[Ian Blanes, Joan Serra-Sagristà, Michael W. Marcellin, and Joan Bartrina-Rapesta]

Divide-and-Conquer Strategies for Hyperspectral Image Processing

[A review of their benefits and advantages]

In the field of geophysics, huge volumes of information often need to be processed with complex and time-consuming algorithms to better understand the nature of the data at hand. A particularly useful instrument within a geophysicist's toolbox is a set of decorrelating transforms. Such transforms play a key role in the acquisition and processing of satellite-gathered information, and notably in the processing of hyperspectral images. Satellite images have a substantial amount of redundancy that not only renders the true nature of certain events less perceivable to geophysicists but also poses an issue to satellite makers, who have to exploit this data redundancy in the design of compression algorithms due to the constraints of down-link channels. This issue is magnified for hyperspectral imaging sensors, which capture hundreds of visual representations of a given target—each representation (called a component or a band) for a small range of the light spectrum. Although seldom alone, decorrelation transforms are often used to alleviate this situation by changing the original data space into a representation where redundancy is decreased and valuable information is more apparent.

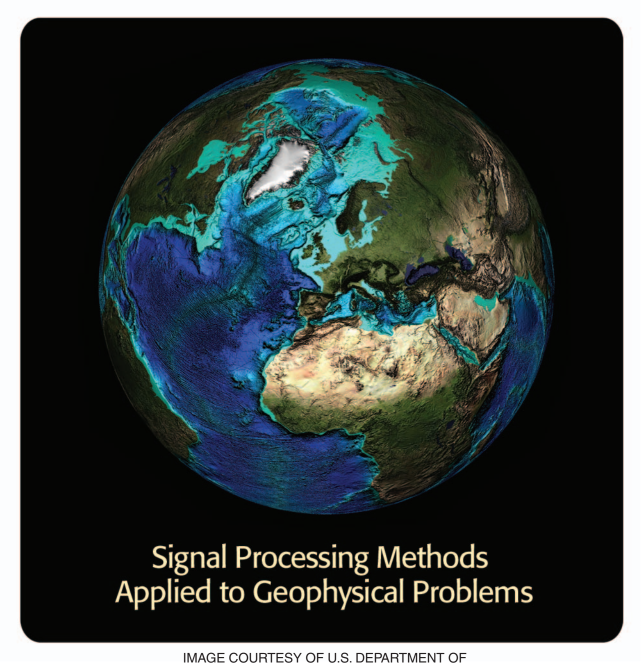


IMAGE COURTESY OF U.S. DEPARTMENT OF COMMERCE/NOAA/NESDIS/NATIONAL GEOPHYSICAL DATA CENTER

INTRODUCTION

The Karhunen-Loève transform (KLT) is a powerful decorrelating transform. Once it is applied, no correlation remains among its outputs. However, the KLT has several drawbacks. It has a very high computational cost as well as high memory requirements and a lack of component scalability, as described below. Because of these facts, it has not achieved widespread use in practice, even though it dates back more than 60 years. To alleviate these drawbacks, researchers have resorted to employing well-known approaches that help achieve a similar performance but without the burdens of the original technique. One of these

Digital Object Identifier 10.1109/MSP.2011.2179416
Date of publication: 9 April 2012

1053-5888/12/$31.00©2012IEEE

well-known approaches is a divide-and-conquer strategy, with hundreds of years of history behind it (the Euclidean algorithm to compute the greatest common divisor of two numbers dates to several centuries BC).

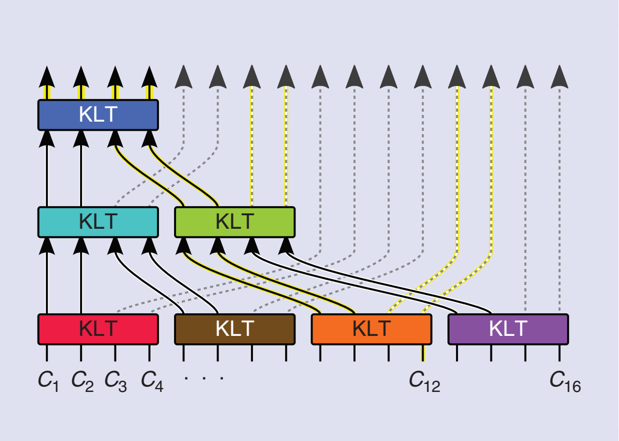
Divide-and-conquer spectral decorrelation is a recent development that allows the KLT to be approximated at a fraction of the computational cost, with lower memory requirements, while also providing some component scalability. Having efficient approximations of the KLT is important because results can be obtained earlier in time and with less hardware costs. Not only that, it allows equipping satellites, which have significant constraints in their computational resources, with better redundancy-removing methods in their image coding units, enabling them to increase the resolution of the images they acquire.

The field of hyperspectral image coding is where divide-and-conquer decorrelation strategies have flourished most vigorously, motivated in part by the large potential benefits. Different research teams have proposed several contributions applicable to this area [1]–[6]. In an historical context, divide-and-conquer spectral decorrelation is a very recent topic, with contributions starting five years ago [7], [8], and with most contributions occurring in the last two years. This article focuses on developments in the use of divide-and-conquer spectral decorrelation, mostly for hyperspectral image coding. Nonetheless, we also show other areas that may also benefit from this approach.

The KLT is a transform that adapts to the statistics of its input to provide decorrelated output vectors. It is defined by

$$y_i = \mathrm{KLT}_{\Sigma_X}(x_i) = Q^T(x_i - \bar{x}).$$

The forward application of the transform consists of the matrix/vector multiplication of Q^T and $(x_i - \bar{x})$, where Q^T is specially crafted in a training stage from the eigendecomposition of the covariance matrix Σ_X of the whole set of input data vectors $X = \{x_i\}_{\forall i}$. The term $\bar{x}$ is the input vector average, used to guarantee centered or zero-mean data. As Q^T and $\bar{x}$ are different for each input, the inverse transform requires that both are preserved as side information along with the output data set $Y = \{y_i\}_{\forall i}$.



[FIG1] Example of a divide-and-conquer strategy for a 16-input KLT. The dependencies of input 12 are highlighted in yellow.

The computational cost of the KLT is dominated by the quadratic cost of the matrix/vector multiplication that occurs on its forward and inverse applications, and partially by the covariance matrix calculation. Divide-and-conquer strategies tackle this issue by (instead of applying one large transform) dividing the KLT into a collection of smaller transforms with a lesser overall cost, and with an important point in mind: smaller transforms have to be arranged in a way that they are applied only where they are more effective. To this end, it is worth noting that transforms provide little overall benefits, if any, in portions of data with low amounts of information regardless of how correlated they are. An example of a possible organization is shown in Figure 1. In this example, a first level of KLT transforms is applied to provide local decorrelation, with the most significant half of the outputs of each transform forwarded to a next level. This process is applied recursively to account for global correlation. Note that, in this example, "less important" portions are indeed successively excluded at each level from the decorrelation process.

In the example, one large transform is replaced by seven smaller transforms each of one fourth of the original size. Since the transform cost is mainly quadratic, each smaller transform has one sixteenth of the original cost, yielding a cost for the whole approach of $7/16 \simeq 45\%$ of the original cost. Larger inputs and more sophisticated methods yield further cost reductions. The approach of the example above also improves component scalability, which is the ability to gain random access to specific components in a compressed codestream, without having to decompress the entire codestream. This ability is greatly affected by computational dependencies in the inverse transform. For example, in Figure 1, there are only eight outputs (highlighted in yellow) required to be able to perform inverse transform operations to obtain input 12, whereas for the KLT all 16 outputs would be required. More generally, having a low degree of computational dependency allows for partial applications of the forward and inverse transforms, which in turn allows decoding of portions of a compressed image without having to process or download the full compressed data. It also may allow online processing, where, as the original image is read, the compressed codestream is progressively produced, without having to allocate memory for the whole image. In practice, online compression also requires careful management of the memory needed for designing the transform (i.e., buffering of training data). This is discussed subsequently in the context of the pairwise orthogonal transform (POT).

With schemes like the one above, a full KLT can be closely approximated by a collection of smaller transforms. However, even for a given divide-and-conquer strategy, there is a combinatorial explosion in the number of possible divide-and-conquer schemes, and not all of them have equal decorrelating performance. For example, with no other constraint than to follow the successive-refining pattern as given in Figure 1, there are as many as $8.77 \cdot 10^{26}$ possible divide-and-conquer schemes for a

16-input KLT (it is estimated that the number of seconds since the Big Bang is of the order 10^{17}). To further exacerbate the situation, actual data do not always follow the Gaussian model on which the theory is based, and therefore the quality assessments that the Gaussian model provides are insufficient to guide the selection of the best possible scheme. In the face of such issues, as will be seen in the next section, researchers in this field have resorted to the use of heuristics and empirical tests to select the "best" strategy for a particular task.

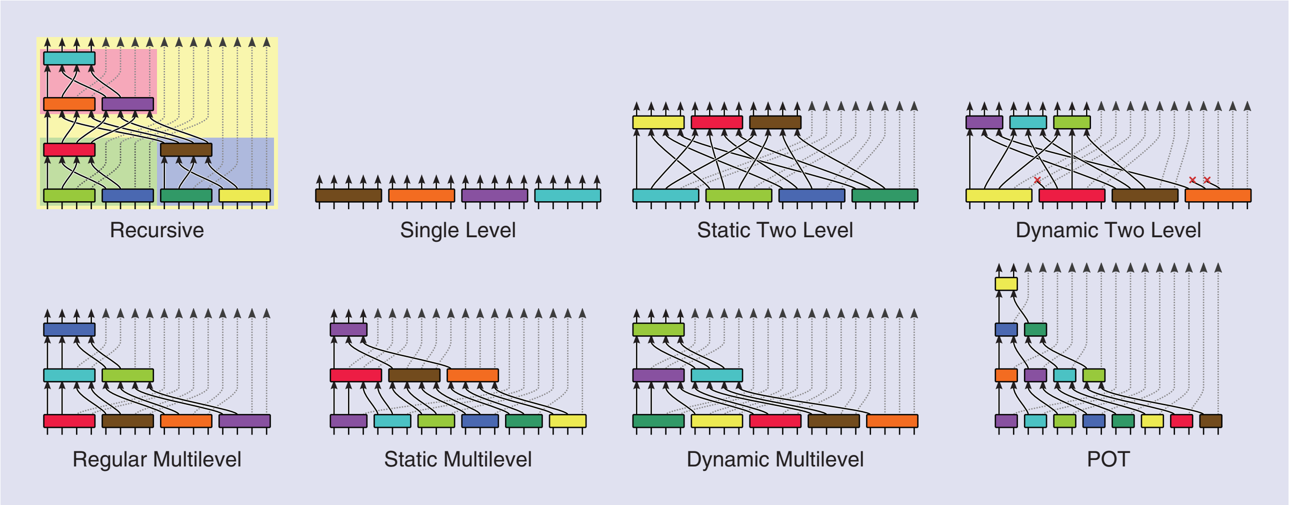
Other tools and methods related to spectral decorrelation, to the KLT, and to divide-and-conquer strategies, not central to this article, but nonetheless worth mentioning, are now reported:

- On a KLT, a direct calculation of the covariance matrix is an expensive operation. In [9], the use of statistical sampling is introduced to reduce this cost to a negligible percentage. Simple random sampling of 1% of the input is usually enough to obtain sufficiently good approximations of a covariance matrix with minimal variation of the KLT transform. Sampling is implicitly used throughout this article whenever possible.
- It is trivial to see that the KLT application can be expressed as a matrix/matrix product if all input elements are transformed at once. In that case, the use of subcubic matrix multiplication algorithms, such as the Strassen algorithm [10], yields a subquadratic per element application of the KLT. Divide-and-conquer strategies are complementary to fast matrix multiplication algorithms, as the former provides computational cost reductions by changing the applied operation by a simpler approximation and may still use the latter in its matrix operations. Results provided in this article do not incorporate these methods, as fast matrix multiplication is still an evolving field and would require a much deeper review of the subject.
- While the KLT is the optimal decorrelating transform under the assumptions of jointly Gaussian data and scalar quantization (but not only under this set of assumptions), others have tried to provide optimal transforms under other criteria. This is the case for the independent component analysis [11], [12], which tries to maximize statistical independence of non-Gaussian signals (originally designed as an extension to the KLT), and also the case for the optimal spectral transform and its variations that minimize end-to-end mean square error (MSE) under high-resolution quantization hypotheses [13], [14]. Minor coding gains can be obtained at the expense of training stages with cost increases of varying degrees.
- Finally, other related tools worth mentioning are wavelet transforms [15], [16]. Wavelets provide moderate spectral decorrelation at low computational cost and will be used in this article to provide a reference framework due to their presence in the hyperspectral image coding literature (see [17] for a detailed review).

REVIEW OF DIVIDE-AND-CONQUER STRATEGIES

The benefits of employing divide-and-conquer strategies in a plethora of disciplines have been well established [18]. In the following sections, we will illustrate the benefits of divide-and-conquer strategies for hyperspectral image processing. Here we provide a chronological review of divide-and-conquer strategies for spectral decorrelation.

Divide-and-conquer strategies on transforms for spectral decorrelation have, as explained above, a relatively short historical time line, originated by recent developments in computing hardware that have enabled a more widespread adoption of the KLT as a decorrelating transform. Once the technological obstacles were overcome, independent research teams developed a variety of strategies almost in parallel, with perhaps one strategy, the recursive subdivision, leading the way. Existing strategies can be classified in four families according to their general traits: recursive, single-level, two-level, and multilevel strategies. These families are described here in chronological order of publication and are thoroughly compared. For the reader's convenience, illustrative diagrams of each family of divide-and-conquer transforms are provided in Figure 2.



[FIG2] Illustrative diagrams of divide-and-conquer strategies.

EIGENTHRESHOLDING, OR WHERE TO "CUT"

There are methods whose purpose is to estimate the number of factors that have influenced the observed data, be it the factors involved in a chemical reaction [19] or the "minerals" present in a hyperspectral scene [20]. Oftentimes, these methods are based on determining how many components should be retained after a KLT, in which case they can be properly categorized as eigenthresholding methods (i.e., a threshold on the eigenvalues of the KLT).

One famous test is the "Scree test" from Cattell [21], which is simply based on plotting, in descendant order, the variances of the KLT outputs and selecting components up to the sharp break in the plot by visual inspection. According to Cattell himself [22], such a method would not have pleased the statistician community, yet the method was widely adopted by psychologists with quite reliable results (his article has received more than 2,900 citations since 1966).

The recursive strategy [7], [8] is the only member of the recursive family and was not originally proposed for remote-sensing image processing, although we have adapted it for hyperspectral image coding. This strategy is based on a successive subdivision of a KLT into three half-sized KLTs. Two half-sized KLTs provide a first level of local decorrelation, while the third one provides partial global decorrelation from the outputs of the other two. This three-element division is applied recursively for each half-sized KLT. The use of this recursion is mathematically convenient to prove a computational complexity below that of the KLT, on the assumption of a Toeplitz covariance matrix. Note that under that assumption, both a KLT and a Fourier transform can be used to diagonalize the covariance matrix. Apart from a good theoretical decorrelating performance, the recursive approach also exhibits experimental performance very close to that of the KLT (as opposed to a Fourier transform). The recursive strategy provides a good starting point into the subject, but, as will be discussed later, there are other strategies that provide a similar approximation penalty/performance with a lower computational cost.

The second family of divide-and-conquer strategies is the one of single-level strategies [3], [4], which are based on a single level of small transforms that provide only local decorrelation. Even if the decorrelation properties of a single-level strategy are limited, since it produces low amounts of side information, it may work well on situations where the size of the side information is a significant portion of the bit rate budget, i.e., at very low bit rates, or when the spatial dimensions are notably small.

The third family of divide-and-conquer strategies to a KLT subdivision is that of two-level strategies [1], [2]. The idea is to achieve decorrelation locally on a first level and globally on a second level, but, as opposed to the former recursive strategy, without any recursion. Instead, this family segments the first level of decorrelation in a larger number of small KLTs, and, in a second level, the important outputs of a first-level KLT are decorrelated together with the equivalent output of the other first-level KLTs. We refer to this approach as a *static* two-level strategy if used as just described, or as a *dynamic* two-level strategy if some pruning is performed after the transform is trained to sever "less contributing" inputs of second-level KLTs. Once more we refer the reader to Figure 2 for a clearer idea of the heuristics.

Finally, the last family of methods is the one of multilevel strategies [3], [5], [6], which includes four different subtypes of strategies. Multilevel strategies are based on a progressive sieving over multiple levels that yields local to global decorrelation over multiple levels. At each level, components are sliced into clusters of KLTs, and for each cluster some of its outputs are forwarded to a next level, until one last level decorrelates together all the remaining components. It is particularly notable that these strategies do not incorporate a permutation of components between each level, and nonetheless, as will be shown below, they still provide good performance.

- The regular strategy is the most naive family member: it includes strong regularity constraints to keep at bay the combinatorial explosion of feasible multilevel structures.
- As was the case for two-level strategies, we can also devise static and dynamic approaches, that help to partially lift the aforementioned constraints with the use of eigenthresholding methods, which are analytical methods used to quantify the relevant outputs of each KLT. "Eigenthresholding, or Where to 'Cut'" introduces pioneering work for these selection models. On the static variant, the possible structures are reduced from millions to a few hundred with eigenthresholding and within-level regularity, e.g., at each level of the multilevel structure, the clusters are all of the same size, and the same number of components is forwarded to the next level. The best structures are empirically selected for and from a training data set.
- On the other hand, the dynamic variant produces one structure of equal cluster size in all levels, but then a different number of important outputs for each small KLT may be selected as the transform is applied.
- The fourth member of this family is the POT, characterized by its minimal structure of two-component KLTs. The POT is a particular case of regular multilevel worth mentioning on its own due to the additional benefits of its minimal structure, specifically, the possibility of operation under strong memory constraints as well as the elimination of the numerically cumbersome eigendecomposition procedure required in the other structures. More details on the POT are provided in the section "Practical Cases."

COMPARATIVE EVALUATION

We have summarized above eight divide-and-conquer heuristic strategies, and while all of the described strategies provide approximations to the KLT, each entails a different tradeoff among distinct performance characteristics. In the current scope, such characteristics of a strategy include: coding performance, computational cost, component scalability, and

HOW TO EVALUATE THE SUCCESS OF A STRATEGY

To properly evaluate the benefits and advantages of the several approaches for a divide-and-conquer strategy for hyperspectral image processing, different criteria may be considered, mostly depending on the process at hand. Here we report those commonly used when addressing hyperspectral image coding.

Coding performance is a tradeoff between quality and bit rate, where the higher the quality for a given bit rate, the better the coding performance is. Quality is computed comparing the original image x with the recovered image $\hat{x}$: several measures can be taken, although in the case of remote-sensing images, it is customary to employ a signal-to-noise ratio (SNR) defined for instance as

$$\mathrm{SNR}_{\sigma^2} = 10 \cdot \log_{10}\left(\frac{\sigma^2}{\mathrm{MSE}}\right), \quad (\mathrm{dB}),$$

where σ^2 is the variance of the input image and MSE is

$$\mathrm{MSE} = \frac{1}{N_x N_y N_z}\sum_i \sum_j \sum_k [x(i,j,k) - \hat{x}(i,j,k)]^2.$$

The bit rate is the normalized length of the compressed file produced by the coding technique after applying the spectral decorrelation transform, and is reported herein using the unambiguous unit: bits per pixel per band (bpppb).

Computational cost is computed taking into account the number of operations that need be performed for applying a given spectral decorrelation transform. The lower the computational cost, the higher the speed of applying that particular transform. It can be measured either in number of operations or in seconds.

Component scalability is defined as the ability to retrieve a single component, as is often needed in remote-sensing applications, e.g., in false color composition for visualization purposes. The lower the number of spectral components (or bands) that are needed for inverting the spectral decorrelation transform if only a single component is to be retrieved, the higher the scalability. Component scalability aims to employ as low a number of components for inverting the spectral decorrelation as possible, both because of memory constraints and because of faster computation.

Memory requirements is a criterion that assesses the peak computer memory capacity needed to apply the spectral decorrelation transform, where lower is better, since this transform is sometimes devised for application on board aircraft or satellites, with restricted memory capability. It is often measured in megabytes.

memory requirements. We direct readers unfamiliar with such characteristics to “How to Evaluate the Success of a Strategy,” providing a brief description of the characteristics and their relevancy as well as an explanation of how to measure them. The different tradeoffs of the existing strategies are shown in Table 1. For the purpose of comparison, the KLT and three common wavelet transforms have also been included. On one side of the comparison there is the KLT, with top coding performance but lower scores on all the other characteristics, while on the other side there are the wavelets, with extremely low cost as well as low memory requirements, but with only moderate spectral decorrelation performance. In between the KLT and wavelets, there are the divide-and-conquer strategies, which provide a gradient of tradeoffs from one extreme to the other.

It is interesting to look at the different tradeoffs, since not all transforms provide reasonable compromises. In Figure 3, the characteristics of each transform are plotted in a three-dimensional space corresponding to coding performance, computational cost (speed), and component scalability. Each point in the plot is projected to the coordinate planes to show the tradeoffs that each method provides for a given pair of characteristics. From the plot, it can be seen that wavelets do not always provide a competitive deal between coding performance and computational cost in relation to divide-and-conquer strategies. In particular, as indicated by the double sided arrow in the speed/coding performance plane, the CDF 9/7 wavelet provides poor performance in relation to its speed. We also refer the reader to “A Few More Degrees of Freedom” for a deeper understanding of the different tradeoffs provided by each spectral transform.

PRACTICAL CASES

Three examples of practical usage furnish evidence of the benefits of divide-and-conquer strategies. The first example is a step-by-step application of the recursive strategy, posed such that interested readers can reproduce it easily. The second example reports the potential of the POT assuming a use case on board a satellite. Due to its extremely low complexity, it may be used to improve coding performance, despite the resource-constrained environment of satellites. The third example reports the use of divide-and-conquer strategies to improve the computational performance of a Reed-Xiaoli (RX) anomaly detector, where the distance operations are normally performed in the KLT domain. The objective of this last example is to show the applicability of divide-and-conquer strategies on geophysical signal processing fields other than hyperspectral image coding.

RECURSIVE STRATEGY EXAMPLE FOR AVIRIS HYPERSPECTRAL IMAGE CODING

In this first example, divide-and-conquer is applied with the recursive strategy using four recursion steps on a remote-sensing image that is later compressed using a lossy image coder. To ensure the reproducibility of this example, the image used is the widely distributed hyperspectral image “Cuprite” from the AVIRIS sensor from NASA’s Jet Propulsion Laboratory, and the image coder is the “Kakadu v6.4.1” implementation of JPEG2000 [24]. The image technical name is “f970619t01p02_r02,” and its size is 614 × 2,206 × 224 pixels (width × height × bands).

A FEW MORE DEGREES OF FREEDOM

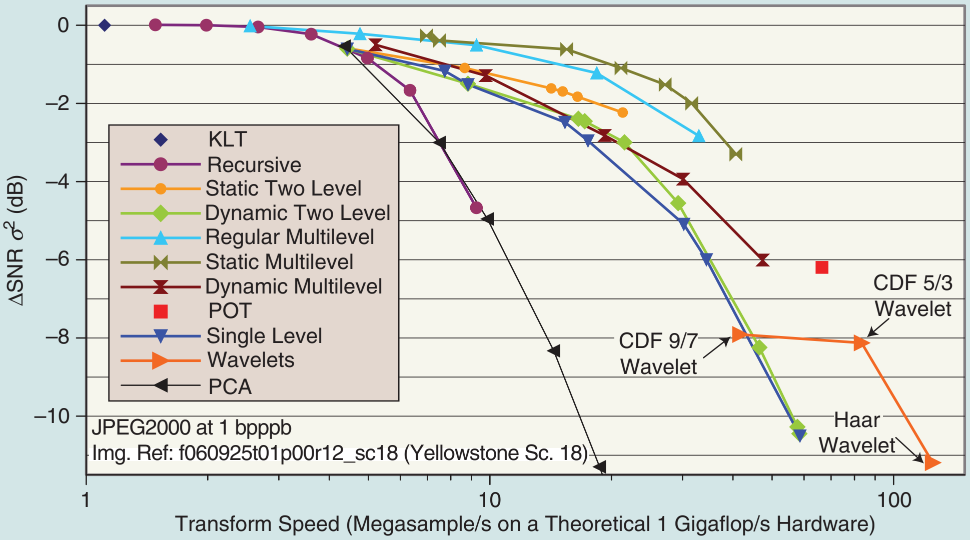
Each divide-and-conquer strategy generally has a few degrees of freedom, e.g., the recursion depth on the recursive strategy, or the number of first-level clusters, and the number of forwarded outputs on the static two-level strategy. Usually, authors of strategies sort through the possible variations within a strategy with empirical experimentation and recommend one or two items as the representative elements.

This is the case of Figure 3 and Table 1, where the performance of a single representative element for each strategy is reported. This approach is necessary for an understandable presentation of ideas, but it is also necessary to bring an element comparison to tractable size.

Nonetheless, it is an interesting exercise to compare structures in the span of their degrees of freedom, even though the comparison has to be limited in scope. Such a comparison is provided in Figure S1, where the relation between coding performance and computational cost is shown (the former measured as the coding performance difference with the original KLT, and the latter measured as the data samples transformed per unit of time). The scope of this comparison has been limited to the relative coding performance difference with respect to a full KLT on just one image when coded in combination with JPEG2000 [24] at the fixed bit rate of 1 bpppb, and only some of the best variations of each strategy are shown. While a narrowly scoped example like this is not useful to extract global conclusions, it is still insightful to the understanding of the particularities of the strategies.

In Figure S1, it can be observed that, by changing the free parameters of structures, a gradation is produced between high cost/coding performance to low cost/coding performance. In this particular case, the best tradeoffs are provided by the recursive strategy on high coding performance, followed, as coding performance decreases, by the regular multilevel strategy, the static multilevel strategy, and the POT. Three wavelets are also included as reference, and two of them provide the best tradeoffs on the lowest coding performance segment.

Results for principal component analysis (PCA) are also included. The KLT and PCA are basically the same operation, but the latter terminology is often used to indicate that the transform only retains the "principal" components (PCs) of its output. Since PCA uses a smaller number of operations than the KLT because not all outputs need to be produced, the question of whether to use PCA as an approximation of the KLT often arises. Using a PCA instead of a divide-and-conquer strategy for KLT has its merits, as, by producing just a few outputs, future processing stages are simplified (in this case JPEG2000 has less bands to code). However, there are two drawbacks. One is that the coding performance and speed relation of the PCA alone is quite bad, so it is very dependent on the speed gains of simpler future stages. The other drawback, and perhaps the most problematic, is that the number of PCs to be kept varies with different images and bit rates, and that number is hard to guess before the coding has taken place.



[FIG S1] Comparison of spectral transforms tradeoffs between coding performance and computational cost (speed) when encoding an image at a target bit rate of 1 bpppb.

The first step towards this example's purpose is to infer the structure of the transform, or equivalently define the sequence of small KLTs that have to be applied. The spectrum at each spatial location is taken as an input vector to the transform, yielding 614 × 2,206 input vectors each of dimension 224. Given the 224-input transform and the four recursions, a sequence of 81 small KLTs, each of 14 inputs, is applied as shown in Figure 4.

[TABLE 1] QUALITATIVE SUMMARY OF SPECTRAL TRANSFORMS, NOTABLY OF THE DIVIDE-AND-CONQUER STRATEGIES. THE PERFORMANCE OF EACH TRANSFORM FOR A GIVEN CRITERION IS RANKED FROM – – – (WORST) TO + + + (BEST), ACCORDING TO QUANTITATIVE DATA FROM [23].

	AUTHORS	PUBLICATION DATE	REFERENCE(S)	CODING PERFORMANCE	COMPUTATIONAL COST	COMPONENT SCALABILITY	MEMORY REQUIREMENTS
KLT							
KLT	KARHUNEN, LOÈVE	1946	[24], [25]	+ + +	– – –	– – –	– – –
DIVIDE-AND-CONQUER STRATEGIES							
RECURSIVE							
RECURSIVE	WONGSAWAT, ORAINTARA, RAO	2006	[7], [8]	+ + +	– –	– –	– – –
SINGLE-LEVEL							
SINGLE LEVEL	BLANES, SERRA-SAGRISTÀ; DU, ZHU, YANG, FOWLER	2009	[3], [4]	+ +	–	+ +	– –
TWO-LEVEL							
STATIC TWO LEVEL	SAGHRI, SCHROEDER, TESCHER	2009	[1], [2]	+ +	+	– –	– – –
DYNAMIC TWO LEVEL	SAGHRI, SCHROEDER, TESCHER	2009	[1], [2]	+ +	+ +	+ + +	– –
MULTILEVEL							
REGULAR MULTILEVEL	BLANES, SERRA-SAGRISTÀ	2009	[3]	+ + +	+	+	– –
STATIC MULTILEVEL	BLANES, SERRA-SAGRISTÀ	2010	[5]	+ + +	+	+	– –
DYNAMIC MULTILEVEL	BLANES, SERRA-SAGRISTÀ	2010	[5]	+ +	+ +	+ +	– –
PAIRWISE ORTHOGONAL TRANSFORM	BLANES, SERRA-SAGRISTÀ	2011	[6]	+	+ + +	+ + +	+ +
WAVELETS							
WAVELETS CDF 9/7	COHEN, DAUBECHIES, FEAUVEAU	1992	[16]	–	+ + +	+	+ + +
WAVELETS CDF 5/3	COHEN, DAUBECHIES, FEAUVEAU	1992	[16]	–	+ + +	+ +	+ + +
WAVELETS HAAR	HAAR	1910	[15]	– –	+ + +	+ + +	+ + +

Once the structure is defined, each small KLT is applied. Here, the small transform on bottom left of the diagram is detailed. Since the elements of the input vector to the first small KLT have mean values other than zero (i.e., 962.9, 1275.8, 1752.7, . . .), the mean is subtracted from each sample to force a zero mean value. As the outputs of a KLT have zero mean, some of the other small transforms do not need this adjustment. Then, the covariance matrix is calculated and diagonalized with a standard QR algorithm, obtaining Q^T. In this case,

$$Q^T = \begin{pmatrix} 0.037 & 0.053 & \cdots & 0.399 \\ 0.216 & 0.350 & \cdots & -0.321 \\ \vdots & \vdots & \ddots & \vdots \\ 0.003 & 0.003 & \cdots & -0.368 \end{pmatrix},$$

which is applied to its input at this point. The 80 remaining KLTs are applied similarly, in a bottom-up order, preserving the dependencies between individual KLTs.

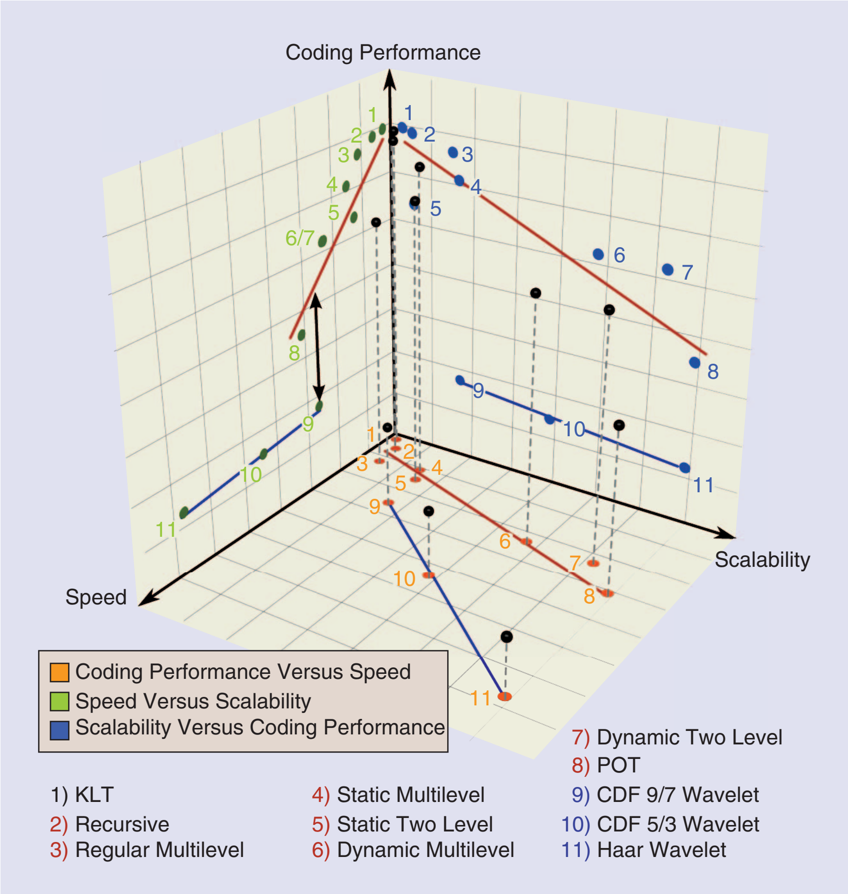
After all of the KLTs have been applied, the result can be assumed mostly free of interband redundancy, and each band can be coded independently. Using the JPEG2000 coding system provides efficient coding within a band and allows a different rate allocation for each band that maximizes the overall coding performance.

With the described approach, and targeting at a rate of one bpppb, the combination of recursive subdivision and JPEG2000 yields an SNR of 54.12 dB, which is only 0.01 dB lower than when a full KLT is used.

To decompress, one simply needs to reverse each individual operation in the inverse order in which they where applied, i.e., first the JPEG2000 codestream is decoded and then, using the side information, each of the small KLTs, starting with the last one applied in the coding process.

Counting the arithmetic operations performed in this example yields a total of 273 Gigaoperations for the KLT, or in other words, more than 4 min on a theoretical 1

Gigaflop/s central processing unit (CPU) just for the spectral transform. At virtually no quality loss, by using the recursive strategy, the total amount of operations is reduced to 31%, or about 1.2 min. By way of contrast, if a DWT CDF 9/7 was used, it would have required fewer than 9 s, but it would have had a significant coding quality penalty of 3.46 dB.



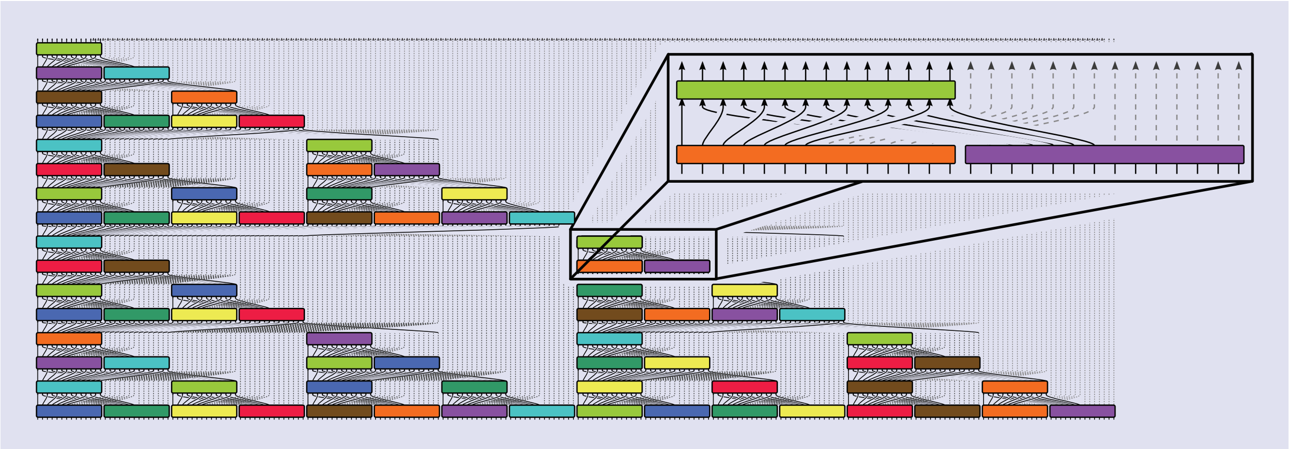
[FIG3] Comparison of spectral transforms tradeoffs among coding performance, computational cost (speed), and component scalability.

While the KLT cost might be quite a nuisance while one waits a few minutes for an image to decode, there are important use cases that forbid such a high cost. For example, with a KLT it is not possible to perform real-time coding of hyperspectral images. This is a quite important use case for remote-sensing acquisition hardware, for which hyperspectral images must be coded efficiently under the hardware constraints of such devices. The following example addresses this use case explicitly.

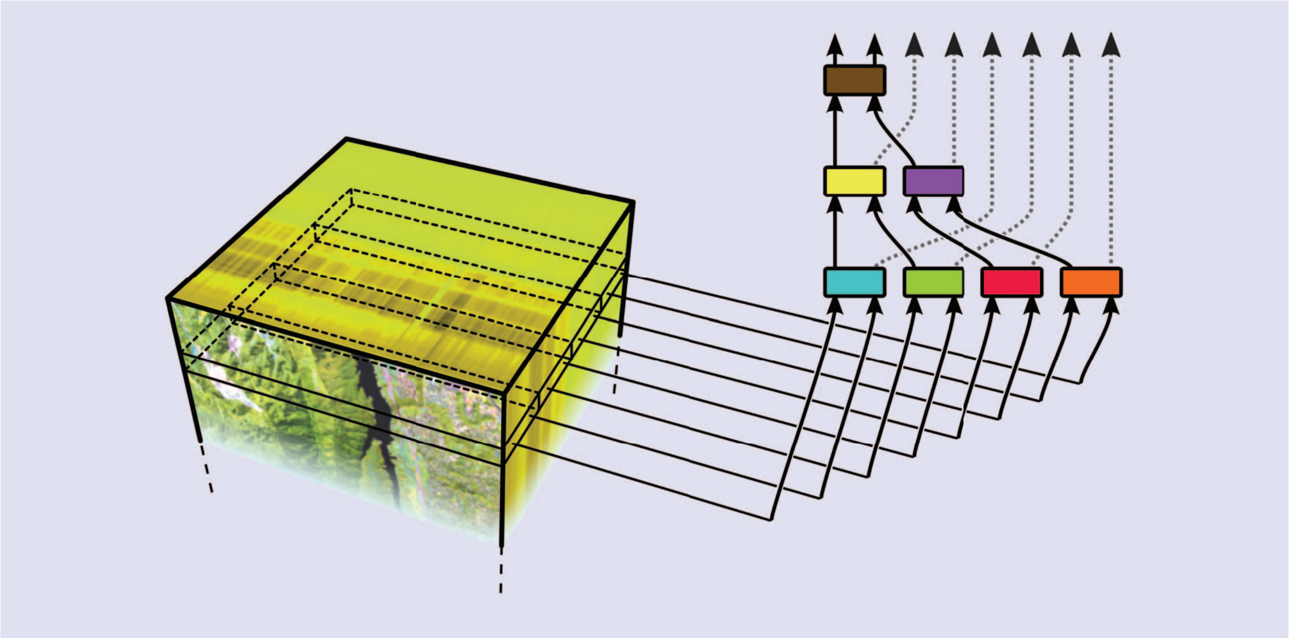
POT STRATEGY EXAMPLE FOR ON-BOARD SATELLITE APPLICATION

Due to the radiation hardening required in space-borne hardware as well as other factors such as smaller economies of scale, weight restrictions, or power-consumption limitations, satellite hardware is often heavily constrained in its capabilities in comparison with the average office desktop computer. The POT divide-and-conquer strategy is specially designed for satellite image coding and employs a minimal multilevel structure, where only two components are decorrelated at once.

In addition to the low computational cost, using a minimalistic strategy allows the reduction of the memory requirements for the transform and a much simpler eigendecomposition stage. High memory requirements for the KLT are caused by the fact that the KLT is a two-stage transform: it is first trained over the whole input, and then it is applied. Having to keep the whole image in memory while the transform is trained is the main source of memory consumption. Taking into account that satellite imaging devices usually capture



[FIG4] Structure of the recursive divide-and-conquer strategy when applied to 224 components with a recursion depth of four.



[FIG5] Line-based application of a POT divide-and-conquer strategy for on-board satellite image coding.

images line-by-line as the satellite moves along its orbit, a natural solution for this problem would be to use blocks of image lines and transform them independently; however, this solution clashes with the relatively high amount of side information that would be required by the KLT of each block. On the other hand, the POT has fewer degrees of freedom and requires a much smaller amount of side information that allows the use of spatial blocks. Such blocks may even be as small as a single image line. In Figure 5, a graphical representation of a single-line application of the POT is shown.

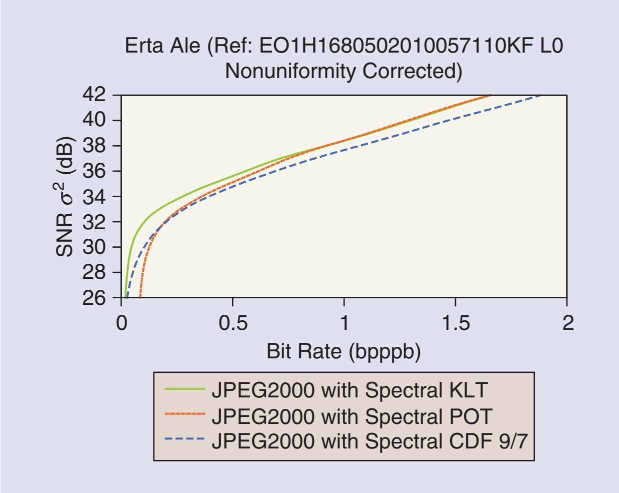
The eigendecomposition stage (i.e., where the covariance matrix is diagonalized and Q^T is obtained) is the most complicated (not the most time-consuming) stage of the whole KLT and, by extension, of a divide-and-conquer approach. This process is usually performed by a QR algorithm, or possibly by a Jacobi eigenvalue algorithm if the process needs to be parallelized. Both algorithms are well-known numerical iterative methods that converge to a solution, and nonetheless both of them present very complicated numerical instabilities that pose significant risks apart from the implementation complexity. In contrast, as the POT uses only two-component KLTs, the transform matrix $Q^T = \begin{pmatrix} p & t \\ q & u \end{pmatrix}$ can be derived in a straightforward manner from $\Sigma_X = \begin{pmatrix} a & b \\ b & d \end{pmatrix}$, to yield $t = -q = (b/|b|)\sqrt{(1/2) - ((a-d)/2s)}$, $p = u = \sqrt{1 - t^2}$, and $s = \sqrt{(a-d)^2 + 4b^2}$. Note that all the elements in Q^T can be produced from t alone, and thus the side information for the POT are the offsets to force a zero mean on each input and the value of t for each two-component transform.

Figure 6 reports a comparison of the performance of the line-based POT in relation to the KLT and the CDF 9/7 wavelet when used in combination with JPEG2000, to encode an image captured by the EO-1 satellite while orbiting over the Erta Ale volcano (Afar Region, Ethiopia). For this data type, the POT provides performance between the KLT and wavelets, with extremely low complexity and memory requirements as needed to operate on satellite equipment. At low-medium to high bit rates (usually the only ones of interest in practice), the performance of the line-based POT is above that of the DWT and especially close to that of the KLT. We attribute this closeness to the KLT due to the relatively low SNR of the Hyperion sensor itself. At very low bit rates, the performance of the POT can be undermined by the required side information. Even though the side information is only, for each row, one parameter per two-component transform and one offset per input, it amounts to 0.07 bpppb for this example, as this is a rather narrow image of only 256 columns. If this were an issue, applying the transform in blocks of two or three lines would address this problem.

Following with the theoretical 1 Gigaflop/s CPU of the previous example, even if not corresponding with most spaceborne hardware, the application of both the forward DWT CDF 9/7 and the forward POT would stay below 3 s, while the KLT would take more than 1.5 min.

HYPERSPECTRAL IMAGE PROCESSING EXAMPLE FOR ANOMALY DETECTION

A third example of the use of a divide-and-conquer strategy is in combination with the conventional RX anomaly detector [25]. Among others, airborne detection of land mines is one of the applications of an anomaly detector. The conventional RX discussed here is the baseline reference in this research field, and more powerful alternatives exist such as support vector methods [26] or Kernel RX [27]. The objective of this example is to provide some insight into how the strategies presented throughout this article can be extended to other



[FIG6] Coding performance of POT in comparison to the KLT and the CDF 9/7 wavelet. Performance measured in variance SNR (SNR σ^2) in relation to the image bit rate.

[TABLE 2] PERFORMANCE OF AN RX DETECTOR WHEN APPLIED USING DIVIDE-AND-CONQUER STRATEGIES.

METHOD	PRESERVATION OF CLASSIFICATION	ANOMALIES PRESERVED	COST
KLT (ORIGINAL)	100%	100%	100%
RECURSIVE	99.81%	90.54%	32%
STATIC MULTILEVEL	99.68%	84.21%	8%
DYNAMIC TWO LEVEL	99.09%	54.38%	4%

fields of interest of the geophysics community in addition to image coding; not to improve the state of the art on said fields, which would be the objective of another article.

An RX anomaly detector is based on how distant a pixel r in a hyperspectral image is from the overall background of that image as measured using the Mahalanobis distance,

$$\mathrm{RX}(r) = (r-\mu)^T \Sigma_X^{-1}(r-\mu),$$

where μ is the average background. If the covariance matrix is a diagonal matrix, then the problem is simplified. By performing the substitution $\Sigma_X^{-1} = Q\Lambda^{-1}Q^T$, the problem becomes

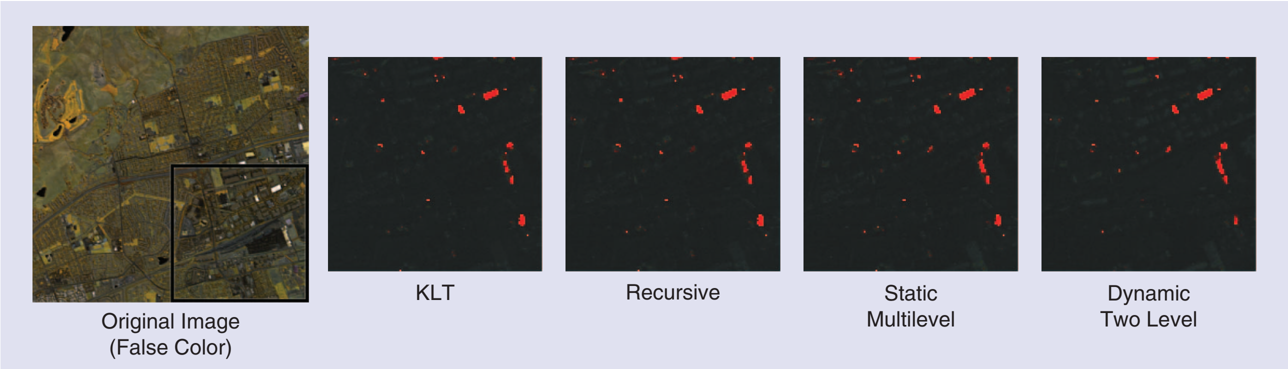
$$\mathrm{RX}(r) = (Q^T(r-\mu))^T \Lambda^{-1}(Q^T(r-\mu)),$$

where $Q^T(r-\mu)$ is the KLT of $(r-\mu)$. As the matrix multiplication by Q^T can be applied approximately by a divide-and-conquer strategy, the computational cost of the anomaly detector is diminished.

In this example, the RX detector is used on a hyperspectral image acquired near the Moffett Federal Airfield in California. An RX detector is an unsupervised classifier that ranks how anomalous each location is, and then locations ranked over a threshold (in this case, the top 2%) are classified as anomalies. Table 2 shows results when using several divide-and-conquer strategies. Detector performance is measured with the KLT as reference, either in the column "Preservation of Classification," which is the percentage of locations that do not change class (anomaly/no anomaly) in comparison with the KLT, or in the column "Anomalies Preserved," which is the percentage of the anomalies that are detected in both classifications. Detector performance decreases along with the method cost, with up to 54% of the anomalies preserved with only 4% of the original cost. It is worth noting that the anomalies that are not preserved are often situated on the edges of anomaly zones, with no significant variations on locations or shapes of the detected zones. Figure 7 shows a visual representation of the detector output.

CONCLUSIONS

Each year, remote-sensing technologies gather an increasing amount of hyperspectral information, which is treated with more sophisticated data processing methods demanding large amounts of computing resources. Spectral decorrelation is a widely used method with a significant computational cost, particularly in the image coding context that is described in this article. We have shown that divide-and-conquer strategies mitigate these issues with schemes that provide approximate decorrelation at a fraction of the original cost as well as with improved component scalabilities and lower memory requirements. We have reported, in three practical cases of divide-and-conquer decorrelation strategies for hyperspectral images, the benefits and advantages of these strategies, and given insights into the applicability of these technologies to adjacent fields, which we hope may foster its use in other research fields.



[FIG7] Visual comparison of an RX detector.

Java and MATLAB implementations of the described methods, together with simple example applications, are available at http://gici.uab.es.

ACKNOWLEDGMENTS

We would like to thank Carole Thiebaut for her valuable comments and suggestions. This work has been partially supported by the European Union, the Spanish Government (MICINN), FEDER, and the Catalan Government, under grants FP7-SPACE FP7-242390, FP7-PEOPLE-2009-IIF FP7-250420, TIN2009-14426-C02-01, FPU AP2007-01555, and 2009-SGR-1224. The computational resources used in this article were partially provided by the Oliba Project of the Universitat Autònoma de Barcelona.

AUTHORS

Ian Blanes (ian.blanes@uab.es) received the B.S., M.S., and Ph.D. degrees in computer science from the Universitat Autònoma de Barcelona, in 2007, 2008, and 2010, respectively. Since 2003, he has been with the Group on Interactive Coding of Images of the Universitat Autònoma de Barcelona, where he currently holds a postdoctoral position. In 2007, he was an intern with Thomson Corporate Research, Princeton, New Jersey. In 2010, he was a visiting postdoctoral researcher at the Centre National d'Etudes Spatiales, France. He was the recipient of the 2007 Awards by the Spanish Ministry of Education as second-best computer-science student of Spain.

Joan Serra-Sagristà (joan.serra@uab.es) received the Ph.D. degree in computer science from Universitat Autònoma Barcelona (UAB), Spain, in 1999. He is currently an associate professor in the Department of Information and Communications Engineering, UAB. From 1997 to 1998, he was with the University of Bonn, Germany. His current research interests focus on data compression, with special attention to image coding for remote-sensing and telemedicine applications. He is an associate editor of *IEEE Transactions on Image Processing* and has coauthored over 100 publications. He was the recipient of the Spanish Intensification Young Investigator Award in 2006.

Michael W. Marcellin (marcellin@ece.arizona.edu) received the B.S. degree in electrical engineering from San Diego State University in 1983, and the M.S. and Ph.D. degrees in electrical engineering from Texas A&M University in 1985 and 1987, respectively. Since 1988, he has been with the University of Arizona, where he currently holds the title of Regents' Professor and is the International Foundation for Telemetering Chaired Professor. His research interests include digital communication and data storage systems, data compression, and signal processing. He has authored or coauthored more than 200 publications in these areas. He has received numerous honors, including six teaching awards.

Joan Bartrina-Rapesta (joan.bartrina@uab.es) received the B.Sc., B.E., M.S., and Ph.D. degrees in computer science from the Universitat Autònoma de Barcelona, Spain, in 2002, 2004, 2006, and 2009, respectively. He was awarded a doctoral fellowship from Universitat Autònoma de Barcelona. He collaborated in the development of BOI, a JPEG2000 Part 1 implementation. His research interests include a wide range of image coding topics, including highly scalable image and video coding systems, region-of-interest coding, rate-distortion optimization techniques, distortion estimation, interactive image and video transmission, and medical image coding.

REFERENCES

[1] J. A. Saghri, S. Schroeder, and A. G. Tescher, "An adaptive two-stage KLT scheme for spectral decorrelation in hyperspectral bandwidth compression," *Proc. SPIE*, vol. 7443, p. 744313, Aug. 2009.

[2] J. A. Saghri, S. Schroeder, and A. G. Tescher, "Adaptive two-stage Karhunen-Loeve-transform scheme for spectral decorrelation in hyperspectral bandwidth compression," *SPIE Opt. Eng.*, vol. 49, p. 057001, May 2010.

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

[4] Q. Du, W. Zhu, H. Yang, and J. E. Fowler, "Segmented principal component analysis for parallel compression of hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 4, pp. 713–717, Oct. 2009.

[5] I. Blanes and J. Serra-Sagristà, "Cost and scalability improvements to the Karhunen-Loève transform for remote sensing image coding," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 7, pp. 2854–2863, July 2010.

[6] I. Blanes and J. Serra-Sagristà, "Pairwise orthogonal transform for spectral image coding," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 3, pp. 961–972, Mar. 2011.

[7] Y. Wongsawat, S. Oraintara, and K. R. Rao, "Integer sub-optimal Karhunen-Loève transform for multi-channel lossless EEG compression," in *Proc. European Signal Processing Conf.*, 2006. [Online]. Available: http://www.eurasip.org/Proceedings/Eusipco/Eusipco2006/papers/1568981760.pdf

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

[9] 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.

[10] V. Strassen, "Gaussian elimination is not optimal," *Numer. Math.*, vol. 13, no. 4, pp. 354–356, Aug. 1969.

[11] C. Jutten and J. Herault, "Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture," *Signal Process.*, vol. 24, no. 1, pp. 1–10, 1991.

[12] P. Comon, "Independent component analysis, a new concept?," *Signal Process.* vol. 36, no. 3, pp. 287–314, Apr. 1994.

[13] I. P. Akam Bita, M. Barret, and D.-T. Pham, "On optimal transforms in lossy compression of multicomponent images with JPEG2000," *Signal Process.*, vol. 90, no. 3, pp. 759–773, 2010.

[14] I. P. Akam Bita, M. Barret, and D.-T. Pham, "On optimal orthogonal transforms at high bit-rates using only second order statistics in multicomponent image coding with JPEG2000," *Signal Process.*, vol. 90, no. 3, pp. 753–758, 2010.

[15] A. Haar, "Zur theorie der orthogonalen funktionensysteme," *Math. Annal.*, vol. 69, no. 3, pp. 331–371, 1910.

[16] A. Cohen, I. Daubechies, and J. C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Comm. Pure Appl. Math.*, vol. 45, no. 5, pp. 485–560, 1992.

[17] J. E. Fowler and J. T. Rucker, "3-D wavelet-based compression of hyperspectral imager," in *Hyperspectral Data Exploitation: Theory and Applications*. Hoboken, NJ: Wiley, 2007, pp. 379–407.

[18] S. Dasgupta, C. H. Papadimitriou, and U. V. Vaziran, *Algorithms*. New York: McGraw-Hill, 2006.

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

[20] 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," in *Proc. 9th Thematic Conf. Geologic Remote Sensing*, Feb. 1993, pp. 395–408.

[21] R. B. Cattell, "The Scree test for the number of factors," *Multivariate Behav. Res.*, vol. 1, no. 2, pp. 245–276, 1966.

[22] R. B. Cattell, "Citation classic—The Scree test for the number of factors," *Curr. Contents*, no. 5, pp. 16–16, 1983. [Online]. Available: http://garfield.library.upenn.edu/classics1983/A1983PY50000001.pdf

[23] I. Blanes, J. Serra-Sagristà, and P. Schelkens, "Divide-and-conquer decorrelation for hyperspectral data compression," in *Recent Advances in Satellite Data Compression,* B. Huang, Ed. Berlin: Springer-Verlag, 2011, pp. 215–232.

[24] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards, and Practice* (Kluwer International Series in Engineering and Computer Science, vol. 642), Norwell: MA, 2002.

[25] 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.

[26] A. Banerjee, P. Burlina, and R. Meth, "Fast hyperspectral anomaly detection via SVDD," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Oct. 2007, vol. 4, pp. IV–101–IV–104.

[27] H. Kwon and N. Nasrabadi, "Kernel rx-algorithm: A nonlinear anomaly detector for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 2, pp. 388–397, Feb. 2005.