

Quadtree based variable rate oriented mean shape-gain vector quantization

Raouf Hamzaoui, Bertram Ganz and Dietmar Saupe

Institut für Informatik, Universität Freiburg

Am Flughafen 17, 79110 Freiburg, Germany.

E-mail:hamzaoui,ganz,saupe@informatik.uni-freiburg.de.

Abstract: Mean shape-gain vector quantization (MSGVQ) is extended to include negative gains and square isometries. Square isometries together with a classification technique based on average block intensities enable us to enlarge the MSGVQ codebook size without any additional storage requirements while keeping the complexity of both the codebook generation and the encoding manageable. Variable rate codes are obtained with a quadtree segmentation based on a rate-distortion criterion. Experimental results show that our scheme performs favorably when compared to previous product code techniques or quadtree based VQ methods.

1 Introduction

Unconstrained vector quantization suffers from complexity and storage limitations which restrict its applicability. Moreover, it has been observed that important features of the image such as edges may not be well represented if codebooks are constrained in size. Several methods have been devised to overcome these problems. One of them is to use product code techniques [1]. Mean shape-gain vector quantization (MSGVQ) is a product code technique first introduced by Murakami *et al.* [2]. The method works as follows. For a block $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \mathbf{R}^n$, define the mean $m = \frac{1}{n} \mathbf{x}^T \cdot \mathbf{1}$, the gain $g = \|\mathbf{x} - m\mathbf{1}\| = \sqrt{(\mathbf{x} - m\mathbf{1})^T \cdot (\mathbf{x} - m\mathbf{1})}$ and, if $g \neq 0$, the shape $\mathbf{s} = \frac{1}{g} (\mathbf{x} - m\mathbf{1})$, where $\mathbf{1}$ is the block of constant intensity $\mathbf{1} = (1, \dots, 1)^T \in \mathbf{R}^n$. Then \mathbf{x} can be decomposed as

$$\mathbf{x} = g \mathbf{s} + m \mathbf{1}.$$

The encoding consists of approximating each block \mathbf{x} by the reconstruction vector

$$\hat{\mathbf{x}} = \hat{g} \hat{\mathbf{s}} + \hat{m} \mathbf{1}$$

that minimizes the distortion error

$$d(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2.$$

Here \hat{m} is a quantization level of the mean from a scalar codebook \mathcal{C}_m of size N_m , \hat{g} is a quantization level of the gain from a scalar codebook \mathcal{C}_g of size N_g , and $\hat{\mathbf{s}}$ is a

code vector of zero mean and norm one which approximates the shape from a vector codebook \mathcal{C}_s of size N_s . Now, let us introduce the residual vector $r(\mathbf{x}) = \mathbf{x} - m\mathbf{1}$. Then we have

$$d(\mathbf{x}, \hat{\mathbf{x}}) = \|r(\mathbf{x})\|^2 - 2\hat{g} \mathbf{x}^T \cdot \hat{\mathbf{s}} + \hat{g}^2 + n(m - \hat{m})^2. \quad (1)$$

From equality (1), the procedure for determining the optimal mean, gain and shape codewords becomes straightforward [3]. First, find the optimal reproduction mean \hat{m}_{k_o} , where

$$k_o = \arg \min_{1 \leq k \leq N_m} |\hat{m}_k - m|. \quad (2)$$

This codeword is independent of the other codewords. Next, find the optimal shape codeword $\hat{\mathbf{s}}_{i_o}$ by maximizing $\mathbf{x}^T \cdot \hat{\mathbf{s}}_i$, $1 \leq i \leq N_s$. Finally, use the result to obtain the optimal gain codeword \hat{g}_{j_o} , where

$$j_o = \arg \min_{1 \leq j \leq N_g} (\hat{g}_j - \mathbf{x}^T \cdot \hat{\mathbf{s}}_{i_o})^2.$$

Oehler and Gray [3] used pruned tree-structured vector quantization for the generation of the three codebooks \mathcal{C}_m , \mathcal{C}_g and \mathcal{C}_s to provide encodings at variable rates. In this paper we take another approach to variable rate MSGVQ:

1. We use a quadtree scheme to segment the image into blocks of different sizes which are encoded by product code codebooks designed for each level of the quadtree. Variable block size image coding with quadtrees has been introduced by Vaisey and Gersho [4, 5]. However, VQ was used only for blocks of size 4×4 . For blocks of larger size, transform coding with VQ of the coefficients was employed.
2. We also extend MSGVQ to include negative gains in the following way. Let $\hat{\mathbf{s}}$ be a codebook vector of zero mean and norm one. Then any vector \mathbf{x} can be approximated as the linear combination $\hat{g}\hat{\mathbf{s}} + \hat{m}\mathbf{1}$, where \hat{g} and \hat{m} are quantization scalars of the coefficients g and m obtained in the least squares optimization

$$\min_{g, m \in \mathbf{R}} \|\mathbf{x} - (g\hat{\mathbf{s}} + m\mathbf{1})\|^2.$$

Although m is still the mean of \mathbf{x} , $g = \mathbf{x}^T \cdot \hat{\mathbf{s}}$ can now take negative values. By allowing negative gains, we expect to improve the overall quantization (see Section 4). However, the price is that for given shape, mean and gain codebooks, searching for the optimal gain and shape codewords cannot be done sequentially as explained above, and requires a more complex procedure.

3. We propose to virtually enlarge the shape codebook by a factor of eight without any supplementary storage requirements by considering rotated and reflected versions of the shape codebook blocks. This will be helpful in producing more edge orientations which may not be well represented in the shape codebook.

The resulting scheme which we call *oriented mean¹ shape-gain* VQ (OMSGVQ) is very similar to fractal image compression [7, 8]. At the encoder the only difference is that in the latter the shape codebook is taken directly from the test image, and need not be transmitted to the decoder. For each image vector only the position of the best matching shape codebook block is required by the receiver. The resemblance between fractal image compression and MSGVQ had been first pointed out by Lepsøy *et al.* [9, 10]. A hybrid quadtree scheme combining fractal coding and OMSGVQ was presented in [11] where a pure OMSGVQ was obtained as a special case. The improvement proposed in this paper is due to the following features.

- A new quadtree scheme based on a rate-distortion criterion with rate targeting.
- Joint optimization of the shape and gain codebooks.
- Optimization of the encoder complexity.

2 The encoding algorithm

2.1 Encoding with oriented blocks

Once a shape codebook has been designed for every vector dimension (see Section 3), one may enlarge each with a factor of 8 by considering all shape codebook vectors in the 8 orientations corresponding to the 8 isometries of the square. It can be shown that the resulting codebooks improve the rate-distortion performance of the MSGVQ quadtree scheme. Of course, a better performance would have been obtained by generating shape codebooks of the same size in the usual way, i.e., without isometries. However, this alternative is not practicable for evident storage and complexity limitations. The use of square isometries is thus a suboptimal solution to shape codebook enlargement which has the merit to avoid additional storage requirements and longer codebook training. Furthermore, even the encoding complexity can be kept reasonable by using the following classification technique well known in fractal image compression [8]. Let $\{I_1, \dots, I_8\}$ be the set of the 8 isometries of \mathbf{R}^n corresponding, when a block is seen as a 2D-array B , to the 8 isometries of the square. As demonstrated in Table 1, for every block \mathbf{x} there exists a unique isometry $I_{\mathbf{x}} : \mathbf{R}^n \rightarrow \mathbf{R}^n$, $I_{\mathbf{x}} \in \{I_1, \dots, I_8\}$ having the following property. If we denote by B_i^* , $i = 1, 2, 3, 4$ the average pixel intensities of the 4 quadrants (upper left, upper right, lower left and lower right) of the 2D-array representation of $I_{\mathbf{x}}\mathbf{x}$, then these averages are ordered in one of the three canonical classes

$$\begin{aligned} \text{Class 1: } & B_1^* \geq B_2^* \geq B_3^* \geq B_4^*, \\ \text{Class 2: } & B_1^* \geq B_2^* \geq B_4^* \geq B_3^*, \\ \text{Class 3: } & B_1^* \geq B_4^* \geq B_2^* \geq B_3^*. \end{aligned}$$

¹The notion *mean/oriented* has been coined by Budge and Baker [6] in the context of mean-removed VQ.

\mathbf{x}	$I_{\mathbf{x}}$	$I_{\mathbf{x}}\mathbf{x}$	class	\mathbf{x}	$I_{\mathbf{x}}$	$I_{\mathbf{x}}\mathbf{x}$	class
	R_{180}		1		R_{180}		3
	H		2		R_{270}		1
	H		3		R_{270}		2
	D^+		1		D^+		3
	R_{90}		2		V		2
	R_{90}		3		V		1
	R_{180}		2		D^-		3
	H		1		D^-		2
	R_{270}		3		D^-		1
	D^+		2		I		3
	R_{90}		1		I		2
	V		3		I		1

Table 1: This table illustrates the concept of block canonical orientations. In the columns labeled \mathbf{x} blocks are shown with their mean quadrant intensities. There are $4! = 24$ possible rankings of these quadrant intensities. In each case the next column labeled $I_{\mathbf{x}}$ lists the isometry transformation that puts the block into its canonical orientation $I_{\mathbf{x}}\mathbf{x}$, which is shown in the following column. There are 8 isometry operators I_1, \dots, I_8 , which in the table are called

- I = identity transformation,
- R_{90} = counterclockwise rotation by 90 degrees,
- R_{180} = counterclockwise rotation by 180 degrees,
- R_{270} = counterclockwise rotation by 270 degrees,
- H = reflection at the horizontal axis,
- V = reflection at the vertical axis,
- D^+ = reflection at the diagonal,
- D^- = reflection at the anti-diagonal.

In the encoding, instead of minimizing the distortion $d(\mathbf{x}, \hat{g}I_i\hat{\mathbf{s}} + \hat{m}\mathbf{1})$ for all $\hat{g} \in \mathcal{C}_g$, $\hat{\mathbf{s}} \in \mathcal{C}_s$, $\hat{m} \in \mathcal{C}_m$, and $i \in \{1, \dots, 8\}$, one may save computing time with a slightly suboptimal technique as follows. In the codebook generation we constrain shape codebook vectors to be in canonical orientation (see Section 3). The optimal mean codeword is still given by equation (2). Then we solve the minimization problem

$$\min_{\hat{g}, \hat{\mathbf{s}}} d(\mathbf{x}', \hat{g}\hat{\mathbf{s}})$$

first for vector \mathbf{x}' being our vector \mathbf{x} in the canonical orientation $I_{\mathbf{x}}\mathbf{x}$, then in the anticanonical orientation $I_{-\mathbf{x}}(-\mathbf{x})$.² Finally, we retain the encoding with the smallest distortion. Since the gains corresponding to both $I_{\mathbf{x}}\mathbf{x}$ and $I_{-\mathbf{x}}(-\mathbf{x})$ are positive, determining the optimal gain and shape codewords can be done sequentially. The code for a block consists of the indices of the optimal gain, shape and mean codewords, and of the index of one of the isometries $I_{\mathbf{x}}^{-1}$ or $I_{-\mathbf{x}}^{-1}$. Also, we decide not to send the bits of the indices of the optimal isometry, gain and shape codewords if $|g| < \epsilon$, where ϵ is a prefixed small bound.

2.2 The quadtree algorithm

The algorithm aims at finding the optimal quadtree encoding for a given compression ratio. The quadtree partition may consist of square blocks of size $2^k \times 2^k$ where $k_{min} \leq k \leq k_{max}$. Suppose now that a code with a total rate of N bits is targeted. For a block B_i of size $2^k \times 2^k$, $k_{min} < k \leq k_{max}$, we define the ratio

$$\lambda(B_i) = \left| \frac{d_k - d_{k-1}}{r_k - r_{k-1}} \right|,$$

where d_k and r_k are the distortion and the rate for the optimum encoding of block B_i , respectively, and d_{k-1} and r_{k-1} are the sum of the distortions and the sum of the rates, respectively, for the optimum encoding of the subblocks obtained by splitting block B_i in 4 subblocks of size $2^{k-1} \times 2^{k-1}$. Thus, $\lambda(B_i)$ denotes the reduction in distortion per additional bit for the code when block B_i is subdivided into 4 quadrants. We start by finding the optimal code for the image partitioned into blocks of maximum size. Then we successively split that block with the highest ratio λ until the total rate of the code becomes larger than N or all blocks in the quadtree partition have minimum size.

3 Codebook generation

The jointly optimized codebook training technique for shape-gain VQ developed by Sabin and Gray [12] can be easily extended to oriented mean shape-gain VQ. Since

$$\mathbf{x}^T \cdot \hat{\mathbf{s}} = r(\mathbf{x})^T \cdot \hat{\mathbf{s}},$$

²The anticanonical orientation of a block \mathbf{x} is the canonical orientation of $-\mathbf{x}$.

we get from equality (1)

$$\begin{aligned} d(\mathbf{x}, \hat{\mathbf{x}}) &= \|r(\mathbf{x})\|^2 - 2\hat{g}r(\mathbf{x})^T \cdot \hat{\mathbf{s}} + \hat{g}^2 + n(m - \hat{m})^2 \\ &= d(r(\mathbf{x}), \hat{g}\hat{\mathbf{s}}) + n(m - \hat{m})^2. \end{aligned} \quad (3)$$

Clearly, the mean codebooks can be constructed directly from the means of the training vectors \mathbf{x} . For the shape and gain codebooks, joint optimization is required. But from (3) we see that we simply have to use the residual blocks $r(\mathbf{x})$ as training vectors which are in addition transformed in the canonical orientation for positive gains, and in the anticanonical orientation for negative gains for original vector \mathbf{x} . Even though the decoder needs both shape codebooks, no bit has to be sent as side information since the decoder can identify the sign of the gain. However in our implementation, we opted for a simpler variant. We constructed only the shape and gain codebooks corresponding to the canonical orientation and used them also for encoding blocks in the anticanonical orientation.

4 Experimental results

A training sequence of 10 images was used to design the codebooks. The test images were not included in the training sequence. We provide coding results for the green band of the USC 512×512 color image of Lenna and for the 512×512 Barbara image. The smallest subblock size was 4×4 and the largest was 16×16 . Training vectors for each block size were obtained after segmenting the training images with a quadtree algorithm based on the following n -fold variance criterion [13]. After partitioning the image into blocks of maximum size, a block $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ is split into 4 subblocks if $\|r(\mathbf{x})\|^2 > \delta$, where δ is a given tolerance value. Increasing the number of training vectors of minimum size is obtained by decreasing the value of δ . Note that this is not the conventional variance criterion because larger blocks have here a stronger weight. For the gain and the mean codebooks, 5 and 7 bits were spent, respectively. Lloyd-Max quantization was used to generate the mean codebooks. All shape codebooks were of size 256. With these settings the product codebook contains a total of $2^{5+7+8+3} = 2^{23}$ blocks of each size. Figure 1 shows the results for the 512×512 Lenna image for three different encodings:

1. with positive gains only and without isometries,
2. with positive and negative gains and without isometries,
3. with positive and negative gains and with isometries.

Figure 2 shows the results for the 512×512 Barbara image. For all tests the bound ϵ was set to 6, 12 and 24 for dimensions 4×4 , 8×8 and 16×16 , respectively.

First, as expected, we observe that negative gains improved slightly the performance of our coder with an increase, however, in complexity. Second, our experiments clearly show the efficiency of isometries. The encoding times measured on an SGI Indigo2 running a MIPS R4400 150 MHz processor were reasonable varying between

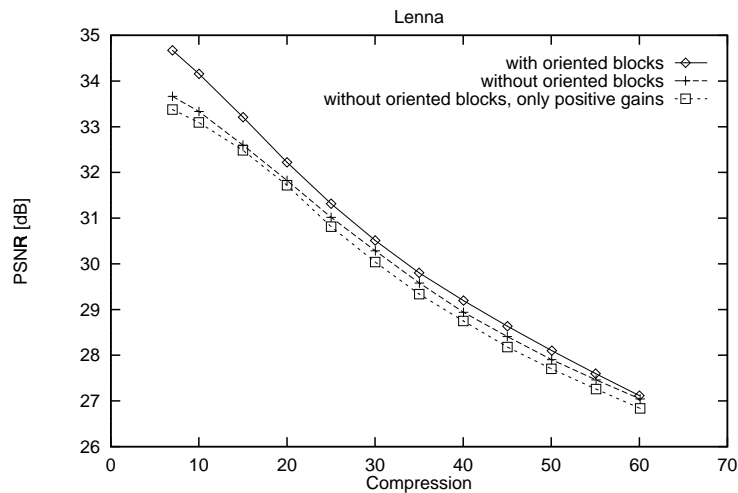


Figure 1: PSNR vs. Compression ratio for the 512×512 Lenna image.

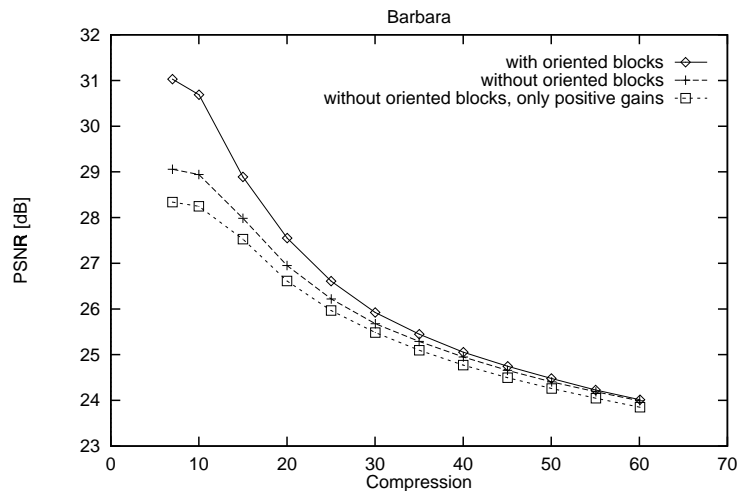


Figure 2: PSNR vs. Compression ratio for the 512×512 Barbara image.

10 and 70 seconds. Figure 3 shows the segmentation of the Lenna image resulting from our rate-distortion quadtree algorithm. We note that our results are superior to those published in previous comparable works. For example, Vaisey and Gersho [5] reported for the Lenna image a PSNR value of 31.36 dB at 0.363 bpp while our best scheme (with isometries and negative gains) yielded for the same rate a PSNR value of 31.88 dB in 47 seconds CPU time. For the Barbara image our PSNR values were roughly 1 dB higher at both reported rates. For the 512×512 Lenna image Oehler and Gray [3] obtained a PSNR value of 33.18 dB at a rate of 0.58 bpp while our scheme provided for this rate a PSNR value of 33.45 dB in 45 seconds CPU time.

The overall perceptual quality of our coded images was satisfying. Particularly, edges were well rendered. We remark, however, that blocking artifacts appeared at low compression ratios. This is typical of block encoding schemes based on non-overlapping blocks including OMSGVQ from this work. Postprocessing techniques (a

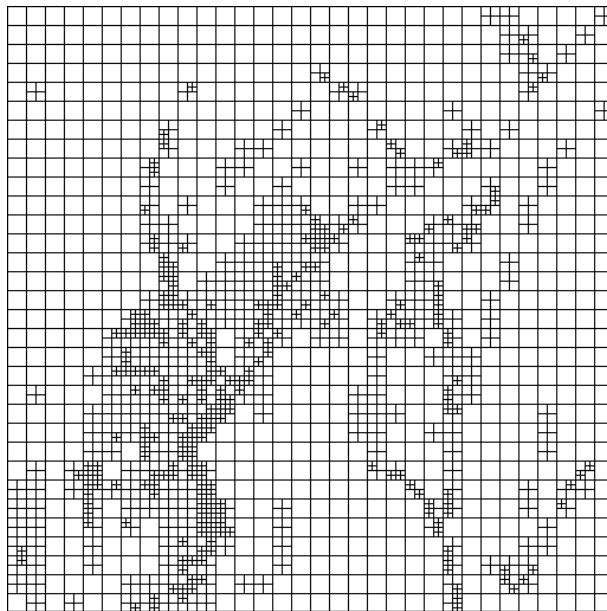


Figure 3: Quadtree segmentation of the 512×512 Lenna image at 0.25 bpp.

simple technique is in [8]) commonly are applied to reduce these artifacts. Application of postprocessing to the encodings studied in this work significantly improved the visual quality and, moreover, increased the PSNR by up to about 0.3 dB for low bitrate encodings. In Figure 4 we show the 512×512 Boat image coded at 0.25 bpp with postprocessing.

Finally, a reduction in bitrate can be obtained by entropy coding the bitstream of the mean information. A method based on DPCM with border pixel estimator is proposed in [14].

5 Conclusion and future work

We have presented a variable rate product code technique for vector quantization of images. A quadtree segmentation based on a rate-distortion criterion with a target rate was used. Product code codebooks were designed for each of the allowed block sizes. In spite of its simplicity, our scheme provided satisfying results, both visually and with respect to mean squared error for a large range of compression ratios. Furthermore, both storage requirements and encoding times were low. Our results contrast those found for fractal image compression in many aspects. First, while there is no need to employ the isometries in fractal image compression since their loss can be easily compensated by making the implicit shape codebook larger [15], storage limitations justify their use in MSGVQ. Second, error propagation at the decoding, which is typical in fractal image compression, does not occur in OMSGVQ where the decoding is noniterative. Moreover, in order to ensure the convergence of the decoding, suboptimal constrained gains are frequently used in fractal image compression. This is not the case in OMSGVQ.



Figure 4: The 512×512 Boat image coded with OMSGVQ at 0.25 bpp and post-processed with the technique in [8]. The PSNR is equal to 30.35 dB.

Future work may include a rate-distortion optimization which uses both pruned tree-structured codebooks and hierarchical encoding. One may also try other adaptive partitioning schemes, which proved superior to the quadtree scheme in fractal image compression [8, 16].

References

- [1] Gersho, A., Gray, R. M., *Vector quantization and signal compression*, Kluwer Academic Publishers, 1992.
- [2] Murakami, T., Asai, K., Yamazaki, E., *Vector quantizer of video signals*, *Electronics Letters* 7 (1982) 1005–1006.
- [3] Oehler, K., L., Gray, R., M., *Mean-gain-shape vector quantization*, in: *Proc. ICASSP-93 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 5, 1993.
- [4] Vaisey, D.J., Gersho, A., *Variable block-size image coding*, in: *Proc. ICASSP-87 IEEE International Conference on Acoustics, Speech and Signal Processing*, 1987.

- [5] Vaisey, D.J., Gersho, A., *Image compression with variable block size segmentation*, IEEE Trans. Signal Processing 40,8 (1992) 2040–2060.
- [6] Budge, S. E., Baker, R. L., *Compression of color digital images using vector quantization in product codes*, in: *Proc. ICASSP-85 IEEE International Conference on Acoustics, Speech and Signal Processing* , pp. 129–132, April 1985.
- [7] Jacquin, A. E., *Image coding based on a fractal theory of iterated contractive image transformations*, IEEE Trans. Image Processing 1 (1992) 18–30.
- [8] Fisher, Y., *Fractal Image Compression — Theory and Application*, Springer-Verlag, New York, 1994.
- [9] Lepsøy, S., *Attractor Image Compression: Fast Algorithms and Comparisons to Related Techniques*, PhD Thesis, The Norwegian Institute of Technology, Trondheim, Norway, June 1993.
- [10] Ramstad, T. A., Lepsøy, S., *Block-based attractor coding: Potential and comparison to vector quantization*, in: *Conference Record of the Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, pp. 1504–1508, 1993.
- [11] Hamzaoui, R., Müller, M., Saupe, D., *VQ-enhanced fractal image compression*, in: *Proc. ICIP-96 IEEE International Conference on Image Processing*, Lausanne, Sept. 1996.
- [12] Sabin, M., J., Gray, R. M., *Product code vector quantizers for waveform and voice coding*, IEEE Trans. Acoust. Speech and Signal Proc. ASSP-32:474-488, 1984.
- [13] Saupe, D., Jacob, S., *Variance-based quadrees for fractal image compression*, to appear in Electronics Letters.
- [14] Boxerman, J. L., Lee, H. J., *Variable block-sized vector quantization of grayscale images with unconstrained tiling*, in : *Proc. SPIE Visual Communications and Image Processing'90*, Vol. 1360, 1990.
- [15] Saupe, D., *The futility of square isometries in fractal image compression*, in: *Proc. ICIP-96 IEEE International Conference on Image Processing*, Lausanne, Sept. 1996.
- [16] Saupe, D., Ruhl, M., *Evolutionary fractal image compression*, in: *Proc. ICIP-96 IEEE International Conference on Image Processing*, Lausanne, Sept. 1996.