Cost and Scalability Improvements to the Karhunen-Loêve Transform for Remote-Sensing Image Coding

Ian Blanes, *Student Member, IEEE,* Joan Serra-Sagristà, *Member, IEEE*

***Abstract*—The Karhunen-Loêve Transform is widely used in hyperspectral image compression because of its high spectral decorrelation properties. However, its use entails a very high computational cost. To overcome this computational cost and to increase its scalability, in this paper we introduce a multi-level clustering approach for the Karhunen-Loêve Transform.**

As the set of different multi-level clustering structures is very large, a two-stage process is used to carefully pick the best members for each specific situation. First, several candidate structures are generated through local search and eigen-thresholding methods, and then, candidates are further screened to select the best clustering configuration.

Two multi-level clustering combinations are proposed for hyperspectral image compression: one with the coding performance of the Karhunen-Loêve Transform, but with much lower computational requirements and increased scalability, and another one that outperforms a lossy wavelet transform, as spectral decorrelator, in quality, cost, and scalability.

Extensive experimental validation is performed, with images from both the AVIRIS and Hyperion sets, and with JPEG2000, 3d-TCE, and CCSDS-IDC as image coders. Experiments also include classification-based results produced by k-means clustering and RX anomaly detection.

***Index Terms*—Low-cost, Karhunen-Loêve Transform, hyperspectral data coding, progressive lossy-to-lossless and lossy compression, component scalability.**

I. Introduction

CURRENT remote sensors gather very large amounts of information: they capture images with sizes ranging from a few megabytes to the gigabyte, and they produce lots of such images, increasing the amount of information produced, which is expensive to store, and slow to manipulate and transmit. Image compression techniques are capable of substantially reducing image sizes, decreasing costs and improving user interaction with the information.

A common kind of images that remote sensors acquire are hyperspectral images, those with a few hundreds of spectral components. In hyperspectral images, each of these components represents the light of a given wavelength range for the same spatial location, such that components are highly correlated, and redundancy exist between them. An appropriate transform applied in the spectral domain removes this redundancy, and improves compression performance.

This work was supported in part by the Spanish and Catalan Governments, and by FEDER, under grants TIN2009-14426-C02-01, TIN2009-05737-E, SGR2009-1224, and FPU2008. Computational resources used in this work were partially provided by the Oliba Project of the Universitat Autònoma de Barcelona.

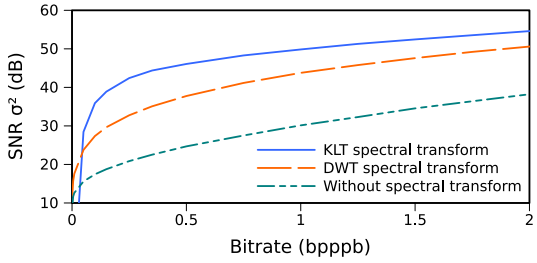


Figure 1. The NASA AVIRIS Low Altitude image coded with JPEG2000, with and without spectral transforms.

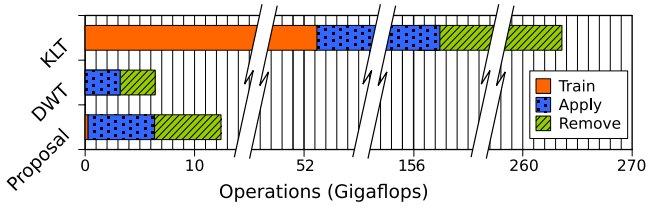


Figure 2. Spectral transform cost for an image of size $224 \times 2048 \times 512$ ($z \times y \times x$). Lossy compression.

Common spectral transforms are the Discrete Wavelet Transform (DWT) and the Karhunen-Loêve Transform (KLT). Their use provides substantial compression improvements, with the KLT yielding significantly better results [1]. However, the better performance of the KLT comes at the price of an increase of two orders of magnitude in computational cost, and at the price of not being scalable, which is the ability of reverting the transform partially. Fig. 1 shows an example of the quality improvements produced by spectral transforms, when used in combination with the JPEG2000 coding system standard.

In this article, we propose a family of variations to the KLT, which will bring computational costs and scalabilities comparable to those of the DWT, but with almost the same compression performance as the original KLT.

Computational costs are an important issue to take into account; for example, on a theoretical 1 Gflops CPU, applying and removing a KLT transform on an image of size $224 \times 2048 \times 512$ ($z \times y \times x$) takes more than 4 minutes. In contrast, it only takes 6 seconds if a DWT is employed as spectral transform. A cost comparison among the KLT, the DWT, and our proposal, when used as spectral decorrelators on an image of this size, is provided in Fig. 2.

High computational costs are also an issue in the Progressive Lossy-to-Lossless (PLL) scenario, in which the coder used produces lossless encodings when enough bytes are allocated, but, as opposed to a pure lossless method, it also provides lower quality versions of the image when not enough bytes are available. In PLL, the Reversible Karhunen-Loêve Transform (RKLT) [2], an approximation of the KLT, has to be employed. For the Reversible KLT, similar cost and scalability improvements will also be shown, and in this case, the RKLT will be compared to a common reversible wavelet, the Integer Wavelet Transform (IWT).

Several improvements on the computational cost of KLT are found in the literature, starting with the Discrete Cosine Transform (DCT), which was introduced as a low-cost alternative to the KLT [3]. The cost reduction stems from the additional assumption of a first-order Markov model. While still widely used in image coding, the DCT performs poorly as spectral decorrelator. Later, the fast Approximate Karhunen-Loêve Transform (AKLT) was introduced [4]. It is based on linear perturbation theory and uses the DCT as the related problem. It was also further improved as the AKLT_2 [5], which extends the AKLT with second order perturbations.

Other strategies for cost reduction try to reduce transform training costs, which amount approximately to one fifth of the total cost. In [1], it is proposed to compute the empirical covariance matrix from a small sample of the whole input, and a subsampling factor of 1% is shown to be adequate for hyperspectral images. This strategy is compatible with our proposal, and will be discussed later. Another strategy to reduce training computational costs is to train once the KLT for a training set of images [6], and afterwards employ the already trained transform for all images, although, with this strategy, the transform is not specifically trained for each image and its coding performance is worse.

On the other hand, the RKLT is a recent subject, and there are few improvements that alleviate its cost. In [7], a recursive clustered structure is used to reduce the computational cost of the RKLT in lossless coding of electroencephalograms, and in [8] RKLT is used for lossless compression of Magnetic Resonance Imaging (MRI) images. Similarly, we developed a multi-level clustered KLT structure for PLL [9]. This structure yields greater savings, and leads to the family of structures discussed in this work.

Two are the objectives of this paper: to reduce the computational cost of the KLT to an acceptable level, and to add component scalability to the transform, both for lossy and PLL image coding. To that purpose, the family of multi-level KLT clustering structures has been analyzed, and a procedure to pick appropriate transforms has been devised. In addition, very extensive experimental testing has been performed for a wide range of hyperspectral images, image coders, and quality measures.

We should end this discussion by saying that our proposal for reducing the computational cost of the KLT may still be too high for its practical use for onboard compression. As noted in [10], most image coding algorithms have excessive complexity for onboard compression, and lossless predictive compression methods are preferred. However, for on-the-ground compression, our proposal provides very competitive computational cost, improved component scalability, and yields high coding and classification-based performance.

This article is organized as follows. Section II introduces the proposed family of variations of the KLT. Then, Section III describes a procedure to select notable elements of the proposed family and describes how to measure their suitability. Section IV contains the experimental results. Section V draws some conclusions.

II. Low-cost scalable Karhunen-Loêve Transform

The KLT is a linear transform, constrained to provide stochastic independence for centered Gaussian sources, and, in this sense, optimality. This comes at the price of having to apply a different transform for each set of sources.

Let X be a matrix that has N rows, one for each source, and M columns, one for each sample. Then, Y, the outcome after applying the transform, is computed as

$$Y = \text{KLT}_{\Sigma_X}(X) = Q^T X, \tag{1}$$

where $\Sigma_X = \frac{1}{M} X X^T$ is the covariance matrix of X, and Q is the orthogonal matrix obtained from the Singular Value Decomposition (SVD) of $\Sigma_X = Q \Lambda Q^{-1}$ ($\Lambda = diag(\lambda_1, \ldots, \lambda_N)$, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$). Note that sources must be manually centered around zero if they do not have zero mean.

Once the transform matrix has been generated (or trained), the KLT application consists only of a matrix multiplication. If a lossy transform is desired, then the multiplication can be naturally applied using fixed-precision numbers. But, if exact recovery is needed, then additional precautions must be taken. The problems arise from the matrix multiplication, which is not exactly reversible because of the small imprecisions caused by the use of floating-point arithmetic. This situation can be resolved by decomposing the matrix multiplication into reversible lifting steps, which can be later applied and reversed without loss [11].

Originally, two lifting decomposition strategies were proposed, the TERM and the SERM [2], where the latter trades an increase of factorization time for less application costs. Another variant was proposed in [12], where the quantization noise introduced by the lifting steps was reduced substantially, but with higher computational costs. We call this latter one the 3TERM.

A. Multi-level clustering

Three are the main sources for the computational cost of the KLT: the covariance matrix calculation, and the forward and reverse application of the transform. The three operations have an effective computational complexity of $\mathcal{O}(n^2)$, where n is the number of spectral components. With this in mind, a divide and conquer approach seems appropriate, as for each division, the total cost is halved. Therefore, dividing a large transform into multiple clusters (or spectral tiles), and applying smaller transforms, yields significant reductions on all three cost sources.

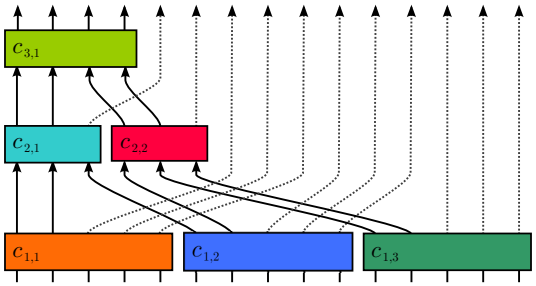


Figure 3. Example of a multi-level clustering KLT structure. This structure decorrelates 15 spectral components with 3 levels of clustering.

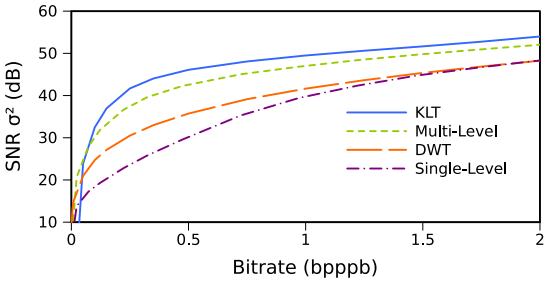


Figure 4. Comparison of compression performance for several spectral transforms: full KLT, DWT, Single-Level KLT with 56 clusters, and multi-level KLT with 56 clusters on the first level and successive levels with half the number of clusters. Image is AVIRIS Yellowstone (Sc 0).

A drawback of this simple approach is that it only provides local decorrelation within each cluster. However, global decorrelation can be achieved by applying additional transform stages, where only the "most important" parts are further decorrelated. This scheme is depicted in Fig. 3. We know from [9] that using a homogeneous multi-level KLT structure, where half of each cluster is forwarded to the next stage, is a good start. In Fig. 4, the benefits of a homogeneous multi-level structure can be observed. Nevertheless, as will be seen later, other structures can be used obtaining further cost reductions.

To define a specific multi-level clustering KLT structure, the notation $\mathcal{C} = \{c_{level,index}\}$ is used. $\mathcal{C}_l$ is the set of clusters in level l of the structure $\mathcal{C}$. For each cluster, let $\mathrm{Size}(c_{i,j})$ be the size of the cluster, and $\mathrm{Th}(c_{i,j})$ the number of elements in said cluster that proceed to the next level. As an example, Fig. 3 scheme is defined by the set $\mathcal{C} = \{c_{1,1}, c_{1,2}, c_{1,3}, c_{2,1}, c_{2,2}, c_{3,1}\}$, with, for instance, $\mathrm{Size}(c_{1,2}) = 5$ and $\mathrm{Th}(c_{2,1}) = 2$.

Due to the large number of possible structures, choosing a specific multi-level structure from all the possible ones is not a trivial problem. For example, for as few as 16 components, the total number of possible structures[1] is $8.77 \cdot 10^{26}$. Moreover, structure parts cannot be selected independently, as the performance of parts of one level depends on the configuration of the parts of the previous level.

How to select a structure as a whole for each specific situation is addressed in Section III, while how to select individual structure parts is addressed here.

[1]Multi-level clustering structures for n components can be counted with $\mathrm{g}(n) = 1 + \sum_{t=2}^{n-1} \mathrm{f}(n,t)\mathrm{g}(t)$, where $\mathrm{f}(n,t)$ is the number of one-level clustering configurations of n components that forward t components to the next level.

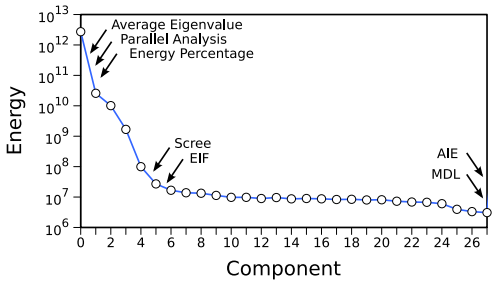


Figure 5. Comparison of the different eigen-thresholding methods. KLT applied on a cluster of 28 components of the AVIRIS image Moffett Field.

The structure specification can be divided into two kinds of problems: determine the size of each cluster in the structure, and, for each cluster, set the threshold on how many components are forwarded to the next stage.

For the first problem, cluster size determination, a local search may be reasonable, as the number of possible cluster sizes for one level is relatively small. Moreover, search cost can be reduced if some constraints are added to cluster sizes, such as homogeneity within a level.

The second problem, deciding how many components are forwarded to the next stage, can be effectively linked to eigen-thresholding methods, which try to determine the amount of components to retain after a KLT, precisely our situation. The set of different eigen-thresholding methods is very large. The ones found to be more relevant to this research are:

Scree test [13]. It is a subjective test based on the visual inspection of a plot of the eigenvalues. The threshold is set where the viewer determines there is the change of trend (*i.e.*, the "elbow" of the plot).

Average Eigenvalue (AE). Components with above-average eigenvalues are selected. Usually only the first component is selected, on occasions two.

Empirical Indicator Function (EIF) [14]. Malinowski discovered a simple function (Eq. 2) that predicted the correct number of factors in his chemical experiments. It yields results very similar to the scree test. It might be a good automated replacement.

$$EIF = \underset{j=0..N}{\operatorname{argmin}} \sqrt{\frac{\sum_{j<i} \lambda_i}{M(N-j)^3}} \tag{2}$$

The following methods were also examined, but had to be discarded due to more efficient alternatives, sensitivity to non-Gaussian noise, or inability to operate on zero-mean data: Energy Percentage (EP), Parallel Analysis (PA) [15], Akaike Information Criterion (AIE) [16], Minimum Description Length (MDL) [17], [18], and Harsanyi-Farrand-Chang (HFC) [19].

Fig. 5 provides an illustrative comparison of the different eigen-thresholding methods. The figure depicts the energy accumulation for an example cluster after a KLT transform, and indicates, according to each method, how many components should be forwarded to the next level.

In the relevant eigen-thresholding methods, two clear trends are observed. The first trend corresponds to the AE method,

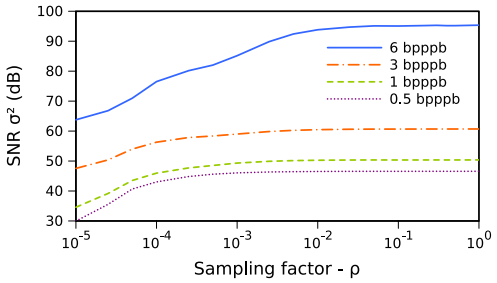


Figure 6. Effects of image subsampling for covariance calculation. The AVIRIS image Jasper Ridge has been compressed at different bitrates with JPEG2000 on lossy mode.

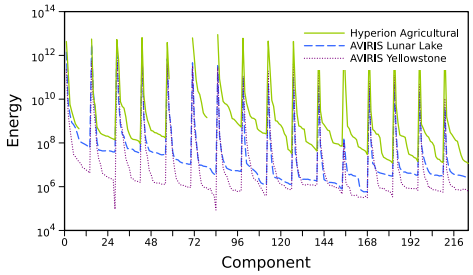


Figure 7. Energy accumulation for two AVIRIS and one Hyperion images. One level of clustering is applied, with 16 clusters, each of 14 components.

where one component is selected most of the times, and, on rare occasions, two components are selected. The other trend corresponds to the scree test and EIF. The scree test has the disadvantage that human intervention is required every time a threshold needs to be determined.

B. Subsampling

Image subsampling in the KLT has been known to substantially reduce the training cost of the KLT [1]. The aim is to use only a small sample of the image spatial locations to compute the covariance matrix. Subsampling factors of $\rho = 0.01$ (1%) are known to provide almost the same compression performance results on unclustered transforms.

For clustered transforms, image subsampling in the KLT training also provides substantial cost reduction. It simply needs to be applied in each of the clusters, since with clustered transforms the number of spatial samples does not change.

We have reproduced the experiments in [1], and also found $\rho = 0.01$ to be an adequate subsampling factor for the multi-level clustering KLT approach. However, as shown in Fig. 6, this factor is bitrate-dependent and could be further decreased for low bitrates. For hyperspectral images, a ρ factor of 0.01 effectively reduces the cost to a small fraction, and leaves application and removal as the main sources of cost.

III. Strategy for candidate selection

In this section we formulate how to select which cluster combinations to explore from all the possible ones, and how to evaluate their suitability.

Strong similarities have been observed on the performance of each multi-level clustered KLT structure across the different imagery used. Fig. 7 shows how the images used in the next section have similar behaviors. While the energy at each component is not equal for the three different images, peaks have the same width, which is a key aspect that allows result extrapolation.

Two strategies for structure selection are proposed. Both strategies produce various candidate structures for each specific situation; further screening leads to the selection of very effective transforms. The idea behind both strategies is to examine each cluster size, from 4 to 28, and forward as many components as the various eigen-thresholding methods suggest, producing a list of different candidate structures.

To reduce the solution space (*i.e.*, candidate list) three constraints are added. First, a constraint of cluster size homogeneity has been added to each clustering level:

$$\forall\, l, i, j,\ \mathrm{Size}(c_{l,i}) = \mathrm{Size}(c_{l,j}).$$

Second, non-regular structures where the cluster number does not divide the number of components are not examined, as they can be easily adapted from regular structures, using one cluster with fewer components. And third, structures with more than one cluster on the top level are discarded, as early results showed huge quality penalties.

As described below, strategies differ on whether a dynamic eigen-thresholding method is used or not:

- Given the previous restrictions, if thresholds are known *a priori*, the result is a fully **static structure**, where the whole structure can be chosen before it is applied, and does not vary with different images.
  To define static structures, we set the same threshold value for each cluster of a level, based on the average output of either the EIF or the AE methods for a training image. As all the clusters in a level have the same size and threshold, static structures can be identified with a list of L elements, corresponding to the number of clusters of each level, their sizes and their thresholds:

  $$\Big[\big(\#\mathcal{C}_1, \mathrm{Size}(c_1), \mathrm{Th}(c_1)\big), \ldots, \big(\#\mathcal{C}_{L-1}, \mathrm{Size}(c_{L-1}), \mathrm{Th}(c_{L-1})\big), \big(\#\mathcal{C}_L, \mathrm{Size}(c_L), 0\big)\Big],$$

  where c_l is any cluster of level l.
  For example, the structure $[(2,5,3),(1,6,0)]$ has two clusters of 5 components on the first level. From each cluster of the first level, 3 components are forwarded to the second level, where the structure only has one cluster of 6 components.
  This notation has the property that two structures with the same prefix in its notation also share the same stages until that point, which is useful to enumerate the possible structures for a corpus of images. In Fig. 8, a tree is shown, where each leaf is a different clustering structure, and each structure is identified by the sequence of integers corresponding to its path from the root to the leaf.
- On the other hand, a **dynamic structure** is produced if either AE or EIF are used to set the threshold individually for each cluster on the fly, and without any training.

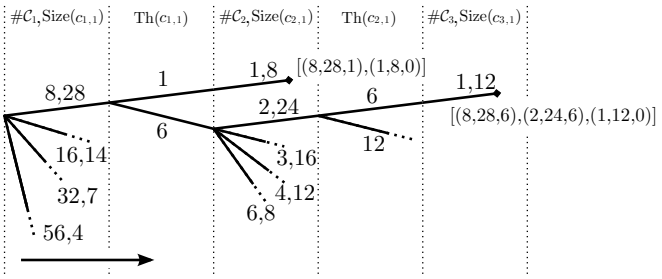


Figure 8. A tree represents all the possible static structures for a specific corpus. This is a piece of the tree for hyperspectral images of 224 components.

Table II
TECHNICAL NAMES FOR AVIRIS AND HYPERION IMAGES USED.

Name	Technical Name	Size ($x \times y \times z$)
AVIRIS Cuprite	f970619t01p02_r02	$614 \times 2206 \times 224$
AVIRIS Jasper Ridge	f970403t01p02_r03	$614 \times 2586 \times 224$
AVIRIS Low Altitude	f960705t01p02_r05	$614 \times 3689 \times 224$
AVIRIS Lunar Lake	f970623t01p02_r07	$614 \times 1431 \times 224$
AVIRIS Moffett Field	f970620t01p02_r03	$614 \times 2031 \times 224$
AVIRIS Yellowstone (Sc 0)	f060925t01p00_r12_sc00	$677 \times 512 \times 224$
AVIRIS Yellowstone (Sc 11)	f060925t01p00_r12_sc11	$677 \times 512 \times 224$
Hyperion Agricultural	EO1H0280342002068110PX	$256 \times 2905 \times 224$
Hyperion Coastal	EO1H0140342002050110PY	$256 \times 2905 \times 224$
Hyperion Coral Reef	EO1H0830742003120110PW	$256 \times 3127 \times 224$
Hyperion Tornado	EO1H0150332002121112PF	$256 \times 3352 \times 224$
Hyperion Urban	EO1H0440342002212110PY	$256 \times 2905 \times 224$

Only structures with all clusters of the same size for all levels are selected (as opposed to static structures, where sizes are allowed to vary between levels). After the first level of clusters, a thresholding method is used, and then, in the next level, enough clusters are allocated to treat all the forwarded components. This scheme is repeated until there is just one cluster in the top level.
Two convenient tweaks are applied to the parameters. First, to guarantee a decreasing number of clusters at each level, thresholds are limited to half the number of components of a cluster. Second, if the number of clusters does not divide the number of forwarded components, then, additional components are selected, starting with those with more energy among all the clusters in that level.

Following the strategies described in this section, a list of candidate structures for hyperspectral images has been built. For static structures, the whole tree, partially depicted in Fig. 8, has been manually expanded, generating more than 200 nodes, and 68 valid structures. As for dynamic structures, there are 8 sub-strategies to create one candidate structure for a given image, that differ on thresholding method (AE, EIF) and cluster size (4,7,14,28).

Once a list of candidate structures has been generated, each candidate needs to be tested against the others to determine the most suitable one. The suitability of a structure will be determined by three criteria: quality, cost, and scalability. Each criterion will be evaluated as follows:

- Quality is evaluated with the Signal-to-Noise Ratio[2] (SNR) produced by the tested structure in relation to the SNR produced by the full KLT transform. In Progressive Lossy-to-Lossless scenario, the relative difference of Compression-Ratio (CR) for lossless compression will also be considered.
- Cost is calculated counting the total floating-point operations required to apply and remove a transform. Table I provides operation counts for one cluster, for the lossy case, and for the three lossless approximations. The total cost of a transform, for N components applied to M spatial locations, can be computed, as shown in Eq. 3, by adding each cluster cost, and the cost of the additional stage to guarantee zero-mean.

$$\mathrm{TotalCost}(\mathcal{C}) = 3MN + \sum_{c \in \mathcal{C}} \mathrm{ClusterCost}(c) \qquad (3)$$

- Scalability has been evaluated measuring the dependencies to decode only one component. In other words, the number of components required for the inverse transform. As the KLT is not scalable at all, clusters must be completely reversed to recover one of its components. Also, clusters with dependencies first need their dependencies reversed.

Each candidate structure has been evaluated according to the previously set criteria, and results of each are discussed in the next section.

IV. EXPERIMENTAL RESULTS

Two sets of hyperspectral imagery have been used to produce experimental coding results. One is the AVIRIS corpus, frequently used in image coding [21], and the other is the corpus from the space-borne Hyperion sensor of the EO-1 program [22]. Images of both sets are signed and have 16 bits per sample. The spectral size of the Hyperion images is 242 components, but the last 18 are uncalibrated, and have been discarded, thus matching the spectral size of 224 components of the AVIRIS images. AVIRIS images have been cropped to a size of $224 \times 512 \times 512$ ($z \times y \times x$), while Hyperion images have not been cropped, and have a fixed width of 256 columns and a variable height ranging from 2905 to 3352 rows. For readability, images are referenced by their short name, and full technical names and details are provided in Table II.

The multi-level clustered versions of the KLT and RKLT are only proposed as a spectral transform, one of the coding stages of a coding process, hence, to properly measure their appropriateness as spectral transforms, the compression performance of the whole coding process has to be assessed. The coders selected for this purpose are: JPEG2000 [23], 3d-TCE [24], and TER [25], an improved version of Consultative Committee for Space Data Systems (CCSDS) recommendation for Image Data Compression [26]. The first one is a standardized multipurpose image coder, which includes support for arbitrary spectral transforms. 3d-TCE is an embedded tarp-based coder, with a performance similar to JPEG2000. And the last one, TER, enhances the CCSDS recommendation, highly

[2]As defined by $\mathrm{SNR} = 10 \log_{10}(\frac{\sigma^2}{\mathrm{MSE}})$.

Table I
DETAIL OF CLUSTER COST (IN FLOPS). ρ IS THE SUBSAMPLING FACTOR, $n = \text{Size}(c)$ IS THE NUMBER OF COMPONENTS OF THE CLUSTER, AND M IS THE NUMBER OF SPATIAL LOCATIONS OF THE ORIGINAL IMAGE. SVD COST IS FROM [20], AND TERM/SERM APPLICATION COSTS ARE FROM [2].

	KLT Variant			
Stage	Lossy KLT	RKLT TERM	RKLT SERM	RKLT 3TERM
Covariance Matrix	$\rho M(n^2+n+4)$			
Singular Value Decomposition	$\simeq 9n^3$			
Factorization	–	$n^3 - \frac{n^2}{2} + \frac{n}{2} - 1$	$\frac{11n^3}{3} - 5n^2 + \frac{7n}{3} - 1$	$\frac{5n^3}{3} + \frac{n^2}{2} - \frac{37n}{6} + 5$
Apply or Remove (done twice)	$M(2n^2-n)$	$M(2n^2+2n-3)$	$M(2n^2+n-1)$	$M(3n^2-3)$

focused on satellite image compression, with volumetric rate-distortion, quality scalability, and other features. All three coders are able to perform both lossy and PLL encodings.

Implementations chosen to perform the experiments are: Kakadu for JPEG2000 [27], QccPack for 3d-TCE [28], TER for CCSDS-IDC [29], and our own open-source implementation of the clustered KLT [30].

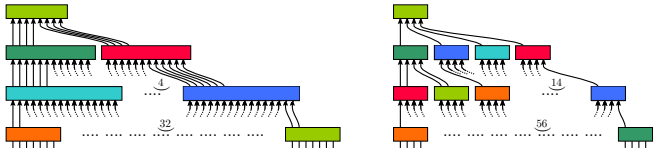
Candidate multi-level clustered KLT structures are evaluated, as spectral transforms, in comparison to the original KLT, and also to wavelets. The wavelet transforms used as spectral transforms are the Cohen-Daubechies-Feauveau (CDF) 9/7 for lossy coding, and the CDF 5/3 for PLL, both of them applied with 5 levels.

The remaining of this section is organized as follows: first the results for candidate multi-level KLT structures are discussed; then, Rate-Distortion (R-D) evolutions and CRs are analyzed for the best structures; and finally, some classification-based results are reported.

All the candidate structures have been tested in order to assess their properties. In Table III, results in relation to the KLT for the candidate structures are presented, and for each structure, measures of the three criteria proposed in the previous section are listed, for both lossy and PLL. As can be seen, dynamic thresholds provide the greater cost reductions and better scalabilities, while having a moderate quality penalty. On the other hand, static thresholds guarantee stable quality with still good computational savings. The most remarkable structures on the candidate list are the one with static thresholding with the following configuration: $[(32,7,2),(4,16,6),(2,12,4),(1,8,0)]$, and the dynamic one based on AE and a cluster size of 4 (AE size 4). The former achieves almost the same quality as the full KLT, while keeping cost lower than previous approaches, and still with a decent scalability. And the latter outperforms a lossy wavelet transform in all criteria, and is very competitive against lossless wavelets. Note that a static structure, due to its nature, might also be more amenable to hardware implementations.

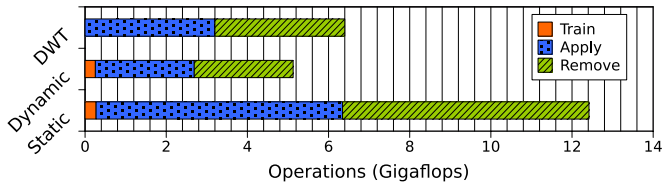
From the AVIRIS corpus, Yellowstone images were acquired in 2006 and calibrated with a different scaling factor [31]. For these Yellowstone images, there is a slightly better static structure: $[(32,7,3),(8,12,5),(2,20,7),(1,14,0)]$, which is capable of obtaining higher qualities with a small cost increase. However, for other images tested, this last structure performs similarly to the other structures, while it has a higher cost.

Further analysis of the performance of the two first structures follows. For brevity, from now on, they will be referred as static and dynamic. Details on each structure organization and costs can be found in Fig. 9 and Fig. 10, respectively.

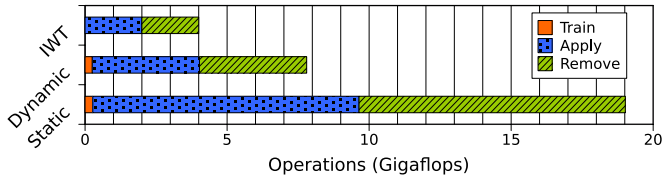


(a) Static (b) Dynamic for the AVIRIS image Cuprite

Figure 9. Best structures of multi-level KLT clustering for hyperspectral images according to the procedure described in Section III.



(a) Lossy



(b) PLL

Figure 10. Costs of the selected structures for an image of size $224 \times 2048 \times 512$ ($z \times y \times x$). $\rho = 0.01$.

Quality-evolution plots of the two selected structures are shown in Fig. 11. Proposed structures perform well, both in lossy and PLL, on all the images tested. Also, no significant changes are appreciated whether JPEG2000, TCE or CCSDS-IDC is used. As expected, the static structure yields almost the same performance as the original transform, besides, on very low bitrates it performs even better, due to the reduced amount of side information required by the smaller KLT matrices. On the Hyperion images, the reduction of side information is not so relevant, as the proportion of side information to the spatial size is lower, and still results are similar. Also as expected, the dynamic structure consistently outperforms the wavelet transform as spectral decorrelator in all the scenarios tested.

Lossless compression rates are reported on Table IV, where it can be observed that little impact is produced by the use of the static clustered approach (usually bitstream sizes differ less than 0.1 bpppb). On the other hand, a dynamic cluster

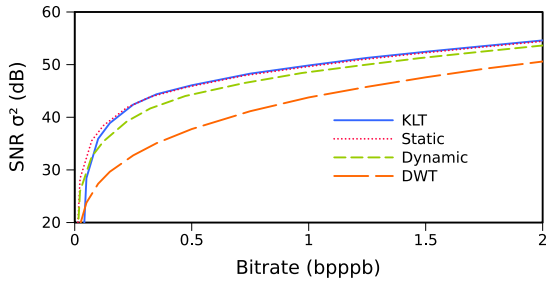
approach produces larger bitstreams, which have CRs between the RKLT and the IWT. Still, results are close to the ones of the RKLT for the dynamic structure. Note that the proposed transforms are also suitable to operate in near-lossless mode, where the peak absolute error is bounded, if a scheme similar to the ones proposed in [32], [33] is employed.

As for classification based results, two very common unsupervised classification techniques are used: k-means clustering [34], and Reed Xiaoli (RX) anomaly detection [35]. In this case, k-means has been configured to produce 10 clusters with spectral angle as distance function; for RX a confidence coefficient of $\gamma = 99\%$ has been set, and results are reported using Preservation of Classification (POC). Fig. 12 reports the performance of both classifiers when tested on images that have been compressed and uncompressed with PLL procedures. It is important to note that, for k-means, classes are more or less homogeneous in size, but for RX, the class that represents anomalies is very small. Therefore, a 1% drop on POC for k-means is very acceptable while it is not for RX.

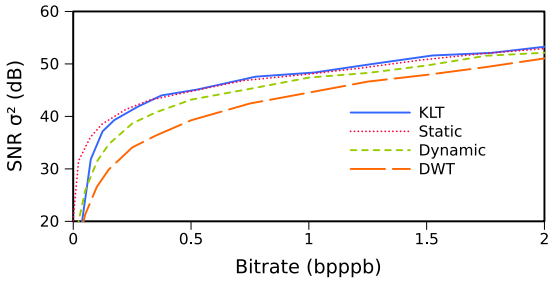
On tests for both classification techniques, the RKLT and its clustered variants perform similarly, except on one sample on the RX results, where the RKLT produces a compression artifact that the RX detector marks as anomaly. Nonetheless, this artifact is produced at a bitrate where the RX performance is already not acceptable. On the other hand, the IWT yields lower classification performance than the RKLT for the same bitrate.

In Fig 13, a visual comparison shows the different performance of a RX anomaly detector depending of the transform used. In order to produce visually appreciable results, the RX detector result is not thresholded and images are highly compressed. While all the techniques used introduced several artifacts on the RX results, these are more noticeable with the use of the dynamic structure or the IWT.

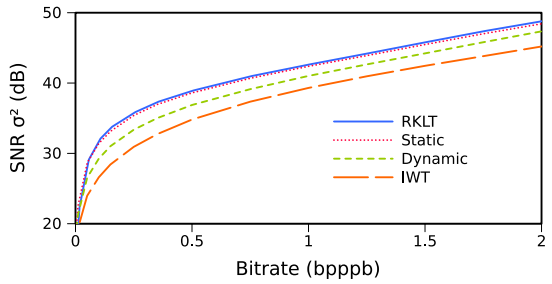
RX anomalies are usually computed after a KLT, and since the transform is a shared part in image coding and RX anomaly detection, [36] propose to improve the preservation of anomalies by simultaneously detecting anomalies while the image is coded with a KLT. In that context, if the evaluation of the proposed transforms for RX anomaly detection proved positive, a similar scheme could be set up to perform both coding and detection.



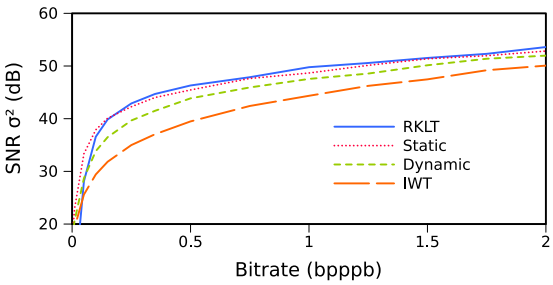
(a) Lossy coding of the AVIRIS image Low Altitude with JPEG2000



(b) Lossy coding of the AVIRIS image Moffett Field with TER (an enhanced version of CCSDS-IDC) [25]



(c) PLL coding of the Hyperion image Tornado with JPEG2000



(d) PLL coding of the AVIRIS Moffett Field with TCE

Figure 11. R-D evolution of the proposed structures for various techniques and images.

V. Conclusions

In this paper, we investigate how to reduce the computational cost of the KLT to acceptable terms through the use of multi-level clusterings. Clustered versions of the KLT provide a good solution, as they require much less computational resources than the KLT, they maintain its coding performance if applied over multiple levels, and are compatible with the use of subsampling in the covariance calculation, a technique that virtually removes transform training costs. With the proposed approaches, significant gains are achieved on the KLT training, and on the forward and inverse application of the transform. Besides, clustered transforms have also quite good scalabilities.

Our objective has been to find suitable multi-level clustering structures, among all the possible clustering combinations, that had low resource consumption, and yielded similar coding performances as the KLT. However, as the set of possible clusterings is very large, some strategy to select candidate members is necessary, and to that purpose, eigen-thresholding methods are useful. Eigen-thresholding methods, which are used to discard noise components in a principal component analysis, are here employed to help determine the number of forwarded components between levels of clusters. Among the eigen-thresholding methods examined, EIF and AE are the

Table III

DETAILED RESULTS FOR STATIC AND DYNAMIC STRUCTURES. CODING METHOD IS JPEG2000. ΔSNR IS THE AVERAGE OF THE SNR DIFFERENCE WITH THE FULL TRANSFORM. IT IS SAMPLED AT $0.1k$ BPPPB FOR $k = 1..20$, AND IS MEASURED IN DB. ΔSIZE IS THE INCREASE OF SIZE FOR LOSSLESS BITSTREAMS. COST ONLY INCLUDES TRAINING, APPLICATION AND REMOVAL OF THE SPECTRAL TRANSFORM, AND IS IN GFLOPS. A CONSTANT SUBSAMPLING FACTOR OF $\rho = 0.01$ HAS BEEN USED FOR ALL KLT TRANSFORMS.

AVIRIS Cuprite:

Reference Transforms						
	Lossy		PLL			Both
Transform	ΔSNR (min/max)	Cost	ΔSNR (min/max)	ΔSize	Cost	Scalability (min/max)
Full KLT Transform	0.00 (0.00 / 0.00)	52.9	0.00 (0.00 / 0.00)	0.00%	79.3	224.0 (224 / 224)
Half Cluster [9]	−0.02 (−0.12 / +0.06)	6.3	−0.02 (−0.11 / +0.03)	−0.56%	9.7	42.0 (42 / 42)
DWT (5 levels)	−3.22 (−5.72 / −2.23)	1.6	–	–	–	36.0 (32 / 38)
IWT (5 levels)	–	–	−3.23 (−6.17 / −1.84)	+8.01%	1.0	16.0 (11 / 17)

Multi-level clustering with dynamic thresholds						
	Lossy		PLL			Both
Transform	ΔSNR (min/max)	Cost	ΔSNR (min/max)	ΔSize	Cost	Scalability (min/max)
EIF size 28	+0.06 (−0.01 / +0.14)	9.1	+0.10 (+0.02 / +0.21)	−0.90%	13.7	64.8 (51 / 72)
EIF size 14	+0.04 (−0.04 / +0.12)	4.7	+0.11 (+0.01 / +0.24)	−0.92%	7.2	41.2 (29 / 51)
EIF size 7	−0.12 (−0.27 / −0.03)	2.5	+0.00 (−0.14 / +0.24)	−0.48%	3.8	26.1 (19 / 32)
EIF size 4	−0.34 (−0.58 / −0.24)	1.6	−0.13 (−0.41 / +0.21)	+0.10%	2.5	18.5 (13 / 25)
AE size 28	−0.05 (−0.13 / +0.02)	7.5	+0.05 (−0.06 / +0.25)	−0.56%	11.3	52.5 (46 / 55)
AE size 14	−0.08 (−0.17 / 0.00)	4.0	+0.04 (−0.08 / +0.27)	−0.57%	6.0	35.0 (27 / 40)
AE size 7	−0.29 (−0.51 / −0.18)	2.0	−0.09 (−0.34 / +0.25)	−0.03%	3.1	18.5 (17 / 19)
AE size 4	−0.34 (−0.89 / −0.22)	1.3	−0.10 (−0.62 / +0.28)	+0.02%	2.0	12.9 (12 / 13)

Multi-level clustering with static thresholds (only the most relevant)						
	Lossy		PLL			Both
Transform	ΔSNR (min/max)	Cost	ΔSNR (min/max)	ΔSize	Cost	Scalability (min/max)
$[(8, 28, 6), (4, 12, 6), (2, 12, 5), (1, 10, 0)]$	+0.06 (−0.06 / +0.16)	7.6	+0.12 (+0.02 / +0.24)	−0.93%	11.6	45.0 (45 / 45)
$[(16, 14, 3), (4, 12, 6), (1, 24, 0)]$	+0.09 (−0.08 / +0.18)	4.5	+0.17 (+0.05 / +0.31)	−1.07%	6.9	41.0 (41 / 41)
$[(32, 7, 1), (4, 8, 1), (1, 4, 0)]$	−0.31 (−0.71 / −0.17)	2.0	−0.10 (−0.49 / +0.27)	−0.03%	3.0	17.0 (17 / 17)
$[(32, 7, 2), (4, 16, 6), (2, 12, 4), (1, 8, 0)]$	+0.04 (−0.09 / +0.14)	3.1	+0.15 (+0.03 / +0.34)	−0.99%	4.8	31.0 (31 / 31)
$[(32, 7, 2), (8, 8, 4), (2, 16, 10), (1, 20, 0)]$	+0.04 (−0.11 / +0.13)	3.1	+0.15 (+0.04 / +0.30)	−0.97%	4.8	35.0 (35 / 35)
$[(32, 7, 3), (8, 12, 5), (2, 20, 7), (1, 14, 0)]$	+0.06 (−0.04 / +0.15)	3.9	+0.12 (+0.02 / +0.25)	−0.93%	6.0	38.0 (38 / 38)
$[(56, 4, 1), (7, 8, 4), (4, 7, 4), (1, 16, 0)]$	−0.14 (−0.35 / −0.07)	1.9	+0.04 (−0.15 / +0.32)	−0.48%	2.9	27.3 (26 / 29)

AVIRIS Yellowstone (Sc 0):

	Lossy		PLL			Both
Transform	ΔSNR (min/max)	Cost	ΔSNR (min/max)	ΔSize	Cost	Scalability (min/max)
Half Cluster [9]	−0.56 (−0.92 / −0.13)	6.3	+0.01 (−0.56 / +0.57)	−0.20%	9.7	42.0 (42 / 42)
DWT (5 levels)	−7.92 (-11.26 / −5.71)	1.6	–	–	–	36.0 (32 / 38)
IWT (5 levels)	–	–	−6.40 (-10.96 / −2.34)	+22.78%	1.0	16.0 (11 / 17)
AE size 4	−3.55 (−5.68 / −2.25)	1.3	−2.00 (−5.14 / +0.50)	+9.09%	2.0	12.9 (12 / 13)
$[(32, 7, 2), (4, 16, 6), (2, 12, 4), (1, 8, 0)]$	−0.97 (−1.48 / −0.27)	3.1	−0.08 (−0.98 / +0.94)	+1.37%	4.8	31.0 (31 / 31)
$[(32, 7, 3), (8, 12, 5), (2, 20, 7), (1, 14, 0)]$	−0.54 (−0.83 / −0.10)	3.9	+0.12 (−0.44 / +0.84)	+0.09%	6.0	38.0 (38 / 38)

AVIRIS Yellowstone (Sc 11):

	Lossy		PLL			Both
Transform	ΔSNR (min/max)	Cost	ΔSNR (min/max)	ΔSize	Cost	Scalability (min/max)
Half Cluster [9]	−0.24 (−0.47 / −0.10)	6.3	+0.22 (−0.22 / +0.84)	−1.14%	9.7	42.0 (42 / 42)
DWT (5 levels)	−5.47 (−8.87 / −3.68)	1.6	–	–	–	36.0 (32 / 38)
IWT (5 levels)	–	–	−3.81 (−8.31 / −0.20)	+15.39%	1.0	16.0 (11 / 17)
AE size 4	−3.43 (−6.47 / −2.25)	1.3	−1.54 (−5.20 / +0.93)	+8.54%	2.0	12.9 (12 / 13)
$[(32, 7, 2), (4, 16, 6), (2, 12, 4), (1, 8, 0)]$	−0.70 (−1.09 / −0.41)	3.1	+0.29 (−0.53 / +1.28)	−0.06%	4.8	31.0 (31 / 31)
$[(32, 7, 3), (8, 12, 5), (2, 20, 7), (1, 14, 0)]$	−0.32 (−0.52 / −0.13)	3.9	+0.39 (−0.11 / +1.23)	−1.19%	6.0	38.0 (38 / 38)

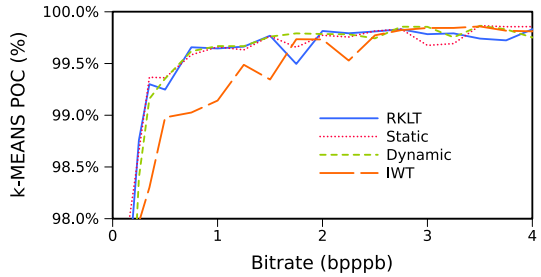
Hyperion Coral Reef:

	Lossy		PLL			Both
Transform	ΔSNR (min/max)	Cost	ΔSNR (min/max)	ΔSize	Cost	Scalability (min/max)
Half Cluster [9]	−0.04 (−0.10 / +0.04)	19.4	−0.04 (−0.11 / +0.05)	+0.04%	29.6	42.0 (42 / 42)
DWT (5 levels)	−2.26 (−2.85 / −1.86)	4.9	–	–	–	36.0 (32 / 38)
IWT (5 levels)	–	–	−2.41 (−2.88 / −2.10)	+6.02%	3.1	16.0 (11 / 17)
AE size 4	−0.91 (−1.63 / −0.72)	3.9	−0.83 (−1.54 / −0.63)	+0.71%	6.0	12.9 (12 / 13)
$[(32, 7, 2), (4, 16, 6), (2, 12, 4), (1, 8, 0)]$	−0.25 (−0.43 / −0.13)	9.5	−0.20 (−0.28 / −0.11)	−0.34%	14.5	31.0 (31 / 31)
$[(32, 7, 3), (8, 12, 5), (2, 20, 7), (1, 14, 0)]$	−0.25 (−0.42 / −0.17)	11.9	−0.21 (−0.29 / −0.16)	−0.69%	18.2	38.0 (38 / 38)

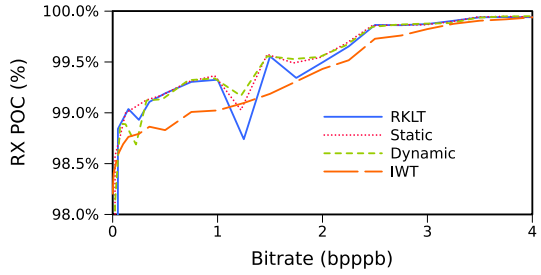
Table IV

BITRATES AT WHICH ENCODERS PRODUCE LOSSLESS BITSTREAMS. MEASURED IN BPPPB. BEST AND WORST IMAGE CODING METHODS ARE HIGHLIGHTED FOR EACH CORPUS. THREE PURE LOSSLESS METHODS ARE ALSO INCLUDED AS REFERENCE: FAST-LOSSLESS (FL) [37], LAIS-QLUT-OPT (LLUT) [38], AND TSP-W2 [31]. FOR THESE RESULTS, THE COMPLETE AVIRIS IMAGES HAVE BEEN USED.

	JPEG2000				FL	LLUT	TSP-W2
	RKLT	Static	Dynamic	IWT			
AVIRIS Cuprite	4.85	4.84	4.89	5.28	4.91	4.29	3.77
AVIRIS Jasper Ridge	4.86	4.92	5.11	5.59	4.95	4.61	4.08
AVIRIS Moffett Field	4.95	5.01	5.22	5.68	4.99	4.62	4.12
AVIRIS Lunar Lake	4.95	4.94	4.98	5.32	4.91	4.37	3.81
AVIRIS Yellowstone (Sc 0)	3.86	3.96	4.23	4.73	3.96	-	3.99
Average	4.69	4.73	4.89	**5.32**	4.74	"4.47"	**3.95**
Hyperion Agricultural	6.02	5.96	6.16	6.40	-	-	-
Hyperion Coastal	5.75	5.72	5.87	6.14	-	-	-
Hyperion Urban	6.06	6.01	6.28	6.56	-	-	-
Average	5.94	**5.90**	6.10	**6.37**	-	-	-



(a) k-means classification



(b) RX anomaly detection

Figure 12. Classification-based measures for the AVIRIS image Lunar Lake and JPEG2000.

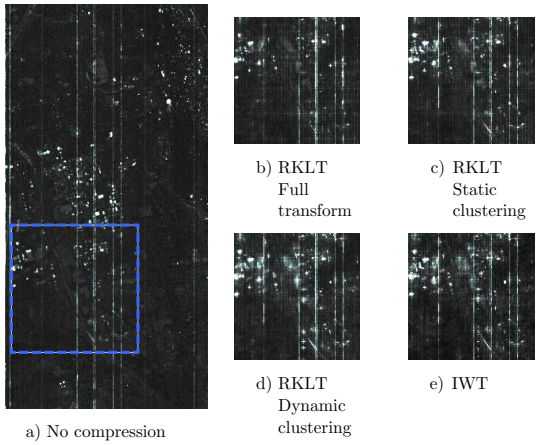


Figure 13. RX anomalies for the Hyperion image Agriculture. The image has been encoded and decoded with one of the spectral transforms and JPEG2000 at 0.3 bpppb.

most suitable.

Moreover, two distinct approaches to select structures have been investigated, differing on whether eigen-thresholding methods are used directly over every image (dynamic structures), or are used only to characterize the transform for a set of images (static structures). We found that static structures tend to have higher coding performances than the dynamic ones, but dynamic structures are the ones with greater cost savings.

Two are the most relevant multi-level clustered structures found, one static, and one dynamic. If compared with a wavelet transform, the former has a competitive cost, and almost the same coding performance as the KLT, while the latter, also as compared with wavelets, has a similar cost, and better performance, although not as good performance as the KLT. The use of one of these two transforms effectively solves our initial problem: computational costs. While the static transform allows applying a KLT with very reasonable resource constraints, the dynamic transform can be taken as a direct replacement of the DWT for spectral coding, improving the DWT in all the three measured criteria: quality, cost, and scalability. For both lossy and PLL, the proposed transforms perform between 16 and 50 times faster than the original KLT.

The best two transforms have been further validated in experimental results, which have shown similar behaviors when used in different images, with different image coders, or in lossy or lossless mode. In addition, the transforms have been used in classification-based experiments, which have shown also similar results.

REFERENCES

[1] B. Penna, T. Tillo, E. Magli, and G. Olmo, "Transform coding techniques for lossy hyperspectral data compression," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 5, pp. 1408–1421, May 2007.

[2] P. W. Hao and Q. Y. Shi, "Matrix factorizations for reversible integer mapping," *IEEE Trans. Signal Process.*, vol. 49, no. 10, pp. 2314–2324, 2001.

[3] N. Ahmed, T. Natarajan, and K. Rao, "Discrete cosine transfom," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.

[4] L. Lan and I. Reed, "Fast approximate Karhunen-Loeve transform (AKLT) with applications to digital image-coding," *Visual Commun. and Image Process. 93*, vol. 2094, pp. 444–455, 1993.

[5] A. Pirooz and I. Reed, "A new approximate Karhunen-Loeve transform for data compression," *Conference Record of the Thirty-Second Asilomar Conference on Signals, Systems and Computers*, vol. 1-2, pp. 1471–1475, 1998.

[6] M. Cagnazzo, L. Cicala, G. Poggi, and L. Verdoliva, "Low-complexity compression of multispectral images based on classified transform coding," *Signal Processing-Image Communication*, vol. 21, no. 10, pp. 850–861, 2006.

[7] Y. Wongsawat, S. Oraintara, and K. R. Rao, "Integer sub-optimal Karhunen-Loève transform for multi-channel lossless EEG compression," *European Signal Processing Conference*, 2006.

[8] Y. Wongsawat, "Lossless compression for 3-D MRI data using reversible KLT," *Int'l Conf. on Audio, Language and Image Processing, 2008. (ICALIP 2008).*, pp. 1560–1564, July 2008.

[9] I. Blanes and J. Serra-Sagristà, "Clustered reversible-KLT for progressive lossy-to-lossless 3d image coding," in *Data Compression Conf. 2009 (DCC 2009)*. IEEE Press, Mar. 2009, pp. 233–242.

[10] A. Abrardo, M. Barni, E. Magli, and F. Nencini, "Error-resilient and low-complexity onboard lossless compression of hyperspectral images by means of distributed source coding," *IEEE Trans. Geosci. Remote Sens.*, 2010, (To appear).

[11] F. A. M. L. Bruekers and A. W. M. van den Enden, "New networks for perfect inversion and perfect reconstruction," *IEEE J. Sel. Areas Commun.*, vol. 10, no. 1, pp. 130–137, 1992.

[12] L. Galli and S. Salzo, "Lossless hyperspectral compression using KLT," *IEEE Int'l Geosci. and Remote Sens. Symp. Proc. (IGARSS 2004)*, vol. 1-7, pp. 313–316, 2004.

[13] R. B. Cattell, "The scree test for the number of factors," *Multivariate Behav. Res.*, vol. 1, pp. 245–76, 1966.

[14] E. R. Malinowski, "Determination of the number of factors and the experimental error in a data matrix," *Analytical Chemistry*, vol. 49, no. 4, pp. 612–617, 1977.

[15] J. L. Horn, "A rationale and a test for the number of factors in factor analysis," *Psychometrika*, vol. 30, pp. 179–185, 1965.

[16] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. 19, no. 6, pp. 716–723, Dec 1974.

[17] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, pp. 461–464, 1978.

[18] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.

[19] J. C. Harsanyi, W. Farrand, and C.-I. Chang, "Determining the number and identity of spectral endmembers: An integrated approach using Neyman-Pearson eigenthresholding and iterative constrained RMS error minimization," *Proc. of the 9th Thematic Conf. on Geologic Remote Sens.*, pp. 395–408, Feb. 1993.

[20] G. Golub and C. van Loan, *Matrix Computations*. The Johns Hopkins University Press, Oct. 1996.
[21] Jet Propulsion Laboratory, NASA, "Airborne Visible InfraRed Imaging Spectrometer website," http://aviris.jpl.nasa.gov/html/aviris.overview.html.
[22] U.S. Geological Survey and NASA, "Earth Observing 1, Hyperion website," http://eo1.usgs.gov/hyperion.php.
[23] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards, and Practice*. Kluwer International Series in Engineering and Computer Science, 2002, vol. 642.
[24] J. Zhang, J. E. Fowler, and G. Liu, "Lossy-to-lossless compression of hyperspectral imagery using three-dimensional TCE and an integer KLT," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, pp. 814–818, Oct. 2008.
[25] F. Garcia-Vilchez and J. Serra-Sagristà, "Extending the CCSDS Recommendation for Image Data Compression for Remote Sensing Scenarios," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 10, pp. 3431–3445, 2009.
[26] Consultative Committee for Space Data Systems, "Image Data Compression," Blue Book, CCSDS 122.0-B-1, Nov. 2005.
[27] D. Taubman, "Kakadu software," http://www.kakadusoftware.com/, 2000.
[28] J. Fowler, "QccPack: An open-source software library for quantization, compression, and coding," in *Applications of Digital Image Processing XXIII*, A. Tescher, Ed., vol. 4115, San Diego, CA, USA, Aug. 2000, pp. 294–301.
[29] Group on Interactive Coding of Images, "TER software; Open Source CCSDS-122-B-1 implementation and extension." http://www.gici.uab.cat/TER, June 2008.
[30] Group on Interactive Coding of Images, "RKLT software," http://gici.uab.cat/, 2009.
[31] A. Kiely and M. Klimesh, "Exploiting calibration-induced artifacts in lossless compression of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 8, pp. 2672–2678, Aug. 2009.
[32] A. Lucero, S. Cabrera, E. J. Vidal, and A. Aguirre, "Evaluating residual coding with JPEG2000 for L-infinity driven hyperspectral image compression," *Satellite Data Compression, Communications, and Archiving*, vol. 5889, no. 1, pp. 12–23, 2005.
[33] G. Carvajal, B. Penna, and E. Magli, "Unified lossy and near-lossless hyperspectral image compression based on JPEG 2000," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 4, pp. 593–597, Oct. 2008.
[34] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statistics and Probability*, L. M. L. Cam and J. Neyman, Eds., vol. 1. University of California Press, 1967, pp. 281–297.
[35] I. Reed and X. Yu, "Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 10, pp. 1760–1770, Oct. 1990.
[36] B. Penna, T. Tillo, E. Magli, and G. Olmo, "Hyperspectral image compression employing a model of anomalous pixels," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 4, pp. 664–668, Oct. 2007.
[37] M. Klimesh, "Low-complexity adaptive lossless compression of hyperspectral imagery," *Proc. SPIE*, pp. 63 000N.1–63 000N.9, Sept. 2006.
[38] J. Mielikainen and P. Toivanen, "Lossless compression of hyperspectral images using a quantized index to lookup tables," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 3, pp. 474–478, July 2008.

Ian Blanes (S'05) Ian Blanes received the B.S. and M.S. degrees in Computer Science in 2007, and 2008 from the Universitat Autònoma de Barcelona, Spain. He was awarded by the Spanish Ministry of Education as second-best computer science student of Spain (2007 awards). In 2007, he held an intern position at Thomson Corporate Research, Princeton, New Jersey. Currently he holds a research grant by Spanish Ministry of Education and is currently working toward the Ph.D. degree. Since 2003, he has been with the Group on Interactive Coding of Images of the Universitat Autònoma de Barcelona.

Joan Serra-Sagristà (S'97-M'05) received the B.S., M.S., and Ph.D. degrees in Computer Science from the Universitat Autònoma de Barcelona, Spain, in 1992, 1994, and 1999, respectively. Since 1992 he has been with the Department of Information and Communications Engineering, at the Universitat Autònoma de Barcelona, Spain, where he is currently an Associate Professor and Director of the Group on Interactive Coding of Images. From September 1997 to December 1998, he hold a DAAD research grant at the University of Bonn, Germany. From June to July 2000 he was a visiting researcher at the U.Bonn, Germany. His current research interests include image coding, data compression, vector quantization, and wavelet-based techniques, with special attention to remote sensing and telemedicine applications. He has coauthored several papers in these areas. Dr. Serra-Sagristà is also a member of SPIE. He is a 2006 recipient of the Intensification Program Young Investigator Award. He has served on the steering committee and on technical program committees of several international conferences, and is a Reviewer for the major international journals in his research field.