Available online at www.sciencedirect.com
ScienceDirect

Signal Processing: *Image Communication* 21 (2006) 850–861

SIGNAL PROCESSING: IMAGE COMMUNICATION
www.elsevier.com/locate/image

Low-complexity compression of multispectral images based on classified transform coding

Marco Cagnazzo, Luca Cicala, Giovanni Poggi*, Luisa Verdoliva

Dipartimento di Ingegneria Elettronica e delle Telecomunicazioni, Università Federico II di Napoli, Italy

Received 7 April 2006; received in revised form 4 August 2006; accepted 21 August 2006

Abstract

Compression of remote-sensing images can be necessary in various stages of the image life, and especially on-board a satellite before transmission to the ground station. Although on-board CPU power is quite limited, it is now possible to implement sophisticated real-time compression techniques, provided that complexity constraints are taken into account at design time. In this paper we consider the class-based multispectral image coder originally proposed in [Gelli and Poggi, Compression of multispectral images by spectral classification and transform coding, IEEE Trans. Image Process. (April 1999) 476–489 [5]] and modify it to allow its use in real time with limited hardware resources. Experiments carried out on several multispectral images show that the resulting unsupervised coder has a fully acceptable complexity, and a rate–distortion performance which is superior to that of the original supervised coder, and comparable to that of the best coders known in the literature.

Keywords: Multispectral images; Remote sensing; Image compression; Region-based coding; Low complexity

1. Introduction

Satellite-borne sensors have ever higher spatial, spectral and radiometric resolution. With this wealth of information comes the problem of dealing with very large volumes of data, in every stage of the data life. The most critical phase is on-board the satellite, where acquired data easily exceed the capacity of the downlink transmission channel, and often large parts of images must be simply discarded, but similar issues arise in the ground segment, where image archival and dissemination are seriously undermined by the sheer amount of data to be managed. To avoid these problems one can resort to data compression, which allows one to reduce the data volume by one and even two orders of magnitude without serious effects on the image quality and on their diagnostic value for subsequent automatic processing. To this end, however, one cannot resort to general purpose techniques as they do not exploit the peculiar features of multispectral[1] remote-sensing images, which is why several ad hoc coding schemes have been proposed in recent years.

*Corresponding author. Tel.: +39 081 7683151; fax +39 081 7683149.
E-mail address: poggi@unina.it (G. Poggi).

[1] Although we focus on multispectral images, most of the considerations and techniques apply just as well to hyperspectral images.

0923-5965/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2006.08.003

The most popular approach is by far transform coding, for several reasons. In fact, transform coding techniques are well established and deeply understood, they provide excellent performances in the compression of images, video and other sources, have a reasonable complexity and, not least, are at the core of well-known and efficient standards such as JPEG and JPEG2000, implemented in coders widely used and easily available to the scientific community [16]. As a matter of fact, a common approach for coding multispectral images [14,11] is to use some decorrelating transforms along the spectral dimension followed by JPEG2000 on the transform bands with a suitable rate allocation among the bands. Viable alternatives include the use of wavelet transform (WT) followed by SPIHT [3,15], schemes based on the more traditional discrete cosine transform (DCT) [12,1], or on other application-oriented transforms, e.g., [4].

Less attention has been devoted to techniques based on vector quantization (VQ) because, despite its theoretical optimality [7], VQ is too computationally demanding to be of any practical use. Nonetheless, when dealing with multiband images, VQ is a natural candidate, because the elementary semantic unit in such images is the spectral response vector (or spectrum, for short) which collects the image intensities for a given location at all spectral bands. The values of a spectrum at different bands are not simply correlated but strongly dependent, because they are completely determined (but for the noise) by the land covers of the imaged cell. This observation has motivated the search for constrained VQ techniques [2,10,9], which are suboptimal but simpler than full-search VQ, and show promising performances.

We focus here on the hybrid coding scheme originally proposed in [5], where VQ is used only in the first and more critical coding phase, to be followed later by simpler transform techniques in order to encode the residuals. In this approach, apart from being a first encoding step, VQ has the central role of segmenting the image in homogeneous classes: the VQ residuals are then grouped according to their class, so that subsequent transform coding techniques can operate on each class separately, and adapt to their specific statistics. Thanks to the joint use of VQ, which fully exploits the strong interband dependencies of multispectral images, and of class-adaptive transforms, which concentrate in a few coefficients most of the residuals' energy, this coding scheme (referred from now on as CBC, for class-based coder) provides a very good performance.

It must be underlined that this is a *supervised* coder, that relies on several pieces of information (to begin with the VQ classifier and the covariance matrices of the classes) which are supposed to be known or computed in advance. These parameters must be either retrieved from a very large database, covering virtually all land covers of possible interest in all atmospheric and illumination conditions, or computed before actual coding, and in both cases, on-board real-time coding is not possible. This could not be an issue until a few years ago, but the steady increases in computation power now opens the door to real-time operations, that is, to compression on-board the satellite, where it is needed the most, before transmission to the ground station.

Given the interesting performance of the CBC, and the opportunities offered by the classification-based approach, in this paper we address the problem of modifying the original coder so as to make it fully unsupervised and suitable for on-board operation. This requires that all coding parameters be computed on-board at compression time and sent as side information along with the coded data, which, in turn, poses two constraints:

1. the combined complexity of the parameter design and image coding phases must remain small enough to allow for real-time operation, and
2. the information overhead associated with the parameters must be very limited so as not to impair the rate–distortion performance.

By keeping into account both the complexity and side information constraints, we designed a new CBC for multispectral images which is fully unsupervised and guarantees low computational complexity and highly competitive performance. Although the design of an actual real-time encoder is out of the scope of this work, the experimental analysis shows that the overall coding complexity is clearly within the reach of modern high-performance on-board processors [17]. In addition, experiments on real-world multispectral images prove that, even in terms of rate–distortion performance, the new unsupervised coder is superior to the original CBC as well as to state-of-the-art techniques such as 3d-SPIHT [15].

The next section reviews the CBC, highlighting both its potential and its weak points for real-time operations, and carefully estimating its encoding complexity. Section 3 describes in detail the variations w.r.t. the original coder and analyzes design complexity and side information issues. Section 4 is devoted to the experimental analysis on some real-world remote-sensing images. First, the simplified design procedure and the side information coding technique will be validated, then, the rate–distortion performance of the proposed technique will be assessed by comparing it with those of the original CBC and of a state-of-the-art wavelet-based coder. Finally, Section 5 draws conclusions.

2. The original supervised coder

In this section we first describe in some detail the CBC coding scheme, and then provide an accurate estimate of its encoding complexity in view of real-time use.

2.1. Coding scheme

CBC comprises three main steps, as summarized in Fig. 1, namely,

1. image segmentation;
2. lossless coding of the segmentation map;
3. lossy coding of the texture.

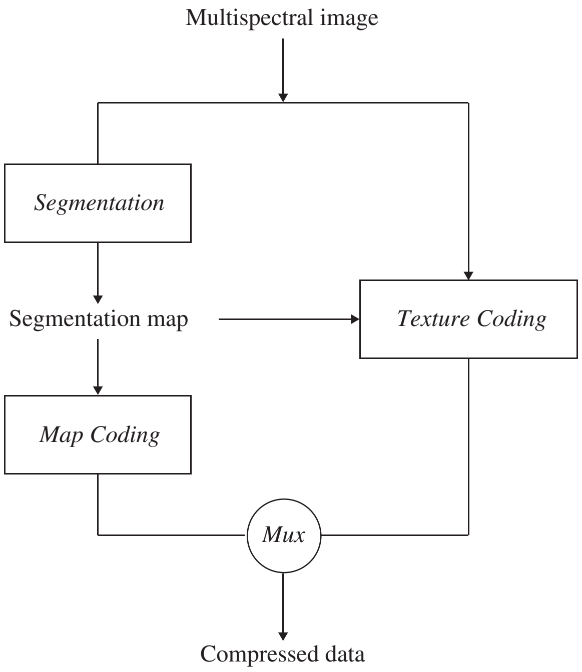

Fig. 1. The class-based coding scheme.

Segmentation amounts to a simple spectral clustering: a set of template spectral vectors is selected in advance, one for each class, and the current pixel is classified, based on its spectrum, according to a minimum-distance criterion. The set of template spectra can be viewed as a VQ codebook, and the segmentation itself as a VQ. Therefore, this segmentation step can be also regarded as a first level of image compression, followed afterwards by the transform coding of the residuals.

The main reason for segmentation, however, is to classify image pixels based on their spectra, so as to collect and encode together VQ residuals of the same class. Each set of residuals, in fact, exhibits homogeneous statistics and can be compressed very efficiently by means of conventional transform techniques.[2] In CBC a Karhunen–Loeve transform (KLT) is first performed along the spectral dimension where, in order to account for the class information, a different transformation matrix is used for each class. Then, the transform bands are scanned in order to form several sequences of coefficients, one for each band and for each class, that will be further subject to a one-dimensional DCT to exploit remaining spatial redundancies within each transform band. To maximize the correlation of coefficients encountered along the scanning path, the image is not scanned line by line but along a Peano curve. Finally, each transform coefficient is sorted by spectral class, KLT band and DCT frequency, and included in a quantization set which is quantized by a specific Lloyd–Max quantizer at a rate decided by means of a greedy bit allocation algorithm. Of course, the segmentation map, that is, the set of VQ indexes, must be transmitted as well as a side information. Since neighboring pixels are highly correlated the map is significantly compressed, without loss of information, by resorting to a predictive scheme followed by Huffman coding.

Experiments show that the classification-based coder guarantees a 2–3 dB improvement at all rates of interest w.r.t. to "flat" transform coder (the same coder with a single all-encompassing class) despite the additional cost required for the transmission of the segmentation map. In addition, the segmentation map

[2] More sophisticated algorithms can produce smoother segmentation maps than VQ, but they lead eventually to worse overall performance [6]. Under this respect, in fact, VQ represents the optimal choice, as it is designed to optimize the rate–distortion performance, namely, to minimize the energy of residuals for an assigned encoding rate.

itself is a valuable piece of information, obtained on the original uncompressed data and automatically embedded in the coding stream, which helps making this coding scheme an interesting tool for the compression of remote-sensing images.

This quick review of CBC (the reader is referred to [5] for further detail) allows us to focus on the main processing steps, that is, VQ classification, KLT spectral transform, spatial transform and quantization in the original coder, several important pieces of information (VQ classifier, KLT matrices, scalar quantizers) are all supposed to be known in advance, but in real-time operations they must be designed on-line on the data to be encoded, which adds to complexity, and transmitted as a side information, which adds to the encoding rate. This new *design* phase will be described thoroughly in the next section, but before that, it is useful to estimate the complexity of the *encoding* phase, so as to have a significant reference for complexity, and also to gain more insight about the coder functioning.

2.2. Estimate of the encoding complexity

Let us consider a multispectral image with N pixels[3]and B spectral bands. For each pixel, the segmenter must find the minimum-distance spectrum among the C template spectra available, where C is the number of classes. With full-search VQ, this operation requires BC multiplications per each spectrum, or C multiplications per sample (mps), our conventional unit of measure for complexity). Even though only small codebooks will be used (say, $C<32$), this is a non-negligible level of complexity. The CBC resorts therefore to tree-structured VQ (TSVQ) [7], which reduces complexity to about $\log_2 C$ mps (the exact value is unpredictable as it depends on the tree structure itself) at little or no cost in terms of segmentation accuracy. In TSVQ, in fact, the classes are organized as a binary tree, and the desired class is reached by means of a sequence of binary decisions, each of which requires the computation of a single scalar product.[4]

After VQ, each residual vector undergoes a KLT, and this requires B^2 multiplications per vector, that is, B mps, quite small for typical multispectral images. The complexity would become relevant for large values of B (that is, hyperspectral images), in which case, however, only a small fraction $B' \ll B$ of the transform bands carry useful information, and one can use a rectangular $B \times B'$ transform with complexity down to B' mps.

Then there is the DCT which, for a sequence of length K, has a complexity of $\frac{1}{2}K\log_2 K$ multiplications. Since all samples must be transformed, irrespective of their class and band, we must carry out about NB/K such DCTs, and hence the normalized complexity is just $\frac{1}{2}\log_2 K$ mps, not large for typical values of K.

The last step is scalar quantization, but its complexity will be neglected. In fact, it requires only comparisons, and the exact number of them depends on the bit rate allocated to the specific set to be quantized and ultimately on the overall encoding rate. As an upper bound, the number of comparisons must be less than the total number of bits sent (because some bits are used for map coding) and hence at 1 bit/sample, which is definitely high for multispectral images, there is just one comparison per sample.

Summing these results, the complexity of the encoding phase can be estimated as

$$Q_{\text{coding}} \simeq \log_2 C + B + \frac{1}{2}\log_2 K. \tag{1}$$

Of course, this is only an approximate estimation of complexity, as many CPU-time consuming phases, like data transfer, etc., are neglected, but allows us to study under what conditions is CBC amenable to on-board implementation. At first, one could require that

$$Q_{\text{coding}} < P_{\text{CPU}}/R_{\text{aq}}, \tag{2}$$

with P_{CPU} the CPU power in multiplications per second, and R_{aq} the data acquisition rate in sample/s. Assuming, for example, a 2.5 Gflops processing unit [17], and a typical data rate of 10^8 sample/s, the coder should be already able to function smoothly. In addition, CBC lends itself naturally to parallel processing, because after the VQ, which works on all the data together, C processes can go on in parallel, one for each class. Therefore, if a parallel implementation is foreseen, complexity constraints become much looser and real-time implementation undoubtedly possible.

[3] The actual number of rows and columns is irrelevant, here.

[4] A plethora of fast VQ techniques also exist, and possibly could further reduce complexity, but their investigation goes beyond the scope of this work.

3. The new unsupervised version

To realize a fully unsupervised version of CBC we must simply design on-line the VQ classifier, the KLT matrices and the scalar quantizers, but the resulting increase in complexity and side information must be carefully checked in order for the coder to remain feasible and efficient. On the up side, since all these pieces of information are designed on the same data they are used on, there should be a performance improvement which could balance or even exceed the rate penalty due to the increased side information.

3.1. VQ classifier

The design of a VQ codebook can be very demanding in terms of CPU power but, since only a limited number of land covers are typically present in a given image, we are interested in a rather small codebook. Moreover, since our classifier is tree-structured, we are not really looking for a size-C codebook, but rather for $C-1$ size-2 codebooks, which are much simpler to design. Finally, the design need not be carried out on all the data to be encoded, but only on a training set of adequate size M_{VQ}. The design is carried out by the well-known generalized Lloyd algorithm (GLA) [7], which requires a few iterations, during which each training vector is encoded by the current codebook. If we assume, for the sake of simplicity, that the classification tree is balanced, that training vectors are evenly divided between the right and left children at each node and that the GLA requires no more than I_{VQ} iterations to converge, the complexity can be estimated as $M_{\mathrm{VQ}}I_{\mathrm{VQ}}B$ multiplications for the design of the root codebook, plus another $2\times(M_{\mathrm{VQ}}/2)I_{\mathrm{VQ}}B$ for the design of the two codebooks at level one of the tree, and so on down to the bottom of the tree, for a total of $M_{\mathrm{VQ}}I_{\mathrm{VQ}}B\log_2 C$ multiplications or (M_{VQ}/N) $I_{\mathrm{VQ}}\log_2 C$ mps.

With $M_{\mathrm{VQ}}=N$ the overall cost would therefore be significant, but even a modest subsampling, that is a relatively large training set, would bring it well under the encoding complexity. For example, assuming quite conservatively $I_{\mathrm{VQ}}=10$, $C=32$, $M_{\mathrm{VQ}}=100\times C$ (a good rule of thumb) and $N=512\times 512$ we have a fully manageable complexity of less than 1 mps.

As for the side information, the VQ codebook is composed of C vectors,[5] with B components each. For any reasonable combination of the relevant parameters (image size, etc.), the cost of sending such a codebook is negligible even with a 16 bit per component coding (and can be further reduced if interband correlation is exploited), hence it will not disrupt performance. On the contrary, a good codebook designed on-line can be significantly superior to its off-line counterpart, since in the latter case the training set is not guaranteed to fit well the actual data, so we could expect some performance gain here.

3.2. Class-adapted KLT matrices

To compute a KLT transform matrix along the spectral direction, we must first estimate the $B\times B$ correlation matrix of the data, and then compute its eigenvectors. Since we use class-adaptive KLT, we need C such matrices, one for each class.

For the estimation part we resort again to some subsampling of the training data. With a training set of M_{KLT} vectors, $M_{\mathrm{KLT}}B\,(B+1)/2$ multiplications are needed, that is, $(M_{\mathrm{KLT}}/N)(B+1)/2$ mps, which is not significant if the training set has a reasonable size, for example, $M_{\mathrm{KLT}}=100C$ as before.

Computing the eigenvectors with standard inversion procedures (tridiagonalization/Jacobi) instead, has a complexity of about $2B^2(B+1)I_{\mathrm{KLT}}$ for each KLT matrix [8], where I_{KLT} is the number of iterations required, for a total of $2B(B+1)$ CI_{KLT}/N mps. For multispectral images, where B is rather small, this is certainly affordable; with $C=32$, $N=512\times 512$ and $I_{\mathrm{KLT}}=10$, for example, complexity remains under 1 mps even for a 20-band image. For hyperspectral images, on the other hand, since only the $B'\ll B$ dominant eigenvectors are required, one can use much faster techniques, such as the power method [8], and keep complexity under control.

Concerning the side information, for each of the C KLT matrices we must send $B(B+1)/2$ parameters which could become a burden only in some non-typical conditions (small images, low coding rates, many classes, many bands). Therefore, care must be taken to avoid such cases and to encode all parameters with as few bits as possible, without significant performance losses. On the pro-side, like before, using matrices computed on-line cannot but improve the compression performance.

[5] Intermediate TSVQ vectors are not needed at the receiver.

3.3. Set-adaptive quantizers

Here, we must definitely depart from the original coding scheme, not so much for the design complexity but because the side information required to transmit all the Lloyd–Max quantizers is obviously unacceptable. In fact, an ad hoc quantizer is used for the first DCT coefficient of the first KLT band of the first class, another one for the second DCT coefficient of the first KLT band of the first class and so on, for a total of *CBK* quantizers in the most general case. Even considering that most of these sets of coefficients will be assigned no encoding bits, and no information needs to be transmitted for them, many quantizers remain to be designed and transmitted that this approach, which made perfect sense in the context of a supervised algorithm, becomes unviable for the new real-time unsupervised coder.

We resort therefore to parametric quantizers: each set of coefficients is modeled as either Gaussian (low frequencies) or Laplace (high frequencies), and characterized by its variance, thus univocally identifying the corresponding parametric quantizer. To preserve the scalability of the original scheme, we keep using tree-structured quantizers, and make them mid-tread to increase robustness, so that at each level there is a ternary node besides the binary ones as shown in Fig. 2. The coefficient variances are then used to perform rate allocation by means of the Huang–Schultheiss algorithm [7].

The design complexity lies only in the variance estimation. With no subsampling, all samples appear in the estimate of a single parameter, hence the complexity is just 1 mps. Subsampling schemes could reduce this cost, but this is clearly not an issue.

Side information is represented by the set variances, and therefore a total of *CBK* parameters, even more than for the KLT matrices. As noted above, however, a large number of these sets will have very low variance, and hence will be assigned no quantization bits. Therefore, two pieces of side information will be actually sent, a flag bit for each set which indicates whether it is active (bit assigned) or not, and the variance for the active sets only. The overall cost is therefore $CBK(1+\alpha(R)\beta)$ where the fraction of active sets $\alpha(R)$ is smaller at low encoding rates R, easing the possible problems, and β indicates the number of bits used to encode each parameter.

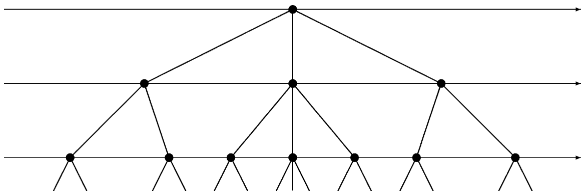

Fig. 2. Tree-structured mid-tread scalar quantizer.

In summary, the total design complexity can be estimated as

$$Q_{\text{design}} \simeq \frac{M_{\text{VQ}}}{N} I_{\text{VQ}} \log_2 C + \frac{M_{\text{KLT}}}{N}(B+1)/2 + \frac{2}{N} C I_{\text{KLT}} B(B+1) + 1. \tag{3}$$

Even though the relative weight of coding and design complexity, as expressed by Eqs. (1) and (3), depends on a number of parameters that can vary wildly, we can already draw some conclusions. Assuming that the training set used in the design phase is much smaller than the whole image, and that $I_{\text{VQ}} < 10$, as always happened in our experiments, the first and second terms of Q_{design} are negligible w.r.t. the corresponding terms of Q_{coding}. Moreover, the contribution due to scalar quantization is always negligible. As for the third term of Q_{design}, it is negligible as well, except when the number of bands is comparable with the spatial dimension of the image, that is, for hyperspectral images. In such a case, however, only a few eigenvectors $B' \ll B$ really need be computed, as already said, and Eq. (3) should be modified accordingly leading again to a perfectly manageable complexity. In conclusion, whether or not design complexity can be considered negligible depends essentially on the ratio between the training set size (for both VQ and KLT) and the image size.

4. Experimental analysis

In this section, we investigate the actual complexity and performance of the proposed coding scheme by means of numerical experiments on real-world remote-sensing images, with the goal to show that the additional complexity required by the new design phase is practically negligible, and hence the whole coder could work in real time on-board a satellite, and that the rate–distortion performance is even superior to that of the original coder. Our initial tests are conducted on the same image used in [5] where CBC was originally proposed, a 6-band Landsat TM multispectral image of a rural region near Lisbon in Portugal, so as to obtain comparable results. Similar experiments were carried out on other test images, and we report the results obtained on a 4-band IKONOS multispectral image of the San Diego, CA, area, which has features that are markedly different from those of the first image, and on a 32-band GER hyperspectral image of a region near the river Rhein in Germany.

4.1. Complexity

In Section 3 we have seen that design complexity depends primarily on the size of the training sets. Therefore, we carry out some experiments to establish what subsampling rates can be considered safe for the design of VQ, KLT and scalar quantizers. More precisely, for various number of classes C and various coding rates, we perform the complete design and coding cycle with smaller and smaller training sets and measure the ensuing performance degradation. Figs. 3 and 4 show the results of two such experiments conducted on the TM image, with 20 classes, and coding rates of 0.25 and 0.50 bit/sample. The abscissa reports the log of the subsampling rate, going from 0 (no subsampling) to 13 (only one sample in 8192 included in the training set). The ordinate reports the signal-to-noise ratio [6](SNR) obtained when the item of interest is designed subject to subsampling while the others are designed on the full training set.

We observe a common behavior in these two cases as well as in other ones not reported here. Reducing the VQ training set size has little or no effect on performance up to $M_{\mathrm{VQ}} = 256$ (that is, a subsampling ratio of 1024), a much smaller size than indicated by our rule of thumb. Beyond that point, performance begins to decline significantly, also because the TSVQ has a hard time identifying 20 different classes in the training set. The KLT appears to be even more resilient to subsampling, with stable performance (but for some obvious random oscillations) in a wide range. Things are definitely different for the scalar quantizers, where even a moderate subsampling causes a clear loss. This happens both because available data are scarcer here, and because this is the last processing step, and errors in this phase cannot be recovered by some further processing. However, remember that SQ has such a limited complexity that subsampling is not at all necessary.

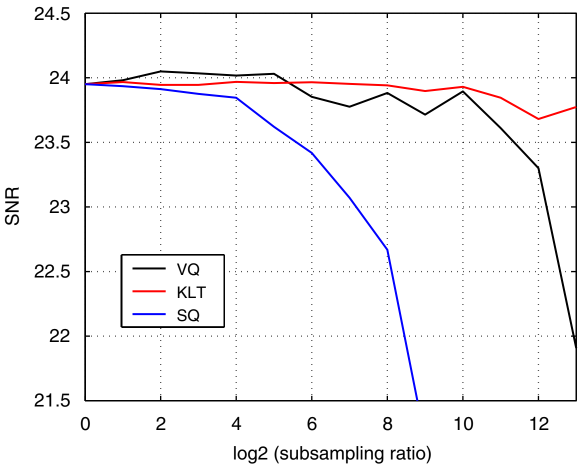

Fig. 3. Effects of subsampling at 0.25 bit/sample. The subsampling ratio goes form 1:1 to 1:8192.

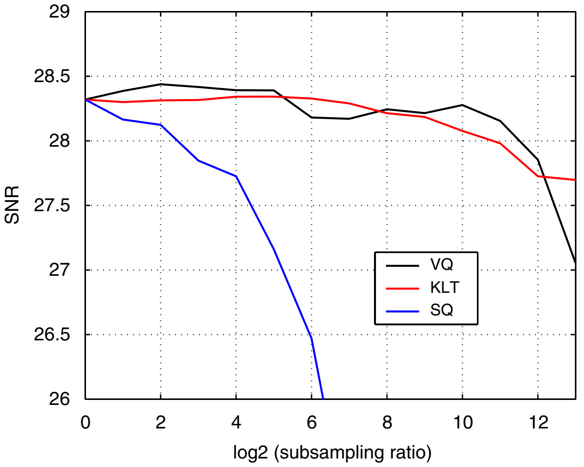

Fig. 4. Effects of subsampling at 0.50 bit/sample. The subsampling ratio goes form 1:1 to 1:8192.

Although joint effects are not analyzed here, experiments show that a moderate subsampling rate for both VQ and KLT, such as our 100 × C rule, entails no appreciable performance impairment. Therefore, in the following, we will consider this rule for subsampling.

Fig. 5 reports the design (solid lines) and coding (dashed lines) complexity for a 512 × 512 pixel image as a function of the number of bands B and of classes C. Curves are computed via Eqs. (1) and (3), considering $I_{\mathrm{VQ}} = I_{\mathrm{KLT}} = 10$ and $K = 64$, and

[6] In all experiments on this image, we follow the conventions adopted in [5], so the bands are all normalized to unit power, and the SNR is defined as $10\log_{10}(1/\mathrm{MSE})$, with MSE the mean squared error. Later on we will use the more common definition with signal variance replacing signal power.

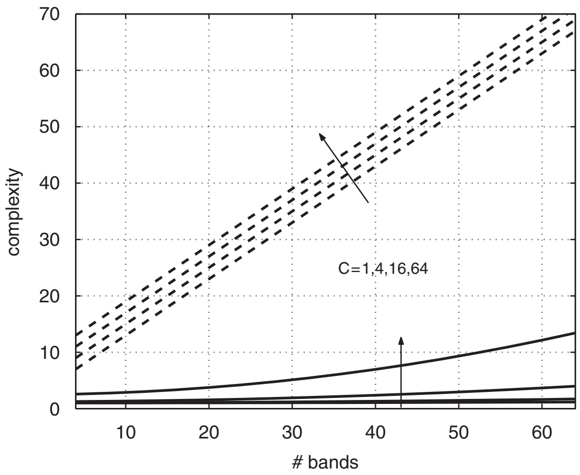

Fig. 5. Coding complexity vs. design complexity for U-CBC.

assuming $M_{\rm VQ} = M_{\rm KLT} = 100C$. In all situations, design complexity is between one and two orders of magnitude smaller than coding complexity, hence definitely negligible. Needless to say, such a proportion varies with the parameter values, and becomes more critical, for example, when a much smaller image is encoded, but this does not seem to be a reasonable instance, and our choice of parameters is instead rather conservative.

4.2. Rate– distortion performance

We can now turn to examine the rate–distortion performance. In Fig. 6 we report the performance on our test image of the original CBC coder (black), our unsupervised version U-CBC (red) and the widely known SPIHT algorithm (blue), originally proposed by Said and Pearlman [13], adapted for multispectral images [15], and made available by the authors at [18]. In this experiment, we actually used a further modified SPIHT coder [3] where KLT is used in place of the WT in the spectral direction, which provides a better performance and allows one to treat images with an arbitrary number of bands.

First of all, we see that both versions of CBC perform at least equally well than the reference SPIHT-based technique, justifying our interest towards the classification-based approach. What is more interesting for us is that, despite the additional side information required for VQ codebook, KLT matrices and quantization parameters, and despite the subsampling introduced in the design phase, our unsupervised version outperforms the original CBC coder (and SPIHT) by 1–2 dB at all rates of interest.

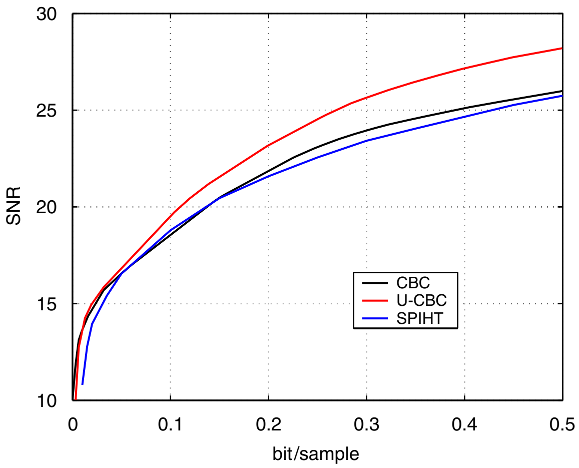

Fig. 6. Rate–distortion performance for the TM test image.

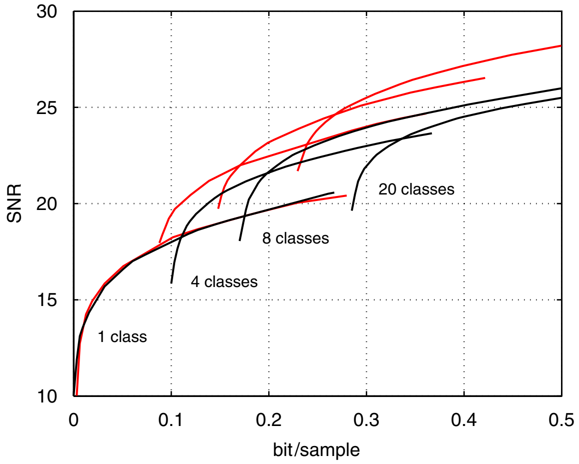

Fig. 7. The RD curves for the TM test image and various number of classes.

In other words, using coding tools tuned on the data to be encoded is more than worth the cost, and this gain comes on top of the opportunity to carry out all operations on-board, with no need for external pieces of information.

Note that CBC curves in Fig. 6 are computed as upper envelopes of actual rate–distortion curves obtained for different values of the number of classes C. In Fig. 7 we report some such curves, with $C = 1, 4, 8, 20$, for CBC (black) and U-CBC (red), in order to gain better insight about the difference in their performance. First of all, the curves for $C = 1$, where no classification is carried out, exhibit the worst performance but for extremely low bit rates. All other curves start from a non-zero initial rate,

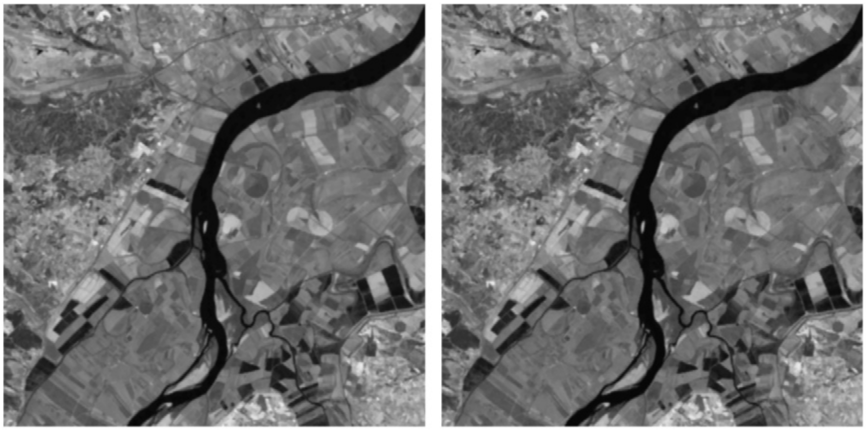

Fig. 8. Band 5 of the TM image before and after encoding at 0.40 bit/sample.

corresponding to the cost of side information, and a relatively high initial value of SNR, corresponding to the first step of encoding provided by VQ. By looking at these starting points, it is obvious that unsupervised CBC benefits from a significant advantage, both in rate and distortion, because of the higher fidelity and smoothness of the classification maps, and such an advantage increases with increasing number of classes. In fact, it is relatively easy to single out 20 meaningful classes within an image, but is very difficult that exactly the same classes are found in another image, although very similar. Therefore the off-line approach of CBC shows some fatigue for large values of C, to the point that the 20-class curve never exceeds the 8-class curve, while the on-line design is effective in all situations. In addition, even though VQ exploits many redundancies in the image, the classified KLT is still able to improve quality, resulting in a steep initial increase of all rate–distortion curves. The use of parametric rather than ad hoc quantizers does not seem to produce appreciable degradations.

To complete this analysis, Fig. 8 shows band 5 of our test image and the same band compressed with unsupervised CBC at 0.40 bit/sample, using a 20-class segmentation. Even at such relatively high compression ratio (20:1) no coding artifacts are visible.

Similar results were obtained for all multispectral images available to us. As an example, let us consider a 448×448 section of the San Diego IKONOS test image, which comprises only four bands and has a higher spatial resolution then the TM image, 4 m instead of 20 m. Fig. 9 reports the rate–distortion performance[7] of U-CBC (red), obtained again as the upper envelope of curves for fixed number of classes, and compares it with the performance of SPIHT, both in the original version (dashed blue) which uses the WT in the spectral dimension, and is now applicable since the number of bands is a power of two, and in our modified version (solid blue) with KLT in the spectral dimension. Results confirm the general behavior observed for the TM image: except for very low bit rates, which are of little interest given the low SNR level, the comparison speaks always in favor of U-CBC, which gains 1–2 dB w.r.t. the KLT-based SPIHT and up to 3 dB w.r.t. the original WT-based version.

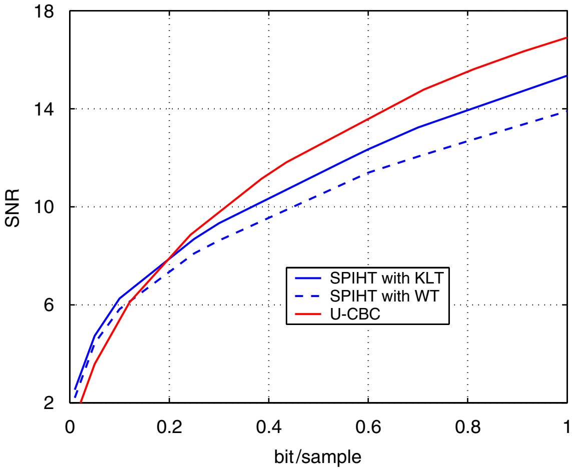

Fig. 9. Rate–distortion performance for the IKONOS test image.

[7] From now on, we adopt the usual definition of SNR as $10\log_{10}(\mathrm{VAR}(X)/\mathrm{MSE})$.

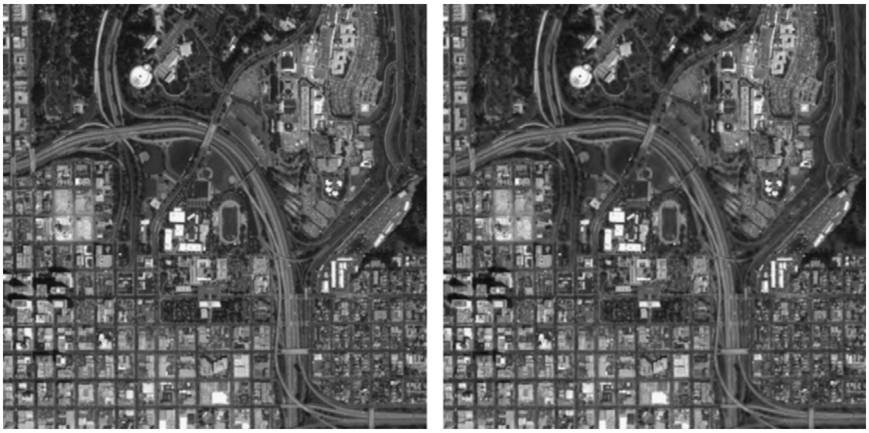

Fig. 10. Green band of the IKONOS image before and after encoding at 0.80 bit/sample.

Fig. 10 reports the original green band of our test image and the same band encoded by U-CBC at 0.80 bit/sample, with 20-class segmentation, confirming again the accuracy of the encoding process. Note that a higher bit rate is necessary here to obtain a fully satisfactory result both because the image is much richer in details and because there is a smaller number of bands to be coded jointly.

Finally, Fig. 11 reports the result of a similar experiment carried out on 32 bands of 512 × 512 section of the GER test image. The behavior here is somewhat different than before. In fact, while U-CBC guarantees a consistent gain w.r.t. conventional SPIHT, more than 1 db at 0.50 bit/sample, the KLT-based SPIHT performs even better, gaining about 0.5 dB at all rates. A possible explanation for this result lies in the nature of the test image considered (see Fig. 12a) which is much more homogeneous than both the multispectral images. Therefore the classified KLT guarantees only a limited gain w.r.t. a global KLT, not sufficient to offset the cost of side information, and the reduced efficiency of one-dimensional coding of coefficients, as opposed to the three-dimensional coding of SPIHT.

Like for the other images, we complete our analysis by showing, in Fig. 12, one band of the 10 bit/sample original image and its compressed counterpart, obtained with unsupervised CBC at 0.30 bit/sample, using a 16-class segmentation. Again, despite a compression ratio of more than 30:1, the subjective (as well as the objective) quality is so good that this can be considered as near-lossless coding.

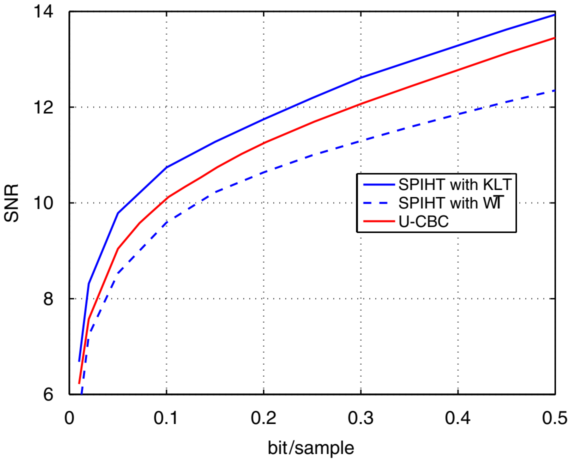

Fig. 11. Rate–distortion performance for the GER test image.

In all the preceding analyses, we used the SNR to compute an objective and synthetic measure of encoding quality. However, in the remote-sensing field, the quality of a compressed image is better measured by its value for subsequent automatic processing steps, like segmentation, classification, detection of particular targets, etc. Therefore, we conclude this section by presenting a sample segmentation experiment, carried out on our first test image, The Landsat TM section shown in Fig. 8, assesses the diagnostic value of images after compression.

We segmented the original image, with six classes, using an unsupervised minimum-distance clustering algorithm. Lacking actual references, this segmentation has been taken as our ground truth. Then,

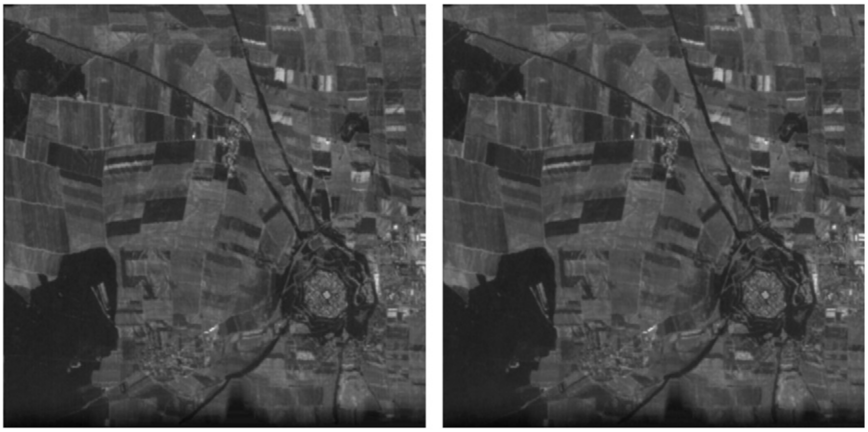

Fig. 12. Band 24 before and after encoding at 0.30 bit/sample.

Table 1
Confusion matrix with the CBC-compressed image

CBC	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Total	User's acc. (%)
Class 1	26,714			7				234	26,955	99.1
Class 2		3222	52						3274	98.4
Class 3		164	31,648		260				32,072	98.7
Class 4	36			40,560		3020	1141	1835	46,592	87.0
Class 5			1763		48,439	507			50,709	95.5
Class 6				445	2566	59,410			62,421	95.2
Class 7				893			14,522	456	15,871	91.5
Class 8	143			1663			598	21,846	24,250	90.1
Total	26,893	3386	33,463	43,568	51,265	62,937	16,261	24,371		
Prod.'s acc (%)	99.3	95.2	94.6	93.1	94.5	94.4	89.3	89.6		

Table 2
Confusion matrix with the SPIHT-compressed image

SPIHT	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Total	User's acc (%)
Class 1	25,970			104				881	26,955	96.3
Class 2		2886	388						3274	88.1
Class 3		333	28,098		3630	11			32,072	87.6
Class 4	137			35,844	20	6563	1262	2766	46,592	76.9
Class 5			3843	12	40,179	6675			50,709	79.2
Class 6			6	6351	7090	48,946	26	2	62,421	78.4
Class 7	2			1703		53	12,808	1305	15,871	80.7
Class 8	1250			2865		1	630	19,504	24,250	80.4
Total	27,359	3219	32,335	46,879	50,919	62,249	14,726	24,458	0	
Prod.'s acc (%)	94.9	89.7	86.9	76.5	78.9	78.6	87.0	79.7		

using the classes found in this first step, the image has been segmented again after being compressed at 0.40 bit/sample with both the CBC and the best reference algorithm, SPIHT with KLT. Finally, confusion matrices have been computed; they are reported in Tables 1 and 2 for the two algorithms.

Results speak definitely in favor of the proposed algorithm, with an overall error rate of about 6% for CBC as compared to about 18% for SPIHT (similar results have been obtained with all images at any rates). It is clear that this huge improvement is only partially due to the better reconstruction quality, considering that the SNR shows only a 2.5 dB improvement at that rate. The main reason, instead, is that CBC comprises a segmentation step which, although not explicitly accounted for in these experiments, affects the compressed data and leads eventually to a superior performance. A smarter approach would be to take *explicitly* into account the segmentation map built during the coding phase on the original uncompressed data, and this procedure would be highly recommended in case of practical operations, but we did not use it here since the comparison would have been unfair. It must be also pointed out that results would be less striking with other applications or segmentation techniques, but spectral clustering is probably the single most frequent processing step used in the analysis of such images.

5. Conclusions

This research had the goal to develop and test a low-complexity and fully unsupervised version of the CBC for multispectral images proposed in [5] so as to make it amenable to real-time use on-board a satellite. To this end, besides modifying a few blocks of the original coder, we had to include the on-line design and transmission of several new pieces of information. The judicious (and carefully tested) use of subsampling allowed us to limit the increase in complexity and hence obtain a definitely viable coder.

Experiments proved U-CBC to have a better rate–distortion performance than the original coder. This was partly a surprise, because more coding parameters must be now transmitted as side information, but the opportunity of fine tuning such parameters on the same image to be coded compensates abundantly this initial disadvantage. The U-CBC provided also better results than 3d-SPIHT, a state-of-the-art low-complexity coder, and (but for one case) than an improved version of 3d-SPIHT, based on spectral KLT.

The original CBC was already an interesting tool, especially because of its distinctive feature of providing an accurate segmentation map together with the class statistics embedded in the coded stream, pieces of information not easily available for the end user. The unsupervised version developed here adds the features of low complexity and improved rate–distortion performance, and makes it a valid candidate for actual use on-board a satellite.

References

[1] G.P. Abousleman, M.W. Marcellin, B.R. Hunt, Compression of hyperspectral imagery using the 3-D DCT and hybrid DPCM/DCT, IEEE Trans. Geosci. Remote Sensing (January 1995) 26–34.

[2] G.R. Canta, G. Poggi, Kronecker-product gain-shape vector quantization for multispectral and hyper-spectral image coding, IEEE Trans. Image Process. (May 1998) 668–678.

[3] P.L. Dragotti, G. Poggi, A.R.P. Ragozini, Compression of multispectral images by three-dimensional SPIHT algorithm, IEEE Trans. Geosci. Remote Sensing (January 2000) 416–428.

[4] Q. Du, C. Chang, Linear mixture analysis-based compression for hyperspectral image analysis, IEEE Trans. Geosci. Remote Sensing (April 2004) 875–891.

[5] G. Gelli, G. Poggi, Compression of multispectral images by spectral classification and transform coding, IEEE Trans. Image Process. (April 1999) 476–489.

[6] G. Gelli, G. Poggi, A.R.P. Ragozini, Multispectral-image compression based on tree-structured Markov random field segmentation and transform coding, in: Proceedings of the 1999 IEEE International Geoscience and Remote Sensing Symposium, vol. 2, June 1999, pp. 1167–1170.

[7] A. Gersho, R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, Dordrecht, 1992.

[8] G. Golub, C. van Loan, Matrix computations, third ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[9] S.E. Qian, Hyperspectral data compression using a fast vector quantization algorithm, IEEE Trans. Geosci. Remote Sensing (August 2004) 1791–1798.

[10] S.E. Qian, A.B. Hollinger, D. Williams, D. Manak, Vector quantization using spectral index-based multiple subcodebooks for hyperspectral data compression, IEEE Trans. Geosci. Remote Sensing (May 2000) 1183–1190.

[11] J.A. Saghri, A.G. Tescher, A.M. Planinac, KLT/JPEG 2000 multispectral bandwidth compression with region of interest prioritization capability, in: Proc. SPIE, 2003, pp. 226–235.

[12] J.A. Saghri, A.G. Tescher, J.T. Reagan, Practical transform coding of multispectral imagery, IEEE Signal Process. Mag. (January 1995) 32–43.

[13] A. Said, W.A. Pearlman, A new, fast, and efficient image codec based on set partitioning in hierarchical trees, IEEE Trans. Circuits Systems Video Technol. (1996) 243–250.

[14] S.S. Shen, J.H. Kasner, Effects of 3D wavelets and KLT-based JPEG-2000 hyperspectral compression on exploitation, in: Proc. SPIE, 2000, pp. 167–176.

[15] X. Tang, W.A. Pearlman, J.W. Modestino, Hyperspectral image compression using three-dimensional wavelet coding: a lossy-to-lossless solution, IEEE Trans. Geosci. Remote Sensing, submitted for publication. Available at ⟨http://www.cipr.rpi.edu/~pearlman/papers/tgrs04_tpm.pdf⟩.

[16] ⟨http://www.kakadusoftware.com/⟩.

[17] ⟨http://www.globalsecurity.org/space/systems/nemo.htm⟩.

[18] ⟨http://www.cipr.rpi.edu/research/SPIHT/⟩.