FRACTAL IMAGE CODING AND MAGNIFICATION USING INVARIANT FEATURES

DETLEF GÖTTING, ACHIM IBENTHAL
Philips Semiconductors, PCALH/VS,
Stresemannallee 101, D-22529 Hamburg
email ibenthal@depcalhp.serigate.philips.nl

ROLF-RAINER GRIGAT
Technical University Hamburg-Harburg, Technische Informatik I
Harburger Schloßstr. 20, D-21079 Hamburg
email grigat@tu-harburg.d400.de

Abstract

Fractal image coding has significant potential for the compression of still and moving images and also for scaling up images. The objective of our investigations was twofold. First, compression ratios of factor 60 and more for still images have been achieved, yielding a better quality of the decoded picture material than standard methods like JPEG. Second, image enlargement up to factors of 16 per dimension has been realized by means of fractal zoom, leading to natural and sharp representation of the scaled image content. Quality improvements were achieved due to the introduction of an extended luminance transform. In order to reduce the computational complexity of the encoding process, a new class of simple and suited invariant features is proposed, facilitating the search in the multidimensional space spanned by image domains and affine transforms.

1. INTRODUCTION

Lossy *image compression* based on the removal of redundant and irrelevant content of the picture information brings about compression factors up to 10. For increasing compression ratios up to factors of 100 and more as required for limited bandwidth transmissions or between the original and the decoded picture are not visually significant for certain textures and structures like hair, trees, clouds and other irregular detail. Fractal image coding performs superior here, since information detail occurring manifold in different scales under slight variations can easily be modelled.

Image enlargement is applied for example to printing posters using resolution limited sources as e.g. TV cameras. Linear magnification factors of 16 and more per spatial dimension are required. Key point is the expansion of image information by a factor of 256 adding subjectively realistic and acute image information, where standard interpolation algorithms lead to a blurred representation.

The coding scheme underlying the investigated algorithms is the fractal block coder[1], where domain blocks are contractively mapped onto range blocks by concatenation of a massic or luminance transform LT and a geometrical transform GT like depicted in Figure 1.

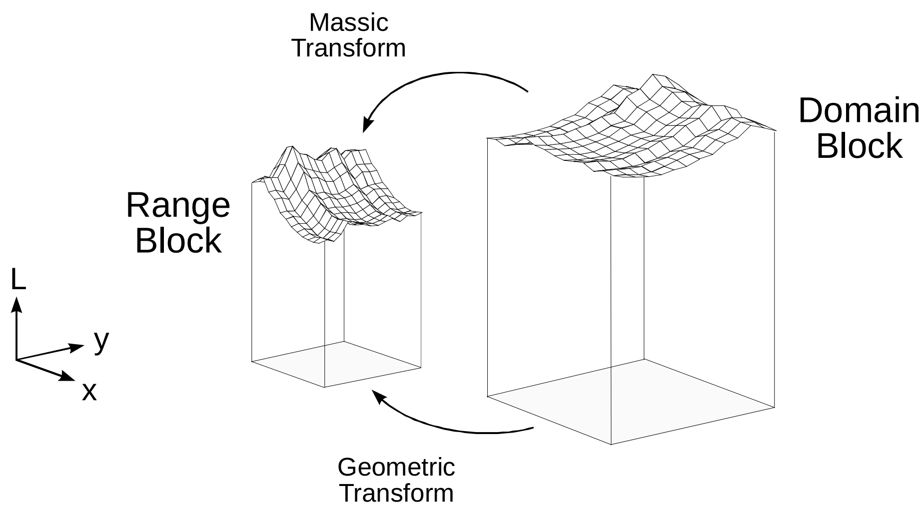

Fig. 1 Principle of fractal block coding.

If the picture is divided in nonintersecting rangeblocks, the compression gained is the amount of information conveyed in the mapping parameters compared to a description by space discrete colour values. Due to the recursive application of the mappings at the decoder, the notion Iterated Function System (IFS) is introduced[2].

Compared to classical DCT based coding schemes, fractal image compression is superior in encoding edges as well as low frequency picture content with lack of perceivable texture. Due to the exploitation of scaling invariance (the domain blocks are scaled down by a factor of 2), the modelling of areas with high frequency detail is difficult. However, such detail often is of irregular nature so that there is no or minor perception of artefacts as long as the decoded picture content is *similar* to the original. This is rather advantageous for image coding. Even if the PSNR values are bad in such areas, impairments are visually concealed, given that there is no well known, recognizable geometrical pattern within the picture, like for example characters or geometrical figures. The quality of fractal image coding with respect to classical DCT based coding schemes is well documented[3].

plexity. In order to find the best encoding parameters there is the task for each range block to find a domain block which gives the best approximation after application of geometric and luminance transform. The search space to be investigated is spanned by the amount of domain block locations, which is equal to the number of pixels of the image, and the variations of all transform parameters. Therefore the major part of our work concentrated on methods to reduce the search effort to a value acceptable for practical implementations. This involves intelligent search methods known from computer science and a definition of simple features invariant against pixel permutations like rotation or mirroring. First of all an extension of the luminance transform is introduced together with a practical implementation strategy. It can be used for example to achieve quality improvements in case of magnification based on IFS.

2. HIGHER ORDER LUMINANCE TRANSFORM

The original coding scheme of A. Jacquin[1] foresees a luminance transform applying luminance scaling of domainblocks and addition of an offset which is equal to adjusting contrast and brightness. Generalized, the massic part of the mapping from domain blocks L_D to range blocks L_R can be expressed as $LT(L_D(x,y)) = s \cdot L_D(x,y) + o$. Finding the best approximation of a range block L_R usually is done by applying least squares error criteria and taking into account the constraint $s < 1$ in order to ensure contractivity of the mapping. However in practice so called eventual contractivity may be sufficient to maintain decoder convergence. Y. Fisher[3] reports experiments where s reached up to 4 and the decoder still converged, although at lower speed. Another phenomenon was observed in these experiments. The best PSNR was found at $s_{max} = 2$, not below 1! One may deduce that the increased parameter space s leads to a better approximation of the collage, which on the decoder side only is constrained by bad fixpoint approximation when s becomes too large. Anyhow, as a conclusion it is worthwhile to increase self similarity between domain and range blocks by extending the luminance transform, especially when the application requires high quality reproduction or even interpolation in case of image magnification. D. Monro[4,5] proposed the Bath Fractal Transform BFT to drastically limit the search area and still be able to model blocks with the help of polynomial functions over the space, helping to increase self similarity. Another way of luminance transformation extension is proposed by G. Vines[3] as a first order luminance transform in conjunction with an orthonormal basis of codebook vectors. In that case the encoding of an image becomes a matter of projection operations, reducing computational effort of domain block search.

In our investigations the luminance transform has been extended by polynomial functions

$$LT(L_D(x,y)) = s_0 \cdot L_D(x,y) + s_1 \cdot x + s_2 \cdot y + s_3 \cdot x \cdot y + s_4 \cdot x^2 + s_5 \cdot y^2 + o \qquad (1)$$

of the spatial coordinates x,y up to second order, also making use of mixed terms. The polynomial part can be regarded as an approximation of the block. The residual part is encoded by IFS means using the whole domain pool in order to avoid a priori quality restrictions. As polynomial and residual part are orthogonal, they can be treated separately.

A fast algorithm for the determination of the luminance transform has been developed by calculating the polynomial coefficients s_0, o, s_1, $s_2, \ldots$ only once for each range and each domain block. The N_R pixels of range blocks L_R and shrinked domain blocks L_D

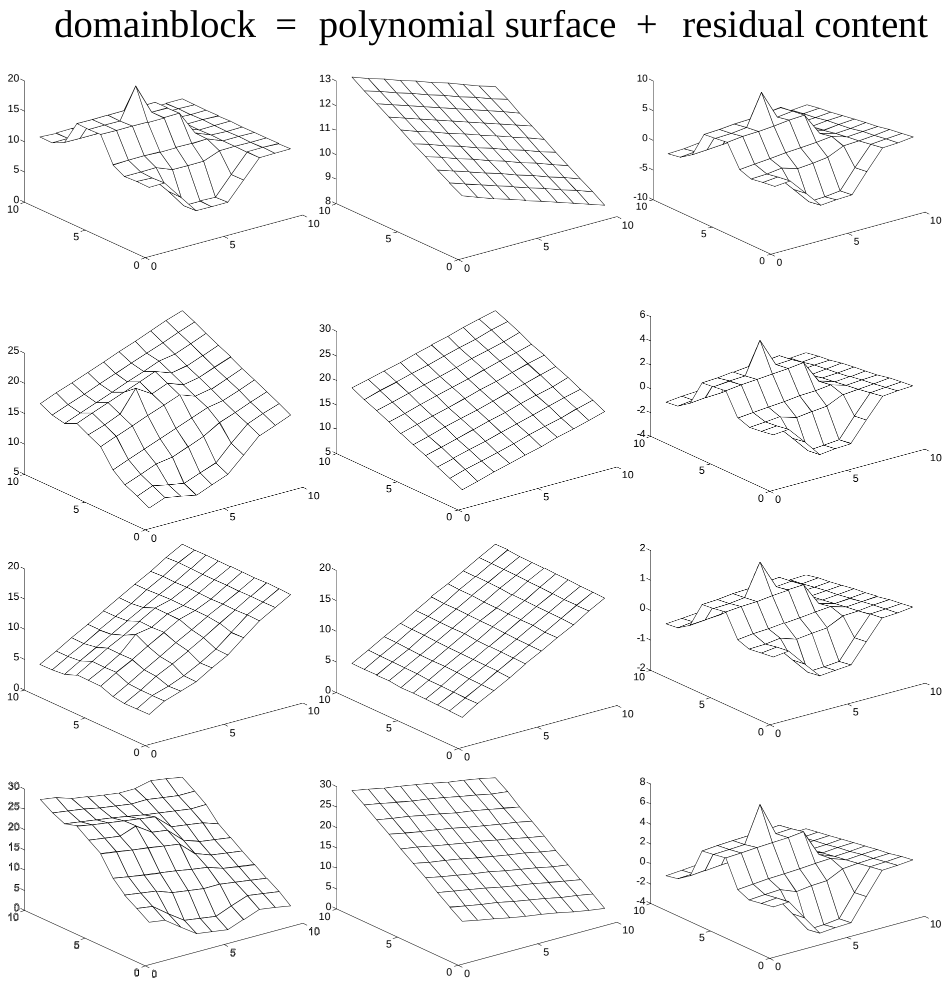

Fig. 2 Example for increasing self similarity by a first order luminance transform.

are serialized and written as onedimensional vectors. Any vector L_B is represented as $L_B = \tilde{L}_B + M \cdot P_B$, composed of the polynomial coefficient vector P_B, the matrix M consisting of the spatial variables $x, y, x \cdot y, x^2, y^2, \cdots$ and the residual signal vector $\tilde{L}_B$. The mapping from domain blocks L_D onto range blocks L_R is expressed by the transformation $L_R = s_0 \cdot L_D + M \cdot P$ leading to

$$\tilde{L}_R + M \cdot P_R = s_0 \cdot [\tilde{L}_D + M \cdot P_D] + M \cdot P \quad . \tag{2}$$

After separation of the polynomial terms

$$P = P_R - s_0 \cdot P_D, \text{ where } P_R = [(M^T \cdot M)^{-1} \cdot M^T] \cdot L_R \text{ and } P_D = [(M^T \cdot M)^{-1} \cdot M^T] \cdot L_D \tag{3}$$

the scale factor s_0 is calculated from the pseudoinverse of the $N_R^2 \times 1$-matrix $\tilde{L}_D$ as

$$s_0 = [(\tilde{L}_D^T \cdot \tilde{L}_D)^{-1} \cdot \tilde{L}_D^T] \cdot \tilde{L}_R = \frac{\sum_{x=0}^{N_R-1} \sum_{y=0}^{N_R-1} \tilde{L}_{Dxy} \cdot \tilde{L}_{Rxy}}{\sum_{x=0}^{N_R-1} \sum_{y=0}^{N_R-1} \tilde{L}_{Dxy}^2} \quad . \tag{4}$$

is illustrated in Figure 2, where four differently looking domainblocks turn out to have the same appearance after subtracting the first order polynomial part.

Obviously there are also two drawbacks of this scheme. The computational complexity and the number of IFS parameters is increased and therefore the compression ratio is decreased. This is acceptable in certain applications, especially in the case of fractal zoom, where blockiness is one major impairment. For standard encoding/decoding applications there are other benefits. Using a quad tree approach for the hierarchical coding of range blocks[3], it is possible to increase the number of large blocks. This is reasonable as long as the amount of additional parameters s_i is smaller than the entire mapping parameters needed for the subblocks. Another benefit is given for small picture sizes like for example the QCIF format used for very low bitrate coding. Changed image statistics and a reduced domain pool are imposing restrictions on the applicability of IFS based coding methods. It is important here to use the domain pool most effectively and therefore to increase self similarity by the extended luminance transform. A matter of future work in this field will be the investigation of other luminance transforms paying a better tribute to the image statistics of such pictures together with a good performance of the zooming function, which is of special interest here.

3. FAST SEARCH USING INVARIANT FEATURE KEYS

The reduction of the enormous encoding effort of fractal encoders gave rise to lots of investigations. Starting from A. Jacquin[1], who used three simple block classifications, shade blocks without a significant gradient, edge blocks featuring a significant gradient across a curve and midrange blocks containing moderate gradients like textures. Y. Fisher[3] uses mean values and variances of the four domain/rangeblock quadrants to derive 72 classes. The idea behind classification is to characterize a range block a priori and search for proper domain blocks only within the subset of the domain pool belonging to the respective class. A comprehensive overview of complexity reduction algorithms is given by A. Saupe and R. Hamzaoui[6], characterizing known methods as being based on discrete (classification, adaptive clustering) and continuous features (1 D functional vectors, feature vectors). The complexity of such search reduction methods equals $\alpha \cdot \mathrm{O}(N_{pel})$, where the efficiency of the particular method decreases α, but still stays linear to the amount of pixels N_{pel} in the image, which corresponds to the amount of domain blocks.

It has already been mentioned, that the range and domainblocks L_R and L_D are serialized for practical issues, forming vectors of N_R and N_D elements. A simple and direct approach is to regard the domain pool as a file of N_{pel} records, each of which contains N_R keys. Following the work of J. L. Bentley et. al.[8,9,10,11], the application of multidimensional search trees reduces the linear effort to expected logarithmic order $\mathrm{O}(\log(N_{pel}))$. Different search space partitionings are known (Figure 3), all leading to logarithmic mean search effort.

D. Saupe[7] mentions the huge memory requirements of $k-d$ trees for domain block search applications. As keys he uses the normalized part of a block L which is orthogonal to the normalized vector representing the DC component $[1\,1\,\cdots\,1]^T/\sqrt{N_R}$. To reduce the memory requirements, D. Saupe has investigated $k-d$ trees in conjunction with Fisher's classification scheme and downsamples the keys to a number of 4 or 16 respectively. Instead of inserting each domain under application of the 8 isometries[1] (identity map, reflection,

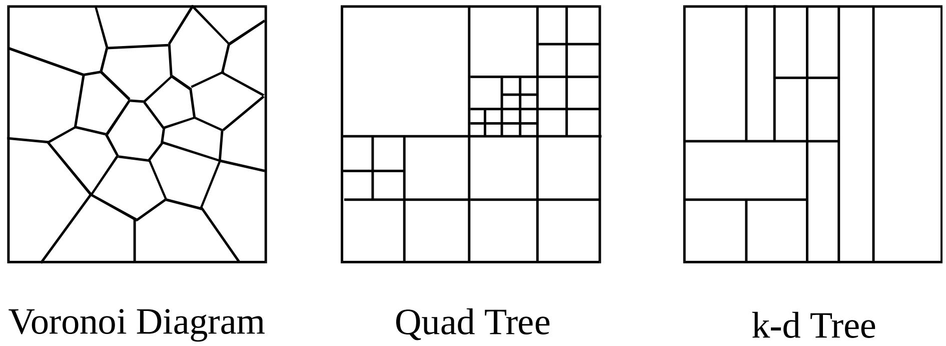

Fig. 3 Search space partitionings.

rotation), he searches each range block 8 times in the different variations. To reduce search time, not only the nearest but the $1+\varepsilon$ nearest neighbour is searched for.

The key point of reducing the search order from order $\alpha \cdot \mathrm{O}(N_{pel})$ to expected $\alpha \cdot \mathrm{O}(\log(N_{pel}))$ is an important step. The notification *expected* is reflecting to the organization of the tree. Constructing it from scratch with no constraints would possibly lead to a degenerated tree, where the mean logarithmic search time could not be achieved. The structure will largely depend on the statistics of the blocks found in an image, thus it is image dependent. So in our algorithm we fixed the tree at the first levels by predeterming keys and their respective division thresholds. A further possibility is to prohibit the division of very small subspaces and either insert new domain entries as a list associated to the respective node or discard them totally and only insert a representative block in the node.

Another important step for the reduction of search effort is the efficient treatment of the isometries, which is proposed in the following. In Figure 4 the basic idea is sketched for a 4×4 pixel block:

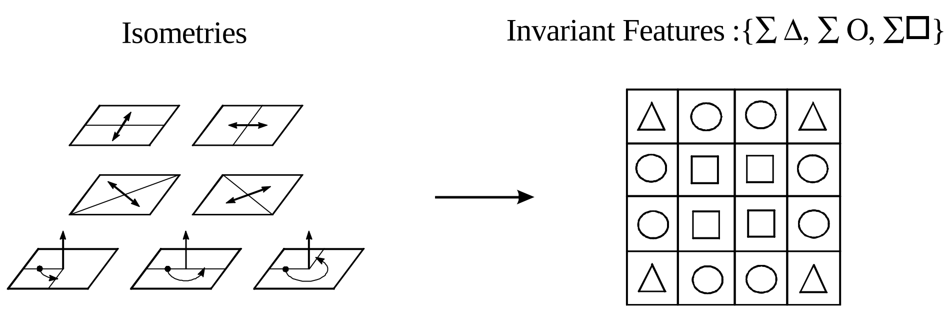

Fig. 4 Isometry invariant features.

All pixel locations of a block, which are mapped onto themselves, are accumulated. For a 4×4 block three features can be extracted that may be used as keys in a $k-d$ tree. Searching under all isometry variations is not necessary any more. A nice side effect is the reduction of keys from 16 down to 3 giving rise to additional speedup as well as reduction

For a practical implementation only the residual luminance components are considered, the polynomial parts of the blocks can be treated separately without putting constraints on tree construction and search. The residual parts of the domain blocks are normalized to $\tilde{L}_{DN} = \tilde{L}_D / \|\tilde{L}_D\|_2$ in order to remove the dynamics in scaling. The invariant keys then become:

$$K_1 = \sum_{\triangle} \tilde{L}_{DN}, \quad K_2 = \sum_{\bigcirc} \tilde{L}_{DN}, \quad K_3 = \sum_{\square} \tilde{L}_{DN} \tag{5}$$

A simplified example for the search is given in Figure 5, where for reasons of a better illustration only two keys are used and a tree has been constructed out of 6 domain blocks. The keys are supposed to have values between 0 and 6. The active key belonging to a node is marked by an arrow, the discriminator threshold p is given below the node. Now a range block R is to be searched, which has key values $K_1 = 4$ and $K_2 = 1$. Starting at the root D_1, the search will develop as:

1. discriminator: K_1 $\Rightarrow$ $K_1(R) = 4 \; > \; p(D_1) = 3$ $\Rightarrow$ search right
2. discriminator: K_2 $\Rightarrow$ $K_2(R) = 1 \; < \; p(D_3) = 3$ $\Rightarrow$ search left
3. discriminator: K_1 $\Rightarrow$ $K_1(R) = 4 \; < \; p(D_5) = 4.5$ $\Rightarrow$ search left
4. no successor $\Rightarrow$ nearest neighbour of R is D_5.

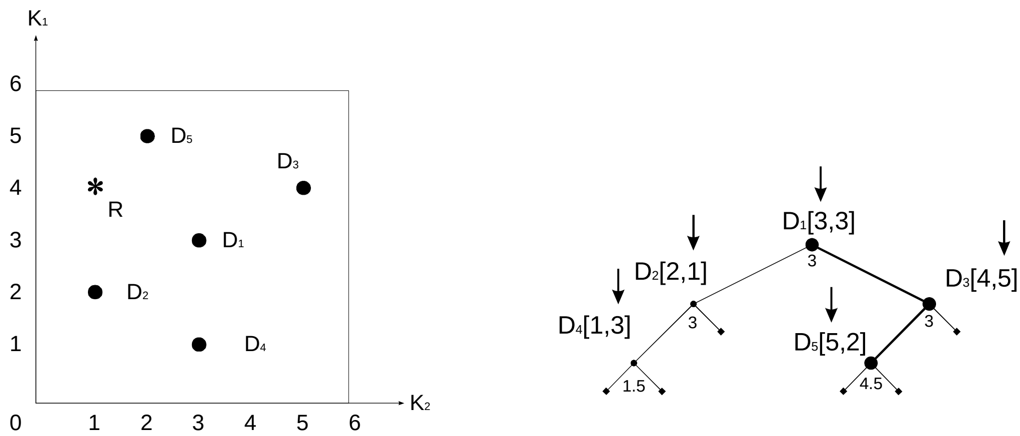

Fig. 5 Example for twodimensional search space and 6 domainblocks.

The computational cheap generation of the invariants is paid by a lack of completeness[12], meaning that there are ambiguities in the discrimination of possible blocks in the feature space. A derivation of more complete and general invariants is given by Nölle et al.[13]. The invariants proposed above are largely simplified versions of them. So arriving at D_5, it could happen that a bad match of $\tilde{L}_{RN}$ by $\tilde{L}_{D_5N}$ is discovered, although the nearest neighbour in the feature space has correctly been found. One possibility to reduce the ambiguity problem is to add other features, for example the variance as Y. Fisher did for his classification scheme. This would enlarge the search tree and thus is not favourable. Another possibility, which we investigated, is monitoring the matching error during the search of a range block and "climb up" the tree if ambiguities would lead into a bad branch. At the same time a threshold check is done if the matching error is for example better than 35 dB, so that the time spent on upclimbing the tree is compensated in other cases. In total the simulation time can be brought down to about 10 seconds for a 512 by 512 Lenna image ratio of 35:1.

Fig. 6 Lenna encoded fractally (left) and accoding to JPEG (right) at compression ratio 35:1.

It should be mentioned that the storage of block information in internal nodes does not follow the common $k-d$ tree scheme, where data are stored in external nodes. For the application of search trees to invariants however it is more practical and does not increase the requirements of memory capacity. In case of larger blocks it is possible to use the same approach by increasing the number of keys. If this is not wanted, pixel weighting and subsampling may be applied to come down to 4×4 blocks again.

4. Fractal magnification

During decoding any encoded IFS may be applied to arbitrary resolutions, since it only conveys the mapping of $N_D \times N_D$ sized domain blocks to $N_R \times N_R$ sized range blocks where $N_D = aN_R$. While a is fixed (typically $a = 2$), in the decoder N_R can be determined without respect to the encoded block format as the IFS is continuous in space. For magnification N_R has to be increased while for an image shrink it has to be decreased. Compared to classical interpolation methods this technique is therefore resolution independent.

For quality issues the above described extended luminance transform has been developed and applied. Like the affine mappings the polynomial approximation surface can be calculated for any block size. Supposed an original range block size N_R is magnified to $N'_R = mN_R$, the polynomial surface is given by the values $p(x/m, y/m)\ \forall x, y \in \{0, 1, \ldots, mN_R - 1\}$. This polynomial surface has to be added to the mapped domain blocks of size $N_D = amN_R = aN'_R$.

Assessments were done by visual evaluation, since PSNR measures are not applicable. Our results have shown, that the quality of magnified pictures is proportional to the polynomial order (up to order 2 has been investigated) of the approximation surface and inverse proportional to the quantization of the polynomial parameters and of N_R with the restricapplication high compression rates are of minor priority, except for very low bitrate coding, where there also is a strong interest of zooming the small QCIF pictures.

For magnification factors higher than 8 per spatial dimension, postfiltering has to be applied for the suppression of blocking artifacts, which visually appear as a grid overlay. In combination with postprocessing methods, predominating benefits of fractal zoom are acute preservation of edges without serration effects and increased brilliance compared with interpolation blur, cf. Figures 7 and 8 featuring Lenna's right eye. In the fractal zoom example no postfiltering has been carried out to demonstrate the blocking effects. By Polidori and Dugelay[14] another trick to cope with the block artefacts has been shown. Overlapping ranges were used and weighted to suppress the effects.

Fig. 7 Fractal magnification 16×16 of Lenna's right eye.

Fig. 8 Convential nonlinear interpolation 16×16 of Lenna's right eye.

5. CONCLUSIONS

The purpose of the extended luminance transform is the increase of self similarity. It can be exploited to improve the picture quality whenever this is required. Examples are fractal zoom, applications with medium compression and high quality or small image formats with rather restricted domain pool. Although more parameters need to be encoded, the extended luminance transform makes sense in quadtree approaches where the amount of large blocks can be increased. In practical implementations a speedup of the encoding process has been achieved by applying modified $k-d$ trees in combination with a simple class of invariant features that are optimal for the isometries commonly used in fractal coding. Computational complexity has been further reduced by a split between parameter calculation of the polynomials and IFS encoding of the residual luminance signal. The encoding time on a HP715 workstation for video coding of QCIF sequences is reduced to one second per frame. Another potential application due to the split is scalable encoding/decoding: the polynomials can be used for a fast low resolution representation of the image while the residual part can be decoded separately and added when available. In case of transmission, the channel encoding part may also take advantage by allowing a higher bit error rate for the fractal part than for the polynomial part. In case of error prone environments first the fractal part

6. REFERENCES

1. A. E. Jacquin, *A Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding*, (PhD Thesis, Georgia Institute of Technology, 1989).
2. M. F. Barnsley and L. P. Hurd, *Fractal Image Compression*, (Academic Press Professional, 1992).
3. ed. Y. Fisher, *Fractal Image Compression*, (Springer, 1994).
4. D. M. Monro, *Proc. ICASSP* (1993), **V** pp. 169-172.
5. D. Monro and S. J. Woolley, *Proc. ICASSP* (1994), **V** pp. 557-560.
6. D. Saupe and R. Hamzaoui, *I.M.A. Conf. Proc. on Image Processing; Mathematical Methods and Applications*, (Sep. 1994).
7. D. Saupe, *Proceedings DCC'95 Data Compression Conference*, (IEEE Computer Society Press, March 1995).
8. J.L. Bentley, *Communications of the ACM*, **18**, (No. 9, 1975), pp 509-517.
9. J. H. Friedman, J. L. Bentley and R.A. Finkel, *ACM Transactions on Mathematical Software*, **3**, (No. 3, 1977), pp. 209-226.
10. J. L. Bentley, *IEEE Trans. on Software Engineering*, **SE-5**, (No. 4, July 1979), pp. 333-340.
11. J. L. Bentley, B. W. Weide and A. C. Yao, *ACM Trans. on Mathematical Software*, **6**, (No. 4, Dec. 1980), pp. 563-580.
12. H. Burkhardt, A. Fenske, H. Schulz-Mirbach, *tm - Technisches Messen*, **59** (1992, 10), pp. 398-407.
13. M. Nölle, G. Schreiber, H. Schulz Mirbach, *Internal Report*, (Technische Universität Hamburg-Harburg, Technische Informatik I, 1994).
14. E. Polidori, J.-G. Dugelay, *poster session on the Fractal Image Encoding and Analysis Advance Study Institute*, (Trondheim, Norway, July 8-17, 1995).