Enhancing Fractal Image Compression with Vector Quantization

Raouf Hamzaoui *Martin Müller* *Dietmar Saupe*
Universität Freiburg, Institut für Informatik, Am Flughafen 17, 79110 Freiburg, Germany

ABSTRACT

A novel hybrid scheme combining fractal image compression with mean-removed shape-gain vector quantization is presented. The algorithm, based on a distance classification fractal coder with fixed cluster centers, decides whether to encode a range block by a cluster center or by a domain block. Our scheme is shown to improve the performance of conventional fractal coding in all its aspects. The rate-distortion curve is ameliorated, and both the encoding and the decoding are faster.

1. INTRODUCTION AND PREVIOUS WORK

Fractal image compression is a promising new technique introduced by the works of Barnsley [1] and Jacquin [2]. A review of the topic is given in [3]. Several extensions have been proposed to improve the performance of Jacquin's original algorithm. Recently some impressive results have been obtained by applying hybrid methods which use traditional block coding techniques [4]. Though evoked by some authors, combining fractal coding with vector quantization (VQ) has not been deeply investigated. In [3] it is only suggested that fractal coding should be employed for sharp-edge blocks, whereas vector quantization would be more advantageous for the other blocks. Gharavi-Alkhansari and Huang [5] point out that vector quantization can be seen as a special case of their generalized transform. However, no emphasis is given to the VQ aspect of their coder. An interesting study was presented in [6] where the performance of a fractal image coder and a product code vector quantizer have been compared.

This paper investigates how to optimally take advantage of a vector quantization codebook in order to enhance the performance of a fractal image coder.

In a previous article [7] we implemented a simple classification scheme that allowed to attain almost full search fidelity by considering only a small number of domain blocks from the whole domain pool. Classes were represented by cluster centers designed either adaptively from the test image or from a set of training images. Clusters were formed by grouping feature vectors of domains and ranges around their corresponding nearest neighbors in the set of cluster centers. The encoding consisted of searching matches inside the same cluster or in the neighboring ones. In this paper we propose to take profit of the precomputed cluster centers which can be considered as an integral part not only of the encoder but also of the decoder. If the least squares approximation of a range block by an affine transformation of the nearest cluster center is "good enough", then the cluster center will serve as a VQ codebook block. Otherwise, the range block will be encoded by a domain block. In this way, the bit rates can be improved by a clever choice of the ratio of the number of cluster centers to the number of domain blocks used in the fractal code. For example, if we denote by N_R the number of range blocks and by N_1 the number of range blocks VQ encoded, then the hybrid scheme will improve the rate of the fractal coder if $N_1 > \frac{N_R}{p-k}$, where 2^p is the number of domain blocks and 2^k is the number of cluster centers. In the above computation, one bit per range has been included to specify the way a range block has been encoded.

2. NOTATIONS AND MATHEMATICAL BACKGROUND

Let us assume that a sampled image is partitioned into nonoverlapping square blocks of size $N \times N$ called range blocks. We consider each range block as a vector R in the linear vector space $\mathbf{R}^n$ where $n = N \times N$.

The domain pool is a collection of square blocks which are typically larger than the ranges and taken also from the image, called domain blocks. The domain pool may be enlarged by including blocks obtained after applying the eight isometries of the square to the domain blocks. By pixel averaging, the size of these blocks is reduced to the size of a range block. The resulting blocks are called codebook blocks. In the encoding process for a range block $R \in \mathbf{R}^n$ a search through the codebook blocks $D_1, \ldots, D_{N_D} \in \mathbf{R}^n$ is required. We let $E(D_i, R)$ denote the least squares error of an approximation of the range block R by an affine transformation of the codebook block D_i, i.e., $E(D_i, R) = \min_{a,b \in \mathbf{R}} \|R - (aD_i + bC)\|$, where C is the block of constant intensity $C = (1, \ldots, 1)$. The codebook block D_i which gives the smallest error $E(D_i, R)$ is selected on condition that the value of the scaling factor a for the codebook block D_i ensures the convergence of the decoding process (e.g., by requiring $|a| < 1$). Now let O be the orthogonal projection operator which projects $\mathbf{R}^n$ onto the orthogonal complement $\mathcal{C}^\perp$, where $\mathcal{C}$ is the linear span of C. Keeping the same notations we have the theorem [8], which sets the mathematical basis for our clustering algorithm.

Theorem 1 *Let $n \geq 2$ and $X = \mathbf{R}^n \backslash \mathcal{C}$. Define the function $\Delta : X \times X \to [0, \sqrt{2}]$ by $\Delta(D, R) = \min(||\phi(R) + \phi(D)||, ||\phi(R) - \phi(D)||)$, where $\phi(Z) = \frac{OZ}{||OZ||}$. For $D, R \in X$ the least squares error $E(D, R)$ is given by $E(D, R) = \langle R, \phi(R)\rangle\, g(\Delta(D, R))$ where $g(\Delta) = \Delta\sqrt{1 - \frac{\Delta^2}{4}}$.*

It follows from the theorem that we may replace the computation and minimization of N_D least squares errors $E(D_i, R)$ by the search for the nearest neighbor of $\phi(R) \in \mathbf{R}^n$ in the set of $2N_D$ vectors $\pm\phi(D_i) \in \mathbf{R}^n$.

3. THE ALGORITHM

The block average intensity classification [9] is a smart way to divide square blocks into three main classes. In the following we describe this classification, since it will be used in our algorithm. For every block $B \in \mathbf{R}^n$ there exists a unique isometry $I_B : \mathbf{R}^n \to \mathbf{R}^n$ (corresponding, when block B is seen as a 2D-array, to one of the 8 isometries of the square) that transforms B such that the average pixel intensities $B_i,\ i = 1, 2, 3, 4$ of the four quadrants (upper left, upper right, lower left and lower right) of its 2D-array representation are ordered in one of the three canonical positions

Major class 1: $B_1 \geq B_2 \geq B_3 \geq B_4$,
Major class 2: $B_1 \geq B_2 \geq B_4 \geq B_3$,
Major class 3: $B_1 \geq B_4 \geq B_2 \geq B_3$.

Using the procedure described in [7], a set of fixed cluster centers was designed from several training images. It can be proved that these cluster centers belong to the space $\mathcal{C}^\perp$, i.e., they have zero mean. These cluster centers are then normalized and denoted by $\{m_1, \ldots, m_{N_m}\}$. The clustering of the image codebook consisted of mapping each feature vector $\phi(I_{D_i}(D_i))$ to its nearest normalized cluster center. To encode a range block R and if we wanted to obtain only positive scaling factors we considered codebook blocks whose feature vectors lied in the cluster whose center was the nearest neighbor of $\phi(I_R(R))$. In our study [7] we demonstrated the superiority of this approach as compared to the previous state-of-the-art classification method of [9]. We now introduce our VQ hybrid scheme which provides a further enhancement. Let $c = \arg\min_i ||\phi(I_R(R)) - m_i||$. If

$$\frac{1}{\sqrt{n}} E(R, I_R^{-1}(m_c)) \leq \delta \qquad (1)$$

or if

$$E(R, I_R^{-1}(m_c)) \leq (1 + \epsilon) E(R, I_R^{-1}(I_{D_i}(D_i))) \qquad (2)$$

holds for all codebook blocks D_i in the cluster with center m_c, then the cluster center m_c is retained for VQ encoding of the range block. Here ϵ and δ are parameters of our method. Condition (1) states that R is well approximated by m_c while condition (2) says that the collage error due to m_c is not much worse than the collage error due to D_i. If neither condition (1) nor condition (2) are satisfied, then the range block will be encoded by the codebook block D minimizing $E(R, I_R^{-1}(I_{D_i}(D_i)))$ in the cluster with center m_c. The code consists of the index c (or the address of D), and the corresponding scaling factor a, offset b, and isometry I_R^{-1}.[1] Note that Theorem 1 ensures that m_c is the cluster center that can best approximate the range block in the least squares sense.[2] Note also that the new scheme reduces the complexity of the already fast algorithm described in [7] since the search for a matching codebook block is only started if the cluster center was not able to provide an acceptable approximation. As explained in [7], the search for a matching codebook block can be extended to neighboring clusters. Also to include negative scaling factors the nearest neighbor of $\phi(I_{-R}(-R))$ must be considered.

We make now some remarks:

- For VQ encoded range blocks no contractivity condition on the scaling factor is required.
- The offset of a VQ encoded range block reduces to the mean of the block.

The decoding proceeds as with a conventional fractal decoder, i.e., through iterations from any initial image with the advantage, however, that the reconstruction of the VQ encoded range regions is already achieved after the first iteration. Thus, in addition to a less complex decoder, a faster convergence is expected.

4. RESULTS AND DISCUSSION

With Kohonen's program package [10] we designed a set of 256 fixed cluster centers corresponding to the nodes of a 16×16 rectangular array. The training sequence was generated from 9 images of size 512×512. We compared the encoding of several 512×512 images that were not used in the training sequence by our distance classification based fractal coder [7] and by the hybrid scheme. For the two schemes both positive and negative scaling factors were considered (option Both in [7]). A fixed range size (4×4) was initially selected to better analyze the performance of the new scheme. Thus we had 16384 nonoverlapping range blocks

[1] In case the scaling factor is zero, the address of D (respectively the index of the cluster center) and the isometry I_R^{-1} are not stored.

[2] Of course quantization effects may make us choose a suboptimal cluster center.

ϵ	δ	N_1	Comp	PSNR	Time
0.1	2	6430	5.06	37.38	56
	3	13149	5.89	36.99	39
0.15	2	7501	5.11	37.30	56
	3	13797	5.95	36.92	39
0.2	2	8539	5.16	37.21	56
	3	14384	6.01	36.86	39
0.25	2	9502	5.22	37.11	56
	3	14883	6.06	36.78	39

Table 1. Performance of the hybrid scheme for the 512 × 512 Lenna image for different settings of the tolerance bounds ϵ and δ. Here N_1 is the number of VQ encoded range blocks. The fourth column indicates the compression ratio. The extra bit per range specifying the encoding way (VQ or fractal) is taken into account. Gzip coding of the corresponding bit plane is used. The last two columns show the PSNR in dB and encoding time in seconds. In comparison, the distance based fractal coder provided a PSNR of 36.92 dB, necessitated 58 seconds for the encoding, and had a compression ratio of 4.74.

Iteration	Hybrid	Fractal
1	25.18	11.29
3	36.59	20.35
5	36.92	36.77
8	-	36.92

Table 2. Convergence of the decoding for the two methods with $\epsilon = 0.15$ and $\delta = 3$ for the 512 × 512 Lenna image. The last two columns give the PSNR in dB for the hybrid scheme and the distance based fractal scheme. The last two rows correspond to the iteration step at which the convergence of the hybrid scheme and the pure fractal scheme occurred.

and 4096 nonoverlapping domain blocks having twice the range size. We opted for a 4-class search (for both methods) since it provides an acceptable trade-off between speed and fidelity. We spent respectively 5 and 7 bits for the scaling factor and the offset for both schemes. All coefficients were uniformly quantized. The maximum value of the scaling factor was set to 1 for the fractal coder.

Table 1 shows the results obtained when varying the tolerance bounds ϵ and δ for the 512×512 Lenna image. The PSNR was computed without postprocessing. All the times reported are measured in seconds on an SGI Indigo2 running an MIPS R4400 150 MHz processor. Note that the number of bits saved was not exactly proportional to the number of range blocks which were VQ encoded. This is due to the fact that the address of the cluster center was not stored if the scaling factor was zero.

Figure 1 shows the bit plane indicating the way each range block was encoded. This bit plane corresponds to the sequence of 16384 bits that must be transmitted to the decoder accordingly.

We have also noticed that the convergence of the decoding was significantly faster for the hybrid scheme (see Table 2).

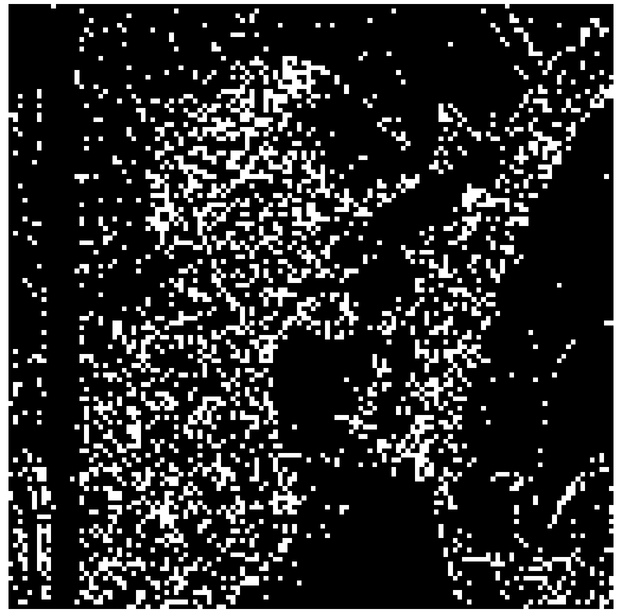

Figure 1. Bit plane for the 512 × 512 Lenna image with $\epsilon = 0.15$ and $\delta = 3$. VQ encoded range blocks are shown in black while fractal encoded range blocks are shown in white.

The results show that for a large range of the tolerance bounds ϵ and δ, all aspects of the fractal coding are improved: the compression ratio is higher, the PSNR is better, and both the encoding and the decoding are faster.

We point out that encoding all range blocks with only cluster centers provided a PSNR of 36.44 dB. Thus it seems that the domain pool was of vital importance for an accurate encoding of some parts of the image.

Our method was also successfully implemented with a two-level quadtree scheme using an additional set of 256 VQ cluster centers of size 8 × 8 generated in a similar way as explained above. The core of the quadtree algorithm for the hybrid scheme can be described as follows. For a given range block $R \in \mathbf{R}^n$ and root mean square tolerance levels δ and t, if

$$\frac{1}{\sqrt{n}} E(R, I_R^{-1}(m_c)) \leq \delta$$

then range block R was encoded with cluster center m_c. Otherwise, letting D denote the best candidate codebook block in all the classes searched, if

$$\frac{1}{\sqrt{n}} E(R, I_R^{-1}(I_D(D))) > t \tag{3}$$

then the range block was not accepted and partitioned into 4 smaller blocks. In the other case, if

$$E(R, I_R^{-1}(m_c)) \leq (1+\epsilon) E(R, I_R^{-1}(I_D(D)))$$

then range block R was encoded with cluster center m_c and otherwise it was encoded with codebook block D.

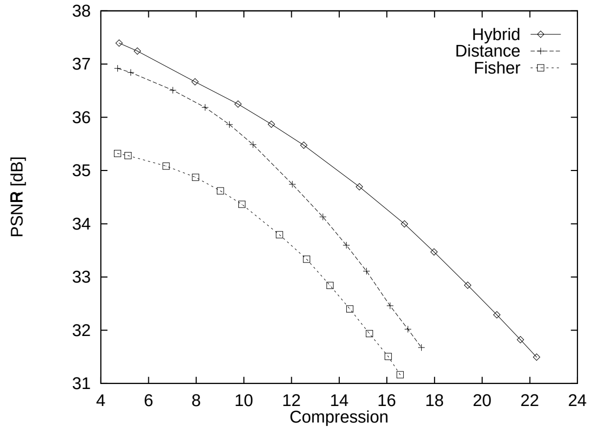

Figure 2. PSNR vs. compression ratio for the 512×512 Lenna image.

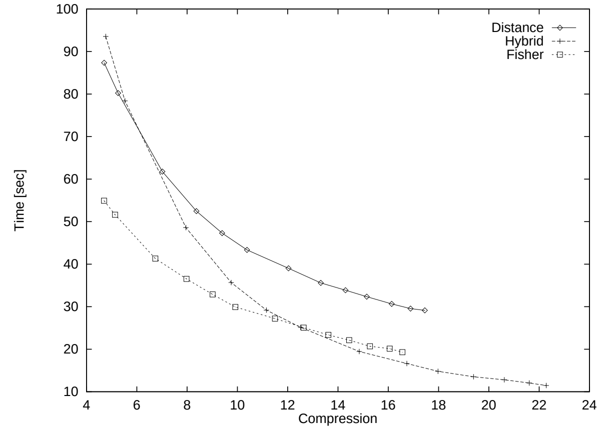

Figure 3. Time vs. compression ratio for the 512×512 Lenna image.

Of course at the lowest level of the quadtree test (3) is not needed. As in [7], the dimension of the 8×8 blocks was reduced by pixel averaging. Thus, classification was carried out at the lowest level of the quadtree, i.e., for the 16-dimensional feature vectors but computations of the least squares errors were done for blocks at the real size. Figure 2 compares the rate-distortion performance of Fisher's scheme (one class from a total of 72 classes was searched) with the distance based scheme [7] and the hybrid scheme (four classes out of 256 were searched). The compression ratio was varied by letting the tolerance level t take the values 20, 18, 16, 14, 12, 10, 8, 6, 5, 4, 3, 2 and 1. For the hybrid scheme, the parameters δ and ϵ were set to t and 0.15 respectively. For the 8×8 ranges, the domain pool consisted of 3969 overlapping domains having twice the range size. Figure 3 shows the time as a function of compression ratio for the same series of tests. Clearly the hybrid scheme had the better rate-distortion performance with a gain of more than 2 dB over Fisher's scheme. Note that a three-class search in Fisher's scheme would give only a small improvement in PSNR but will triple the encoding times. We conclude by mentioning that with the option Best (see [7]), our two distance based schemes can be made significantly faster at the expense of a small loss in image quality.

5. CONCLUSION

We have introduced an improvement of fractal coding by making use of a set of VQ codebook blocks which served as a block classifier for the image domain pool and as an alternative means of coding, when they were able to provide a satisfying distortion. Experimental results show that our hybrid scheme yields superior performance over conventional fractal coding.

REFERENCES

[1] Barnsley, M., Hurd, L., *Fractal Image Compression,* AK Peters, Wellesley, 1993.

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

[3] Jacquin, A. E., *Fractal image coding: A review,* Proceedings of the IEEE 81,10 (1993) 1451–1465.

[4] Barthel, K. U., Schüttemeyer, J., Voyé, T., Noll, P., *A new image coding technique unifying fractal and transform coding,* in: *Proc. IEEE ICIP-94,* Austin, Texas, Nov. 1994.

[5] Gharavi-Alkhansari, M., Huang, T., *Fractal based techniques for a generalized image coding method,* in: *Proc. IEEE ICIP-94,* Austin, Texas, Nov. 1994.

[6] Lepsøy, S., *Attractor Image Compression: Fast Algorithms and Comparisons to Related Techniques,* PhD Thesis, The Norwegian Institute of Technology, Trondheim, Norway, June 1993.

[7] Hamzaoui, R., *Codebook clustering by self-organizing maps for fractal image compression,* in: *NATO ASI Conf. Fractal Image Encoding and Analysis,* Trondheim, July 1995, to appear in a special issue of the journal Fractals. Also available by anonymous ftp in ftp.informatik.uni-freiburg.de in /documents/papers/fractal/Hamz95.ps.gz

[8] Saupe, D., *From classification to multidimensional keys,* in: *Fractal Image Compression — Theory and Application,* Y. Fisher (ed.), Springer-Verlag, New York, 1994.

[9] Fisher, Y., *Fractal Image Compression — Theory and Application,* Springer-Verlag, New York, 1994.

[10] The Self-Organizing Map, Version 3.1, Helsinki University of Technology, Laboratory of Computer and Information Science, April 7, 1995.