EOS/SPIE Visual Communications and PACS for Medical Applications '93, Berlin, Germany, 1993

Fast hierarchical codebook search for fractal coding of still images

Bernd Hürtgen and Christoph Stiller

Institute for Communication Engineering
Aachen University of Technology
Melatener Str. 23, 52056 Aachen, Germany
Tel.:+49–241–80 7683, Fax:+49–241–8888 196
Email: huertgen@ient.rwth-aachen.de

ABSTRACT

This paper presents a method for fast encoding of still images based on iterated function systems (IFSs). The major disadvantage of this coding approach, usually referred to as *fractal coding*, is the high computational effort of the encoding process compared to e.g. the JPEG algorithm [1]. This is mainly due to the costly "full search" of the transform parameters within a fractal codebook.

We therefore propose an hierarchical encoding scheme which is based upon a two level codebook search and a structural classification of its entries. By this way only a small subset of the codebook has to be considered, which increases encoding speed significantly. Refining the initial codebook and applying a second search even increases the reconstruction quality compared to the full search but with a fraction of its computational effort.

1. INTRODUCTION

Due to the increasing number of digital image processing applications the need for efficient coding of pictorial data becomes evident. Image coding using a fractal approach has attracted some degree of interest in the last years. Most common block oriented coding techniques usually revolve around *transform coding*, e.g. [2] and *vector quantization*, e.g. [3]. The fractal coding concept, originally proposed by Barnsley [4] and at first implemented by Jacquin [5, 6, 7], is based on a blockwise approximation of the original image by contraction mappings of itself using affine transformations.

For a given image the encoding process consists of finding among a class of *a priori* defined contractive transformations one which leaves this image approximately invariant. According to Banach's fixed point theorem the sequence of reconstructed images converges for any arbitrary initial image to the fixed point of the transformation which is the original image. Compression is achieved if the transform parameters can be described more compactly than the original image.

The paper is organized as follows: The second section describes the mathematical foundations concerning encoding and decoding of still images using the fractal approach. Following this, the generation of the fractal codebook is described and some of its statistical properties are derived. In section 4 a two step search algorithm and a structural classification of the codebook entries are introduced. Recent results presented in section 5 demonstrate the efficiency of the proposed coding scheme.

2. MATHEMATICAL FOUNDATIONS

Let $(\mathbf{X}, d)$ be a complete metric space with metric d, and $W : \mathbf{X} \rightarrow \mathbf{X}$ be a transformation which maps the space $\mathbf{X}$ onto itself. W is called *contractive* or a *contraction mapping* if there exists a real constant $\lambda \in [0, 1)$ such that

$$d(W(\mathrm{x}), W(\mathrm{y})) \leq \lambda\, d(\mathrm{x},\mathrm{y}) \quad \forall\, \mathrm{x}, \mathrm{y} \in \mathbf{X} \tag{1}$$

holds. According to [4] λ is called the *contractivity factor* of W. Banach's fixed point theorem ensures, that in this case there exists a unique attractor

$$A = W(A) \tag{2}$$

which is invariant with respect to W. Furthermore, the attractor A is the limit of various approximating sequences of sets which can be constructed from W. This means that A is *fixed point* of the sequence of iterates $\{A^n\}$ $(A^n \in \mathbf{X}\ \forall\, \mathrm{n} \in \{0, 1, 2, \cdots\})$ with

$$A^{n+1} = W(A^n) = W^{\circ(n+1)}\big(A^0\big) \tag{3}$$

for any arbitrary initial element A^0.

Let us now think of B as an image which we want to encode. Our goal is then to find a contractive transformation W such that the given image B is invariant with respect to the transformation W[1]. Equations (2) and (3) give us an idea how B can be reconstructed from the fractal code W:

Let A^0 be any arbitrary initial image. Due to the contractivity of the mapping W, the sequence of images $\{A^n\}$, which can be constructed according to equation (3), tends towards a final image, which is the attractor of the mapping and can be written as

$$A^\infty = \lim_{n\to\infty} A^n = \lim_{n\to\infty} W^{\circ n}\big(A^0\big). \tag{4}$$

Due to the presupposition of invariance formulated in equation (2), the attractor of the transformation A^∞ and the given image B we want to encode, must be identical:

$$d(A^\infty, B) = \lim_{n\to\infty} d(A^n, B) = 0 \tag{5}$$

This means that the given image B can solely be reconstructed from the knowledge of the appropriate transformation W. Therefore W often is referred to as *fractal code* since it can be interpreted as representation of the given image in the *fractal domain*.

---

[1] Since former methods directed to computer animation generate naturally looking images from a set of given contractive transformations, this procedure is known as the *inverse problem*.

3. CODEBOOK GENERATION

Grey tone images of size $NB \times NB$ can be modeled as points in the linear vector space $\mathbf{X} = \mathbb{R}^{\mathbf{NB}\times\mathbf{NB}}$, which is the set of all real $NB \times NB$—matrices. For natural images, which are considered here, no practical algorithms are known in order to find a transformation which maps the entire image onto itself. Therefore the image is segmented into N^2 rectangular non overlapping blocks of size $B \times B$ pixels and the transformation $W = \cup_{k=1}^{N} w_k$ is defined block wise.

According to the theory of *recurrent iterated function systems (RIFS)* [8, 9], which are the mathematical basis for fractal image coding, it is important to note that the w_k do not operate on the entire image but are restricted only to parts of it denoted by D_k. This means that $w_k(D_k) = R_k$ maps the area of block D_k on the area of block R_k which is illustrated in figure 1. Following Jacquin’s proposal, e.g. [7], the blocks R_k, which are to be encoded, are termed *range blocks* and the blocks denoted by D_k are the so called *domain blocks*. The displacement between the block R_k and D_k, which is part of the transformation w_k, is denoted by $\vec{b}^{(k)}$ For our application we imply that the R_k’s are disjoint and $\cup_{k=1}^{N} R_k$ covers the entire area of the image.

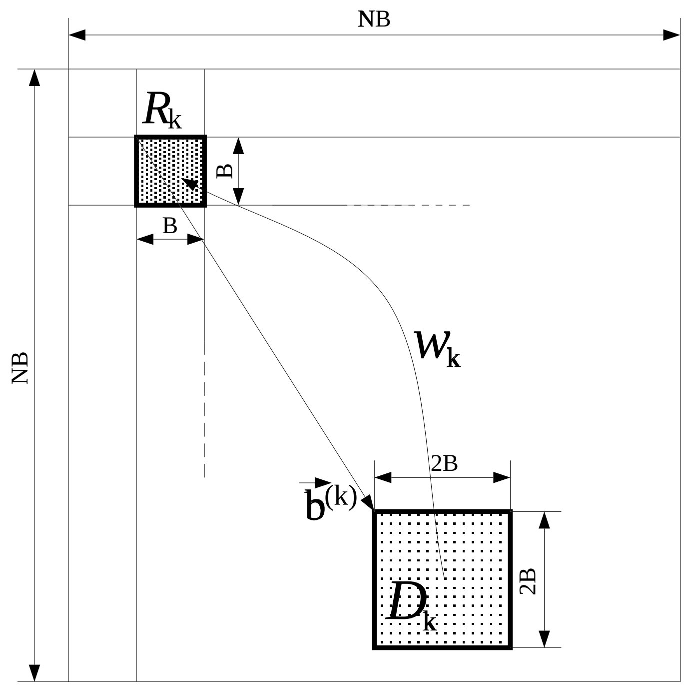

Figure 1: Image tiling

Since we are interested in *encoding* of images, it is evident that the fractal code should be as compact as possible. This means that it can be stored more efficiently (requiring less amount of data) than the original image. The local mappings w_k are therefore restricted to *affine transformations*.

In order to keep the computational effort low, the values for the parameter of the local transformations w_k cannot be chosen arbitrarily but have to be calculated from a *a priori* given set of allowed values. Therefore the domain blocks are always twice the size in each direction as the range blocks. Additionally not all affine transformations are allowed but only the eight possible isometrics which map a square onto another one. The components of the offset vector $(b_x, b_y)^T$ are restricted to be a multiple of the block size of R_k and it must be ensured that both, the range- and the domain blocks lie within the considered image area. All these limitations cause the number M of possible transformations w_k to be finite. This enables us to construct a *fractal codebook* of which each entry defines one possible combination of transform parameters describing a mapping from one block within the considered image onto another. Since the transformations are applied to image blocks, the codebook entries $L_k = w_k(D_k)$ are image blocks too and may be considered as *fractal basis functions*. Encoding now consists of finding for every range block R_k this basis function L_l which fits best in the sense of the used distortion measure d.

$$d(R_k, L_l) \rightarrow \min \quad \forall \quad l \in \{1, 2, \ldots, M\}. \tag{6}$$

Due to the restrictions we made to the possible mappings, it is not very likely to find a transformation W which is a) contractive and b) maps the given image or parts of it exactly onto itself. The goal is therefore to construct a mapping $\widetilde{W}$ in a way that its fixed point $\widetilde{A}$ is *close* — in the sense of a proper chosen distance measure d — to the given image A which has to be encoded. If this is possible, the *collage theorem* [10] ensures, that if we can find an approximation $\widetilde{W}$ to W, the attractor

$$\widetilde{A} = \widetilde{W}\left(\tilde{A}\right) \approx A \tag{7}$$

will be an approximation to A.

One common distance measure is the *Euclidean distance* which can be written as

$$d_E(R, L) = \frac{1}{B^2} \sqrt{\sum_{i=0}^{B-1} \sum_{j=0}^{B-1} \left(R(x_R + i, y_R + j) - L(x_L + i, y_L + j)\right)^2} \tag{8}$$

with R and L denoting two image blocks of size $B \times B$ pixels with their upper left corner at position (x_R, y_R) and (x_L, y_L) respectively. Since we want to represent grey scale values, two additional parameter $s^{(k)}$ and $o^{(k)}$ are introduced, which allow adjusting contrast and brightness of the considered image partition. They have to be taken into account when calculating the distortion measure, which is done by minimizing for each range block R_k the distance

$$d_E\left(R_k,\ s^{(k)} L_l + o^{(k)}\right) \rightarrow \min \tag{9}$$

with respect to the scaling parameter $s^{(k)}$, the offset parameter $o^{(k)}$, and the optimal fractal basis function L_l. By using the Euclidean metric the *least mean square algorithm* offers the optimal solution for $s^{(k)}$ and $o^{(k)}$ if the range block R_k and the corresponding library block L_l are given.

4. CODEBOOK SEARCH

The idea of fractal block coding suffers from its great computational effort due to the numerous distance calculations. As is shown in the following, a hierarchical search algorithm combined with a structural block analysis is capable of reducing the computational effort to a fraction but still yields almost the same reconstruction quality. As mentioned above, the great but finite number of possible transformations enables us to perform a "full search" in order to find the best library block for each range block. For typical applications the number of range- and domain blocks is about 4096 each. With eight possible isometric mappings the huge number of $4096 \times 4096 \times 8 \approx 134 \times 10^6$ distance calculations have to be performed. This simple example illustrates that it is necessary to improve encoding speed in order to make the fractal coding scheme feasible. We therefore propose to reduce the computational burden in the following way:

- Not all library blocks are tested for an appropriate match and
- the calculation of the distance d_E according to equation (8) is partially replaced by a simpler one in the sense of the computational costs.

The properties of the fractal codebook which are described in section 4.1 lead us to a spiral-shaped search addressed in section 4.2 and to a hierarchical search scheme presented in section 4.3. Replacing d_E by a simpler distance measure is based upon the fact that a good match presupposes range and library block to be of similar grey level distribution. For this purpose in section 4.4 a structural distance measure d_S is proposed which can be derived much easier in terms of computational complexity than the Euclidean distance d_E.

4.1. Codebook properties

Due to the restrictions we made to the transformations w_k they do not operate on the entire image but only on the corresponding image area D_k. This means that self similarity is not exploited in a global but in a local sense only. Figure (2) depicts the probability density of the offset vector $\left(b_x^{(k)}, b_y^{(k)}\right)^T$. It can be noted, that the spatial distance between a domain block from which the library block is taken and the block to be encoded (range block) is typically small, i.e. *near* library blocks more likely provide a good match than *far* ones. It can be seen that the most likely offset vector is $\vec{b} = (0,0)^T$. This effect is exploited by restricting the allowed range for the offset vector $\vec{b}$ or by applying a hierarchical scheme, both described in the following sections.

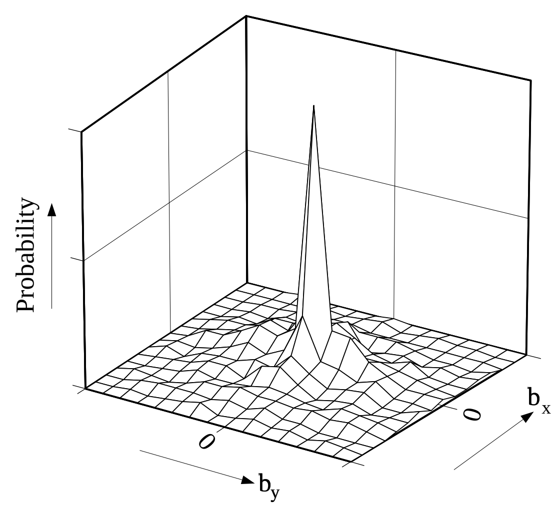

Figure 2: Probability density for offset vector $\vec{b} = \left(b_x^{(k)}, b_y^{(k)}\right)^T$

4.2. Restricted search area

From figure 2 it can be seen, that library blocks which are spatially far away from the considered range block are very unlikely to provide a good match. Therefore the search order is adapted to the probability density function of the codebook offset vector $\left(\boldsymbol{b}_x^{(k)}, \boldsymbol{b}_y^{(k)}\right)^{\boldsymbol{T}}$, so that those codebook candidates which are most likely are tested first. This is performed by a spiral-shaped search order emerging from the codebook entry $(\mathbf{0}, \mathbf{0})^{\boldsymbol{T}}$ of smallest offset vectors to larger ones. The search ends after a predefined number of tests. The library block providing the best match up to this point is taken. As can be

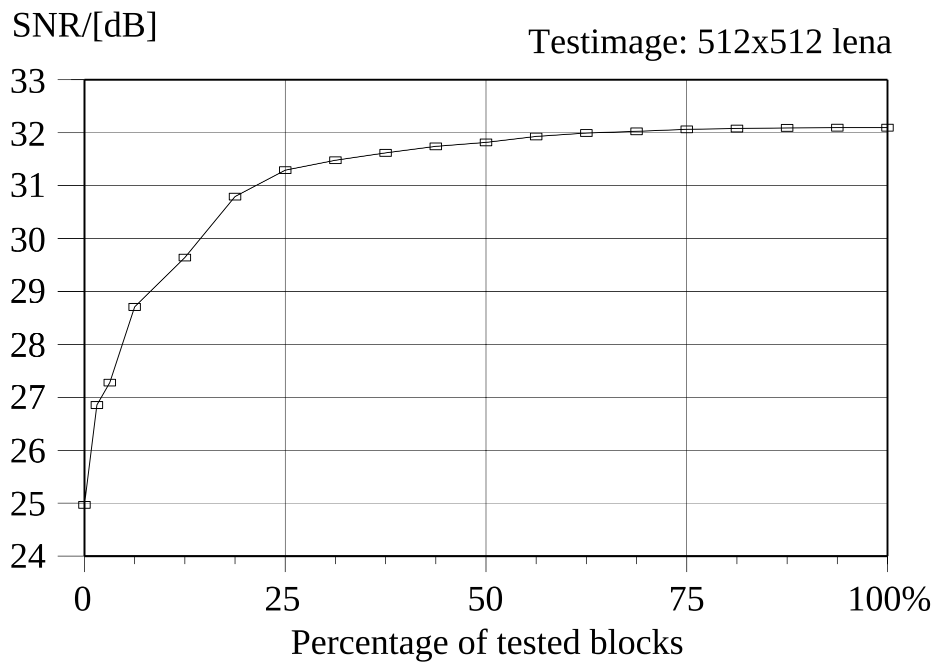

Figure 3: Dependency of Signal to noise ratio (SNR) from number of tested blocks. Full Search corresponds to 100 %.

seen in figure 3 the reconstruction quality only slightly decreases, but encoding speed which is roughly proportional to the number of library tests drastically increases. Another way to break off the search is to provide a minimum distortion level $\boldsymbol{\varepsilon}$ dependent on the desired reconstruction quality which has to be fulfilled for each library block. So the library block $\boldsymbol{L_l}$ is taken as approximation for the considered range block $\boldsymbol{R_k}$ for which

$$\boldsymbol{d_E}\left(\boldsymbol{R_k}, \boldsymbol{s}^{(k)}\boldsymbol{L_l} + \boldsymbol{o}^{(k)}\right) \leq \boldsymbol{\varepsilon} \qquad \textbf{(10)}$$

holds first.

4.3. Hierarchical Search

The second way to decrease the number of necessary block comparisons is to generate a mask which determines those codebook entries, which are considered in the search for an optimal library block. In contrast to the restricted search described above, also some codebook entries which are "far" away from the range block are taken into account. According to the distribution of the offset vectors in the neighborhood of the considered range block a full search starting at codebook entry $(\mathbf{0}, \mathbf{0})^{\boldsymbol{T}}$ is performed.

Since far codebook entries are less likely to provide a good match, not each of them but only a small subset is tested. The set of allowed offset vectors is referred to as search mask exemplarily depicted in figure 4. One can see that the structure of this mask is adapted to the probability density function of the offset vector $\vec{b}$, since in the neighborhood of the codebook entry $(0,0)^T$ many blocks, but for large offset vectors only a few are tested.

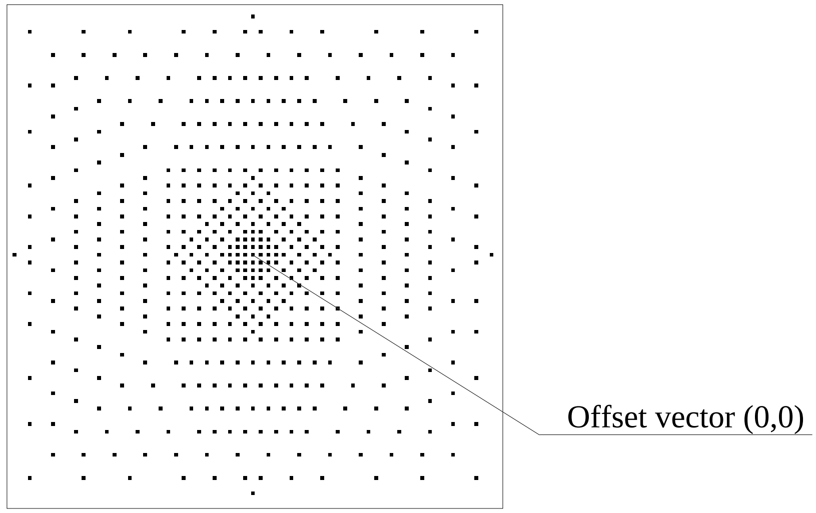

Figure 4: Search mask with 11% occupancy

Because not all codebook entries are considered, the optimal match may not be found. To improve this scheme, a second step is performed by applying a full search in the neighborhood of the codebook entry which was found to be best in the first step. Results as depicted in table 1 show, that the number of necessary block comparisons is significantly reduced but the reconstruction quality is only slightly affected when compared with the full search.

image: 512x512 lena	*percentage of block comparisons*	*SNR*
full search	100%	32.2dB
hierarchical search	12%	31.9dB
restricted search	12%	29.7dB

Table 1 Comparison between hierarchical, restricted, and full search

4.4. Structural classification

Despite of the hierarchical search described in section 4.3, many calculations of the Euclidean distance d_E according to equation (8) have still to be performed in order to find a satisfactory match for every range block. For typical block sizes, e.g. 8×8 pixels, the calculation of each d_E needs about 64 subtractions and multiplications. Encoding speed therefore can be increased not only by reducing the number of distance calculation as with the hierarchical scheme, but also by reducing the computational complexity of the used distortion measure d.

It can be shown that only library blocks of similar spatial grey level distribution as the considered range blocks are likely to provide a good match. In order to exploit this property, a structural classification of all range and library blocks based upon the local mean is proposed. From these a structural distance measure d_S is derived which can be used to determine those blocks for which it is worth to calculate the Euclidean distance d_E. This means, that the Euclidean distance calculation is only performed for those block pairs which have a similar spatial grey level distribution.

Let $B \times B$ be the number of pixels within a square sized image partition R with its upper left corner at position (x_R, y_R). The mean grey level m_R is then defined as

$$m_R = \frac{1}{B^2} \sum_{i=0}^{B-1} \sum_{j=0}^{B-1} R(x_R + i, y_R + j). \tag{11}$$

Let R_0, R_1, R_2, R_3 be the non overlapping square subblocks of the image block R with $R = \cup_{v=0}^{3} R_v$ and m_{R_v} $\forall$ $v \in \{0, 1, 2, 3\}$ be the mean of each subblock. When using the block mean m_R and the four corresponding local mean values m_{R_v}, a *feature vector* $\mathbf{X}_R = (\chi_0, \chi_1, \chi_2, \chi_3)$ can be determined in the following way:

$$\chi_v = \begin{cases} 1, & m_{R_v} > m_R \\ 0, & m_{R_v} \leq m_R \end{cases} \tag{12}$$

The simplest way to derive a distance measure between two blocks R and L based on the spatial grey level distribution is to compare the single components of the associated feature vectors $\mathbf{X}_R$ and $\mathbf{X}_L$. So the distance measure

$$d_S(R, L) = \begin{cases} 0, & \mathbf{X}_R = \mathbf{X}_L \\ \infty, & \mathbf{X}_R \neq \mathbf{X}_L \end{cases} \tag{13}$$

simply results in a decision whether the feature vectors are the same or not.

The encoding process is now modified in the following way: In a first step a feature vector $\mathbf{X}$ for each range and library block is calculated according to equation (12). Then, in order to find the best match for each range block, the hierarchical codebook search is applied as described in section 4.3. The difference is, that the costly calculation of the Euclidean distance $d_E(R, L)$ between a range block R and a library block L is only performed for those block pairs, for which the structural distance $d_S(R, L) = 0$ holds. Since the calculation of the feature vectors for each range and library block only has to be done once at the beginning of the encoding process, it does not increase the computational cost significantly. On the other hand a great number of calculations of $d_E(R, L)$ can be avoided. Encoding speed can be increased by a factor of about five to ten, because the calculation of the structural distance $d_S(R, L)$ requires much less computational effort.

Figure (5) shows the distribution of the feature vectors obtained from a wide variety of test images. The square patterns at the bottom depict the possible spatial grey level distribution of the considered block and are represented by the number $\boldsymbol{u}$. The black area denotes that the mean value of the corresponding subblock is lower compared to the mean value of the entire block.

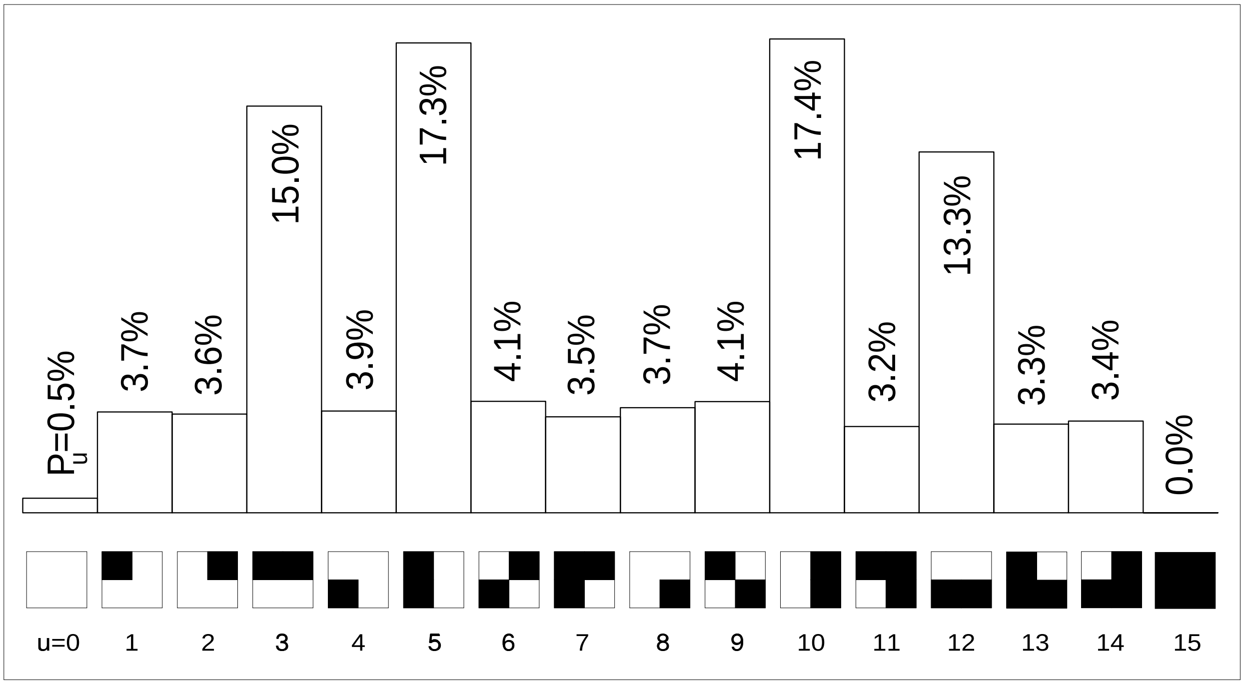

Figure 5: Distribution of feature vectors

Because the Euclidean distance only has to be calculated in the case of zero structural distance $\boldsymbol{d_S}$ between two blocks, the number of remaining necessary calculations of $\boldsymbol{d_E}$ can be estimated in the following way:

Let

$$\boldsymbol{P_u}(\mathrm{X}) = \boldsymbol{Prob}[2^3\chi_3 + 2^2\chi_2 + 2^1\chi_1 + 2^0\chi_0 = \boldsymbol{u}] \tag{14}$$

be the probability that the feature vector $\mathbf{X}$ is of type $\boldsymbol{u}$. The feature vectors of range and library blocks were found to have the same distribution depicted in figure 5. Hence the probability that two blocks $\boldsymbol{R}$ and $\boldsymbol{L}$ have the same feature vector $\mathbf{X} = \mathbf{X}_{\boldsymbol{R}} = \mathbf{X}_{\boldsymbol{L}}$ can be determined by.

$$\boldsymbol{Prob}[\mathrm{X}_{\boldsymbol{R}} = \mathrm{X}_{\boldsymbol{L}}] = \sum_{i=0}^{15} \sum_{j=0}^{i} \boldsymbol{P_i P_j} \tag{15}$$

Coding results obtained from several standard test images are depicted in table 2. The left and middle column show the resulting reconstruction quality in terms of signal to noise ratio (SNR) without and with a structured classification, respectively. The right column depicts the relative number of remaining necessary Euclidean distance calculations when the classification scheme as described above is applied.

It can be seen, that by performing this classification the number of costly distance calculations can be reduced to about 12 % without any significant loss in reconstruction quality.

512x512 test image	SNR no struct. classification	SNR with struct. classification	$\overline{Prob[\mathrm{X}_R = \mathrm{X}_L]}$
lena	32.2dB	32.0dB	11.9%
baltimore	25.2dB	25.5dB	12.5%

Table 2 Reduction of necessary Euclidean distance calculations

4.5. Codebook refinement

The hierarchical search addressed in section 4.3 and the structural block classification described in section 4.4 are both techniques to speed up the computational costly codebook search. Apart from this another point of interest is to increase the reconstruction quality. Obviously a full search in the codebook results in the best quality for the given set of *a priori* defined transformations.

One possibility to improve encoding quality is to reduce the size of range and domain blocks. By using smaller blocks it is more likely to obtain a good match because the distance between the original image and its fractal approximation usually becomes smaller and therefore reconstruction quality increases. Since in some cases, e.g. flat image areas, the approximation with large blocks is sufficient, it does not seem to be meaningful to decrease the block size for the entire image but only for those areas where no good match could be achieved. A very well suited procedure seems to be a quadtree segmentation as described in [11]. It has been shown that reconstruction quality in terms of signal to noise ration (SNR) could be improved by 1–3 dB without any increase in data rate.

Another way to enhance quality is to enlarge the set of allowed transformations, which we refer to as *codebook refinement* in the following. Due to the contractivity constraint, the scaling parameter $s^{(k)}$ (see equation (9)) has to be in the range $s^{(k)} \in (-1.0, \cdots, 1.0)$. The offset parameter $o^{(k)}$ has to be determined in a way that the resulting values representing the grey level are in the range $z(x, y) = [0, \cdots, 255]$ if 8 bpp images are considered. So the only parameter which can be tuned is the range of the allowed offset vector $\left(b_x^{(k)}, b_y^{(k)}\right)^T$. As mentioned in section 3, the single $b_{x,y}$ are only allowed to be multiple of the used block size. We therefore propose to enlarge the used codebook by allowing a finer quantization for the offset vectors in order to increase reconstruction quality. This can be done in the following way:

In a first step the optimal set of transformations W is determined by performing a full search or applying a more sophisticated scheme as described in the previous sections. In a second step the previously determined optimal offset vector is slightly varied. If there exists any vector $\vec{b} + \vec{\delta_b}$ for which the appropriate distance measure becomes significantly smaller, then the vector $\vec{b} + \vec{\delta_b}$ is taken as new offset vector. By this way the set of *a priori* defined transformations is enlarged since the offset vector is no longer constrained to multiples of the used range block size.

The results we obtained show, that reconstruction quality can be increased by 2–3 dB with an additional computational expense of about 5–10 %. Also it should be mentioned that due to the vector $\vec{\delta_k}$ about 4–6 extra bits are needed for representation of one block transformation w_k.

5. SIMULATION RESULTS

Coding simulations were carried out on the original 512x512 8 bits per pixel "lena" image shown in figure 7. The properties of the transformations, especially the distribution of the offset vector which has been derived from a large set of test images are depicted in figure 2. Emerging from these results it has been shown in figure 3 that the reconstruction quality only slightly decreases if the codebook search is restricted to the neighborhood of the considered range block. Some improvements are achieved by introducing an hierarchical search which is based upon a search mask as depicted in figure 4. Simulation results presented in table 1 show that the hierarchical search scheme nearly reaches full search performance with a fraction of its computational effort. Further modifications lead us to a structural classification of the codebook. Since only library blocks having the same spatial grey level distribution are likely to provide a good match a structural classification is introduced and its results are depicted in table 2 for some different test images.

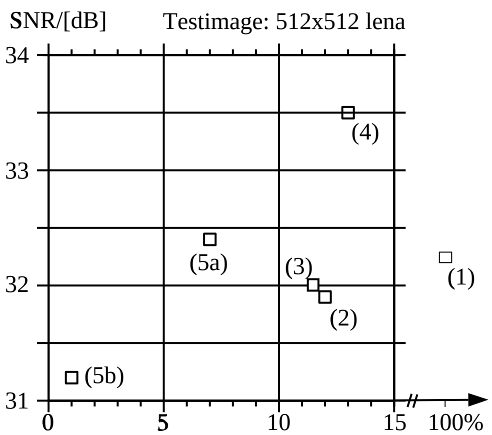

Figure 6: Coding results for some search schemes
1: full search
2: hierarchical search
3: full search with structural classification
4: hierarchical search with codebook refinement
5a: as 4 but with additional structural classification
5b: as 5a but with reduced search mask

Figure 7: Original 512x512 8 bpp "lena" image

A brief summary of our coding results is shown in figure 6. It can be seen that in all cases the computational effort, which is mainly caused by the block tests in order to obtain a satisfactory match, is drastically reduced but the reconstruction quality nearly remains the same. By applying a codebook refinement the reconstruction quality even is beyond the full search boundary but with a fraction of its costs.

6. CONCLUSIONS

Based on thorough studies of the fractal codebook a fast hierarchical search algorithm is described which significantly increases encoding speed while retaining full search performance. In contrast to the latter only a subset of all possible library blocks is tested for a satisfactory match. The selection of this subset containing about five to ten percent of all library blocks is based upon the following properties of the fractal codebook:

- The spatial distance between a library block and the block to be encoded (target block) is typically small, i.e. *near* library blocks more likely provide a good match than *far* ones. Therefore the search order is adapted to the probability density function of the codebook entries using a weighted search mask.
- A good match presupposes target and library blocks to be of similar structure which has been exploited by a codebook classification. On the basis of a local mean analysis a structural distance measure has been developed which partially replaces the costly calculation of the used Euclidean distance.

Emerging from the optimal match the initial codebook has been locally refined and a second search is applied. By this means the reconstruction quality compared to the full search could even be increased with a fraction of its computational effort.

Further investigations are directed to apply the proposed fast hierarchical search algorithm to the field of video coding. Especially in this case it is necessary to perform a fast encoding procedure due to the high encoding rate required.

REFERENCES

[1] CCITT SGVIII Joint Photographics Experts Group (JPEG), ISO/IEC JTC1/SC2/WG8. JPEG Technical Specification, Revision 5, January 1992. JPEG 8–R5.

[2] R. J. Clarke. *Transform coding of images*. Academic Press, London, 1985.

[3] N. M. Nasrabadi and R. A. King. Image coding using vector quantization: A review. *IEEE Transactions on Communication*, COM–36(8):957–971, August 1988.

[4] M. F. Barnsley. *Fractals Everywhere*. Academic Press Inc., London, 1988.

[5] A. E. Jacquin. A novel fractal based block-coding technique for digital images. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing ICASSP'90*, volume 4, pages 2225–2228, 1990.

[6] A. E. Jacquin. Fractal image coding based on a theory of iterated contractive image transformations. In *Proceedings SPIE Visual Communications and Image Processing '90*, volume 1360, pages 227–239, 1990.

[7] A. E. Jacquin. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Transactions on Image Processing*, 1(1):18–30, January 1992.

[8] M. F. Barnsley, J. H. Elton, and D. P. Hardin. Recurrent iterated function systems. *Constructive Approximation*, 5:3–31, 1989.

[9] M. F. Barnsley and A. E. Jacquin. Application of recurrent iterated function systems to images. In *Proceedings SPIE Visual Communications and Image Processing '88*, volume 1001, pages 122–131, 1988.

[10] M. F. Barnsley, V. Ervin, D. Hardin, and J. Lancaster. Solution of an inverse problem for fractals and other sets. In *Proc. Natl. Acad. Sci. USA*, volume 83, pages 1975–1977, April 1986.

[11] B. Hürtgen, F. Müller, and C. Stiller. Adaptive fractal coding of still pictures. In *Proceedings of the International Picture Coding Symposium PCS'93*, page 1.8, Lausanne, Switzerland, 1993.