Applying a Genetic Algorithm Solution to Improve Compression of Wavelet Coefficient Sign

Antonio Martí[1], Otoniel López[2], Francisco Rodríguez-Ballester[1], and Manuel Malumbres[2]

[1] Universidad Politécnica de Valencia, Camino de Vera s/n, Valencia Spain,
[2] Universidad Miguel Hernández, `otoniel@umh.es` Avda. Universidad s/n 03205, Elche, Spain

Abstract. Discrete Wavelet Transform has been widely used in image compression because it is able to compact frequency and spatial localization of image energy into a small fraction of coefficients. In recent years coefficient sign coding has been used to improve image compression. The potential correlation between the sign of a coefficient and the signs of its neighbors opens the possibility to use a sign predictor to improve the image compression process. However, this relationship is not uniform and constant for any image. Typically, image encoders use entropy coding to compact the wavelet coefficients information. This work analyzes the impact of the input parameters over a genetic algorithm that distributes into contexts (sets) the wavelet sign predictors in such a way that the overall aggregate entropy will be reduced as much as possible and a as a consequence higher compression rates can be achieved.

Keywords: Genetic Algorithm, Wavelets, image coding, sign coding

1 Introduction

Genetic algorithms were first introduced by Holland in [1] and they are nowadays well known and very popular to find optimal/suboptimal solutions in very large and complex problems [2].

In a genetic algorithm, the evolution usually starts from a population of randomly generated individuals and runs in generations. In each generation the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness value), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached.

In this work we are looking for an optimal/suboptimal solution of our particular problem related with wavelet image compressors. This kind of image compressor is based in the use of a mathematical transform called Discrete Wavelet

Transform (DWT). Wavelet transforms have proved to be very powerful tools for image compression, since many state-of-the-art image codecs, including the JPEG2000 standard [3], employ DWT into their algorithms. The energy of a wavelet transform coefficient is restricted to non-negative real numbers, but the coefficients themselves are not, and they are defined by both a magnitude and a sign. Shapiro stated in [4] that a transform coefficient is equally likely to be positive or negative and thus one bit should be used to encode the sign. In recent years, several authors have begun to use context modeling for wavelet sign coding [5–7], showing that despite the equiprobability of wavelet sign values, some sign correlation can be found among wavelet coefficients, resulting in overall compression ratio improvements.

In a previous work [8] we have observed that the sign of a wavelet coefficient may be strongly correlated with the sign of some neighbor coefficients. However, this relationship is not uniform and constant for any image, or even consistent within the same image. By increasing the number and kind of images under analysis, the relationship between the signs of the neighbor coefficients may be generalized or, on the other hand, it is possible that, when increasing the number of images, some relationships become contradictory.

In this work, after obtaining the sign prediction for a given neighborhood and its success probability, we will use a Genetic Algorithm (GA) to distribute all the neighborhood permutations into r sets in such a way that the overall aggregate entropy will be reduced as much as possible and a as a consequence higher compression rates can be achieved by the entropy encoder used in the wavelet-based image encoder. We will analyze the impact of the input parameters in the performance of the GA.

The paper is organized as follows: in section 2 we define the optimization problem. In section 3 we propose a GA algorithm that matches the problem definition and we perform a detailed analysis GA parameters that supply the best sign context distribution in the shortest time. The paper then presents the main results in section 4 and finally general conclusions are highlighted.

2   Context-based sign coding approach

Most wavelet image codecs do not consider the use of sign coding tools since the wavelet coefficients located at the high frequency subbands form a zero-mean process, and therefore they are equally likely positive as negative.

Schwartz, Zandi and Boliek [9] were the first authors to consider wavelet coefficient sign coding, using the sign of one neighboring pixel in their context modeling algorithm. The main idea behind this approach is to find correlations along and across edges.

The HL subbands of a multi-scale 2-D wavelet decomposition are formed from low-pass vertical filtering and high-pass horizontal filtering. The high-pass filtering detects vertical edges, thus the HL subbands mainly contain vertical edge information. Oppositely defined are the LH subbands that contain primarily horizontal edge information.

As Deever explained in [10], given a vertical edge in an HL subband, it is reasonable to expect that neighboring coefficients along the edge have the same sign as the coefficient being coded. This is because vertical correlation often remains very high along vertical edges in images. When a low-pass filter is applied along the image columns, it results in a series of similar rows, as elements in a row tend to be very similar to elements directly above or below due to the high vertical correlation. Subsequent high-pass filtering along similar rows is expected to yield vertically correlated transform coefficients.

It is also important to consider correlation across edges, being the nature of the correlation directly affected by the structure of the high pass filter. For Daubechies' 9/7 filters, wavelet coefficient signs are strongly negatively correlated across edges because this filter is very similar to a second derivative of a Gaussian as derived from theory of zero crossings and edge detection [11]. So, it is expected that wavelet coefficients will change sign as the edge is crossed. Although the discrete wavelet transform involves sub sampling, the sub sampled coefficients remain strongly negatively correlated across edges. In this manner, when a wavelet coefficient is optimally predicted as a function of its across-edge neighbors (e.g. left and right neighbors in HL subbands), the optimal prediction coefficients are negative, indicating an expected sign change. This conclusion is general for any wavelet with a shape similar to a second derivative of a Gaussian.

In Fig. 1 we plot the spatial distributions of signs in the HL subband of two popular test images: Barbara and Lena. The visible sign structures suggest that the sign bits of wavelet coefficients are compressible.

(a) Sign map for Barbara   (b) Sign map for Lena
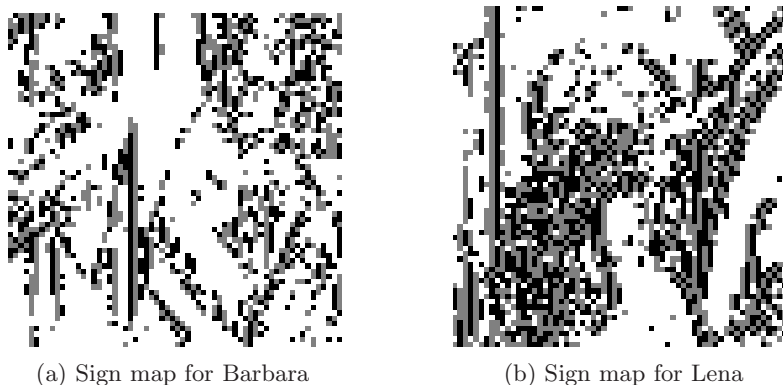
Fig. 1. Sign patterns in the HL subband for Barbara (left) and Lena (right). Black, gray and white dots for negative, positive and non-significant coefficients, respectively

To estimate sign correlation in a practical way, we have applied a 6-level Dyadic Wavelet Transform decomposition of the source image and then a low quantization level to the resulting wavelet coefficients. Taking into account that the sign neighborhood correlation depends on the subband type (HL,LH,HH) as

Deever assesses in [7], we have used different neighbors in each subband type exploiting the correlation along and across the edges (see Table 1). So, for a particular subband type, we have defined n neighbors that can hold one of the three possible sign values, that is positive, negative and null (zero). This lead us to a set of 3^n different Neighbor Sign Patterns (NSP) for each subband type, being n the number of neighbors used in the sign prediction.

Neighborhood	HL subband Neighbors	LH subband Neighbors	HH subband Neighbors
3	N, NN, W	W, WW, N	N, W, NW
4	N, NN, W, WW	W, WW, N, NN	N, W, NW, NNWW
4b	N, NN, NNN, W	W, WW, WWW, N	N, W, NW, NNWW
5	N, NN, NNN, W, WW	W, WW, WWW, N, NN	N, W, NW, NNWW, NNNWWW

Table 1. Neighborhood analyzed

Other encoders like JPEG2000 and the one proposed by [12] use four neighbors (N,S,E,W) for the context formation, but since most non-embedded encoders use a Morton order (Z-order)[13] in the coding stage, no information is available about S and E neighbors and they cannot be employed in context formation. This represents a restriction when looking for sign correlation among the neighborhood which it is shared by most of the non-embedded encoders.

So, for each subband type and each neighborhood we can obtain the sign prediction table that contains the sign predictions, the number of occurrences and the probability of success for every $NSP[k]$, $k = 1..3^n$. Then, when coding the sign of a wavelet coefficient in a particular subband, first we will get the sign value of the corresponding neighbor set in order to form the current NSP. Then we will look up this NSP in the table to find the sign prediction of the current wavelet coefficient. Finally, what we are going to encode is the correctness of this prediction. The performance of our binary entropy encoder will depend on the behavior of our sign predictor, the higher success prediction rate, the higher compression rate is achieved.

In order to improve the final compression performance of our entropy encoder, we propose the use of up to ten contexts for each subband type. So, for each subband type, we distribute the provided NSPs predictions into r sets (contexts) in such a way that the overall aggregate entropy will be reduced as much as possible and a as a consequence higher compression rates can be achieved. We have fixed to ten as a maximum the number of context in order not to increase to much the image encoder complexity, as in [6] where authors reduced and fixed the context number to five. We will use a GA algorithm to distribute all NSPs into r sets.

3   Genetic Algorithm for Wavelet Sign Prediction

The goal of our GA proposal will be to find a good context distribution of the NSPs prediction table for each subband type SB_t, where $t = \{HL, LH, HH\}$. So, for a particular wavelet coefficient $C_{i,j}$ with sign value $SC_{i,j} = \{+, -, NULL\}$, that belongs to subband SB_t, the prediction table will provide a sign prediction, $\hat{SC}_{i,j}$, based on its Neighborhood Sign Pattern, $NSP[k]$, where k represents the specific sign pattern of the neighborhood of the current wavelet coefficient, $C_{i,j}$ and the set (context) number where the NSP will be assigned for the entropy encoder.

There is no univocal relationship between a neighbor sign permutation and the sign prediction, i.e not always for a same $NSP[k]$ pattern, the sign prediction of the current wavelet coefficient, $\hat{SC}_{i,j}$, is always positive or negative. However, it is possible to find out that for a particular $NSP[k]$ sign pattern, $\hat{SC}_{i,j}$ is more probably to be positive than negative or vice versa. But, the problem is still more complex, because a sign prediction for a neighbor sign pattern could fit well for an image and not for others. Evenmore, the context distribution and the number of context to use will affect the encoder compression performance. Therefore, the idea is to find suboptimal neighbor sign pattern predictions and context distribution that better fit for a representative set of images, so we can capture the canonical wavelet sign redundancy introduced by a particular wavelet filter. In this manner, once the universal prediction table is found, it could be used at both encoder and decoder sides.

At this point, the motivation of using GAs to compress the sign of wavelet coefficients is twofold. First, when the number of selected neighbors for the analysis of sign correlation grows or when there is a great number of images to be used in the analysis and also when the number of context to distribute the NSPs in grows, the search space is excessively wide. Second, it is not intuitive to find a way of combining the predictions obtained for several images, and mix them in an unique prediction table.

The context distribution problem is similar to the mathematics problem of finding the ways to partition a set of k objects into r sets. The number of possible partitions is called Stirling number of the second kind [14] and is denoted by $S(k, r)$ (see equation 1). For example, if we use three neighbors for the sign prediction we have 27 NSPs (3^3) and distributing then into 5 contexts, the Stirling number of the second kind is 613382071584090090, which means that we have that number of possibilities to distribute the 27 NSPs into 5 contexts.

$$S(k, r) = \frac{1}{r!} \sum_{j=0}^{r} (-1)^{r-j} \binom{r}{j} j^k \qquad (1)$$

## 4   Genetic Algorithm Design and Analysis

In this work, we analyze the effect of several parameters over the results (fitness and convergence) obtained by the GA. The parameters analyzed are:

- Population size
- Mutation probability
- Neighborhood
- Context number (sets)

   We have executed the GA using the following variations. For each execution we have performed 100 iterations because of the influence of the random seed over the generation of the best individual. This lead us to a total of 100,000 executions.

- Population size: 10, 25, 50, 100, 150, 200, 300, 400
- Mutation probability: 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001 (from 5% to 0.1 ‰).
- Neighborhood: 3, 4, 4b, 5
- Context number (sets): 4, 5, 6, 7, 8, 9

   In Fig. 2 we show the normal probability graphic for the fitness values. As it can be seen, they are not normal and so we will use other kind of analysis, mainly based on dispersion graphics.
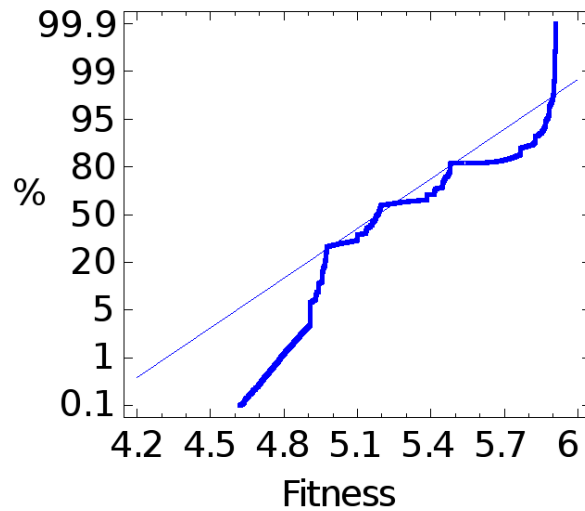
Fig. 2. Normal probability graphic

   In Fig. 3 we show the best result obtained in each GA execution, where the results are ordered as a function of the GA parameter values used in the the following script.
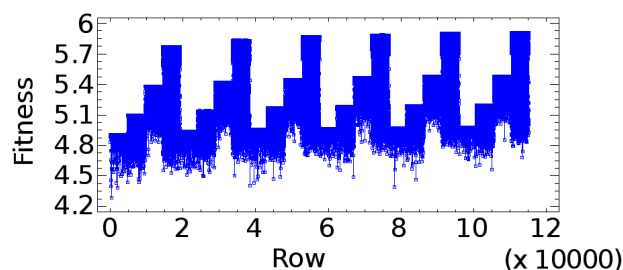
Fig. 3. Chronological sequence of GA fitness results

```
For context (from 4 to 9) do
 For neighbors in (3; 4; 4b; 5) do
  For population_size in (10; 25; 50; 100; 150; 200; 300; 400) do
   For mutation_probability in
      (0.05; 0.01; 0.005; 0.001; 0.0005; 0.0001) do
      Run GA #images.txt #contexts #neigbors
          #population_size #mutation_probability
```

where images.txt contains the wavelet coefficients sign prediction for each NSP and image set.

In Fig. 3 we can see 6 successive stairs. All stairs have a similar shape, but the results are different on each stair. The number of stairs correspond to the number of contexts analyzed, i.e 4, 5, 6, 7, 8, 9. Looking at that figure we can conclude that we obtain a better compression as the number of context used in the entropy encoder increases and also, that the rest of parameters has nearly no influence over context parameter.

In each stair, there are 4 steps which corresponds to the neighborhood analyzed, i.e 3, 4, 4b and 5. The effect of this parameter is quite significative and it is represented by the difference between steps. This difference can be seen on both the minimum and maximum fitness values. At each step or neighborhood we can appreciate an upper limit which is the maximum compression independently of the rest of GA parameters (population size and mutation probability). Obviously, this limit is higher for the neighborhood that uses 5 neighbors. Otherwise, there is a lower bound on each step. The lower bound for 3 and 4b neighborhood is quite similar, being this lower bound higher for the neighborhood 4. This effect is more clearly shown in Fig. 4 where fitness values are represented as a function of the neighborhood. At this point, we could not asses which neighborhood, 4b or 5 is the best one to be used in the compression algorithm because neighborhood 5 obtains better results but with a greater dispersion than neighborhood 4b. Moreover, compression algorithm complexity increases as the number of neighbors does.

The results variation in each neighborhood shown in Fig. 4 is due to the intrinsic GA parameters like population size and mutation probability. In order
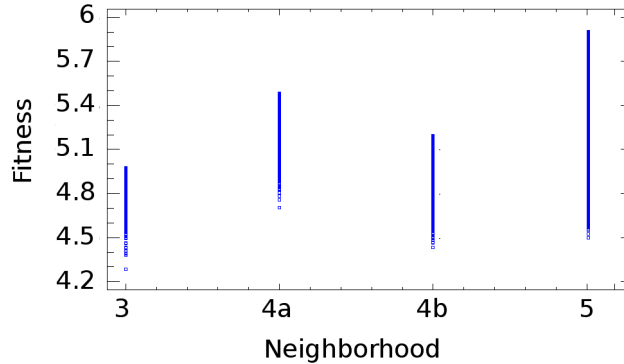
Fig. 4. Dispersion fitness vs neighborhood

to easily analyze the results, we will reduce the analysis to the context number 6 and neighborhood 4b and 5. In Fig. 5 (a) and Fig. 5 (b) we can see the dispersion graphic for the fitness values as a function of the mutation probability for those executions that used the context number 6. Fig. 5 (a) represents executions using 4b neighborhood and Fig. 5 (b) represents executions using 5 neighborhood. As it can be seen for the range from 0.0005 to 0.005, the mutation probability does not affect to the fitness values obtained by the GA. Using lower mutation probability it is possible to obtain suboptim results, but an important number of executions obtain very low fitness values. On the other hand, with mutation probabilities higher than 0.005 the algorithm fluctuates in a random way and the fitness results get worse considerably, both the maximum value and the mean value.

Analyzing the results when the mutation probability values are in the range from 0.0005 to 0.005, we can observe that results are grouped (gathered), i.e. there are less variability and also we can observe that we obtain better results when using neighborhood 5.

Finally we will analyze the effect of population size over the fitness results. In Fig. 6 we can see the dispersion graphic for all executions and as shown there is a great variability in the results, but this variability and also the range are quite similar for all population sizes except for population size of 10 that presents the worst results.

Previous analysis show that the best results are obtained for the neighborhood 5 and mutation probabilities from 0.0005 to 0.005. Fig. 7 shows the dispersion graphic for mutation probabilities in the range of 0.0004 to 0.006 using neighborhood 5 and 6 contexts. We can see three groups of results. First, with a population size of 10 individuals, the GA is not able to achieve as good results as when using greater population sizes. Secondly, population sizes of 25, 50 and 100 individuals present optimal and identical results. Third, for populations sizes of 150, 200, 300 and 400 the graphic does not allow to see if the results are as good as for the previous group. In this last group, the GA algorithm achieve
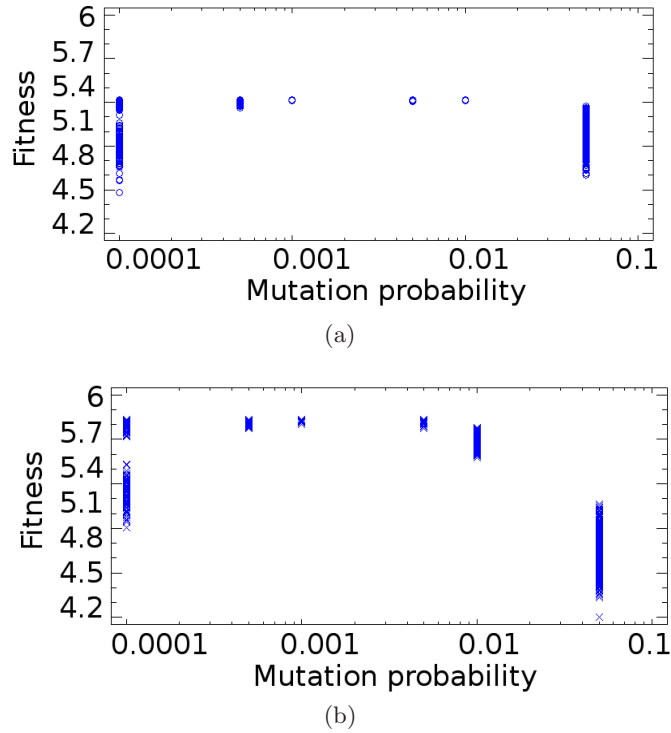
(a)

(b)

Fig. 5. Dispersion fitness vs probability using neighborhoods a) 4b, b) 5

the optimum result, but in a great number of executions, the result is not the optimum and there is a greater dispersion.

In order to determine if there are differences when using population sizes of 25, 50, 100, 150, 200, 300 and 400, we present Table 2 with mean, median and standard deviation values. As it can be seen, both standard deviation and variation coefficient grow as the population size does.

5   Conclusions

In this paper we have analyzed the input parameter of a GA that is able to distribute the sign prediction for a given neighborhood into r sets obtaining the lower aggregate entropy. From the analysis performed, we can conclude that the algorithm obtains better results as the number of context used increase. Even more, the neighborhood used for the wavelet sign prediction has a greater impact over the results, being the neighborhood that uses 5 neighbors the one that better results obtains. Regarding intrinsic parameters of the GA, when we use mutation probabilities in the range from 0.0005 to 0.005 we obtain less variability in the
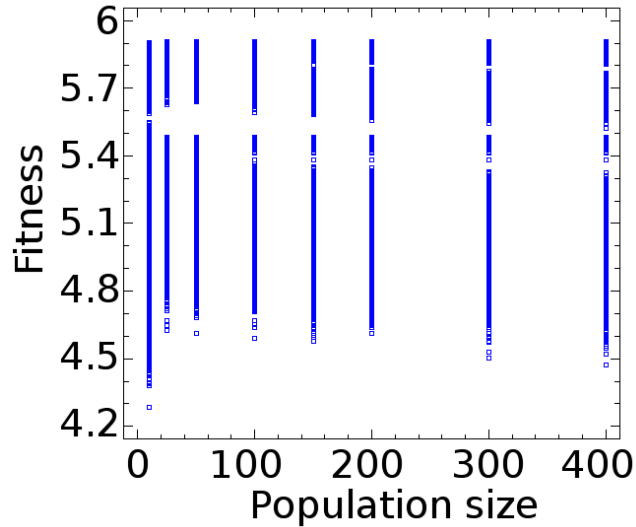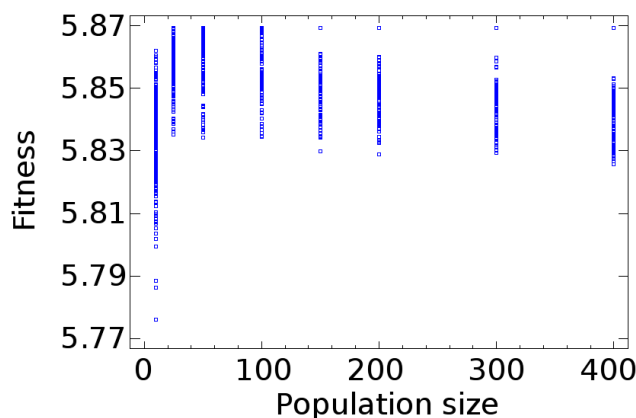
Fig. 6. Dispersion fitness vs population size

Population	25	50	100	150	200	300	400
Average	5.85746	5.86096	5.86153	5.86004	5.86034	5.85782	5.85712
Standard Deviation	0.00671	0.00835	0.01034	0.01182	0.01181	0.01397	0.01517
Variation Coefficient	0.1146%	0.1426%	0.1765%	0.2017%	0.2016%	0.2385%	0.2591%
Minimum	5.83509	5.8343	5.83435	5.82985	5.8289	5.82945	5.82575
Maximum	5.86922	5.86922	5.86922	5.86922	5.86922	5.86922	5.86922
Range	0.03413	0.03492	0.03487	0.03937	0.04032	0.03977	0.04347

Table 2. Execution results statistics for mutation probability from 0.0004 to 0.006, using neighborhood 5 and 6 contexts.

execution results. Finally, we also have analyzed the population size, being those in the range from 25 to 100 the ones that better results and behavior obtain.

References

1. Holland, J. In: Adaption in Natural and Artificial Systems. University of Michigan Press (1975)
2. Goldberg, D.E. In: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley (1989)
3. ISO/IEC 15444-1: JPEG2000 image coding system (2000)
4. Shapiro, J.: A fast technique for identifying zerotrees in the EZW algorithm. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing **3** (1996) 1455–1458
5. Wu, X.: High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression. In: Proc. of 31st Asilomar Conf. on Signals, Systems, and Computers. (1997) 1378–1382

Fig. 7. Dispersion fitness vs population size for ($mutation probability > 0.0004$) and ($mutation probability < 0.006$) and ($neighborhood = 5$) and ($context = 6$))

6. Taubman, D.: High performance scalable image compression with EBCOT. IEEE Transactions on Image Processing **9**(7) (July 2000) 1158–1170
7. Deever, A., Hemami, S.S.: What's your sign?: Efficient sign coding for embedded wavelet image coding. In: Proc. IEEE Data Compression Conf., Snowbird, UT. (2000) 273–282
8. Lopez, O., Martinez, M., Piol, P., Malumbres, M., Oliver, J.: E-ltw: An enhanced ltw encoder with sign coding and precise rate control. In: Image Processing (ICIP), 2009 16th IEEE International Conference on. (Nov 2009) 2821–2824
9. Schwartz, E.L., Z, A., Boliek, M.: CREW: Compression with reversible embedded wavelets. In: In Proc SPIE. (1995) 212–221
10. Deever, A., Hemami, S.S.: Efficient sign coding and estimation of zero-quantized coefficients in embedded wavelet image codecs. IEEE Transactions on Image Processing **12**(4) (April 2003) 420–431
11. Mallat, S., Zhong, S.: Characterization of signals from multiscale edges. IEEE Transactions on Pattern Analysis and Machine Intelligence **14**(7) (July 1992) 710–732
12. Tian, C., Hemami, S.S.: An embedded image coding system baed on tarp filter with classification. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Montreal, Canada. (May 2004)
13. Morton, G.M.: A computer oriented geodetic data base and a new technique in file sequencing. Technical report, IBM Ltd. (1966)
14. Riordan, J. In: Introduction to Combinatorial Analysis. Princeton University Press (1958)