EFFICIENT MEMORY DATA ORGANIZATION FOR FAST STILL IMAGE DECODING IMPLEMENTATION

Anissa Zergaïnoh-Mokraoui, Pierre Duhamel+*

*L2TI, Institut Galilée, Université Paris 13, +LSS\CNRS, SUPELEC,
*99, Avenue Jean Baptiste Clément, 93 430 Villetaneuse, France
+Plateau de Moulon, 91 192 Gif sur Yvette, France
email: zergainoh@galilee.univ-paris13.fr

ABSTRACT

This paper proposes a new still image codec. The encoder has the following structure: a set of pixels of the image is selected and transmitted, together with their position. Then, the value of the image at other places is obtained by a prediction algorithm at the decoder. A useful theoretical covariance model adapted to the image to be encoded is proposed, avoiding the transmission of additional information. The selected pixels and their corresponding positions on the image are encoded using lossless coding algorithms. The computational time of the decoding process is significantly reduced according to the efficient structured memory organization of (i) the image covariance values and (ii) the ordered distances concerning the search of nearest pixels. Experimental results performed on a set of test images show that the rate-distortion results are competitive to the best coders JPEG 2000 and SPIHT with arithmetic coding.

Index terms *– Image coding, image reconstruction, prediction methods, image sampling*

1.INTRODUCTION

Image coding topic receives an increasing interest since the multimedia applications are more and more growing demand in terms of memory resources for storage purpose and\or a higher speed transmission. State of the art, on the framework of still image coding, shows that the most powerful conventional image coders are all derived from the wavelet transform paradigm. Three underlying components are currently implemented in the traditional wavelet coders: (i) the wavelet transform decorrelating and compacting the energy into few coefficients, (ii) the quantization procedure, and (iii) the entropy coding step. Considerable research works are performed on these different components. Based on these remarks, many best image wavelet coders have been developed such as the EZW algorithm ([1]), followed by the SPIHT algorithm ([2]) and the EBCOT algorithm ([3] [4]) adopted by the JPEG200 image compression standard ([5]). This paper proposes a new still image coding approach based on irregularly sub-sampled images.

Usually, digital images contain both homogeneous regions in smooth areas and non-homogenous regions in detail areas. The conventional sampling process operates without any consideration of the type of the different image areas. Only one selected sampling rate, above the Nyquist rate, is applied over the whole image allowing therefore the extraction of equally spaced pixels. However, the regular sampling generates redundant information in homogenous regions. Rather than restricting the sampling rate to regular one (above Nyquist rate), this paper proposes to operate with variable sampling rate adapted to the image content.

Irregular sub-sampling image approach retains a small fraction of pixels (non-redundant), thus reducing the memory size of the original image since the number of the retained pixels on the original image has decreased. The problem of selecting the appropriate pixels on the image grid in accordance with the minimization of the mean square error between the original image and its prediction is considered. An adaptive sub-sampling algorithm closely related to the prediction algorithm is proposed. The distribution of the samples is thus closely related to the content of the image to be encoded. The proposed codec is summarized by Fig. 1.

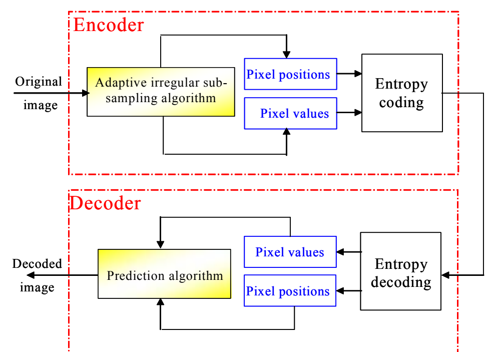


Fig. 1 Still image codec

In the literature many reconstruction algorithms are available (see e.g. [7], [8], [9], [10]). However some algorithms require a high computational load and/or too many parameters to be transmitted to the decoder affecting therefore the performance of the codec. The main objective of this paper is to propose a new competitive still image codec.

The performance of the proposed codec in terms of encoding efficiency and fast decoding implementation is mainly due to the modifications brought to the prediction algorithm proposed in [11]. In order to reduce the encoding cost of the proposed approach, we propose a useful theoretical covariance model which approximates suitably the experimental model. The covariance values are computed only once and are stored in an efficient way in the memory. These values are accessible during all the decoding process. Moreover an efficient search algorithm of the nearest neighbors is proposed. It is based on an efficient memory data organization reducing the computational load of the decoder.

The rest of the paper is organized as follows. Section 2 introduces the adaptive sub-sampling algorithm adopted by the encoder. Section 3 presents the general formulation of the basic prediction algorithm and provides its main drawbacks. Section 4 focuses on the modifications brought to the basic algorithm.

Section 5 provides the simulation results. Section 6 concludes our work.

2. ADAPTIVE SUB-SAMPLING ALGORITHM ADOPTED BY THE ENCODER

This section deals with the optimal distribution problem of the retained pixels on the image grid for the encoding purpose. Before presenting the sub-sampling algorithm, we introduce some notations required for our later developments.

Let denote I the original image to be encoded. It is defined as follows:

$$I = \{u(x_i, y_i) \text{ for } x_i = 1,...,M;\, y_i = 1,...,N\} \quad (1)$$

where $u(x_i, y_i)$ is the gray level of the pixel located at the position (x_i, y_i) on the image grid I.

The prediction of the pixel, located at position (x_i, y_i) on the original image grid I, is denoted $\hat{u}(x_i, y_i)$. The prediction error between the original pixel and its prediction is given as follows:

$$\varepsilon(x_i, y_i) = \hat{u}(x_i, y_i) - u(x_i, y_i) \quad (2)$$

Let denote I_0 the original image irregularly sub-sampled as described in the following section.

In order to reconstruct the initial image with a minimum loss of information, an adaptive sub-sampling algorithm is proposed. Fig. 2 and Fig. 3 summarize the different optimization steps of the adaptive sub-sampling algorithm.

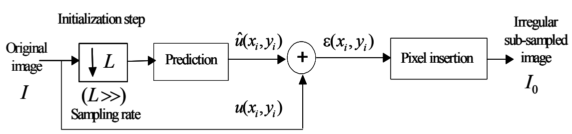


Fig. 2 Initialization step of the adaptive sub-sampling algorithm

As an initialization step, the adaptive sub-sampling algorithm begins by selecting a very weak sampling of pixels regularly spaced on the original image I (sampling rate denoted by L in Fig. 2). The retained initial amount of pixels must be as smaller as possible than the fixed set of the pixels to keep for the coding process.

The second step operates on this regular sampled grid. The missing pixels are reconstructed using a prediction algorithm which will be presented in the next section. The predicted pixels introducing the maximal prediction errors ($\varepsilon(x_i, y_i)$) are considered as candidate pixels to be encoded. The threshold prediction error is implicitly fixed by the desired amount of sampling to reach. Therefore, the pixels are inserted on the image grid providing naturally an irregular sub-sampled image denoted I_0. The distribution of the samples is closely related to the content of the image to be encoded.

The last optimization step, provided by the block diagram of Fig. 3, is considered as a refinement step of the pixels around their close neighborhood. Indeed, at the initialization step, the weak set of pixels has been regularly distributed without any specific criterion. Therefore each previously selected pixel, belonging to the sub-sampled image I_0 (Fig. **2**), is replaced by one of its eight close neighbor pixels. For each of eight replacements, the dropped pixels on the image grid I_0 around the same neighborhood are predicted. Among these eight considered pixels, the pixel which introduces a maximal prediction error in its neighborhood is retained on the sub-sampled grid I_0.

The refinement process is applied several times on the fixed available amount of pixels, until the global mean square prediction error converges to a minimal mean square error.

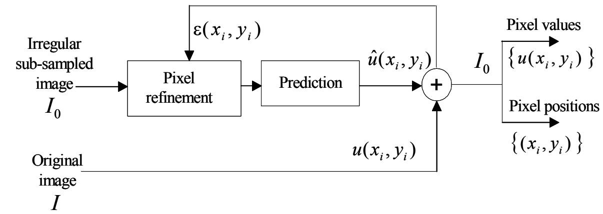


Fig. 3 Refinement steps of the adaptive sub-sampling algorithm

3. THE BASIC PREDICTION ALGORITHM AND ITS MAIN DRAWBACKS

The unknown gray level value, located at position (x_k, y_k) on the irregular sub-sampled image grid, is estimated using a weighted linear combination of n available pixels in the neighborhood, denoted D_{x_k, y_k}, around the unknown pixel. The estimator is given as follows:

$$\hat{u}(x_k, y_k) = \sum_{i=1}^{n} a_{i, D_{x_k, y_k}} u(x_i, y_i) \quad \text{with } u(x_i, y_i) \in D_{x_k, y_k} \quad (3)$$

where $a_{i, D_{x_k, y_k}}$ is the *i-th* weight associated to the pixel $u(x_i, y_i)$ located at position (x_i, y_i) on the image grid I_0. The parameter n represents the neighbors of the unknown pixel belonging to D_{x_k, y_k}.

The best linear unbiased predictor is obtained by determining the weights $a_{i, D_{x_k, y_k}}$ those minimizing the mean square error (MSE) between the original image I and its prediction $\hat{I}$. The formalism of Lagrange leads to the following system of equation:

$$\begin{cases} \sum_{i=1}^{n} a_{i, D_{x_k, y_k}} Cov(u(x_j, y_j), u(x_i, y_i)) + \mu_{D_{x_k, y_k}} = Cov(u(x_j, y_j), u(x_k, y_k)) \\ \sum_{i=1}^{n} a_{i, D_{x_k, y_k}} = 1 \end{cases}$$

$$\forall j \in N \quad (4)$$

where $Cov(u(x_j, y_j), u(x_i, y_i))$ denotes the covariance value between two pixels $u(x_i, y_i)$ and $u(x_j, y_j)$ located respectively at positions (x_i, y_i) and (x_j, y_j) on the image grid. $\mu_{D_{x_k, y_k}}$ is the Lagrange multiplier.

Let us analyse what happens, in terms of encoding cost and computational load, if the codec uses directly this basic prediction algorithm:

(i) The prediction of one pixel by the decoder requires the knowledge of the following information which therefore must be transmitted by the encoder side:

- the coordinates of the retained pixels;
- the grey levels of the retained pixels;
- the image covariance values.

(ii) In point of view of the computational load, at each time when the decoder predicts one pixel located at position (x_i, y_i), the computation of the unknown set of coefficients $a_{i, D_{x_k, y_k}}$ is necessary. For this, the decoder follows the listed steps:

- searching the n nearest neighbors around the unknown pixel;
- constructing the matrix of the image covariance values;

- solving the system of equation (4) which consists to inverse the image covariance matrix of size $(n+1)\times(n+1)$.

The first point (i) affects considerably the global encoding cost of the compression method. While the second point (ii) increases significantly the global computational load of the decoder.

4. MODIFIED PREDICTION ALGORITHM FOR FAST DECODING IMPLEMENTATION

To overcome the drawbacks of the basic prediction algorithm, this section proposes different modifications brought to the algorithm as listed below.

4.1. Reduction of the number of neighbors

The number of neighbors required by the prediction algorithm is restricted to three. As a first consequence, this reduces the size of the matrices to be handled by equation (4).

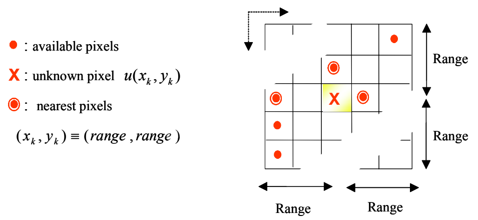


Fig. 4 A master sliding window

4.2. Useful theoretical covariance model

Inspired from kriging methods developed in mining applications ([6]), we propose the construction of a theoretical covariance model which approximates suitably the experimental covariance of the image given by the following equation:

$$Cov(h) = 1 - \frac{1}{N(h)} \sum_{i \le j} (u(x_i, y_i) - u(x_j, y_j))^2 \tag{5}$$

where $N(h)$ is the number of pairs separated by the distance h between two pixels. This method, as it will be described below, avoids the transmission of the covariance values to the decoder.

This step is important since the selected covariance model determines the quality of the reconstructed image. In the majority of experiments performed on several image realizations, the linear covariance model is generally well adapted in a domain image restricted within a range usually lower or equal to 20 (see Fig. 4).

We propose to use the following linear covariance model where h is the normalized Euclidian distance between two pixels:

$$Cov(h) = 1 - h \tag{6}$$

Assume that three nearest pixels, belonging to the sliding domain D_{x_k, y_k}, are selected ($u(x_1,y_1)$, $u(x_2,y_2)$, $u(x_3,y_3)$) around the unknown pixel $u(x_k, y_k)$. The system of equation (4) is presented in a matrix form as follows:

$$\begin{bmatrix} a_{1,D_{x_k,y_k}} \\ a_{2,D_{x_k,y_k}} \\ a_{3,D_{x_k,y_k}} \\ \mu_{D_{x_k,y_k}} \end{bmatrix} = \begin{bmatrix} 1 & Cov(h_{1,2}) & Cov(h_{1,3}) & 1 \\ Cov(h_{2,1}) & 1 & Cov(h_{2,3}) & 1 \\ Cov(h_{3,1}) & Cov(h_{3,2}) & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} Cov(h_{1,k}) \\ Cov(h_{2,k}) \\ Cov(h_{3,k}) \\ 1 \end{bmatrix} \tag{7}$$

where $h_{i,j}$ is the normalized Euclidian distance between two pixels ($u(x_i, y_i)$, $u(x_j, y_j)$) belonging to D_{x_k, y_k} as follows:

$$h_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \Big/ b \tag{8}$$

The weight coefficients, given by equation (7), are computed analytically (e.g. using symbolic computations) once for all. The expressions are not provided in this paper due to lack of place. Each coefficient is a weighted combination of covariance values. Before concentrating on the computation of these covariance values in a fast way, the following paragraph focuses on the fast search for the nearest pixels

4.3. Efficient search algorithm of the nearest pixels

The Euclidian distance between the coordinates of the pixels is used to determine the nearest pixels around the pixel to be predicted. The algorithm is based on the construction of a master sliding window (see Fig. 4) depending on the fixed range: $D_{range,range} = [0, 2\times range] \times [0, 2\times range]$.

Only once, as an initialization step, the proposed search algorithm constructs and sorts in an increasing order the $(2\times range+1)^2$ Euclidian distances $h_{i,j}$ computed between each point belonging to the master window and its center point $(range, range)$ corresponding to the point that we want to predict (see Fig. 4).

The memory organization for the fast search of neighbors is provided by Fig. 5 where the first column concerns the $(2\times range+1)^2$ distances previously sorted only once. Then, the search algorithm updates the second and third columns.

For the prediction of any pixel $\hat{u}(x_l, y_l)$, the algorithm updates the second column as follows:

$$\begin{cases} x_i = x_i + x_l - range \\ y_i = y_i + y_l - range \end{cases} \text{with } (x_l, y_l) \in D_{x_l, y_l} \tag{9}$$

The third column is updated with the grey level of the pixel (if available) or the character string "*Nap*" if non available pixel. Since the distances are already sorted, the decoder recovers directly the three nearest retained pixels on the image grid.

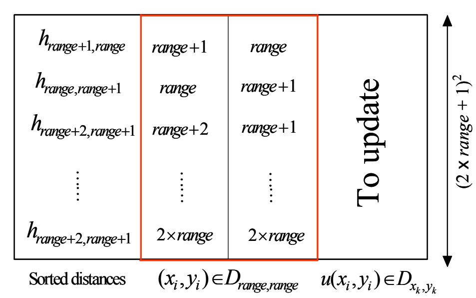


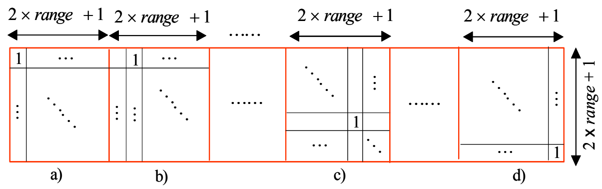
Fig. 5 Fast search algorithm of the nearest pixels using efficient memory organization

4.4. Memory organization of the covariance values

Once the closest neighbors are obtained, one needs to be able to compute their covariance values in a fast way. As an initialization step, all possible covariance values are computed once for all on the master sliding window $D_{range,range}$ according to the following relation:

$$Cov(h_{i,j}) = 1 - h_{i,j} \text{ for all } i, j \in [0, 2\times range] \tag{10}$$

These covariance values are then efficiently organized and stored in a memory (see Fig. 6). The first block, denoted by (a) in Fig. 6, concerns all the covariances computed between the first left pixel in the sliding window with all pixels in the sliding window and so on. For each pixel prediction, the decoder has only to recover the desired covariances from the memory according to the position of the nearest pixels.



for $k = 0,..., 2 \times range$; $l = 0,..., 2 \times range$,

$a)\ 1 - dist((0,0),(k,l))$ $b)\ 1 - dist((0,1),(k,l))$ $c)\ 1 - dist((i,j),(k,l))$

$d)\ 1 - dist((2 \times range + 1, 2 \times range + 1),(k,l))$

Fig. 6 Efficient organization of the image covariance values

5. ENCODING PROCESS

This section concerns the encoding process. According to the previous sections, only two kinds of information have to be encoded: the retained pixel values and their corresponding positions on the image grid.

The positions of the pixels are encoded using a bi-level map grid where (i) the retained pixels are encoded by “1” and (ii) the rejected pixels by “0” at their corresponding positions on the sub-sampled image. Many lossless coding methods are available in the literature (arithmetic coding, RLE, VLC, LZ, …). However the lossless JBIG2 compression method has been retained since its compression ratio is higher compared to other lossless methods ([12]).

The retained pixel values are gathered in a vector. The Lempel-Ziv-Markov chain-Algorithm (LZMA) combined to the arithmetic coding algorithm has been chosen to encode these retained pixels. LZMA is based on a variable dictionary scheme and presents a high compression ratio particularly for this kind of data. The software for a fast decoding version is provided in ([15]).

6. SIMULATION RESULTS

This section presents and compares the performances of the proposed still image codec. The simulation results are performed on traditional test images Lena, boat, Goldhill and peppers images. The respective reference software of the best coders SPIHT ([13]) and JPEG2000 ([14]) is used to compare the performance, in term of rate-distortion, of our codec. The quality of the decoded image is measured using the Peak Signal to Noise Ratio between the original and the decoded image.

The adaptive sub-sampling algorithm is applied on each test image. The sampling, in the provided examples, has been fixed to 4.9% of pixels. The initial regular sampling has been fixed to 0.45% of pixels.

Size	Images	Rate (bpp)	SPIHT (dB)	JPEG2000 (dB)	New codec (dB)
512×512	Lena	0.636	38.29	38.28	38.21
512×512	Boat	0.654	36.09	35.96	35.30
512×512	Peppers	0.635	36.66	36.63	37.27
512×512	Goldhill	0.646	34.25	34.30	34.61

Table 1 Comparisons of the PSNR results

Table 1 summarizes the PSNR results achieved by the different coders for a fixed rate. The final sub-sampling optimization step for Lena, Boat and Peppers has been achieved respectively for 6, 2 and 6 global iterations. The results are similar for Lena, less good for Boat, better for Peppers and Goldhill. The results are closely related to the complexity of the image. Obviously, this method is not very competitive for very low bit rate, since a very small fraction of pixels (e.g. smaller than 1%), will not lead to an acceptable distortion, except on very specific images. However, it is seen that, for medium rates, it can be very efficient. It is likely that the good performances on peppers and Goldhill is due to its smoothness, while boat, being very busy, would require a higher number of points to be transmitted, and the method would be efficient only for higher rates.

7. CONCLUSION

This paper has proposed a new still image codec. The presented approach is based on a variable and adaptive sampling rate depending on the image content. The retained pixels are selected according to the smaller prediction error. A useful theoretical covariance model with no side information needed by the decoder has been proposed. Focusing on the implementation complexity of the prediction algorithm, we have proposed an efficient memory organization of the covariance values and the distances allows obtaining a significant saving in computational time. The PSNR results show that the proposed method is competitive to the best coders. However, many questions arise for future investigations such as (i) from which fraction of pixels the compression method remains valid with a competitive distortion (ii) for a fixed bit budget and a given distortion how many pixels the compression method must retain.

8. REFERENCES

[1] J. M. Shapiro, Embedded image coding using zerotrees of wavelet coefficients, *IEEE Transactions on signal processing,* 41(12) : 3445-3462, 1993.

[2] A. Said, W. A. Pearlman, A new fast and efficient image codec based on set partitionning in hierarchical trees, *IEEE Transactions on circuits and systems for video technology*, 6(3) : 243-250, 1996.

[3] D. Taubman, High performance scalable image compression with EBCOT, *IEEE Transactions on Image Processing*, 9(7) : 1151-1170, July 2000.

[4] D. Taubman, E. Ordentlich, M. Weinberger, G. Seroussi, Embedded block coding in JPEG 2000, *Signal Processing-Image Communication*, 17(1) : 49-72, January 2002.

[5] ISO/IEC 15444-1: JPEG2000 image coding system, 2000.

[6] D. G. Krige, A statistical approach to some basic mine valuation problems on the Witwatersrand, *Journal chem. Metal. Min. S. Afr*., S2, pp. 119-139, 1951.

[7] L. L. Schumaker, Fitting Surfaces to Scattered Data, pp. 203-268, G., Lorentz, Georg Gunter, eds. Chui, Schumaker, 1976.

[8] R. Franke, G. Nielson, Smooth interpolation of large sets of scattered data, *International of Numerical Methods in Engineering*, vol. 15, 1691-1704.January 1980.

[9] D. Shepard, A two dimensional interpolation function for irregularly spaced data, proceedings 23rd Nat. conference ACM, 517-523, 1968.

[10] H. Le Floch, C. Labit, Irregular image sub-sampling and reconstruction by adaptive sampling, *ICIP*, vol. 3, 379-382, 1996.

[11] A. Zergaïnoh, J.-P. Astruc, Hybrid method for still image coding, *Eusipco*, 5-8 september 2000, Tamper Finland.

[12] http://www.jpeg.org/jbig/index.html

[13] http://www.cipr.rpi.edu/research/SPIHT/spiht3.html

[14] http://www.kakadusoftware.com/

[15] http://www.7-zip.org/sdk.html