Color LAR codec: a color image representation and compression scheme based on local resolution adjustment and self-extracting region representation

Olivier Déforges, Marie Babel, Laurent Bédat, Joseph Ronsin

▶ **To cite this version:**

Olivier Déforges, Marie Babel, Laurent Bédat, Joseph Ronsin. Color LAR codec: a color image representation and compression scheme based on local resolution adjustment and self-extracting region representation. IEEE Transactions on Circuits and Systems for Video Technology, 2007, 17 (8), pp.974-987. hal-00168359

HAL Id: hal-00168359
https://hal.science/hal-00168359

Submitted on 27 Aug 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Color LAR codec: a color image representation and compression scheme based on local resolution adjustment and self-extracting region representation

Olivier Déforges, *Member, IEEE,* Marie Babel, *Member, IEEE,* Laurent Bédat, and Joseph Ronsin

***Abstract*— We present an efficient content-based image coding called LAR (Locally Adaptive Resolution) offering advanced scalability at different semantic levels i.e. pixel, block and region. A local analysis of image activity leads to a non-uniform block representation supporting two layers of image description. The first layer provides global information encoded in the spatial domain enabling low bit rate while preserving contours. The second layer holds texture information encoded in the spectral domain enabling scalable bitstream in accordance with the required quality. This basic LAR coding leads to an efficient progressive compression, evaluated through subjective quality tests. Its non-uniform block representation also allows a hierarchical region representation providing higher semantic functionalities. More precisely, the segmentation process can be simultaneously performed at both the coder and the decoder from only the luminance component highly compressed by the first coding layer. This solution provides a representation at a region level while avoiding any contour encoding overhead. Region enhancement can then be realized through the second layer. Furthermore, very high compression of the chromatic components is achieved thanks to this region representation. In this scheme, a low-cost chromatic control, first introduced during the segmentation process, increases the consistency of region representation in terms of color.**

***Index Terms*— Scalable coding, Gray-level and color images segmentation, Region representation based coding, Region of Interest Coding.**

I. Introduction

The main objective when designing an image coding method is to find a solution that is powerful in terms of information compression. However, this feature alone is no longer sufficient for many of the most recent developments. For instance, image and video broadcasting requires scalable compression methods able to adapt the data stream to transmission and receiver capacities. In this context, MPEG4-SVC will soon be adopted as a new standard dedicated to scalable video coding [1]. The scalability referred to here corresponds to three aspects: spatial, resolution and quality [2].

Another type of application refers to remote images and video data base access. To cope with the tremendous amount of available data, automatic or semi-automatic indexing methods are required. The semantic level of the index as defined in MPEG-7 [3] directly depends on the ability of analyzing tools to describe and understand the image content. One natural way of doing this consists of describing the scene in terms of object composition. Relevant solutions have been proposed for this type of representation, such as binary trees by Salembier et al. [4], [5]. To obtain a flexible view with various levels of accuracy, a hierarchical representation is generally used, going from a fine level comprising many regions, to a coarse level comprising only a few objects.

Some unified frameworks try to combine efficient coding with a high, semantic level representation. This concept was first introduced by Kunt *et al.* as a second generation codec [6]. Many other methods have been proposed since then. On the one hand, some of them are designed for image coding [7], [8] and are based on edge representation. On the other hand, a new generation of video coding is object- or region-based [9]–[12]. Thanks to the use of a region shape and a texture description that corresponds to entire objects or to some of their parts inside images, these approaches improve traditional schemes in terms of image information coding.

Regions are defined as convex parts of an image sharing a common feature (motion, textures etc). Objects are defined as entities with a semantic meaning inside an image [13]. For region representation two kinds of information are necessary - **shape** (contours) and **content** (texture). As regards video representation, a third dimension can be added - **motion**.

The region-based approach tends to link digital systems and human vision as regards image processing and perception. This type of approach provides advanced functionalities such as interaction between objects and regions, or scene composition. Another important advantage is the ability, for a given coding scheme, of both increasing compression quality on highly visually sensitive areas of images (**R**egion **O**f **I**nterest) (ROI) and decreasing the compression quality on less significant parts (background) [14]. The actual limited bandpass of channels compared to the data volume required for image transmission leads to a compromise between bit-rate and quality. Once the ROIs are defined and identified, this rate/quality bias can be not only globally but also locally adjusted for each ROI: compression algorithms then introduce only low visual distortions in each ROI, while the image background can be represented with high visual distortions.

Despite the benefits of region-based approaches, current standards are based on traditional information transformation techniques. There are four reasons for this.

1) Shape description, using polygons, produces an information overhead, which can be fairly significant at low

All authors are with the IETR Image Group/INSA Rennes, 20 avenue des Buttes de Cöesmes, CS 14315, 35043 Rennes, France (e-mail: {odeforge,mbabel,lbedat,ronsin}@insa-rennes.fr)

Copyright (c) 2007 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org.

bit rates. To reduce this overhead, we need to limit the number of regions to obtain rudimentary simplified regions. This often leads to a decrease in the accuracy of the shape description. For the same reason, hierarchical representations are generally not supported and are limited to their top levels, as the coding of the structure itself can become prohibitive.

2) Region-based methods mainly preserve the "shape" component and often neglect the "content" component. Consequently, for a given representation, an encoded shape becomes independent of its content.
3) On one hand, region-based representation requires the use of complex image segmentation algorithms. This step generally forms a major obstacle to the achievement of a real time processing system. On the other hand, the use of basic image segmentation algorithms impact strongly on the accuracy of the region description.
4) Common region-based coding schemes authorize the encoder only to define region representation; the decoder does not have any decision-making function. This type of approach cannot, therefore, be used for certain classes of application such as image database browsing, where the operator would define and select his own regions of interest.

Our current work deals with finding a new direction in coding methods, trying to link the aforesaid traditional methods with region-based approaches. We also consider the problem of color image compression as a whole process: the objective is to not duplicate the same scheme for the three color components independently, because this has been shown to be sub-optimal. The paper therefore presents a global approach for both the encoding of color images and region-level representation, i.e. unifying concepts of shape and content.

The next section introduces the Locally Adaptive Resolution (LAR) method as a content-based scalable image codec based on a variable size block representation. It involves two successive main layers: a first layer encodes the global image information at low bit rates, and a second one compresses the local texture. Starting from a coding solution suitable for luminance images, we propose a few adjustments to the processing of chromatic components.

The concept of self-extracting regions is then presented in section III. To gradually enhance the quality of rebuilt images while using scalable coding, the idea is to insert a segmentation stage computed at both the coder and the decoder. This stage uses only first-layer rebuilt images and is efficient because the low bit rate LAR images keep their global content, in particular object contours. A segmentation method is proposed, handling low bit rate luminance images and based on the adjacency graph theory. It leads to an hierarchical region representation at no-cost, as no further information is transmitted to describe regions. ROI coding then enables the second compression layer for the selected regions only. This local enhancement is straightforward in our scheme as the regions and the full LAR codec share the same variable size block representation.

In section IV, we extend the self-extracting region principles to color images. Actually, chromatic information can be used to improve segmentation results. On the other hand, region representation deduced from only the low bit rate luminance LAR image can be used to encode the two chromatic components at a region level. We also investigate a third method which consists of creating a segmentation based mainly on the luminance component and controlled at the coder by additional chromatic information. This approach introduces a low overhead because of the control data transmitted to the decoder. At the same time this solution provides better region representation in terms of color consistency and therefore improves chromatic components compression at a region level.

Finally, a last section is dedicated to conclusions and perspectives.

II. Flat LAR codec presentation

The basic idea for LAR is that the local resolution of an image, i.e. the pixel size, can depend on local activity. On smooth luminance areas, resolution can be lowered. On the other hand, when local activity is high, resolution can be increased. Furthermore, one image I can be considered as a two-component overlay:

$$I = \bar{I} + \underbrace{(I - \bar{I})}_{E} \tag{1}$$

where $\bar{I}$ represents the global image information (typically the local mean value) estimated on a given support, and E represents local variation (local texture) around it. As a result, the dynamic for E depends on two main factors:

1) local activity inside the image,
2) dimension for the support of $\bar{I}$.

Given that an image can be roughly considered as consisting of fairly homogeneous areas and contours, then E has a low dynamic in uniform areas through the adaptation of its support. Inversely, E has a strong dynamic on contours, since support for $\bar{I}$ can be larger than one pixel.

The LAR method is based on a two-layer codec, with a spatial layer for $\bar{I}$ coding and a spectral layer for image error E coding (texture), called respectively flat coder and spectral coder. In this way, the codec naturally offers at least two levels of scalability. Figure 1 shows the overall principle.

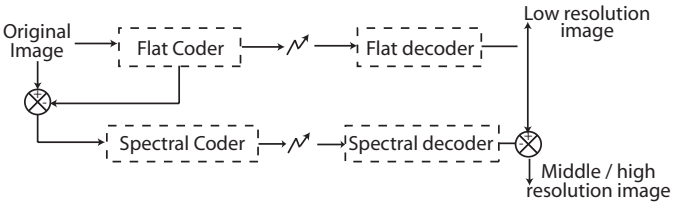


Fig. 1. Overall two-layer LAR coding scheme - flat + spectral coders.

The following sections describe the contents of the different encoded layers. The space selected for color representation is the traditional one for lossy coding, namely Y:Cr:Cb in 4:4:4 format. Various considerations have motivated this choice: decorrelation of information while observing Y:Cr:Cb components, uniformly distributed entropy on chromatic components [15], simplicity of this transformation and, finally, simplicity while using this representation space (linear transformation, integer values).

A. Flat coder

"Spatial" means that the representation and compression process is performed directly in the spatial domain. In order to provide a unique representation and compression of global information in the image, this coder clearly aims at the highest compression rates. On the one hand, it characterizes contours from the rest of the image and, on the other hand, it adapts supports for $\bar{I}$ in such a way that the reconstructed image is subjectively acceptable with a reduced error E in uniform areas. In our case, supports correspond to square blocks.

The flat coding scheme is given in Figure 2. It is based on a process that partitions images into variable-size blocks where each block is rebuilt by its mean luminance value. An arithmetic encoder compresses each symbol (block size, prediction errors, error image etc.) generated by the various representation and coding steps.

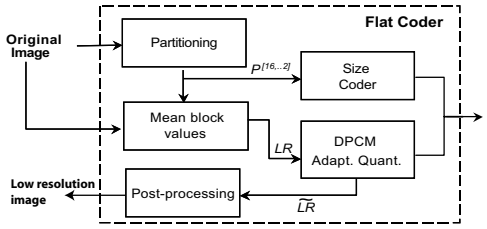


Fig. 2. Principle scheme for flat coder.

The successive steps in the operating technique will be described in the following sections.

1) Partitioning: Systems based on a variable-size block representation rely on a homogeneity criterion and a specific partition. To avoid overlapping, a common partition solution is a Quadtree topology.

The proposed approach involves Quadtree partitioning $P^{[N_{max}\dots N_{min}]}$ with all square blocks having a size equal to a power of two, where N_{max} and N_{min} represent respectively the maximum and minimum authorized block sizes. Thus, the partitioning process consists of first splitting the image into uniform N_{max} square blocks and then building a Quadtree on each block.

In fact, many methods rely on such a variable-size block representation. In particular, MPEG4-AVC/H.264 intra mode authorizes a partition $P^{[16,4]}$ (it splits images into 4×4 or 16×16 blocks), where size selection operates to produce the best bit rate/distortion from a PSNR point of view [16]. Methods based on tree structure operate from the highest level (or maximal size) by cutting nodes down into sons when the homogeneity criterion is not met. Although several homogeneity tests can be found in literature [17], [18], in most cases they rely on computing a L_1 or L_2 norm distance between the block value and the value of its four sons.

Here, we suggest a different criterion, based on edge detection. Among the various possible filters, we opted for a morphological gradient filter (the difference between maximum and minimum luminance values on a given support), because of its fast, recursive implementation and the resulting limitation of the absolute value of texture E (see §II-A.2).

$I(x,y)$ represents later a point with coordinates (x,y) in an image I of size $N_x \times N_y$. Let $I(b^N(i,j))$ be the block $b^N(i,j)$ of size $N \times N$ in I such that:

$$b^N(i,j) = \{(x,y) \in N_x \times N_y \text{ such as } N \times i \leq x < N \times (i+1), \text{ and } N \times j \leq y < N \times (j+1)\}. \quad (2)$$

Let a Quadtree partition be $P^{[N_{max}\dots N_{min}]}$, and $min[I(b^N(i,j))]$ and $max[I(b^N(i,j))]$ be respectively the minimum and maximum values in block $I(b^N(i,j))$.

For each point, the block size is given by:

$$Siz(x,y) = \begin{cases} max(\{N\}) \text{ if } \exists\ N\ \in [N_{max}\dots N_{min}] \\ \quad \text{such as } |max[I(b^N(\lfloor \frac{x}{N} \rfloor, \lfloor \frac{y}{N} \rfloor))] \\ \quad -min[I(b^N(\lfloor \frac{x}{N} \rfloor, \lfloor \frac{y}{N} \rfloor))]| \leq Th \\ N_{min} \text{ otherwise,} \end{cases} \quad (3)$$

where Th represents the homogeneity threshold.

The above image of sizes immediately produces a rough segmenting map for the image, where blocks sized with N_{min} are mainly located on contours and in highly textured areas. Later, we will see that this characteristic forms the basis of the various coding steps.

For color images, the selected solution consists of defining a unique regular partition locally controlled by the minimal size among the three Y:Cr:Cb components. Then, for all the pixels $p(x,y) \in I$, the image of sizes Siz is obtained by

$$Siz(x,y) = min\,[Siz_Y(x,y), Siz_{Cr}(x,y), Siz_{Cb}(x,y)]. \quad (4)$$

Thresholds Th for the luminance component and color components can be independently defined. For a single threshold Th, the minimum is mainly supplied from the Y component. In the remainder of the paper, we have considered a configuration with a single threshold.

2) Mean block values: The flat coder provides a low resolution color image (LR_Y:LR_{Cr}:LR_{Cb}) using the mean block value for each component. For all pixels $p(x,y)$, each LR image component is thus defined by

$$LR(x,y) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{m=0}^{N-1} I(\lfloor \frac{x}{N} \rfloor \times N + k, \lfloor \frac{y}{N} \rfloor \times N + m), \quad (5)$$

where $N = Siz(x,y)$.

As the mean value of each block is naturally included in the range of its minimal and maximal values, one specific property of decomposition is that, for blocks with a size larger than N_{min} (partition $P^{[N_{max}\dots N_{min}[}$), the reconstruction error $E(x,y)$ is bounded by $E(x,y) = |I(x,y) - LR(x,y)| \leq Th$, for all $p(x,y) \in P^{[N_{max}\dots N_{min}[}$.

Therefore, for each image component, entropy error, mean square error and PSNR admit a limit:

$$H(E) \leq \log_2(Th) \text{ bits}, MSE \leq Th^2, PSNR \geq 10 \log \frac{255^2}{Th^2} \text{ dB}. \quad (6)$$

3) Predictive DPCM encoding of mean values: Our Quadtree-like partition associated with the representation of mean block values leads to non-uniform subsampling of the image: uniform areas are then extensively subsampled whereas high activity areas are subject to only light subsampling. In addition to the compression rate introduced by such image subsampling, the global bit rate is reduced while performing the prediction and quantization of block values. These two steps are detailed below.

a) Mean value prediction of luminance block: The mean block luminance value is encoded directly in the spatial domain by a DPCM (Differential Pulse Code Modulation). Subsequently, the encoding process requires only a regular raster scan of the image, and the resulting block representation delivers *a priori* information about activity inside various areas. This can be used for an adaptive prediction.

Our technique is inspired by traditional lossless coding methods. Many of these schemes rely on this kind of predictor to get the best compromise between efficiency and simplicity. More particularly, we have implemented various predictors such as MED (Median Edge predictor) of LOCO-I [19], and DARC (Differential Adaptive Run Coding) proposed in [20]. Finally, we opted for a Graham predictor [21] adapted to fit our context. This adaptation mainly consists of linear prediction on uniform areas, and non-linear prediction on edges, performed at block level. Local gradient drives prediction allowing its optimization in accordance with context.

The predictor follows the relation (7). The estimated value of each block $b^N(i,j)$ is only computed for the top-left pixel $p(x,y)$ where $x = i \times N$ and $y = j \times N$. Then the estimated value $\breve{LR}_Y(x,y)$ is deduced from existing reconstructed values $\tilde{LR}_Y$ obtained after quantization.

$$
\begin{array}{l}
\breve{LR}_Y(x,y) = \\
\begin{cases}
\tilde{LR}_Y(x-1,y) \\
\quad \text{if } |\tilde{LR}_Y(x-1,y-1) - \tilde{LR}_Y(x,y-1)| \\
\quad\quad < |\tilde{LR}_Y(x-1,y-1) - \tilde{LR}_Y(x-1,y)| \\
\quad \text{and if} \\
\quad\quad A_N < |\tilde{LR}_Y(x-1,y-1) - \tilde{LR}_Y(x-1,y)| \\
\tilde{LR}_Y(x,y-1) \\
\quad \text{if } |\tilde{LR}_Y(x-1,y-1) - \tilde{LR}_Y(x-1,y)| \\
\quad\quad < |\tilde{LR}_Y(x-1,y-1) - \tilde{LR}_Y(x,y-1)| \\
\quad \text{and if} \\
\quad\quad A_N < |\tilde{LR}_Y(x-1,y-1) - \tilde{LR}_Y(x,y-1)| \\
(\tilde{LR}_Y(x-1,y) + \tilde{LR}_Y(x,y-1))/2 \quad \text{otherwise.}
\end{cases}
\end{array}
\tag{7}
$$

Parameters A_N grows with N, where $A_1 = 0$, $A_2 = 10$, $A_4 = 20$, $A_8 = 40$ and $A_{16} = 80$. A_N values have been empirically determined. This leads to favor non-linear prediction on small blocks and linear prediction on biggest ones.

b) Quantization of mean block values: Compression techniques based on rate/distortion optimization try to achieve the best compromise between bit rate and global image reconstruction error based on $PSNR$ or MSE. They do not take account of human visual perception. It is experimentally well established that the eye is much less sensitive to luminance and color variations in contour areas (high visual frequencies [22], [23]) than in uniform areas (low visual frequencies). Ricco's law shows that the visual perception threshold for a luminance stimulus inside an area is inversely proportional to the dimension of the area. In other words, visual degradations generated by linear quantization of a block [24] are inversely proportional to its size.

Our coding scheme integrates this principle, performing adapted block size quantization. If q_N represents the quantization step for sized blocks to N, a relation such as $q_N = \frac{q_{N/2}}{2}$ between quantization steps for size N and $N/2$ leads to an almost constant visual quality on the image as a whole.

Let $E_{LR_Y}(x,y)$ be the error prediction, $\hat{E}_{LR_Y}(x,y)$ and $\tilde{E}_{LR_Y}(x,y)$ respectively the quantized and dequantized errors and q_N the applied quantization step for blocks sized to N. This produces the following:

$$
\begin{cases}
E_{LR_Y}(x,y) = LR_Y(x,y) - \breve{LR}_Y(x,y), \\
\hat{E}_{LR_Y}(x,y) = Q\left(E_{LR_Y}(x,y)\right) = round\left[\frac{E_{LR_Y}(x,y)}{q_N}\right], \\
\tilde{E}_{LR_Y}(x,y) = Q^{-1}\left(\hat{E}_{LR_Y}(x,y)\right) = q_N.\hat{E}_{LR_Y}(x,y), \\
\tilde{LR}_Y(x,y) = \breve{LR}_Y(x,y) + \tilde{E}_{LR_Y}(x,y).
\end{cases}
\tag{8}
$$

The whole block is then filled by the reconstructed value.

Quantization steps q_N given in Table I correspond to commonly used values introducing limited distortions.

TABLE I
SIZES AND QUANTIZATION STEPS

Size	16×16	8×8	4×4	2×2	1×1
q_N	2	4	8	16	32

c) Mean value prediction of a color component block: The main advantage of spatial coding is the possibility of using correlations between the three components. Optimizing the mean value prediction of the chromatic component block takes advantage of the first transmitted LR_Y component. This estimation is formalized below. Let $GradMin_Y$ be

$$
\begin{array}{l}
GradMin_Y(x,y) = \\
\quad \min\left[\left|\tilde{LR}_Y(x,y) - \tilde{LR}_Y(x,y-1)\right|, \right. \\
\quad\quad \left|\tilde{LR}_Y(x,y) - \tilde{LR}_Y(x-1,y)\right|, \\
\quad\quad \left.\left|\tilde{LR}_Y(x,y) - \frac{\tilde{LR}_Y(x,y-1) + \tilde{LR}_Y(x-1,y)}{2}\right|\right].
\end{array}
$$

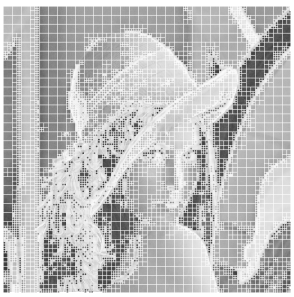
Then we have

$$
\begin{array}{l}
\breve{LR}_{Cr/b}(x,y) = \\
\begin{cases}
\tilde{LR}_{Cr/b}(x,y-1) \\
\quad \text{if } GradMin_Y(x,y) = |\tilde{LR}_Y(x,y) - \tilde{LR}_Y(x,y-1)| \\
\tilde{LR}_{Cr/b}(x-1,y) \\
\quad \text{if } GradMin_Y(x,y) = |\tilde{LR}_Y(x,y) - \tilde{LR}_Y(x-1,y)| \\
\frac{\tilde{LR}_{Cr/b}(x-1,y) + \tilde{LR}_{Cr/b}(x,y-1)}{2} \text{ otherwise.}
\end{cases}
\end{array}
\tag{9}
$$

This prediction optimization gives a significant gain of approximately 20% compared to direct coding.

4) Post-processing: A reconstructed LR image presents specific perceptible blocking effects, mainly in the luminance component. These distortions are much less crucial than the ones produced by methods based on the decomposition of fixed block sizes such as JPEG, MPEG-2 or MPEG-4. These standard methods are content-independent: uniform areas and contours are processed indifferently. In our case, these block effects are due to the non-uniform subsampling which nevertheless preserves the overall content information. This being so, low-complexity post-processing adapted to the variable size block representation can be applied, firstly to smooth uniform areas and secondly to introduce interpolation on edges. Uniform area smoothing is achieved through a linear interpolation adapted to image partitioning. The resultant images are of excellent visual quality in uniform areas (blocks with sizes ranging from 4×4 to 16×16). To achieve edge interpolation (2×2 blocks), we have selected a directional interpolation algorithm designed by D. Muresan [25], based on the optimal adaptive recovery of missing values. This technique, developed by Golomb [26], was initially applied by Shenoy and Parks to interpolation [27]. Figure 3 shows post-processing used to smooth uniform areas while preserving sharp edges [28].

(a) Source image Lena 512×512 - 8 bpp



(b) Quadtree partition: 13888 blocks

(c) Low resolution reconstructed image on $P^{[16...2]}$: 0.2 bpp - Compression ratio: 40, PSNR 30.9 dB

(d) Reconstructed image after post-processing, PSNR 31.4 db

Fig. 3. Results for a partition $P^{[16...2]}$, $Th = 30$.

B. Spectral coder

1) General principles: Error image E (texture) resulting from the flat coder representation is then compressed in a frequency transform space by a second layer, called "spectral coder". The support for E is considered to be the same as the one used for $\bar{I}$, allowing an *a priori* characterization of E, and an adaptation of the coding scheme.

The coding technique is based on an adaptive block size DCT approach where the size is provided by partition $P^{[N_{max}...N_{min}]}$ from the flat coder (see Figure 4). Only AC coefficients need to be transmitted. The first coder layer already supplies mean block values, *i.e.* the DC coefficients.

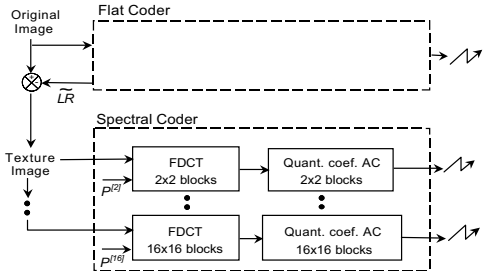


Fig. 4. Principle of a spectral coder with a partition $P^{[16...2]}$.

The major steps in this process are:

- application of the DCT transform to adapt its support to the block size,
- coefficient coding: intra-block zigzag scanning, then encoding non zero values through "run length" (RLC), including specific tags for maximal run length,
- the quantization matrix is adapted to block size.

By definition, the coding scheme is scalable, enabling separate bit stream transmission according to block sizes. Consequently, it is possible to achieve semantic scalability, for instance enhancing only contours, when sending the texture information for N_{min} sized blocks.

2) Energy of fixed/variable size blocks: By construction (equation 6), blocks containing smooth texture present a bounded error, mainly concentrated in the smallest blocks of size $N_{min} \times N_{min}$. Consequently, the mean energy of AC coefficients for partition $P^{[N_{max}...N_{min}[}$ remains lower than the mean energy obtained with traditional approaches using fixed block size (see Figure 5). Moreover, for all blocks in partition $P^{[N_{max}...N_{min}[}$, significant AC coefficients are mainly concentrated in low frequencies. On the other hand, even if the $N_{min} \times N_{min}$ blocks contain AC coefficients with a high dynamic, a rough quantization can be applied without introducing visual distortions.

3) AC Coefficient quantization: Initially we implemented quantization matrices selected from JPEG, by truncating or extrapolating their coefficients in accordance with block size. This solution was not efficient enough, particularly for high bit rate compression because JPEG quantization tables were set up to process low energy and high energy blocks. Another simple quantization law Q was implemented consisting of a linear quantization based on two parameters Q_N and Δ_{Q_N}. A $N \times N$ block contains $2N-1$ diagonals. The k^{th} diagonal of a $N \times N$ block is defined by the set of pixels $\{p(k-i,i)\}$ such as

$$\begin{cases} i \in \{0,\ldots,k\} \text{ for } k \in \{0,\ldots,N-1\} \\ i \in \{k-(N-1),\ldots,N-1\} \text{ for } k \in \{N,\ldots,2N-2\}. \end{cases}$$

For any AC coefficient lying on the k^{th} diagonal, its quantization step is given by: $Q = Q_N + k.\Delta_{Q_N}$.

(a) JPEG: partition $P^{[8]}$

(b) LAR: partition $P^{[16...2]}$

Fig. 5. Energy distribution with fixed/variable size block.

C. Results for the flat LAR codec

1) Image quality: It has been observed that the flat coder is sufficient to encode efficiently chromatic components. A compression format in 4:2:0, where Cr and Cb are subsampled by a factor of 2 in both directions, is equivalent to a partition $P^{[2]}$ for chromatic components in our approach. The extension of this subsampling to a non-uniform partition introduces almost imperceptible distortions.

Figure 6 presents images in which the two chromatic components have been compressed using only the flat coder, while the Y component is still the original.

(a) Lena image, Cr/Cb 16 bpp (b) Block encoded Cr/Cb: 0.063 bpp - PSNR Cr/Cb: 39.1 dB (c) Baboon image, Cr/Cb 16 bpp (d) Block encoded Cr/Cb: 0.226 bpp - PSNR Cr/Cb: 35.6 dB

Fig. 6. Reconstructed images with chrominance component encoding by flat coder (original Y).

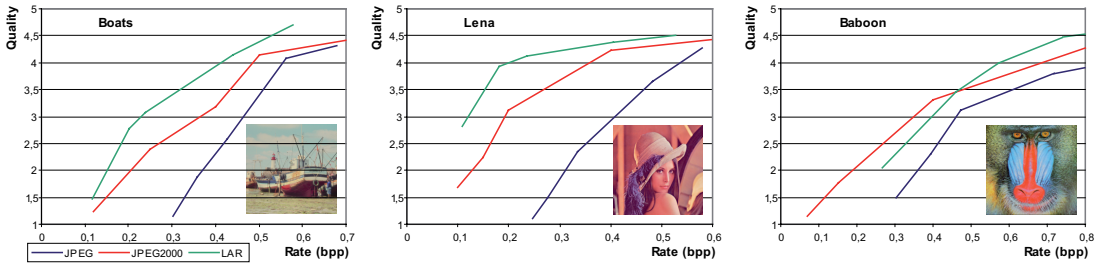


Fig. 7. Comparative test results for visual perception.

From a general point of view, an evaluation of reconstructed images based on bit rate/distortion criteria does not give any indication of the quality of the reconstructed images. Comparative tests of subjective quality have been carried out at the IRRCyN laboratory [29].

This study compared three approaches, namely JPEG, JPEG2000 (codec *ImageXpress*, simple profile) and LAR. Tests were carried out on eight standard images (*Lena, Baboon, Boats, House, Pepper, Fruits, Airplane, Barbara*) compressed at different bit rates. To ensure a rigorous evaluation, the environment was fully standardized in terms of distance from the screen, luminosity, monitor calibration, ambient lighting and color temperature. The elementary protocol for image evaluation was as follows:

1) original image displayed for 6 seconds,
2) uniform grey for 2 seconds,
3) compressed image displayed for 6 seconds,
4) uniform grey for 2 seconds.

Each observer (fourteen in all) was required to grade the observed quality on a scale of one (very bad quality) to five (very good quality). The subjective perception of the LAR method produced higher results in 7 out of 8 series. Figure 7 shows the results obtained for three of these series.

Besides the rate/quality features, these three coders do not have the same characteristics in terms of scalability: in particular, the JPEG selected mode is non-progressive. To obtain the JPEG test image set, each image has to be independently computed, by varying the only parameters, namely the quantization ones. JPEG2000, on the other hand, produces a full embedded bitstream [30] in a single encoding pass. Associated curves reflect continuous quality function. The LAR coder provides an intermediate solution, with scalability achieved by layers namely the spatial and spectral ones. The spectral layer provides also additional levels of semantic scalability. As an example, Figure 8 gives comparative visual results between the three coders.

2) Algorithm complexity: The LAR codec described in this section has low computational complexity. Indeed, block size estimation based on a morphological gradient is implemented with fast recursive erosion and dilation operations. The other main stage of the flat LAR coder, namely the DPCM scheme, is performed at block level. In fact, the flat coder is approximately equivalent to a JPEG coder in terms of operations. Thereby, full compression (flat + spectral) of a 512×512 Y image is performed in 14 ms, running on a 2GHz pentium IV. Finally, as chromatic components encoding requires only the flat coder, the extension of this coding scheme to three components does not multiply system complexity by three.

An alternative has been developed for both the flat and spectral layers. It consists of an original pyramidal decomposition with refined prediction. It supports the highest levels of scalability [31] and enables a fully lossless encoding mode [32]. This solution complies with the self-extracting region representation presented in the next section.

III. Self-extracting region representation at no-cost

In most cases, region representations are available in a decoder only when the coder first performs the segmentation and then transmits the region shapes. To avoid the prohibitive cost of this description, a suitable solution consists of performing the segmentation directly, in both the coder and decoder,

(a) Bike source image, 24 bpp

(b) JPEG encoding, 0.25 bpp, PSNR: 32.75 dB / 35.49 dB / 35.98 dB

(c) LAR encoding, 0.25 bpp, PSNR: 32.58 dB / 36.47 dB / 37.41 dB

(d) JPEG2000 encoding, 0.25 bpp, PSNR: 32.48 dB / 39.47 dB / 40.21 dB

(e) LAR encoding, 0.50 bpp, PSNR: 35.63 dB / 37.73 dB / 38.77 dB

(f) JPEG2000 encoding, 0.50 bpp, PSNR: 35.57 dB / 41.57 dB / 42.92 dB

Fig. 8. Visual quality comparison (PSNR Y/Cr/Cb).

using only a low bit rate compressed image. Once the region representation is built, either the coder or decoder can select regions of interest for enhancement to higher quality. This process fits perfectly with a scalable coding scheme starting with a low-quality image and progressively refining it through successive compressed bitstreams.

A segmentation can be considered on a compressed image whenever distortions introduced by the encoding stage remain limited. At low bit rates, standard methods generate degradations, in particular upon contours, preventing reliable segmentation results. The flat LAR, based on a coherent representation in terms of contours and uniform areas, avoids such damaging degradation. In fact, our approach can be compared to the split and merge segmentation technique based on Quadtree and presented in [33]: the image is first split into homogeneous blocks and then these blocks are merged to build regions. In our case, we can consider that the image has already been split by the flat coder. The segmentation process is then reduced to merge operations. The direct rebuilding of regions from block representation ensures compatibility between shape and region content. This will be used later for ROI enhancement and, in the next section, for chromatic components compression at a region level.

This section deals only with grey-level images (one component) and focusses mainly on the description of the proposed segmentation algorithm based on adjacency graphs. The section ends with the ROI coding issue.

A. Segmentation methods by adjacency graphs

1) Problematic: Let $S = \{(x,y) | 1 \leq x \leq N_x, 1 \leq y \leq N_y\}$ be the spatial coordinates of a pixel in a $N_x \times N_y$ image. The segmentation of the image into K regions consists of finding a partition Δ^K of S such that:

$$S = \bigcup_{k=1}^{K} R_k^K, \tag{10}$$

with $R_i^K \cap R_j^K = \emptyset$, $\forall (i,j) \in \{1 \dots K\}^2$ for $i \neq j$.

Let S^K be the set of regions in partition Δ^K. Starting from an initial partition Δ^{K_0} ($K_0 \leq N_x \times N_y$), the goal of the merge process is to transform Δ^{K_0} into a partition Δ^K ($K < K_0$) complying with homogeneity criteria, through successive region merging.

Partitioning the set of elementary regions S^{K_0} into subsets requires a relationship $\Re$ on S^{K_0}. The subsets then form equivalence classes. In the following, $\left[R_i^{K_0}\right]_{\Re^K}$ denotes a region of S^K inside partition Δ^K, initially associated with region R_i of S^{K_0}.

2) Adjacency graph: Regions must of course create spatially connected sets. The adjacency relation is therefore a key feature for segmentation. Let A_i^K be the set of connected regions at $\left[R_i^{K_0}\right]_{\Re^K}$ in partition Δ^K.

The traditional data structure for partition representation is the Region Adjacency graph (RAG) [34]. The RAG^K of a K-partition is defined as a undirected graph, $G^K = (V, E)$, where $V = \{1, \dots K\}$ is the set of nodes and $E \subset V \times V$ is

the set of edges. Each region is represented by a node and the edge (i, j) exists between two nodes (regions) $R_i^K, R_j^K \in V^2$ if the regions are adjacent.

3) Hierarchical classification and metric: Region merging based on a homogeneity criterion can be considered as a hierarchical classification problem, searching for the most similar elements under a distance D, then measuring clusters between classes with a given criterion. Hierarchical classification can be represented by a tree structure. The hierarchy is said to be *indexed* if, for each set H belonging to hierarchy H, the inclusion relation $H \subset H'$ involves $D(H) \leq D(H')$. A given hierarchical layer of the tree corresponds to one merge between a node and a set of connected nodes.

Classification methods generally use the same process, with merge operations performed two by two and in accordance with a minimal distance criterion [35], [36]. This criterion simply consists of merging the two regions presenting the minimal distance at the current level of hierarchy. Consequently, such an approach creates a binary partition tree for the indexed hierarchical classification [4].

The usefulness of binary partition trees lies in their ability to control exactly the final number of regions. However, this advantage is relative because, to obtain a "correct description" of an image, the number of regions required depends on its complexity.

The major weakness of this approach is mainly its complexity. Even if fast algorithms exist to classify distance based on stack files, these methods remain very time-consuming [35]. Complexity increases when the ultrametric distance (the new distance created by the merging of two regions) is based on a new distance measurement calculated from the merged regions. This new measurement is then used to calculate new minimal distances.

B. Proposed segmentation method

The previous method is based on a minimal distance criterion merging only two regions at a given level of the hierarchy. To solve the complexity problem, we wanted to relax this constraint by allowing several merges at the same time since the distance between regions is less than a given threshold.

To avoid oversegmentation on contours, another suitable characteristic would be as follows: for a given threshold, small regions tend to merge more than large ones. For this purpose, we have introduced the concept of weighted distance. Therefore, contrary to common methods that use symmetrical distances, our approach considers non-symmetrical distances between two regions. Another improvement in the segmentation process relies on the definition of a new distance based on joint mean/gradient criteria.

The segmentation process is performed at block level. In our case Δ^{K_0} is set to $P^{[N_{max}...N_{min}]}$ and S^{K_0} corresponds to the luminance blocks produced from the flat coder. As Δ^{K_0} and S^{K_0} information is available in the decoder, the same segmentation process is performed in both coder and decoder. In the text below, we will show that only this information is required to build a hierarchical region representation, justifying the term "no-cost" representation.

1) Weighted distance: To produce a potential merge based on region size, we have introduced the notion of weighting distances according to the surface areas of regions. If $Cost\left(\left[R_i^{K_0}\right]_{\Re^K}, \left[R_j^{K_0}\right]_{\Re^K}\right)$ defines distance between two classes, the weighted distance is given by

$$\begin{aligned} &Cost'\left(\left[R_i^{K_0}\right]_{\Re^K}, \left[R_j^{K_0}\right]_{\Re^K}\right) = \\ &\quad Cost\left(\left[R_i^{K_0}\right]_{\Re^K}, \left[R_j^{K_0}\right]_{\Re^K}\right) \log_{10}\left(Surf\left(\left[R_i^{K_0}\right]_{\Re^K}\right)\right), \end{aligned} \tag{11}$$

where $Surf\left(\left[R_i^{K_0}\right]_{\Re^K}\right)$ designates the surface of the region $\left[R_i^{K_0}\right]_{\Re^K}$, namely the number of pixels that make up the region. This leads to the following relation:

$$\begin{aligned} &Surf\left(\left[R_i^{K_0}\right]_{\Re^K}\right) > Surf\left(\left[R_j^{K_0}\right]_{\Re^K}\right) \Leftrightarrow \\ &Cost'\left(\left[R_i^{K_0}\right]_{\Re^K}, \left[R_j^{K_0}\right]_{\Re^K}\right) > Cost'\left(\left[R_j^{K_0}\right]_{\Re^K}, \left[R_i^{K_0}\right]_{\Re^K}\right). \end{aligned} \tag{12}$$

A direct effect of these non-symmetrical distances is that RAG is no longer an undirected graph: between two connected nodes there are two edges, each with a specific weight which varies depending on the direction of the adjacency relation.

2) Mean and gradient weighted distances: A traditional distance called $Cost_M\left(\left[R_i^{K_0}\right]_{\Re^{K_N}}, \left[R_j^{K_0}\right]_{\Re^{K_N}}\right)$ is obtained from the differences of the mean value of grey levels in regions and is equal to $\left|\overline{\left[R_i^{K_0}\right]}_{\Re^{K_N}} - \overline{\left[R_j^{K_0}\right]}_{\Re^{K_N}}\right|$, where $\overline{\left[R_i^{K_0}\right]}_{\Re^{K_N}}$ is the mean value of the class $\left[R_i^{K_0}\right]_{\Re^{K_N}}$.

The ultrametric distance is easily updated within the hierarchy, as only the following two characteristics have to remain - surface area and the mean value of a region.

However, a region-merging criterion based solely on these mean values leads to false contouring in uniform areas. To compensate for this, a distance $Cost_{Gr}$ has been added to this distance $Cost_M$.

The distance $Cost_{Gr}$ is based on the measurement of local gradients between two adjacent regions. This local gradient is computed at block level along shared borders, as the mean of the differences in luminance blocks. Thus, estimating the gradient requires consideration of adjacency, not only at the current level of partition, but also at the local level of initial partition Δ^{K_0}. This is particularly simple for a Quadtree partition where common border length between two blocks simply corresponds to the minimal size between these two blocks. Obviously a gradient-based cost function is more difficult to estimate than a mean-based cost function. Nevertheless, this particular solution prevents the need for processing at pixel level and operates only in the data structure associated with the RAG.

The total distance can be expressed as a weighted sum of two distances $Cost_M$ and $Cost_{Gr}$. Our experiments have shown that the best results were obtained when the contribution of each distance was approximately the same. So the total distance has simply been fixed as the mean value between distances $Cost_M$ and $Cost_{Gr}$.

3) Homogeneity criterion: For each scan of the graph, the fast merge algorithm consists of taking, for each region, the one closest to a given distance and merging them if the distance is below a threshold: resulting region label is equal to the lowest label of the merged regions. The process is iterated

until there are no longer any possible merges. The schematic algorithm is as follows:

K_0: initial partition (blocks)
$Nb_{merging} = 0;\ K = K_0;$
Do
  $Nb_{merging_prev} = Nb_{merging}; i = 1;$
  Do
    If $\left[R_i^{K_0}\right]_{\Re^K} \in RAG^K$
     Find $\left[R_j^{K_0}\right]_{\Re^K} \in A_i^K$
      such as $Cost\left(\left[R_i^{K_0}\right]_{\Re^K}, \left[R_j^{K_0}\right]_{\Re^K}\right) \leq$
      $Cost\left(\left[R_i^{K_0}\right]_{\Re^K}, \left[R_l^{K_0}\right]_{\Re^K}\right), \forall \left[R_l^{K_0}\right]_{\Re^K} \in A_i^K$
    End If
    Increment i;
  while $i \leq K_0$;
  $i = 1;$ (13)
  Do
    If $\left[R_i^{K_0}\right]_{\Re^K} \in RAG^K$
     Compute $Cost'\left(\left[R_i^{K_0}\right]_{\Re^K}, \left[R_j^{K_0}\right]_{\Re^K}\right)$
     If $Cost'\left(\left[R_i^{K_0}\right]_{\Re^K}, \left[R_j^{K_0}\right]_{\Re^K}\right) < Th_{Cost}$
        Merge $\left[R_i^{K_0}\right]_{\Re^K}$ and $\left[R_j^{K_0}\right]_{\Re^K}$;
        $K = K - 1$; Increment $Nb_{merging}$;
     End If;
    End If;
    Increment i;
  while $i \leq K_0$;
while $Nb_{merging_prev} < Nb_{merging}$;

Th_{Cost} denotes the tuning parameter providing the level of simplification for image representation.

4) Indexed Hierarchy: The proposed method does not create an indexed hierarchy, because merged regions at a given level of the tree can present higher distances than some merged regions at upper levels. However, by iteratively increasing thresholds for successive partitions, an indexed hierarchy is obtained, with as many indexed levels as thresholds. Typically three thresholds are used, but the choice of levels remains unlimited.

5) Suppressing small components: The non-symmetrical distances prevent over-segmentation on edge areas. Moreover, relative stability is observed in the resulting number of regions, while applying the same segmentation threshold to images with very different complexity. This is because the more complex the image, the more it will generate smaller regions in its initial partition. At the same time, these regions will have a stronger tendency to merge. Nevertheless, in a final step, we use a traditional merge process for remaining small regions. Its only parameter is a surface area value and, using this parameter, each region will merge with the nearest one in terms of distance.

6) Segmentation complexity: The next section will present several examples of segmented images allowing a qualitative evaluation of the method. From the complexity point of view, the merge algorithm (see expression 13) converges rapidly (generally, from 5 to 8 iterations). Calculating distance $Cost_{Gr}$ accounts for almost one-half of total calculating time. The implementation of the algorithm has not yet been optimized. For a 512×512 image with 20000 blocks inside the initial partition, segmentation takes approximately one second on a PC at 2GHz while integrating the concept of gradient distance.

7) ROI coding of local texture: One direct application for self-extracting region representation is found in a coding scheme with local enhancement in regions of interest. From the segmentation map simultaneously available in both coder and decoder, either device can define its own ROI as a set of regions. Thus, an ROI will simply be specified by the labels of its regions. The definition of an ROI is generally performed at the highest level of the segmentation hierarchy (limited number of regions). For an ROI composed of n regions, only n labels are required to fully describe it: this represents a very small number of bytes. The method provides both a *semi-automatic tool for ROI selection*, and *probably the best solution for its concise definition*. Each region, and consequently each ROI, consists of a set of blocks defined in the initial partition. Then *the enhancement of an ROI is straightforward as it merely requires execution of the spectral codec for the validated blocks, i.e. those inside the ROI*. This means that there is immediate, total compatibility between the shape of the ROI and its coding content because the ROI acts as a direct On/Off control for block-level enhancement.

IV. Representation and Coding of Color Images

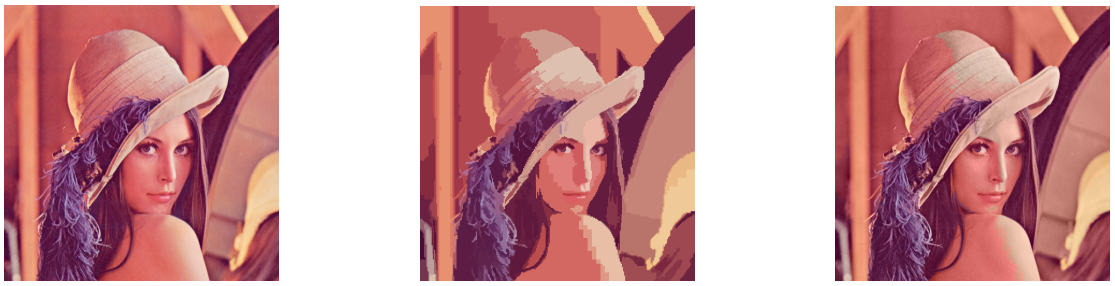
The image representation and coding scheme described above refer only to the luminance component. Turning to the two chromatic components for color images, we aim to improve either the segmentation process by using three component information, or the encoding of the chromatic components by using region representation.

A. Three component image segmentation

Color information produces more reliable segmentation, even if it is more often performed in color space R:G:B or L:a:b [37]. To improve the segmentation results in our case, the most natural solution consists of compressing the three components (Y/Cr/Cb) by the flat coder and sending the corresponding information to the decoder. This means that the segmentation process can be performed with all three components. This operating mode has been verified and provides excellent region descriptions.

B. Region-based coding of color images

Another possible solution in the compression process consists of using the region representation provided by a single component LR_Y to encode the two color components at a region level. The process is implemented as follows: the spatial layer first encodes luminance component Y and the segmentation process produces a region representation in both coder and decoder. The region-based compression of chromatic components then begins by selecting one level of the segmentation hierarchy and considering only one value per region (mean value). For these values, adding a predictive coding step followed by a quantizing step induces a final cost of approximately 4 bits per region. This means that the data volume in bytes for two chromatic components is more or less equal to the number of regions. Figure 9 shows examples of several reconstructed images with the original Y



(a) Lena image, Cr/Cb 16 bpp (b) 300 regions (c) Region coded Cr+Cb: 0.009 bpp, PSNR: 35.79/35.15 dB

Fig. 9. Reconstructed picture with chromatic components encoded through region representation $Th_{Cost} = 50$.

(not encoded) and the Cr/Cb region level encoded. There is a surprising quality/compression rate ratio for the chromatic components and the corresponding coding cost is, at the very most, equivalent to about one hundredth of a bit per pixel, making it non-significant. It is also noticeable that visible impairments are introduced by initial segmenting problems when part of an object attaches itself to another region, or more generally by the merging of several regions (Figure 9). Chromatic components can then be simply improved. All that is required is error encoding at block level through the flat coder, on the whole image or locally inside a given ROI.

C. Segmentation supervised by chromatic control

To take advantage of both region compression for Cr/Cb and the enhancement of quality segmentation using color information, a "chromatic control" principle is defined in an advanced encoding process. The general idea is as follows. The merge segmentation process based on the transmitted low-resolution luminance image is still controlled by the luminance criterion but is also now supervised in the coder by an additional criterion involving chromatic components. This supervises each merge attempt. Clearly, chromatic control generates binary information for each merge attempt and this has to be transmitted. However, since there is high correlation between intensity and chromatic components within an object in an image, the control symbols are of low entropy and the process generates only a low cost. The corresponding color image coding scheme is given in Figure 10.

1) Algorithm description: The merge algorithm or, to be more precise, the search for the nearest region inside the Y picture, remains unchanged. Let $CtrChr\left(\left[R_i^{K_0}\right]_{\Re^K}\right)$ be the binary information transmitted for each merge attempt, and $CoefChrom$ a multiplicative coefficient applied to Th_{Cost}. The merge criterion is then:

If $Cost'\left(\left[R_i^{K_0}\right]_{\Re^K},\left[R_j^{K_0}\right]_{\Re^K}\right) < Th_{Cost}$
  If $CostC'r/b_M\left(\left[R_i^{K_0}\right]_{\Re^K},\left[R_j^{K_0}\right]_{\Re^K}\right) < CoefChrom.Th_{Cost}$
    $CtrChr\left(\left[R_i^{K_0}\right]_{\Re^K},\left[R_j^{K_0}\right]_{\Re^K}\right) = 1$;
    Merge $\left[R_i^{K_0}\right]_{\Re^K}$ and $\left[R_j^{K_0}\right]_{\Re^K}$;
  Else
    $CtrChr\left(\left[R_i^{K_0}\right]_{\Re^K},\left[R_j^{K_0}\right]_{\Re^K}\right) = 0$; Reject $\left[R_j^{K_0}\right]_{\Re^K}$;
  End If
End If

(14)

Chromatic control involves only the mean distance. This is justified in our case as it provides efficient segmentation for Cr/Cb region encoding while limiting algorithm complexity.

$CoefChrom$ is applied as a multiplicative coefficient to Th_{Cost}. Th_{Cost} maintains its role as an overall image simplifying parameter while $CoefChrom$ acts as the adapted chromatic control. From a color point of view, it may be considered as an adjustment cursor located between control cost and segmentation quality:

- if $CoefChrom \to \infty$, then all control values are 1, the associated entropic cost is null, and segmentation only operates from luminance,
- if $CoefChrom = 1$, then the threshold is the same for the three components, and segmentation involves medium chromatic control,
- if $CoefChrom < 1$, the segmentation process closely resembles a direct process applied to the three components, and the control cost increases with the entropy of binary values.

In practical terms, a value of 0.5 gives excellent quality for color segmentation. Even when high-level control is used, the resulting cost remains very low as shown in Tables IV, III and II.

Obviously a merge decision must be accompanied by a mark for a rejected region, to ensure that this region does not remain the best merge candidate based on luminance criterion. Thus, a new graph has been introduced corresponding to a subgraph of RAG; it is called the Region Dissimilarity Graph (RDG). Consequently, the associated set D_i^K supplies the set of rejected candidates for a given class.

2) Results and effects of segmentation parameters: To assess the influence of parameters on Th_{Cost} and $CoefChrom$, we have independently varied these two parameters to observe both segmentation quality and coding cost for color images. The size of "monarch" picture is 768×512 pixels on 24 bits, the number of initial blocks is 23052 for a Y picture compressed by a flat coder at 0.19 bpp, and the cost for direct encoding by the same block layer of components Cr and Cb is 0.125 bpp with an average PSNR of 39.8 dB. Quantitative results are summarized in Tables II, III and IV, and the resultant reconstructed segmented images are shown in Figure 12. Other examples of segmentation are shown in Figure 11.

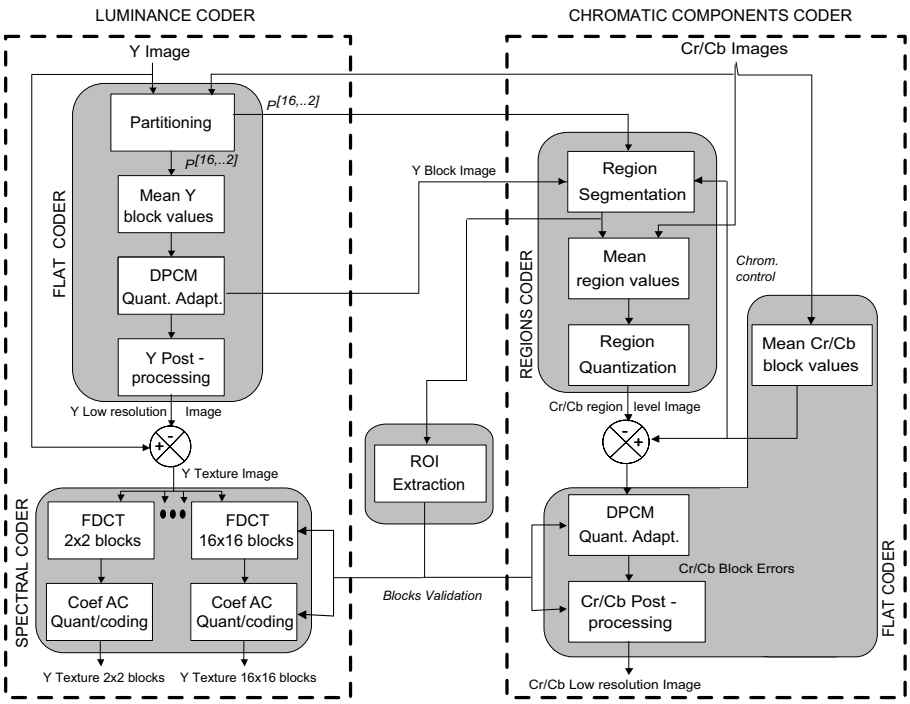


Fig. 10. LAR coder for color image based on supervised segmentation

Fig. 11. Segmentation results on Baboon, Fruits and Lena images ($CoefChrom = 0.5$).

TABLE II
EFFECT OF SEGMENTATION PARAMETERS ON "MONARCH", $Th_{Cost} = 200$.

$\mathbf{Th_{Cost}}$	**Level 3 :** $Th_{Cost} = 200$		
CoefChrom	**2.0**	**1.0**	**0.5**
Nb of regions	30	109	318
Control (bpp)	0.0030	0.0170	0.0596
Cr/Cb regions (bpp)	0.0006	0.0029	0.0085
Total cost contr+reg (bpp)	0.0036	0.0200	0.0681
PSNR Cr/Cb (dB)	29.79	32.84	34.30
Cr/Cb err. flat cod. (bpp)	0.125	0.124	0.115
Cr/Cb total cost blocks level (bpp)	0.129	0.144	0.183

TABLE III
EFFECT OF SEGMENTATION PARAMETERS ON "MONARCH", $Th_{Cost} = 75$.

$\mathbf{Th_{Cost}}$	**Level 3 :** $Th_{Cost} = 75$		
CoefChrom	**2.0**	**1.0**	**0.5**
Nb of regions	83	178	480
Control (bpp)	0.0026	0.0125	0.0480
Cr/Cb regions (bpp)	0.0019	0.0047	0.0128
Total cost contr+reg (bpp)	0.0045	0.0172	0.0615
PSNR Cr/Cb (dB)	30.83	33.35	35.34
Cr/Cb err. flat cod. (bpp)	0.125	0.120	0.112
Cr/Cb total cost blocks level (bpp)	0.130	0.135	0.174

Several remarks can be made about these results.

- Region chromatic components coding is highly compressed,
- The influence of parameter *CoefChrom* on the resultant number of regions is greater when partitioning leads to few regions, thereby increasing the control cost. On the other hand, when a large number of regions are produced, the control cost remains low compared to the cost for

TABLE IV
EFFECT OF SEGMENTATION PARAMETERS ON "MONARCH", $Th_{Cost} = 25$.

$\mathbf{Th_{Cost}}$	**Level 3 :** $Th_{Cost} = 25$		
CoefChrom	**2.0**	**1.0**	**0.5**
Nb of regions	807	909	1345
Control (bpp)	0.0018	0.0071	0.0258
Cr/Cb regions (bpp)	0.0210	0.0241	0.0354
Total cost contr+reg (bpp)	0.0228	0.0212	0.0612
PSNR Cr/Cb (dB)	33.49	35.00	36.62
Cr/Cb err. flat cod. (bpp)	0.115	0.113	0.106
Cr/Cb total cost blocks level (bpp)	0.138	0.144	0.168

chromatic region coding, but there is always a significant gain from a distortion point of view.

- When Chromatic value errors are encoded at block level in partition, i.e. by the flat coder, the additional cost introduced by region level coding is partially overcome by the prediction gain in the block layer.

Figure 13 shows examples of reconstructed images with the original Y component (not encoded) and the Cr/Cb components encoded at region level for various bit rates. It is clear that chromatic coding with a sufficient number of regions enables reconstruction of acceptable quality in a low bit rate situation, avoiding error encoding by the flat coder. The relative stability of the resultant numbers of regions in different images is also noticeable.

D. Mask constraint region coding

The chromatic control technique has been extended to restrict segmentation to any binary shape of a given object. Such a solution is mandatory, for example, when an ROI is defined manually or automatically in the coder (VOP in MPEG-4).

Mask constraint region coding is easily implemented in our scheme. It first consists of integrating the binary mask image into block size estimation in order to produce blocks that are located on each side of the mask boundaries. Segmentation is then performed with shape control preventing the merging of a region located inside an object with a region outside it. The object is finally defined by the set of regions within it. The process is shown in Figure 14, with enhancement applied only to the object.

V. CONCLUSION AND FUTURE PROSPECTS

This paper has presented an original color image coding scheme combining scalable compression and image description at region level. The LAR coder consists of two successive encoding layers, both based on image partitioning into variable-size blocks. The flat coder (first layer) efficiently compresses images at low bit rates but preserves overall information as well as contours. The second layer encodes local texture with respect to the initial partition. This particular structure enables a semantic scalable coding, able to separate flat areas from contours.

In the context of progressive transmission of compressed bitstreams, the principle of self-extracting region representation has been introduced for grey level images. It consists of a segmentation process performed only from highly compressed images in both the coder and the decoder. This solution prevents costly transmission of the segmentation map to provide the region shapes. An original segmentation algorithm has been then proposed, leading to an efficient hierarchical region-based description of the image. A region is built as a set of blocks among the initial partition. This means that one important aspect of our approach is that it ensures full compliance between the two layers in the image coding and the shape of the regions. One direct issue is ROI coding: an ROI is rapidly and easily defined as a set of regions in either the coder or the decoder. Local image quality enhancement is then achieved by allowing only for the selected blocks in the second encoding layer.

The extension of the previous method to color images has two direct purposes: to improve segmentation when considering chromatic information, or to encode chromatic components at region level. We have also proposed a mixed solution combining these two advantages. The supervision of the segmentation process, driven by the luminance component in accordance with a chromatic consistency criterion, is designed to obtain an accurate region description with regard to color. These regions represent a reliable support for encoding chromatic components at region level. Actually, this method provides unequalled rates for chromatic components compression. Moreover, mask constraint region coding is an available extension of our scheme. Segmentation is then driven by the additional shape information, so that the associated object is made up of the set of regions within it.

Current work concerns the extension of this region-based LAR method to highly scalable video coding. We also intend to adapt the solution to the future video standard, i.e. MPEG-4 SVC.

REFERENCES

[1] J. Reichel, H. Schwarz, and M. Wien, "Joint Scalable Video Model JSVM-4," *ISO/IEC JTC1/SC29/WG11, JVT-Q006, Nice, France*, October 2005.

[2] H. Schwarz, T. Hinz, H. Kirchhoffer, D. Marpe, and T. Wiegand, "Technical description of the HHI proposal for SVC CE1," *ISO/IEC JTC1/SC29/WG11, MPEG04/M11244, Palma de Mallorca, Spain*, October 2004.

[3] M.-. R. Group, *MPEG-7: Context, Objectives and Technical Roadmap, V.12*, ISO/IEC JTC 1/SC29/WG11 MPEG99/N2861, D-Lib Magazine, Vancouver, July 1999.

[4] P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval," *IEEE Transaction on Image Processing*, vol. 9, pp. 561–576, January 2000.

[5] C. Ferran and J. Casas, "Object representation using colour, shape and structure criteria in a Binary Partition Tree," in *IEEE International Conference On Image Processing, ICIP'05*, vol. III, Genova, September 2005, pp. 1144–1147.

[6] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second Generation Image Coding Techniques," *Proceedings of the IEEE*, vol. 73, no. 4, pp. 549–575, April 1985.

[7] X. Ran and N. Farvardin, "A perceptually motivated three-component image model–part ii: Application to image compression," *IEEE Transactions on Image Processing*, vol. 4, no. 4, pp. 430–447, 1995.

[8] M. Wakin, J. Romberg, H. Choi, and R. Baraniuk, "Image Compression using an Efficient Edge Cartoon + Texture Model," in *Data Compression Conference*, Snowbird, UT, April 2002, pp. 43–52.
[9] L. Torres and M. Kunt, *Video Coding : The Second Generation Approach*. Kluwer Academic, 1996.
[10] M. Chaumont, N. Camas, and S. Pateux, "Fully scalable object based video coder based on analysis-synthesis scheme," in *IEEE International Conference On Image Processing, ICIP'2003*, vol. 2, Barcelona, September 2003, pp. 69–72.
[11] M. Chaumont, S. Pateux, and H. Nicolas, "Object-based video coding using a dynamic coding approach," in *IEEE International Conference On Image Processing, ICIP'2004*, vol. 2, Singapore, September 2004, pp. 1105–1108.
[12] Y. Wo, G.-Q. Han, and D.-F. G. J.-W. Zhang, "A new segment method for segment-based image coding," in *Fourth International Conference on Machine Learning and Cybernetics*, vol. 9, Guangzhou, August 2005, pp. 5376–5381.
[13] T. Ebrahimi and M. Kunt, "Visual Data Compression for Multimedia Applications," *Proceedings of the IEEE*, vol. 86, no. 6, pp. 1109–1125, June 1998.
[14] N. Peterfreund and Y. Zeevi, "Nonuniform Image Representation in Area-Of-Interest Systems," *Trans. on Image Processing*, vol. 4, no. 9, pp. 1202–1212, September 1995.
[15] B. Deknuydt, J. Smolders, L. V. Eycken, and A. Oosterlinck, "Color Space Choice for Nearly Reversible Image Compression," in *SPIE, Visual Communications and Image Processing '92*, vol. 1818, Boston, November 1992, pp. 1300–1311.
[16] H264 MPEG-4 10 AVC, *Joint Committee Draft (CD)*, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEGn 3rd Meeting: Fairfax, Virginia, USA, May 2002.
[17] C. Shaffer and H. Samet, "Optimal Quadtree Construction Algorithms," *Computer Vision, Graphics, Image processing*, vol. 37, no. 3, pp. 402–419, March 1987.
[18] P. Strobach, "Tree-Structured Scene Adaptive Coder," *IEEE Trans. on Communication*, vol. 38, no. 4, pp. 477–486, April 1990.
[19] M. Weinberger, G. Seroussi, and G. Sapiro, "LOCO-I : a low complexity, context-based, lossless image compression algorithm," in *IEEE Data Compression Conference*, March 1996, pp. 141–150.
[20] B. Gandhi *et al.*, "Differential Adaptive Run Coding," ISO Working Document ISO/IEC JTC1/SC29/WG1 N204, Tech. Rep., 1995.
[21] R. Graham, "Predictive Quantizing of Television Signals," *IREWESCON Convention Record*, vol. 22, no. 4, pp. 147–157, August 1958.
[22] M. A. Losada and K. Mullen, "The Spatial Tuning of Chromatic Mechanisms Identified by Simultaneous Masking," *Vision Research*, vol. 34, no. 3, pp. 331–341, February 1994.
[23] K. D. Valois, M. Webster, and E. Switkes, "Orientation and Spatial-Frequency Discrimination for Luminance and Chromatic Gratings," *Journal of the Optical Society of America*, vol. 7, no. 6, pp. 1034–1049, June 1990.
[24] L. Bédat, A. Saadane, and D. Barba, "Masking Effects of Perceptual Colour Components on Achromatic Grating," in *ECVP 97*, Helsinki, August 1997.
[25] D. Muresan and T. Parks, "Optimal Recovery Approach to Image Interpolation," in *IEEE International Conference on Image Proceedings, ICIP'01*, vol. 3, Thessaloniki, Greece, October 2001, pp. 848–851.
[26] M. Golomb and H. Weinberger, *On Numerical Approximation*. R. E. Langer ed., The University of Wisconsin Press, 1959, ch. Optimal Approximation and Error Bounds, pp. 117–190.
[27] R. Shenoy and T. Parks, "An Optimal Recovery Approach to Interpolation," *IEEE Trans. Signal Processing*, vol. 40, no. 8, pp. 1987–1996, August 1992.
[28] D. Muresan, "Review of Optimal Recovery," DMMD, Tech. Rep. TR-2002-10, 2002.
[29] P. L. Callet and D. Barba, "Perceptual color image quality metric using adequate error pooling for coding scheme evaluation," in *SPIE Human Vision and Electronic Imaging Conference*, vol. 4662, San Jose, USA, January 2002, pp. 173–180.
[30] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards, and Practice*. Kluwer Academic Publishers, 2001.
[31] M. Babel, O. Déforges, and J. Ronsin, "Adaptive Multi-Resolution Scheme for Efficient Image Compression," in *Picture Coding Symposium, PCS'03*, Saint-Malo, France, April 23-25 2003.
[32] ——, "Interleaved S+P Pyramidal Decomposition with Refined Prediction Model," in *IEEE International Conference on Image Processing, ICIP'05*, vol. 2, Genova,Italy, Septembre 2005, pp. 750–753.
[33] P. Willemin, T. Reed, and M. Kunt, "Image Sequence Coding by Split and Merge," *IEEE Trans. on Communication*, vol. 39, no. 12, pp. 1845–1855, December 1991.
[34] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*. Chapman & Hall, 1993.
[35] K. Haris, S. Efstratiadis, N. Maglaveras, and A. Katsaggelos, "Hybrid Image Segmentation Using Watersheds And Fast Region Merging," *IEEE Trans. on Image Processing*, vol. 7, no. 12, pp. 1684–1699, December 1998.
[36] J.-M. Beaulieu and M. Goldberg, "Hierarchy in Picture Segmentation: a Stepwise Optimization Approach," *IEEE Trans. on Patern Analysis and Machine Intelligence*, vol. 11, no. 2, pp. 150–163, February 1989.
[37] H. Cheng, X. Jiang, Y. Sun, and J. Wang, "Color image segmentation: advances and prospects," *Pattern Recognition*, vol. 34, no. 12, pp. 2259–2281, December 2001.

Olivier Déforges Olivier Déforges is a professor at National Institute of Applied Sciences of Rennes (INSA). He received a Ph.D. degree in image processing in 1995. In 1996, he joined the Department of Electronic Engineering at the INSA of Rennes, Scientific and Technical University. He is a member of the Institute of Electronics and Telecommunications of Rennes (IETR), UMR CNRS 6164. His principal research interests are image and video lossy and lossless compression, image understanding, fast prototyping, and parallel architectures.

Marie Babel Marie Babel is an associate professor at the National Institute of Applied Sciences (INSA) of Rennes.She graduated in Electronic Engineering from the INSA of Rennes and in Signal and Image Processing from the University of Rennes 1. She received a Ph.D. degree in image processing in 2005. She is a member of the IETR Laboratory. Her principal research interests include image and video lossy and lossless coding, advanced video schemes and image transmission securization processes.

Laurent Bédat Laurent Bédat received the M.S. degree in signal and image processing from the University of Rennes in 1993, and the Ph.D. degree in 1998 from the University of Nantes. Since 1999, he has been an Assistant Professor at the National Institute for the Applied Sciences, Rennes, France. He is a member of the image processing group of IETR Laboratory. His current activities in education concern analog electronics, Microcontroller, operating system and networks.

Joseph Ronsin Joseph RONSIN is Professor in the Department of Electronic and Computer engineering of Institut National des Sciences Appliquées (INSA - National Institute of applied sciences) and is co-responsible for research of Image and Remote sensing group of Institute of Electronics and Telecommunications of Rennes (www.ietr.org). The unit is linked to French National Center for Scientific Research (CNRS).

Fig. 12. Segmentation results based on parameters Th_{Cost} and $CoefChrom$ on Monarch image.

(a) Original Cr+Cb: 16 bpp

(b) Cr+Cb coded with flat coder: 0.125 bpp, compression ratio $\tau = 128$

(c) Cr+Cb coded with 1345 regions, $Th_{Cost} = 25$, $CoefChrom = 0.5$: 0.0612 bpp, $\tau = 261$

(d) Cr+Cb coded with 807 regions, $Th_{Cost} = 25$, $CoefChrom = 2.0$: 0.0228 bpp, $\tau = 702$

(e) Cr+Cb coded with 178 regions, $Th_{Cost} = 75$, $CoefChrom = 1.0$: 0.0172 bpp, $\tau = 930$

(f) Cr+Cb coded with 30 regions, $Th_{Cost} = 200$, $CoefChrom = 2$: 0.0036 bpp, $\tau = 4444$

Fig. 13. Chromatic components encoding by region representation on Monarch image.

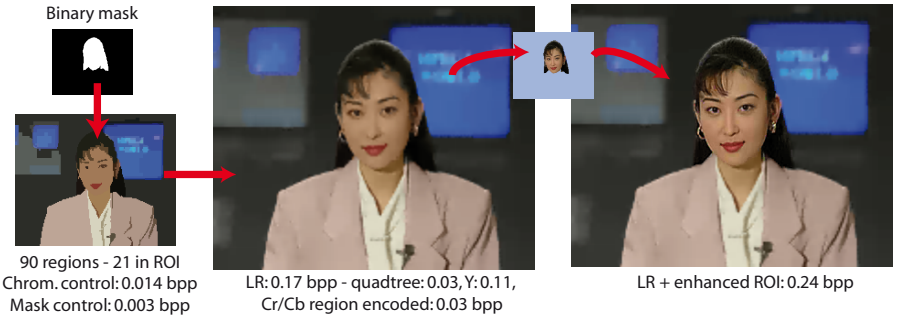


Fig. 14. ROI encoding on Akiyo with chromatic and object mask constraints.