International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:2, No:7, 2008

Reversible, Embedded and Highly Scalable Image Compression System

Federico Pérez González, Iñaki Goirizelaia Ordorika, and Pedro Iriondo Bengoa

***Abstract*—** In this work a new method for low complexity image coding is presented, that permits different settings and great scalability in the generation of the final bit stream. This coding presents a continuous-tone still image compression system that groups loss and lossless compression making use of finite arithmetic reversible transforms. Both transformation in the space of color and wavelet transformation are reversible. The transformed coefficients are coded by means of a coding system in depending on a subdivision into smaller components (CFDS) similar to the bit importance codification. The subcomponents so obtained are reordered by means of a highly configure alignment system depending on the application that makes possible the re-configure of the elements of the image and obtaining different importance levels from which the bit stream will be generated. The subcomponents of each importance level are coded using a variable length entropy coding system (VBLm) that permits the generation of an embedded bit stream. This bit stream supposes itself a bit stream that codes a compressed still image. However, the use of a packing system on the bit stream after the VBLm allows the realization of a final highly scalable bit stream from a basic image level and one or several improvement levels.

***Keywords*—** Image compression, wavelet transform, highly scalable, reversible transform, embedded, subcomponents.

I. Introduction

SINCE Shapiro [1] in 1993 proposed the general structure of EZW and subsequently SPIHT [2] was released, the image codification using the wavelet transform by means of the decomposition method of Mallat [7] has been aim of studying in the image and video compression field, giving as a result standards such as JPEG2000 [4] and introducing capacities in MPEG-4 visual [6]. Some of the wavelet based-encoders [6] [10] have developed reversible compression from no linear transformations that represent integers with integers [8] and using the lifting scheme of Sweldens [9].

In this work a new method for image codification based in low complexity wavelet transforms is presented, that permits different settings and great scalability in the generation of the final bit stream. This codification presents a continuous-tone still image compression system that groups loss and lossless compression making use of finite arithmetic reversible transforms.

The system is valid for the transmission in systems with low bit-rate and, besides its high scalability, permits the perfect rebuilding of the original signal. Moreover, it has low complexity, so that permits simple hardware and software systems.

Manuscript received May 20, 2005. This work has been supported in part by MCYT&FEDER under project DPI 2002-2399.

Federico Pérez González is with the Faculty of Engineering, Department of Automatic Control of the Basque Country University, Bilbao, Spain (phone: +34-946-01-4005; fax: +34-946-01-4187; e-mail: jtppegof@bi.ehu.es).

Iñaki Goirizelaia Ordorika is with the Faculty of Engineering, Department of Telecommunications of the Basque Country University, Bilbao, Spain (e-mail: jtpgoori@bi.ehu.es).

Pedro Iriondo Bengoa is with the Faculty of Engineering, Department of Automatic Control of the Basque Country University, Bilbao, Spain (e-mail: jtpirbepf@bi.ehu.es).

II. General Description of the Compression System

The elements of the compression system are in the following figure: (Fig. 1.)

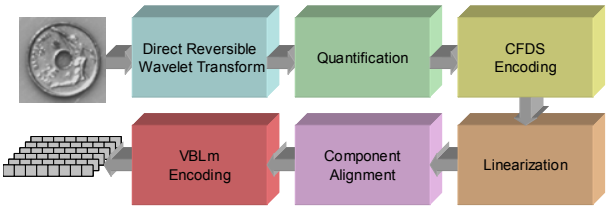

Fig. 1 General description of the compression system

To describe the compression system an image input of one component (gray scale) is assumed. The applied wavelet transforms are reversibles. The transformed coefficients are then quantified, in the case of compression with losses. Later, the transformed coefficients are coded by means of a codification system depending on a subdivision into smaller components (CFDS) similar to the bit importance codification. The subcomponents so obtained are linearized and then reordered by means of a highly configurable alignment system depending on the application that makes possible the re-configure of the elements of the image and obtaining different importance levels from which the bit stream will be generated. The subcomponents of each importance level are coded using a variable length entropy codification system (VBLm) that permits the generation of an embedded bit stream. This bit stream supposes itself a bit stream that codes a compressed still image.

A. Reversible Wavelet Transform

The set of the wavelet transformations that encode integer-to-integer is a reduced subgroup into the wavelet transformations [7]. This type of transformation has been

chosen due to the possibility of perfect rebuilding and its simple arithmetic. Those transformations among the set of reversible transformations that fill up a series of conditions have been looked for:

1. *Integer transformations*: Reversibility has been usually applied on transformations on which the whole set of symbols generated in the lifting process where integers.
2. *Operations with powers of 2*: It has been sought that the arithmetic operations of multiplication and division where the easiest, because of that, transformations on which the constants of multiplication and division were powers of 2 have been chosen, in a way that in the integer number arithmetic, these operations could be done with only bit shifts.
3. *The coefficients resulting from low frequencies*: They are matched with averages from the original coefficients.
4. *The coefficients resulting from high frequencies*: They are matched with subtractions of the original coefficients plus and a prediction element.

Attending to the lifting scheme [8], these transforms could be re-written in the direct transform case as

$$\begin{cases} d^1_{j+1}[k] = c_j[2k+1] - c_j[2k] \\ c_{j+1}[k] = c_j[2k] + \lfloor d^1_{j+1}[k]/2 \rfloor \\ d_{j+1}[k] = d^1_{j+1}[k] + p_{j+1}[k] \end{cases}$$

And the inverse transform

$$\begin{cases} d^1_{j+1}[k] = d_{j+1}[k] - p_{j+1}[k] \\ c_j[2k] = c_{j+1}[k] - \lfloor d^1_{j+1}[k]/2 \rfloor \\ c_j[2k+1] = c_j[2k] + d^1_{j+1}[k] \end{cases}$$

As a result, the well-known S, TS, TT and S+P transforms has been used, where the prediction elements are in expressed in the Table I.

The reversible transformation is used on the image following the model of Mallat [7]. So, different subband types are obtained (LL, HL, LH and HH) for several levels of partitioning levels (Fig. 2.)

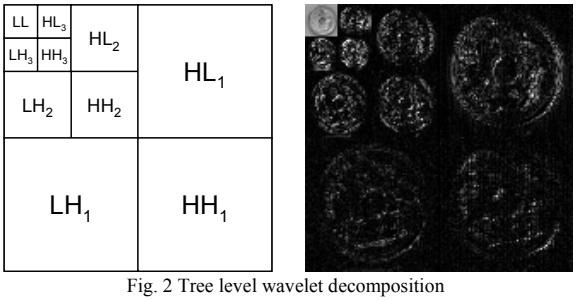

Fig. 2 Tree level wavelet decomposition

B. Quantification system

The quantification system has only sense when a loss in the quality of the signal is assumed. The quantification system of the transformed coefficients is built directly over the coefficients of the subbands depending on the type of the subband. Quantification is applied in a way that produces the smallest loss of information. According to this, the quantification to apply on the subbands with the greatest frequency component (subbands HL, LH and mainly HH) can be bigger than that used on those of smaller frequency component.

The quantification that must be used on the subbands will be a lineal quantification with a threshold of quantification with a value equal to a power of two (2^{r_s}). So, the quantification over the components of a subband s will be only shifts of r_s bits on the transformed coefficients.

$$c^c_s[k] = c_s[k] >> r_s$$

TABLE I
TRANSFORMS CLASSES

Transform	*Prediction element*
S	$p_{j+1}[k] = 0$
TS	$p_{j+1}[k] = \left\lfloor \begin{pmatrix} c_{j+1}[k-1] \\ -c_{j+1}[k+1] \\ +2 \end{pmatrix} \times \frac{1}{2} \right\rfloor$
SP	$p_{j+1}[k] = \left\lfloor \begin{pmatrix} 2 \times c_{j+1}[k-1] \\ +c_{j+1}[k] \\ -3 \times c_{j+1}[k+1] \\ +2 \times d^1_{j+1}[k+1] \\ +4 \end{pmatrix} \times \frac{1}{8} \right\rfloor$
TT	$p_{j+1}[k] = \left\lfloor \begin{pmatrix} 3 \times \begin{pmatrix} c_{j+1}[k-2] \\ -c_{j+1}[k+2] \end{pmatrix} \\ -22 \times \begin{pmatrix} c_{j+1}[k-1] \\ -c_{j+1}[k+1] \end{pmatrix} \\ +32 \end{pmatrix} \times \frac{1}{64} \right\rfloor$

C. Component Encoding

The coefficients studied till the moment are integer positive and negative numbers coded in two's complement. This codification must fill the property of being a one-to-one correspondence on a code that permits variable length. The codification that we are going to use is that named "*Codification in Function of the Partitioning in Subcomponents*" (CFDS). This type of codification is a modification of the codification in bit significance used in the systems of compression that work in bit planes such as CREW [3] or JPEG2000 [4].

With the CFDS codification we try to code integer values dividing their resolution in different fields (*subcomponents*).

The coefficients coded with CFDS are partitioned in M subcomponents of lengths n_i in such a way that $n = \sum_1^M n_i$, n keeps being the total resolution of the original coefficients.

In principle, the subcomponents with smaller weigh will have less priority in the need for codification regarding a compression system, because the energy stored in them is smaller than that stored in subcomponents of greater weigh. If a low weigh subcomponent is lost, the problem poses when the codification of the sign is lost with the loss of the subcomponent. Faced this circumstance, a solution to minimize the error of subcomponent loss when the sign is in the lost subcomponent, is to code the affected coefficients after trunk as a zero.

Looking on the case of the association in two subcomponents of factor 1:3 (first subcomponent of 3 bits and second subcomponent of 1 bit) shown in Table II, to minimize the error owed to the loss of subcomponents of smaller importance it would be convenient that the decodification of `1000`, was `0`.

What is proposed with the CFDS codification is to have two codes for the zero as in bit significance, but now the encoding for the zero will be the zero itself and on the other hand the codification matching to putting the lowest weight bit of the second subcomponent to one.

The election of the second subcomponent is owed to the nature of the wavelet transform, where a big part of the coefficients has absolute value close to zero, and therefore, are usually coded between the two first coefficients.

TABLE II
CFDS Coding

Decimal	Two's complement	Minimal length with sing	Bit significance	CFDS (1:3)
-8	1000			
-7	1001	1111	1111	1111
-6	1010	1110	1110	1110
-5	1011	1101	1101	1101
-4	1100	1100	1100	1100
-3	1101	111	0111	0111
-2	1110	110	0110	0110
-1	1111	11	0011	0011
0	0000	0, 10	0000, 0001	0000, 1000
1	0001	01	0010	0001
2	0010	010	0100	0010
3	0011	011	0101	0100
4	0100	0100	1000	0101
5	0101	0101	1001	1001
6	0110	0110	1010	1010
7	0111	0111	1011	1011

Beginning in the codification in bit significance, it is necessary to reassign the symbols of codification to pass to the CFDS codification. It would be enough with the codification of 1 as 0x01 (it is a representation of zero of bit significance), the codification of 2 as the one of the 1 for bit significance, the one of 3 as the one of 2 for bit significance, etc., till reaching the decode corresponding to the new zero (4 in our example).

D. Linearization System

With the algorithm of linearization we pass from a two-dimensional component layout to a one-dimensional component layout. The linearization of the components is made subband by subband beginning with the LL subband and following with the group of three frequency subbands HL_i, LH_i y HH_i in that order. To order the coefficients, a *Z-distribution* (Fig. 3,) is followed in a such a way that the arrangement is done with groups of 2x2 of four coefficients.

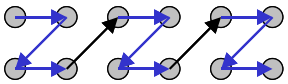

Fig. 3 Z-Distribution linearization

E. Component Alignment

The scalable nature of the current algorithm of compression is based in the concept of arrange or embedding of the coefficients, that is made before the process of entropy codification. The *alignment process* is basically a process of arrangement of planes of generated subcomponents, having into account the resolution of each of the already used in the process of *CFDS codification*. The elements used in embedding are the bands of frequency and the *planes of subcomponents*.

The term *band of frequency* describes a set of coefficients in each wavelet subband partition described before.

The planes of subcomponents are formed with the values of the subcomponent for every subband. According to this, the planes of subcomponents have as much elements as the subbands, but the resolution of the elements of the subband matches to the resolution of the used subcomponent.

The *alignment* is the shifting of the planes of subcomponents of the subbands referred to the planes of subcomponents of other subbands. The process of alignment is similar to the process of quantification, because each subband is treated in a different way.

The alignment process places (arrange) the planes of subcomponents depending on an alignment algorithm. The types of alignment fix the relations among the different planes of subcomponents. For example, the figure 4 show a normalized alignment.

The result of the process of alignment are the *importance levels*. The importance level is the minimum unit of codified data with a meaning in bit stream generated. Before the entropy coding, the importance level is a plane of subcomponent or a group of them of the embedded data.

Each importance level will be identified with a number, an index named *order of the importance level*. The importance levels with greater weigh (more significant) will have an smaller order, and importance levels of smaller weigh (less significant) will have a bigger order. The first order is always the zero order, and it matches with the plane of subcomponent of greater weigh of the LL subband. The order number, the total number of orders of importance levels and the

subcomponents that correspond to each level of importance depend on type of alignment that is being used.

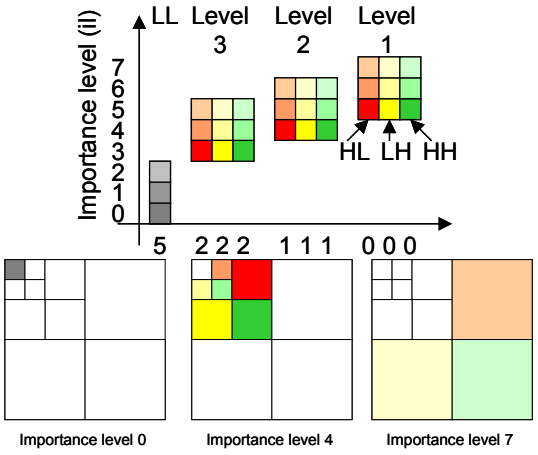

Fig. 4 Normal alignment

F. Variable Bit Length Coding Modified

The *VBLm* (*VBLmodified*) codification method is a modification of the VBL system (*Variable Bit Length coding*) introduced by Sahni and Vemuri [10].

The steps to follow in the encoding are four:

1. *Create segments*: set consecutive symbols with the same resolution.
2. *Create informations*: indicate the identifying data of each segment.
3. *Group informations*: process of compression itself. It a combine algorithm.
4. *Code the sequence*: creation of the coded frame.

The *information* are pairs of data of each segment. These data are:

a) *resolution*: bits per symbol (b_i). The resolution contains the minimum number of bits necessary to encode the symbols of the segments. It is coded with four bits, so the minimum resolution is 0 and the maximum 15.

b) *repetition*: segment lengths (l_i).

Every information is coded in a group of 8 bits. So, each segment generates an overload 8 bit of header for the codification of the information.

The total length corresponding to the codification of a segment would be of 1 byte (8 bits) of the information plus $l_i \times b_i$ of the sequence of symbols.

The total number of bits needed to code n segments is:

$$S_n = 8\,n + \sum_{i=1}^{n} l_i\, b_i$$

The *combine algorithm* becomes a seeking algorithm. The combine algorithm is applied directly to the informations and not to the segments. Talking about $S_q^{\min}$ as the space required for an optimum combination of the first q segments and defining $S_0^{\min} = 0$.

$$S_n^{\min} = \min\left\{ S_{n\ \min}^{m,r} \right\}$$

In a VBLm encoding, the total number of bits needed to code a sequence of n final segments/informations is of:

$$R_T = 2 + n + \left(\sum_{i=1}^{n} l_i b_i + 7\right)/8$$

The structure of the bit stream for a importance level after the VBLm encoding is shown in figure 5.

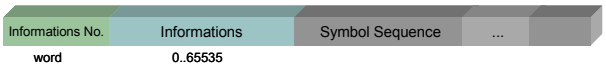

Fig. 5 Stream VBLm format

III. High Level Scalability System

As a result of the encoding done using the method used till the moment, the generated bit stream is structured in importance levels coded by means of the VBLm algorithm. The length of the coded importance levels is variable and very different depending on the size of the image, the levels of wavelet partitioning, the energy scattering of the image and the alignment algorithm used. Since, given that this algorithm generates an embedded bit stream, the trunk of the bit stream anywhere in it doesn't hinder the complete rebuilding of the image although there appears higher or lower degradation depending on the amount of energy the truncated bit stream keeps. However, the objective of the scaling algorithm is to generate several bit streams from the original binary in a way that makes possible to give size in an specific way the number and the size of the basic and improvement original bit streams. Each bit stream generated is named *packet*. The scaling algorithm becomes then a packing algorithm.

The first packet generated matches to the scaling basic level, and the other packets are enhance levels.

Each packet generated contains one, part or several importance levels coded using the same structure used by VBLm. Although, owe to the concrete spatial definition represented with the symbols of the bit train resulting from VBLm coding it is possible to rebuild in a full way or partially the wavelet coefficients. Having into account the high spatial component of the wavelet transformation, it is possible to rebuild partially the original image without having all the coefficients. So, we can say that each packet generated is an elemental and consistent of image that can rebuild on its own rebuild the whole image with higher or lower precision depending on the number and precision of the components it is able to rebuild.

A. Packing Process

The packing process choose the VBLm informations necessary to fill the size of the bit stream of the packet from the VBLm bit streams of consecutive importance levels.

The order of importance levels is kept in the process of the packaging, in such a way that if a packet doesn't fill a importance level, that level will be filled up, if it is possible, in the next packet.

The selection of VBLm informations is done depending on

their resolution, higher resolution implies higher energy and higher image information. The informations are chosen sequentially depending on their position into the level of importance treated.

The rejected informations are treated as if they had null resolution in such a way that, although they are not an extra load for the final bit stream, they have payload.

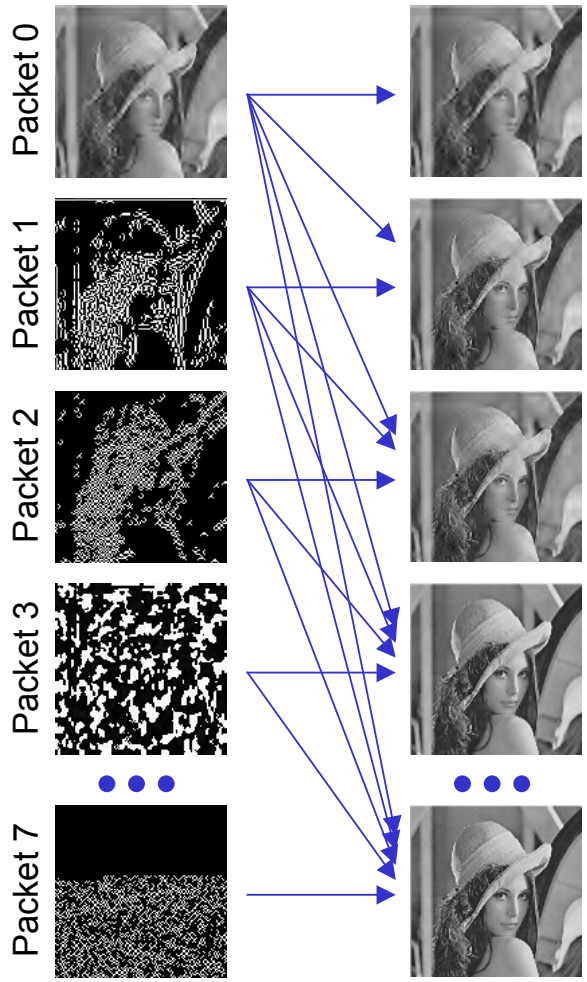

Fig. 7 Scaled images

B. Packet Size

The packing process here shown permits the generation of packets with defined lengths. It is possible to fix the size of the basic packet and of the enhanced levels in the process of setting of the packet. If there is only one enhanced level, it is possible to generate a packet with remaining information after the basic level. If there is no basic level it would work with different consecutive enhanced levels.

In the packing process, the rejected informations are treated like zeros, that are presented as informations, but not like symbols. That is, the packaging system produces additional overload on the compression process.

If a packet is truncated because of problems into the transmission or problem to store it, the remaining information is consistent owe to the characteristics of the VBLm coding.

C. Basic and Scaled Images

The packets generated after the packing process can be grouped into *images*. There can be distinguished two types of images, the basic and the scaled images. The basic images contain the packet with the basic scaling level and present an scaling level of 0. The scaled images contain only improvement packets and present an scaling level greater than zero.

So, the final bit stream of basic and scaled images presents the structure shown in figure 8.

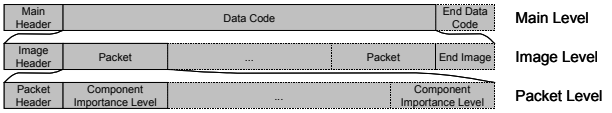

Fig. 8 Image structure

We can see in the figure 7 an example of an image scaled codification. In this example an image sequence code one lossless original image (each image have within an alone packet).

REFERENCES

[1] J.M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients" *IEEE transactions of Signal Procesing*, vol. 41, pp. 3445-3462, Dec. 1993.

[2] A. Said and W.A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees" *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 243-250, Jun. 1996.

[3] M.P. Boliek, M.J. Gormish, E.L. Schwartz and A.F. Keith, "RICOH CREW Image Compression Standard" *RICOH Silicon Valley, Inc.*, Mar. 1999

[4] ISO/IEC, ITU-T, "Information technology – JPEG2000 image coding system" *ITU-T Rec. T800, ISO/IEC 154444-1*, 1999.

[5] ISO/IEC, "Information technology – Generic coding of audio-visual objects: part 2 visual" *ISO/IEC 14486-2*, 2003.

[6] S. Mallat, "A wavelet tour of signal processing. Second edition", *Academic Press, San Diego*, 1999

[7] R. Calderbank, I. Daubechies, W. Sweldens and B.L. Yeo, "Wavelet transforms that map integers to integers", *Journal of Applications and Components*, vol. 5, 1998

[8] W. Sweldens, "The lifting scheme: Construction of second generation wavelets", *SIAM Mathematical Analisys*, vol. 29. No. 2, pp. 511-546, 1997

[9] M.D. Adams and F. Kossentini, "Reversible integer-to-integer wavelet transform for image compression: Performance, evaluation and analysis" *IEEE Transactions on Image Processing*, vol. 9, no. 6, Jun. 2000.

[10] S. Sahni, B.C. Vemuri, F. Chen, C. Kapoor, C. Leonard, J. Fitzsimmons, "State of the art lossless image compression algorithms", *IEEE Proceedings of the International Conference on Image Processing*, Chicago, Illinois, pp. 948-952, Nov. 1998