Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2007, Article ID 78323, 13 pages
doi:10.1155/2007/78323

Research Article

Costs and Advantages of Object-Based Image Coding with Shape-Adaptive Wavelet Transform

Marco Cagnazzo, Sara Parrilli, Giovanni Poggi, and Luisa Verdoliva

Dipartimento di Ingegneria Elettronica e delle Telecomunicazioni, Università Federico II di Napoli, Via Claudio 21, 80125 Napoli, Italy

Received 19 August 2006; Revised 27 November 2006; Accepted 5 January 2007

Recommended by Béatrice Pesquet-Popescu

Object-based image coding is drawing a great attention for the many opportunities it offers to high-level applications. In terms of rate-distortion performance, however, its value is still uncertain, because the gains provided by an accurate image segmentation are balanced by the inefficiency of coding objects of arbitrary shape, with losses that depend on both the coding scheme and the object geometry. This work aims at measuring rate-distortion costs and gains for a wavelet-based shape-adaptive encoder similar to the shape-adaptive texture coder adopted in MPEG-4. The analysis of the rate-distortion curves obtained in several experiments provides insight about what performance gains and losses can be expected in various operative conditions and shows the potential of such an approach for image coding.

1. INTRODUCTION

Object-based image coding is an increasingly active area of research, dating back to early works on second generation coding techniques [1] and gaining momentum more recently thanks to the driving force of the MPEG-4 video coding standard [2]. The major conceptual reason for object-based coding is that images are *naturally* composed by objects, and the usual pixel-level description is only due to the lack of a suitable language to efficiently represent them. Once objects have been identified and described, they can be treated individually for the most diverse needs. For example they can be assigned different coding resources and different error-protection levels based on their relative importance for the user [3, 4], can be edited in various ways by high-level applications, or can be used for subsequent classification tasks (e.g., biometric applications).

In some instances, object-based coding is obviously the most reasonable solution. In the context of MPEG-4 video coding, for example, when a number of arbitrarily shaped foreground objects move in front of a fixed background, which is a full-frame sprite, conventional coding is clearly inefficient. Additionally, there exist applications (e.g., [5]) in which data are available only for part of the image frame, and one has no choice but to either code an arbitrarily-shaped object or artificially pad the object out to a full-frame. Similar to object-based coding, but at a lower level of abstraction, is region-based coding, where the focus is not on objects, meant as semantic units, but rather on image regions, defined by their statistical properties. Statistically homogeneous regions can be singled out by pixel-level segmentation techniques with the aim to encode them efficiently, or the user himself can identify a region of interest (ROI) to encode it at higher priority or with different techniques than the background, as envisaged in several applications and standards [6–9].

Of course, before resorting to object-based coding, and to a particular suite of algorithms, one should be well aware of its potential advantages and costs. In terms of coding efficiency, the object-based description of an image presents some peculiar costs which do not appear in conventional coding. First of all, since objects are separate entities, their shape and position must be described by means of some segmentation map, sent in advance as side information. In addition, most coding techniques become less efficient when dealing with regions of arbitrary size and shape. Finally, each object needs its own set of coding parameters, which adds to the side information cost. On the positive side, an accurate segmentation carries with it information on the graphical part of the image, the edges, and hence contributes to

the coding efficiency and perceived quality. Moreover, component regions turn out to be more homogeneous, and their individual encoding can lead to actual rate-distortion gains.

In any case, to limit the additional costs, or even obtain some performance improvement, it is necessary to select appropriate coding tools, and to know in advance their behavior under different circumstances.

In this work, we focus on a wavelet-based shape-adaptive coding algorithm. The main coding tools are the shape-adaptive wavelet transform (SA-WT) proposed by S. Li and W. Li [10], and a shape-adaptive version of SPIHT (SA-SPIHT) [11] (similar to that formerly proposed in [12] and further refined in [13]) which extends to objects of arbitrary shape the well-known image coder proposed by Said and Pearlman [14]. The attention on wavelet-based coding is justified by the enormous success of this approach in conventional image coding [15, 16], leading to the new wavelet-based standard JPEG-2000 [7], and more recently video coding [17]. As for the choice of the specific coding scheme, S. Li and W. Li's SA-WT is by now a de facto standard, and SPIHT guarantees a very good performance, and is widespread and well known in the compression community. In addition, the algorithm analyzed here is very similar to the standard texture coder of MPEG-4 [2]. Of course, this is not the only reasonable choice, and other coding algorithms based on shape-adaptive wavelet have been proposed in recent years [18–22], sometimes with very interesting results, but a comparison with some of these algorithms, deferred to the last section, is of marginal interest here. The main focus of this work is to analyze the quite general mechanisms that influence the efficiency of wavelet-based shape-adaptive coding and to assess the difference in performance with respect to conventional wavelet-based coding.

In more detail, we can identify three causes for the additional costs of object-based coding: the reduced energy compaction of the WT and the reduced coding efficiency of SPIHT that arise in the presence of regions with arbitrary shape and size, and the cost of side information (segmentation map, object coding parameters). Note that this classification is somewhat arbitrary, since the reduced energy compaction of WT does influence the efficiency of SPIHT, nonetheless it will help us in our analysis. As for the possible gains, they mirror the losses, since they arise for the increased energy compaction of the WT, when dominant edges are removed, and for the increased coding efficiency of SPIHT when homogeneous regions have to be coded.

A theoretical analysis of such phenomena is out of the question, and in the literature attempts have been made only for very simple cases, like 1D piecewise-constant signals [23]. Therefore, we measure losses and gains by means of numerical experiments carried out in controlled conditions. This allows us to isolate with good reliability the individual contributions to the overall performance, point out weaknesses and strengths of this approach, and hence give insight about the behavior of the proposed coding scheme in situations of practical interest.

In order to assess losses and gains related to the SA-WT only, we remove the cost of side information, and use an "oracle" coder which mimics the progressive bit-plane coding of SPIHT but knows in advance the location of significant coefficients within each bit-plane, thereby removing all sorting-pass costs.[1] Within this framework, we use several classes of images and of segmentation maps, both synthetic and natural, so as to study all the relevant phenomena. Subsequently, for the same set of images and maps, we add the actual coding phase: the additional gains and losses can be therefore attributed to SA-SPIHT or to its interactions with the SA-WT.

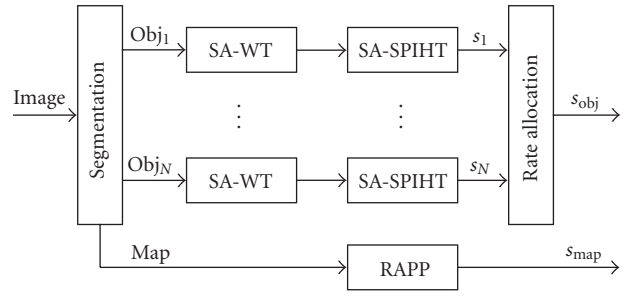


Figure 1: The object-based coding scheme under investigation.

The manuscript is organized as follows. In Section 2 some more detail on the coding scheme is provided. In Sections 3 and 4 we analyze losses and, respectively, gains of object-based coding by means of numerical experiments on suitable images and segmentation maps. Section 5 presents some results for a real-world image with its own segmentation maps, and Section 6 compares performance with those of other coding schemes described in the literature. Finally, Section 7 draws conclusions.

2. THE CODING SCHEME

We implemented an object-based coding scheme with the following elementary steps (see Figure 1):

(1) image segmentation;
(2) lossless coding of the segmentation map (object shapes);
(3) shape-adaptive wavelet transform of each object;
(4) shape-adaptive SPIHT coding of each object;
(5) optimal post-coding rate allocation among objects.

The accurate segmentation of the image is of central importance for the success of object-based coding, and is by itself a very challenging task and a "hot" topic. However, faithful image segmentation is not of interest here and is not investigated. Moreover, to study the effects of different object geometries on the coding performance, we need to change rather freely the geometrical/statistical parameters of objects, and therefore resort, in most of the analysis, to artificial

[1] Note that the very same oracle coder works for all bit-plane oriented coders that use S. Li and W. Li's SA-WT, like for example [19, 22].

regular segmentation maps, independent of the actual image content. Only in our final experiments we do consider meaningful segmentation maps.

The segmentation maps are encoded without loss of information, because of their importance, by means of a modified version of the RAPP algorithm [24], originally proposed for palette images, which proves very efficient for this task. The cost for coding the map, as well as all other side information costs, can become significant and even dominant in some instances, and hence must be always taken into account in the overall performance.

As for the SA-WT, we resort to S. Li and W. Li's algorithm, as already said, which is almost universally used in the literature and also adopted in the MPEG-4 standard. For a detailed description we refer to the original paper [10], but it is worth recalling here its most relevant features. First of all, the number of coefficients equals the number of pixels in the original object, so there is no new redundancy introduced. Second, spatial relationships among pixels are retained, so there are no new spurious "frequencies" in the transform. Finally, the SA-WT falls back to ordinary WT for rectangular objects. All these reasons, together with its simple implementation and experimentally good performance, justify the success of this algorithm. In the implementation, we use five levels of decomposition, Daubechies 9/7 biorthogonal filters, and the global subsampling option which secures experimentally the best performance.

After SA-WT, we use the well-known SPIHT algorithm, in the shape-adaptive extension proposed in [11]. Again, we refer the reader to the original paper [14] for a description of SPIHT, but it is worth recalling that it is a bit-plane coder of the wavelet coefficients. For each bit-plane there are essentially two tasks, locating the significant bits, and specifying their value (also the coefficient signs must be encoded of course). Other algorithms of interest here share the same general approach, and differ only in the way significant bits are located. Our shape-adaptive version of SPIHT is very similar to the basic algorithm with the differences that only active nodes, that is nodes belonging to the support of the SA-WT transform, are considered, and that the tree of coefficients has a single ancestor in the lowest frequency band.

After coding, the rate-distortion (RD) curves of all objects are analyzed so as to optimally allocate bits among them for any desired encoding rate, like in the post-compression rate allocation algorithm of JPEG-2000. This process is intrinsically performed in conventional coding, while it is a necessary step in object-based coding, and also an extra degree of freedom as bits could be also allocated according to criteria different from RD optimization.

3. MEASUREMENT OF LOSSES

3.1. Methodology

The performance of a transform-based compression algorithm depends essentially on the efficiency of the transform, which is therefore the first item we must quantify.

In the context of data compression, the goal of a transform is to compact as much signal energy as possible in a small number of transform coefficients. After a suitable bit allocation, this translates in an SNR (signal-to-noise ratio) improvement which, for a Gaussian signal, an isometric transform, and in the high-resolution limit, is equal to the coding gain (CG) [25, 26], defined as $10\log_{10}\sigma^2_{\mathrm{AM}}/\sigma^2_{\mathrm{GM}}$, that is, the ratio (in dB) between the arithmetic and geometric means of the transform coefficients, or transform subbands in the wavelet case. Although the above-mentioned conditions are rarely met in practice, the CG provides a good insight about the actual gain provided by the transform. In addition, it can be easily extended [27] to encompass nonisometric transforms, such as that based on the biorthogonal Daubechies filters. Unfortunately, in the case of shape-adaptive WT, such a measure is not meaningful at all, because the transform is nonisometric in an unpredictable way. This depends on the need to transform signal segments composed by a single pixel: in S. Li and W. Li's algorithm, this generates a single coefficient which is put in the low-pass transform band and, in order not to introduce discontinuities in otherwise flat areas, is multiplied by a constant. This multiplication (which can occur many times in the SA-WT of an object) modifies the transform energy and makes the coding-gain measure all but useless.

For this reason, we propose here an alternative methodology[2] to compare the efficiency of SA-WT and its conventional (or "flat") version. The basic idea is to apply both the shape-adaptive and the flat transforms to the same image, quantize the resulting coefficients in the same way, and compare the resulting RD curves. In order for the comparison to be meaningful, the transforms must operate on exactly the same source, and hence *all* objects of the image must undergo the SA-WT and be processed together. The total number of coefficients produced by the SA-WT is equal to the number of image pixels and hence to the number of WT coefficients. These two sets of coefficients (which cannot be directly compared because of the energy mismatch) are sent to an oracle encoder which implements a bit-plane quantization scheme like that of SPIHT and most other engines used in object-based coders. All these algorithms spend some coding bits to locate the significant coefficients in each plane (sorting pass, in SPIHT terminology), and some others to encode their sign and to progressively quantize them (refinement pass). Our oracle coder knows in advance all significance maps and spends its bits only for the sign and the progressive quantization of coefficients. As a consequence, the rate-distortion performance of this virtual coder depends only on how well the transform capture pixel dependencies, what we call transform efficiency.

As an example, consider the RD curves of Figure 2. Although the object-based coder (solid red) performs clearly worse than the flat coder (solid blue), at least at low rates, their oracle counterparts (dashed red and dashed blue) perform nearly equally well. This means that, as far as the

[2] Preliminary results have been presented in [28].

transforms are concerned, the shape-adaptive WT is almost as efficient as the conventional WT, and therefore the losses must be ascribed to coding inefficiencies or to the side information. Actually, since the cost of side information is known, we can also easily compute the losses caused by SA-SPIHT inefficiencies, the second major item we are interested to measure.

There are two reasons why shape-adaptive SPIHT could be less efficient than flat SPIHT:

(i) the presence of incomplete trees of coefficients;
(ii) the interactions with the SA-WT.

Much of the efficiency of SPIHT, especially at low-rates, is due to the use of zerotrees, that is, trees of coefficients that are all insignificant with respect to a given threshold and can be temporarily discarded from further analysis. A single information bit can therefore describe a whole zerotree, comprising a large number of coefficients. With an arbitrarily shaped object, the support of the transform can be quite irregular, and incomplete zerotrees can appear, which lack some branches and comprise less coefficients than before. As a consequence, the zerotree coding process becomes less efficient, at least at the lowest rates.

The second item concerns a more subtle phenomenon, the fact that the reduced WT energy compaction affects indeed *both* quantization and sorting. In fact, when the WT does not compact efficiently, the energy is more scattered throughout the trees and more bits are spent sorting in order to isolate the significant coefficients at each iteration. Hence, computing these losses as due to SA-SPIHT is somewhat arbitrary, but it is also true that a different coder could be less affected by this phenomenon.

3.2. Experimental results

To measure losses, we encode some natural images of the USC database [29] with both the oracle and the actual object-based coders using synthetic segmentation maps of various types formed by square tiles, rectangular tiles, wavy tiles, irregular tiles. Test images (512 × 512 pixels, 8 bit/pixel) are shown in Figure 3, while Figure 4 shows some examples of segmentation maps. By using such synthetic maps, which are not related to the actual image to be coded, we introduce and measure only the *losses* due to object shape and size, while no gain can be expected because object boundaries do not coincide with actual region boundaries.

In the first experiment we segment the natural images in square tiles of size going from 512×512 (whole image) down to 32×32 (256 objects), and encode them as described before. In Figure 5 we report the rate-distortion curves obtained by the object-based coders for each tile size: solid lines refer to the actual coder, and dashed lines to the oracle coder. Note that the flat case corresponds to the 512 × 512 coder, that is, conventional WT and SPIHT. Curves refer to the image Lena of Figure 3(a), as will always be in the following unless otherwise stated, but similar results have been obtained with all other images. A first important observation is that the quantization rate is always a small fraction, about one fourth, of the total rate, at least in the range considered here.[3] As a consequence, the same relative loss of efficiency is much more critical for SPIHT than for the WT. In this experiment, however, losses are always quite limited. Performances worsen as the tile size decreases, but the rate increment is always less than 20% (except a very low rates) and the PSNR gap is less than half dB at high rates, and about 1 dB at lower rates. Most of these losses are due, directly or indirectly, to the reduced compaction ability of the SA-WT, since the zerotrees are always complete, and the fixed cost of side information, 0.013 bit/pixel in the worst case, is quite small. Note, however, that this last cost cannot be neglected if one looks at very low rates.

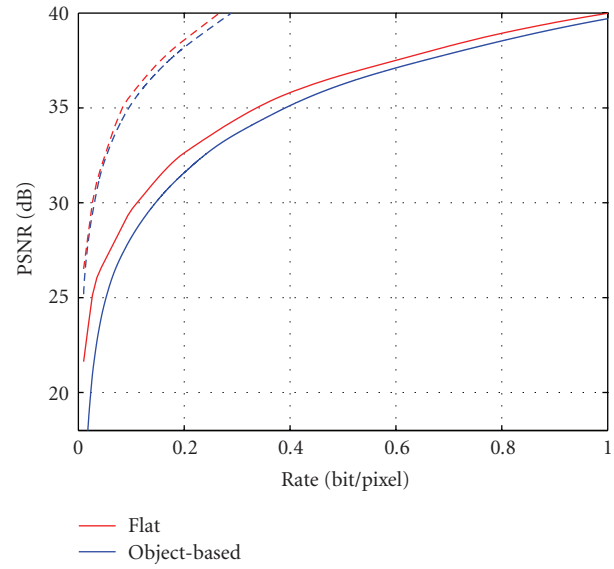


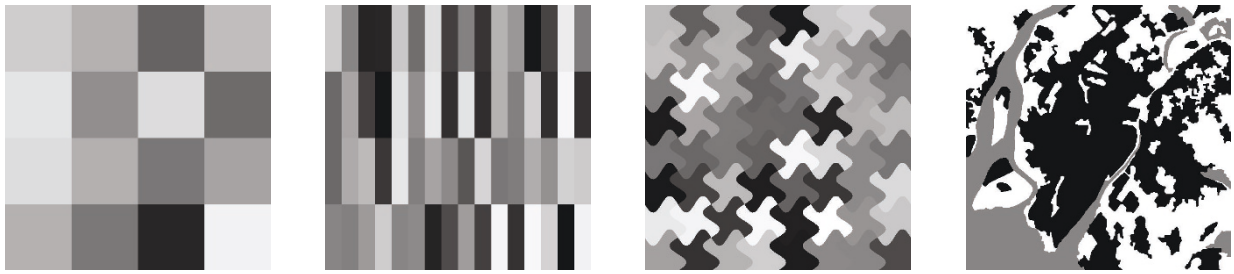
Figure 2: RD curves for flat (red) and object-based (blue) coders. Solid and dashed lines are, respectively, for actual and oracle coders.

To begin investigating the influence of region shapes, in the second experiment we consider rectangular tiles of fixed size (4096 pixels) but different aspect ratios, from 64 × 64 to 512×8. The RD curves are reported in Figure 6, together with those for the flat case, and show that the aspect ratio does matter, but only when very short segments are considered. Indeed, the performance is very close for 64 × 64, 128 × 32, and even 256 × 16 tiles, while it becomes significantly worse for 512 × 8 tiles, because the WT cannot compact much energy anymore with segments as short as 8 pixels. For example, the PSNR loss at high rate is 1.15 dB for the 512 × 8 case and less than 0.6 dB for all the other cases. One might suspect that the sharp decline in performance in the 512 × 8 case is also related with our use of 5 levels of decomposition when 3 or 4 would have been more appropriate for such short segments. In fact, this mismatch produces several single coefficients, after some levels of WT, which are further filtered

[3] At higher rates, the RD slope is the same in all cases because we are only coding noise-like residuals, and hence the analysis looses interest.

(a) (b) (c) (d)

Figure 3: Test images from the USC database: (a) Lena, (b) peppers, (c) baboon, (d) house.



(a) (b) (c) (d)

Figure 4: Some maps used in the experiments: (a) square 128×128 tiles, (b) rectangular 128×32 tiles, (c) wavy tiles with C = 1, A = 16, (d) out-of-context map.

and lead to an artificial increase in energy. However, all our experiments show that adapting the number of decomposition levels to the object size has no measurable effects on the performance, and that a fixed 5-level SA-WT is the optimal choice, at least for our 512×512 images.

Let us now consider more complex tiles, obtained by remodeling the boundaries of a 64×64 square as sine-waves with amplitude A pixels, and frequency C cycles/tile. One such segmentation map, obtained for A = 16 and C = 1, is shown in Figure 4(c). In Figure 7, we report the RD curves for some significant values of A and C, together with the reference curves for square 64×64 tiles and for flat coding. As expected, the performance worsens as the tiles become less regular. At high rates the impairment is not dramatic, with a PSNR loss that lies between 1 and 2 dB, while the situation is much worse at low rates, with losses of 4-5 dB or, for a given PSNR, a coding rate that doubles with respect to flat coding. Apparently, such losses are mainly due to the side information and SA-SPIHT inefficiencies, and only in minimal part to the SA-WT, since the RD curves for the oracle coder are all very close, but we should not forget the WT-SPIHT interactions, and will soon come back to this topic.

In our fourth experiment, we use segmentation maps obtained for unrelated (remote-sensing) images of the same size as ours. These maps, one of which is shown in Figure 4(d), present many elementary tiles, with quite different size and shape, some with regular boundaries and some not. Figure 8 shows RD curves for this case, which resemble closely those of Figure 7, and for which the same comments apply, suggesting that the wavy-tiles segmentation can be a good tool to mimic actual segmentation maps.

To take a closer look at these results, let us consider Table 1 where we have collected the individual contributions of side information, quantization, and sorting pass to the overall coding cost, at a PSNR of 30 dB, corresponding to the low-rate range. We see that the increase of the quantization cost with respect to the flat case is quite steep, from 15% up to 100%, due to the reduced compaction ability of the transform. As for the sorting cost, it also increases with respect to the flat case. The increase is obviously larger in the last six cases, when the tile geometry is more challenging, but also nonnegligible in the first six cases, with square and rectangular tiles. This is quite telling, because with straight boundaries there are no incomplete trees to impair performance, and hence all losses must be charged to the reduced energy compaction. Therefore, one can even hypothesize that transform inefficiencies are the ultimate cause of most of the overall losses, even though the effects are more evident in the sorting pass, a conjecture that we will further analyze shortly. As a synthetic measure of performance, we reported in the last column the overall rate increase with respect to flat coding, including all contributions, which is quite large in all realistic cases, confirming that object-based coding can be very penalizing at low rates.

The picture, however, is quite different at high rates. Table 2 is similar to Table 1 except that all costs are computed

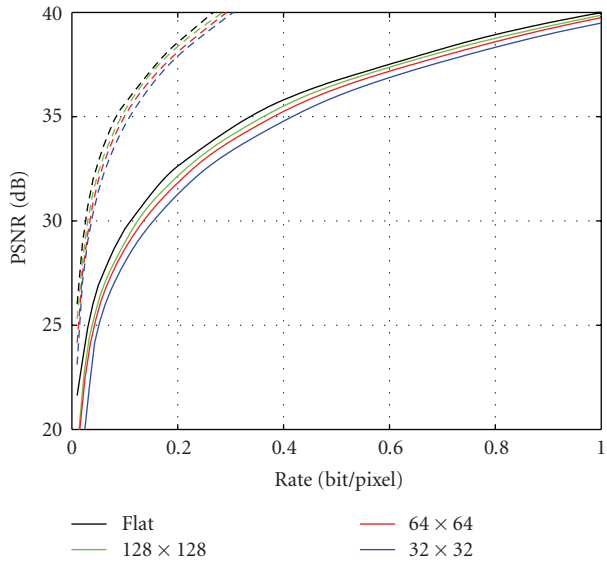


Figure 5: RD performance with square-tile segmentation. Solid and dashed lines are, respectively, for actual and oracle coders. Black lines are for flat (conventional) coding of the whole image, colored lines are for object-based coding.

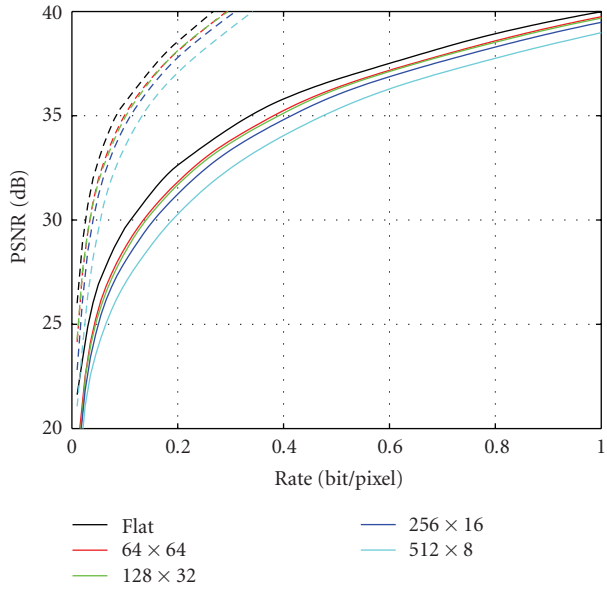


Figure 6: RD performance with rectangular-tile segmentation.

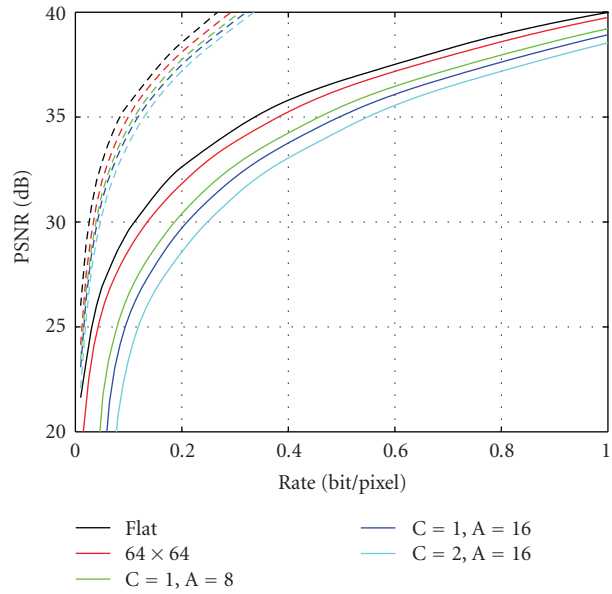


Figure 7: RD performance with wavy-tile segmentation.

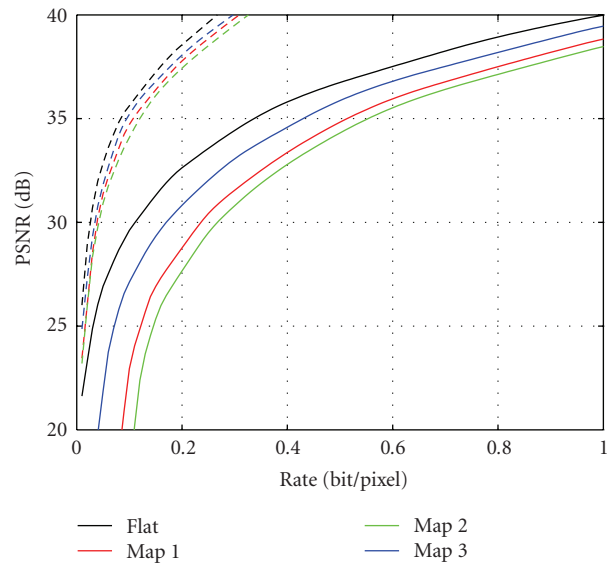


Figure 8: RD performance with out-of-context segmentation maps.

at a PSNR of 38 dB, hence at the right end of our range. It is obvious that the cost of side information becomes less relevant, and even in the more challenging situations the cost of quantization and sorting presents only a limited increase. In the last column, we report a more familiar measure of performance, the PSNR loss with respect to flat coding at 0.8 bit/pixel, which is never more than 2 dB, and quite often under just 1 dB showing that, at high rates, object-based coding can be used without paying much attention to the rate-distortion performance. It is also worth remembering that, in most practical situations where object-based coding is used, there is only a small number of objects, and therefore these measures of loss can be assumed as upper bounds.

We conclude this section with one last insightful experiment, which sheds some more light on the nature of SPIHT losses. S. Li and W. Li's SA-WT, when applied to all objects of an image, like the simple example of Figure 9(a), produces transforms that do not fit together, namely, cannot be put

Table 1: Indicators of losses at low rates (PSNR = 30 dB).

	Absolute rates			Percent increase		
Tiling	Side.i.	Quant.	Sorting	Quant.	Sorting	Total
Whole image	—	0.026	0.085	—	—	—
128 × 128	0.003	0.030	0.091	15.4	7.3	11.7
64 × 64	0.005	0.034	0.096	30.9	13.1	21.6
32 × 32	0.013	0.037	0.104	42.9	22.0	38.7
128 × 32	0.005	0.034	0.100	31.2	17.8	25.2
256 × 16	0.005	0.040	0.110	53.5	29.3	39.6
512 × 8	0.005	0.054	0.131	106.9	54.0	71.1
C = 1, A = 8	0.032	0.038	0.116	48.4	36.3	67.5
C = 1, A = 16	0.044	0.041	0.125	58.6	46.7	89.1
C = 2, A = 16	0.060	0.047	0.141	80.6	65.8	123.4
Map 1	0.083	0.038	0.127	48.3	49.9	123.4
Map 2	0.105	0.042	0.135	61.2	59.2	154.0
Map 3	0.042	0.034	0.105	33.0	24.0	63.0

Table 2: Indicators of losses at high rates (PSNR = 38 dB).

	Absolute rates			Percent increase		Δ PSNR
Tiling	Side.i.	Quant.	Sorting	Quant.	Sorting	@ 0.8 b/p
Whole image	—	0.176	0.488	—	—	—
128 × 128	0.003	0.184	0.498	4.2	2.0	0.15
64 × 64	0.005	0.195	0.512	10.6	4.9	0.31
32 × 32	0.013	0.204	0.534	15.5	9.4	0.62
128 × 32	0.005	0.194	0.519	10.2	6.3	0.37
256 × 16	0.005	0.209	0.542	18.2	11.0	0.60
512 × 8	0.005	0.241	0.590	36.4	20.9	1.14
C = 1, A = 8	0.032	0.211	0.563	19.3	15.2	0.95
C = 1, A = 16	0.044	0.221	0.589	25.2	20.6	1.35
C = 2, A = 16	0.060	0.234	0.622	32.6	27.3	1.82
Map 1	0.083	0.209	0.591	18.5	21.1	1.33
Map 2	0.105	0.225	0.611	27.5	25.2	1.89
Map 3	0.042	0.197	0.544	11.7	11.3	0.78

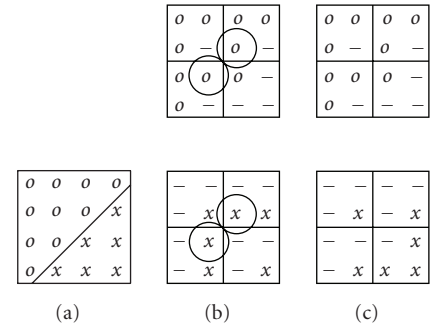


Figure 9: Object overlapping in the transform domain. The 4×4 original image with two objects (a) is subject to 1 level of SA-WT: the supports of the two objects overlap with S. Li and W. Li SA-WT (b) but not with the fitting SA-WT (c).

together in a single image as the pieces of a mosaic, because some coefficients overlap, as the circled coefficients shown in Figure 9(b). This is unavoidable if all single coefficients must be put in the low-pass band after filtering. However, we can modify the algorithm and put single coefficients either in the low-pass or high-pass band depending on their coordinates. This way, we might sacrifice part of the SA-WT efficiency, but obtain object transforms that fit together as shown in Figure 9(c). After all the SA-WTs have been carried out, we can encode the coefficients by using SA-SPIHT on each object, or conventional SPIHT on all the coefficients arranged as a single image. The flat and object-based coders thus operate exactly on the same set of coefficients, and all possible impairments can be ascribed to SA-SPIHT coding inefficiencies. The RD curves obtained with flat and SA-SPIHT for various segmentation maps are reported in Figure 10, and show clearly that the efficiency gap between shape-adaptive and flat SPIHT is always very limited, and at high rates never exceeds 0.3 dB.[4] This seems to be a conclusive proof that the losses arising in the sorting pass, although dominant with respect to those of the quantization pass, are mostly related to the reduced compaction ability of the SA-WT.

4. MEASUREMENT OF GAINS

4.1. Methodology

The rate-distortion potential of object-based coding strongly depends on the ability of the segmenter to single out accurately the component objects. When this happens, in fact, the segmentation map describes automatically many expensive high-frequency components, related to the edges between different objects. In terms of SA-WT, this means dealing with a signal (within the object) that is much smoother that the original signal, since strong edges have been removed, which leads in turn to a much increased efficiency because most of the encoding resources, especially at low rates, are normally used for describing edges. Of course, the actual success of this approach depends on many factors, such as the profile of edges, the statistical properties of the signal within the objects, and the accuracy of segmentation.

In order to measure the potential performance gains, we get rid of the dependence on the segmentation algorithm, which is not the object of this analysis, by building some mosaics in which neighboring tiles are extracted from different images. Of course, one must keep in mind that this condition is very favorable for object-based coding since objects are clear-cut and we know their shape perfectly. Our mosaics vary not only for the form of the tiles, but also for the source images from which they are drawn, that can be

(i) synthetic images where the signal is polynomial in the spatial variables;

[4] As an aside, our experiments show also that the performance of this new scheme (fitting SA-WT + flat SPIHT) is very close to that of our object-based algorithm. However, this new scheme is not object-based anymore.

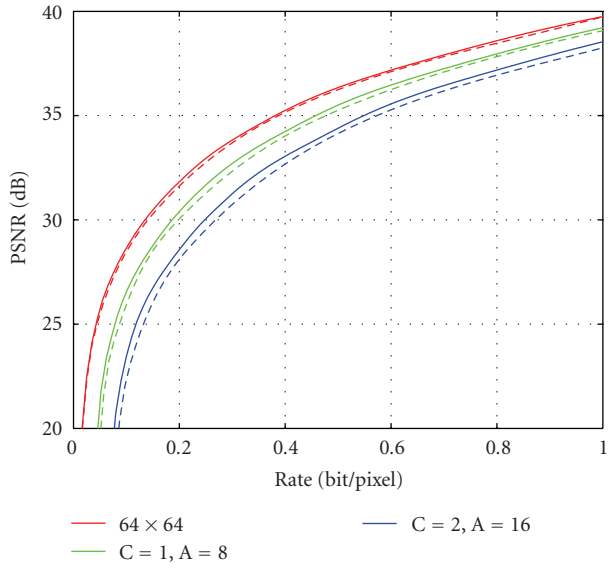


FIGURE 10: RD performance with fitting SA-WT. Solid lines are for flat coding of the mosaic formed by the object transform, dashed lines are for actual object-based coding.

(ii) natural images from the USC database;
(iii) natural textures from the Brodatz database, also available at [29].

Some examples are shown in Figure 11. By changing the source images we go from the most favorable case, like that of Figure 11(a), where all tiles are from polynomial images, to the most challenging, like that of Figure 11(d), where even within the tiles there are strong signal components at the medium and high frequencies due to the original textures. In between these extremes, there are more realistic cases where the objects are drawn from natural images predominantly smooth, like Figure 11(b), or with significant texture components, like Figure 11(c).

4.2. Experimental results

Figure 12 shows the PSNR differences between the object-based and the flat coders when mosaics are composed by wavy tiles of size 64×64 and boundary parameters $C = 1$ and $A = 16$ with the same source images as those shown in Figure 11. For the first mosaic, there is a very large gain of 8–10 dB at medium-high rates, and up to 20 dB at low rates (out of the scale of our figure). This is remarkable but not really surprising, given the smooth sources and the fact that Daubechies wavelets are perfectly fit for polynomial signals.

More interesting are the results obtained with the natural mosaics, with a gain at all bit-rates of about 5 dB in the first case, and almost 2 dB in the second case. Considering that these are natural images, this speaks strongly in favor of the potential of object-based coding, even with all the *caveat* due to the favorable experimental conditions. Also, remember that the observed gain is obtained despite the losses due to the use of SA-WT with small wavy tiles (see again Figure 7). As expected, results are less favorable for the fourth mosaic, where the presence of many high-frequency components within the tiles reduces the gain to the point that it compensates the shape loss but little more.

Figure 13 shows results obtained with the same source images but with square 128×128 tiles. The general behavior is very similar to the former case, but all gains are now much smaller because of the reduced number of objects and the straight boundaries, and even with the polynomial mosaic there is only a 2 dB gain at high rates.

5. PERFORMANCE WITH REAL-WORLD IMAGES

In order to isolate and analyze in depth the phenomena of interest, the experiments carried out in the preceding sections dealt with ideal and sometimes limiting cases. Now, we focus on the performance of the whole coding scheme in real-world situations, thus including the image segmentation, with all its inaccuracies.

In these experiments, we consider the image peppers of Figure 3(c) because its segmentation in a reasonably small number of meaningful objects is somewhat simpler. As a side effect, some objects comprise just one or a few smooth and coherent surfaces, which makes peppers a more favorable case with respect to other, more complex, images. In any case, the choice of what represents an object is somewhat arbitrary, and therefore we use several segmentation maps, with a different number of objects, shown in Figure 14 from the most detailed (25 objects) to the simplest one (just 4 objects, including the background).

Our object-based coding scheme provides the RD curves shown in Figure 15 together with the curve for the flat coder. Results might seem a bit disappointing at first, since the flat coder is always the best, but this is easily justified. In fact, even neglecting the unavoidable segmentation inaccuracies, it must be considered that, with ordinary images, the object boundaries are rarely clear-cut, due to the combination of the object 3D geometry and the illumination, and also to the limited resolution of the sensors that causes some edge smearing. Of course, this erodes the gains of removing strong edges. In addition, when objects have a semantic meaning, their interior is typically not uniform (just think of the bright glares within each pepper), and therefore the WT does not benefit much from the segmentation. On the other hand, when the segmentation map becomes very accurate, so as to single out regions that are actually uniform, the cost of side information increases significantly. In this light, the object-based RD curves of Figure 15 can be considered reasonably good, with a loss of no more than half dB at medium-high rates, and somewhat more at the lower rates, when the cost of side information is proportionally more relevant.

It is also interesting to consider the visual quality of compressed images, and to this end, in Figure 16 we show the image peppers compressed at 0.05 bit/pixel with WT/SPIHT (Figure 16(a)) and with our object-based coder using the simple segmentation map of Figure 14(b) (Figure 16(b)). Such a low rate was selected in order to emphasize the differences of the two approaches, which at higher rates tend

(a) (b) (c) (d)

FIGURE 11: Some mosaics used in the experiments, with square 128×128 tiles: (a) polynomials, (b) house + peppers, (c) Lena + baboon, (d) textures.

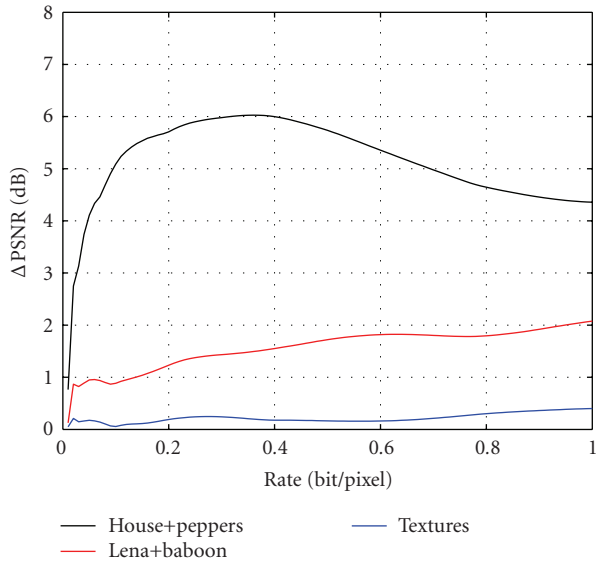


FIGURE 12: PSNR gain of OB-coding with respect to flat coding for wavy-tile mosaics.

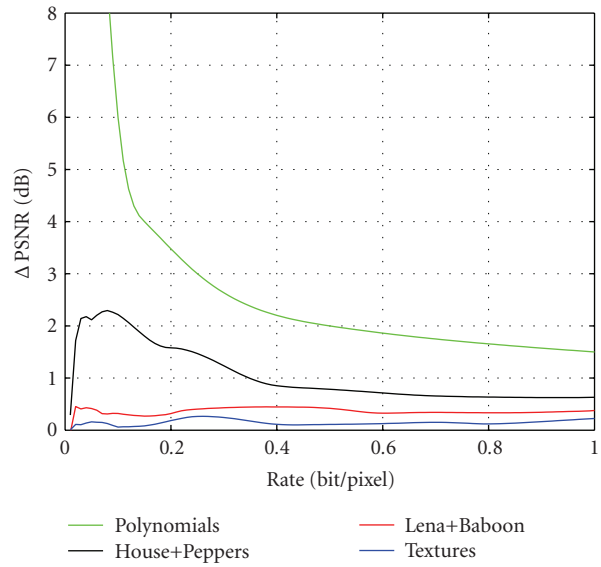


FIGURE 13: PSNR gain of OB-coding with respect to flat coding for square-tile mosaics.

to disappear. The first image has a better PSNR (26.3 versus 25.2 dB), but the second one has a superior perceptual quality, at a first look, because major edges have been better preserved. At a closer inspection, however, the object-based coded image presents a slightly worse texture quality, due to the lower effective rate available, and especially some annoying artifacts at the diagonal boundaries, which appear unnaturally rugged. This last problem could be easily overcome by some directional filtering. Needless to say, if one concentrates most coding resources on a single object considered of interest, neglecting the background, the object-based approach shows an overwhelming superiority.

To conclude this section, let us consider an example of compression of multispectral images, where the segmentation produces regions with nearly uniform statistics, the cost of the segmentation map is shared among many bands, and hence the conditions are such that object-based coding can actually provide some rate-distortion gains. We use a 6-band 512×512-pixel Landsat TM multispectral image of a region near Lisbon, one band of which is shown in Figure 17(a), while Figure 17(b) shows the segmentation map used in this experiment. Figure 18 compares the rate-distortion performance of the best flat and best object-based technique (see [30] for more details). After recovering from the initial handicap due to side information, the object-based technique provides a small but consistent performance gain over the flat technique.

6. COMPARISON WITH OTHER OBJECT-BASED WAVELET CODERS

The object-based coder we have analyzed uses what are probably the most well-known and widespread tools in this field, but other object-based coders have been proposed recently, and it is therefore interesting to carry out a performance comparison. We therefore repeated the experiments of Figure 15 using various algorithms: WDR [18], TARP [21], OB-SPECK [19], and BISK [22], implemented in the

(a) (b) (c) (d)

Figure 14: Segmentation maps for image peppers with (a) 25, (b) 16, (c) 8, and (d) 4 objects.

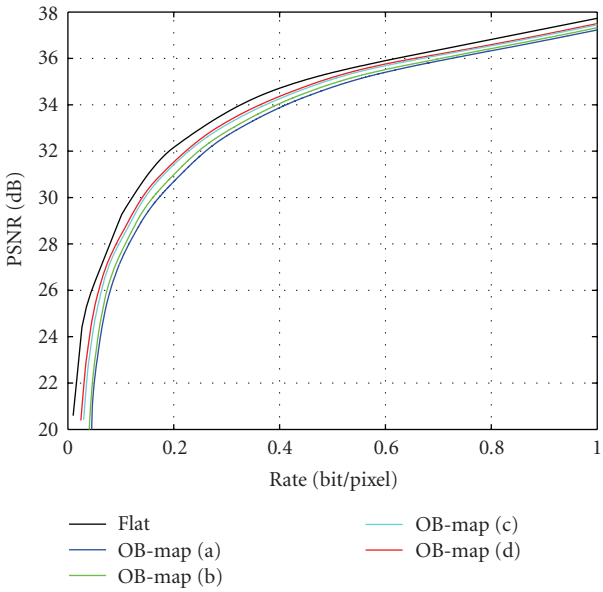


Figure 15: RD performance of flat and object-based codings for image peppers.

Qcc library [31] freely available at [32]. All these algorithms are based on an SA-WT [5] very similar to S. Li and W. Li's SA-WT, and encode the coefficients by means of embedded bit-plane coding algorithms.

The best performance is exhibited by BISK, based on the shape-adaptive version of SPECK, from which it differs for two main innovations: the use of a more flexible binary rather than quaternary splitting of blocks, and the introduction of a bounding box to help discard nodes outside the object of interest. BISK proves also superior to SA-SPIHT, as appears from the curves of Figure 19, obtained with the map of Figure 14(d). The gap, however, is partially due to BISK use of arithmetic coding for the output stream. When we introduce a similar coding step after SPIHT the difference becomes very limited, Figure 20. This had to be expected, if losses are mostly related, directly or indirectly, to the compaction ability of the SA-WT, and this is the same for the two coders.

7. CONCLUSIONS

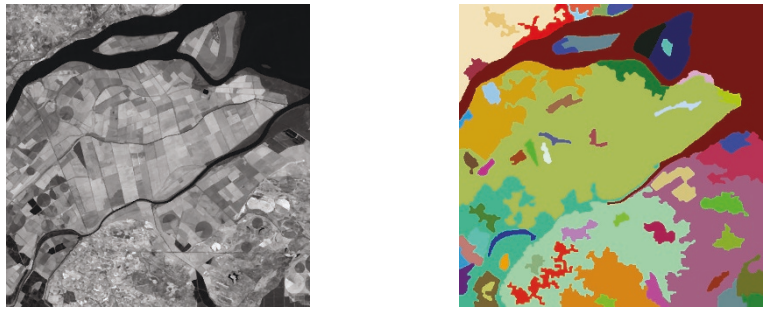
Wavelet transform is a de facto standard in image coding, and SPIHT is one of the most efficient, simple, and flexible algorithms for the encoding of wavelet coefficients. It is therefore only natural to consider their shape-adaptive versions to address the problem of object-based image coding, and to wonder how efficient they are when used for this new task.

Our aim was to assess the rate-distortion performance of such an object-based coder by means of numerical experiments in typical situations of interest, and single out, to the extent possible, the individual phenomena that contribute to the overall losses and gains. Since the usual coding gain does not make sense for S. Li and W. Li's SA-WT, we measured its compaction ability by analyzing the RD performance of a virtual oracle coder which spends bits only for quantization. This was a very important step because SA-WT losses turned out to be quite significant, especially at low rates. Although the quantization cost is by itself only a small fraction of the total cost, the reduced compaction ability of SA-WT has a deep effect also on the subsequent coding phase, the sorting pass of SPIHT. In fact, our experiments revealed this to be the main cause of SPIHT losses, while the presence of incomplete trees plays only a minor role. This is also confirmed by the fact that SA-SPIHT performs about as well as more sophisticated coding algorithms, and suggests that algorithms that code significance maps equally well perform equivalently at shape-adaptive coding regardless of how carefully their coding strategies have been tailored to accommodate object boundaries, and hence improving boundary handling is largely a wasted effort.

As for the gains, our analysis showed that they can be significant when the image presents sharp edges between relatively homogeneous regions but also that this is rarely the case with real-world images where the presence of smooth contours, and the inaccuracies of segmentation (for a few objects) or its large cost (for many objects) represent serious hurdles towards potential performance gains.

(a) (b)

Figure 16: Image peppers compressed at 0.05 bit/pixel with (a) flat and (b) object-based codings.



(a) (b)

Figure 17: Band 5 of the Landsat TM multispectral image (a) and its segmentation map (b).

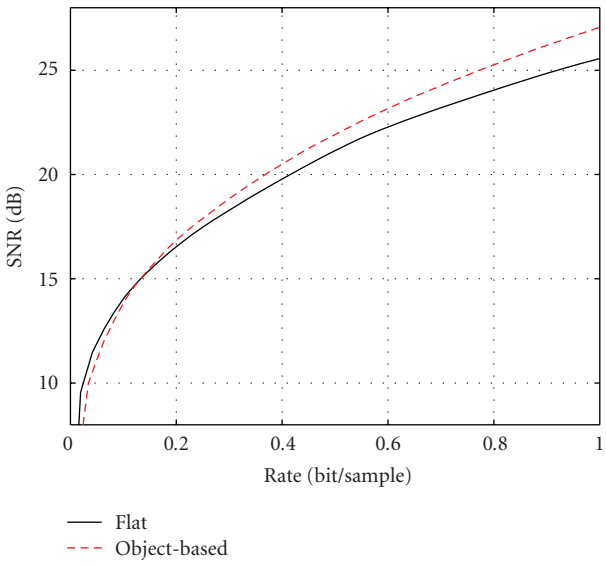


Figure 18: RD performance of flat and object-based codings for the Landsat TM image.

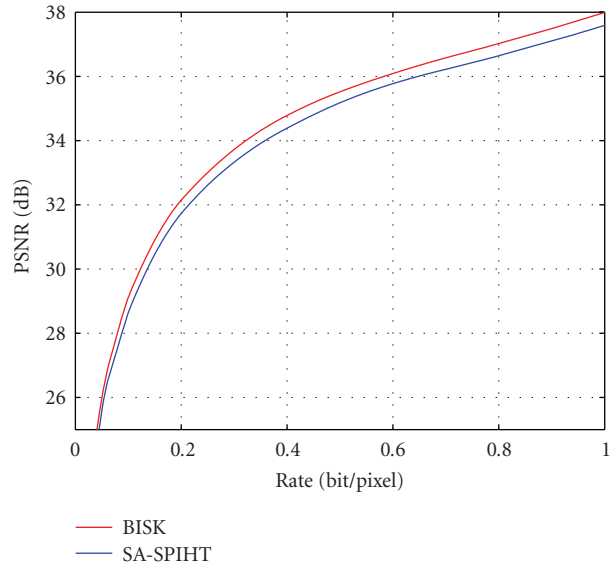


Figure 19: RD performance of BISK and SA-SPIHT for image peppers.

The experimental evidence (the bulk of which was not presented here) allows us to provide some simple guidelines for the use of wavelet-based OB-coding, by dividing operative conditions in three major cases.

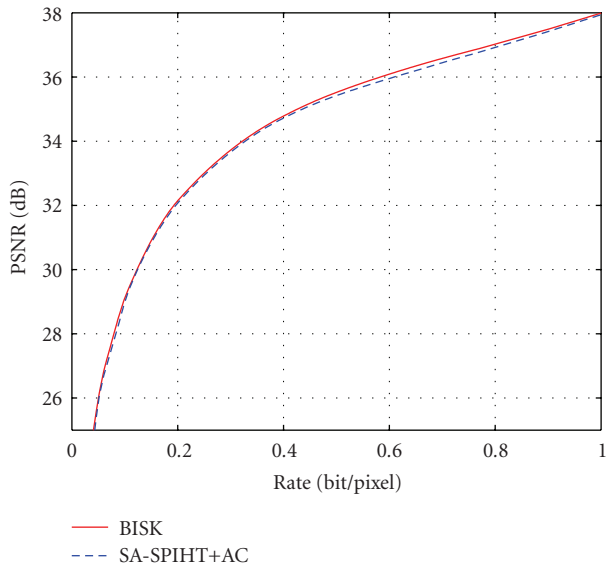


FIGURE 20: RD performance of BISK and SA-SPIHT with arithmetic coding for image peppers.

(1) A few large (say, over ten thousand pixels) objects with smooth contours

RD losses and gains are both negligible, hence performance is very close to that of flat wavelet-based coding, the best currently known. In this case, which is the most explored in the literature, resorting to wavelet-based coding, with S. Li and W. Li's transform and SA-SPIHT or BISK, is probably the best solution.

(2) Many small objects (or a few large objects with very active boundaries) at low rates

There are significant RD losses, both because of the reduced compaction ability of SA-WT and because the coding cost of the segmentation map is not irrelevant. This is the only case, in our opinion, where the wavelet-based approach leaves space for further improvements, as the introduction of new tools explicitly thought to encode objects rather than signals with arbitrary support.

(3) Many small objects (or a few large objects with very active boundaries) at high rates

Here, the losses due to the SA-WT and the side information become almost negligible, and the performance comes again very close to that of flat coding, making wavelet-based coding very competitive again.

This list accounts mainly for the losses, as performance gains are currently achievable only for some specific source, like multispectral images. Further improvements require probably a tighter interaction between coding and segmentation, with a better description of the graphical part of the image, for example by taking into account the profile as well as the position of the boundaries, or even an RD-driven segmentation.

ACKNOWLEDGMENT

The authors wish to express their gratitude to the anonymous reviewers whose many constructive comments have helped improve the paper.

REFERENCES

[1] M. Kunt, M. Benard, and R. Leonardi, "Recent results in high-compression image coding," *IEEE Transactions on Circuits and Systems*, vol. 34, no. 11, pp. 1306–1336, 1987.

[2] ISO/IEC JTC1, *ISO/IEC 14496-2: coding of audio-visual objects* April 2001.

[3] M. Madhavi and J. E. Fowler, "Unequal error protection of embedded multimedia objects for packet-erasure channels," in *Proceedings of the IEEE International Workshop on Multimedia Signal Processing*, pp. 61–64, St. Thomas, Virgin Islands, USA, December 2002.

[4] T. Gan and K.-K. Ma, "Weighted unequal error protection for transmitting scalable object-oriented images over packet-erasure networks," *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 189–199, 2005.

[5] J. E. Fowler and D. N. Fox, "Wavelet-based coding of three-dimensional oceanographic images around land masses," in *Proceedings of IEEE International Conference on Image Processing (ICIP '00)*, vol. 2, pp. 431–434, Vancouver, BC, Canada, September 2000.

[6] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, 2000.

[7] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, 2001.

[8] G. Xie and H. Shen, "Highly scalable, low-complexity image coding using zeroblocks of wavelet coefficients," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 6, pp. 762–770, 2005.

[9] X. Sun, J. Foote, D. Kimber, and B. S. Manjunath, "Region of interest extraction and virtual camera control based on panoramic video capturing," *IEEE Transactions on Multimedia*, vol. 7, no. 5, pp. 981–990, 2005.

[10] S. Li and W. Li, "Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 5, pp. 725–743, 2000.

[11] M. Cagnazzo, G. Poggi, L. Verdoliva, and A. Zinicola, "Region-oriented compression of multispectral images by shape-adaptive wavelet transform and SPIHT," in *Proceedings of IEEE International Conference on Image Processing (ICIP '04)*, vol. 4, pp. 2459–2462, Singapore, October 2004.

[12] A. Kawanaka and V. R. Algazi, "Zerotree coding of wavelet coefficients for image data on arbitrarily shaped support," in *Proceedings of the Data Compression Conference (DCC '99)*, p. 534, Snowbird, Utah, USA, March 1999.

[13] G. Minami, Z. Xiong, A. Wang, and S. Mehrotra, "3-D wavelet coding of video with arbitrary regions of support," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 9, pp. 1063–1068, 2001.

[14] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, 1996.

[15] O. Egger, P. Fleury, T. Ebrahimi, and M. Kunt, "High-performance compression of visual information-a tutorial review—I: still pictures," *Proceedings of the IEEE*, vol. 87, no. 6, pp. 976–1011, 1999.

[16] B. E. Usevitch, "A tutorial on modern lossy wavelet image compression: foundations of JPEG 2000," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 22–35, 2001.

[17] T. Sikora, "Trends and perspectives in image and video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 6–17, 2005.

[18] J. Tian and R. Wells Jr., "Embedded image coding using wavelet difference reduction," in *Wavelet Image and Video Compression*, P. Topiwala, Ed., pp. 289–301, Kluwer Academic, Norwell, Mass, USA, 1998.

[19] Z. Lu and W. A. Pearlman, "Wavelet coding of video object by object-based SPECK algorithm," in *Proceedings of the 22nd Picture Coding Symposium (PCS '01)*, pp. 413–416, Seoul, Korea, April 2001.

[20] Z. Liu, J. Hua, Z. Xiong, Q. Wu, and K. Castleman, "Lossy-to-lossless ROI coding of chromosome images using modified SPIHT and EBCOT," in *Proceedings of IEEE International Symposium on Biomedical Imaging (ISBI '02)*, pp. 317–320, Washington, DC, USA, July 2002.

[21] J. E. Fowler, "Shape-adaptive tarp coding," in *Proceedings of IEEE International Conference on Image Processing (ICIP '03)*, vol. 1, pp. 621–624, Barcelona, Spain, September 2003.

[22] J. E. Fowler, "Shape-adaptive coding using binary set splitting with K-D trees," in *Proceedings of IEEE International Conference on Image Processing (ICIP '04)*, vol. 5, pp. 1301–1304, Singapore, October 2004.

[23] P. Prandoni and M. Vetterli, "Approximation and compression of piecewise smooth functions," *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, vol. 357, no. 1760, pp. 2573–2591, 1999.

[24] V. Ratnakar, "RAPP: lossless image compression with runs of adaptive pixel patterns," in *Proceedings of the 32nd Asilomar Conference on Signals, Systems & Computers*, vol. 2, pp. 1251–1255, Pacific Grove, Calif, USA, November 1998.

[25] V. K. Goyal, "Theoretical foundations of transform coding," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 9–21, 2001.

[26] J. Katto and Y. Yasuda, "Performance evaluation of subband coding and optimization of its filter coefficients," in *Visual Communications and Image Processing*, vol. 1605 of *Proceedings of SPIE*, pp. 95–106, Boston, Mass, USA, November 1991.

[27] B. E. Usevitch, "Optimal bit allocation for biorthogonal wavelet coding," in *Proceedings of the 6th Data Compression Conference (DCC '96)*, pp. 387–395, Snowbird, Utah, USA, March-April 1996.

[28] M. Cagnazzo, G. Poggi, and L. Verdoliva, "Costs and advantages of shape-adaptive wavelet transform for region-based image coding," in *Proceedings of IEEE International Conference on Image Processing (ICIP '05)*, vol. 3, pp. 197–200, Genova, Italy, September 2005.

[29] http://sipi.usc.edu/database/.

[30] M. Cagnazzo, R. Gaetano, G. Poggi, and L. Verdoliva, "Region based compression of multispectral images by classified KLT," in *Proceedings of IEEE International Conference on Image Processing (ICIP '06)*, Atlanta, Ga, USA, October 2006.

[31] J. E. Fowler, "QccPack: an open-source software library for quantization, compression, and coding," in *Applications of Digital Image Processing XXIII*, vol. 4115 of *Proceedings of SPIE*, pp. 294–301, San Diego, Calif, USA, July 2000.

[32] http://qccpack.sourceforge.net/.