Predictive depth coding of wavelet transformed images

J. Lehtinen[a]

[a]Turku Centre for Computer Science, Lemminkäisenkatu 14 A , 20520 Turku, Finland

ABSTRACT

In this paper, a new prediction based method, *predictive depth coding* (PDC), for lossy wavelet image compression is presented. It compresses wavelet pyramid composition by predicting the number of significant bits in each uniformly scalar quantized wavelet coefficient and then coding the prediction error with arithmetic coding. Adaptively found linear prediction context covers spatial neighbours of the coefficient to be predicted and corresponding coefficients on lower scale and in different orientation pyramids. In addition to the number of significant bits, the sign and the bits of non-zero coefficients are coded. The compression method is tested with standard set of images and the results are compared with SFQ, SPIHT, EZW and context based algorithms.

Keywords: wavelet, image, compression, lossy, predictive, depth, coding

1. INTRODUCTION

While the DCT-based JPEG[1] is currently the most widely used lossy image compression method, much of the lossy still image compression research has moved towards wavelet-transform[2] -based compression algorithms. The benefits of wavelet-based techniques are well-known and many different compression algorithms,[3–5] which have good compression performance, has been documented in literature.

General system for lossy wavelet image compression usually consists of three steps: 1) signal decomposition, 2) quantization and 3) lossless coding. Signal decomposition separates different frequency-components from the original spatial domain image by filtering the image with subband filter-bank. Quantization is the phase, where some of the image information is purposely lost to achieve better compression performance. In the third phase, the quantized wavelet coefficients are coded using as few bits as possible. Usually this is done in two phases: modelling the information to symbols and then entropy encoding the symbols. The entropy coding can be optimally implemented with arithmetic coding,[6] and thus the compression performance largely depends on the modelling algorithm.

Signal composition can be done in different ways using 1-dimensional wavelet transform for the columns and the rows of the image in some order or by directly using 2-dimensional transform. One of the most popular compositions is to use 1-dimensional transforms for the rows and columns of the image and thus divide the image to four different orientation bands. Then the same signal-composition is done recursively to the up-left quadrant of the image to construct a pyramid (dyadic) composition (Fig.1). It is also possible to apply the signal-composition to other orientation subbands and this way to construct more complicated wavelet packet signal-compositions.

The 1-dimensional discrete wavelet transform (DWT) is basically very simple: discrete signal is filtered using low- and high-pass filters, and the results are scaled down and concatenated. The filters are constructed from a mother wavelet-function by scaling and translating the function. Few of the most used functions in lossy image compression are biorthogonal 9-7 wavelet, Daubechies wavelet family and coiflets. Usually the transforms itself are not lossy, but produce floating point result, which is often rounded to integers for coding.

The quantization of the wavelet coefficient matrix can be done in many ways. The most obvious way of quantizing the matrix is to use uniform scalar quantizer for the whole matrix, which usually gives surprisingly good results. It is also possible to use different quantizers for the different subbands or even regions of the image, which leads to variable quality image compression.[7] A more sophisticated way of quantifying the matrix would be to use vector quantization, that is computationally more intensive, but in some cases gives better results. Many techniques also implement progressive compression by integrating the modelling and the scalar quantization.

Many different approaches for modelling has been documented in literature. The most obvious one is to entropy code the quantized coefficients directly using some static, adaptively found or explicitly transmitted probability distribution. Unfortunately this leads to relatively poor compression performance, as it does use any dependencies

Further author information: E-mail: jole@jole.fi, WWW: http://jole.fi/

between the coefficients. Shapiros landmark paper[4] used quad-trees to efficiently code zero regions of the scalar-quantized coefficient matrix and called them zerotrees. SPIHT[3] developed the zerotree approach further by implicitly sending zerotrees embedded into code-stream and achieved very good compression performance. Also spatial contexts has successfully been used to select probability distribution used for coding the coefficients. One of best methods currently is based on space-frequency quantization, that tries to find optimal balance between scalar quantization and spatial quantization achieved with zerotree-like structures.

In lossless image compression, signal prediction and prediction-error coding are efficient and widely used methods. In this paper the usage of these methods in lossless wavelet image coding is evaluated and *a new method for modelling quantized pyramid composition is introduced.* The modelling algorithm is evaluated by implementing a wavelet image compression scheme with the following components: 1) biorthogonal 9-7 transform is used to construct a pyramid composition, 2) uniform scalar quantization is used for quantization and 3) after the modelling arithmetic coding with adaptive probability estimation is used.

It is possible to try directly predict the values of the wavelet coefficients, but the prediction results would probably be very inaccurate. Because the goal is to code the quantized coefficients accurately and thus the prediction error would be quite big, the efficient error coding would be impossible. Instead of trying to predict the exact coefficients, the PDC-algorithm tries to estimate the number of bits needed to represent the coefficients and then codes the coefficient itself separately.

In this paper the number of significant bits in an coefficient is called *depth*. Each coefficient is coded as the depth-prediction error, the sign and the actual bits of the coefficient. The prediction is made with simple linear predictor, which covers six spatial neighbours, a coefficient on the lower scale band and two coefficients on the different orientation bands. The weight of the linear predictor are adaptively learned in the compression. The signs are coded using simple context model. Also several coding contexts for the error-coding of zero depth coefficient is used.

In section 2 gives a brief introduction to all components of the PDC compression scheme. Biorthogonal 9-7 wavelet transform is introduced, the structure of the pyramid composition and the dependencies between the coefficients are explained and finally scalar quantization and entropy coding principles are briefly introduced. Section 3 explains the modelling algorithm in detail. First the depth is explained and depth prediction context is defined. Then adaptive prediction context coefficient estimation is explained. Modelling of the estimation errors is discussed and finally spatial sign coding context is defined and the coding of absolute values of actual coefficients are explained. The compression performance of PDC is tested and compared to some other image compressing techniques in section 4. Furthermore the memory requirements and the speed of the PDC are estimated. Conclusions and results are shown in section 5.

2. BACKGROUND

2.1. The structure of the compression method

The PDC-compression algorithm consists of six different steps:

1. The spatial domain image is transformed to pyramid composition structure (section 2.2).
2. Resulting coefficient matrix is quantized using scalar quantization (section 2.3)
3. Quantized coefficients are converted to depth representation and average values of the depth are calculated for the different subbands (section 3.2)
4. The significant bits of the absolute coefficient values (section 3.5) and the signs (section 3.4) are entropy-coded (section 2.4) using several coding contexts.
5. The depth of each coefficient is predicted using the same information that will be available for the decompression in the same stage when predicting the same coefficient (section 3.2)
6. The prediction errors are calculated and entropy-coded (section 3.3)

The decompression is done similarly, but in reverse order: 1) the depth-prediction error are decoded, 2) depths are predicted and the correct depth values are calculated using the decoded prediction errors, 3) the signs and the absolute values of the coefficients are decoded 4) different components of the coefficients are combined and 5) the reverse transform is used to produce the spatial domain image.

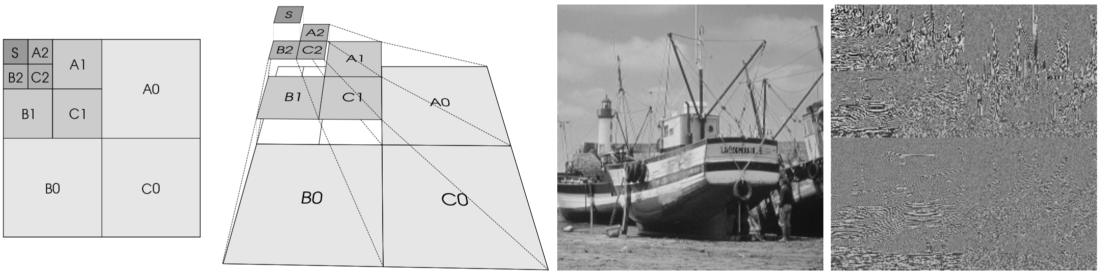

Figure 1. Images from left to right: 1) pyramid composition structure, 2) three orientation pyramids A,B and C, 3) spatial domain test image and 4) the same image in wavelet domain: the absolute values of the coefficients are represented in logarithmic scale.

2.2. Pyramid composition structure

In the PDC, the image is transformed to a subband pyramid-composition for quantization and compression using biorthogonal 9-7[2] wavelet functions. Wavelet transform is done hierarchically for the image by filtering the rows with low- and high-pass filters and then combining the down-scaled results side by side. Then the same operation is done to the columns of the image. The result of these two operations is a matrix with the same dimensions as the original image, but it is divided into four parts: S) Top left part contains a scaled down low-frequency components of the image. This part is also called scaling coefficients and basically it looks like scaled down version of the image. A) Top right part of the image has the vertical high-frequency details of the image and B) bottom left corner the horizontal details. C) Bottom right corner of the image contain diagonal high-frequency components.

Pyramid composition (Fig. 1) is created by doing the transformation described above recursively to the scaling coefficients on the top left corner of the coefficient matrix. This can be repeated while the dimensions of the scaling coefficients are even. The three pyramids can be seen in the pyramid composition: A,B,C. Pyramid levels correspond to different scale details of the spatial domain image and the different pyramids to details having different orientations. Usually the details in spatial image have features of multiple orientations and scales and thus they affects to the coefficients on all pyramids on multiple levels. These dependencies imply that coefficients are not independent and their values or magnitude can be somewhat predicted from the other coefficients.

2.3. Scalar quantization

Scalar quantization of the coefficients is simply a function from the set of all possible coefficients to a smaller set of representative coefficients, which approximate the original coefficients. Uniform scalar quantization is simplification of general case, where each quantized coefficient represents uniform set of original coefficients. Thus the uniform scalar quantization Q of the coefficients c can be simply implemented as

$$Q(c, q) = \lfloor q(c + 0.5) \rfloor / q,$$

where the quantization parameter q determines how much the data is quantized.

2.4. Entropy coding

Entropy H gives a lower bound[6] on the average code length of a symbol having probability p for any instantaneously decodable system :

$$H(p) = p \log_2(1/p).$$

This implies that if our modelling system can model the image to a stream of N symbols having independent probabilities, $p_0, p_1, \ldots, p_{N-1}$, an optimal entropy coder can compress it to $\sum_{i=0}^{N-1} p_i \log_2(1/p_i)$ bits.

The assumption here, when calculating the length of optimal representation of symbol-stream was that the probabilities of the symbols are known. It is of course possible to first do all the modelling and calculate the probabilities for the symbols, but usually adaptive probability estimation gives very good results. Moreover, it is

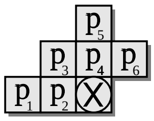

Figure 2. Spatial context for the predicted coefficient (marked with X) consists of depths of the six surrounding coefficients.

not necessary to code the probability distribution itself, if the probability distribution estimate is calculated from already sent symbols.

Arithmetic coder[6] represents a stream of symbols as one range between $[0, 1)$. This is done by recursively dividing the range in the proportion of symbol probabilities and selecting a subrange corresponding to the symbol to be coded. The output of the algorithm is the binary representation of some number in the final range. The compression performance is very close to the optimum defined by entropy.

3. PREDICTIVE DEPTH CODING

3.1. PDC algorithm

The PDC-algorithm models the quantized wavelet pyramid composition coefficients to allow them to be efficiently entropy-coded. Also the whole compression scheme consisting of biorthogonal 9-7 wavelet transform, scalar quantization, the actual modelling algorithm and a dynamic arithmetic coder is in this paper referred as the PDC as described in section 2.1. In this section the modelling phase of the scheme is described in detail.

The coefficients are compressed one by one starting from the highest pyramid level and continuing downwards level by level in all orientation pyramids. More precisely the scaling coefficients (on the level S in Fig. 1) are compressed first in row major order*, then similarly the bands on the next pyramid level in order C,B,A (that is C2,B2,A2 in the example) and finally similarly all the other levels below it.

For each coefficient, it's depth is predicted. Then the actual depth and the prediction error is calculated. The prediction error is coded together with the sign and the actual absolute value of the coefficient. In addition to an individual coefficient, the compression algorithm needs to know some coefficients on the coefficients spatial neighbourhood, on different orientation pyramids on the same spatial location and on the parent level. Also the information on the mean depth of coefficients on current level and its parent level is needed.

3.2. Depth prediction

The depth D of the coefficient c is here defined simply as the number of bits needed in the representation of the absolute value of a coefficient.

$$D(c) = \begin{cases} \lfloor \log_2(|c|) + 1 \rfloor, & \text{if } c > 0 \\ 0, & \text{if } c = 0 \end{cases}$$

Thus the coefficient can be represented as the depth, followed by the absolute value and the sign. In this representation the insignificant bits are not needed.

In PDC, the prediction is made with linear predictor $P(p_1, p_2, \ldots, p_N, w_1, w_2, \ldots, w_N) = \sum_{i=1}^{N} w_i p_i$, where p_i form the prediction context, which is weighted by corresponding w_i $(sum_{i=1}^{N} w_i = 1)$. The prediction context is divided to three parts: 1) spatial context describing the depths of six surrounding coefficients, 2) orientational context consists of the depths of two orientational coefficients on the same level and in same spatial location, 3) a depth estimate calculated from the depth of the parent coefficient and the mean depths of the levels.

The prediction context in the PDC contains the depths of 9 coefficients: 6 spatial neighbours, 2 orientational neighbours and 1 coefficient from the parent-level. The six spatial neighbours (Fig. 2) used in the prediction are selected because they give the best results with the PDC-algorithm. The orientational context is selected to be the depths of the coefficients in the same spatial location on the two other orientation pyramids on the same level. Experiments with larger orientational context did not give any better results. Only one coefficient is used on the

*By traversing the coefficient of the band from left to right and from top to bottom upper level - the one in the same spatial location and in the same orientation pyramid as the coefficient to be predicted. Because the mean-depths of the levels are very different, the depth of the parent coefficient is divided by the mean-depth of its level and multiplied by the mean-depth of the current level.

In the compression phase, only the depths known to the decompressor can be used in predicting and for the rest of the prediction context zero weights must be used. For example for the first coefficient of a band the whole spatial context is undefined and only the coefficients on the orientational context and a coefficient on the previous level, can be used (if they are known).

The weights W_i for the linear prediction-context are calculated adaptively in the prediction algorithm:

- $\forall i \in [1, 9] : W_i \leftarrow 1$
- while not finished
  - $\forall i \in [1, 9] : w_i \leftarrow W_i$
  - $\forall i \in \{\text{unknown,undefined}\} : w_i \leftarrow 0$
  - $s \leftarrow \sum_{i=1}^{9} w_i$
  - $P \leftarrow \begin{cases} (\sum_{i=1}^{9} w_i D_i)/s & \text{if } s > 0 \\ 0 & \text{if } s = 0 \end{cases}$
  - $\forall i \in [1, n] : W_i \leftarrow \begin{cases} W_i * \alpha & \text{if } |p_i - D| \leq |P - D| \\ W_i/\alpha & \text{if } |p_i - D| > |P - D| \end{cases}$
  - $\forall i \in [1, n] : W_i \leftarrow \begin{cases} \beta \text{ if } W_i > \beta \\ \gamma \text{ if } W_i < \gamma \\ W_i \quad \text{otherwise} \end{cases}$

One good selection for the parameters is $\alpha = 1.1, \beta = 100$ and $\gamma = 0.3$. In the algorithm D denotes the depth of the current coefficient and D_i the depths of the its neighbours, where the index corresponds to the location in prediction context. The resulting prediction is denoted with P. For the first prediction the s would be 0 and thus the first coefficient should not be predicted at all.

3.3. Coding the prediction errors

For each coefficient, the prediction error of the depth should be coded. Because the linear prediction produces a floating-point value it must be rounded before error calculation. The prediction error can then be directly coded with arithmetic coder.

On the lower pyramid levels most of the quantized coefficient values are zero and thus also the zero prediction is very common. Better compression can be achieved by adjusting the error coding probability distributions, when the prediction is very near to zero. In the PDC this is implemented by using six independent probability distributions. The distribution is selected by classifying the predictions to the following classes: $[0, 0.01)$, $[0.01, 0.05)$, $[0.05, 0.13)$, $[0.13, 0.25)$, $[0.25, 0.5)$ and $[0.5, \infty)$. This selection of coding contexts seem to work quite well for natural images, but the number of classes and the edges can be varied quite much with only small changes in compression performance.

3.4. Coding the sign

Although the signs of the coefficients seem to be very random and their distribution is almost even, some compression can be achieved using two neighbours (north and west) as coding context. Thus nine different probability distributions is used, as each sign can be either $+$, $-$ or 0 (if the quantized coefficient is zero, the sign is unknown and insignificant).

3.5. Coding the quantized coefficients

In addition to the sign and the depth of the coefficient, the actual quantized absolute value must be coded. The value could be coded simply with arithmetic coding using the depth as a context, but in the PDC even simpler approach is taken: the bits of the absolute value are coded one my one. A binary arithmetic coder can be used, but the bits can also be saved without entropy coding.

As the depth of the coefficient is known, only the bits after the first one-bit of the absolute value must be coded. The bits could be saved without any entropy coding as their distribution is very even. As the distribution of absolute values is very skewed, for example value of 2 is more common than 2, and thus different coding context is used for the first bit after the first one-bit

4. TESTING

4.1. Compression performance

As the purpose of the PDC is to apply the prediction coding principles in wavelet-coding, the absolute compression performance results are compared to some advanced wavelet compression techniques. Furthermore, as the compression scheme consists of many different sub-components, their influence on compression performance is evaluated separately. The peak signal to noise ratio (PSNR)[5] is used as image quality measure.

The compression performance (Table 4.1 of the PDC is measured by compressing three 512×512, 8-bit test images: lena, barbara and goldhill, using 0.1-1.8 bits per pixel (bpp). The performance is compared to the compression results of the standard JPEG,[1] a context based wavelet compression technique (C/B),[8] space frequency quantization (SFQ)[9] and a wavelet image compression technique based on set partitioning in hierarchical trees (SPIHT).[3]

The JPEG codes the image in fixed 8×8 blocks by first transforming the image block using discrete cosine transform (DCT) and then applying scalar quantization to the coefficients. The quantized coefficients are coded in zig-zag order using Huffman-coding. Context based technique compresses the scalar quantized wavelet pyramid composition coefficients using independent probability distributions that depend on the spatial context of the coded coefficients. The technique is relatively simple, yet very efficient. SFQ is based on the usage of zerotrees[4] in wavelet pyramid composition coding. The idea of the SFQ is to find optimal balance between spatial quantization done with zerotrees and scalar quantization. The resulting compression performance is very good. The SPIHT is also based on the idea of zerotrees, but it develops the zerotree structure coding further by coding the tree information implicitly as the decision information of the set partitioning algorithm, which constructs the zerotree. Although the method is somewhat complicated, the results are very good and moreover the modelling algorithm is so powerful, that the entropy-coding can be omitted in some applications. The algorithm can also be easily optimized with little modifications.[7]

The compression performance graphs (Fig. 3) show that generally the PDC algorithm is not as good as the current state of the art algorithms, but the differences in the compression performance are not very significant. In comparison with the JPEG algorithm, the inflexibility of the fixed sized block structure and the advantages of the wavelet transform can be clearly seen as better compression performance and scalability to smaller bit-rates.

BPP	PSNR(dB)		
	lena	barbara	goldhill
0.50	49.8	49.8	49.8
0.25	45.8	45.8	45.8
0.15	42.3	42.5	42.1
0.10	39.9	40.0	39.2
0.05	36.6	35.8	34.9
0.01	29.5	26.4	27.6

Table 1. The PDC compression results

4.2. Coding benefits of different components

The modeller used in the PDC uses several different techniques gain better compression performance. Many of these techniques can be left out from the PDC or integrated to other wavelet image compression systems. The individual coding gains can be measured by leaving the corresponding coding technique out from the PDC and comparing the compression performance to the original results (Fig. 4).

It seems that the sign prediction and separate context for the second bit coding are not absolutely necessary. If one would like to reduce the amount of arithmetic coding done, the performance loss from directly saving the sign - and the coefficient bits is not very big. Also it seem to be important to use several context for coding the coefficients that are predicted to be near zero.

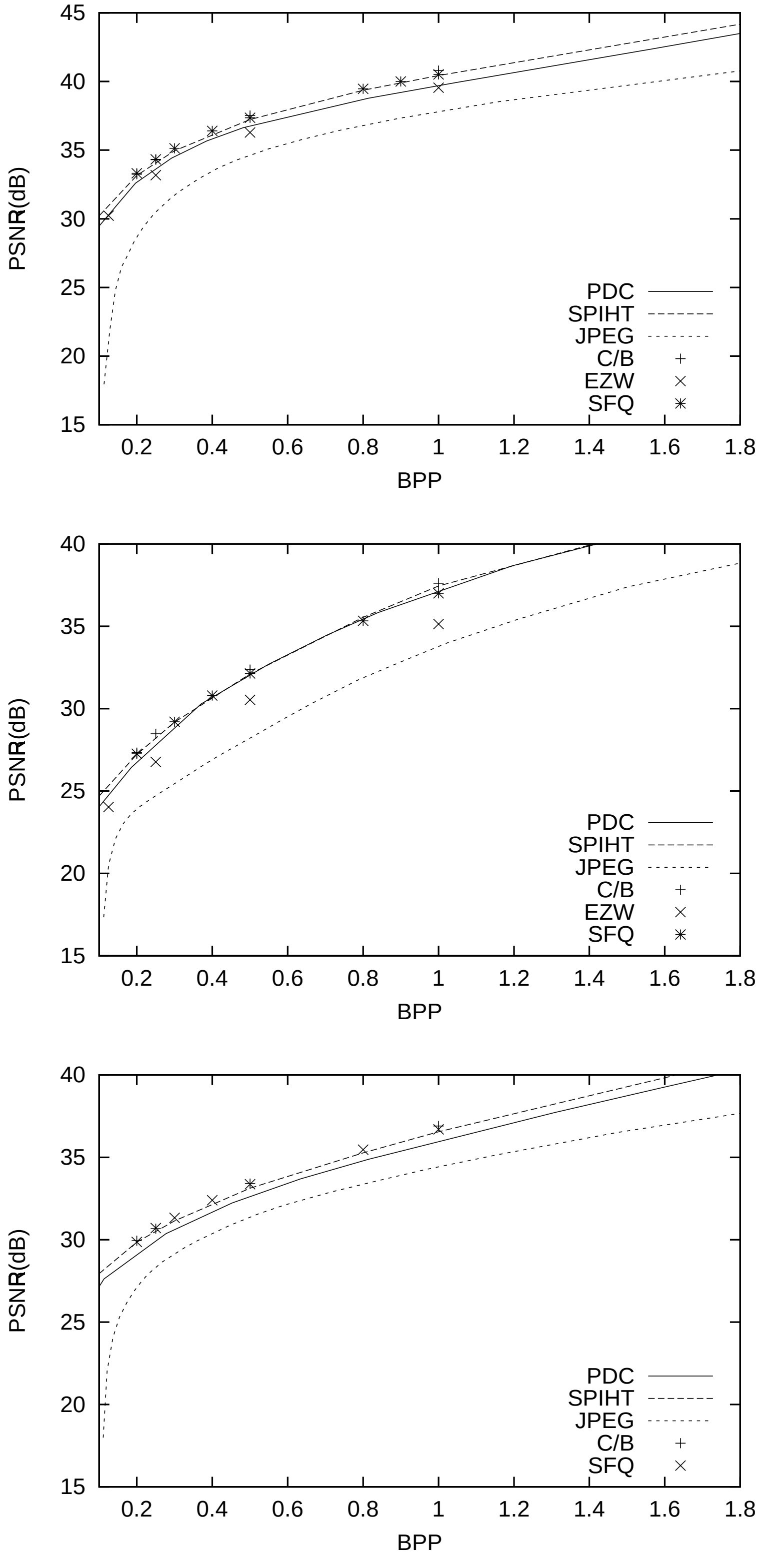

Figure 3. Compression performance of the PDC with three different 512×512 8-bit test images is measured. The test images used are from top to bottom: lena, barbara and goldhill.

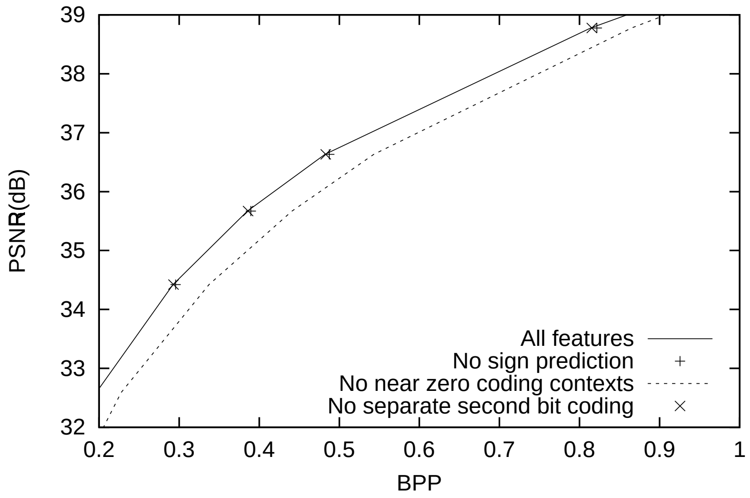

Figure 4. The compression performance of the PDC on the lena test-image, when leaving out some of the features.

4.3. Efficiency

One of the advantages of the PDC is that the algorithm does not require any extra memory, like many other wavelet transform coding algorithms. Still the wavelet transformed image itself must be in memory to calculate the prediction. New methods for wavelet transforms having reduced memory requirements have been documented in literature,[8] and the PDC algorithm could probably be easily changed to work with them, as only the prediction context must be changed.

The speed of the PDC depends of three different steps: 1) the wavelet transform, 2) modelling and 3) arithmetic coding. The two most time consuming calculations in the modelling step are coefficient prediction and prediction context coefficient adjustment. In the prediction one must calculate each estimate as a linear combination of the prediction context, which requires 9 floating point multiplications, additions and several memory address calculations and reads. The learning step is about three to five times more complex than the prediction. The learning phase could be approximated by making the update of the scaling coefficients sparse.

5. CONCLUSIONS

A new method for wavelet transform coding has been proposed. The predictive depth coding technique demonstrates that the conventional prediction coding principles from lossless image-coding can be applied to lossy wavelet transform coding. The PDC also demonstrates that sign compression with simple two-sign context is possible, yet relatively inefficient. The compression technique is very simple and does not need any extra memory.

Even though the prediction coding is possible, on the basis of these compression results it is questionable, if it should be used in lossy wavelet image compression. One possibility is to integrate some prediction coding principles in some more efficient transform-coding technique to achieve better compression performance with the resulting hybrid algorithm.

REFERENCES

1. W. Pennebaker and J. Mitchell, *Jpeg : Still Image Data Compression Standard*, Van Nostrand Reinhold, 1992.
2. M. Vettereli and J. Kovačević, *Wavelets and Subband Coding*, Prentice Hall, Englewood Cliffs, NJ, 1995.
3. 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* **6**, pp. 243–250, June 1996.
4. J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing* **31**, December 1993.

5. F. Bar, *Dummy reference to be corrected later*, Not published, 0.
6. S. K., *Introduction to Data Compression*, Morgan Kaufmann, 1996.
7. A. Järvi, J. Lehtinen, and O. Nevalainen, "Variable quality image compression system based on SPIHT," *to appear in Signal Processing: Image Communications* , 1998.
8. C. Chrysafis and A. Ortega, "Efficient context-based entropy coding for lossy wavelet image compression," in *DCC, Data Compression Conference*, (Snowbird, UT), March 25 - 27 1997.
9. Z. Xiong, K. Ramchandran, and M. T. Orchard, "Space-frequency quantization for wavelet image coding," *IEEE Trans. Image Processing* , 1997.