SPATIAL ANALYSIS-SYNTHESIS FOR IMPROVEMENT OF WAVELET CODERS

Benjamin Le Guen[1,2], *Stéphane Pateux*[1], *Jacques Weiss*[2]

[1] France Télécom R&D, 4, rue du Clos Courtel, 35512 Cesson-Sévigné Cedex, France
[2] Supélec-SCEE/IETR-AC, avenue de la Boulaie, CS 47601, 35576 Cesson-Sévigné Cedex, France

ABSTRACT

This paper describes an adaptive image compression scheme built upon a conventional wavelet coder. The principle is to warp the input signal so as to minimize its coding cost. First, the warping parameters are estimated through an analysis step. Then, the warped image is sent to the basic coder. The warping parameters are transmitted independently. Finally, a synthesis step reconstructs the original image by inverse warping.

This approach of the adaptivity issue is different from previous methods seeking to adapt the wavelet rather than the signal. Our technique enables to take advantage of optimized wavelet coders such as JPEG2000. Moreover, the minimization procedure does not make any assumption about the regions of high coding cost. An implementation of the principles is presented where the warping transform is modelized by a regular 2D-Mesh. Compression results of the Analysis-Synthesis (A-S) scheme show a significant visual quality improvement for non-texture images compared to JPEG2000.

1. INTRODUCTION

JPEG2000 [1] is a powerful coding scheme which led to highly optimized and widely spread fast implementations. However, the core of this latest compression norm is the 2D-separable Discrete Wavelet Transform (DWT) which sub-optimality has been established (see [2] for example). Indeed, the DWT takes advantage of the correlations along the horizontal and vertical directions only, but natural images include regularity curves of various shapes. This work is particularly interested in improving the reconstruction quality of contours provided by JPEG2000. In other words, we want to get rid of the ringing artefacts which impair the visual reconstruction quality of contours at low rates.

A great effort has been made through the past decade to find new representation basis possessing directional properties. A first class of approaches aims to project the image on a dictionary of *fixed* anisotropic atoms, such as Contourlets [3]. But the redundancy and the non-adaptivity of these techniques motivates the research on new *adaptive* basis. Adaptivity can be looked for in a variety of ways. We essentially distinguish between two kinds of approaches. On the one hand, adaptivity can be obtained by extracting from the signal a relevant geometrical content *beforehand*. Whether this extraction resorts on detection of contours [4] or regular curves [2], these techniques are based on geometrical *a priori*. This does not insure a fair modelization of energetical properties in the transform domain. On the other hand, recent approaches proposed to express an energy in the transform domain as a function of adaptivity parameters. Minimizing this energy leads to optimal parameters in a certain sense. For this purpose, most techniques [5, 6, 7] implement an exhaustive search. For complexity and compacity concerns, this imposes to segment the image into blocks and *independently* compute for each of these blocks a *limited* number of parameters. Hence, it leads to a discontinuous and low-level adaptivity. Note that some techniques, like [2, 8], propose regularization procedures to optimize the parameters in a Rate/Distorsion (R/D) sense.

In this paper, we propose a new approach to the adaptivity issue. Instead of modifying the core structure of a conventional wavelet coder, we choose to *warp* the input image so as to adapt it to the coder. It can be argued that a similar purpose is pursued in [2] and [4] on a *Block* basis. However this block-based processing requires to modify the wavelet coder to insure a special treatment for border cases. Also, the signal adaptation we propose is not based on any geometric *a priori* but aims to minimize the coding cost of the warped image. The next section describes our Analysis-Synthesis (A-S) scheme with a particular focus on this energy formulation and minimization considering any warping model. In the third section, we present the specificities of an implementation with a 2D Mesh. Different R/D improvements of the model are proposed. The last section will present comparative compression results.

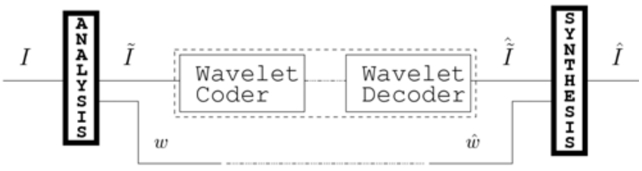


Figure 1: The A-S scheme. Analysis warps the input image so as to adapt the resulting signal to any conventional wavelet coder.

2. A-S SCHEME

2.1 Notations

Let us define a reversible transformation w that maps a position $\tilde{\mathbf{p}}$ in a Domain $\tilde{\mathscr{D}}(\subset \mathbb{Z}^2)$ - which we call the *Warped* Domain - to a position $\mathbf{p}$ in the *Image* Domain $\mathscr{D}(\subset \mathbb{Z}^2)$. Given w, the original Image I and the Warped Image $\tilde{I}$ are related as follows:

$$\tilde{I}(\tilde{\mathbf{p}}) = I(w(\tilde{\mathbf{p}})) \Leftrightarrow I(\mathbf{p}) = \tilde{I}(w^{-1}(\mathbf{p}))$$

With these notations, our proposed coding scheme is presented in Fig. 1. It is composed of three blocks:

Analysis: estimates the parameters of w. This estimation follows an energy formulation which aims to minimize the coding cost of $\tilde{I}$ in a conventional wavelet basis.

Codec: $\tilde{I}$ is encoded and decoded using any conventional wavelet coder. Parameters of w are coded and transmitted independently.

Synthesis: takes as inputs the decoded warping transformation $\hat{w}$ and warped image $\hat{\tilde{I}}$ and reconstructs the original image by inverting the warping, i.e. $\hat{I} = \hat{\tilde{I}}(\hat{w}^{-1})$.

In this section, we will only assume w to be a parametric model composed of a set of N_p undefined parameters $\{\mathbf{p}_i\}_{i=1..N_p}$. In the next paragraph, we focus on the Analysis Step. The key issue of this step is to express the coding cost of the Warped Image $\tilde{I}$ as a function of the parameters. Because geometrical features of natural images strongly contributes to the coding cost of I, one can expect the computed w to have some relation with the image geometry. This will be supported by the analysis results provided in section 3.

2.2 Analysis

2.2.1 *Energy Formulation*

Let $\psi_{j\mathbf{n}}$ refer to a 2D discrete analysis Wavelet scaled with a factor 2^{-j} ($j \in \mathbb{Z}_+$) and translated to a position $\mathbf{n} \in \mathbb{Z}^2$. The dot product of $\tilde{I}$ with $\psi_{j\mathbf{n}}$ gives a wavelet coefficient $c_{j\mathbf{n}}$.

Considering the set of wavelet coefficients $\left\{c_{j\mathbf{n}}\right\}_{j=0..J-1}$ for a user-fixed J, we define the *Coding Cost* CC_J as:

$$CC_J = \sum_{j=0}^{J-1} \gamma_j^2 \cdot \sum_{\mathbf{n}} c_{j\mathbf{n}}^2 \tag{1}$$

γ_j is a weight which can be adjusted in relation to the probability law $P(c_{j\mathbf{n}})$ of the coefficients at the scale 2^{-j}. This formulation is particularly well fitted to Gaussian distributions in the subbands. Indeed, if we assume $P(c_{j\mathbf{n}})$ to be a gaussian probability law $\mathcal{N}(0, 1/\gamma_j^2)$, then the *Coding Cost* of the transformed signal is $CC_J = -\sum_{j=0}^{J-1} log_2(P(c_{j\mathbf{n}}))$ which can be reduced as in Eq. (1).

To make the minimization of CC_J tractable, we need to express it as a function of w. Now, an inverse wavelet transform gives:

$$\tilde{I}(\tilde{\mathbf{p}}) = \sum_{j=0}^{j_{max}} \sum_{\mathbf{n}} c_{j\mathbf{n}} \cdot \psi_{j\mathbf{n}}^*(\tilde{\mathbf{p}}) \tag{2}$$

where $\psi_{j\mathbf{n}}^*$ refers to the synthesis kernel corresponding to $\psi_{j\mathbf{n}}$, and j_{max} to the greatest possible decomposition level of $\tilde{I}$.

The right part of Eq. (2) can be decomposed to yield:

$$\sum_{j=0}^{j_0-1} \sum_{\mathbf{n}} c_{j\mathbf{n}} \cdot \psi_{j\mathbf{n}}^*(\tilde{\mathbf{p}}) = \tilde{I}(\tilde{\mathbf{p}}) - \sum_{j=j_0}^{j_{max}} \sum_{\mathbf{n}} c_{j\mathbf{n}} \cdot \psi_{j\mathbf{n}}^*(\tilde{\mathbf{p}}) \tag{3}$$

for any $j_0 \in [0, j_{max}]$.

As the signal $\sum_{j=j_0}^{j_{max}} \sum_{\mathbf{n}} c_{j\mathbf{n}} \cdot \psi_{j\mathbf{n}}^*$ is the approximation of $\tilde{I}$ obtained by setting to 0 each coefficient $c_{j\mathbf{n}}$ for $j \in \{0..j_0 - 1\}$, we will refer to it as $\tilde{I}_{j_0}$.

From Eq. (3), Parseval theorem gives:

$$\sum_{j=0}^{j_0-1} \sum_{\mathbf{n}} c_{j\mathbf{n}}^2 = \sum_{\tilde{\mathbf{p}}} (\tilde{I}(\tilde{\mathbf{p}}) - \tilde{I}_{j_0}(\tilde{\mathbf{p}}))^2$$

We then obtain the desired relation between CC_J and w:

$$CC_J(w) = \sum_{j_0=1}^{J} \eta_{j_0} \cdot \sum_{\tilde{\mathbf{p}}} (I(w(\tilde{\mathbf{p}})) - \tilde{I}_{j_0}(\tilde{\mathbf{p}}))^2, \tag{4}$$

with the new weights η_{j_0} verifying the following relations:

$$\gamma_j^2 = \sum_{j_0=j+1}^{J} \eta_{j_0} \Rightarrow \left\{ \begin{array}{l} \eta_J = \gamma_{J-1}^2 \\ \eta_{j_0} = \gamma_{j_0-1}^2 - \gamma_{j_0}^2 \end{array} \right.$$

In practice, under the gaussian assumption, all the weights η_{j_0} are positive. Indeed, for most natural images, the variance $1/\gamma_j^2$ in a subband j increases with j, which means $\gamma_{j-1}^2 > \gamma_j^2 \ \forall j$.

Because CC_J is now expressed as a function of w, we can search for the set of parameters $\{\mathbf{p}_i\}_i$ which minimizes it.

2.2.2 *The Minimization Algorithm*

Each signal $\tilde{I}_{j_0}$ in Eq.(4) can be computed only *after* w and $\tilde{I}$ have been computed. For that reason, we decide to consider $\tilde{I}$ as a new variable in the minimization process, from which each $\tilde{I}_{j_0}$ can be computed. Then, we formulate the minimization issue as a joint optimization where the best couple $(w, \tilde{I})$ is looked for, subject to the constraint: $\tilde{I} = I(w(\tilde{\mathbf{p}}))$. We propose to solve this optimization problem through an iterative Expectation Maximization-like procedure. At each iteration, an estimate of w is updated given the current observation of $\tilde{I}$. This `Update-Warping` process is based on the minimization of Eq.(4). In return, $\tilde{I}$ is refined given the updated observation of w:

```
COMPUTE-WARPING(I)
1  $n \leftarrow 0, w \leftarrow Id, \tilde{I}^{(0)} \leftarrow I$ // Initialization
2  while ($n < n_{max}$)
3      do $n \leftarrow n+1$
4         $w^{(n)} \leftarrow$ UPDATE-WARPING$(I, w^{(n-1)}, \tilde{I}^{(n-1)})$
5         $\tilde{I}^{(n)} = I(w^{(n)})$
6  return $w$
```

From the previous algorithm, it is clear that the optimization complexity is related to step 4. Knowing $\tilde{I}^{(n-1)}$, it is possible to compute each current approximation $\tilde{I}_{j_0}^{(n-1)}$ and minimize Eq. (4) taking $w^{(n-1)}$ as the initial guess for $w^{(n)}$.

Further, it is possible to *simplify* the energy to minimize. Indeed, under the condition that $\sum_{j_0=1}^{J} \eta_{j_0} \neq 0$ (verified in practice by the gaussian model), we can demonstrate that:

$$\underset{w^{(n)}}{\arg\min} CC_J(w^{(n)}) = \underset{w^{(n)}}{\arg\min} \sum_{\tilde{\mathbf{p}}} (I(w^{(n)}(\tilde{\mathbf{p}})) - \tilde{I}_{ref}^{(n)}(\tilde{\mathbf{p}}))^2, \tag{5}$$

$$\text{with} \quad \tilde{I}_{ref}^{(n)}(\tilde{\mathbf{p}}) = \frac{\sum_{j_0=1}^{J} \eta_{j_0} \cdot \tilde{I}_{j_0}^{(n-1)}(\tilde{\mathbf{p}})}{\sum_{j_0=1}^{J} \eta_{j_0}}.$$

The problem expressed in the right hand side of Eq. (5) is the minimization of a *Displaced Frame Difference* between $\tilde{I}$ and a current *reference* frame $\tilde{I}_{ref}^{(n)}$. It is a well-known problem in the Video Coding Community, where w does not refer to a spatial but a temporal operator. A variety of solutions has been proposed, which complexity depends on the chosen motion model. In the next section, we will present the specificities of an implementation using an active mesh as the model for w.

2.3 Synthesis

The synthesis step consists on inverting the warping carried out at the end of the analysis step. Whatever warping model may be chosen, the complexity of this processing remains very low and can be performed in real-time by today's processors. This low complexity upgrade on the decoding side is very important for the clients in applications such as Video Streaming or Broadcasting.

3. IMPLEMENTATION WITH A MESH

3.1 Warping Transform Modelization

Let us recall that the warping transform w is a coordinate mapping between $\tilde{\mathcal{D}}$ and $\mathcal{D}$. If we modelize this transform with any active mesh [9], two types of parameters must be taken into account: geometric parameters, i.e. the positions of the nodes in $\tilde{\mathcal{D}}$ and $\mathcal{D}$, and connectivity parameters, i.e. how these nodes are linked to one another. All these parameters have a cost.

Here, we choose to work with a *regular* mesh so that no connectivity parameter must be transmitted. Let $\{\mathbf{p}_i\}_{i=1..N_p}$ be the N_p positions of its nodes in $\mathcal{D}$. In accordance with Fig. 2, each position $\mathbf{p}_i$ is arbitrarily mapped to a position $\tilde{\mathbf{p}}_i$ in $\tilde{\mathcal{D}}$. We decide to place the positions $\tilde{\mathbf{p}}_i$ on a uniform grid of squares in $\tilde{\mathcal{D}}$. The positions of the mesh in $\tilde{\mathcal{D}}$ are completely known once the edge size S_e of a square is given. Eventually, the only warping parameters to transmit are S_e and the N_p displacements from the uniform grid in $\mathcal{D}$.

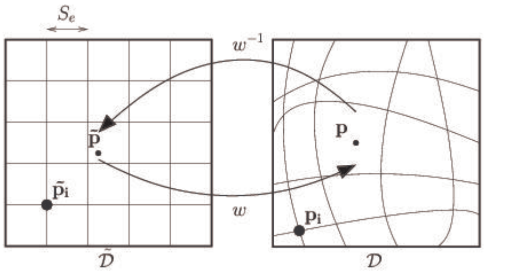


Figure 2: Modelization of w with a quadrangular mesh.

Letting $\phi(\tilde{\mathbf{p}})$ refer to a 2D shape function defined in $\tilde{\mathscr{D}}$ (e.g. the bilinear function), the warping transform w is defined as follows:

$$w(\tilde{\mathbf{p}}) = \sum_{i=1}^{N_p} \phi(\tilde{\mathbf{p}} - \tilde{\mathbf{p}}_i) \cdot \mathbf{p}_i \tag{6}$$

Notice that all the tests presented in this paper were carried out with a *quadrangular* mesh, as represented in Fig. 2.

3.2 Analysis applied to a Mesh

3.2.1 Estimation of Nodes Displacements in $\mathscr{D}$

The computation of the positions $\{\mathbf{p}_i\}_i$ follows the algorithm described in paragraph 2.2.2. As we said, the technical issue of this algorithm is the update of w at each iteration. In this work, we choose to implement this step using a gradient descent algorithm (see [9] for details). The linearization of Eq. (5) gives a sparse linear system. The solution of this system is a set of displacements for each position $\mathbf{p}_i$. Initially, the nodes are placed uniformly in $\mathscr{D}$ so that $\mathbf{p}_i = \tilde{\mathbf{p}}_i \; \forall i \in \{1,..,N_p\}$ (i.e. $w = Id$). At each iteration, the positions are updated *globally* by solving the linear system. Experiments have shown that no significant displacement occurs after 10 to 15 iterations. In term of complexity, this procedure can be compared to a motion estimation (with an active mesh) between a current frame and a reference frame, with the reference frame being updated at each iteration. The basic algorithm is greedy but can be sped up in numerous ways, which is not the topic of this article.

3.2.2 Visual Results

Fig. 3 shows the outputs of the analysis step when applied to images *Cameraman* and *Barbara* 256×256 with $S_e = 4$. For illustration convenience a deformation energy such as the one introduced in [9] was added to the minimization process. This energy acts like a spring force between each pair of nodes and forces the mesh not to be overly deformed.

On the right, the estimated meshes in $\mathscr{D}$ are shown. We here recall that our energy formulation **do not integrate any assumption about the Image geometry**. It is clear however that the computed meshes do have some geometrical properties. Particularly, we notice a significant concentration of nodes around contours, so that it is possible to recognize the main geometric features of the images.

On the left, the warped Image $\tilde{I}$ in $\tilde{\mathscr{D}}$ is represented. Two main observations can be made. Firstly, we notice that the warping acts as a magnifying mirror on discontinuities: contours are stretched in the direction orthogonal to the regularity. This is a property we could expect because stretching means smoothing and smooth discontinuities can be represented with fewer wavelet coefficients. Secondly, we observe that w has a tendency to produce "stairs-like" shapes in $\tilde{\mathscr{D}}$. This is particularly noticeable on the tripod in *Cameraman* but can be observed by zooming on almost all contours. Again, this is a result we could expect because conventional wavelets efficiently represent horizontal and vertical regularity lines.

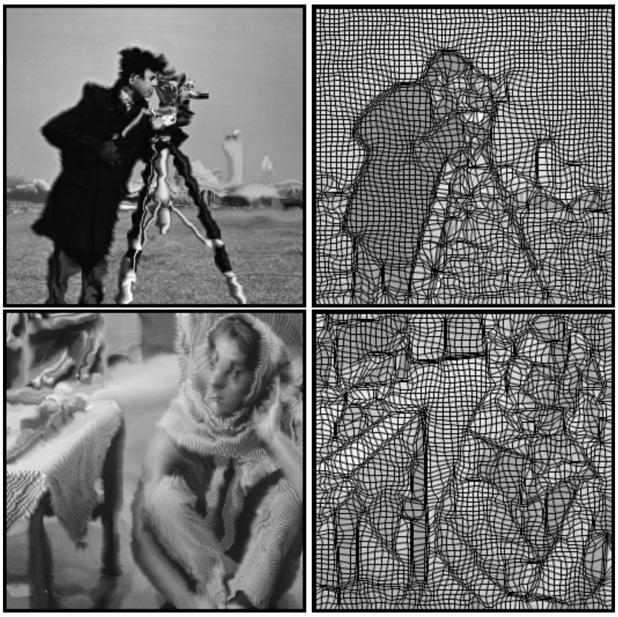


Figure 3: Analysis Results. [RIGHT] Mesh obtained in $\mathscr{D}$, [LEFT] The warped Image $\tilde{I}$ in $\tilde{\mathscr{D}}$.

3.3 R/D Improvements

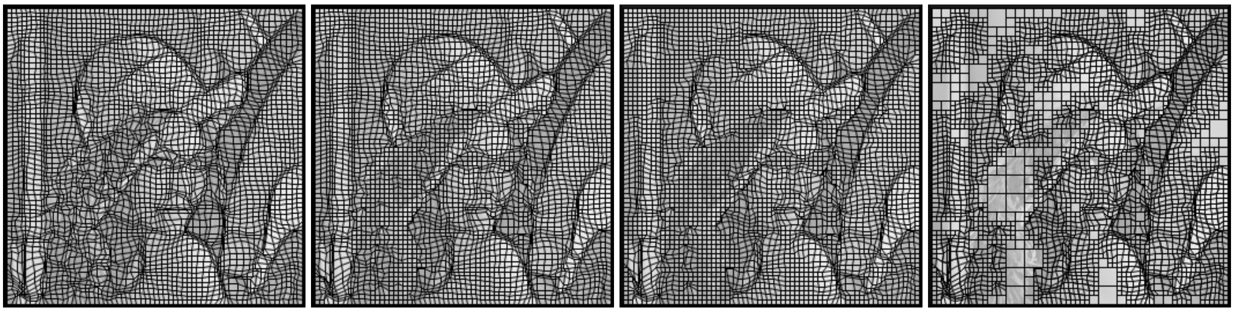
3.3.1 Limitation of Texture Distortion

As we work in a discrete setting, a non-isotropic transformation w, as defined in Eq. (6), cannot comply with the reversibility assumption due to the resampling implied. Indeed, such a transformation authorizes non-reversible warpings, such as stretchings or contractions. Considering the properties of the estimated mesh, this non-reversibility has a different *visual* impact on reconstructed *contours* and reconstructed *textures*:

Contours: we noticed that w has a general tendency to stretch the discontinuities. This means that the resolution is increased on these areas when going from $\mathscr{D}$ to $\tilde{\mathscr{D}}$. After inverse warping, the loss introduced has an impact on the MSE, but not visually.

Textures: textures are here considered as very fine features - like the feathers on *Lena*'s hat or the checked materials in the image *Barbara* - that cannot be extracted well by a mesh. Because these regions sometimes have a significant gradient activity, they can provoke displacements of nodes during the estimation process. Also, these regions can be close to contours where the nodes have moved. In those cases, the non-reversibility of the resampling has a bad impact on the MSE and produces some smoothing on the reconstructed texture areas. *If the user has a precise knowledge of the original image*, the visual loss on textures can be disturbing. We now propose a simple post-processing after analysis to limit this loss.

Let us denote by I_{new} the image obtained after inverting the warping: $I_{new}(\mathbf{p}) = \tilde{I}(w^{-1}(\tilde{\mathbf{p}}))$. Experiments have shown that the image *residue* $I_{res} = I_{new} - I$ has most of its energy in texture areas. The idea is to analyze I_{res} on a block basis in order to detect the regions of highest variances. In a R/D sense, it can then be preferable not to perform any warping in these regions: **the nodes are put back to their initial location on the uniform grid**. More precisely, let B_s denote a block of size $s \times s$ and T_{var} a tolerance threshold. Then, we compute the variance in each block B_s of I_{res}. If this variance is greater than T_{var}, the region is recognized as a texture region. T_{var} is an adjustment parameter which must be set appropriately. We observed that $T_{var} \approx 30$ gives a fair separation of textures and contours errors. Fig. 4(a) shows an estimated mesh on *Lena* and Fig. 4(b) the result of the described post-processing taking $T_{var} = 36$. We notice that most of the quadrangles covering the feathers have been forced to squares, whereas contours remain well extracted.



(a) Original Mesh
4098 bytes

(b) Texture loss limitation
3848 bytes

(c) Precision reduction
3256 bytes

(d) Generated Quadtree
3187 bytes

Figure 4: R/D Post-processings on the mesh.

3.3.2 *Mesh Entropy Reduction*

The previous post-processing has a good side effect: forcing some quadrangles to be squares increase the number of null displacements and reduces the coding cost of the mesh. In this paragraph, we propose another simple trick to reduce its cost while keeping its geometrical properties. Indeed, the estimated mesh contains many quadrangles which are very close in shape to their initial square. In a R/D sense, it can be preferable to constrain these quadrangles to be squares. More precisely, let $\{\Delta \mathbf{p}_i, i = 1..4\}$ be the estimated displacements of the 4 nodes of a quadrangle, and T_{disp} a tolerance threshold. Then, for each quadrangle, we compute the deformation criteria $\sum_{i=1}^{4} \|\Delta \mathbf{p}_i\|^2$. If this criteria is lower than T_{disp}, then the quadrangle is forced back to its initial square. Fig. 4(c) shows the result of applying this procedure after the texture loss reduction process. Experiments have shown that $T_{disp} \approx S_e^2/10$ was generally a good compromise.

After performing the two previous processings, the set of symbols representing the mesh includes a large number of zeros. Rather than encoding all these zeros, we finally propose to represent and encode the mesh as a quad-tree structure. The construction of this quad-tree is an up-to-bottom procedure. The value of the root is decided at the level of the mesh. If the mesh has at least one non-square quadrangle, we set this value to 1. Then, a dyadic decomposition of the nodes is made, and the decision process is repeated on each of the 4 dyadic sets of quadrangles. Otherwise, we set this value to 0 and end the process on the current branch. Fig. 4(d) presents the quad-tree obtained with this procedure. Comparing the cost of the quad-tree to the cost of the initial estimated mesh, we notice a reduction of the coding cost of about 20%. This rate saving is important because more bits can be used for the encoding of $\tilde{I}$.

The quad-tree representation has an important visual interpretation: it clearly delineates regions for which the analysis step brings a gain from regions which are better encoded by JPEG2000 alone. The notions of gain or quality are closely related to the values of T_{var} and T_{disp} and can be left to the appreciation of the user. In the extreme case that $T_{var} = 0$ or $T_{disp} = \infty$, the quad-tree is reduced to its root (i.e. 1 bit) and the coding scheme reduces to JPEG2000. This case is better suited to images only composed of textures. At the other end of the range of possibilities, $T_{var} = \infty$ or $T_{disp} = 0$, and the leaves of the quad-tree are the entire set of estimated displacements $\{\Delta \mathbf{p}_i\}_i = 1..N_p$. This choice is particularly relevant for cartoon images with no texture. Between those 2 extreme cases, the user can choose to give more weight to contours or textures in the visual quality of reconstructed images.

4. COMPRESSION RESULTS

4.1 Coding Parameters

The compression results presented here were obtained by applying the A-S scheme on *Lena*, *Barbara*, *Peppers* and *Airplane*. All the input images have a dimension of 512×512. We used the same coding parameters for all tests. For the analysis step, we set $S_e = 16$ and $J = 4$. After analysis, the meshes were processed as described in paragraph 3.3 setting $T_{var} = 36$ and $T_{disp} = S_e^2/10$. Non-zeros displacements were quantized with a half-pixel precision and the set of quad-tree symbols was encoded using arithmetic coding. $\tilde{I}$ was encoded with JPEG2000 VM8.0 with its default parameters.

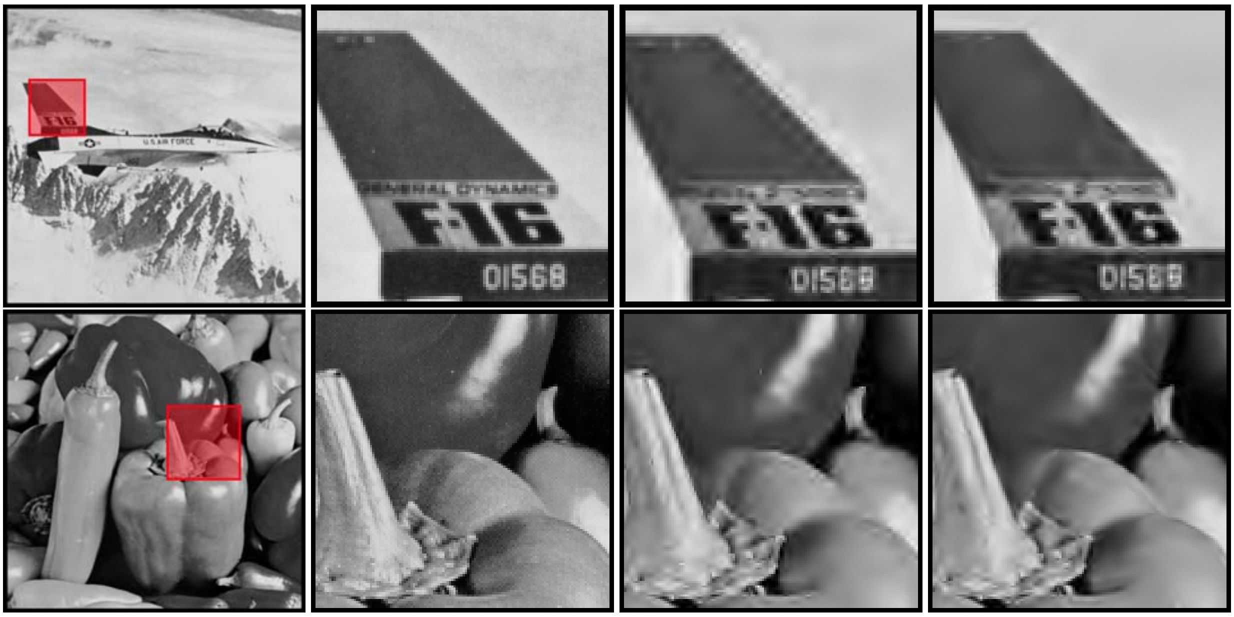
4.2 Quality Assessment

The non-reversibility of w was mentioned in paragraph 3.3.1 and a method was proposed to limit the visual impact on textures. Although this method enables to increase PSNR values, the goal of the A-S scheme upgrade is clearly not to obtain high PSNR values because the loss on contours will always have a bad impact on these values. However, all the experiments we carried out showed that this loss on contours has no visual impact. Therefore, to assess fairly the efficiency of our approach we will present two different curves denoted C_u and C_{new}. The first one is the usual PSNR curve measuring the distortion between the decoded image $\hat{I}$ and the original image I . The second one measures the distortion between $\hat{I}$ and I_{new} (we recall that $I_{new}(\mathbf{p}) = I(w^{-1} \circ w(\mathbf{p}))$). We claim that the second curve better fits the subjective visual quality at low to medium rates. A similar evaluation strategy was proposed in [10] in the context of scalable t+2D wavelet video coding. Let us notice that the usual PSNR value of I_{new} with respect to I is about 44 dB for every tested images and visually no distortion is perceivable.

4.3 Results

Fig. 5 shows the numerical compression results obtained with the A-S scheme. The cost of the mesh is included in the curves. Values are compared to those obtained with JPEG2000 alone. We also present in Fig. 4 different zooms on reconstructed contours which illustrate the visual gain provided by the A-S scheme compared to JPEG2000. The same gain is noticed on most contours for the tested images.

Even though the A-S scheme is not expected to provide high PSNR values, we observe that C_u remains very close to the JPEG2000 curve for each image at all rates. This gives a good indication that the visual quality of reconstructed *textures* is very close in both cases. To assess the visual quality of reconstructed *contours* at low to medium rates, we recommend to compare the JPEG2000 curve with C_{new}. The set of tested images covers a fair range of possible natural content. In the case of *Barbara*, we do not observe a significant gain of the A-S scheme. The reason is that the image is a complex combination of geometrical objects and textures which cannot be well described altogether by a mesh with $S_e \approx 16$. A larger T_{var} would better reconstruct textures, but at the price of the visual gain on contours. In the case of *Lena* and *Airplane*, we do observe a significant visual gain on contours as well as in the overall quality of reconstructed images. This gain is well indicated by C_{new} at low rates. These two images are composed of one dominant object surrounded by a reasonable amount of texture. The contours of the object are finely extracted during the analysis and



Original Image | Original (zoom) | JPEG2000 at 0.2 bpp | A-S scheme at 0.2 bpp

subsequent post-processings do not alter this precision. Finally, the results obtained for *Peppers* illustrate all the potential of the A-S scheme. Indeed, *Peppers* is very like a cartoon image: it is composed of geometrical objects and almost no texture appears. Therefore, no conflict between geometry and textures occurs during the estimation of the mesh or the post-processings, and contours can be represented very finely with a small number of coefficients.

5. CONCLUSION

We have presented a new adaptive coding scheme built upon a conventional wavelet coder. Adaptivity is not obtained by changing the representation basis but by changing the input image. An analysis step computes the warped version of the original image which has the lowest coding cost. The warped image is then sent to the wavelet coder. After reception of the decoded warped image, the original image is reconstructed through a synthesis step. An implementation of this scheme has been described where the warping transformation is modelized by a regular quadrangular mesh. Different R/D improvements have been proposed. Compression results have shown that the A-S scheme brings a significant gain in the visual quality of reconstructed contours. The extension of the A-S scheme to video coding is currently in development. For future work, we wish to investigate the scalable properties of the new representation. Also, replacing the wavelet coder by a DCT coder is under study.

REFERENCES

[1] E. Skodras, C. Christopoulos, and T. Ebrahimi, “The JPEG 2000 still image compression standard,” *IEEE Signal Processing Magazine*, vol. 18, pp. 36–58, Sept. 2001.

[2] E. Le Pennec and S. Mallat, “Sparse Geometric Image Representations with Bandelets,” *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 423–438, Apr. 2005.

[3] M. N. Do and M. Vetterli, “The Contourlet Transform: An Efficient Directional Multiresolution Image Representation,” *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2091–2106, December 2005.

[4] D. Taubman and A. Zakhor, “Orientation adaptive subband coding of images,” *IEEE Transations on Image Processing*, vol. 3, pp. 421–436, July 1994.

[5] W. Ding, F. Wu, and S. Li, “Lifting-based Wavelet Transform with Directionally Spatial Prediction,” December 2004.

[6] G. Peyré and S. Mallat, “Discrete bandelets with geometric orthogonal filters,” September 2006.

[7] D. Wang, L. Zhang, A. Vincent, and F. Speranza, “Curved Wavelet Transform for Image Coding,” *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2413–2421, August 2006.

[8] V. Chappelier and C. Guillemot, “Oriented wavelet transform on a quincunx pyramid for image compression,” September 2005.

[9] Y. Wang and O. Lee, “Active mesh - a feature seeking and tracking image sequence representation scheme.,” *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 610–624, September 1994.

[10] U. Benzler and M. Wien, “Results of SVC CE3 (Quality Evaluation),” Proposal for MPEG2004/M10931, ISO/IEC JTC1/SC29/WG11, July 2004.

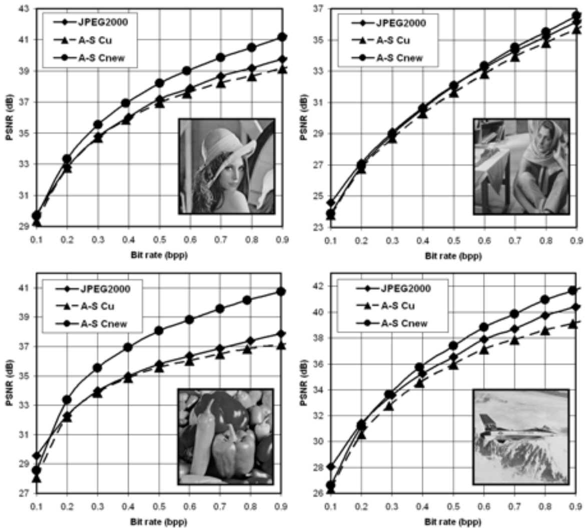


Figure 5: PSNR Curves. The mesh cost is 0.027 bpp for *Lena*, 0.029 bpp for *Barbara*, 0.043 bpp for *Peppers* and 0.035 bpp for *Airplane*.