IMPROVING LOSSY IMAGE COMPRESSION WAVELET TRANSFORM BASED ALGORITHMS BY PREDICTING DISCARDED COEFFICIENTS

Roberto Gil Pita, Manuel Rosa Zurera, Fernando Pérez Cruz

Dpto. de Teoria de la Señal y Comunicaciones - Escuela Politécnica - Universidad de Alcalá
Ctra. Madrid-Barna, km. 33.6, 28871 Alcalá de Henares - Madrid - Spain
e-mail: robgilpita@eresmas.com, manuel.rosa@uah.es

ABSTRACT

Most of lossy image compression methods like SPIHT or JPEG2000 are based on determining the intervals in which each coefficient can be reached. Once these intervals are known, the decoder selects the value that minimizes the mean square error of the possible values. When a coefficient is not considered significant, because its sign is not known, the selected reconstruction value is zero. This ignorance of the sign for non significant coefficients makes the interval of the possible values to be twice as much the rest of the Wavelet coefficients.
In this paper an alternative for the selection of the reconstruction value for the non significant coefficients is analyzed. It is based on predicting their actual values using the neighbors' values. With this new improvement it has been possible to reach average reductions of about 5 percent, being higher than reductions reached by JPEG2000 over SPIHT.

1 Introduction

The JPEG committee has recently released its new image coding standard, JPEG 2000 [1] [2], which will serve as a supplement for the original JPEG standard introduced in 1992. This new coding system achieves excellent compression performance, somewhat higher (and, in some cases, substantially higher) than SPIHT [3] with arithmetic coding, a popular benchmark for comparison purposes based on the EZW algorithm [4].

These three algorithms are based on the Wavelet transform [5], The Wavelet coefficients are quantized and entropy coded in order to form the output bit stream. At the decoder, the bit stream is first entropy decoded and dequantized. The result of these two operations is a set of possible values for each coefficient. So as to select the correct value from the possible ones, trying to minimize the mean square error is usual. Once the estimation of the coefficients is obtained, the inverse transform is applied, thus resulting in the reconstructed image data.

The quality loss is introduced on the quantization step. Low coefficients are considered zero valued, ignoring the sign of the coefficient in the quantization. So, the interval of the possible values of these coefficients is twice the rest of the Wavelet coefficients. The quality loss introduced by this kind of coefficients is, therefore, higher than the left ones.

We propose an alternative method for estimating the reconstruction value for these non-significant coefficients based on predicting their real values by their known valued neighbors. On the next section, we will study the statistical conditions of these values, building up from the study of the sign. If we could determinate the sign of these coefficients, it would be possible to choose the value with more accuracy and the global quality of the recovered image would be higher.

2 Estimation of the residual value of the non-significant coefficients

This section is divided in two subsections. The first one introduces the concepts necessaries to the understanding of our improvement. And the second one explains our alternative method.

2.1 Autocorrelation of the signs of the Wavelet coefficients

The Wavelet decomposition is carried out by bidimensional filtering and decimation. The spectrum is divided in four areas. Repeating this process, we are selecting the frequency areas, adjusting them to make the positive valued frequencies embrace the whole spectrum. According to this, we are passing from a frequency domain with some determinated characteristics in the original image, to blocks of coefficients where each scale and orientation possess some frequency characteristics that are a portion of the original ones. This way there will be soft shaped images producing scales with an important component of high frequency.

On the other hand, at least a high pass filtering is used for obtaining the different Wavelet coefficients. So, the mean value of each an every orientation and scale is zero. Putting together these two characteristics of the Wavelet coefficients, we can conclude that statistical properties in each orientation and scale are different. Therefore, this will produce correlations in the signs of the coefficients. Either to (a) have a high degree of high frequencies and, therefore, a lot of probability of changing sign in adjacent coefficients, either to (b) have low frequencies and low probability of a sing change. In addition, these properties will remain uniform for every orientation and scale, being possible to find different conditions on each direction. All this is demonstrated by calculating the autocorrelation of the signs of the Wavelet coefficient in each orientation and scale. Finding out that, depending on the selected one, there are high or low correlations, positive or negative for each neighbor of the coefficient.

We can summarize, therefore, that the sign of a coefficient is correlated with the sign of its neighbors, so that it is feasible to predict its sign with those around. When we finish the image decoding, the result is a bunch of coefficients (with a precision marked by the last threshold), with a reconstruction value that minimizes the mean square error, surrounded of a set of coefficients that, not being able to know their sign, a reconstruction value of zero and an interval of precision twice as much as that of the rest of the coefficients are assumed. If we could know the sign of these non-significant coefficients, a reconstruction value different from zero would subsequently diminish the square error in a remarkable way.

2.2 Obtaining the residual value estimation coefficients

We will take over the problem in the following way: once finished the encoding we will look for the non-significant coefficients, and we will select among them, those having at least one significant neighbor. The neighbors to be considered will be the eight adjacent coefficients in the horizontal, vertical and diagonal directions.

As seen in the previous subsection, the statistical properties of the signs are different for each orientation and scale. We will consider the scales with 64x64 coefficients or more. For an image of 512x512 pixels, we have three 256x256 coefficients groups for the first scale, three 128x128 coefficients groups and three 64x64 groups for the third scale. We don't consider bigger scales for the gain in the usual compression rates for not being significant.

This way, we will focus on the analysis of a chosen scale and orientation. For each group, we will have N non significant coefficients with at least one significant neighbor. x_n will be the residual value of the n coefficient. The neighbors of x_n are the adjacent eight ones numbered from left to right and from top to bottom, and s_n^i ($i = 0$ ...7, $n = 0$ ...N-1) are the sign of the i neighbor of the n coefficient. s_n^i will be +1 for significant and positive coefficients, -1 for significant and negative ones and 0 for non-significant ones. To calculate the prediction value of a non-significant coefficient, its eight neighbors will be taken and multiplied by certain a_i coefficients to obtain the normalized reconstruction value. Then we only have left multiplying by the threshold to obtain the estimation of actual value for the point. To obtain the a_i values, minimization of the mean square error is carried out.

According to this, our approach to the problem should be the following: Being e_n the derived error of the estimation of the n coefficient, we can write:

$$e_n = x_n - \sum_{i=0}^{7} s_n^i \cdot a_i \cdot U_n \tag{1}$$

Where U_n is the threshold reached by the coefficient n. We must take into account that the algorithm can be stopped at any given moment, so that some coefficients' threshold shouldn't necessary be the same one than others'. This will be managed by the factor A. Therefore, now we have a system of N equations with eight variables that will be solved by the least square method. Thus, the square error will be the following:

$$\sum_{n=0}^{N-1} e_n = \sum_{n=0}^{N-1} (x_n - \sum_{i=0}^{7} s_n^i \cdot a_i \cdot U_n)^2 \tag{2}$$

In order to find the values a_i that minimize this expression, we should carry out the partial derivative regarding each a_i. This way we will obtain a system of eight equations with eight variables to be solved.Taking the partial derivative of the previous expression regarding to a_j, j equations are obtained:

$$\frac{\partial \sum_{n=0}^{N-1} e_n}{\partial a_j} = \sum_{n=0}^{N-1} 2 \cdot (x_n - \sum_{i=0}^{7} s_n^i \cdot a_i \cdot U_n) \cdot s_n^j \cdot U_n = 0 \tag{3}$$

$$\sum_{n=0}^{N-1} x_n \cdot s_n^j \cdot U_n = \sum_{n=0}^{N-1} \sum_{i=0}^{7} s_n^i \cdot a_i \cdot U_n^2 \cdot s_n^j \tag{4}$$

Now we consider A as $T \cdot V_n$, where V_n is worth 1 or 2, depending whether this coefficient has already been tested in the current algorithm loop or not, and T will be the last obtained threshold. Inverting the order of the sums and re-arranging the expression we obtain expression (5):

$$\frac{1}{T} \cdot \sum_{n=0}^{N-1} x_n \cdot s_n^j \cdot V_n = \sum_{i=0}^{7} a_i \cdot \sum_{n=0}^{N-1} s_n^i \cdot V_n^2 \cdot s_n^j \tag{5}$$

Naming S_n^i the product $s_n^i \cdot V_n$, we obtain:

$$\frac{1}{T} \cdot \sum_{n=0}^{N-1} x_n \cdot S_n^j = \sum_{i=0}^{7} a_i \cdot \sum_{n=0}^{N-1} S_n^i \cdot S_n^j \tag{6}$$

If $R_{ij} = \sum_{n=0}^{N-1} S_n^i \cdot S_n^j$ and writing the eight resulting lineal equations in a matrix form:

$$\begin{pmatrix} R_{00} & R_{01} & \dots \\ R_{10} & R_{11} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ \vdots \end{pmatrix} = \frac{1}{T} \cdot \begin{pmatrix} \sum_{n=0}^{N-1} x_n \cdot S_n^0 \\ \sum_{n=0}^{N-1} x_n \cdot S_n^1 \\ \vdots \end{pmatrix} \tag{7}$$

Once coding is concluded, the inverse of the matrix $\bar{\bar{R}}$ and the vector in the right side of equation (7) is calculated, and multiplied in order to obtain the vector $\bar{A}$. These eight coefficients are always smaller than unity, and they will be coded using a floating point representation and sent them to the receiver. So, once the decoding is concluded, they will be read from the file, and each scale and orientation non significant coefficients will be tested looking for those having significant neighbors. The reconstruction value for each non significant coefficient will be:

$$\hat{x_n} = T \cdot \sum_{i=0}^{7} s_n^i \cdot a_i \cdot V_n \tag{8}$$

To code the coefficients, a structure of one byte has been used, where the first bit represents the sign of the element, next three bits the exponent and last five bits the mantissa. We must take into account that the inclusion of these coefficients must be considered in coding, being $8 \cdot 8 \cdot 3 \cdot 3 = 576$ bits added at the end of the process.

3 Results

In table 1 we have the gain represented for different compression rates using the image Lenna, for each scale and orientation. Gain is expressed maintaining the same compression rate, therefore negative values may appear. So as to calculate the compression gain, an estimation of the slope of the bpp-PSNR curve has been made, so that it has been possible to change increments in PSNR to increments in bpp.

In tables 2 and 3 reduction and gain achieved with the application of this approach to different images are shown. As we can see, the gain is not stable for all rates; it is centered on certain values of PSNR. On the other hand, as this values of PSNR are usual, we find out a method that gives out some very good results (up to 7% in some cases). On the other hand, another conclusion is that the variations are quite abrupt. To achieve the maximum gain with this method, the best option is to stop the coding at the changes of bit plane. According to this method, we can reach higher gains.

This is shown in table 4, where we can see the best reduction values for the working images. This time it is a real reduction, since we targeted that no matter what number of bits are used, image quality is exactly the same for both algorithms. As we can see, the results reached by the method begin to be comparable to

	Bit plane 5	Bit plane 4	Bit plane 3
Scale 0 Hor.	0,001 dB	0,031 dB	0,037 dB
Scale 0 Dia.	-0,006 dB	0,004 dB	0,015 dB
Scale 0 Ver.	0,066 dB	0,071 dB	0,036 dB
Scale 1 Hor.	0,024 dB	0,016 dB	0,004 dB
Scale 1 Dia.	0,004 dB	-0,001 dB	0,000 dB
Scale 1 Ver.	0,011 dB	0,011 dB	0,004 dB
Scale 2 Hor.	0,005 dB	0,004 dB	0,000 dB
Scale 2 Dia.	0,002 dB	-0,001 dB	0,001 dB
Scale 2 Ver.	0,013 dB	0,000 dB	0,000 dB
Total	0,118 dB	0,136 dB	0,098 dB
Reduction (%)	2,712%	3,146%	1,867%

Table 1: Gain for each scale and orientation using the image "lenna" 512x512 (dB)

Bpp	Barbara	Finger	Goldhill	Lenna	Peppers
0,1	2,5%	0,5%	1,1%	0,3%	0,5%
0,2	1,3%	2,4%	3,8%	1,4%	1,4%
0,5	1,7%	2,3%	2,8%	1,1%	6,2%
1,0	1,5%	5,3%	1,6%	0,3%	2,8%

Table 2: Reduction for different images using different rates (%)

Bpp	Barbara	Finger	Goldhill	Lenna	Peppers
0,1	0,072	0,014	0,032	0,014	0,017
0,2	0,050	0,094	0,125	0,065	0,031
0,5	0,114	0,116	0,130	0,051	0,154
0,75	0,247	0,126	0,066	0,064	0,134
1,0	0,121	0,368	0,098	0,014	0,188

Table 3: Gain in PSNR for different images using different rates (dB)

low	Barb.	Finger	Goldh.	Lenna	Peppers
Rate (bpp)	0,144	0,510	0,081	0,078	0,070
PSNR (dB)	26,34	28,01	27,53	29,37	28,83
Reduction	6,42%	2,89%	3,17%	0,52%	1,10%
medium	**Barb.**	**Finger**	**Goldh.**	**Lenna**	**Peppers**
Rate (bpp)	0,669	1,054	0,527	0,168	0,401
PSNR (dB)	34,82	32,33	33,68	32,64	33,27
Reduction	4,47%	6,83%	3,67%	2,98%	7,15%
high	**Barb.**	**Finger**	**Goldh.**	**Lenna**	**Peppers**
Rate (bpp)	1,162	1,934	1,164	0,341	1,153
PSNR (dB)	39,20	37,59	37,87	35,81	37,53
Reduction	2,25%	3,69%	2,44%	3,38%	6,65%

Table 4: Best cases for different images using different rates in low, medium and high quality

those achieved by Said and Pearlman with their algorithm SPIHT over the one proposed by Shapiro (about 20%). In addition to that, the best results are gotten in the most commonly used areas in image compression (between 30 and 40 dB).

Besides, when comparing two images recovered with different methods, we can notice that they are different, although having the same quality. When implementing the prediction, we alter many values that are distributed by the first three scales and, therefore, we modify the high frequencies in most of the image shaping out most of the hardest contours. In the original method, however, symbols have been taken out until reaching the same quality, which means modifying the value of a small number of coefficients, but in a significant way. In other words, it adjusts some areas of the image neglecting others.

4 Conclusions

The objective of this paper is the search of possible improvements to the well known last generation image coding standards. Throughout our work we have tried to explain the statistical properties of the residual values. We have proposed a new alternative for determining the values of the wavelet coefficients at the decoder.

Using prediction schemes we have obtained higher rate reductions. But the correct way for understanding these results is to compare them with those reached by JPEG 2000. The gain of JPEG 2000 over SPIHT is 0,05 dB with Lenna. With our approach, gain over SPIHT with Lenna is 0,06 dB, which is about the same amount.

The coding gain is not related to the coding scheme, so similar quality improvements should be obtained applying our approach to the JPEG 2000 standard.

References

[1] "Jpeg 2000 part 1: Final draft international standard (iso/iec fdis15444-1)," *ISO/IEC JTC1/SC29/WG1 N1855*, August 2000.

[2] "Jpeg 2000 requeriments and profiles," *ISO/IEC JTC1/SC29/WG1 N1271*, March 1999.

[3] William A. Pearlman Amir Said, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Tecnology*, vol. 6, no. 3, pp. 243–249, June 1996.

[4] Jerome M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on image processing*, vol. 41, no. 12, pp. 3445–3462, December 1993.

[5] Pierre Mathieu Ingrid Daubechies Marc Antonini, Michel Barlaud, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205–220, April 1992.