Compression of Multispectral Images by Three-Dimensional SPIHT Algorithm

Pier Luigi Dragotti, Giovanni Poggi, and Arturo R. P. Ragozini

***Abstract*—We carry out low bit-rate compression of multispectral images by means of the Said and Pearlman's SPIHT algorithm, suitably modified to take into account the interband dependencies. Two techniques are proposed: in the first, a three-dimensional (3-D) transform is taken (wavelet in the spatial domain, Karhunen–Loeve in the spectral domain) and a simple 3-D SPIHT is used; in the second, after taking a spatial wavelet transform, spectral vectors of pixels are vector quantized and a gain-driven SPIHT is used. Numerous experiments on two sample multispectral images show very good performance for both algorithms.**

***Index Terms*—Compression, multispectral images, remote sensing, transform coding.**

I. Introduction

REMOTE sensing images are of interest for a large number of applications, such as geology, earth-resource management, pollution monitoring, meteorology, and military surveillance. As a consequence, there is a constant growth both in the number and in the performance of remote sensing facilities, which produce larger and larger amounts of data that have to be transmitted, processed, and stored efficiently. The problem stems from the raw size of the images considered, and is especially urgent when dealing with multispectral[1] images where the amount of data to be managed further increases with the number of bands. As an example, a single multispectral image acquired by the Thematic Mapper (TM) sensor carried on board of the Landsat V satellite, amounts to more than 200 MB. New sensors are bound to have higher radiometric precision, as well as better spatial and spectral resolution. Images composed by tens or even hundreds of spectral bands are highly valuable because of the wealth of information they provide on the nature of the ground, but their management can easily overwhelm the available resources. These problems can be significantly reduced by using some form of data compactation or data compression [1].

Data *compactation*, or lossless coding, is a reversible processing in which the original data can always be recovered from the encoded data without any loss of information. Everyday examples are the "compress" routine of Unix or the "zip" routines available on DOS/Windows. Reversibility is no doubt a desirable property since the original data are always more rich of information than any processed (possibly "enhanced") version. On the other hand, since lossless coding exploits only the statistical redundancy of the image, the compression ratio achieved rarely exceeds 2 : 1, not enough to cope with serious data management issues.

A much better performance is guaranteed by data *compression*, or lossy coding, which easily achieves compression ratios of 10 : 1 and more. As the name suggests, however, in this case, one has to tolerate some loss of information (errors) in exchange for the increased efficiency: the higher the compression ratio the lower the image quality as measured by means of some suitable distortion function.

A number of techniques have been proposed in the last few years for the compression of multispectral images. All of them try to take advantage, in various degrees, of the peculiarities of these images, especially of the high intraband and interband redundancies they exhibit. They can be roughly classified in two families: techniques based on vector quantization (VQ), and techniques based on transform coding.

Vector quantization [2], [3] is theoretically the optimal block coding strategy. Indeed, it is the direct application of the principles of information theory, and all other block coding techniques (e.g., transform coding) can be seen as structurally constrained forms of VQ. However, unconstrained VQ is characterized by a computational complexity that grows exponentially with the block size. As a consequence, practical coding schemes based on VQ are forced to use small blocks, thereby exploiting the statistical dependencies among only a small number of pixels and/or spectral bands. In [4], for example, VQ operates on purely spectral blocks, thus neglecting any spatial dependency; on the contrary, in [5] it is used with purely spatial blocks, and the spectral dependencies are exploited through nonlinear block prediction; a similar hybrid scheme based on VQ and address prediction is proposed in [6]. In all of these cases, a small block size is considered (no more than 16 pixels) with detrimental effects on the performance. In [7], instead, large three-dimensional (3-D) blocks are represented as the Kronecker product of smaller vectors, which are then jointly vector quantized: although the encoder is not optimal anymore, the use of larger blocks leads to a better performance.

To obtain a good encoding performance with limited complexity, many researchers rely on transform coding techniques [1], where a linear transform decorrelates the input data and concentrates most of the power in a few coefficients so that subsequent quantization is more efficient. For example, in [8] the Karhunen–Loeve transform (KLT) is used to decorrelate the data in the spectral domain, followed by a two-dimensional (2-D) discrete cosine transform (DCT) in the spatial domain;

Manuscript received May 11, 1998; revised February 8, 1999.

The authors are with the Dipartimento di Ingegneria Elettronica e delle Telecomunicazioni, Università "Federico II", Napoli 80125, Italy.

Publisher Item Identifier S 0196-2892(00)00407-1.

[1]In the following, unless otherwise stated, we use "multispectral" also in place of "hyperspectral," irrespective of the number of component bands.

in [9], instead, a hybrid predictive/transform coding scheme is proposed. In [10], to take into account the nonstationarity of the image, pixels are first classified (by means of tree-structured VQ) so as to use subsequent transform coding (KLT+DCT) later on groups of homogeneous pixels.

In the transform coding framework, wavelet transform [11], [12] deserves a special treatment because of its peculiar characteristics. Indeed, due to its implementation as a recursive filtering procedure, it can easily work on large blocks (even on the whole image) thereby providing an excellent power concentration. This is not possible with the KLT which is much more complex and needs prior statistical information. Moreover, unlike other simple transforms, notably the DCT, the wavelet transform does not spread the power associated with discontinuities in the whole transform domain thus providing a simple and elegant way to deal with nonstationarity. These properties (among the others) motivate an intense research on wavelet transform (WT) in the image coding field [13], also with reference to multispectral images (e.g., [14]). Only recently, however, with the embedded zerotree wavelet (EZW) encoding technique proposed by Shapiro [15], and subsequently refined by Said and Pearlman [16], WT has been incorporated into a very successful encoding technique.

In this paper, whose first results appeared in [17], we extend the Said–Pearlman algorithm or SPIHT (set partitioning in hierarchical trees) to the multispectral case and propose two new encoding algorithms. Indeed, the SPIHT possesses a number of desirable properties (good performance, low complexity, embedded encoding) which make it a perfect candidate for the task of compressing multispectral images, possibly on board. Although the extension to 3-D images is conceptually straightforward, it presents new problems (mainly concerning implementation issues) and new opportunities (offered by the strong interband dependencies) that have to be addressed specifically.

We propose and compare here two alternative techniques for the compression of multispectral images. Both use the wavelet transform in the spatial domain but they differ in how they take into account spectral dependencies: by means of KLT in the first case, by means of tree-structured VQ in the second case. After transform and (in the latter case) VQ, both use the zerotree coding approach by applying suitably modified versions of the SPIHT on the resulting images of coefficients.

After a brief review of the SPIHT algorithm in Section II, the proposed techniques are described in detail in Section III (except for the VQ codebook design which is deferred to the Appendix). Section IV is devoted to the assessment of the performance, also in comparison with those of some reference techniques. Finally, Section V draws conclusions.

II. The SPIHT Algorithm

In this section, we review just the basic concepts and terminology concerning the SPIHT that are relevant for the remainder of the paper; for a thorough description of the algorithm the reader is referred to the original papers of Shapiro [15] and Said and Pearlman [16].

All transform coding techniques consist of three main steps (see Fig. 1): 1) transform, 2) quantization, and 3) inverse

Fig. 1. Block scheme of transform coding.

transform. The transform is usually unitary (norm-preserving) so that quantization errors in the transform domain correspond to equivalent errors on the reconstructed signal. It has the main goal of concentrating the input power, spread over a large number of samples, into as small a number of transform coefficients as possible. In fact, for a given total power, it is much more convenient to quantize a single large coefficient than many small ones.

Unfortunately, an "ideal" transform does not exist. For wide-sense stationary sources, the Karhunen–Loeve transform is well-known to be optimal in a statistical sense [3] since it concentrates as much power as possible in the first coefficients. However, it is computationally expensive, requires a transmission overhead, and does not work well on nonstationary images. The data-independent DCT overcomes many of these problems and works extremely well on flat low-pass regions of the image. However, it does a poor job in the presence of discontinuities (edges, boundaries between regions, impulses) spreading the block power on a large number of coefficients.

Wavelet transform (WT) represents, in a way, the answer to the limited ability of the DCT (and other transforms) to represent efficiently both the low-pass part and the details of an image. A small low-pass version of the image is extracted which accounts for most of the power, while the details can be recovered by a series of directional high-pass bands (see Fig. 2). What is especially important, an impulse in the original image contributes significantly to just a small number of coefficients in the WT, corresponding to the same spatial location at various resolutions, and therefore it can be encoded very efficiently. Although no claim of optimality can be made for the WT, it is no doubt a powerful tool for signal processing in general and image coding in particular [12], [13].

After the transform, be it WT, KLT, or DCT, there is the problem of conveniently assigning the encoding resources; as seen before, one should quantize larger coefficients first, but the location of such coefficients is not known *a priori*. The SPIHT algorithm tackles the bit assignment problem at its root. The basic idea is to sort all coefficients in order of decreasing magnitude so that an almost "perfect" bit assignment can be achieved. More precisely, coefficients are grouped according to their significance with respect to a set of octavely decreasing thresholds $2^N, 2^{N-1}, \cdots, 2^0$: a coefficient whose magnitude exceeds or equals 2^n is said to be significant for that threshold. The encoder sends a sequence of significance maps which locate significant coefficients at level 2^N, 2^{N-1}, etc., and uses quantization bits only for coefficients found significant thus far. For example, in the sequence of coefficients (–5, 0, –2, 7, 2, 1, –3, –1), the first map comprises the linear coordinates $\{1, 4\}$ to indicate that the first and the fourth coefficients are significant with respect to the threshold 4; for each of these coefficients, the sign and a quantization bit are immediately sent. Likewise, the second map is $\{3, 5, 7\}$, and allows the encoder to send the sign and a quanti-

(a)

(b)

Fig. 2. Example of wavelet transform: (a) "House" original and (b) wavelet transform, the low-pass subband is in the upper-left corner.

zation bit for the corresponding coefficients, followed by a further bit for those of the first map. The process continues with the following maps as desired. Thanks to the sorting information, quantization bits can be used almost optimally by specifying bits in order of significance. The quantization is still suboptimal for several reasons (the sorting is only partial, the thresholds are not optimized, the quantization is scalar and greedy) but very satisfactory.

Of course, the sorting information must be sent as well for the decoder to work properly, and this can be very expensive: sending an explicit list of coordinates will not work. So, the core of the SPIHT is an efficient strategy for encoding the sorting information, which takes full advantage of the properties of wavelet transform. The transform coefficients are organized in

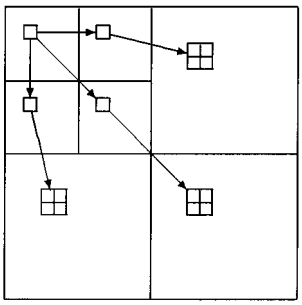


Fig. 3. Tree structure used in the SPIHT algorithm.

trees (see Fig. 3) with a baseband coefficient as root; each tree corresponds roughly to a square region of the original image, the root itself can be considered as the low-pass version of that region while the descendants correspond to finer and finer details. Three "spatial orientation trees" depart from the root, in the horizontal, vertical, and diagonal subbands. Now, since the image is for the most part smooth, significant coefficients are usually concentrated in the upper levels of the trees and only rarely in the bottom layers. This fact is exploited by recursively encoding the significance of subtrees rather than that of single coefficients. If all coefficients in a subtree are insignificant for a given threshold a single bit, say "0," will signal this occurrence; otherwise a bit "1" is sent and all subtrees are analyzed in turn. If the input image is indeed smooth, it will often happen that large "zerotrees" will be pruned at once allowing for a very efficient localization of the significant coefficients. On the other hand, even in the presence of impulses or edges, only a small number of high-pass coefficients will be significant and the recursive pruning with its "divide and conquer" approach will single them out efficiently.

As said before, a precise description of the SPIHT is beyond the scope of this paper and this qualitative synthesis overlooks many details, which can be easily found in [16]. In the same paper, an experimental performance analysis is carried out on the well-known test image Lena (and on a different test image) showing extremely good rate-distortion results. Besides being efficient, the SPIHT is relatively simple and fast, does not require prior information and provides embedded coding, namely, the output data produced at a given rate can be used to decode a meaningful (but lower quality) image at any desired lower rate. For all these reasons, it seems sensible to extend the SPIHT, whenever possible, to other image coding applications, like video coding or, as in the present case, multispectral image coding.

III. Proposed Techniques

In this section, we propose two techniques that extend the SPIHT to the case of multispectral images, the first, based on 3-D transform coding, and, the latter, on gain-shape VQ.

A. The 3-D Transform SPIHT

The obvious way to extend the SPIHT to a 3-D source is to take a 3-D wavelet transform of the image and, from then on, to apply the usual encoding procedure, with 3-D instead of 2-D

trees. The interband (spectral) dependence will be automatically exploited the same way the spatial dependence is.

This solution is certainly viable, and maybe the best possible, when the source has a significant size in the third dimension[2] as in the case of video sequences or 3-D medical images [18], but it does not work with images comprising only a few bands like the Landsat TM images used in the experiments of Section IV: a WT in the spectral dimension would not make much sense in this case. On the other hand, even though the WT possesses a number of desirable properties, it does not satisfy any optimality criterion for image compression in general or even for SPIHT-like image compression algorithms in particular. Shapiro himself suggests [15] that other subband decompositions and other hierarchical structures could lead to interesting results. The recent work of Xiong *et al.* [19], for example, shows that equally good results can be achieved using an EZW-like image coder with the DCT in place of the WT. As a matter of fact, one ends up using a fixed transform, like the WT, only because of the complexity of a data-driven transform, but when complexity is not an issue it is more reasonable to use the KLT which guarantees optimal (at least in a statistical sense) power concentration.

For these reasons, in the 3-D implementation of the SPIHT for multispectral images, we resort to the usual WT in the spatial domain but to a KLT in the spectral domain (this is also the solution adopted by Said and Pearlman in their color image coder.) Of course, some additional computation is needed to evaluate the transformation matrix as well as a small overhead to encode it, but for vectors of small size they are both negligible. As a matter of fact, given the limited increase in computation and encoding costs, it is also possible to consider more than one KLT so as to better match the statistics of the image [10]. Therefore, we consider two alternative transform sequences:

- spectral KLT on the whole image, followed by spatial WT of the transform bands;
- spatial WT of the original bands, followed by subband-adaptive spectral KLT.

In the latter case, one takes advantage of the implicit classification carried out by the wavelet transform, which separates the coefficients according to their resolution (fine/coarse) and orientation (horizontal/vertical/diagonal). It is reasonable to expect that different subbands have different statistics, and that the use of subband-matched KLT matrices will improve performance. However, only numerical experiments will make clear whether such an improvement is actually worth the increased overhead in terms of computation and side information.

After the transform, we have to define a 3-D hierarchical structure to run the SPIHT algorithm. Based on a few preliminary experiments we selected the structure shown schematically in Fig. 4. Each coefficient in the baseband of the first KLT band is the root of a tree and has four children: three of them located in the horizontal, vertical and diagonal lowest-resolution subbands, the fourth in the baseband of the second KLT band. The same-band children generate the usual spatial orientation trees, while the spectral child is root of another 3-D tree. Note that only the baseband coefficients have a spectral child, otherwise multiple-parent coefficients would arise and the quantization algorithm would not work properly.

[2]However, note that this approach poses serious memory management problems. For a 3-D image of size, say, 512 × 512 × 512 with 16-bit coefficients, the transformed image alone requires 256 MB of memory and the lists maintained by the SPIHT (list of significant and insignificant pixels, list of insignificant sets) can grow much larger than that at high rates.

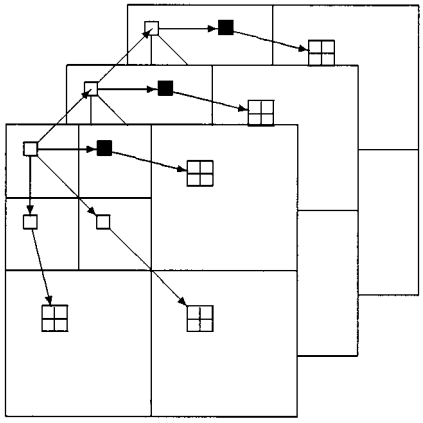


Fig. 4. Tree structure used in the 3-D-SPIHT algorithm. The "black" coefficients form a spectral vector like those used in the gs-SPIHT.

B. Gain-Shape SPIHT

The use of vector quantization to encode image subbands is quite common [13] and has its rationale in the well-known principle of information theory which states that it is always possible to improve the encoding performance by quantizing vectors (blocks of samples) instead of scalars. While this is certainly true, one should be aware that such improvements come at the price of an exponential increase of the complexity (in terms of both memory and computation) with the vector size, which readily exceeds any reasonable bound. Although a large number of techniques have been devised to carry out VQ with limited complexity [3], the problem remains, so much so that transform coding still keeps the lead in most applications.

When dealing with small vectors, however, where complexity problems can be better controlled, VQ represents an extremely appealing tool: its structural freedom allows one to exploit all intrablock dependencies, be they linear or nonlinear, whereas only linear dependencies are exploited in transform coding.

Therefore, we devised a second SPIHT-based technique for multispectral images where vector quantization is used to encode spectral vectors of coefficients. More precisely, after the WT of all bands of the image is carried out, the homologous coefficients from all the bands are stacked to form vectors, one for each point, and these are then vector quantized to a given precision. In Fig. 4, one such vector has been evidenced by using black rather than white squares for its components in the various bands. The SPIHT can now work on these vectors like it did on scalars, with just a few changes: the magnitude of the coefficient is replaced by the norm of the vector, while the sign information disappears altogether, to be replaced by the direction of the vector. To implement the VQ-based SPIHT in this

form, highly resemblant of the original algorithm, it is necessary to use a gain-shape VQ codebook in which all codevectors are obtained as products of the form

$$\hat{\mathbf{x}}_i j = \hat{g}_i \hat{\mathbf{s}}_j \tag{1}$$

where $\hat{s_j}$ is a unit-norm shape vector (the "direction" of the codevector) and $\hat{g}_i$ is a scalar gain factor. This choice not only allows us to introduce only minimal changes to the basic encoding structure of the SPIHT, but also reduces considerably both encoding complexity and memory storage with respect to unstructured VQ.

As a matter of fact, even working at moderate bit-rates, as we do in this work, a prohibitively large VQ codebook could turn out to be necessary to accurately encode the most important vectors, like those in the baseband.[3] By using a product codebook, one can hugely reduce such storage requirements: if the gain and shape codebooks have size N_G and N_S, respectively, a total of $N = N_G N_S$ product codevectors are available, but the memory occupation is only proportional to $N_G + N_S$. The encoding performance is slightly inferior to that of an unconstrained codebook of the same size N but the complexity saving is well worth this price.

Progressive transmission is easily achieved by resorting to tree-structured codebooks where a better and better approximation of the input vector is obtained with each new bit of information, just like in ordinary scalar quantization. In particular, to preserve the embedding, both the gain and shape codebooks need be tree-structured. While the joint design of ordinary gain and shape codebooks is a deeply understood problem and algorithms exist [3] that provide jointly (locally) optimal codebooks, the same is not true for tree-structured gain and shape codebooks. In the Appendix we propose a simple algorithm for the design of the tree-structured gain and shape codebooks which also solves the related resource assignment problem (how many bits should be devoted to the gain, how many to the shape, and in which order).

Once the design problems are solved, the encoding algorithm is conceptually very similar to the original SPIHT: once a vector is found significant, because its gain (the norm) exceeds a given threshold, one or more bits of shape (the direction) are immediately sent which, together with the average gain for that class of significance, forms the initial reproduction vector. Then, significant vectors are progressively refined, like in the ordinary SPIHT, by sending gain and shape bits in the order singled out at design time until all resources are consumed.

Also in the case of the gs-SPIHT, like for the 3-D-SPIHT described before, it is possible to devise a subband-adaptive version where a different gain-shape codebook is designed and used for each WT subband. Again, numerical experiments of the next section will clarify the usefulness of such a variation.

IV. Numerical Results

We present here the results of a number of experiments carried out to assess the performance of the proposed algorithms, also in comparison with that of some reference techniques proposed in the literature [7], [8], [10].

Fig. 5. Band 5 of the TM test image.

We use a TM multispectral image of a region near Lisbon in Portugal. There are six bands in the range 0.4–2.5 μm, each consisting of 2401 lines of 2401 pixel quantized at 8 bit per pixel (bpp), plus a thermal band of different spatial resolution which is not considered in the experiments for the sake of simplicity. To test the algorithms under conditions of higher spectral correlation we also use a 63-band hyperspectral image acquired by the GER (geophysical environmental research) airborne sensor which portrays an agricultural area in Germany near the river Rhein. Each band consists of 1953 lines of 512 pixels quantized at 16 bpp; in particular, we use 16 bands (from 6 to 21) that have a constant spectral resolution of 25.4 nm, and only 9 bpp of meaningful information. In both cases, a square region of 512 $\times$ 512 pixels which exhibits all the land covers present in the image is selected and used as a test set, while the rest of the image or part of it is used as a training set to design the VQ codebook. The wavelet transform in the SPIHT has five levels of decomposition and is based on the symmlet-16 orthogonal wavelet [20]. Fig. 5 shows band 5 of the TM test image, while Fig. 6 shows band 12 of the GER test image.

Performance is assessed through the analysis of suitable rate-distortion curves. For now, we use the mean square error distortion measure, as is customary in the data compression field, but later on, we will consider other distortion criteria, more closely related to possible applications of compressed images, such as maximum error and probability of misclassification.

Rates are given on a per-band basis, i.e., 8 bpp means 8 bit per pixel for each band. The mean square error is defined as

$$\mathrm{MSE} = E[(x - \hat{x})^2] \tag{2}$$

[3]However, note that VQ is intrinsically lossy, hence perfect reconstruction is not possible and the quality saturates at medium/high rates when sending more bits will not improve the reconstruction any further.

Fig. 6. Band 12 of the GER test image.

or, equivalently, by the SNR

$$\mathrm{SNR} = 10 \log_{10} \frac{E[x^2]}{\mathrm{MSE}} \tag{3}$$

where $E(\cdot)$ denotes statistical average. In practice, lacking an accurate probabilistic model, statistical averages are replaced by sample averages. All bands are normalized to zero mean and unitary power prior to encoding in order to obtain homogeneous subjective results; otherwise, a minimum-MSE strategy would privilege high-power bands and underencode low-power ones. To facilitate the analysis of the results we will often use average (on B bands) rate and distortion, namely

$$\begin{aligned} R &= \frac{1}{B} \sum_{b=1}^{B} R_b, \\ \mathrm{MSE} &= \frac{1}{B} \sum_{b=1}^{B} \mathrm{MSE}_b, \\ \mathrm{SNR} &= 10 \log_{10} \frac{1}{\mathrm{MSE}}. \end{aligned} \tag{4}$$

Under the normalized-power condition, such averages indeed make sense.

A. Selecting the Encoding Algorithm

In this subsection we analyze the results of several experiments in order to select the encoding scheme that better satisfies the contrasting requirements of good performance, and low design and encoding complexity.

Let us begin with the Landsat TM image. Fig. 7 shows the encoding results of the 3-D-SPIHT; the average SNR already exceeds 12 dB at 0.2 bpp (a compression ratio of 40), and improves rapidly with the rate, up to more than 16 dB at 0.5 bpp. Such values of the SNR (by definition, the SNR is 0 at 0 bpp) correspond to medium and high quality images, as the reader can verify by taking a look already at Figs. 13 and 14 discussed later.

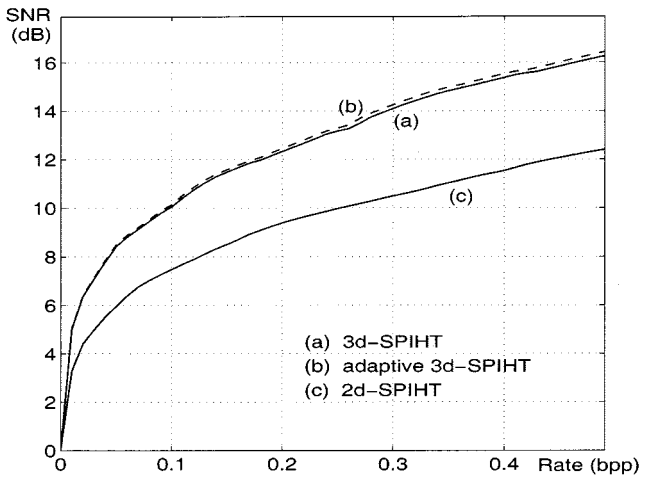


Fig. 7. Rate-distortion performance of the 3-D-SPIHT for the TM test image: (a) 3-D-SPIHT, (b) subband-adaptive 3-D-SPIHT, and (c) 2-D-SPIHT (band by band encoding).

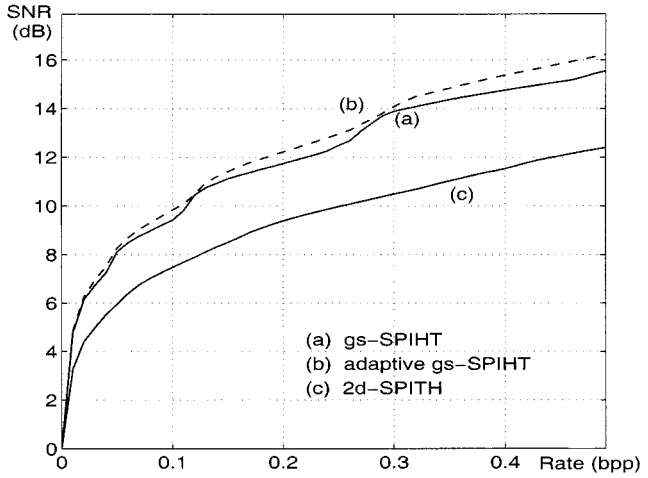


Fig. 8. Rate-distortion performance of the gs-SPIHT for the TM test image: (a) gs-SPIHT, (b) subband-adaptive gs-SPIHT, and (c) 2-D-SPIHT (band by band encoding).

Compared to the 2-D-SPIHT (band by band SPIHT), whose results are also shown in the figure, the 3-D version guarantees a 2–4 dB improvement at all rates. The 3-D-SPIHT is implemented both with the usual KLT-WT sequence (solid line), and with the WT-KLT sequence (dashed line), in which case a different KLT is used for each subband. It is clear that the negligible performance gain granted by this latter approach is not worth the extra complexity, so it will not be considered anymore in the experiments.

Fig. 8 shows results for the gs-SPIHT, and compares the use of a single codebook for all subbands (solid line) with that of subband-matched codebooks (dashed line); as a reference, 2-D-SPIHT performance is also reported again. With respect to the 3-D-SPIHT algorithm, only small differences in the performance are observed, hence the choice between the two should be guided mainly by other considerations, such as implementation complexity, and prior information on the source. For the gs-SPIHT, however, the subband adaptivity does provide some limited improvement over the single-codebook

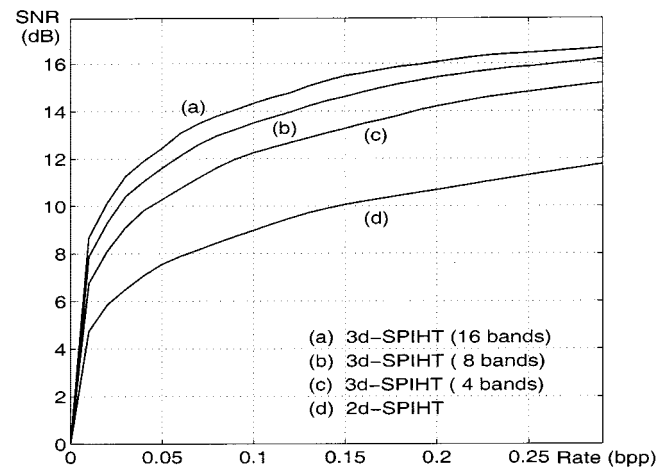


Fig. 9. Rate-distortion performance of the 3-D-SPIHT for the GER test image; the 16 bands are encoded in groups of $B = 16$, 8, 4 or 1 (2-D-SPIHT) bands.

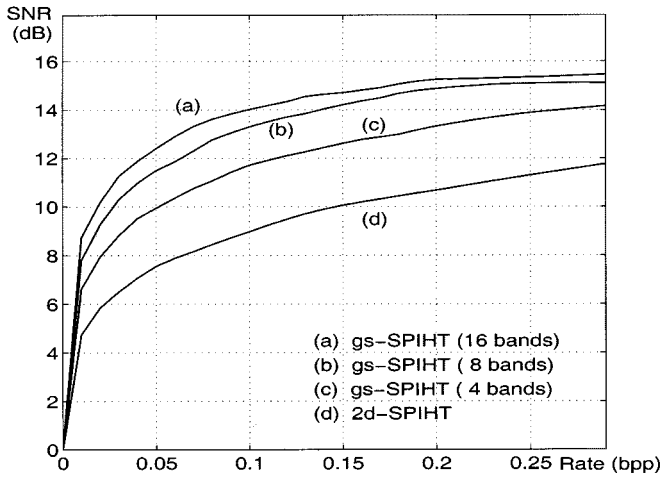


Fig. 10. Rate-distortion performance of the gs-SPIHT for the GER test image; the 16 bands are encoded in groups of $B = 16$, 8, 4 or 1 (2-D-SPIHT) bands.

case, so it could be considered as a convenient choice when complexity is not an issue.

Results for the GER image are presented in Fig. 9 (3-D-SPIHT, single KLT) and Fig. 10 (gs-SPIHT, single codebook). The general behavior is similar to that observed in the TM image case, but the same values of the SNR are obtained at much smaller rates, as a consequence of the stronger interband dependency which characterizes this image.

Dealing with a 16-band image, we now have the opportunity to study the effects of jointly encoding many bands; in particular, we use groups of $B = 1$ (isolated encoding), 4, 8, and 16 bands. Of course, the performance improves with B in general, but the gain is smaller and smaller as B increases. So, going from $B = 4$ to $B = 8$ guarantees a 1–1.5 dB improvement, while going from $B = 8$ to $B = 16$ adds just another 0.5 dB on the average. Moreover, higher B's call for more encoding resources, which are not always available. In the 3-D-SPIHT, which maintains large lists of coordinates, memory is the critical resource, and $B > 16$ could not even be implemented (with 128 MB of memory). The gs-SPIHT is less memory and computation intensive because it has to manage a single spectral vector in

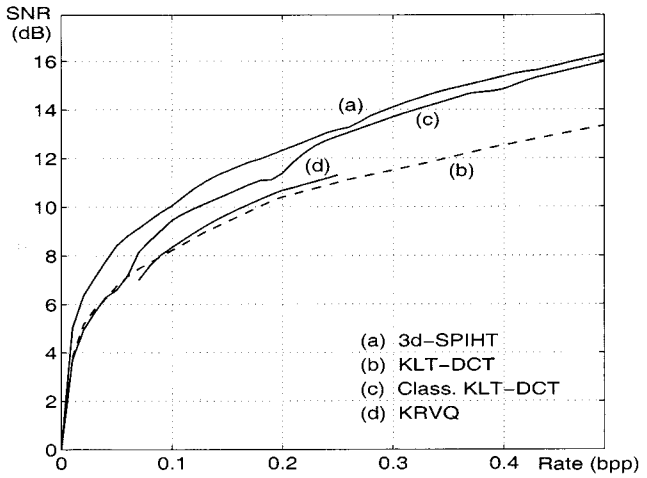


Fig. 11. Rate-distortion performance of various techniques for the TM test image: (a) 3-D-SPIHT (proposed), (b) macroblock-adaptive KLT-DCT [8], (c) class-adaptive KLT-DCT [10], and (d) Kronecker-representation VQ [7].

place of B individual scalars and hence maintains much smaller lists of coordinates. However it exhibits the saturation phenomenon already described in Section III, and clearly visible above 0.2 bpp in the curves $B = 8$ and $B = 16$.

Summing up all this experimental evidence, 3-D-SPIHT and gs-SPIHT appear to provide approximately equivalent performance. The former requires much less design effort and memory storage (the KLT matrix can even be computed on line) so it will be our choice for subsequent experiments. Nonetheless, when central memory is the more precious resource, or when nonlinear dependencies are dominant, the gs-SPIHT should be preferred. In the following experiments we will use $B = 6$ for the TM image, and $B = 16$ for the GER image.

B. Comparison with Conventional Methods

The performance of the selected technique (3-D-SPIHT with single KLT) is now compared to that of three "conventional" methods, based on transform coding and vector quantization, proposed specifically for the compression of multispectral images. The technique proposed by Saghri *et al.* [8] (referred to as KLT-DCT, here) uses spectral KLT followed by spatial DCT; to account for the nonstationarity of the image, a different KLT matrix is computed and used for each image macroblock. Gelli and Poggi [10] use VQ to classify each pixel of the image, and then encode the residuals by means of class-adaptive spectral KLT, and spatial DCT. Finally, Canta and Poggi [7] carry out VQ by means of a Kronecker-product gain-shape codebook (KRVQ) which enables the use of large 3-D blocks with reasonable complexity.

Fig. 11 presents encoding results for the TM image: 3-D-SPIHT outperforms all reference techniques at all rates gaining more than 2 dB on the average over both KLT-DCT and KRVQ. Only the classification-based technique (Class. KLT-DCT) has a comparable performance but it requires a considerable training to design the classifier and the class-adaptive KLT. A similar behavior is observed for the GER image (Fig. 12) with the only difference that here the performance gain is significant also with respect to the Class (KLT-DCT) technique. In summary, the

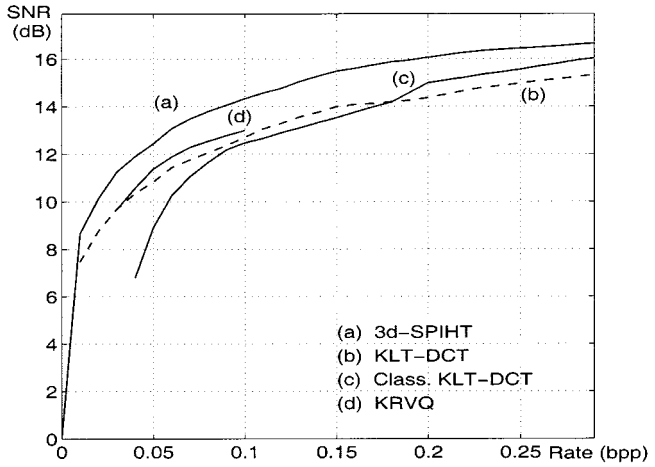


Fig. 12. Rate-distortion performance of various techniques for the GER test image: (a) 3-D-SPIHT (proposed), (b) macroblock-adaptive KLT-DCT [8], (c) class-adaptive KLT-DCT [10], and (d) Kronecker-representation VQ [7].

3-D-SPIHT seems to be preferable to conventional methods (at least the set we could test) because of its limited complexity, its fully progressive transmission ability, the absence of any training phase, and also, as the experiments show, for its better rate-distortion performance.

C. Assessing the Image Quality

As said before, the mean square error (and related SNR) is a convenient performance measure because of its universality, tractability, and fairly good agreement with other measures, but cannot be the only performance criterion when images must be used for further (possibly unknown) processing and their semantic value should be carefully preserved. Therefore, we are going to consider here some alternative criteria.

Subjective as it can be, it is important to first gain some insight about the visual appearance of images encoded at various bit rates. In each of Figs. 13–16 we show a section of the original, encoded, and difference image (the errors are multiplied by five and shifted around 128 for display) of a single band of a test image.

Fig. 13 shows band 5 of the TM test image encoded at 0.2 bpp: many fine details are lost and the borders between different regions are often blurred; the difference image is highly structured (confirming that not all relevant information has been encoded) showing peaks of error in the border regions. Hence, this image is likely not to be accurate enough for the most demanding applications. On the contrary, at 0.5 bpp (Fig. 14) the difference image is almost flat and also fine details and borders are faithfully encoded, despite the high compression ratio, 16 : 1.

The GER image presents a different behavior: both at 0.2 bpp (Fig. 15) and at 0.3 bpp (Fig. 16) the encoding quality is very high, and also the difference images hardly show any structure, looking more like white noise. With respect to the original, the most relevant differences are in homogeneous regions, which appear to be more flat after compression. Our guess is that already at 0.2 bpp all (or most of) the structural information has been correctly reproduced, and all further encoding resources go to refine the noise present in the original image. In other words,

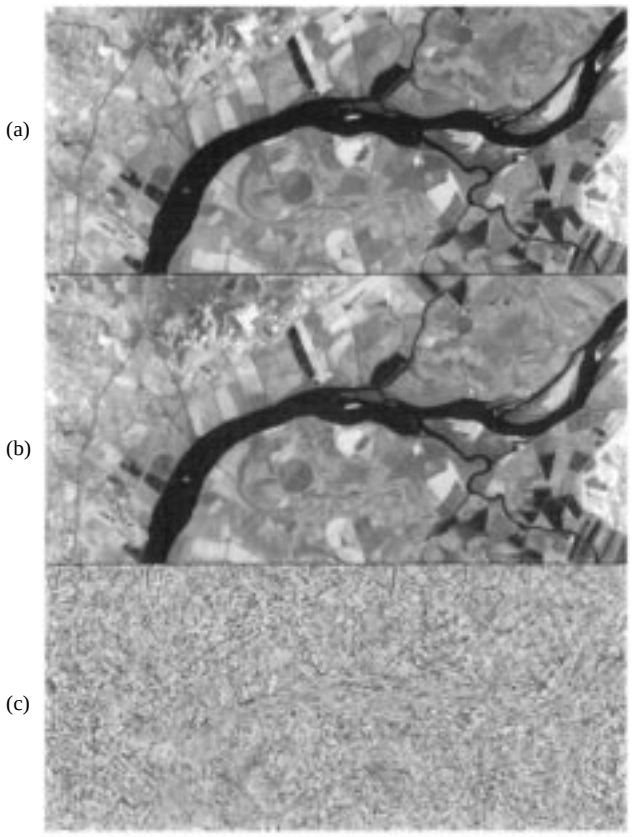


Fig. 13. Detail of band 5 of the TM test image: (a) original, (b) compressed at 0.2 bpp, and (c) Fig. 5 difference image (five times amplified).

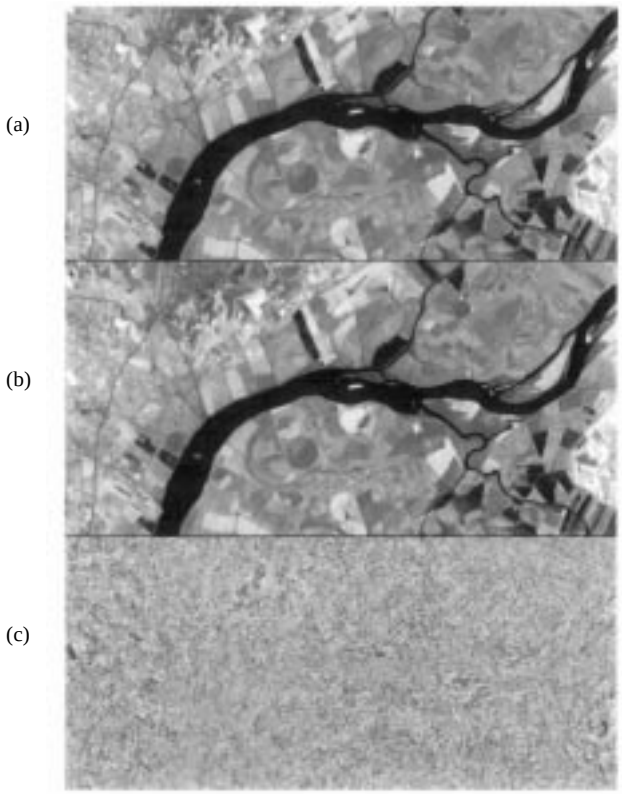


Fig. 14. Detail of band 5 of the TM test image: (a) original, (b) compressed at 0.5 bpp, and (c) difference image (five times amplified).

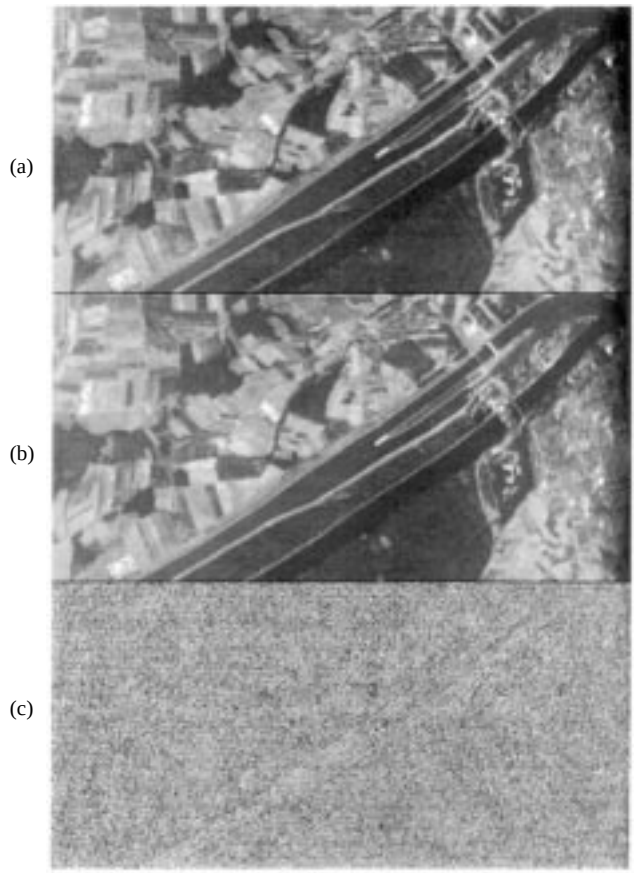


Fig. 15. Detail of band 12 of the GER test image: (a) original, (b) compressed at 0.2 bpp, and (c) difference image (five times amplified).

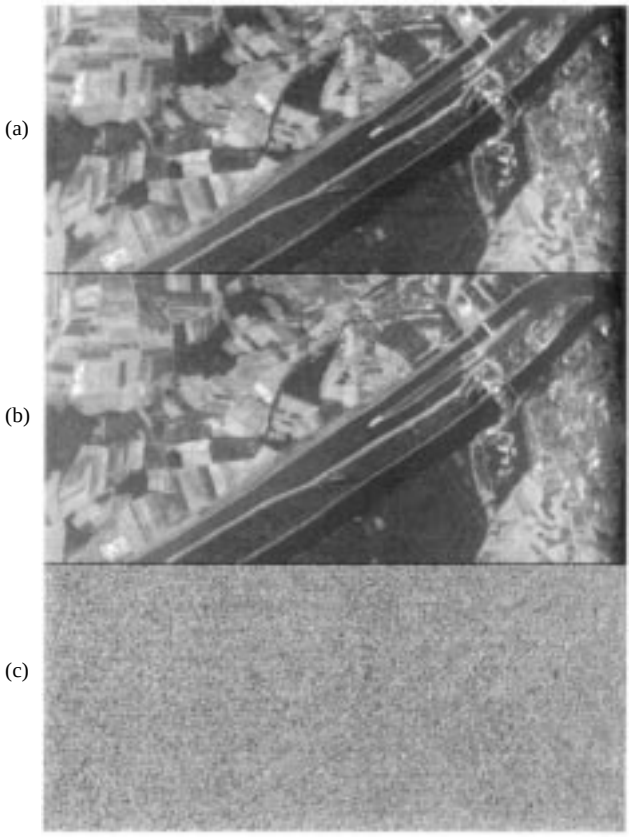


Fig. 16. Detail of band 12 of the GER test image: (a) original, (b) compressed at 0.3 bpp, and (c) difference image (5 times amplified).

low rate compression has the effect of removing sensor noise, and the compressed image *looks* even better than the original. To be more conservative, it is at least reasonable to conclude that lossless encoding is often an unnecessary (but expensive) requirement.

The l^{∞} norm, namely, the maximum encoding error over the whole image, is also a performance measure often considered in the literature. In fact, it is assumed that a small maximum error is a guarantee that the diagnostic value of the images is fully preserved; in such a case it is customary to speak of *near-lossless* coding. The 3-D-SPIHT proposed here becomes lossless for sufficiently high rates, but we want to investigate its behavior at lower bit-rates. Fig. 17 reports the global maximum error on the TM image as a function of the rate. Although generally decreasing with the rate, it is still pretty large (44 out of 256 levels) even at 0.5 bpp. However, this is mainly due to a small number of outliers, as is clear by the analysis of Fig. 18 which shows the histogram of the errors at 0.3 bpp: the probability of errors larger than 20 is clearly negligible, even though the maximum error is 64. Therefore, we experimented with a simple error correction scheme, in which the coordinates (band and spatial location) of each of the L pixels with the largest errors are sent, together with their value in the original domain quantized at 5-bit precision. By choosing L one controls the amount of side information required. In this way, much better results are obtained, as shown in Fig. 17. Taking again the case of 0.5 bpp as a reference, the maximum error decreases to 25 and 20, respectively, when 0.001

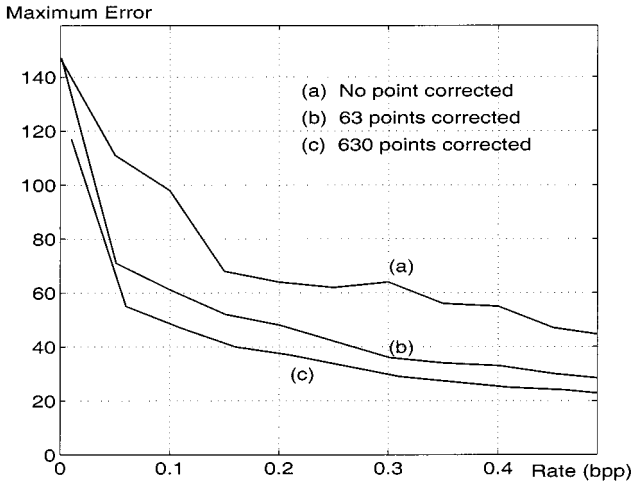


Fig. 17. Maximum error as a function of the rate for the TM test image: (a) no correction of the outliers, (b) correction of 63 values (0.001 bpp of side information), and (c) correction of 630 values (0.01 bpp of side information).

bpp and 0.01 bpp of side information are sent. The same considerations hold for the GER image, Fig. 19, with the only difference that the errors are generally smaller here when compared to the maximum value of 511, and the results at 0.3 bpp are similar to those obtained at 0.5 bpp for the TM.

Finally, to analyze the usefulness of compressed images for subsequent applications, we carried out some classification experiments. A simple minimum distance classifier was designed on the original image for a variable number of classes: $N =$

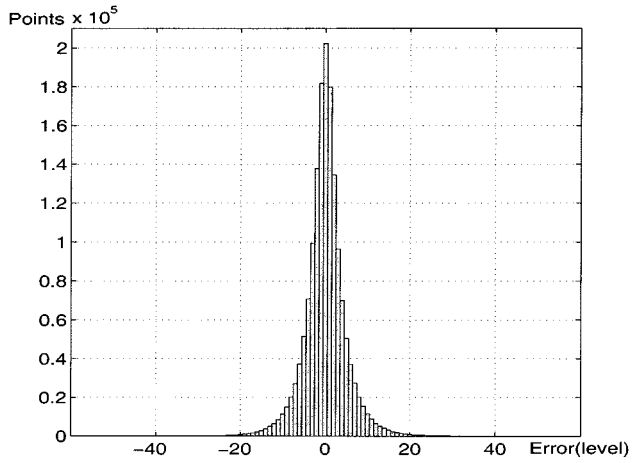


Fig. 18. Histogram of the encoding errors at 0.3 bpp for the TM test image.

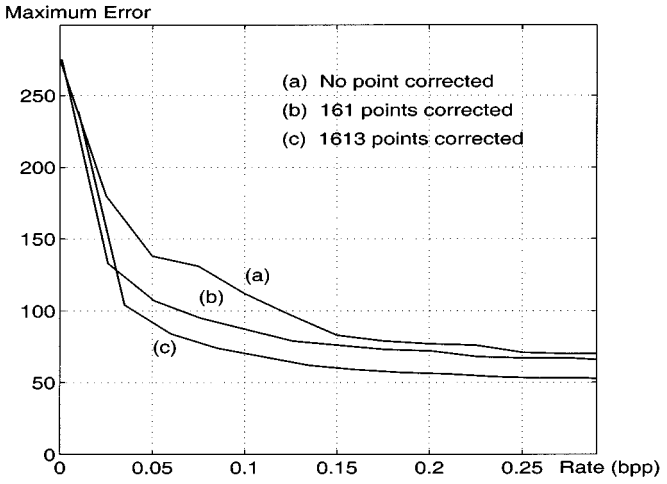


Fig. 19. Maximum error as a function of the rate for the GER test image: (a) no correction of the outliers, (b) correction of 161 values (0.001 bpp of side information), and (c) correction of 1613 values (0.01 bpp of side information).

TABLE I
ERROR RATES IN THE CLASSIFICATION OF THE TM TEST IMAGE COMPRESSED AT 0.1, 0.3, AND 0.5 bpp USING TWO TO EIGHT CLASSES

classes	misclassification rate (%)		
	0.1 bpp	0.3 bpp	0.5 bpp
2	8.91	5.38	3.95
3	11.42	6.89	4.93
4	16.30	9.60	6.89
5	18.17	10.75	7.79
6	20.82	12.73	9.32
7	28.44	17.95	13.37
8	31.46	20.11	15.15

$2, \cdots, 8$. In the absence of ground truth data, the classification of the original image is assumed as a reference against which to compare the results of the classification of the compressed images.

We report the probability of misclassification for $N = 2$ to 8 classes and for some values of the rate in Table I for the TM image and Table II for the GER image. The results are in good agreement with the SNR and maximum error performance, namely, the TM image needs at least 0.3 bpp to achieve a reasonable level of accuracy, and results improve steadily with the rate. Much better results are obtained for the GER image already at 0.2 bpp, while going to 0.3 bpp does not reduce significantly the error. When the number of classes increases, the misclassification rate becomes relatively large, but this comes as no surprise given the very high activity of these images. Better results could probably be achieved using context-based classifiers and having some prior knowledge on the images.

TABLE II
ERROR RATES IN THE CLASSIFICATION OF THE GER TEST IMAGE COMPRESSED AT 0.1, 0.2, AND 0.3 bpp, USING TWO TO EIGHT CLASSES

classes	misclassification rate (%)		
	0.1 bpp	0.2 bpp	0.3 bpp
2	4.16	2.42	2.21
3	9.26	5.56	5.15
4	13.33	8.15	7.52
5	17.74	10.86	10.47
6	21.14	12.83	12.31
7	23.02	13.99	13.27
8	27.57	16.96	15.65

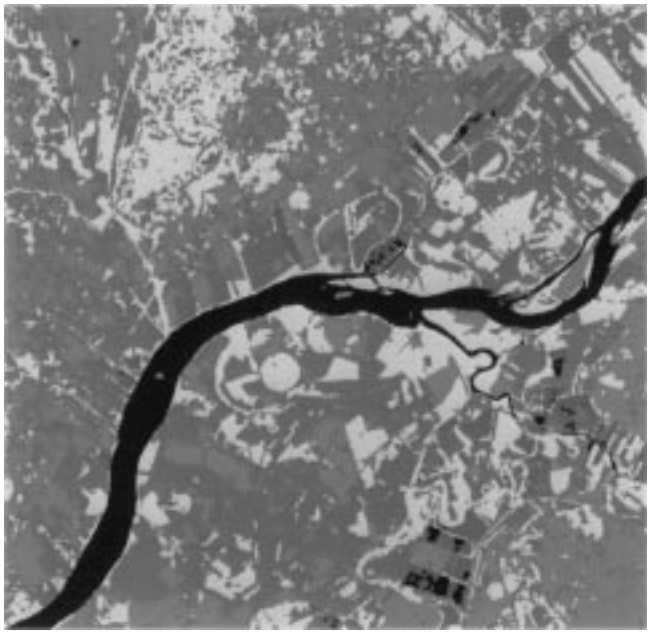
In particular, note that these results are rather conservative since the original image is characterized itself by a nonzero misclassification rate with respect to the (unknown) ground truth, due to the presence of noise. Compression tends to smooth out edges (introducing errors) but also to filter noise (correcting errors). Only the net increase should be really accounted for. It is not easy to tell the classification of the original from classification of the compressed image, as evidenced by Fig. 20 with reference to the TM, $N = 5$ classes, and 0.3 bpp.

V. CONCLUSIONS AND FUTURE RESEARCH

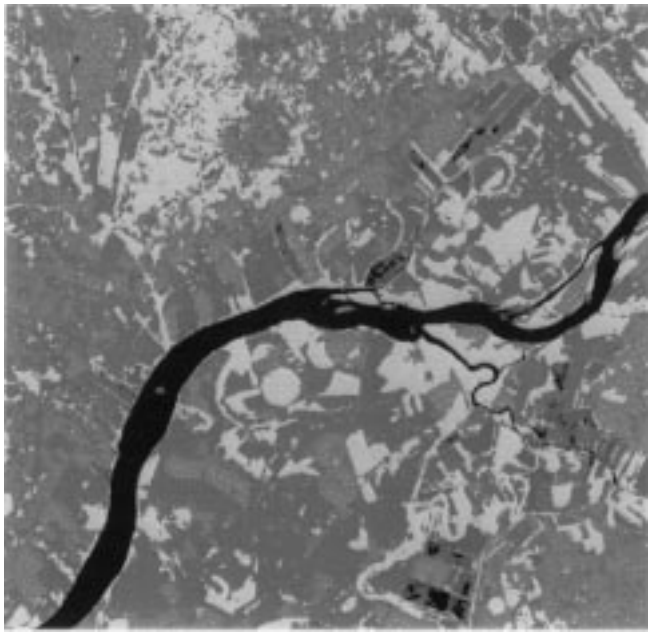
In this paper, we proposed several modified SPIHT algorithms for the compression of multispectral images, using either KLT or VQ in the spectral dimension in order to take advantage of the strong interband dependencies that such images exhibit. Our experiments show an approximately equivalent rate-distortion performance for the various algorithms proposed, but the nonadaptive KLT-based version (3-D-SPIHT) seems to be preferable because of its lower design complexity. It also compares favorably with other techniques proposed recently for the same task, both when multispectral and hyperspectral images are considered.

To better assess the absolute quality of the compressed images, we considered, besides the usual SNR measure, visual inspection, maximum error measure, and misclassification rate. According to all measures, it is clear that the encoding is far from being lossless or near-lossless, as is also obvious given the compression ratios used, 16 : 1 or more for the TM image, 30 : 1 or more for the GER. However, it is also clear that often these requirements are not necessary, as the degradation introduced by compression is much less than the intrinsic noise level of the images.

Work is currently under way to investigate in more depth this point. In particular, a joint project with other research groups in



(a)



(b)

Fig. 20. Five-class minimum-distance classification of (a) the TM test image and (b) the TM test image compressed at 0.3 bpp.

Italy is under development with the goal (among others) to assess the quality of compressed remote sensing images for environment monitoring application. More immediate experiments will concern the use of contextual classifier (based on Markov random field models) which can partly make up for the noise of the image and provide more reliable results.

APPENDIX
VQ CODEBOOK DESIGN

In this Appendix, we describe the design of the tree-structured gain-shape codebooks used in the gs-SPIHT algorithm. The reader is assumed to be already familiar with the basic concepts of vector quantization [3], and especially with the GLA (or LBG algorithm) for the codebook design.

We want to design a product codebook where both component codebooks are tree-structured, so as to carry out low-complexity progressive encoding of large vectors. In order to obtain a good performance the two codebooks should be designed jointly. However, we show that a truly optimal design is not feasible, and propose a simple, greedy, design procedure which provides satisfactory experimental performance.

For the sake of clarity, we will now make a number of simplifying assumptions, to be removed later. So, let us suppose that both component codebooks, say $\mathcal{G}$ and $\mathcal{S}$, are described by *balanced* trees, and have the same size N; in addition, let us assume that for each input vector $\boldsymbol{x}$ the encoder sends alternately one bit for the gain and one for the shape. Under these conditions, a joint codebook design procedure is readily outlined. Let

$$\{\mathcal{G}^n, \quad n = 2^0, 2^1, \cdots, 2^B = N\} \tag{5}$$

be the set of codebooks corresponding to the various layers of the gain tree, where

$$\mathcal{G}^n \equiv \{\hat{g}_1^n, \hat{g}_2^n, \cdots, \hat{g}_n^n\}. \tag{6}$$

The last layer corresponds to the full-resolution codebook $\mathcal{G}^N$, while $\mathcal{G}^1$ is composed of a single element, $\hat{g}_1^1$ which is the root of the tree. Analogous definitions will hold for the shape codebooks, with the only difference that these will be composed by unit-norm vectors, rather than scalars.

Now, it is a trivial task to design the optimal size-1 codebooks $\mathcal{G}^1$ and $\mathcal{S}^1$ over the training set $\mathcal{T} \equiv \{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_M\}$. It results in

$$\hat{g}_1^1 = ||\overline{\boldsymbol{y}}||, \quad \hat{\mathbf{s}}_1^1 = \frac{\overline{\boldsymbol{y}}}{||\overline{\boldsymbol{y}}||} \tag{7}$$

where $\overline{\boldsymbol{y}}$ is the centroid of $\mathcal{T}$. At 0 bits, all input vectors $\boldsymbol{x}$ are reproduced as $\hat{g}_1^1 \hat{\boldsymbol{s}}_1^1$. If a single encoding bit per vector is available, for the assumptions made above it goes to refine the gain, so we should now design $\mathcal{G}^2$. This is a straightforward task as well, since it simply consists in the application of the ordinary GLA. The gain root is split in two new values, say, $\hat{g}_L = \hat{g}_1^1 + \epsilon$ and $\hat{g}_R = \hat{g}_1^1 - \epsilon$, which, given $\hat{\boldsymbol{s}}_1^1$, partition the training set in $\mathcal{T}_L$ and $\mathcal{T}_R$ according to a minimum distortion criterion. Then, the values of $\hat{g}_L$ and $\hat{g}_R$ and the corresponding partition are updated until convergence. A similar procedure is carried out to design all other layers, for example, $\mathcal{S}^2$ given $\mathcal{G}^2$, where each couple of codewords is designed on the subtraining set relative to their parent codeword.

Note that, contrary to what happens in conventional product-codebook VQ, each codebook is designed *given*, rather than *jointly with*, some other codebooks. This makes the design very easy, just like in ordinary TSVQ, but also introduces some performance loss.

Now, let us remove the simplifying assumptions considered above, beginning with the fixed encoding path (alternation of gain and shape bits). Indeed, it is easy to envision situations where an unequal resource assignment is more desirable. The design procedure needs only minor changes. Suppose $\mathcal{G}^n$ and $\mathcal{S}^m$ are the last codebooks already designed; now, both $\mathcal{G}^{2n}$

given $\mathcal{S}^m$ and $\mathcal{S}^{2m}$ given $\mathcal{G}^n$ are designed and the corresponding encoding distortion over the training set is computed: the codebook that leads to the smallest distortion is accepted, while the other one is simply discarded. Starting from the roots, the whole encoding path is singled out.

Another unnecessary constraint is that the tree be balanced and, more basically, that it grow one layer at a time rather than one node at a time. By removing such a constraint we are bound to design a sequence of gain and shape codebooks $\{\mathcal{G}^1, \mathcal{G}^2, \mathcal{G}^3, \cdots\}$ and $\{\mathcal{S}^1, \mathcal{S}^2, \mathcal{S}^3, \cdots\}$ where the $(n+1)$st is obtained by the nth through the split of a single node. Again, given $\mathcal{G}^n$ and $\mathcal{S}^m$ one has to decide which node of which tree to split. This is conceptually simple, as we will see, but computationally heavy, and in practice we will use a simplified version. At any given moment, through the encoding rule, the two codebooks partition the training set (and the input space) in $n \times m$ regions (Fig. 21): $\mathcal{T}_{ij}$ comprises all the training vectors that are encoded by $\hat{g}_i^n$ and $\hat{\boldsymbol{s}}_j^m$. Accordingly, we define

$$\mathcal{T}_{i,*} \equiv \bigcup_j \mathcal{T}_{i,j}, \quad \text{and} \quad \mathcal{T}_{*,j} \equiv \bigcup_i \mathcal{T}_{i,j}. \tag{8}$$

Following the usual approach for the growth of unbalanced TSVQ codebooks [3] we now want to carry out the split that maximizes the slope along the empirical rate-distortion curve, namely, the ratio $|\Delta D/\Delta R|$, where ΔD is the decrease in distortion and ΔR is the increase in rate deriving from the split. Taking node i of the gain tree, for example, these quantities are readily evaluated as

$$\Delta R = \frac{M_i}{M} \tag{9}$$

$$\Delta D = \frac{1}{M} \sum_{\boldsymbol{y} \in \mathcal{T}_{i,*}} \cdot \left[\|\boldsymbol{y} - \hat{g}^n(\boldsymbol{y})\hat{\boldsymbol{s}}^m(\boldsymbol{y})\|^2 + \|\boldsymbol{y} - \hat{g}^{n+1}(\boldsymbol{y})\hat{\boldsymbol{s}}^m(\boldsymbol{y})\|^2 \right] \tag{10}$$

where M_i indicates the cardinality of $\mathcal{T}_{i,*}$, while $\hat{g}^n(\boldsymbol{y})$ and $\hat{\boldsymbol{s}}^m(\boldsymbol{y})$ are the gain and shape selected by the encoder for $\boldsymbol{y}$ in the codebooks $\mathcal{G}^n$ and $\mathcal{S}^m$. In a similar way, the same quantities are evaluated for the generic node j of the shape tree. The split that produces the maximum rate-distortion benefit is then accepted and the corresponding codebook is updated accordingly. We explicitly note that this is a *greedy* procedure, since the long-term effects of a split are not taken into account.

Unfortunately, this procedure is computationally very intensive, opposite to our goal. Therefore, as a reasonable compromise between complexity and performance, we decided to grow the gain codebook one layer at a time and to retain the regular structure used in the original SPIHT. On the contrary, the shape codebook is grown one node at a time; the choice between growing a layer of the gain codebook or a node of the shape codebook is again based on the slope of the rate-distortion curve. Finally, since in the SPIHT algorithm a vector whose norm is in the range $T/2-T$ is automatically assigned the initial reproduction gain $3T/4$ upon discovery, we consider the root gain codebook $\mathcal{G}^1$ as composed by the collection of all these initial values $3T/4$, $3T/8$, $\cdots$ rather than by a single root value. It goes by itself that this difference does not change the structure of the codebook design algorithm, but only its implementation.

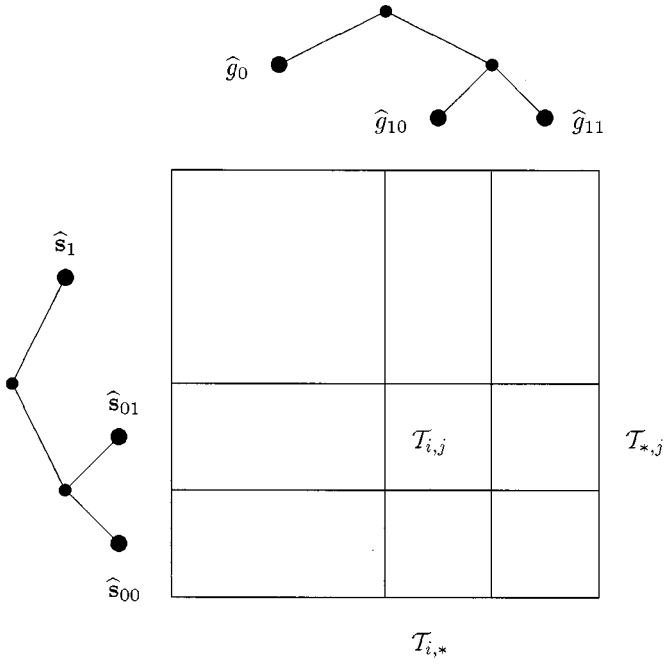


Fig. 21. Schematic representation of the tree-structured gain and shape codebooks and of the induced partition of the input space.

REFERENCES

[1] A. N. Netravali and B. G. Haskell, *Digital Pictures, Representation and Compression*. New York: Plenum, 1988.

[2] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proc. IEEE*, vol. 73, pp. 1551–1588, Nov. 1985.

[3] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.

[4] R. L. Baker and Y. T. Tse, "Compression of high spectral resolution imagery," *Proc. SPIE*, no. 974, pp. 255–264, 1988.

[5] S. Gupta and A. Gersho, "Feature predictive vector quantization of multispectral images," *IEEE Trans. Geosci. Remote Sensing*, vol. 30, pp. 491–501, May 1992.

[6] G. R. Canta and G. Poggi, "Compression of multispectral images by address-predictive vector quantization," *Signal Process.: Image Commun.*, vol. 11, pp. 147–159, Dec. 1997.

[7] ——, "Kronecker-product gain-shape vector quantization for multispectral and hyperspectral image coding," *IEEE Trans. Image Processing*, vol. 7, pp. 668–678, May 1998.

[8] J. A. Saghri, A. G. Tescher, and J. T. Reagan, "Practical transform coding of multispectral imagery," *IEEE Signal Processing Mag.*, pp. 32–43, Jan. 1995.

[9] G. P. Abousleman, M. W. Marcellin, and B. R. Hunt, "Compression of hyperspectral imagery using the 3-D DCT and hybrid DPCM/DCT," *IEEE Trans. Geosci. Remote Sensing*, vol. 33, pp. 26–34, Jan. 1995.

[10] G. Gelli and G. Poggi, "Compression of multispectral images by spectral classification and transform coding," *IEEE Trans. Image Processing*, vol. 8, pp. 476–489, Apr. 1999.

[11] S. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 2091–2110, Dec. 1989.

[12] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[13] P. C. Cosman, R. M. Gray, and M. Vetterli, "Vector quantization of image subbands: A survey," *IEEE Trans. Image Processing*, vol. 5, pp. 202–225, Feb. 1996.

[14] B. R. Epstein, R. Hingorani, J. M. Shapiro, and M. Czigler, "Multispectral KLT-wavelet data compression for Landsat thematic mapper images," in *Proc. Data Compression Conf.*, Snowbird, UT, Apr. 1992, pp. 200–208.
[15] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.
[16] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, June 1996.
[17] F. Amato, C. Galdi, and G. Poggi, "Embedded zerotree wavelet coding of multispectral images," in *Proc. Int. Conf. Image Processing*, Santa Barbara, CA, Oct. 1997, pp. 612–615.
[18] A. Bilgin, G. Zweig, and M. W. Marcellin, "Efficient lossless coding of medical image volumes using reversible integer wavelet transforms," in *Proc. 1998 Data Compression Conf.*, Snowbird, UT, Mar. 1998.
[19] Z. Xiong, O. Guleryuz, and M. T. Orchard, "A DCT-based embedded image coder," *IEEE Signal Processing Lett.*, Nov. 1996.
[20] J. J. Benedetto and M. W. Frazier, *Wavelets: Mathematics and Applications*. Boca Raton, FL: CRC, 1994.

Pier Luigi Dragotti was born in Naples, Italy, on May 11, 1971. He received the Laurea degree in telecommunications engineering in March 1997, from the University Federico II, Naples. From March to September 1996, he was a Visiting Student at Stanford University, Stanford, CA, at the Signal Compression and Classification Group directed by Prof. R. Gray. From October 1997 to July 1998, he attended the Doctoral School in Communications Systems at the Swiss Federal Institute of Technology, Lausanne, Switzerland, where he is currently a Ph.D. student in the Audiovisual Communications Laboratory directed by Prof. Martin Vetterli. His research interests include image and video compression, joint source-channel coding, scalar and vector quantization.

Giovanni Poggi was born in Naples, Italy, on February 17, 1963. He received the Laurea degree in electronic engineering in July 1988 from the University Federico II, Naples.

From 1990 to 1998, he was a Researcher in the Department of Electronic and Telecommunication Engineering, University Federico II, and since 1998, has been an Associate Professor in the same institution. In 1992, he was a Visiting Scholar in the Department of Electrical Engineering, Stanford University, Stanford, CA. His research activity is in signal processing, in particular, on vector quantization techniques for low rate encoding of still images, compression of medical images, and, more recently, classification and compression of remote sensing images.

Dr. Poggi has been a reviewer for IEEE TRANSACTIONS ON IMAGE PROCESSING and IEEE TRANSACTIONS ON CIRCUIT AND SYSTEMS FOR VIDEO TECHNOLOGY.

Arturo R. P. Ragozini was born in Genoa, Italy, on August 3, 1972. He received the Laurea degree in telecommunications engineering in May 1996, from the University Federico II, Naples, Italy. Since the academic year 1996–1997, he has been a Ph.D. student in the Department of Electronic and Telecommunication Engineering of the University Federico II. In the academic year 1998–1999, he was a Visiting Student at the ENST, Paris, France, working on the performance evaluation of PNNI-driven ATM networks. His research interests are in the compression and classification of remote sensing images and in the modeling of telecommunication systems.

From July 1996 to January 1997, he was a Frequency Management Engineer at Eutelsat, Paris, France.