine N PCs for full PCA, this heuristic dictates that M' should be at least $10N$. In this case, with $M' \gg N$, $\mathcal{C}_{\mathbf{\Sigma}}$ still dominates $\mathcal{C}_{\mathbf{\Lambda}}$ in (6).

An alternative approach would be to decrease N in (6) to $N' < N$. This can be accomplished by removing water-absorption bands and other spectral bands likely to be mostly noise before determining the covariance matrix. This approach is reasonable since such high-noise bands are not typically useful in many hyperspectral applications, provided that identification of the high-noise bands can be easily accomplished with little computation. In the case of water-absorption bands, this is certainly the case, since water absorption occurs at known frequency ranges.

It should be noted that subsampling the hyperspectral volume to M' or N' results in a transform that is only approximately the spectral PCA of the original image. As a consequence, there is likely to be a deleterious effect on performance resulting from the approximate transform. (Penna, Tillo, Magli, and Olmo 2006a) suggests that the degradation to rate-distortion performance is minimal; below, we investigate this as well as the effect on data-analysis performance in the form of anomaly detection.

B. IPCA

When the number of PCs P preserved for compression is small, IPCA may be preferred, since, in this approach, the calculation of $\mathbf{\Sigma}$ is not explicitly performed. Rather, the complexity required to find an eigenvector increases linearly with N and M. Our approach to IPCA employs a neural network and well-known Hebbian learning. The neural network has a simple feedforward structure—one input layer with N neurons, an output layer with one neuron, and weights connecting the input neurons to the output neuron. After convergence, the output neuron provides the largest eigenvalue with the weights representing the corresponding eigenvector.

Let $\mathbf{w}_1$ denote the first eigenvector to be estimated. Denote the neural-network input as $\mathbf{x}$ (a pixel vector from the hyperspectral image) and the output as $\lambda_1 = \mathbf{w}_1^T \mathbf{x}$. According to the Hebbian learning rule (Haykin 1999), the weight update is

$$\Delta \mathbf{w}_1 = \alpha \lambda_1 \mathbf{x}, \tag{8}$$

where α is the learning rate; for simplicity here, we set $\alpha = 1$. The weight vector must be normalized to prevent unlimited growth. Therefore, the additional equations for update are

$$\mathbf{w}_1 \leftarrow \mathbf{w}_1 + \Delta \mathbf{w}_1, \tag{9}$$

$$\mathbf{w}_1 \leftarrow \frac{\mathbf{w}_1}{\|\mathbf{w}_1\|}. \tag{10}$$

In sequential mode, $\mathbf{w}_1$ is updated after each pixel is presented to the network. On the other hand, in batch mode, $\mathbf{w}_1$ is not updated until after all the pixels from the image are presented to the network. We find that batch mode provides more accurate results; i.e., the final result is closer to the actual eigenvector. As a consequence, we consider batch-mode training exclusively in the following.

After $\mathbf{w}_1$ is estimated, the second eigenvector $\mathbf{w}_2$ can be determined similarly. However, the dataset is first projected onto the first eigenvector $\mathbf{w}_1$, and this projection removed before $\mathbf{w}_2$ is determined. Likewise, to estimate eigenvector i, $\mathbf{w}_i$, projections along the previous $i-1$ eigenvectors are removed before the data is presented to the neural network. In other words, the update equation when estimating $\mathbf{w}_i$ for $i > 1$ is

$$\mathbf{x} \leftarrow \mathbf{x} - \mathbf{w}_{i-1}\mathbf{w}_{i-1}^T\mathbf{x}. \tag{11}$$

This process, akin to Gram-Schmidt orthogonalization, guarantees extraction of a set of mutually orthogonal eigenvectors.

The detailed IPCA algorithm is as follows:

Step 1: Let $i = 0$ and $\mathbf{w}_0 = \mathbf{0}$.

Step 2: Let $i \leftarrow i+1$ and use (11) to update $\mathbf{x}_m$, $1 \leq m \leq M$.

Step 3: Let $k = 0$. Initialize $\mathbf{w}_i^{(0)}$ with a random vector orthogonal to all $\mathbf{w}_j$ for $1 \leq j \leq i$.

Step 4: Let $k \leftarrow k+1$. If $k = K_{\text{max}}$, set $\mathbf{w}_i \leftarrow \mathbf{w}_i^{(k-1)}$ and go to Step 2 for the next eigenvector. Otherwise, use (8)–(10) to update $\mathbf{w}_i^{(k)}$.

Step 5: If $\left\|\mathbf{w}_i^{(k)} - \mathbf{w}_i^{(k-1)}\right\| < \eta$, $\mathbf{w}_i^{(k)}$ is considered to have converged to the i^{th} eigenvector, so set $\mathbf{w}_i \leftarrow \mathbf{w}_i^{(k)}$. If $i = P$, terminate. Otherwise, go to Step 2 to find the next eigenvector. On the other hand, if $\left\|\mathbf{w}_i^{(k)} - \mathbf{w}_i^{(k-1)}\right\| \geq \eta$, go to Step 4 for the next iteration on $\mathbf{w}_i^{(k)}$.

It should be noted that this one-output neural network finds the eigenvectors sequentially, one after another, while data orthogonalization is achieved by simple matrix multiplication outside the neural network. On the other hand, Generalized Hebbian learning (Sanger 1989) uses multiple outputs to find multiple eigenvectors simultaneously—its weight-update equation thus includes data orthogonalization (Haykin 1999). There are two advantages to our algorithm: 1) it is more flexible in generating an arbitrary number of eigenvectors as needed because of its sequential nature; and 2) it requires fewer multiplications since data orthogonalization is accomplished in a single step.

Let $K_1, K_2, \ldots, K_P$ be the number of iterations required to find the first P eigenvectors, respectively, in the algorithm above. Clearly, these values will depend on the threshold η. If η is very small, then many iterations will be needed to find an eigenvector; however, the maximum difference between the estimated and true eigenvectors will be guaranteed to be small. We have found that, for $\eta = 10^{-3}$, each component of the estimated eigenvectors differs from its corresponding true value by less than 10^{-4}, while the number of iterations remains small. Since the number of multiplications in each iteration is MN, the complexity for finding the P eigenvectors via IPCA is

$$\begin{aligned} \mathcal{C}_{\text{IPCA}} &= O\left(\sum_{i=1}^{P} K_i MN\right) \\ &= O\left(\bar{K}PMN\right), \end{aligned} \tag{12}$$

where $\bar{K}$ is the average number of iterations,

$$\bar{K} = \frac{1}{P}\sum_{i=1}^{P} K_i. \tag{13}$$

Clearly, when the number of eigenvectors P to be estimated is large, then IPCA will not result in reduced complexity as compared to PCA. The largest P that still yields reduction in complexity with respect to PCA can be approximately determined by the following. By comparing (12) to (6), we see that, for complexity reduction, we need

$$\bar{K}PMN < MN^2. \tag{14}$$

Thus, P is approximately bounded as

$$P \lessapprox \left\lfloor \frac{N}{\bar{K}} \right\rfloor. \tag{15}$$

For example, if $N = 224$, and $\bar{K} = 5$, then $P \lessapprox 44$.

Clearly, to achieve significant reduction of computational complexity using IPCA, $\bar{K}$ needs to be small. We have found that, when determining the first several eigenvectors, the neural network tends to converge quickly. That is, K_i tends to be small—say, less than 5—for small values of i. However, many more iterations are typically needed to find the eigenvectors for larger values of i. This effect is due to the fact that the eigenvectors $\mathbf{w}_i$ are mutually orthogonal, and it becomes more and more difficult for a randomly initialized $\mathbf{w}_i$ to converge to a vector that is orthogonal to all the previous eigenvectors. One approach to partially alleviating this difficulty is to initialize $\mathbf{w}_i$ to a random vector that is orthogonal to all other eigenvectors already determined, a process which is easily accomplished via a Gram-Schimdt orthogonalization (as we have done above). To make $\bar{K}$ sufficiently small so as to significantly reduce complexity as compared to PCA, each $\mathbf{w}_i$ is updated with a certain number of iterations not greater than K_{max}, a fixed upper bound. Since K_{max} is itself small, $\bar{K} \approx K_{\text{max}}$. This upper bound entails encoding the image using a spectral transform that may be somewhat distant from the true spectral PCA. Below, we investigate the impact of this approximate spectral transform on rate-distortion as well data-analysis performance. First, however, let us note that IPCA can be applied in conjunction with spatial subsampling, leading to a complexity of

$$\mathcal{C}_{\text{SS+IPCA}} = O(\bar{K}PM'N) \tag{16}$$

when $M' \ll M$ pixels are used to drive the IPCA design procedure.

IV. Performance Evaluation

To evaluate the performance of low-complexity PCA, three AVIRIS radiance datasets of size 512×512 with 224 bands were used. We use Scene 1 of the Cuprite, Jasper Ridge, and Moffett radiance datasets[3] cropped spatially to size 512×512 from the upper left corner; band 100 for each of these datasets is depicted in Fig. 3(a)–(c). In addition to rate-distortion performance, we are also concerned with object information that is preserved in the reconstructed data. For such data-analysis performance, we focus on anomaly detection, wherein an anomaly is defined as a small object or material whose spectral signature is very different from the background and, as such, most likely indicates an unknown target. Anomaly detection provides a reasonable test of the performance of lossy compression at information preservation because such small objects or materials are prone to be sacrificed during compression; yet such anomalies may be critical in certain remote-sensing applications. Here, the well-known RX algorithm (Reed and Yu 1990) is employed for anomaly detection, and, since the exact spatial structure of objects is unknown, detection is conducted at the pixel level (Chang and Chiang 2002). Due to the lack of real ground truth for the AVIRIS datasets considered, we employ the detection map resulting from the original data as "ground truth" in the experimental results below. Specifically, we compare the detection map resulting from the reconstructed data to that from the original data, using as a measure of similarity the spatial correlation coefficient, ρ. The closer ρ is to 1.0, the better the detection performance is considered to be. The anomaly-detection maps for the datasets under consideration are depicted in Fig. 3(d)–(f).

As discussed in Sec. II, the calculation of the covariance matrix $\mathbf{\Sigma}$ entails the majority of the computation associated with the traditional eigendecomposition implementation of PCA, and subsampling the dataset spatially is a straightforward approach to cutting this cost. For the results here, we employ the heuristic of Sec III-A that $M' \gtrapprox 10N$. With $N = 224$ and $M = 512^2 = 262,144$, we have that M' should be roughly 1% of pixels; we note that the results of (Penna, Tillo, Magli, and Olmo 2006a) were also carried out using a 1% subsampling rate. Consequently, for the datasets considered, we set $M' = 2621$, and choose the M' pixels at random. To gauge the effect of spatial subsampling on the spectral transform, we employ both traditional SVD-based PCA and IPCA on the subsampled dataset to generate the eigenvectors and thus the spectral transform. For IPCA, no more than 5 iterations were used to update each eigenvector; i.e., $K_{\text{max}} = 5$, and $\bar{K} \approx K_{\text{max}}$. For this K_{max}, IPCA will have approximately the same computational complexity as SVD when $P = 44$. In the following, "PCA" refers to a spectral transform composed of eigenvectors determined from SVD of the true covariance matrix $\mathbf{\Sigma}$; "SS+PCA" refers to a spectral transform composed of eigenvectors determined from SVD of $\widehat{\mathbf{\Sigma}}$ estimated via a random spatial subsampling of 1% of the pixels; and "SS+IPCA" refers to a spectral transform composed of eigenvectors determined from IPCA with $K_{\text{max}} = 5$ applied to $\widehat{\mathbf{\Sigma}}$ estimated from a random spatial subsampling of 1% of the pixels.

Fig. 4 compares the first three eigenvectors for the Jasper Ridge dataset as calculated by the various approaches. The discrepancy between the eigenvectors of the true covariance matrix as opposed to those calculated from the covariance as estimated using only 1% of the dataset is small— specifically, the spectral angles between the eigenvectors resulting from subsampling and those of the true covariance matrix are under 3° in all cases.

Table II investigates rate-distortion performance when the low-complexity PCA approaches are used to design spectral transforms for PCA+JPEG2000 coding. In Fig. 2, it was illustrated that rate-distortion performance varies with the number of PCs retained for coding, with the best performance typically occurring for some $P < N$, i.e., when only some subset of PCs were retained. Along these lines, (Du and Fowler 2007) presents a simple linear model designed to predict the number of PCs yielding maximal SNR as a function of the bitrate used for coding. This model is used here, and Table II compares the rate-distortion performance for coding with this number of PCs ($\widetilde{P}^*$) using the various spectral-transform designs. We see that the drop in SNR due to using the low-complexity PCA approaches is typically not greater than 0.1 dB. Thus, the impact on rate-distortion performance from the use of low-complexity PCA appears to be rather minimal.

Table III presents a quantitative evaluation of anomaly-detection performance for PCA+JPEG2000 with low-complexity PCA. We see that, when $\widetilde{P}^*$ PCs are retaining and coded, the correlation coefficient ρ is largely unchanged when low-complexity PCA is employed. Interestingly, a larger variation in ρ is seen between traditional SVD-based PCA and its low-complexity counterpart when all PCs are retained and coded.

In terms of computational complexity, for spatial subsampling of 1%, the complexity of subsampled PCA, $\mathcal{C}_{\text{SS+PCA}}$ is 1% of that of traditional SVD-based PCA, $\mathcal{C}_{\text{PCA}}$. On the other hand, the relative speedup of IPCA depends on the number of PCs retained; for $\widetilde{P}^* = 21$, 25, and 38 as used in Tables II and III, $\mathcal{C}_{\text{SS+IPCA}}$ of (16) is 0.47%, 0.56%, and 0.85% of $\mathcal{C}_{\text{PCA}}$, respectively. For the datasets considered here, $\mathcal{C}_{\mathbf{T}}$, the cost of applying $\mathbf{T}$ as a spectral transform, is actually larger than either $\mathcal{C}_{\text{SS+PCA}}$ or $\mathcal{C}_{\text{SS+IPCA}}$, the costs of determining $\mathbf{T}$.

Removing spectral bands with low quality (i.e., water-absorption bands) offers an alternative approach to reduce complexity. For the three AVIRIS datasets, bands 1–3, 108–113, 153–169, and 222–224 can be discarded, leaving 195 bands for further processing. Just this simple band removal reduces the complexity by about 24%, even when all pixels are used to compute $\mathbf{\Sigma}$. Table IV compares performance for PCA+JPEG2000 coding under such "bad-band" removal to the case in which all bands are coded. Here, PCA is applied using SVD with no spatial subsampling, and rate-distortion as well as anomaly-detection performance is tabulated. We see that distortion, as measured by mean squared error (MSE) as calculated over only the 195 bands in question, is largely unaffected by the bad-band removal, while the spatial correlation coefficient ρ remains unchanged or slightly improved.

[3] http://aviris.jpl.nasa.gov/

V. Conclusion

PCA has been shown to offer significantly improved rate-distortion and data-analysis performance as compared to a DWT when used as a spectral transform in JPEG2000 compression of hyperspectral imagery. However, the high computational cost of designing the transform matrix impedes the adoption of PCA-based transforms in practice. In this paper, we have investigated several approaches for reducing computational complexity of PCA. Most straightforwardly, one can subsample the hyperspectral image spatially before calculating the covariance matrix as needed in the traditional eigendecomposition procedure for PCA design; such spatial subsampling produces a substantial reduction in complexity with the cost of a transform that only approximates the true spectral PCA of the dataset. Additionally, the spectral dimension can be decreased by explicitly removing the water-absorption bands and other bands of low sensor quality before PCA is applied. Finally, the cost of computing the covariance matrix can be avoided entirely by a direct iterative calculation of the eigenvectors. Such an IPCA approach is feasible when the number of PCs to be determined is small, as such may be the case for coding at low bitrates. The experimental results we have presented reveal that both rate-distortion performance as well as data-analysis performance in the form of anomaly detection incur little impact as a result of using the low-complexity PCA implementations considered here.

Acknowledgment

The authors thank Mr. Wei Zhu for various contributions to the work presented here.

References

Baldi, P. F. and K. Hornik (1995, July). Learning in linear neural networks: A survey. *IEEE Transactions on Neural Networks 6*(4), 837–857.

Chang, C.-I. and S.-S. Chiang (2002, June). Anomaly detection and classification for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing 40*(6), 1314–1325.

Chang, C.-I. and Q. Du (2004, March). Estimation of number of spectrally distinct signal sources in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing 42*(3), 608–619.

Diamantaras, K. I. and S. Y. Kung (1996). *Principal Component Neural Networks: Theory and Applications*. New York, NY: John Wiley & Sons, Inc.

Dremmel, J. E. (1997). *Applied Numerical Linear Algebra*. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Du, Q. and J. E. Fowler (2007, April). Hyperspectral image compression using JPEG2000 and principal component analysis. *IEEE Geoscience and Remote Sensing Letters 4*(2), 201–205.

Fleury, M., R. P. Self, and A. C. Downton (2004, April). Development of a fine-grained parallel Karhunen-Loève transform. *Journal of Parallel and Distributed Computing 64*(4), 520–535.

Fowler, J. E. and J. T. Rucker (2007). 3D wavelet-based compression of hyperspectral imagery. In C.-I. Chang (Ed.), *Hyperspectral Data Exploitation: Theory and Applications*, Chapter 14, pp. 379–407. Hoboken, NJ: John Wiley & Sons, Inc.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation* (2 ed.). Upper Saddle River, NJ: Prentice Hall.

Hirai, Y. and K. Nishizawa (2000, July). Hardware implementation of a PCA learning network by an asynchronous PDM digital circuit. In *Proceedings of the International Joint Conference on Neural Networks*, Volume 2, Como, Italy, pp. 65–70.

Hotelling, H. (1933, September). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology 24*(6), 417–441.

ISO/IEC 15444-2 (2004). *Information Technology—JPEG 2000 Image Coding System—Part 2: Extensions*. ISO/IEC 15444-2.

Kim, B.-J., Z. Xiong, and W. A. Pearlman (2000, December). 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 10*(8), 1374–1387.

Kulkarni, P., A. Bilgin, M. W. Marcellin, J. C. Dagher, J. H. Kasner, T. J. Flohr, and J. C. Rountree (2006). Compression of earth science data with JPEG2000. In G. Motta, F. Rizzo, and J. A. Storer (Eds.), *Hyperspectral Image Compression*, Chapter 12, pp. 347–378. New York: Springer.

Kumar, A. P., V. Kamakoti, and S. Das (2007, February). System-on-programmable-chip implementation for online face recognition. *Pattern Recognition Letters 28*(3), 342–349.

Makhoul, J., S. Roucos, and H. Gish (1985). Vector quantization in speech coding. *Proceedings of the IEEE 73*(11), 1551–1588.

Murahashi, Y., H. Hotta, S. Doki, and S. Okuma (2004, July). Pulsed neural networks based on delta-sigma modulation suitable for hardware implementation. In *Proceedings of the International Joint Conference on Neural Networks*, Volume 4, Budapest, Hungary, pp. 2607–2612.

Pearlman, W. A., A. Islam, N. Nagaraj, and A. Said (2004, November). Efficient, low-complexity image coding with a set-partitioning embedded block coder. *IEEE Transactions on Circuits and Systems for Video Technology 14*(11), 1219–1235.

Penna, B., T. Tillo, E. Magli, and G. Olmo (2006a, August). A new low complexity KLT for lossy hyperspectral data compression. In *Proceedings of the International Geoscience and Remote Sensing Symposium*, Volume 7, Denver, CO, pp. 3525–3528.

Penna, B., T. Tillo, E. Magli, and G. Olmo (2006b, January). Progressive 3-D coding of hyperspectral images based on JPEG 2000. *IEEE Geoscience and Remote Sensing Letters 3*(1), 125–129.

Pickering, M. R. and M. J. Ryan (2006). An architecture for the compression of hyperspectral imagery. In G. Motta, F. Rizzo, and J. A. Storer (Eds.), *Hyperspectral Image Compression*, Chapter 1, pp. 1–34. New York: Springer.

Prasanna, C. S. S., N. Sudha, and V. Kamakoti (2005, January). A principal component neural network-based face recognition system and its ASIC implementation. In *Proceedings of 18th International Conference on VLSI Design*, Kolkata, India, pp. 795–798.

Reed, I. S. and X. Yu (1990, October). Adaptive multiple-band CFAR detection of an optical patterm with unknown spectral distribution. *IEEE Transactions on Acoustics, Speech, and Signal Processing 38*(10), 1760–1770.

Rucker, J. T., J. E. Fowler, and N. H. Younan (2005, July). JPEG2000 coding strategies for hyperspectral data. In *Proceedings of the International Geoscience and Remote Sensing Symposium*, Volume 1, Seoul, South Korea, pp. 128–131.

Said, A. and W. A. Pearlman (1996, June). A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology 6*(3), 243–250.

Sanger, T. D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks 2*(6), 459–473.

Schreiber, R. (1986, April). Solving eigenvalue and singular value problems on an undersized systolic array. *SIAM Journal on Scientific and Statistical Computing 7*(2), 441–451.

Subramanian, S., N. Gat, A. Ratcliff, and M. Eismann (2000, February). Real-time hyperspectral data compression using principal components transformation. In *Proceedings of the AVIRIS Earth Science & Applications Workshop*, Pasadena, CA.

Tang, X. and W. A. Pearlman (2005, March). Scalable hyperspectral image coding. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Volume 2, Philadelphia, PA, pp. 401–404.

Taubman, D. S. and M. W. Marcellin (2002). *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Boston, MA: Kluwer Academic Publishers.

Zhu, W. (2007). On the performance of JPEG2000 and principal component analysis in hyperspectral image compression. Master's thesis, Mississippi State University.

PLACE
PHOTO
HERE

Qian Du received the Ph.D. degree in electrical engineering from the University of Maryland Baltimore County, Baltimore, in 2000. From 2000 to 2004, she was with the Department of Electrical Engineering and Computer Science, Texas A&M University–Kingsville. In fall of 2004, she joined the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, where she is currently an associate professor. Her research interests include remote sensing image analysis, pattern classification, data compression, and neural networks. Dr. Du is a member of IEEE, SPIE, ASPRS, and ASEE.

PLACE
PHOTO
HERE

James E. Fowler received the B.S. degree in computer and information science engineering and the M.S. and Ph.D. degrees in electrical engineering in 1990, 1992, and 1996, respectively, all from the Ohio State University.

In 1995, Dr. Fowler was an intern researcher at AT&T Labs in Holmdel, NJ, and, in 1997, he held an NSF-sponsored postdoctoral assignment at the Université de Nice–Sophia Antipolis, France. In 2004, he was a visiting professor in the Département TSI at École Nationale Supérieure des Télécommunications (ENST), Paris, France. He is currently a professor in the Department of Electrical & Computer Engineering at Mississippi State University in Starkville, MS, and is also a researcher in the GeoResources Institute (GRI) at Mississippi State.

Dr. Fowler is an Associate Editor for IEEE SIGNAL PROCESSING LETTERS, an Associate Editor for the INTERNATIONAL JOURNAL OF IMAGE & VIDEO PROCESSING, and a member of the program committee for the DATA COMPRESSION CONFERENCE.

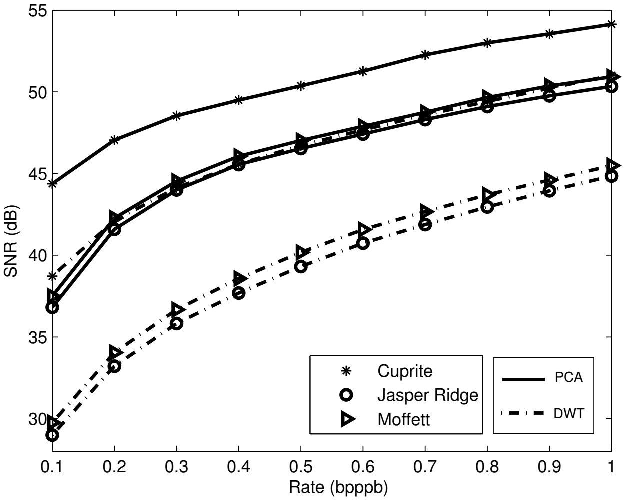


Fig. 1. Rate-distortion performance for JPEG2000 using a PCA-based spectral transform (PCA+JPEG2000) and a DWT-based spectral transform (DWT+JPEG2000).

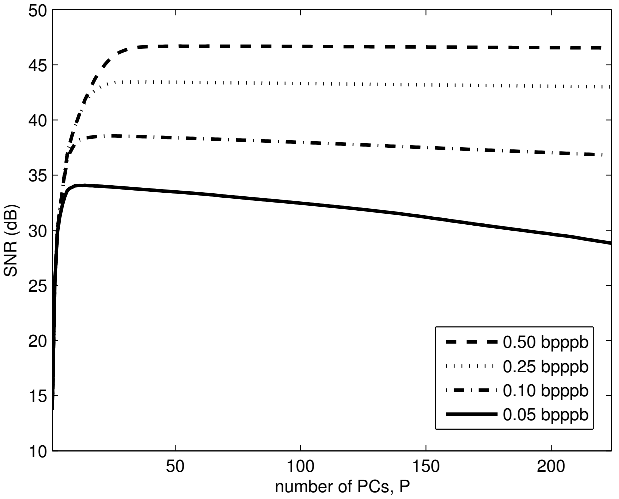


Fig. 2. Impact of the number of PCs on SNR for PCA+JPEG2000 coding of Jasper Ridge.

TABLE I

SNR Performance (in dB) at Various Rates for Coders Using a 5-Level Spectral DWT Followed by a 5-Level 2D Spatial DWT Using the Popular 9/7 Biorthogonal Wavelets.

	0.05 bpppb			0.1 bpppb			0.25 bpppb			0.5 bpppb		
	JPEG2000	SPECK	SPIHT	JPEG2000	SPECK	SPIHT	JPEG2000	SPECK	SPIHT	JPEG2000	SPECK	SPIHT
Cuprite	35.2	34.7	34.9	38.7	38.3	38.5	43.2	42.9	43.0	46.7	46.4	46.5
Jasper Ridge	25.2	24.9	24.8	29.0	28.6	28.5	34.6	34.2	34.2	39.3	38.9	39.0
Moffett	26.0	25.6	25.6	29.8	29.3	29.3	35.5	35.1	35.2	40.2	39.9	39.9

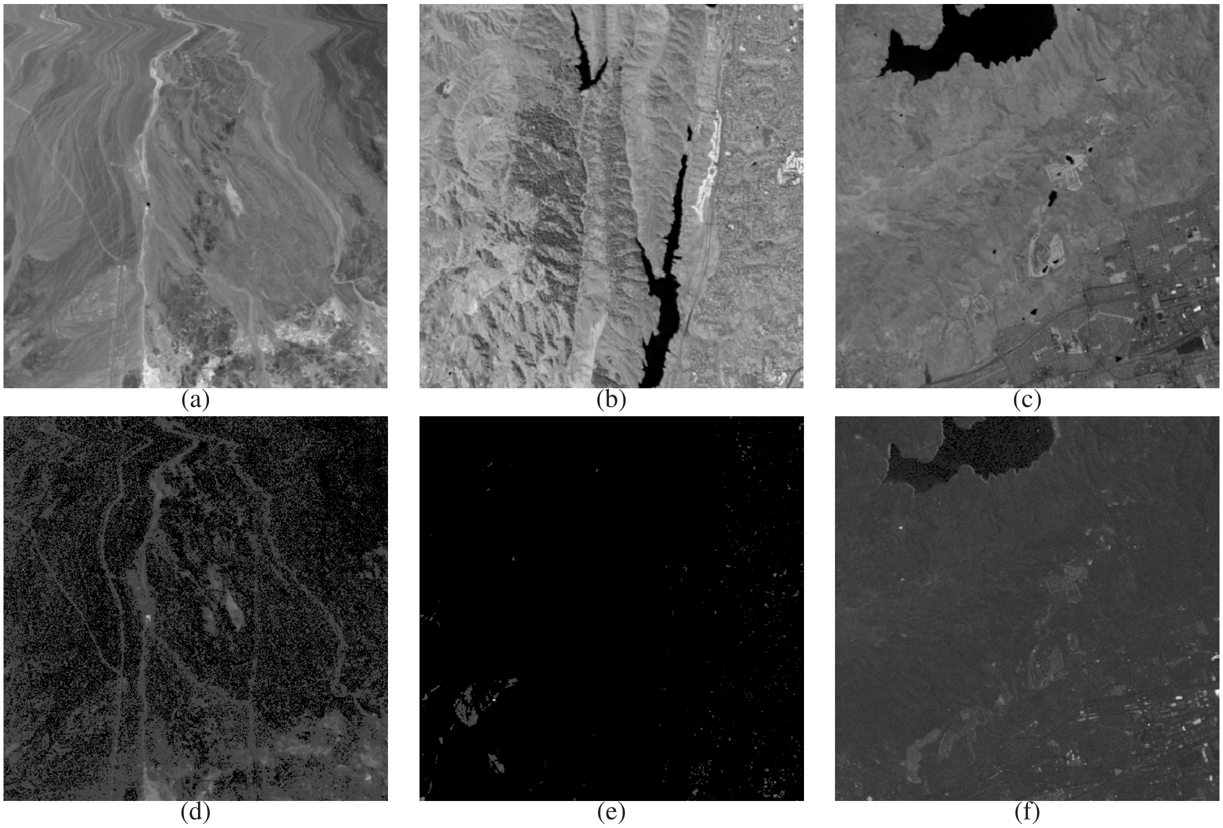
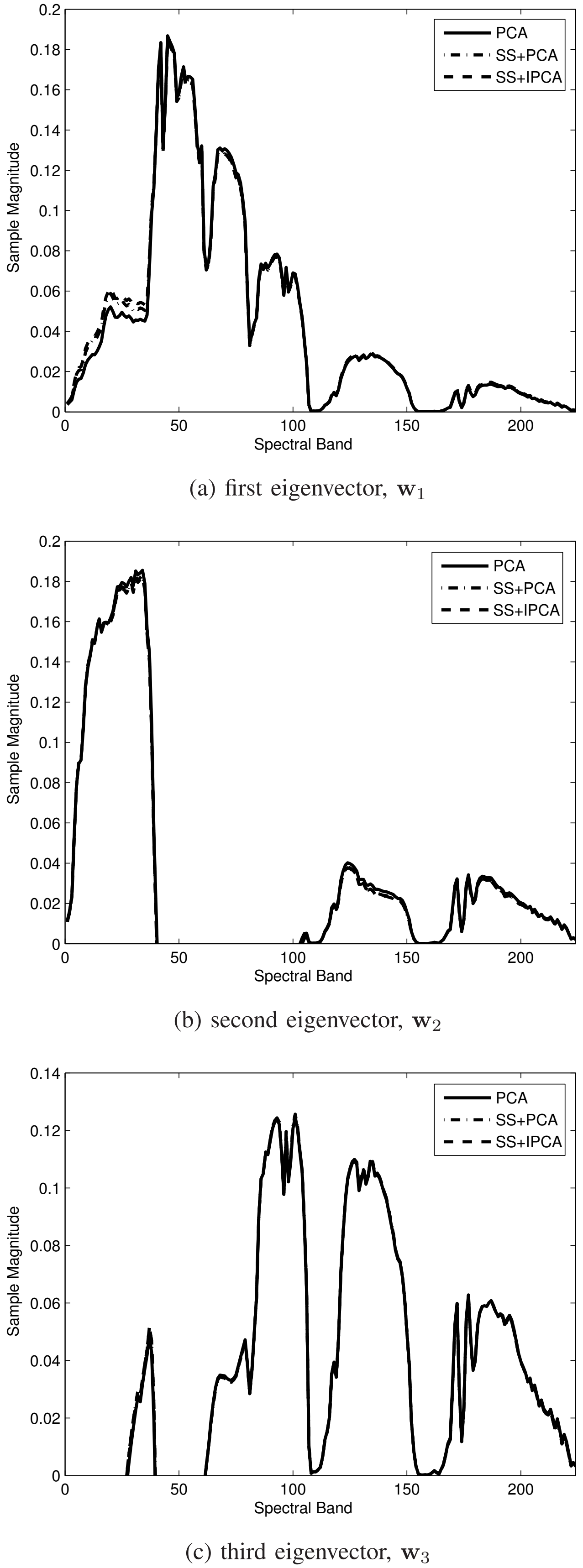


Fig. 3. (a)–(c) Band 100 from original images. (d)–(f) Corresponding anomaly-detection maps (contrast-enhanced for visualization purposes). (a) & (d) Cuprite; (b) & (e) Jasper Ridge; (c) & (f) Moffett.



(a) first eigenvector, $\mathbf{w}_1$

(b) second eigenvector, $\mathbf{w}_2$

(c) third eigenvector, $\mathbf{w}_3$

Fig. 4. Comparison of the first three eigenvectors for the Jasper Ridge dataset.

TABLE II

SNR Performance (in dB) of PCA+JPEG2000 With Spectral Transform Designed Via Low-Complexity PCA

		0.05 bpppb		0.1 bpppb		0.25 bpppb	
#PCs		$\widetilde{P}^* = 21$	all	$\widetilde{P}^* = 25$	all	$\widetilde{P}^* = 38$	all
Cuprite	PCA	43.1	40.0	45.3	44.4	48.2	47.9
	SS+PCA	43.1	40.0	45.1	44.3	48.1	47.8
	SS+IPCA	43.1	–	45.2	–	48.1	–
Jasper Ridge	PCA	34.0	28.8	38.6	36.8	43.4	43.0
	SS+PCA	34.0	28.7	38.5	36.8	43.4	43.0
	SS+IPCA	33.9	–	38.5	–	43.3	–
Moffett	PCA	43.1	40.0	45.3	44.4	48.2	47.9
	SS+PCA	43.1	40.0	45.1	44.3	48.1	47.8
	SS+IPCA	43.1	–	45.2	–	48.1	–

$\widetilde{P}^*$ is the number of PCs from the linear model of (Du and Fowler 2007); “all” refers to retaining all $N = 224$ PCs

TABLE III

Anomaly-Detection Performance Measured Via Correlation Coefficient ρ of PCA+JPEG2000 With Spectral Transform Designed Via Low-Complexity PCA

		0.05 bpppb		0.1 bpppb		0.25 bpppb	
#PCs		$\widetilde{P}^* = 21$	all	$\widetilde{P}^* = 25$	all	$\widetilde{P}^* = 38$	all
Cuprite	PCA	0.73	0.69	0.75	0.65	0.75	0.69
	SS+PCA	0.73	0.70	0.75	0.70	0.75	0.71
	SS+IPCA	0.73	–	0.75	–	0.76	–
Jasper Ridge	PCA	0.96	0.88	0.97	0.95	0.98	0.91
	SS+PCA	0.95	0.86	0.97	0.66	0.98	0.64
	SS+IPCA	0.95	–	0.97	–	0.98	–
Moffett	PCA	0.77	0.59	0.86	0.74	0.87	0.43
	SS+PCA	0.75	0.64	0.84	0.67	0.87	0.59
	SS+IPCA	0.77	–	0.84	–	0.88	–

$\widetilde{P}^*$ is the number of PCs from the linear model of (Du and Fowler 2007); “all” refers to retaining all $N = 224$ PCs

TABLE IV

Performance for Bad-Band Removal for PCA+JPEG2000

		0.05 bpppb		0.1 bpppb		0.25 bpppb		0.5 bpppb	
bands coded		224	195	224	195	224	195	224	195
Cuprite	MSE	320.12	312.85	107.70	104.52	56.33	52.99	32.09	32.25
	ρ	0.69	0.69	0.65	0.64	0.69	0.69	0.65	0.65
Jasper Ridge	MSE	1996.74	1966.86	285.11	328.58	69.96	69.20	32.90	33.33
	ρ	0.88	0.86	0.95	0.94	0.91	0.89	0.90	0.90
Moffett	MSE	2375.07	2331.31	314.62	359.77	77.96	75.50	37.64	37.65
	ρ	0.59	0.66	0.74	0.76	0.43	0.44	0.77	0.83