Joint Lossless Coding and Reversible Data Embedding in a Multiresolution Still Image Coder

Jean Motsch, Marie Babel, Olivier Déforges

JOINT LOSSLESS CODING AND REVERSIBLE DATA EMBEDDING IN A MULTIRESOLUTION STILL IMAGE CODER

J. MOTSCH[1]*, M. BABEL*[2]*, and O. DÉFORGES*[2]

[1]CREC Saint-Cyr/LESTP, France [2]INSA RENNES, France
IETR UMR CNRS 6164
Image and Remote Sensing Group
jean.motsch@st-cyr.terre.defense.gouv.fr, {Olivier.Deforges, Marie.Babel}@insa-rennes.fr

ABSTRACT

Modern still image codecs furnish more than just good distortion-rate performances. They must also provide some services. Scalability in resolution and quality, error resilience and embedded bitstreams were among the first one to be available. There is still room for enhancement, especially when it comes to security-oriented features. Data embedding is necessary, as for inserting metadata, or to copyright a picture. We present the use of a very simple reversible data embedding method in a multiresolution still image codec framework. Experimental results show the usefulness of such an adequation of techniques from different domain. Moreover, the embedding overhead is evaluated and shows to be very acceptable.

1. INTRODUCTION

Lossless image coding is still useful for archiving and medical applications. In this area, coding frameworks have to fulfill several goals: dealing with huge amount of data, providing efficient lossless compression performance, and furnishing new services. These services include content protection, secure transmission [1], and metadata insertion. Most of the time, services are provided independently of the coding scheme, introducing some overhead cost.

Data embedding hides data (*i.e.* the *payload*) in a digital picture. This must be as unnoticeable as possible. For that purpose, image quality should be high after data embedding. Measurement of data embedding algorithm performances is done using three criteria: first, the payload capacity limit, *i.e.* the maximal amount of data that can be embedded, then visual quality, depending on the distortions introduced by the algorithm, and, at last, complexity, or the computational cost of the algorithm.

Pixel-based methods rely on pixel modification following specific patterns. Examples are LSB (Least Significant Bit) modification, statistical methods like patchwork, or fractal modification. Frequency-based methods uses the spectrum's invariance under geometrical transformations. Usually, spread spectrum techniques are used for different kinds of transforms, like Fourier-Mellin, Laguerre, Fresnel, Haar. Joint compression-insertion methods are frequency-based methods using the transformation performed by the still image coder. DCT-based method can be embedded in JPEG, using DCT coefficient inversion, addition and spread spectrum. JPEG-2000 wavelet coefficients can also be watermarked.

This paper deals with the joint use of an existing lossless coding scheme, called LAR, and a data embedding method, Difference Expansion (DE). Section 2 and section 3 introduce briefly the LAR codec and the DE, respectively, while section 4 describes a way to embed data in the LAR bitstream. Section 5 presents some results and discussions, and section 6 concludes our topic.

2. LAR LOSSLESS CODEC

The LAR codec is a multipurpose coding solution for still images. The LAR framework provides scalability both in resolution and quality, embedded bitstream from lossy to lossless coding, with computational low complexity, using grayscale and color imaging with state-of-the-art distortion-rate performance [2]. This codec adapts resolution to the local activity of the image. So low resolutions correspond to smooth areas, and high resolutions correspond to high frequency areas (edges). That adaptation is described by a block-based quadtree partition, with block size from 2x2 to 16x16 pixels, or more, if needed.

Figure 1: Scalability both in resolution and quality for the LAR-iSP.

The LAR Interleaved S+P (LAR-iSP) [3] uses a hierarchical decomposition of the picture to allow resolution scalability, quality scalability, and lossless compression, as depicted in figure 1. LAR-iSP consists in a bottom-up pyramidal construction formed by the S-Transform of diagonally adjacent pixel, as shown by figure 2. Coding is performed with two top-down decompositions of the pyramid, using enriched context predictor (360-degrees) introduced by Wu[4] and error of prediction coding. The first pass computes the low resolution image and the second one provides the texture. Efficiency is obtained trough the use of inter- and intra-level prediction. An implicit context modeling by the quadtree decomposition provides minimum redundancy. Thus, LAR-iSP coder performs better than state-of-the-art coders for lossless coding, especially on medical images.

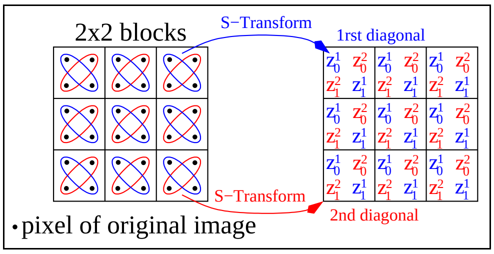


Figure 2: Interleaved S+P transforms. z_0^i resp. z_1^i is mean resp. gradient value of ith diagonal

3. DIFFERENCE EXPANSION

Difference Expansion (DE) is a method introduced by Tian [5] that embed one bit per pixel pair based on S-Transform. Pixels are first paired according to a predetermined pattern and S-Transform is applied. One bit b of data is inserted in the gradient h such as: $h' = 2 \times h + b$. The new gradient h' is then used (instead of h) in the inverse transform to produce the marked two pixels. This operation is called DE. In order to avoid over/underflows problems during the inverse transformation and to extend capacity, gradients are divided into three sets: *expandable* gradients which are marked by the DE, *changeable* gradients which cannot be transformed with the DE but embed one bit by saving their LSB and replacing it such as: $h' = 2 \times \lfloor \frac{h}{2} \rfloor + b$, and at last *non-changeable* gradients which cannot be used at all. Moreover, to be a system as distortionless as possible, some expandable gradients, that are also changeable, should be changed instead of expanded. The support is adapted to the target payload by a looped algorithm in order to distinguish expandable gradients that have to be expanded and expandable gradients that have to be changed. During the process, a binary map, the location map, is created and used to locate expandable and changeable/non-changeable gradients.

This technique provides among the best capacity-distortion performances presented in the literature, with the advantage of being fully reversible. Indeed, the original image can be losslessly restored thanks to the binary location map of gradients included in the embedded data. Further developments include Hierarchical DE, which embed data by performing several iterations using ordinary DE.

4. DATA EMBEDDING IN LAR-ISP

LAR-iSP and DE both use S-Transform during their computation. The main goal is to perform the data insertion without degrading coding performances. In order to adjust the DE to LAR-iSP, some minor modifications are introduced compared to the original DE method as described in section 3.

1. S-Transform is applied to the first and the second diagonals of 2x2 pixels blocks partitioning the picture,and DE is performed on the z_1^i values. We choose coefficients belonging to a given block size, so that the support depends on the quadtree partition and thereafter it depends on the image content. The first diagonal is marked with the highest priority, then the second one if a larger capacity is required.
2. In the original *DE*, the separation of expandable gradients is done by an iterative process. To speed up the process, all expandable gradients are expanded, and thanks to endmarks inserted in the bitstream, the decoder knows the location of expandable gradients.

5. EXPERIMENTAL RESULTS

All experiments were performed by embedding respectively a payload of 1024 bits and a payload as large as possible. Payloads are pseudo-random binary sequences generated by the `rand()` C function with different seeds. This simulates the encryption of a message using symmetric cryptography. Otherwise stated, images are 512x512 pixels, 8bpp.

5.1 Capacity-distortion performances

Table 1 presents the available capacity for different marked block sizes for image `lena`. Note that the maximal payload, using pixels from 2x2 block size, is smaller than other block sizes, even if the maximal capacity is larger. Difference in the size of compressed location map explains this apparent paradox. Measure of that difference is the DE overhead.

block size	2	4	8	16	4–16
max capacity (bits)	34816	30848	32640	32768	96256
DE overhead (bits)	7676	994	1042	1170	3042
DE overhead (bpp)	0.03	0.004	0.004	0.004	0.01
max payload (bits)	27142	29854	31598	31710	93214
max payload (bpp)	**0.10**	**0.11**	**0.12**	**0.12**	**0.36**
PSNR (dB)	**32.93**	**43.41**	**46.58**	**48.60**	**40.91**
Mobj	583	70	43	45	89
Nobj	3.36	3.75	3.77	3.77	3.73

Table 1: Embedded payload size, capacity PSNR, objective quality measure and notation vs. marked block sizes for image `lena`

Figure 3 presents distortion-capacity curves for LAR+DE, original DE (equivalent to LAR+DE considering all blocks), hierarchical DE and G-LSB [6]. LAR+DE is performed for several block sizes. What we have just noticed appears also with other images: there are more distortions when 2x2 block size is used. For other block sizes, the capacity-distortion rate is better than other lossless data embedding methods like G-LSB, and sometimes better than the original hierarchical (multipass) DE (almost for small payload sizes) as it is shown by figure 3. Note that

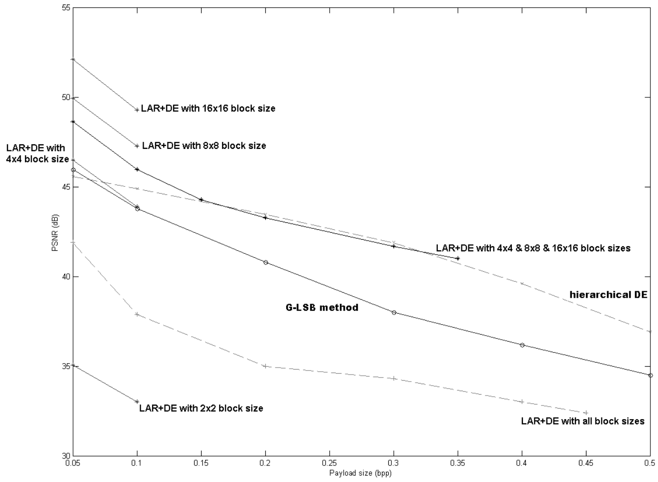


Figure 3: Capacity vs. distortion for image `lena` for different embedding methods

block size	2x2		4x4		8x8		16x16		4x4–16x16	
	bpp	dB	bpp	dB	bpp	dB	bpp	dB	bpp	dB
`lena`	0.11	33	0.11	43	0.12	47	0.12	49	0.36	41
`pcb`	0.13	28	0.10	45	0.09	49	0.07	51	0.27	43
`subdur`	0.07	35	0.11	44	0.11	48	0.13	49	0.34	42

Table 2: Capacity-distortion measures for various block's size. The columns marked bpp and dB gives resp. capacity and PSNR.

standard (single pass) DE doesn't provide good results in this case.

Table 2 presents capacity-distortion measure for natural image `lena`, synthetic look-like image `pcb` and medical image `subdur`. It confirms the need to mark the pairs of pixels present in the 4x4 to 16x16 blocks, in order to achieve a good tradeoff between capacity and distortion.

5.2 Visual quality

Figure 4 presents an example of embedded picture, with pairs of blocks of size 4 to 16 tagged. Overall quality is fairly good. An evaluation of objective visual quality has been performed using KOMPARATOR[7] and results are shown on the last 2 lines on table 1. Mobj is an objective measure of the visual quality, like a visual-PSNR, while Nobj aims at giving an objective note for the image quality, with notes from 1 (very bad) to 5 (very good). It shows that visual quality is better when pairs of pixels are chosen in blocks with size ranging from 4x4 to 16x16.

It is known that the Human Visual System is less sensitive at low and high spatial frequencies and modifying high frequencies could provide quite good invisibility. However, high frequencies imply edges, and pixels of edges are very pertinent because they define the semantic of the picture. This is why there are more distortions with 2x2 block size than with others: the DE modifies edges unrespectfully. We called this artefact an *inter-block incoherence*. Its effect is to locally inverse the orientation of the gradient, introducing new visible edges. Nevertheless, 2x2 blocks could be an adapted support to achieve a good capacity-distortion rate, but modified blocks have to be chosen depending on their neighborhood.

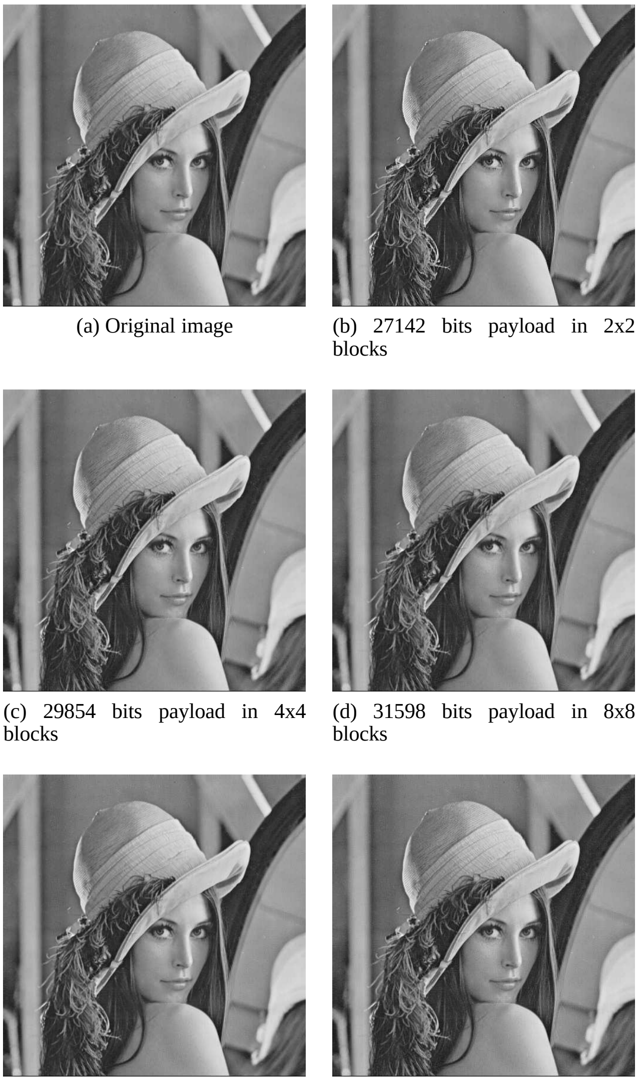
5.3 Coding cost

In order to evaluate the impact of the data embedding in the coding scheme, we have computed the size of the actual bitstream needed to code losslessly the different pictures. Table 3 sums up the results obtained. Note that due to randomness, the minimum coding cost for the payload is its own size. DE overhead cost (the location map) must also be taken in account.

block size	payload [bpp]	DE overhead [bpp]	coding cost [bpp]	coding overhead [bpp]	coding overhead [%]
			4.31		
2	0.10	0.03	4.50	0.06	60
4	0.11	0.004	4.51	0.08	73
8	0.12	0.004	4.51	0.08	67
16	0.12	0.004	4.49	0.07	58
4–16	0.36	0.01	4.73	0.05	14

Table 3: LAR-iSP+DE coding costs. First line represents the coding cost of the original picture.

The interesting features in the table are the coding overhead costs, that is, what is the cost to code the embedded payload. Added to the DE overhead, it gives the overhead of the LAR+DE. As DE overhead has been discussed in section 5.1, we focus on the coding overhead. The coding over-



(a) Original image

(b) 27142 bits payload in 2x2 blocks

(c) 29854 bits payload in 4x4 blocks

(d) 31598 bits payload in 8x8 blocks

(e) 31710 bits payload in 16x16 blocks

(f) 93214 bits payload in simultaneous 4x4, 8x8 and 16x16 blocks

Figure 4: Reversibly embedded *lena* with the maximum payload for each block size.

head reflects mostly the predictor's adaptation to the new gradient value, as modified by the DE embedding scheme. Table 3 shows that coding overhead is related to the payload, The more pixels are marked, or, equivalently, the more gradients are modified, the better the predictor behaves. In fact, as the gradients distribution becomes uniform, the predictor performs better.

6. CONCLUSION

In this paper, we have adapted a fast and efficient reversible data embedding algorithm for the LAR-Interleaved S+P compression framework, introducing the DE. Both this codec and the data embedding algorithm explore the redundancy in the digital picture to achieve respectively either better than state-of-the-art compression rates or reversible data embedding. We show the need to insert data using pairs present in block with size ranging from 4x4 to 16x16. We obtained resulting capacity-distortion rates of embedded images belong to the best in the literature about lossless data embedding. Nevertheless, the coding cost of the embedded data shows to be manageable.

REFERENCES

[1] M. Babel, B. Parrein, O. Déforges, N. Normand, J. P. Guédon, and J. Ronsin, "Secure and progressive transmission of compressed images on the internet: application to telemedicine," in *SPIE*, pp. 126–136, 2005.

[2] O. Déforges, M. Babel, L. Bédat, and J. Ronsin, "Lar codec: a representation and image compression scheme based on adaptive resolution scheme and self-extracting region representation," *IEEE Trans. on Circuits and Systems for Video Technology* **17**, july 2007.

[3] M. Babel, O. Déforges, and J. Ronsin, "Interleaved S+P pyramidal decomposition with refined prediction model," in *ICIP*, 2005.

[4] X. Wu, "Lossless compression of continuous-tone images via context selection, quantization and modelling," in *IEEE Transactions on Image Processing*, pp. 656–664, 1996.

[5] J. Tian, "Reversible data embedding using a difference expansion," in *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 890–896, august 2003.

[6] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Reversible data hiding," in *ICIP*, pp. 157–160, 2002.

[7] P. L. Callet and D. Barba, "Robust approach for color image quality assessment," in *VCIP*, pp. 1573–1581, 2003.