Embedding Multilevel Image Encryption in the LAR Codec

Jean Motsch, Olivier Déforges, Marie Babel

EMBEDDING MULTILEVEL IMAGE ENCRYPTION IN THE LAR CODEC

Jean MOTSCH

CREC Saint-Cyr/LESTP, France
IETR UMR CNRS 6164
Image and Remote Sensing Group
jean.motsch@st-cyr.terre.defense.gouv.fr

Olivier DÉFORGES, Marie BABEL

INSA RENNES, France
IETR UMR CNRS 6164
Image and Remote Sensing Group
{Olivier.Deforges, Marie.Babel}@insa-rennes.fr

ABSTRACT

A still image codec should not only to be good from compression point of view, it might also provide services. The LAR *interleaved S+P* codec allows loss-less to lossy coding, presenting both resolution and distortion scalability. The proposed method brings image encryption at no cost in the LAR framework. It exploits the embedded quadtree decomposition of the LAR to provide multilevel protection to the whole bit-stream. Theoretical aspects are considered and experimental results show the effectiveness of the process.

1. INTRODUCTION

Evolution in still image codecs shows the need for new embedded services. The JPEG2000 standard [1] introduces some features that are among the ones requested for a good codec:

- error resilience for transmission in noisy environments, such as wireless and the Internet,
- region of interest coding,
- scalability, both in resolution and distortion,
- and, of course, good distortion-bit–rate performances.

Data protection, through encryption for example, is also a need. Good codecs should provide not only the best distortion–bit-rates performance, but also include the aforementioned services.

This paper deals with embedded image protection. Whereas most methods relies on coding whole or part of the image, using block or stream ciphering, we propose a method based on using the quadtree decomposition bit-stream as a way to protect the content of the image. The main idea is to transmit the data without the quadtree decomposition, using the quadtree as the key to decrypt the image. Figure 1 gives a rough idea of the control that can be done on the quadtree bit-stream. As the quadtree bit-stream is multilevel, the authorization system manages different levels of quality for the same original bit-stream. As an example, low resolution of the picture is always provided to take the decision of buying the picture while highest resolution is furnished after payment. As for most data encryption method, one way to confirm the usefulness of this method is to show that a brute force attack on the bit-stream is too time consuming.

This paper is organized as follows. Section 2 presents shortly the context of the LAR framework and its relationship to quadtree. Section 3 emphasizes on some theoretical results on quadtree partitioning and their coding cost. Section 4 shows some experimental results and comments, and section 5 does conclude the topic.

This work is partially supported by French Ministry of Research and Technology through ANR Project TSAR

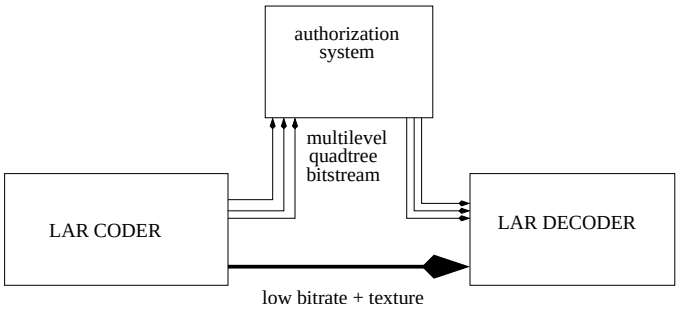


Figure 1: Quadtree bit-stream control

2. LAR FRAMEWORK

2.1. LAR codec

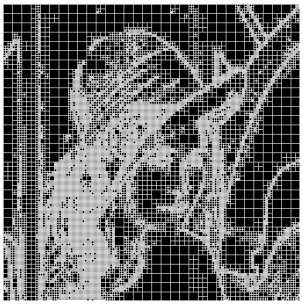
The LAR (Locally Adaptive Resolution), based on a variable block-size decomposition, leads to an efficient lossy image compression technique [2].

The LAR compression method – initially designed for lossy grey-scale image coding – is a two layer codec : a spatial coder and a complementary spectral one. The spatial coder provides a low bit-rate compressed image whereas the spectral coder codes the texture. Self extraction of regions of interest is also possible [2]. Extensions also enables middle and high quality coding and region-level based chromatic image coding [3]. Furthermore, subjective quality image evaluation show that the low resolution LAR picture is better than the same bit-rate JPEG2000 coded picture [2].

A minimal-redundancy pyramidal decomposition, the LAR-APP introduces a multiresolution framework [4]. Finally, an *interleaved S+P* pyramidal decomposition with a refined prediction model shows state-of-the-art results combined to resolution and SNR scalability [5].

The basic principle of the spatial coder is that the local resolution, *i.e.* the pixel size, depends on the local activity. This leads to build a resolution variable image using a quadtree structure. The pixel's size (*e.g.* from 32×32 to 2×2) is computed thanks to a local morphological gradient. A single parameter, the threshold on the contrast, drives the resolution process. Figure 2 presents an example of quadtree decomposition. Through this decomposition, the pixel size provides implicitly the block's nature: small blocks are located on contours whereas large ones describes smooth areas. An efficient post-processing [6] removes perceptible blocks artifacts.

(a) Original picture



(b) LAR quadtree

Figure 2: Example of LAR quadtree decomposition with threshold equals to 30

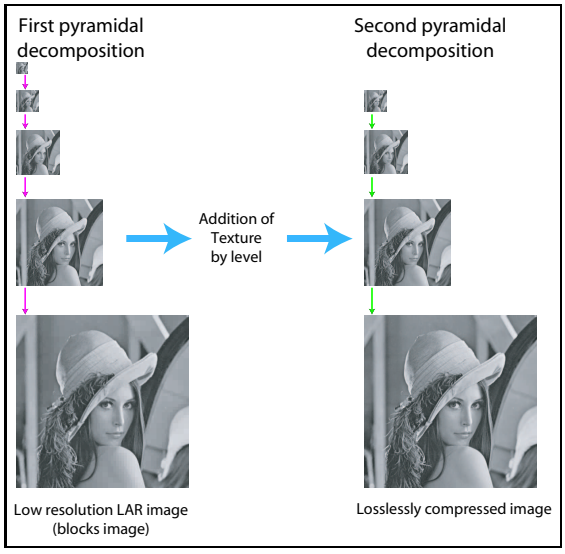


Figure 3: Pyramidal decomposition in LAR *interleaved S+P*

2.2. LAR Interleaved S+P bit-stream

The LAR *interleaved S+P* codec is a two passes codec:

1. first pass constructs a low bit-rate, low resolution and high distortion picture. It plays the role of the spatial coder and it can be stopped at every level of the pyramid.

2. second pass, equivalent to the spectral coder, provides the texture at every level of the pyramid.

Figure 3 shows the pyramidal decomposition on a picture. This two passes process allows to refine the picture and provides both resolution and distortion scalability.

In order to reflect the pyramidal decomposition, the bit-stream is organized in sub-streams, one for each level of the pyramid. There are three different type of sub-streams: the quadtree's refinement, the refinement on the low resolution image (first pass) and the local texture values. Figure 4 gives a basic idea of the LAR *interleaved S+P* bit-stream.

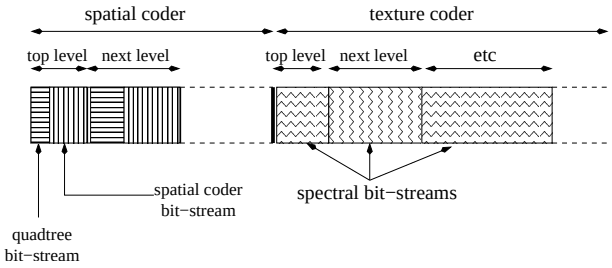


Figure 4: LAR *interleaved S+P* bit-stream description

2.3. Error on the quadtree

The LAR *interleaved S+P* shows to be reactive on errors made on its embedded quadtree decomposition. Figure 5 illustrates two important properties of the codec for data encryption. Firstly, without a perfect knowledge of the quadtree, the reconstructed picture is far from the original one. Secondly, guessing the top level quadtree decomposition doesn't help obtaining a good quality picture.

(a) An error at the first block of level 4 (16×16blocks)

(b) Assuming a uniform quadtree $QP^{[16..4]}$

Figure 5: LAR *interleaved S+P* reconstructed pictures

3. QUADTREE REPRESENTATION: THEORETICAL STUDY

3.1. On the number of quadtree partitions

We consider an image I having M lines and N columns with $S = MN$ pixels. Every pixel is coded using q bits and therefore take $Q = 2^q$ different values.

$\text{QP}^{[L_T..L_B]}(I)$ is a quadtree partition of the image I using blocks size ranging from $L_T \times L_T$ for the top level to $L_B \times L_B$ at the bottom. $\left|\text{QP}^{[L_T..L_B]}(I)\right|$ is the number of different quadtree partitions possible for image I.

Let $l_T = log_2 L_T$, $l_B = log_2 L_B$ and $l = l_T - l_B + 1$ the number of levels in the quadtree partition. We note $\Omega^{[L_T..L_B]}$ the number of different quadtree possible on a $L_T \times L_T$ block. $\omega^{[L_T..L_B]} = log_2 \Omega^{[L_T..L_B]}$ is the number of bits needed to code this quadtree partition.

For sake of illustration, we will consider a 256×256 8 bits picture using a $\text{QP}^{[32..2]}$ partition with 5 levels.

The number of quadtree partition $\Omega^{[L_T..L_B]}$ is related to the following recursively defined function Φ:

$$\Phi(0) = 1 \quad (1)$$

$$\Phi(n) = 1 + \Phi^4(n-1) \quad (2)$$

In fact, $\Omega^{[L_T..L_B]} = \Phi(l-1)$. Table 1 gives the first values of $\Phi(n)$. Notice that $\log_2 \Phi(n)$ is the number of bits required to code the value of $\Phi(n)$. For $n > 0$, $\Phi(n)$ can be roughly approximated by $\Phi_a(n) = 2^{(4^{n-1})}$. This establishes the quadtree coding cost $C(\text{QP}^{[L_T..L_B]})$ to about 4^{l-2} bits per square $L_T \times L_T$ block.

n	$\Phi(n)$	$\log_2 \Phi(n)$	$\Phi_a(n)$
0	1	0	1
1	2	1	2
2	17	4.09	2^4
3	83522	16.35	2^{16}
4	4.86×10^{19}	65.40	2^{64}

Table 1: $\Phi(n)$ and $\Phi_a(n)$ values for small n

For our example, $\Omega^{[32..2]} = 4.86 \times 10^{19}$ and we need up to 65.40 bits to code the quadtree partition on an 32×32 block.

The number of quadtree partition for the whole image I is simply computed considering that every $L_T \times L_T$ block contains a $\text{QP}^{[L_T..L_B]}$ partition. $\left|\text{QP}^{[L_T..L_B]}(I)\right|$ is then given by:

$$\left|\text{QP}^{[L_T..L_B]}(I)\right| = \Phi(l-1)^{\frac{MN}{L_T^2}} \quad (3)$$

Usually, coding cost are expressed in bit per pixel (bpp). This cost is given by:

$$\frac{\log_2 \Phi(l-1)}{L_T^2} \quad (4)$$

Following with our example, $\left|\text{QP}^{[32..2]}(I)\right| \approx 2^{4186}$. It also means that the maximum coding cost will be 4186 bits or 0.064 bpp.

3.2. Quadtree coding

Coding of quadtree partitions is computed using a progressive method. A one is transmitted for every block that has to be splitted in 4, while a zero is transmitted otherwise. And this is done recursively. Figure 6 shows an example of quadtree. The quadtree partitions are $\text{QP}^{[8..2]}$. The bit-stream to code the first level is 0111. The second, and last, level, is coded with 011010010010. The bit-stream is 16 bits long, or 0.0625 bpp. In fact, there are $\Phi 2^4 = 83521$ possible quadtree partitions that need at most 16.35 bits to be coded, or 0.0634 bpp. This is the upper bound of the coding cost.

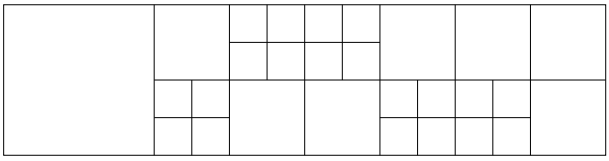


Figure 6: $\text{QP}^{[8..2]}$ on a 8×32 image

3.3. Remarks

Section 3.1 demonstrates the huge number of quadtree partitions that are possible in coding an image. However, two important remarks have to be presented.

Firstly, the result shown in equation (3) relies on the hypothesis that blocks are independent in the image. This assumption implies that the coding cost of the quadtree given by equation (4) is a maximum coding cost. If there is some correlation among blocks, the entropy of the generated bit-stream will be less than the coding cost. An entropic coder will then be able to reduce the size of the bit-stream canceling the remaining redundancy. Experimental results of section 4 show that the entropy after entropic coding is still important.

Secondly, the space of all images is still order of magnitudes bigger than the space of partitions. In the example we used, every pixel is coded with 8 bits but the maximum quadtree coding cost is 0.064 bit. That explains that a quadtree cannot be used as a unique identifier of an image. Indeed, numerous images share the same quadtree.

4. EXPERIMENTATION

Experiments are conducted using pictures lena, airplane, baboon, goldhill, man, pepper and woman, all of size 512×512 and 8 bits coded. Quadtree decomposition are $\text{QP}^{[64..2]}$ *i.e.* 6 levels in the pyramid. In that case, the maximum quadtree entropy is 0.06387 bpp.

Figure 7 shows the influence of the threshold on the entropy of the quadtree bit-stream. The threshold drives the process of quadtree decomposition. That is, as the threshold increases, the number of regions decreases, with bigger blocks. Usually, the threshold has a fixed value of 30 (on a 256 greyscale). Notice that the quadtree entropy remains above 0.03 bpp, about half its maximum value.

If the quadtree entropy is 0.03 bpp, for a 512×512 picture, the quadtree entropy is about 7864 bits for the whole image. That means, the number of possible quadtrees is about $2^{7864} \approx 10^{2367}$. Even if decoding a bit-stream takes 1 µs, then the brute force approach will be largely too long.

Conversely, if the protection of the picture may stay for 100 years, that means, $3.156 \times 10^9 s$, the entropy of the quadtree must be only more than 51.5 bits.

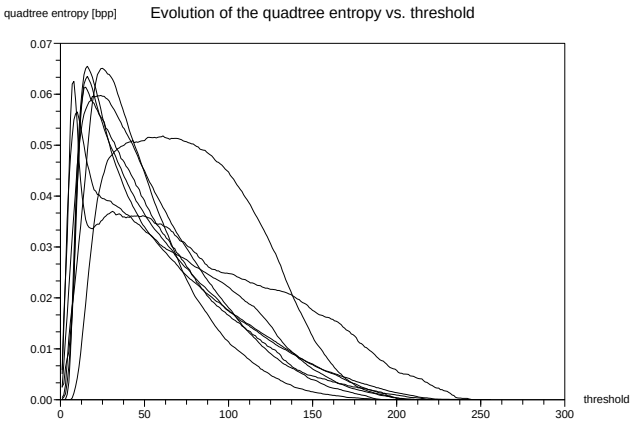


Figure 7: Quadtree entropy vs. threshold

Figure 8 shows results on investigating on the quadtree entropy as a function of the number of blocks. This figure is shaped like the $h_2(p) = -p\log_2(p) - (1-p)\log_2(1-p)$. It exhibits the fact that the quadtree entropy is maximum approximately when the number of blocks is half the maximum number of possible blocks.

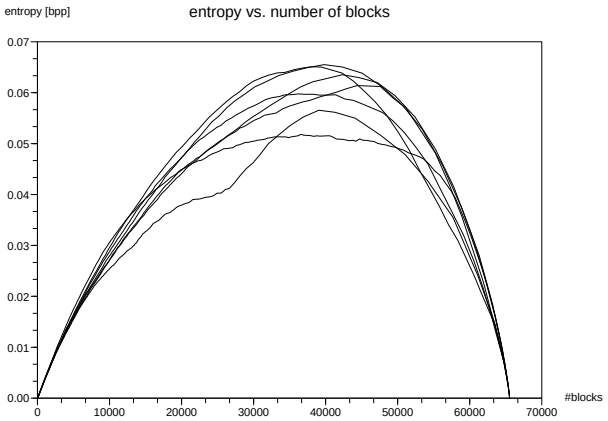


Figure 8: Quadtree entropy vs. number of blocks

As the quadtree is progressively transmitted in the bit-stream, the entropy of each level of the quadtree coding is computed and table 2 shows the results obtained with threshold value equal to 30. The entropy of first level is null when there is no 64×64 blocks in the quadtree. The second level has an entropy about 50 that might be sufficient to provide a reasonable data protection. The entropy increases as the block's size decreases, making the reconstruction without the partition more and more difficult. Notice also that for image airplane, the 17 bits of entropy are not sufficient. In all case, the top level of the pyramid might be decoded, furnishing only a very low resolution picture with high distortion.

image	64×64	32×32	16×16	8×8	4×4
lena	0	61	656	2916	8886
airplane	17	175	601	1922	7243
baboon	0	0	21	1647	10356
goldhill	0	65	218	2659	12218
man	0	35	466	2824	11014
pepper	0	41	667	3148	8245
woman	0	86	560	2810	9483

Table 2: Entropy of quadtree levels given in [bits] for various 512×512 images

5. CONCLUSION AND PERSPECTIVES

This paper presented an original image encryption method with following properties: embedded in the original bit-stream at no cost, allowing multilevel access authorization combined with state-of-the-art still picture codec. Multilevel quadtree decomposition provides a way to select the quality of the picture decoded. Theoretical and experimental results shows the effectiveness of this tool on an information theory point of view.

One main goal of this work is the implementation of an archiving system for high resolution art pictures of the Louvre's museum. This digital library will provide a selective access with different quality of images.

6. REFERENCES

[1] D. Taubman and S. Marcellin, *Jpeg2000: Image Compression Fundamentals, Standards, and Practice*, Kluwer Academic Publishers, 2002.

[2] O. Deforges and J. Ronsin, "Region of interest coding for low bit-rate image transmission," in *ICME*, july 2000, vol. 1, pp. 107–110.

[3] O. Deforges and J. Ronsin, "Supervised segmentation at low bit rates for region representation and color image compression," in *ICME*, 2002, vol. 1, pp. 665–668.

[4] M. Babel, O. Deforges, and J. Ronsin, "Lossless and lossy minimal redundancy pyramidal decomposition for scalable image compression technique," in *ICME*, 2003, vol. 3, pp. 249–252.

[5] M. Babel, O. Deforges, and J. Ronsin, "Interleaved s+p pyramidal decomposition with refined prediction model," in *ICIP*, 2005, vol. 2, pp. 750–753.

[6] D. Muresan and T. Parks, "Optimal recovery approach to image interpolation," in *ICIP*, 2001, vol. 3, pp. 848–851.