Image Compression using Wavelet Transform and Multiresolution Decomposition

to appear in IEEE Trans. on Image Processing, Vol. 5, No. 1, January 1996

A. Averbuch[†], D. Lazar[†] M. Israeli[‡]

[†] School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel

[‡] Faculty of Computer Science, Technion, Haifa 32000, Israel

Abstract

A scheme for image compression of black and white images based on the wavelet transform is presented. The multiresolution and multifrequency nature of the discrete wavelet transform is proved to be a powerful tool for representing the images which are decomposed along the vertical and horizontal directions using the pyramidal multiresolution scheme. The wavelet transform decomposes the image into a set of subimages (which are called shapes) with different resolutions corresponding to different frequency bands. Hence, different allocations are tested assuming that details at high resolution and diagonal directions are less visible to the human eye. The resulted coefficients are vector quantized using the LGB algorithm. By using an Error Correction method, which approximates the error of the quantization of the reconstructed coefficients, we optimize the tradeoff between low distortion and high compression rate while keeping a low computational cost.

In the following work, several types of compression techniques were tested.

In the first experiment, several images were trained together and common table codes were created. Using these table, the black and white images that were part of the training sequence achieved compression ratio of 60-65 and PSNR of 30-33.

In order to investigate the compression on images that are not part of the training set, many images of faces (which are not calibrated) were trained together and yielded global tables code. Images of faces outside the training set were compressed and reconstructed using the resulted tables. The compression ratio was 40 and the PSNR were 30-36. Images taken from the training set had similar values of compression and quality.

Finally, another approach of compression, based on the wavelet decomposition and vector bit allocation, was also examined. At each shape, the variances of the vectors were computed. The vectors at each shape were grouped and trained together according to the magnitude of their variance. Vectors which had higher variances were quantized using longer tables. Hence, the vector coefficients were quantized at each shape using several tables- each vector by the appropriate one. The connection of each vector to its quantization table was saved in a map file. The major improvement was by making the process much more efficient and fast, since smaller tables were used, and fewer

comparisons for finding the closest vector in the table had to be made - especially if the coefficients of the vector were not important enough. The bottleneck of searching large tables was eliminated.

The compression ratio and the quality of the faces compressed and reconstructed outside the training set had similar results to the previous ones: compression ratio of 35-36 and PSNR of 35-37, although faces which were reconstructed from the training set were slightly better comparable to the previous methods.
Distinct wavelet filters were tested, and the best results were achieved by applying biorthogonal filters. The results presented in this paper are better, comparable to the best results published recently [3, 10], in terms of peak signal-to-noise ratio as well as the subjective quality.

1 Introduction

Image compression means reducing the volume of data needed in order to represent the image. Image compression techniques are divided into two main approaches: transforms (DCT, JPEG, FFT, Wavelet) and non transforms (PCM, DPCM). The PCM and DPCM methods enables us to compress the data without transforming it, based on the signal's model. Compression can be achieved by transforming the data, projecting it on a basis of functions, and then encoding the resulted coefficients. One approach of encoding is based on quantizing the coefficients using scalar or vector quantization. Subband coding method even splits the coeffients according to distinct areas of behavior, while each "square" is quantized seperately according to its own special parameters and predetermined compression ratio. From a comparison between different compression methods which are described in [10], it appears that while achieving an image representation of 0.5 bpp, the accompanied PSNR is usually 27-29 dB. A compression based on the wavelet transform resulted with PSNR of 29.11-32.10 and compression ratios of 10-38 [3].
The compression method which is being used in this paper is based on applying a vector quantization on the wavelet coefficients resulted from the wavelet transform of the trained images, using the pyramidal multiresolution architecture (Mallat [11]). The multiresolution tree decompositions based on the wavelet transform is given by figure 1(b). Quantizing and encoding the wavelet coefficients is based on the assumption that details at high resolution are less visible to the human eye and therefore can be eliminated or be reconstructed with low order precision. The wavelet coefficients are coded according to their location (in the sense of pyramidal level and texture shape) and their importance for the quality of the final reconstructed image. Higher bits rate per pixel and lenghtes of vectors are allocated at shapes which are found to be more important, and Error Correction method, which approximates the error of the quantization of the reconstructed coefficients, is applied.

The compression of images using quantization of the wavelet coefficients is demonstrated in this work by several experiments.

A predetermined compression ratio to be used at each shape in the pyramidal structure is given, and different lengthes of vectors are tested, in order to find the best lengthes which enable both better quality and low CPU computational cost. After determining the appropriate allocations at each level and shape, several images are trained together and common table codes are created. In order to check our approach, we begin with processing images that were taken from the training set. Using these table, the black and white images that were part of the training sequence achieved compression ratio of 60-65 and PSNR of 30-33. The pyramidal decomposition had 4 levels, and the coefficients representing the texture of the image at the lower decomposition level were not quantized at all.

Then, further experiments based on the same quantization approach were tested in order to compress images that are taken outside the training set. Many images of faces (not calibrated) were trained together and yielded global (universal) tables code. The lengthes of the vectors were different, as well as the number of the levels in the decomposition (3). Also, since we had only 3 levels, the compression was reduced and the texture's coefficients were uniformly quantized (using 8 bits per coefficient). Images of faces outside the training set were compressed and reconstructed using the resulted tables. The compression ratio was 40 and the PSNR were 30-36, which proved the independent of the training set. Images taken from the training set had similar values of compression and quality.

In the previous methods, the length of the vectors at the tables for each shape was established experimentally, and all the vectors in the shape had to be quantized according to same large table. At the second part of this work, another approach of vector quantization based on vector bit allocation is explained. The vectors at each shape are seperated according to their variances. Using the bit allocation formula, each vector gets a number of bits by which it will be represented. Vectors with similar amount of bits are trained together and create an appropriate table code. It is obvious, that vectors with higher variances (which implies that they are more important to the image reconstruction) will be quantized using larger tables, while unimportant vectors will be approximated by shorter tables. The vector coefficients were quantized at each shape using several and smaller tables which made the compression process much more efficient and fast, and the decision of the vectors length was computed mathematically and not determined by heuristic experiments. The compression and quality of faces compressed and reconstructed outside the training set had similar results to the previous ones: compression ratio of 35-36 and PSNR of 35-37, although faces which were reconstructed from the training set were slightly better comparable to the previous methods. It is interesting to observe that inspite of the differences between the two approaches, the quality results were pretty similar.

In the paper, section II describes the wavelet transform which is used in this work. Section III brings the energy distribution and the statistical properties of the wavelet coefficients which yielded the bit allocations for each shape and the number of decomposition levels that should be used. Obtaining the desired bit allocation is accompanied to the considerations which determined the lengthes of the training tables and their vectors. In addition, the improvement of the vector quantization in producing sufficient quality with shorter vector and tables, by approximating the quantization error, is explained. The results of the comparison in the use of different wavelet filters are also summerized at this section.

Section IV brings experimental results for determining the lengthes of the training tables and their vectors and demonstrates the process on images taken within the training set. The changes needed for creating global tables are also considered, as well as experimental results for compressing and restoring faces taken outside the training set. Section V explains the compression method and its quantization based on the vector bit allocation, as well as results yielded applying those ideas on the wavelet coefficients.

2 Background

The wavelet transform [5, 6, 7] is a recent method of signal analysis and synthesis. It analyzes signals in terms of *wavelets*-function limited both in the time and frequency domain. In comparison, the classical Fourier analysis method analyzes signals in terms of sine and cosine wave components that are not limited in time.

The wavelet transform was introduced to compensate for the inconvenience of the window Fourier transform. It is computed by expanding the signal over a family of functions which are the dilates and translates of a single function. The wavelet transform enables to decompose an arbitrary signal into its localized contributions labelled by a scale parameter. It decomposes an arbitrary function into a two-parameter family of elementary wavelets that are obtained by shifts in the time variable but also by dilations that are both on the time and the frequency variables. This localization tecniques enrich substantially our ability to deal with all types of signal analysis. The locality behavior of the transform is in contrast to the global nature of the classic Fourier transform. The complete representation is calculated by decomposing the signal on a wavelet orthonormal basis and this gives an intermediate representation between a Fourier and a spatial representation.

Multiresolution representations, which is a part of the wavelet representation, are very effective for analyzing the infromation content of images. We use the properties of the operator which approximates a signal at a given resolution We know that the difference of information bewteen the approximation of a signal at the resolutions 2^{j+1} and 2^j can be extracted by decomposing this signal on a wavelet orthonormal basis of $L^2(\Re^n)$. A wavelet orthonormal basis is built by dialting and translating a unique function. This decomposition defines an orthogonal multiresolution representation. It is computed based on convolutions with quadrature mirror filters.

Wavelets are functions generated from one basic function by dilations and translations. A particularly interesting development is the recent discovery of orthonormal bases of wavelets. For particular functions $\psi \in L^2(\Re)$, the family $\psi_{jk}(x) = 2^{-\frac{j}{2}}\psi(2^{-j}x - k), \ \ j,k \in Z$, consistutes an orthonormal basis for $L^2(\Re)$.

The orthonormal wavelet bases can be associated with *multiresolution analysis*[11]. This analysis consists of ladder of spaces $\cdots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \cdots$ where their union spans $L^2(\Re)$ and their intersection is $\{0\}$. They satisfy the following conditions:

1. $f \in V_j \leftrightarrow f(2^j x) \in V_0$
2. There exists $\phi \in V_0$ such that the $\phi_{0n}(x) = \phi(x-n)$ consistute an orthonormal basis of V_0.

The spaces V_j can be considered as different approximations spaces: for a given f, the successive projections $Proj_{V_j} f$ describe approximations of f with resolution 2^j.

If we define $\phi_{jk}(x) = 2^{-\frac{j}{2}}\phi(2^{-j}x - k)$, then it follows that, for every j, $(\phi_{jk})_{k\in Z}$ consistute an orthonormal basis for V_j, such that $Proj_{V_j}(f) = \sum_{k\in Z} < f, \phi_{jk} > \phi_{jk}$.

Since $\phi \in V_0 \subset V_{-1}$ then $\phi(x) = \sum_{n \in Z} c_n \phi(2x - n)$ where from orthogonality we have that $\sum_n c_n c_{n+2k} = 2\delta_{k0}$. Therefore, the orthonormal wavelet basis associated to this multiresolution analysis is defined by $\psi(x) = \sum_{n \in Z} (-1)^n c_{-n+1} \phi(2x - n)$. In [5, 6] it is proved that $\psi_{jk}(x) = 2^{-\frac{j}{2}} \psi(2^{-j}x - k), \;\; j, k \in Z$ consistute an orthonormal basis for $L^2(\Re)$. Moreover, for every fixed j, $\{< f, \psi_{jk} >: k \in Z\}$ express the difference between the approximation of f with resolution 2^j and 2^{j-1} such that $Proj_{V_{j-1}}(f) = Proj_{V_j}(f) + \sum_{k \in Z} < f, \psi_{jk} > \psi_{jk}$.

The wavelet coefficients $< f, \psi_{jk} >$ are computed by:

$$< f, \psi_{jk} >= \sum_n g_{n-2k} < f, \phi_{j-1,n} > \tag{1a}$$

$$< f, \phi_{jk} >= \sum_n h_{n-2k} < f, \phi_{j-1,n} > \tag{1b}$$

Eqs. (1a) and (1b) describes convolution and decimation. The Eqs. (1a) and (1b) show how to compute a coarser approximation from a finer one, as well as, the difference in information between the two successive approximations. Recovering the finer approximation from the coarser one. We obtain

$$\begin{aligned} < f, \phi_{j-1,m} > &= \sum_k [< f, \phi_{jk} >< \phi_{jk}, \phi_{j-1,m} > + < f, \psi_{jk} >< \psi_{jk}, \phi_{j-1,m} >] \\ &= \sum_k [h_{m-2k} < f, \phi_{jk} > + g_{m-2k} < f, \psi_{jk} >] \end{aligned} \tag{2}$$

The whole decomposition and reconstruction scheme (Eqs. (1a)-(2)) is known as a *subband coding scheme with exact reconstruction.* If ϕ and ψ have compact support then h and g represents FIR filters (low and high pass respectively).

In 2D wavelet analysis one introduces, like in the 1D case, a scaling function $\Phi(x, y)$ such that:

$$\Phi(x, y) \;=\; \Phi(x)\Phi(y)$$

where $\Phi(x)$ is a 1D scaling function.

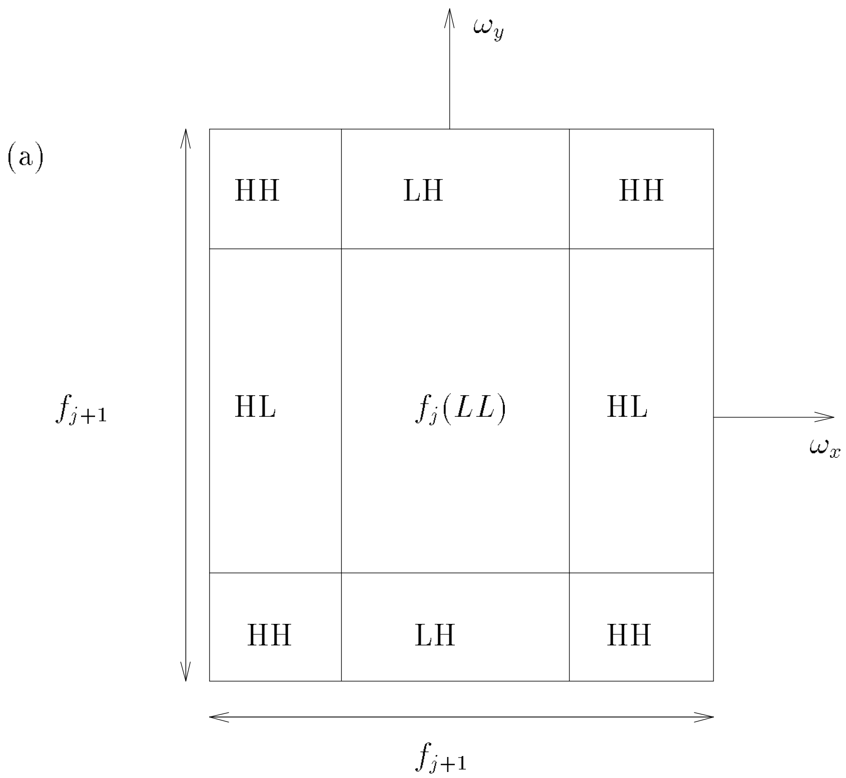
Let $\Psi(x)$ be the one-dimensional wavelet associated with the scaling function $\Phi(x)$. Then, the three 2D-wavelets are defined as:

$$\Psi^H(x, y) = \Phi(x)\Psi(y)$$

$$\Psi^V(x, y) = \Psi(x)\Phi(y)$$

$$\Psi^D(x, y) = \Psi(x)\Psi(y)$$

Suppose that $\Phi(x)$ and $\Psi(x)$ are a perfect low-pass and a perfect band-pass filter respectively. Figure 1(a) shows how the discrete signal, f_{j+1}, is decomposed in the frequency domain into $f_j(LL), LH, HL, HH$: f_j corresponds to the lowest frequencies at level j (LL), LH gives the vertical high frequencies (shape 1 - horizontal edges) at the resolution 2^{j+1}, HL the horizontal high frequencies (shape 2 - vertical edges) and HH the high frequencies in both directions (shape 3 - corners and diagonals).



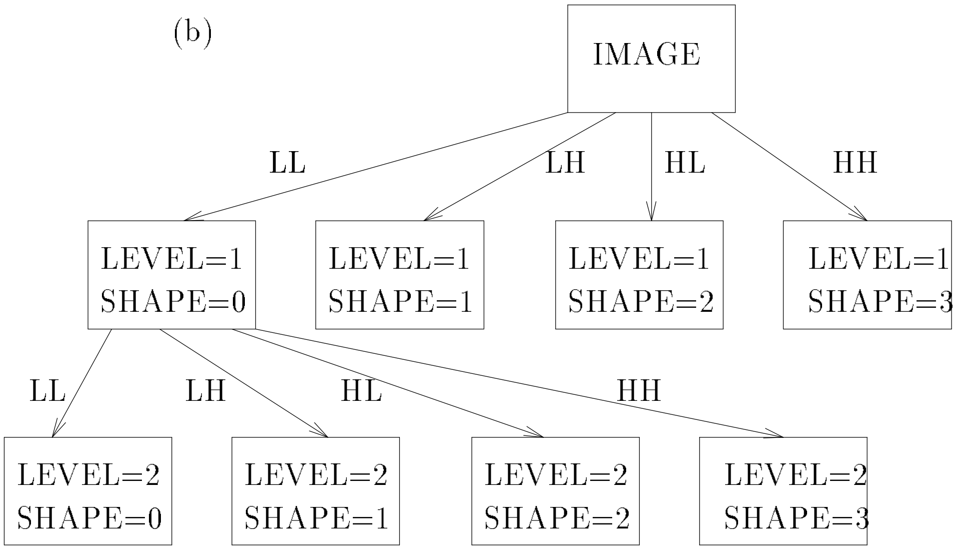


Figure 1: (a)Frequency support of the images f_{j+1}, f_j (LL) and the detail images LH, HL, HH. (b)Image decomposition into two levels.

3 Quantization

Applying wavelet transform to images does not reduce the amount of data to be compressed. On the contrary, most of the pixels are represented by coefficients that are much larger and necessitate longer bit allocations than the original signal's coefficients. A common way to reduce the amount of bits required for the compression phase is to quantize the coefficients and apply Huffman Code on the quantized coefficients or their representing indices.

3.1 Principle of Vector Quantization

The encoding and decoding scheme in Figure 2 demonstrates the use of Vector Quantization together with Wavelet Transform.
The encoding is performed by approximating the sequence to be encoded by a vector belonging to the codebook. Several techniques and methods are known for createing these codebooks. One of them is the LGB algorithm [12]. The codebook is created and optimized using classification based on a training set comprised of vectors belonging to different images; it converges iteratively to a locally optimal codebook.

3.1.1 The LGB algorithm

(0) Initialization: Given N=number of levels, a distortion threshold $\epsilon \geq 0.0$, an initial N-level reproduction alphabet $\hat{A}_0$, and a training sequence $\{x_j; j = 1, \ldots, n\}$. Set $m = 0$ and $D_{-1} = \infty$.
(1) Given $\hat{A}_m = \{y_i; i = 1, \ldots, N\}$, find its minimum distortion partition $\mathcal{P}(\hat{A}_m) = \{S_i; i = 1, \ldots, N\}$ of the training sequence : $x_j \in S_i$ if $d(x_j, y_i) \leq d(x_j, y_l)$ for all l.
Compute the resulting average distortion,

$$D_m = D(\{\hat{A}_m, \mathcal{P}(\hat{A}_m)\}) = \frac{\sum_{j=1}^{n} \min_{y \in \hat{A}_m} d(x_j, y)}{n}$$

(2) If $(D_{m-1} - D_m)/D_m \leq \epsilon$, halt with $\hat{A}_m$ and $\mathcal{P}(\hat{A}_m)$ describing final quantizer. Otherwise continue.
(3) Find the optimal reproduction alphabet $\hat{x}(\mathcal{P}(\hat{A}_m)) = \{\hat{x}(S_i); i = 1, \ldots, N\}$ for $\mathcal{P}(\hat{A}_m)$. Set $\hat{A}_{m+1} \triangleq \hat{x}(\mathcal{P}(\hat{A}_m))$. Replace m by $m + 1$ and go to (1).
Observe above that while designing the quantizer, only partitions of the training sequence (the input alphabet) are considered. Once the final codebook $\hat{A}_m$ is obtained, however, it is used on new data outside the training sequence with the optimum nearest-neighbour rule, that is, an optimum partition of K-dimensional Euclidean space.

3.1.2 Choice of $\hat{A}_0$

There are several ways to choose the initial reproduction alphabet $\hat{A}_0$ required by the algorithm. The splitting technique is useful when one wishes to design quantizers of successively higher rates until achieving an acceptable level of distortion. This method considers M-level quantizer with $M = 2^R, R = 0, 1, \ldots,$ and continues until achieving an initial guess for an

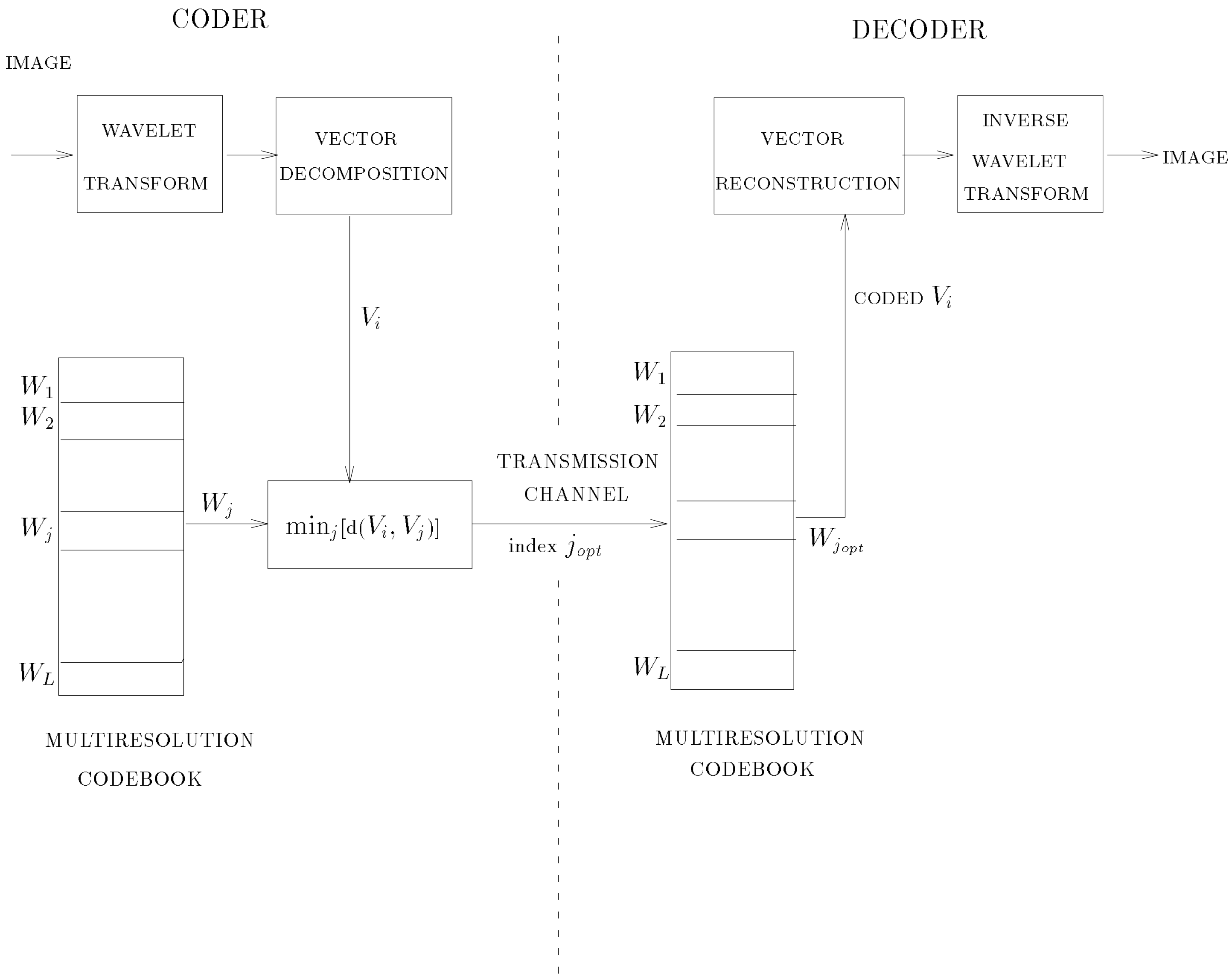


Figure 2: Encoding/Decoding Scheme

N-level quantizer as follows:
(0) Initialization: Set $M = 1$ and define $\hat{A}_0 = \hat{x}(A)$, the centroid of the entire alphabet (the centroid of the training sequence if a sample distribution is used).
(1) Given the reproduction alphabet $\hat{A}_0(M)$ containing M vectors $\{y_i; i = 1, \ldots, M\}$, "split" each vector y_i into two close vectors $y_i + \epsilon$ and $y_i - \epsilon$ where ϵ is a fixed perturbation vector. The collection $\tilde{A}$ of $\{y_i + \epsilon, y_i - \epsilon, i = 1, \ldots, M\}$ has $2M$ vectors. Replace M by $2M$.
(2) Is $M = N$? If so, set $\hat{A}_0 = \tilde{A}(M)$ and halt. $\hat{A}_0$ is then the initial reproduction alphabet for the N-level quantization algorithm. If not, run the algorithm for an M-level quantizer on $\hat{A}(M)$ and then return to step (1).

Comparing different methods yielded that choosing $\hat{A}_0$ by "splitting" performs better, and thus was chosen to be the initial alphabet method in this work.

3.2 Coding Evaluations

Let M_1 and M_2 be the dimensions (width and height) of the original image, R be the number of bits per pixel (bpp) of the compressed image, while the original image is represented by K bpp (in our case - 8 bpp). We also declare $f(i,j)$ to be the pixel's value or the wavelet coefficient value at coordinate (i,j).

Among the various numerical measures the most popular are :

$$Compression\ Ratio = C.R. = \frac{K}{R} \tag{1}$$

$$Energy = \sum_{i=0}^{M_1-1} \sum_{j=0}^{M_2-1} f^2(i,j) \tag{2}$$

$$Mean\ Square\ Error = MSE = \frac{\sum_{i=0}^{M_1-1} \sum_{j=0}^{M_2-1} [f(i,j) - \hat{f}(i,j)]^2}{M_1 \times M_2} \tag{3}$$

$$Error\ Root\ Mean\ Square = ERMS = \sqrt{MSE} \tag{4}$$

$$Peak\ Signal\ to\ Noise\ Ratio = PSNR = 20 \log_{10} \frac{255}{\sqrt{MSE}} \tag{5}$$

$$Signal\ to\ Noise\ Ratio = SNR = 10 \log_{10} \frac{\sum_{i=0}^{M_1-1} \sum_{j=0}^{M_2-1} [f(i,j)]^2}{\sum_{i=0}^{M_1-1} \sum_{j=0}^{M_2-1} [f(i,j) - \hat{f}(i,j)]^2} \tag{6}$$

$$Number\ of\ Big\ Errors = Number\ of\ Differences\ Greater\ than\ 21 \tag{7}$$

In this work, we evaluate both the images and their transform coefficients. The SNR and Energy measures are used for coefficients evaluations at each level and shape, while the other measures are for image evaluations.
By comparing the numerical and the subjective results we can say that in most of the cases they support each other, and fine visual quality is achieved when the PSNR is higher than 30 and ERMS is smaller than 6.5 (MSE smaller than 42.25).

bpp	Level	Shape	N	K	MAX SNR	LGB SNR
1.0	3	1	2	1	1.68	2.32
1.0	3	1	4	2	-	3.75
1.0	3	1	8	3	-	4.40
1.0	3	1	16	4	-	5.47
1.0	4	1	2	1	2.10	2.45
1.0	4	1	4	2	-	3.88
1.0	4	1	8	3	-	4.64
1.0	4	1	16	4	-	6.38
2.0	3	1	4	1	6.40	6.82
2.0	3	1	16	2	-	8.67
2.0	4	1	4	1	6.83	7.33
2.0	4	1	16	2	-	9.93
3.0	3	1	8	1	7.89	11.85
3.0	3	1	64	2	-	13.75
3.0	4	1	8	1	9.09	12.48
3.0	4	1	64	2	-	16.80

Table 1: Scalar and Vector quantizations results using Lenna's coefficients at shape 1 in levels 3,4. N is the number of the vectors in the table and K is the length of each vector. In the LGB case, the compression R bpp is calculated by $N = 2^{KR}$. MAX SNR brings the SNR of the coefficients created by Max quantizer.

3.3 Scalar Quantization versus Vector Quantization

The wavelet coefficients can be quantized, among other methods, by scalar (SQ) or vector (VQ) quantization.

Comparing SQ versus VQ should take into consideration two aspects: compression ratio achieved and quality evaluations. Using vectors of coefficients instead of scalars gives higher compression ratios, since for allocating R bits per pixel one only has to choose N (the number of vectors in the table) and K (the length of each vector) such that $N = 2^{KR}$.
Table 1 compares between SQ and VQ. In comparison to the SQ, we used the values which were computed by Max's quantizer([13]). The quantizer uses the values which are best for achieving 1.0-4.0 bits per pixel representation while quantizing samples that have pdf of Lapalacian, Gaussian or Gamma functions. According to our analysis of the wavelet pdf, the Gamma's values were used.

When using $K = 1$, VQ becomes SQ but, as shown in Table 1, scalar quantizations which were achieved by LGB gives better results then using Max's approximations of the Gamma function, since the pdf of the wavelet is a little different and the LGB algorithm adapts to the wavelet's pdf very easily. Also, it is observed that by using vector quantization a much higher SNR is achieved comparing with using scalar quantization of the same compression ratio. Therefore, the present application uses vector quantization.

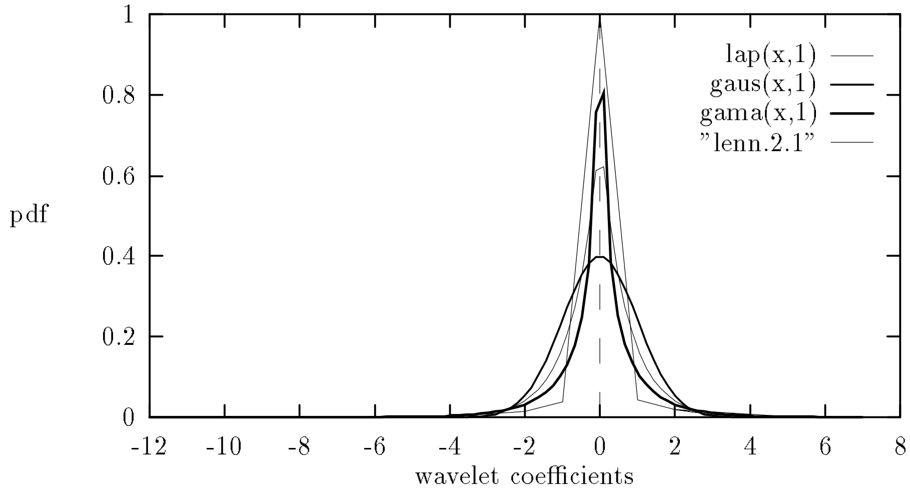


Figure 3: Wavelet coefficients and approximations. The wavelet coefficients were sampled from shape 1 in level 2 of Lenna's decomposition and were scaled by the local variance. The graph of lenn.2.1 demonstrates the wavelet pdf distribution. Similar results were given using other images. The graphes of lap(Laplacian), gaus(Gaussian) and gama demonstrates the approximations.

3.4 Statistical Distribution of the Wavelet Coefficients

The probability density function (pdf) of the wavelet coefficients at each shape (direction) is important in order to understand the behavior of the different resolutions. The results are demonstrated using the wavelet coefficients of Lenna's image but decomposing many other images yielded similar results.
Figure 3 shows the wavelet pdf at level 2 and direction shape=1 (horizontal edges) together with three approximations : Gaussian, Laplacian, and the Gamma function.

$$Gaussian(x, \sigma_x) = \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp[-x^2/2\sigma_x^2] \tag{8}$$

$$Laplacian(x, \sigma_x) = \frac{1}{\sqrt{2}\sigma_x} \exp[-\sqrt{2}|x|/\sigma_x] \tag{9}$$

$$Gamma(x, \sigma_x) = \frac{\sqrt[4]{3}}{\sqrt{8\pi\sigma_x|x|}} \exp[-\sqrt{3}|x|/2\sigma_x] \tag{10}$$

Examining the coefficients of the black and white images shows that the graph of the wavelet coefficients pdf remains similar for all the shapes at the same level (see Figure 4) or at different levels, excluding the lower resolution sub-image (shape 0).

Thus, we may conclude that the pdf of the wavelet coefficients is closely approximated by the Gamma function at all levels and shapes used in our work. Another fact that can be

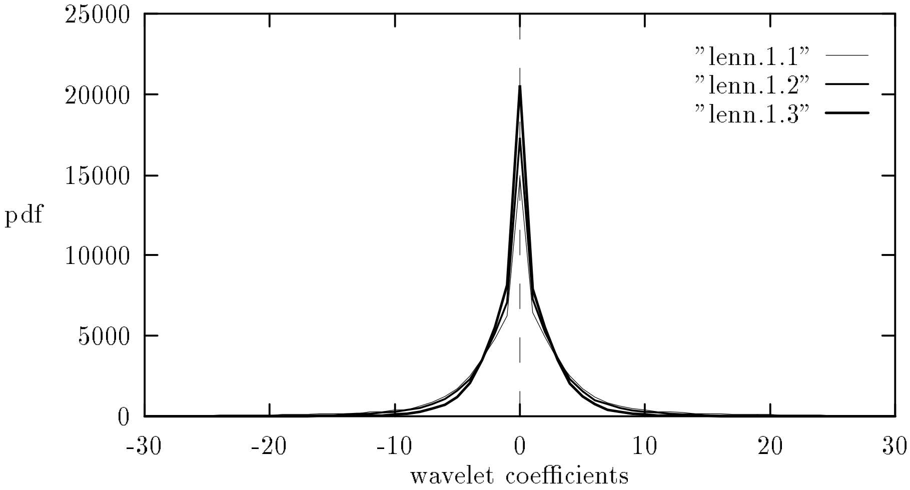


Figure 4: Wavelet coefficients distribution: Shapes 1,2,3 at level 1 from Lenna are demonstrated. Similar results were also given by other images.

LEVEL	SHAPE	MIN	MAX	AVR	VAR	(Min-Avr)/Var	(Max-Avr)/Var
1	1	-100.8	106.3	0.13	8.090	-12.4	13.1
1	2	-53.28	81.37	0.03	5.057	-10.5	16.0
1	3	-34.07	41.33	0.00	3.136	-10.3	12.4
2	1	-343.3	207.8	0.23	26.65	-12.8	7.78
2	2	-198.4	157.6	0.16	15.65	-12.6	10.0
2	3	-113.9	137.7	-0.14	12.17	-9.34	11.3
3	1	-479.6	587.8	-2.16	68.05	-7.01	8.66
3	2	-324.2	257.8	-0.34	38.74	-8.36	6.66
3	3	-256.5	315.1	0.13	35.85	-7.15	8.78
4	1	-931.6	1070	-4.46	196.1	-4.72	5.48
4	2	-593.8	682.2	-3.61	98.13	-6.01	6.98
4	3	-484.5	721.0	0.39	89.85	-5.39	8.02
4	0	-1743	1609	-155	780.5	-2.03	2.26

Table 2: Lenna's Wavelet Coefficients

observed in the graphs is that 90% of the coefficients are very small and fall in a very narrow dynamic range around the origin.

Table 2 demonstrates the behavior of the coefficients. It shows the statistics of each level and shape and describes the smallest and maximal coefficient (MIN and MAX) at each shape, the values of the average and variance (AVR and VAR) and the range of each shape after normalizing by the average and variance.
The importance of each level is determined by table 2 since:

1. The range of the coefficients (which is [MIN,MAX]) and their variances grow as the resolution becomes coarser which proves that the coefficients at the lower levels have higher importance for the image reconstruction.

2. After subtracting the average and scaling by the variance, the range of the coefficients becomes smaller and very similar for every shape at each level (see last 2 columns at table 2). This fact (which is demonstrated in figure 4) enables us to quantize different shapes with similar considerations.

3. At the coarser level, shape 0 (which contains the smoothed, low pass, coefficients) consists of much wider range and larger coeffiecients and variances than the other shapes. That subimage contains the most important coefficients, thus they will be quantized using a scalar quantization which might result in smaller compression but also in higher accuracy of reconstruction.

3.5 Energy Distribution

From the behavior of the wavelet coefficients and their distribution it can be seen that most of the coefficients fell in a narrow range around the origion, and the higher levels contained smaller coefficients and variances. Looking at tables 2,3, we see that at each level, shapes 1,2 contain more energy and larger coefficients values and variances than shape 3. Also, level 1 contains less energy and smaller coefficients than level 2, which contains less energy than level 3, etc. Tables 2,3 bring the results of analysing the wavelet coefficients of the faces Lenna and John. The same behavior is expected when analysing other images.

3.6 The Importance of Level 1

In the previous sections the importance of the different levels was discussed. Lower levels were proved to be more important, a fact that arise special attention to the upper level, level 1, which is both holds the least important coefficients and the largest one. Level 1 contains 75% of the coefficients. Quantizing that level efficiently is very critical to the total compression process. In addition, during the reconstruction process, the error caused by quantizing its coefficients do not interfere with the other levels reconstruction quality (see Table 4).
We can see that the diagonal coefficients (shape 3) are less important for the reconstruction since they are responsible for patterns that can hardly be seen by the eye, especially in the upper levels. By excluding them, we can still achieve PSNR of 43(!), while at the same time by removing the coefficients of shapes 1 or 2 the PSNR decreases to be 34-39.

LEVEL	SHAPE	LENNA	JOHN
1	1	0.563	0.173
1	2	0.220	0.126
1	3	0.094	0.034
Sum of Level 1		0.877	0.333
2	1	1.528	0.227
2	2	0.527	0.161
2	3	0.318	0.083
Sum of Level 2		2.373	0.471
3	1	2.492	0.230
3	2	0.807	0.157
3	3	0.691	0.075
Sum of Level 3		3.99	0.462
4	1	5.179	0.401
4	2	1.298	0.441
4	3	1.088	0.093
Sum of Level 4		7.565	0.935
4	0	85.19	97.79

Table 3: Lenna's and John's faces wavelet coefficients : energy distribution. For each level and shape the relative energy is computed by (eneregy of the coefficients at that shape)/(total energy of the coefficients).

Reconstruction Type	PSNR	ERMS
without shape 1 in level 1	35.52	4.26
without shape 2 in level 1	39.69	2.63
without shape 3 in level 1	43.09	1.78
without shapes 1,2 in level 1	34.13	5.00
without shapes 1,2,3	33.60	5.32

All the coefficients that were left
(at levels 2-4) were kept unchanged.

Table 4: The importance of Level 1 in Lenna's Reconstruction. Other images gave similar results which yielded the conclusion that the coefficients in Level 1 might not be necessary.

Table 4 also enables us to decide to neglect and ignore all the coefficients in level 1, since it is still possible to achieve high PSNR (higher than 33).

3.7 Error Correction (EC)

Applying the LGB on the wavelet coefficients did not always produce satisfactory results. Better results could be obtained by allocating longer vectors and tables which necessitated heavy computational load related to searching and accessing the tables. The goal is to achieve high SNR of the reconstructed coefficients and high compression ratios without allocating enormous tables that degrades the performance.
One way to improve the reconstructed results is by using Error Correction (EC) method which approximates the error of the quantization (Figure 5): The quantization error of the reconstructed image is computed and then the LGB quantization is applied on the error. We get an approximation of the error and add it to the previous reconstructed image and this reduces the differences from the original one. This process yields better SNR of the quantized coefficients and higher PSNR of the reconstructed image, as well as several fine qualities of compression:

1. If we decide in advance, to use allocation of 0.5 bpp in a certain shape, it can be done either by allocating N, K (number of vector in the table, and their length) for the shape's coefficients such that $N = 2^{0.5K}$ is held, or by allocating 0.25 bpp for the shape's samples and 0.25 bpp for the error samples. The last choice yields a much higher SNR of reconstruction. In Table 6, one can see that the final SNR is doubled when we use EC.

2. Splitting the quantization into two steps - quantizing the coefficients and quantizing the error - enables us to use shorter vectors, a fact that is very important for faster computations, as well as achieving high SNR.

3. The values representing the error approximation are very small and thus its wavelet coefficients are small too. This also enables fast computation and fine reconstruction.

3.8 Comparing Different Wavelet Filters

Distinct QMF filters are known. This fact raises the problem which filter to use. We have already investigated the behavior of the wavelet coefficients, but the nature of the coefficents is characterized by the filter according to which they were created. Table 5 presents the compression achieved by applying different types of wavelets. This work uses the biorthogonal wavelet since although its PSNR was a little smaller than the best achieved, its visual quality, judged by several observers, was better.

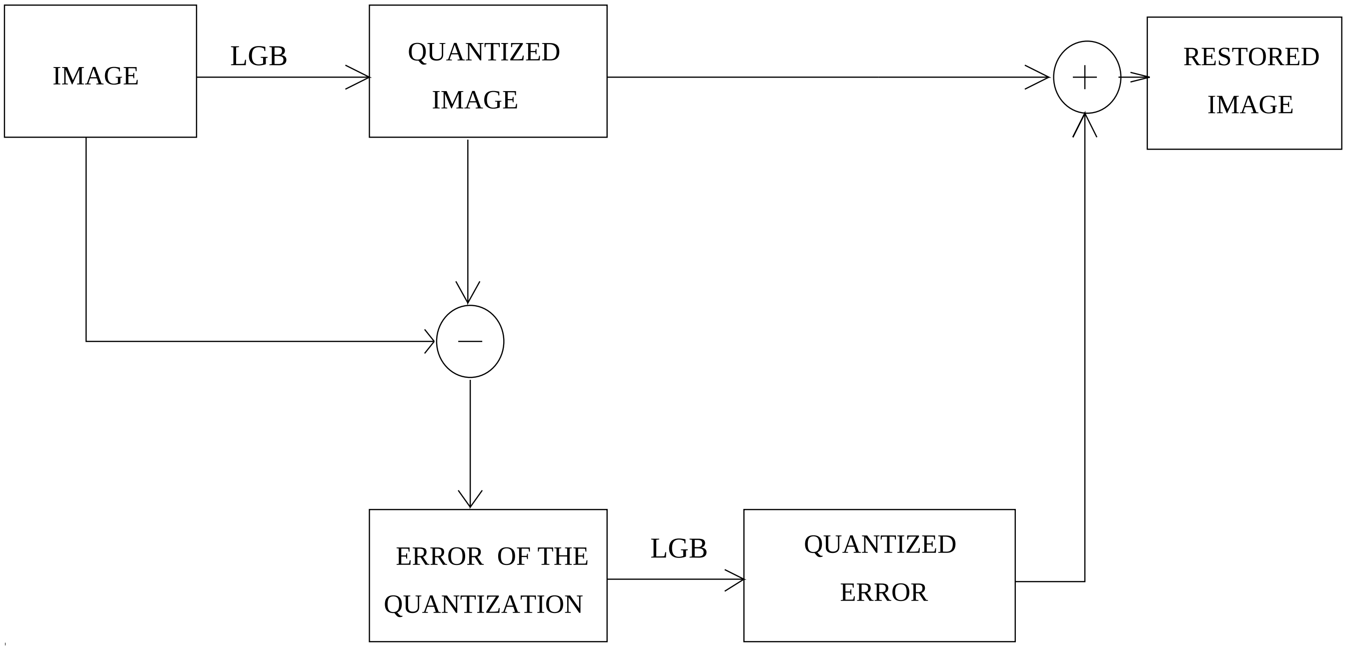


Figure 5: Error Correction Process

Filter Type	Filter length	PSNR	ERMS	Compression: Entropy		Huffman Coding		Without Huff	
				bpp	CR	bpp	CR	bpp	CR
Daubechies	2	29.87	8.18	0.129	61.80	0.129	61.58	0.168	47.51
Daubechies	6	32.06	6.35	0.126	63.14	0.127	62.87	0.168	47.51
Daubechies	8	32.38	6.12	0.127	62.51	0.128	62.11	0.168	47.51
Daubechies	10	32.70	5.90	0.128	62.24	0.129	61.94	0.168	47.51
Daubechies	12	33.10	5.64	0.127	62.59	0.128	62.28	0.168	47.51
Daubechies	16	33.10	5.64	0.128	62.44	0.128	62.09	0.168	47.51
Daubechies	20	32.97	5.72	0.128	62.03	0.129	61.67	0.168	47.51
Byelkin	18	33.41	5.44	0.128	62.05	0.129	61.74	0.168	47.51
Coiflet	6	31.98	6.41	0.128	62.43	0.128	62.15	0.168	47.51
Coiflet	18	33.09	5.64	0.128	62.41	0.128	62.11	0.168	47.51
Coiflet	30	32.83	5.81	0.127	62.99	0.127	62.68	0.168	47.51
BiOrthog	9	32.84	5.81	0.128	62.29	0.129	61.99	0.168	47.51

Table 5: Using different wavelet filters while reconstructing the Lenna's image. The compression achieved is described in three ways: the Entropy measure, the compression achieved after applying the Huffman Code on the resulted vector tables, and the compression resulted without using the Huffman Code. CR brings the compression ratio which is brought in the previous column by bits per pixel (bpp).

4 Vector Quantization of Wavelet Coefficients - Experimental Results

The sensitivity of the quantized representation is higher at low frequencies which need larger table allocations and a higher reconstruction quality than the higher frequencies. This is true for all images because the energy is generally concentrated in the lower frequencies. It is interesting to remark that the human visual system is also much more sensitive to contrast at low frequencies. The evolution of the contrast sensitivity with respect to the frequency of the stimuli is qualitatively the same for a human observer and for our quantized wavelet coefficients.
Thus, using the wavelet transform pyramidal scheme enables us to apply this approach while achieving higher quality and better SNR during quantizing the lower levels with larger allocations. The performance of the wavelet transform as a subband coding which splits each level for three directions (horizontal, vertical and diagonal) implies that shapes 1,2 (horizontal and vertical) should be quantized better than shape 3 (diagonal). At the lower levels, error correction will be used for improving the quality of reconstructed coefficients. The coarser resolution subimage texture (shape 0) is kept unquantized: this group of coefficients contains most of the information and energy of the image (see Table 3). Quantization errors resulted when reconstructing these coefficients yield heavy disturbances at the image. Hence, at this work, shape 0 remains unchanged or quantized using an uniform quantization which minimizes the error and each coefficient needs 10-13 bits (range of $2^{10} - 2^{13}$). Using vector quantization on the coefficients enables to control the total compression ratio by changing N, K that are allocated for each direction.

In the following section we explain the idea of deciding which length of N and K to use at each level and shape.

4.1 Choosing Allocations and Vector Lengthes

Let N be the number of vectors in the quantization table, and K be their length. In order to reach a final predetermined compression ratio, different allocations (resulted in different N, K) were tested while coarser levels had greater allocations and achieved a satisfactory SNR in their reconstruction. After deciding which bit allocations to use at each level, different lengthes of N, K that achieved these allocations (according to $N = 2^{KR}$) were tested. The number of total coefficients ($N \times K$) at the table should not have exceeded a bound which is level dependent, otherwise the computational complexity will be very heavy. Thus this work used N, K that reached the SNR around 12. If the shape was important for the final reconstruction, EC was also applied, and thus SNR higher than 20 was achieved without including longer vectors at the table codebook.

The dependency between SNR and the vector length (at constant allocation $R = 0.5$) is given in Figure 6.

Table 6 presents the final allocations and SNR which are resulted by applying similar analysis to all shapes in Lenna's coefficents. Close results are given applying tose allocations on other images.

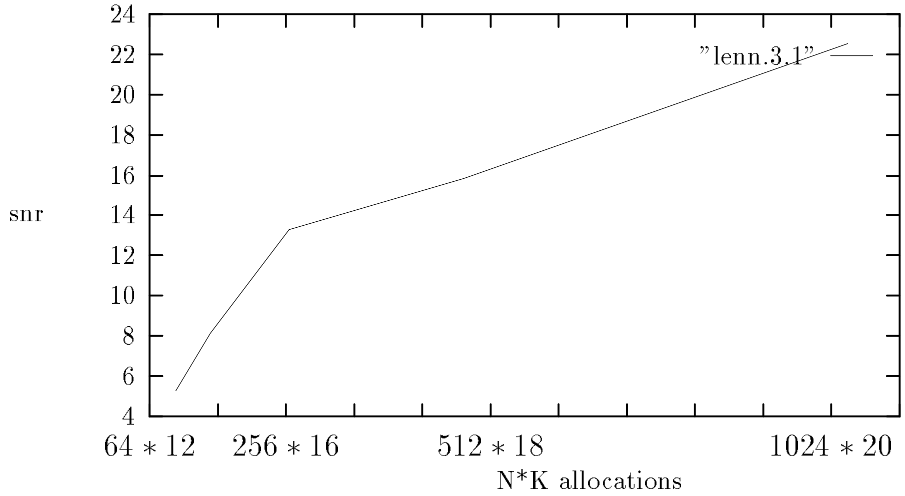


Figure 6: The influence of different allocations on SNR: Different $N \times K$ which satisfies $N = 2^{RK}$ were tested for R=0.5 bpp at shape 1 in level 3. SNR of 13 was reached by allocating N=256 and K=16.

4.2 Allocating Local Tables

The previous section explained the decision of vector length at each shape. The selection of R, N, K for each shape is based on that approach Also, the coefficients were devided into 4 levels. The coefficients of Level 1 were not used, and shape 0 (at level 4) remained unchanged. Applying those allocations on Lenna's coefficients yielded the results at table 6 as well as PSNR of 31.76 and compression ratio of 58.56bpp.
Several images were trained together using those allocations. Those images, which were part of the training sequence, were restored using the reulted tables and achieved compression ratio of 60 and PSNR of 30.17-32.28 (table 7).
The results are brought in figures 7 and 8.

4.3 Allocating Global Tables

In order to create global tables, where images outside the training sequence are being compressed, many images of faces (not calibrated) were trained together. In order to overcome several problems created by quantization such as blurring and stripes, the depth of the multiresolution decomposition tree was based on 3 levels. This fact reduces the compression ratio: shape 0 at level 3 cannot be compressed at the same bit rate as level 4, whose compression was based on the decomposition to different shapes. Level 3, shape 0 was scalar quantized, using 8 bits per sample, contrary to the previous training which kept the coefficients in shape 0 of level 4 unquantized. Different allocations were used, based on the same approaches which were mentioned previously, and the new allocations, as well as the final results of the compressed images inside and outside the training set are brought at tables 8,9.

LEVEL	SHAPE		N	K	bpp Entropy	Huffman	without Huff	SNR	SNR Diff	Final SNR
2	1		512	36	0.202	0.203	0.25	11.09		
		EC	512	36	0.122	0.122	0.25		8.53	19.22
2	2		512	36	0.176	0.177	0.25	10.16		
		EC	512	36	0.158	0.160	0.25		7.63	17.96
2	3		256	80	0.073	0.074	0.10	7.73		7.73
3	1		256	16	0.413	0.414	0.50	13.06		
		EC	256	16	0.267	0.267	0.50		12.09	25.87
3	2		256	16	0.369	0.371	0.50	12.39		
		EC	256	16	0.315	0.320	0.50		12.03	22.44
3	3		256	16	0.377	0.378	0.50	12.54		
		EC	256	16	0.316	0.321	0.50		12.82	24.13
4	1		64	24	0.210	0.214	0.25	10.76		
		EC	64	24	0.091	0.096	0.25		8.53	19.30
4	2		64	24	0.204	0.208	0.25	9.93		
		EC	64	24	0.097	0.101	0.25		11.69	24.38
4	3		64	24	0.209	0.213	0.25	11.59		
		EC	64	24	0.078	0.086	0.25		10.89	23.03

Table 6: The behavior of different levels in Lenna- Local quantization of wavelet coefficients that were derived from BiOrthogonal filter of length 9. For each level and shape N, K represent the number of vectors (N) and their length (K) at the quantization tables. When EC is used, SNR Diff shows the quality of the error approximation. The column of Final SNR brings the SNR of the coefficients at each shape after applying the EC. The Entropy measure is computed, while the columns of Huffamn and without Huffman bring the experimental results of compression before and after applying the Huffman Coding.

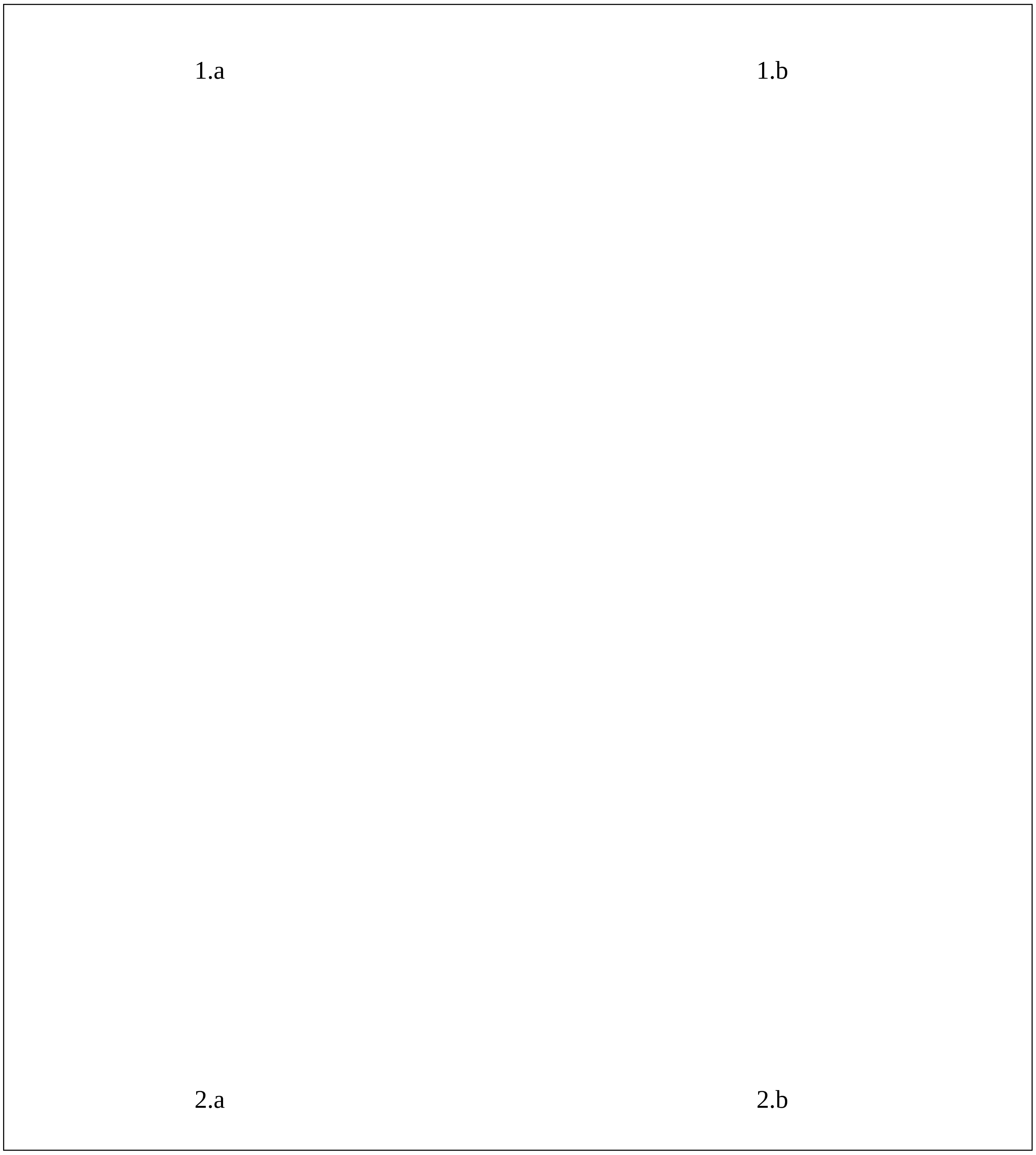


Figure 7: Training several images together and restoring them from the resulted tables. "LENNA": (1.a) original image. (1.b) compression of 0.136 bpp. "JOHN" : (2.a) original image. (2.b) compression of 0.134 bpp.

1.a 1.b

2.a 2.b

Figure 8: Training several images together and restoring them from the resulted tables. "HAGIT": (1.a) original image. (1.b) compression of 0.137 bpp. "CAROL" : (2.a) original image. (2.b) compression of 0.145 bpp.

IMAGE	PSNR	ERMS	Big Err.	Max Diff.	Min Diff.	Compression					
						Entropy		Huffman Coding		Without H.C.	
						bpp	CR	bpp	CR	bpp	CR
LENNA	31.76	6.58	4112	76	-73	0.124	64.03	0.136	58.56	0.168	47.51
JOHN	32.28	6.19	3407	82	-64	0.124	64.50	0.134	59.31	0.168	47.51
CAROL	30.17	7.90	5869	101	-98	0.133	59.71	0.145	54.96	0.172	46.43
HAGIT	27.39	10.88	14867	111	-175	0.125	63.99	0.137	58.02	0.168	47.51

Table 7: The results of training several images together and restoring them from the resulted tables. The compression achieved is described in three ways: the Entropy, the compression achieved after applying the Huffman Code on the resulted vector tables, and the compression resulted without using the Huffman Code. CR brings the compression ratio which is brought in the previous column by bits per pixel (bpp).

IMAGE	PSNR	ERMS	Big Err.	Max Diff.	Min Diff.	Compression	
						bpp	CR
FACE 1	35.05	4.50	699	96	-68	0.21	38.22
FACE 2	36.05	4.02	590	89	-58	0.20	39.23
FACE 3	30.73	7.40	6490	103	-71	0.22	36.34
FACE 4	35.60	4.22	727	139	-58	0.20	38.97

Table 8: The results of restoring several images of faces. FACE 4 is taken from the training set. All the other faces were taken outside the training set.

The results are brought in figures 9 and 10.

5 Vector Bit Allocation Using Several Tables At Each Shape

In our previous compression approach, for each shape, one table was created by the LGB, and all the vectors in that shape were quantized based on that table. Hence, before applying the Huffman coding, equal length binary code words were used for representing the quantizer output. Here we bring another approach, which takes into consideration the variance and the energy of the coefficients at each vector. The idea is that since different vectors (at each shape) have different energy concentrations, they should be quantized using different codebooks of different sizes. Obviously, the vectors with higher energy (which are more important for a fine image reconstruction) should be quantized using the larger codebook, whereas the vectors with a lower energy may be quantized using a smaller codebook.
After deciding on the total amount of bits representing the shape, several lengthes of vectors are examined as was done in the previous method, but for each one of them we change the

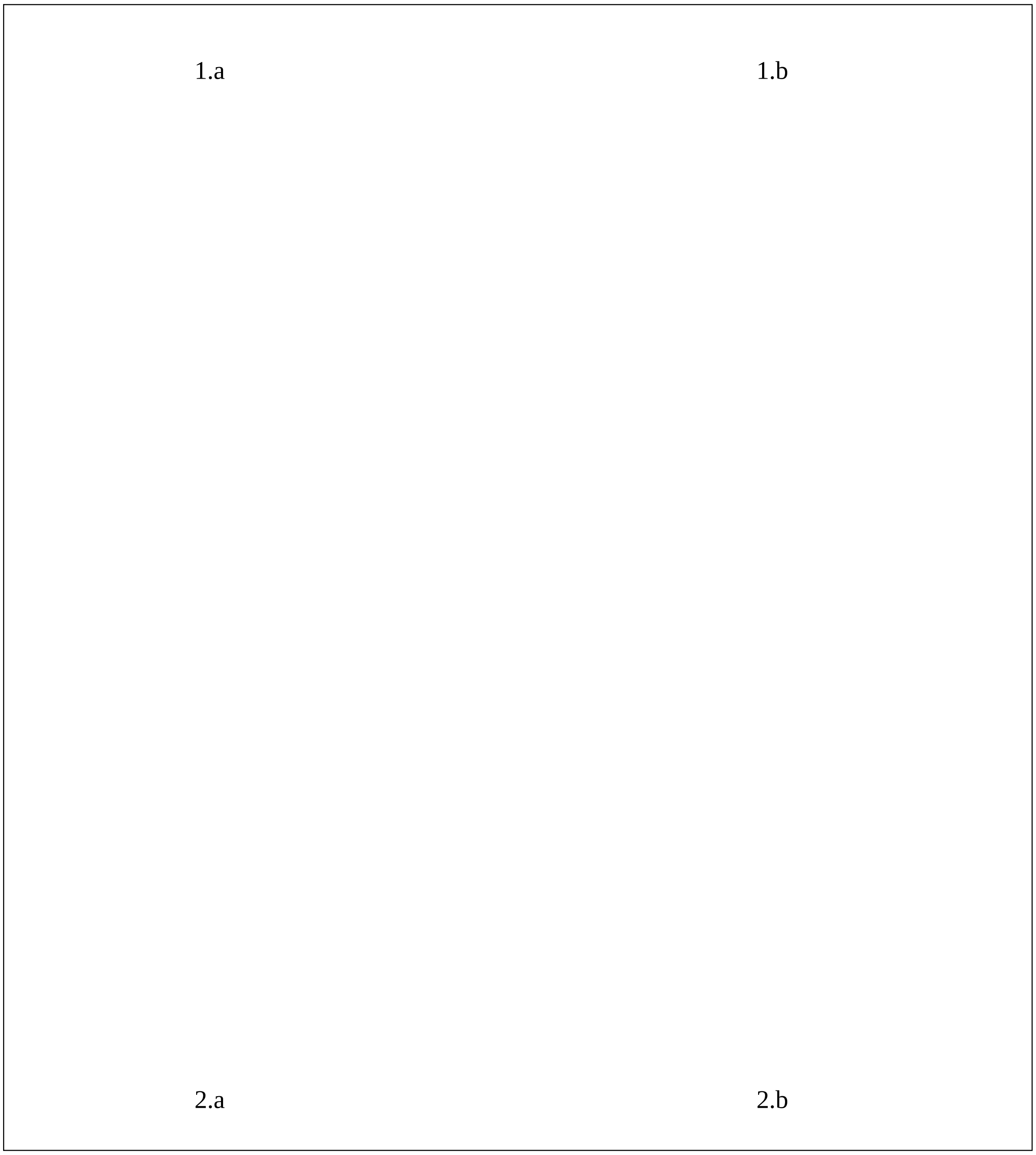


Figure 9: Restoring faces taken outside the training set. FACE 1: (1.a) original image. (1.b) compression of 0.21 bpp. FACE 2: (2.a) original image. (2.b) compression of 0.20 bpp.

1.a 1.b

2.a 2.b

Figure 10: Restoring faces taken outside (1) and inside (2) the training set. FACE 3: (1.a) original image. (1.b) compression of 0.22 bpp. FACE 4: (2.a) original image. (2.b) compression of 0.20 bpp.

Level	Shape		N	K
			128	28
2	1	EC	128	28
			128	28
2	2	EC	128	28
			128	28
2	3	EC	-	-
			128	14
3	1	EC	128	14
			128	14
3	2	EC	128	14
			128	14
3	3	EC	-	-

Table 9: Vector Allocations for global trainings. The EC line brings the allocations for the error correction step.

number of tables that should be used for the shape.

5.1 A Vector Bit-Allocation

Assuming that R is the target average rate in bits per pixel in a given shape, M, N are the dimensions of the shape and K is the length of the vectors, then $[(M \times N)/K] = N_V$ is the number of vectors in the shape and $(M \times N) \times R$ is the total bits available for coding the shape's data.

If $\sigma_{i,k}^2$ is the variance of the ith component of the kth transform vector, we determine the variance of the vector as:

$$\sigma_k^2 = \sum_{i=0}^{K-1} \sigma_{i,k}^2 \tag{11}$$

and if $b_{i,k}$ is the number of bits allocated for the ith components of this vector, than the total bits allocated for this vector should be as follows:

$$b_{i,k} = R + 0.5(\log_2 \sigma_{i,k}^2 - \frac{1}{N \times M} \sum_{k=0}^{N_V - 1} \sum_{i=0}^{K-1} \log_2 \sigma_{i,k}^2) \tag{12}$$

$$b_k = \sum_{i=0}^{K-1} b_{i,k} = K \times R + 0.5 \times K \times (\log_2 \sigma_k^2 - \frac{1}{N_V} \sum_{k=0}^{N_V - 1} \log_2 \sigma_k^2) \tag{13}$$

At each shape, after determining the codebook size for each k, the LGB algorithm can be used to design each codebook. The vectors which belong to the same codebook (in the meaning of the same b_k) are trained together, and a table code of size 2^{b_k} is created. At the compression step, a map file determines for each vector by which table it should be quantized. The process is demonstrated at figure 11.

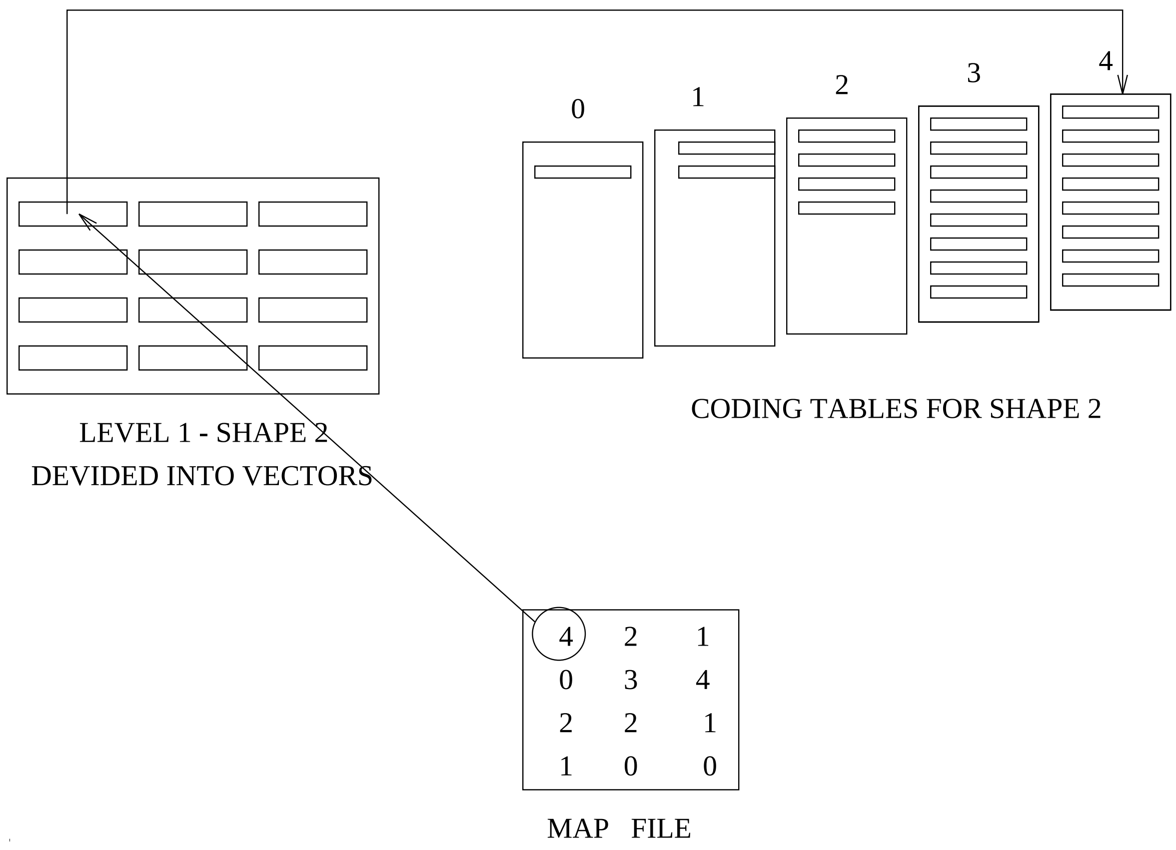


Figure 11: Using a map file, the first vector should be compressed searching a close vector in Table Code number 4.

R	K	$tablenumber$	SNR	bytes
0.25	8	8	3.13	171
0.25	8	9	3.31	187
0.25	8	10	3.31	198
0.25	8	11	3.37	197
0.25	8	12	3.43	180
0.5	8	8	3.69	205
0.5	8	9	3.94	227
0.5	8	10	3.99	241
0.5	8	11	4.16	249
0.5	8	12	4.19	239
1.0	8	8	7.35	260
1.0	8	9	8.10	301
1.0	8	10	8.88	333
1.0	8	11	8.70	354
1.0	8	12	8.56	360

Table 10: Different allocations using several tables at level 3, shape 2

5.2 Implementation of the Vector Transform using Bit-Allocation

In the implementation of vector transform using bit-allocation, a set of look-up tables are used to reproduce quantized transform vectors. In the following, several parameters were checked: R which is the final predetermined average compression rate, K the length of the vectors in a given shape, and $tablesnum$ is the number of tables which are accompanied to each shape. The quality of the results is evaluated according to the values of the SNR of the reconstructed coefficients, the resulted compression after applying Huffman Coding, and the $PSNR$ of the reconstructed images- taken inside and outside the training set.

Table 10 brings the results of several tests applied on the coefficients of shape 2 in level 3. The parameter $tablenumber$ denotes the highest bpp rate which is allocated for the most important vectors. $tablenumber = n$ means that there are n tables accompanied the shape, and table number n has length of 2^n vectors and belongs to vectors that will be compressed to n bpp. The column of $bytes$ presents the number of bytes needed after applying the Huffman Coding to represent the shape. From this table we can conclude that the more tables we use - the better quality of reconstruction we get, but larger tables (which belong to more important vectors) make the compression process much slower, since more vectors should be compared to in order to locate the index of the closest vector in the table. Also, one can see that achieving similar quantity of bytes is accompanied to achieving similar values of SNR, not in relation to the allocations of R, K and $tablenumber$. Hence, using the bit-allocation formula enables using shorter allocations and number of tables- without decreasing the resulted quality and compression.

Observing the results of the other shapes yields similar conclusions.

$LEVEL$	$SHAPE$	R	K	$tablesnumber$	SNR
2	1	1.0	8	8	4.66
2	2	0.5	8	16	4.56
2	3	-	-	-	-
3	1	1.0	16	12	7.29
3	2	1.0	8	12	5.83
3	3	1.0	8	8	4.03

Table 11: Final Allocations while creating Global Tables

			Big	Max	Min	Compression	
IMAGE	PSNR	ERMS	Err.	Diff.	Diff.	bpp	CR
FACE 1	35.47	4.29	610	109	-78	0.22	36.00
FACE 2	36.49	3.81	549	85	-62	0.22	35.80
FACE 3	36.72	3.71	622	130	-63	0.22	35.23

Table 12: The results of restoring several images. FACE 3 is taken from the training set. All the other faces were taken outside the training set.

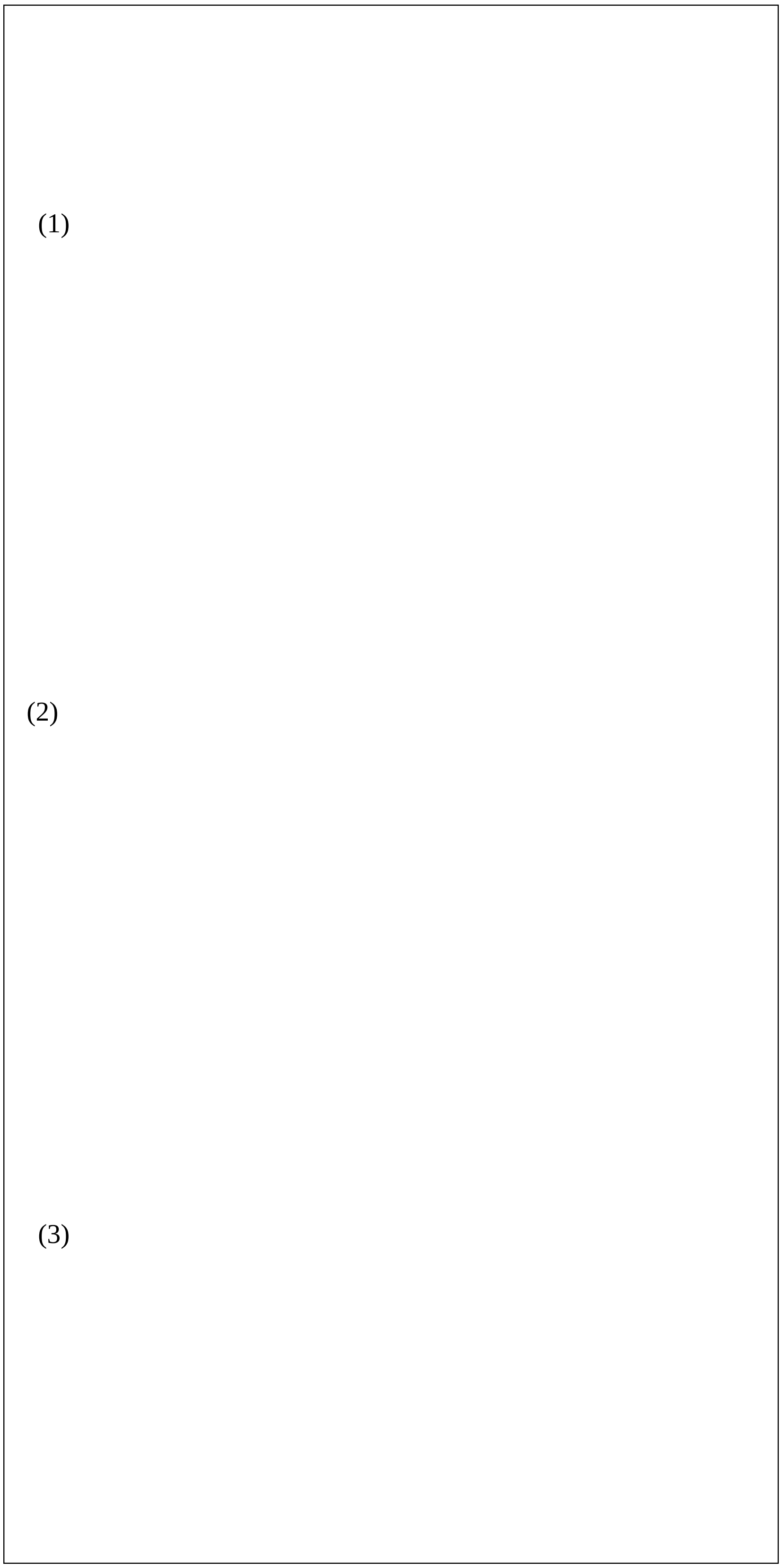
5.3 Allocating Global Tables

Table 11 brings the final allocations which were used while creating the training tables. The SNR column gives the quality of the reconstructed coefficients of an image taken outside the training set. EC is used in shape 1 of level 3. Table 12 shows the final results of compressing images of faces taken within and outside the training set. Our new method, in addition to being much faster, needed smaller tables and allocations, as well as achieving finer quality results.
The results are brought in figures 12 and 10.

6 Conclusions

In this paper, we have developed image coding scheme combining the wavelet transform and the use of the LGB vector quantization.
The wavelet transform used here exploits the assumption that the high frequencies are less visible to the human eye. Indeed, the proposed method enables high compression bit rates while maintaining good visual quality through the use of distinct bit allocations in the sub-images.
Applying the compression method on black and white images within the training set yielded compression ratio of 60-65 and the PSNR was usually above 30. Using similar methods on images of faces and reconstructing compressed faces taken outside the training sequence resulted with compression ratio of 38-40 and PSNR of 30-36.

Figure 12: Restoring faces taken outside (1,2) and inside (3) the training set, using Vector Bit Allocation. The compression is 0.22 bpp.



A compression using vector bit allocation produced similar results, as well as much lower computational cost.

References

[1] N.M. Nasrabadi, R.A. King, *Image Coding Using Vector Quantization: A Review* , IEEE Trans. on Comm, vol 36,No. 8 ,August 1988.

[2] M. Barlaud, P. Mathieu, M. Antonini, *Wavelet Transform Image Coding Using Vector Quantization* ,6 th Workshop on MDSP, Monterey California , Sep. 1989.

[3] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, *Image Coding Using Wavelet Transform* , IEEE Transaction on Image Processing, VOL. 1, No. 2, April 1992, pp. 205-220; IEEE ICASSP 90, Albuquerque USA, pp.2297-2300.

[4] J.C. Feauveau, P. Mathieu, M. Barlaud, M. Antonini, *Recursive Biorthogonal Wavelet Transforn for Image Coding* ,IEEE ICASSP 1991, pp. 2649-2652.

[5] I. Daubechies, *Orthonormal Bases of Compactly Supported Wavelets* , Comm. Pure Appl.Math.41(1988) pp.909-996.

[6] A. Cohen, I. Daubechies and J.C. Feauveau, *Biorthogonal Bases of Compactly Supported Wavelets* , AT&T Bell Laboratories, Technical report, no TM11217-900529-07.

[7] M. Vetterli, C. Herley, *Wavelets and Filter Banks: Relationships and New Results*, IEEE ICASSP, Albuquerque, USA, April 1990.

[8] P.H. Westerink, D.E. Boekee, J. Biemond, J.W. Woods, *Subband Coding of Image Using Vector Quantization* , IEEE Trans. on Comm., Vol. 36, pp. 713-719, 1988.

[9] P.H. Westerink, D.E. Boekee, J. Biemond, J.W. Woods, *Progressive Transmission of Images Using Subband Coding* , IEEE ICASSP 89, pp. 1811-1814.

[10] Peter H. Westerink, Jan Biemond and Dick E. Boekee, *Subband Coding of Color Images.*

[11] S. Mallat, *A Theory for Multiresolution Signal Decomposition: The Wavelet Representation* , IEEE Trans. on Pattern Anal. and Match. intel. (PAMI) ,Vol. 11, No. 7, July 1989, pp. 674-693.

[12] Y. Linde, A. Buzo, R.M. Gray, *An Algorithm for Vector Quantizer Design* , IEEE Transaction on Communications,VOL. COM-28, No.1, pp.84-95, Jan. 1980.

[13] J. Max, *Quantizing for minimum Distortion* , IEEE Trans. on Info. Theory, VOL. IT 6, March 1960.

[14] A. Averbuch, D. Lazar, M. Israeli *Image Compression using Wavelet Transform and Multiresolution Decomposition* , DCC, pp.459, March 1993.