Evolving better wavelet compression schemes

Andreas Klappenecker* and Frank U. May

Universität Karlsruhe
Institut für Algorithmen und Kognitive Systeme,
Am Fasanengarten 5, D–76 128 Karlsruhe, Germany
e-mail: wavelet@ira.uka.de

ABSTRACT

Wavelet based compression schemes belong to the general class of transform coding schemes. We show how the genetic programming approach can be used to optimize such a compression scheme in the sense of rate-distortion. The results of optimized wavelet based compression schemes are compared with the JPEG compression standard. A prototype implementation of the method is realized as a distributed, parallel implementation on a heterogeneous Unix network.

Keywords: Wavelets, compression, optimization, genetic programming.

1 INTRODUCTION

Lossy image data compression is an impressive application of wavelet algorithms. The aim is to implement an efficient compression scheme, which is flexible enough to cover a great variety of bit rates while achieving a minimum of distortion. This goal can only be attained if the scheme is adapted to the human visual system as well as to the image class considered. We show how the genetic programming paradigm[8] can be used to optimize wavelet based compression schemes in the sense of rate-distortion with an arbitrary computable distortion function.

Wavelet based compression schemes belong to the general class of transform coding schemes. A transform coding scheme can be divided into three major steps: transformation of the input signal, quantization of the transform coefficients, and entropy coding of the quantizer output. The transformation in our compression scheme can be described by a decomposition tree (see below) or alternatively by the corresponding basis.

Coifman, Meyer, Quake, and Wickerhauser (CMQW)[1] have developed an algorithm that allows to choose a basis with minimum "information cost" from a library of bases. Unfortunately, in their approach the information cost function has to be additive. The cost functions considered in CMQW[1] include a bit counting measure and an additive measure based on entropy. More recently, Ramchandran and Vetterli[13] extended this algorithm in order to optimize both rate and distortion. However, they still require the distortion measure to be additive.

We propose a method to optimize wavelet based compression schemes in rate-distortion sense allowing an

* This research was supported by Deutsche Forschungs-Gemeinschaft (DFG) under project Be 887/6–2.

arbitrary (possibly non-additive) computable distortion function. Moreover, our procedure does not only adapt the basis, but improves the whole compression scheme including the crucial point of choosing the appropriate quantizers.

2 WAVELET COMPRESSION SCHEMES

Recently, there has been a lot of interest in wavelet based compression and many different methods were proposed. In this section, we briefly describe the compression method used here and fix some notation. For simplicity, we restrict our description to one-dimensional signals. The algorithms can easily be extended to higher dimensions by using tensor products.[3,7,12] The generalization to two-dimensional images is straightforward.

The input signal is transformed in order to decorrelate consecutive samples. To increase the compression ratio, the entropy of the resulting coefficients is reduced by quantization. We use simple scalar quantizers[4,15] to allow efficient implementation in hardware. Finally, the quantized coefficients are passed through a simple entropy coder like an arithmetic coder.[9]

Transform. Denote by S_k the shift operator $S_k\,(s_n)_{n\in\mathbb{Z}} = (s_{n-k})_{n\in\mathbb{Z}}$. Recall that a pair of sequences $G = (g_i)_{i\in\mathbb{Z}}$, $H = (h_i)_{i\in\mathbb{Z}} \in \ell^2(\mathbb{Z})$ is called a *Conjugate Quadrature Filter*[2,7,14] (CQF) pair iff $\{\, S_{2k}\,G,\ S_{2k}\,H \mid k \in \mathbb{Z}\,\}$ is an orthonormal basis of $\ell^2(\mathbb{Z})$, and the sequences G and H are related by $g_n = (-1)^n\,\bar{h}_{1-n}$. Associated with (G, H) are projectors $F_G, F_H : \ell^2(\mathbb{Z}) \to \ell^2(\mathbb{Z})$ operating on an arbitrary sequence $s = (s_i)_{i\in\mathbb{Z}} \in \ell^2(\mathbb{Z})$ as follows:

$$F_G\,s = \left(\sum_{m\in\mathbb{Z}} \overline{g_{m-2l}}\,s_m\right)_{l\in\mathbb{Z}} \qquad \text{and} \qquad F_H\,s = \left(\sum_{m\in\mathbb{Z}} \overline{h_{m-2l}}\,s_m\right)_{l\in\mathbb{Z}}.$$

The operators F_G, F_H project a signal sequence s onto subspaces of $\ell^2(\mathbb{Z})$. We use this pair of projections as a basic decomposition step. Of course, we may apply this decomposition recursively to the resulting subspaces. For example, applying another pair of operators F_D, F_E to the subspace $F_G\,\ell^2(\mathbb{Z})$ yields another two subspaces $F_D F_G\,\ell^2(\mathbb{Z}), F_E F_G\,\ell^2(\mathbb{Z})$, etc. We call the resulting hierarchy of subspaces a *decomposition tree.* As a consequence of the CQF property, it is possible to perfectly reconstruct the signal $s \in \ell^2(\mathbb{Z})$ from its projections onto the leaves of the decomposition tree. These projections are called *subbands*.

Quantization and Coding. After projecting the signal onto the leaves of the decomposition tree, a quantization operation is applied to each subband. Scalar quantization applies a non-linear mapping to each coefficient in a subband. This mapping approximates the coefficients with a value chosen from a small finite set.[4,15] The quantization has two effects: On the one hand, the entropy is reduced and thus the compression ratio is increased, on the other hand distortion is introduced. The amount of distortion and the compression ratio depend on the cardinality and the values of the quantizer output range.[4,15] The actual coding is done by an adaptive arithmetic coder.[9,16]

Representation. For illustration, we introduce a tree representation for a compression scheme including the decomposition tree and the quantizers. We label the leaves of the tree with the quantizer names and the internal nodes with the name of the corresponding CQF pair. Taking two CQF pairs (G, H) and (D, E), we can e.g. project $\ell^2(\mathbb{Z})$ onto three subspaces $F_D F_G\,\ell^2(\mathbb{Z})$, $F_E F_G\,\ell^2(\mathbb{Z})$, and $F_H\,\ell^2(\mathbb{Z})$, followed by quantization with the scalar quantizers Q_1, Q_2, and Q_3 respectively. The corresponding tree is shown in figure 1 as tree (a).

3 CROSSOVER AND MUTATION

Our method to optimize the parameters of our compression scheme in rate-distortion sense is based on the principle of evolution. The idea is to have a big population of compression schemes and to generate new members

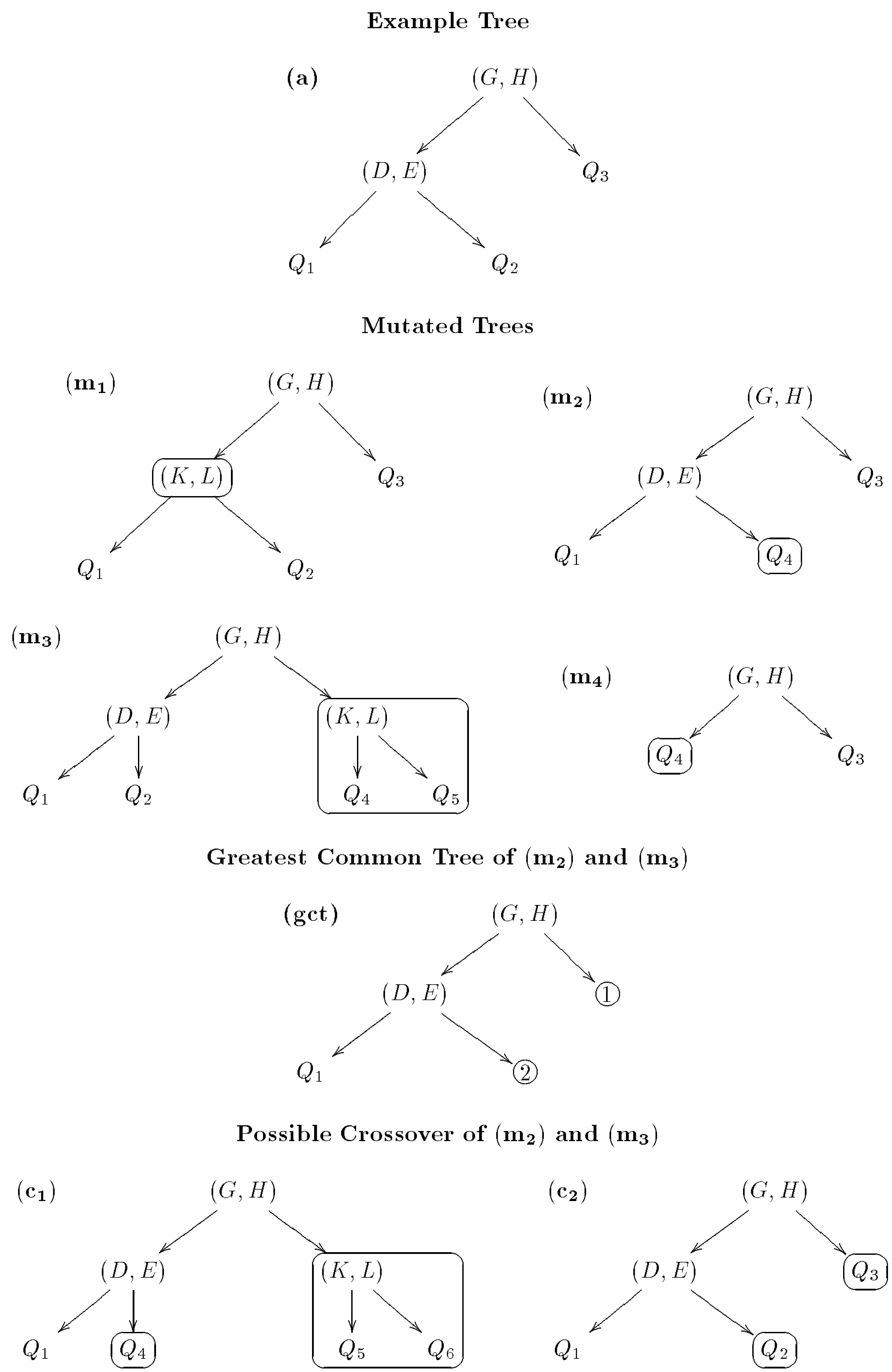


Figure 1: Examples for crossover and mutation

by *crossover* and *mutation* operations from randomly selected members of the population. In this random selection the *fittest* members (in the sense of rate-distortion) are preferred.

Mutation. The crossover and mutation operations are best explained by example. The mutation operation creates a new tree by changing an old tree at one randomly selected node. Four different kinds of mutation may occur. Trees $(\mathrm{m}_1), \ldots, (\mathrm{m}_4)$ in figure 1 show mutations of tree (a) and are examples for all kinds of mutation. Tree (m_1) is generated by substituting the CQF pair (D, E) in tree (a) by another randomly selected CQF pair (K, L). Modifying some parameters of the quantizer Q_2 in tree (a) yields tree (m_2) with a new quantizer Q_4. These mutations do not alter the structure of the decomposition tree. In tree (m_3) the structure is changed by substituting quantizer Q_3 with a new randomly generated subtree of depth one. As a result of the mutation operation even a complete subtree may be substituted by a randomly chosen quantizer, as shown in tree (m_4).

Crossover. The crossover operation uses two "parent" compression schemes to generate two "child" compression schemes. First, the *Greatest Common Tree* (GCT) of the parents is determined. Roughly speaking, the GCT represents the common structures of both parents. In a tree search starting at the root, all equally labelled nodes are collected. A more detailed description is given in appendix A. For example, the GCT of the trees (m_2) and (m_3) is shown in tree (gct) in figure 1.

The nodes where the two parents differ are marked with $(1), (2)$ in the Greatest Common Tree (gct). These nodes are called *crossover nodes.* The childs are composed of the Greatest Common Tree and the possible subtrees found in the parents at the crossover nodes. To form the first child one of the two possible subtrees is randomly chosen at each crossover node. If the Greatest Common Tree has N crossover nodes, then the two possible choices at each crossover node result in 2^N possibilities to form the first child tree. In order to use all structures present in the parent trees, the second child is formed by selecting the alternative subtree at each crossover node. Thus, the second child is completely determined once the first child is formed.

In figure 1 the trees (c_1) and (c_2) are one possibility for selecting two childs of the parents (m_2) and (m_3).

4 FITNESS AND SELECTION

The *selection of the fittest* plays a key role in evolutionary processes. The fitness of a particular compression scheme depends on two quantities: the compression ratio and the distortion introduced. The *compression ratio* is defined as the ratio of the size of the original signal to the size of the compressed signal (measured in bytes). The *distortion* resulting from the quantization operation can be measured with an arbitrary computable distortion function, for example, the distance measures induced by ℓ^p-norms. In our examples the signals are two-dimensional images. A commonly used distortion measure for an original image (x_i) and a distorted image (y_i) is the *peak-to-peak signal-to-noise ratio* (PSNR) measured in dB:

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{d^2}{\mathrm{MSE}} \right),$$

where d is the *dynamic range,* i.e. the difference between maximal and minimal value of the original image, and MSE is the mean square error defined as

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=0}^{n-1} (x_i - y_i)^2.$$

Clearly, a distortion measure for images should reflect the limitations of the human visual system.[6] For example, high frequency noise is not noticeable in smooth images. Several visually based distortion measures have been proposed in literature, e.g. Macq[10] or Mannos and Sakrisson.[11] Due to non-linearities in the human visual system, we cannot expect those measures to be additive. Unfortunately, there is currently no common agreement on a particular measure.

Fitness. The calculation of fitness is best described by using the rate-distortion diagram in figure 2. The diagram shows a part of a population of compression schemes applied to the same image. Each compression scheme is represented by one point in the diagram. A member of the population is called *Pareto-optimal,* if there is no other member in the population with both higher PSNR value and higher compression ratio. The Pareto-optimal members are the fittest members in the population (marked with a line in the diagram).

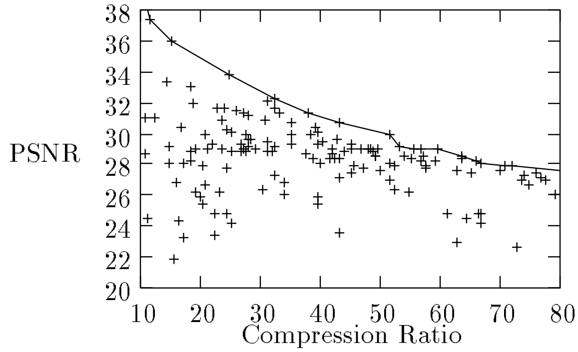


Figure 2: Fitness of a population.

A *fitness value* is assigned to each member of the population by the following algorithm:

1. Set $f := 1$.
2. Assign fitness f to all Pareto-optimal members.
3. Increase f by one.
4. Continue with step 2, but do not consider members with fitness value already assigned.

This algorithm partitions the population in classes of the same fitness values.

Selection. The evolutionary process uses an operation that randomly selects a member for reproduction. In this selection the fitter members should be preferred. We first select a fitness class. In this random selection the probability distribution is adapted such that the fitter classes have higher probability. Then a member out of this class is selected. Here we use a probability distribution which is uniform with respect to compression ratio. This policy avoids clustering of individuals around a specific compression ratio.

5 EVOLUTION

We are now able to describe the complete optimization strategy. The method uses a simple model of an evolutionary process. Essentially, this method incorporates an adaptive search mechanism. Starting with a randomly generated population, the mutation and crossover operations described in section 3 are used to generate new members of the population. Fitness proportionate selection of the parent members leads eventually to an overall improvement of the fitness of the whole population.

In order to control the size of the population, some "bad" members are deleted after adding the newly generated members. To select the bad members, we use the same selection as described in section 4, but with the complementary probability distribution of the fitness values. The optimization procedure can be stated algorithmically as in figure 4. In each iteration step about 5 % of the population are removed and newly generated by reproduction.

Due to the random operations, evolutionary methods require large populations. The computation of the fitness values for these large populations is computationally expensive, as a complete compression/decompression run

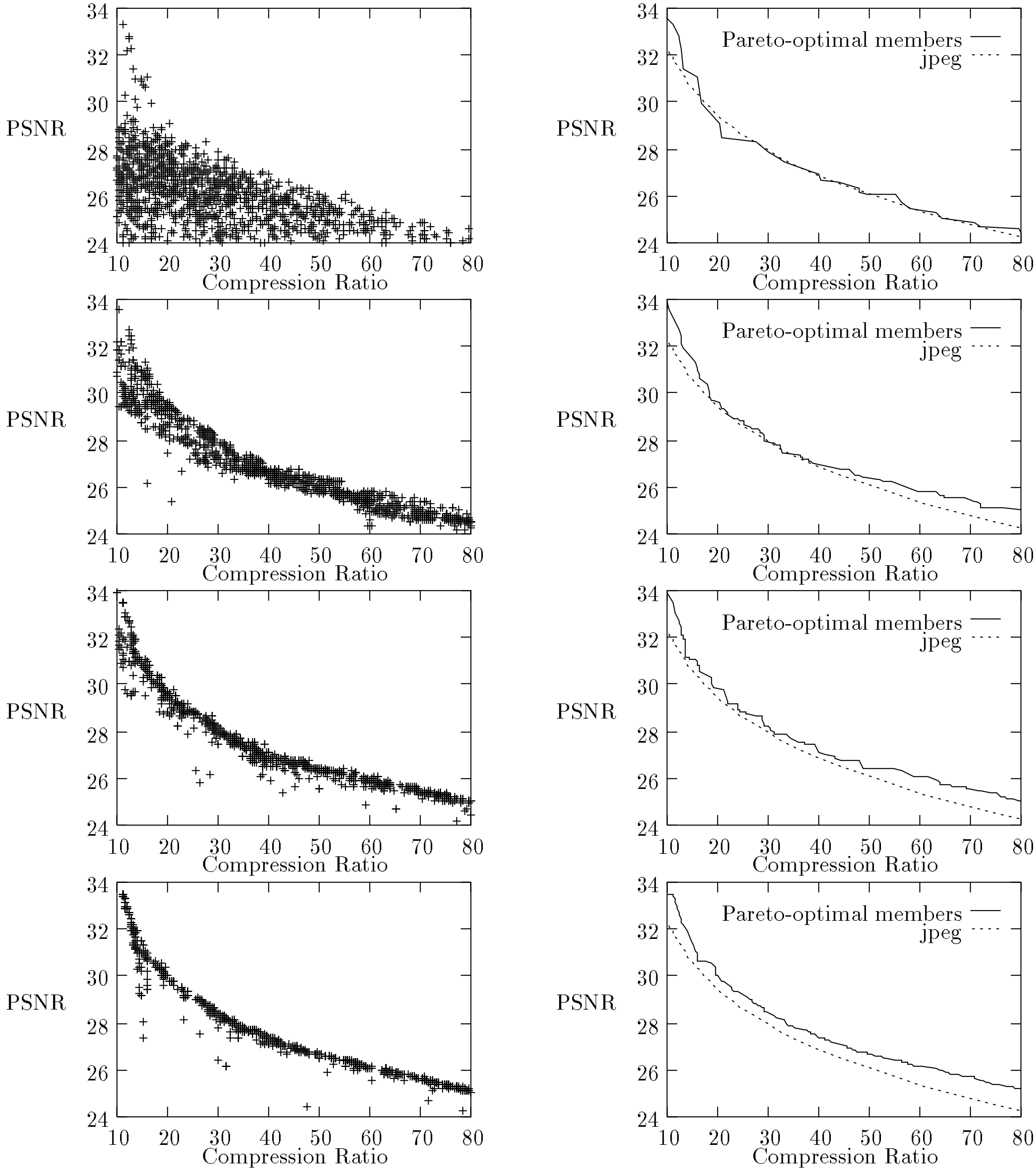


Figure 3: Some snapshots of the evolution.

1. `Generate a random initial population.`
2. `Compute the fitness values (cf. section 4).`
3. `Randomly select parents for crossover and mutation operations. The resulting compression schemes are added to the population.`

   The probability distribution is adapted according to fitness in rate distortion sense (cf. section 4).
4. `Compute the fitness values.`
5. `Remove randomly chosen members from the population.`

   The probability distribution is adapted according to inverse fitness.
6. `Continue with step 2.`

Figure 4: The evolution procedure

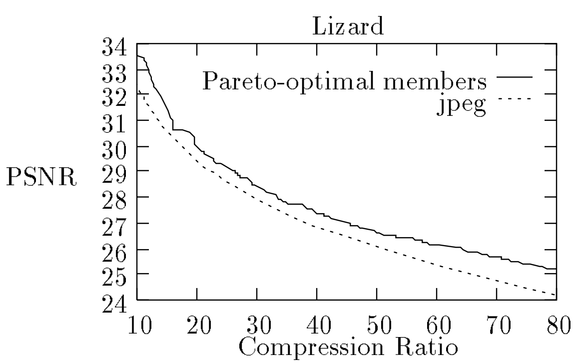
is necessary for each member. However, this big number of independent fitness evaluations can be computed in parallel. Our implementation runs on a Unix-cluster with 18 workstations. This allows the calculation of up to 100 fitness values per minute.

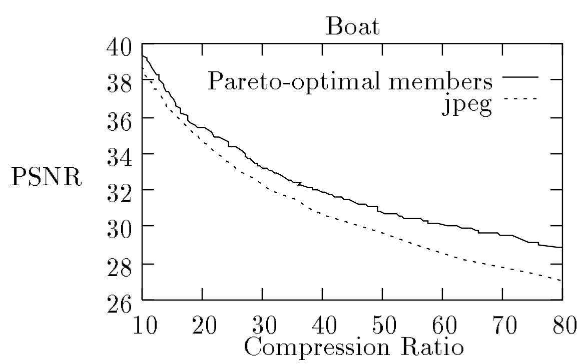
To give an idea of the progress of evolution, figure 3 from top to bottom shows some snapshots from the initial population to the 700th iteration step. The left diagrams show a part of the population (each member plotted as a cross). The diagrams on the right show the corresponding Pareto-optimal members in comparison with the JPEG compression standard.[5] The improvement of the whole population is obvious.

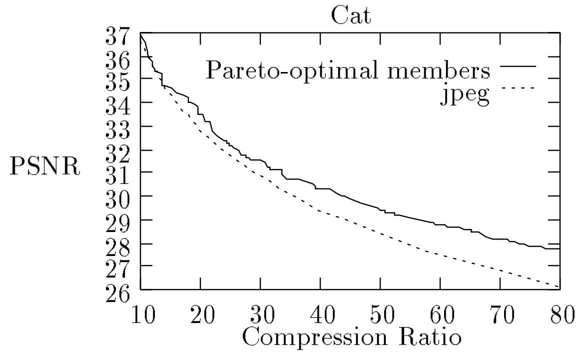
6 RESULTS

We compare the results of the proposed method for three test images with the JPEG compression standard[5] (with different levels of quality and optimized Huffman tables). A constant population size of 1800 members was used in all examples. The procedure in figure 4 was iterated about 500 times. In each iteration 5 % of the population were renewed.

The Pareto-optimal wavelet compression schemes perfom better than the JPEG compression standard in the whole range of compression ratios. For higher compression ratios the advantage of the wavelet based compression schemes is even more striking. The corresponding original and compressed images at the compression ratios 10, 20, 40 and 80 can be obtained from our WWW server `http://iaks-www.ira.uka.de/iaks-beth/wavelet`.







7 CONCLUSION

We have proposed an optimization method for wavelet based compression schemes. Our method is not dependend on specific features of the distortion function, nor on particular choices of the wavelet bases, the quantizers or the coder. Therefore, our optimization method is particularly well-suited for evaluating the performance of different designs of wavelet based coders/decoders. Our wavelet based compression schemes perform better than the JPEG compression standard.

A THE GREATEST COMMON TREE

In this section we define the Greatest Common Tree of two given binary trees. We denote the root of tree (x) by `root`(x). The left and right subtrees are denoted by `l_subtree`(x) and `r_subtree`(x) respectively. This is illustrated in the following figure.

(x) `root`(x)

`l_subtree`(x) `r_subtree`(x)

A new tree is constructed with a constructor `tree` from a new node and two existing subtrees such that the following identity holds:

(x) = `tree`(`root`(x), `l_subtree`(x), `r_subtree`(x)).

A leaf is a subtree created from a node l and two empty subtrees `nil`, i.e. `tree`(l, `nil`, `nil`). Using these notations, the Greatest Common Tree can be defined recursively as follows:

```
gct(a, b) ::=
    if a = nil or b = nil
       then gct(a, b) = nil
       else  if root(a) = root(b)*
                then gct(a, b) = tree(root(a),
                                      gct(l_subtree(a), l_subtree(b)),
                                      gct(r_subtree(a), r_subtree(b)))
                else  gct(a, b) is a crossover node.
```

*Two nodes are considered equal if they represent the same CQF pair or the same quantizer.

8 REFERENCES

[1] R. R. Coifman, Y. Meyer, S. Quake, and M. V. Wickerhauser. Signal processing and compression with wavelet packets. In Y. Meyer and S. Roques, editors, *Progress in Wavelet Analysis and Applications*, pages 77–93. Ed. Frontieres, 1992.

[2] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 41:909–996, 1988.

[3] I. Daubechies. *Ten Lectures on Wavelets*. CBMS-NSF Reg. Conf. Series Appl. Math. SIAM, 1992.

[4] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

[5] Independent JPEG Group. JPEG compression software, release 5beta2. Available on the Internet at `ftp://ftp.uu.net/graphics/jpeg`.

[6] G. Hauske. *Systemtheorie der visuellen Wahrnehmung*. Teubner, Stuttgart, 1994.

[7] M. Holschneider. *Wavelets – An Analysis Tool*. Oxford University Press, 1995.

[8] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1993. 2nd printing.

[9] G. G. Langdon. An introduction to arithmetic coding. *IBM J. Res. Devel.*, 28(2):135–149, 1984.

[10] B. Macq. Weighted optimum bit allocation to orthogonal transform of picture coding. *IEEE Sel. Areas in Comm.*, 10(5), 1992.

[11] J. L. Mannos and D. J. Sakrisson. The effects of a visual fidelity criterion on the encoding of images. *IEEE Trans. Inf. Theory*, 20(4):525–536, 1974.

[12] Y. Meyer. *Ondelettes et Opérateurs*, volume 1. Hermann, Paris, 1990.

[13] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Trans. Image Proc.*, 2(2):160–175, 1993.

[14] M. J. T. Smith and T. P. Barnwell. Exact reconstruction techniques for tree structured subband coders. *IEEE ASSP*, 34:434–441, 1986.

[15] R. Veldhuis and M. Breeuwer. *An Introduction to Source Coding*. Prentice Hall, 1993.

[16] I. A. Witten, M. N. Radford, and J. G. Cleary. Arithmetic coding for data compression. *Comm. ACM*, 30(6):520–540, 1987.