REFERENCES

[1] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression.* Englewood Cliffs, NJ: Prentice-Hall, 1971.

[2] A. Buzo, A. H. Gray, Jr., R. M. Gray, and J. D. Markel, "Speech coding based upon vector quantization," *IEEE Trans. Inform. Theory,* vol. IT-28, no. 5, pp. 562–574, 1980.

[3] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Trans. Inform. Theory,* vol. 35, no. 2, pp. 299–315, 1989.

[4] J. Lin, "Vector quantization for image compression: Algorithms and performance," Ph.D. dissertation, Brandeis Univ., Waltham, MA, 1992.

[5] J. Lin and J. A. Storer, "Design and performance of tree-structured vector quantizers," *Inform. Processing Mgmt.,* vol. 30, no. 6, pp. 851–862, 1994.

[6] J. Lin, J. A. Storer, and M. Cohn, "Optimal pruning for tree-structured vector quantization," *Information Processing Mgmt.,* vol. 28, no. 6, pp. 723–733, 1992.

[7] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.,* vol. 28, pp. 84–95, 1980.

[8] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory,* vol. IT-28, no. 2, pp. 129–136, 1982.

[9] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proc. IEEE,* vol. 73, pp. 1551–1588, 1985.

[10] K. L. Oehler, E. A. Riskin, and R. M. Gray, "Unbalanced tree-growing algorithms for practical image compression," in *Proc. IEEE ICASSP,* 1991, pp. 2293–2296, vol. 4.

[11] E. A. Riskin and R. M. Gray, "A greedy tree growing algorithm for the design of variable rate vector quantizers," *IEEE Trans. Signal Processing,* vol. 39, no. 11, pp. 2500–2507, 1991.

[12] ——, "Look ahead in growing tree-structured vector quantizers," in *Proc. IEEE ICASSP,* 1991, pp. 2289–2292, vol. 4.

Extended Lapped Transform in Image Coding

Ricardo L. de Queiroz and K. R. Rao

***Abstract*—A modulated lapped transform with extended overlap (ELT) is investigated in image coding with the objective of verifying its potential to replace the discrete cosine transform (DCT) in specific applications. Some of the criteria utilized for the performance comparison are reconstructed image quality (both objective and subjective), reduction of blocking artifacts, robustness against transmission errors, and filtering (for scalability). Also, a fast implementation algorithm for finite-length-signals using symmetric extensions is developed specially for the ELT with overlap factor 2 (ELT-2). This comparison shows that ELT-2 is superior to both DCT and the lapped orthogonal transform (LOT).**

I. INTRODUCTION

While block transforms became very popular in the image coding field, the lapped orthogonal transform (LOT) [1] arose as a promising competitor to such transforms as the discrete cosine transform (DCT) [2], which is the block transform used in most image and video coding algorithms [2]. The advantage of lapped transforms [3] resides on the length of their basis functions, providing improved filtering capabilities and reduction of blocking artifacts commonly present in block transform coding at low bit rates. Furthermore, the concept of lapped transforms was established and proven to be equivalent to the concept of paraunitary FIR uniform filter banks [3], [4]. Under this point of view, both the LOT and the DCT are considered as special choices of paraunitary filter banks [3], [4]. Cosine modulated filter banks [4] allow perfect reconstruction (PR) in paraunitary analysis-synthesis systems, using a modulation of a lowpass prototype by a cosine train. By a proper choice of the phase of the modulating cosine, Malvar developed the modulated lapped transform (MLT) [5], which led to the so-called extended lapped transforms (ELT) [3], [6]. The ELT allows several overlapping factors, generating a family of PR cosine modulated filter banks. Other cosine-modulation approaches have also been developed (see, for example, [4], [7], and references therein) and the most significant difference among them is the lowpass prototype choice and the phase of the cosine sequence.

Manuscript received July 12, 1993; revised June 8, 1994. This work was supported in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Brazil, under Grant 200.804-90-1. The associate editor coordinating the review of this paper and approving it for publication was Prof. Nasser M. Nasrabadi.

The authors are with the Electrical Engineering Department, University of Texas at Arlington, Arlington, TX, 76019 USA.

IEEE Log Number 9411133.

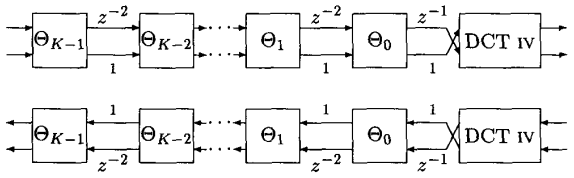


Fig. 1. Flow graph for the direct (top) and inverse (bottom) ELT. Each branch carries $M/2$ samples.

Let M and L be the number of channels and filters' length, respectively, where, for the ELTs, $L = 2KM$, and K is the overlap factor. The analysis filters ($f_m(n)$) are time-reversed versions of the synthesis filters ($g_m(n)$) as in any paraunitary filter bank (for $m = 0, 1, \cdots, M-1$ and $n = 0, 1, \cdots, L-1$). The ELT class is defined by [3], [6]

$$g_m(n) = f_m(L-1-n) = h(n)\cos\left\{\left(m+\frac{1}{2}\right)\cdot\left[\left(n-\frac{L-1}{2}\right)\frac{\pi}{M}+(N+1)\frac{\pi}{2}\right]\right\} \quad (1)$$

for $m = 0, 1\cdots, M-1$ and $n = 0, 1, \cdots, L-1$. $h(n)$ is a symmetric window modulating the cosine sequence and the impulse response of a lowpass prototype (with cutoff frequency at $\pi/2M$) which is translated in the frequency domain to M different frequency slots in order to construct the uniform filter bank. We will use ELT with $K = 2$, which will be designated as ELT-2, and assume row-column separable implementation of the transform. Therefore, one-dimensional (1-D) analysis of the transform implementation is sufficient for two-dimensional (2-D) applications.

The lattice-style algorithm [3] is shown in Fig. 1 for an ELT with generic overlap factor K. The stages Θ_n contain the plane rotations and are defined by

$$\Theta_n = \begin{bmatrix} -\mathbf{C}_n & \mathbf{S}_n\mathbf{J} \\ \mathbf{J}\mathbf{S}_n & \mathbf{J}\mathbf{C}_n\mathbf{J} \end{bmatrix}, \quad (2)$$

$$\mathbf{C}_n = diag\left\{\cos(\theta_{0,n}), \cdots, \cos\left(\theta_{(M/2)-1,n}\right)\right\},$$
$$\mathbf{S}_n = diag\left\{\sin(\theta_{0,n}), \cdots, \sin\left(\theta_{M/2-1,n}\right)\right\}$$

where $\mathbf{J}$ is the $M/2 \times M/2$ counter-identity (reversing) matrix [3]. Also, $\theta_{i,j}$ are rotation angles such that Θ_n is of the form indicated in Fig. 2, containing $M/2$ orthogonal butterflies. We use the optimized angles presented in [3].

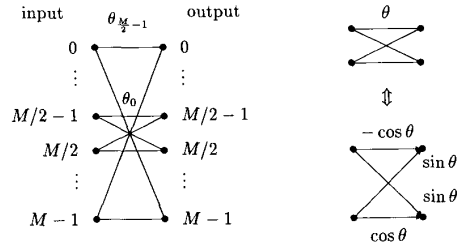


Fig. 2. Implementation of plane rotations stage showing the displacement of the $M/2$ butterflies.

II. Theoretical Motivation

The coding gain G_{TC}, in dB, of a transform/subband scheme is defined as [8]

$$G_{TC} = 10 \log_{10} \left(\frac{\frac{1}{M} \sum_{i=0}^{M-1} \sigma_i^2}{\left(\prod_{i=0}^{M-1} \sigma_i^2 \right)^{1/M}} \right) \tag{3}$$

with σ_i^2 as the variance of the ith subband signal (transform coefficient) for $i = 0, \cdots, M-1$. Implementation complexity (C) will be measured here by the number of floating-point operations (additions plus multiplications) per sample (FLOPS) required to implement the 1-D transform. Another way to generate transforms with longer overlap is based on the hierarchical connection of two-band filter banks following the paths of a binary tree. For parallel M-band systems, the full-tree is applied. If S stages of filter banks are cascaded, the resulting filter bank will have $M = 2^S$ channels and the resulting filters have length $(M-1)(L_{2B}-1)+1$, where L_{2B} is the length of the filters in the two-band filter bank used as a basic cell for the hierarchical structure.

In Table I are shown G_{TC} and C for the DCT, LOT, and ELT-2. Additionally, we included the same parameters for the tree-structured filter banks based on two-band systems with filters with eight and 16 taps. We used Smith and Barnwell conjugate quadrature filters (CQF) [9] and Johnston's quadrature mirror filters (QMF) [10]. Note that eight-tap filters lead to equivalent filter banks whose filters have length closer to ELT-2. The "IDEAL" entry in this table refers to ideal brick-wall filters, which only can be implemented using infinite-length filters. The input signal was assumed to follow an AR(1) model with adjacent-sample correlation $\rho = 0.95$. From Table I, we can see that ELT-2 has coding gain similar to 8A tree-structured filter banks at a much lower complexity. The complexity of the DCT is unbeatably low, and LOT and ELT-2 are regarded as improvements for which the trade-off between costs and benefits has to be taken into account. LOT has proven to be superior to DCT, leading to more pleasant images even at high compression rates. We will show later that ELT-2 surpasses LOT in performance.

One of the incentives to study ELT's of larger overlap for image coding resides in their longer basis functions and, therefore, in their potential for better spectral selectivity of each subband filter. In fact, the ELT-2 has filters with good stopband attenuation. The filtering capability is supposed to be reflected by G_{TC} measurements, in the case of nonlayered image coding. However, in compatible coding [11], filtering performance is a plus. In this approach, an image, let us say, of size $2N \times 2N$ is encoded using any transform/quantization method and the receiver makes the option of decoding the $2N \times 2N$ image or a reduced version of it, such as a $N \times N$ image. A straightforward method can be: inverse transform followed by anti-aliasing filtering and subsampling. As a faster and more efficient approach, one can retain $M/2 \times M/2$ low-frequency coefficients in each block (out of $M \times M$) and perform a pruned inverse transform, resulting in a reconstructed image at a lower spatial resolution. This is equivalent to transform domain filtering followed by subsampling. Fig. 3 shows the frequency response of the first four filters for the eight-channel DCT, LOT, and ELT-2. They are combined into one band, which is the frequency response of the lowpass filter actually implemented when four (out of eight) coefficients are retained.

TABLE I
G_{TC} IN dB AND IMPLEMENTATION COMPLEXITY (C) IN FLOPS FOR VARIOUS TRANSFORMS AND BLOCK SIZES. FOR THE TREE STRUCTURED FILTER BANKS, FULL-TREE IS APPLIED AND $M = 2^S$, WHERE S IS THE NUMBER OF STAGES OF THE TREE

	C	G_{TC}	C	G_{TC}	C	G_{TC}
	$M=4$		$M=8$		$M=16$	
DCT	3.5	7.57	5.25	8.83	7.13	9.46
LOT	7	7.95	9.50	9.20	11.75	9.69
ELT-2	9	8.39	11	9.48	13	9.90
QMF-8A	16	8.31	24	9.32	32	9.67
CQF-8A	18	8.44	27	9.47	36	9.84
QMF-16B	32	8.52	48	9.56	64	9.92
CQF-16B	34	8.54	51	9.58	68	9.94
IDEAL	∞	8.56	∞	9.59	∞	9.94

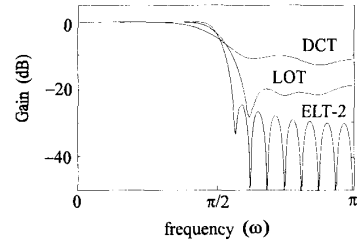


Fig. 3. Frequency response of a lowpass filter with cut-off frequency = $\pi/2$ produced by the first four filters of a $M = 8$ transform. Plots for DCT, LOT, and ELT-2 are shown.

III. Fast Size-Limited ELT-2 Algorithm

One of the major problems for the use of the ELT-2 resides on the transform applied to blocks near the borders of the image. As the transform contains overlap, samples outside the image boundaries may be included in the analysis section. On the other hand, extra transformed blocks are needed in the synthesis process to reconstruct the signal. Periodic extensions can solve this problem, at the expense of inserting artificial edges caused by unequal luminance levels in the extremes of the image. The symmetric extension method is useful for linear-phase filter banks [12]. However, the ELT does not have linear-phase filters. In [13] a nonorthogonal solution is found using sample extensions (including symmetric extension) and post-processing techniques. Malvar devised a fast and orthogonal algorithm [3] implying change of the ELT-2 near the signal boundaries. In Fig. 4 we introduce a fast implementation flow-graph for the ELT-2

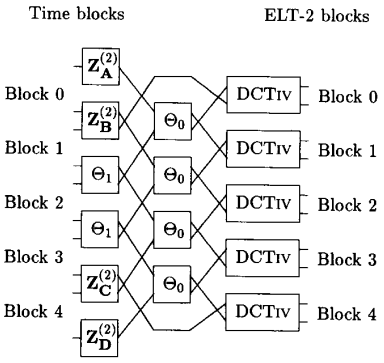


Fig. 4. Flow graph for finite-length signals for ELT-2 ($K = 2$). Each branch carries $M/2$ samples. Forward transform is performed by following the flow-graph from left to the right, while inverse transform is performed by following the flow-graph in the opposite direction and substituting the $\mathbf{Z}$ matrices by their inverses.

using symmetric extensions, where

$$\mathbf{Z}_A^{(2)} = (\mathbf{S}_1 - \mathbf{C}_1)\mathbf{J} \tag{4}$$

$$\mathbf{Z}_B^{(2)} = \begin{bmatrix} -\mathbf{C}_1 & \mathbf{S}_1\mathbf{J} \\ (\mathbf{S}_0 - \mathbf{C}_0)\mathbf{S}_1 & (\mathbf{S}_0 + \mathbf{C}_0)\mathbf{C}_1\mathbf{J} \end{bmatrix} \tag{5}$$

$$\mathbf{Z}_C^{(2)} = \begin{bmatrix} \mathbf{J}(\mathbf{C}_0 - \mathbf{S}_0)\mathbf{C}_1 & \mathbf{J}(\mathbf{C}_0 + \mathbf{S}_0)\mathbf{S}_1\mathbf{J} \\ \mathbf{J}\mathbf{S}_1 & \mathbf{J}\mathbf{C}_1\mathbf{J} \end{bmatrix} \tag{6}$$

$$\mathbf{Z}_D^{(2)} = \mathbf{J}(\mathbf{S}_1 + \mathbf{C}_1). \tag{7}$$

The algorithm corresponds to the application of a symmetric extension as if the signal was folded around the borders. In these flow-graphs, each branch carries $M/2$ samples and analysis is accomplished by following the paths from left to the right, while the synthesis (inverse transform) is achieved by following the paths from right to the left, replacing the $\mathbf{Z}$ matrices by their inverses. Note that $\mathbf{Z}_A^{(2)}$ and $\mathbf{Z}_D^{(2)}$ are simple counter-diagonal matrices and their inverses have the same basic format. $\mathbf{Z}_B^{(2)}$ and $\mathbf{Z}_C^{(2)}$ are composed of $M/2$ butterflies, similar to those in Fig. 2, but the lattice is no longer orthogonal [see (5) and (6)]. Their inverses are obtained by inverting each of the butterflies. As a result, both analysis or synthesis have the same fast algorithm. The DCT-IV and Θ_n matrices do not need replacement in synthesis because they are both symmetric and orthogonal.

A comparison of Malvar's solution against ours is carried in Fig. 5 using a JPEG baseline coder and replacing DCT by ELT-2. Although Malvar's solution is PR and orthogonal, the effects of quantization on the borders are intense and unpleasant. Our nonorthogonal algorithm also suffers from similar effects; however, the distortion is just noticeable in one of the borders ($\mathbf{Z}_D^{(2)}$ has a somewhat high condition number) and on the last one or two columns/lines of pixels. Thus, the distortion is virtually invisible because it is masked by the background.

IV. Image Coding Simulation with Information Loss

Most communications protocols packetize data into cells and provide cell prioritization to protect more important data, as cell losses can occur. This is so, for example, for ATM networks, which are gaining acceptance lately. For image and video transmission in such networks, compressed data can be unrecoverably lost and a more robust encoding approach is necessary, as well as developing reconstruction procedures. For such environments we propose the use of the ELT-2 as a simple and effective way to recover the lost data.

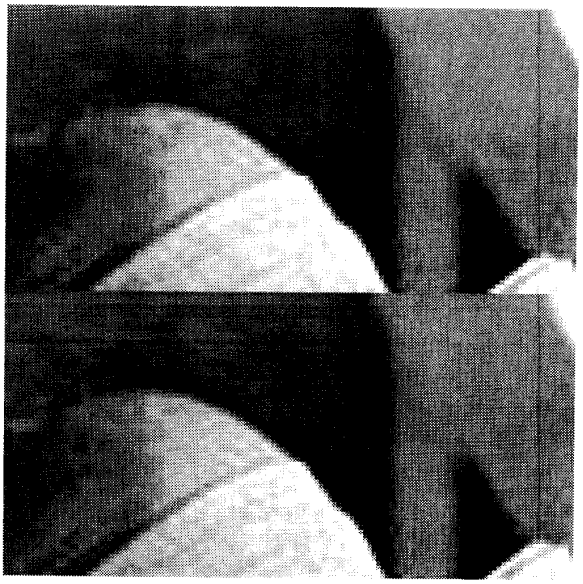


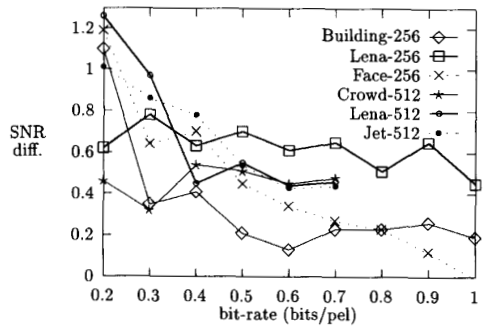
Fig. 5. Comparison of fast ELT-2 implementation methods for finite-length-signal regarding sensitivity to quantization errors at the borders. A 256 × 256-pels image is encoded at 0.7 bits/pel using the JPEG coder and a portion of each reconstructed image is shown. Top: proposed method; bottom: orthogonal method.

In order to simplify the problem and its modeling, we assume that [15]: i) the DC coefficients of the transform are prioritized and transmitted with enough protection so that they are not susceptible to errors; ii) if a cell loss occurs, the information regarding all AC coefficients of only one block is lost. In this case, see [15] for details on packet losses and on efficient methods for estimating the lost AC coefficients using the LOT. The ELT-2 is expected to perform better than LOT because of its larger overlapping, since the spatial region affected by the lost-block will increase making the error locally less intense. However, this is only partially true, since the robustness of the ELT-2 comes from its nonlinear-phase-filters allied with the longer overlap. The error using ELT-2 is sparser compared to a filter bank with the same filters' length but with linear-phase (such as a LOT with extended overlap [16], which we refer as LOT-2). We transformed an image using various transforms and deleted all the coefficients of a single block except for the DC term. After respective inverse transforms, Fig. 6 shows a zoom of the region where a lost-block occurred, using DCT, LOT, LOT-2, and ELT-2. From the DCT results, we can clearly see where the lost-block is located and, comparing all the transforms, we can see that the ELT-2 performed fairly better than its competitors in view of its effective error masking properties.

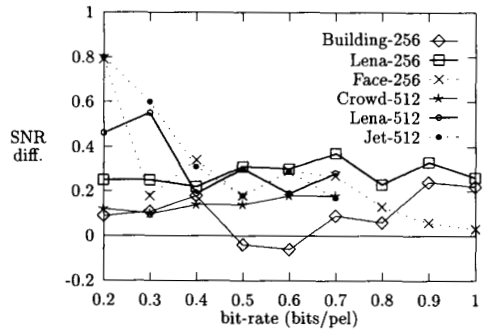
To carry out our tests, we selected the JPEG baseline coder (JPEG) [14] because its algorithm is popular and well understood. We assume it will resemble image/video coders actually used in packetized transmission of images. Then, we replaced the 8 × 8 DCT by ELT-2 or LOT with $M = 8$.

The objective SNR tests were carried out using several images at different bit-rates and the results are shown in Fig. 7, assuming no cell-losses. It can be seen that the ELT-2 outperformed both LOT and DCT. Fig. 8 shows the 256 × 256 pels image Lena coded at 0.8 bit/pel (bpp) using JPEG for both the DCT and the ELT-2. For a better visualization, a dramatic error rate was used, and we simulated 5 % rate of lost blocks (51 blocks are lost). The errors occur randomly, but we forced the position of the errors used for the DCT to be repeated

Fig. 6. Trivial image reconstruction when all AC coefficients of a single block are lost. The AC coefficients are set to zero in this block and a zoom of reconstructed images for each transform are shown. Top left corner, DCT. Top right corner, LOT. Bottom left corner, LOT-2. Bottom right corner, ELT-2.



(a)



(b)

Fig. 7. SNR difference (in dB) between ELT-2 and other transforms using JPEG for several bit rates. The test images have either 256×256-pels or 512×512-pels, and their names and sizes are indicated. (a) $\mathrm{SNR}_{\mathrm{ELT\text{-}2}} - \mathrm{SNR}_{\mathrm{DCT}}$. (b) $\mathrm{SNR}_{\mathrm{ELT\text{-}2}} - \mathrm{SNR}_{\mathrm{LOT}}$.

for the ELT-2, so that we can compare the effects of a block-loss for both transforms in several positions in the test image.

(a)

(b)

Fig. 8. Image compression results for 256×256-pels and eight bpp image Lena, using JPEG to compress the image to a rate of 0.8 bpp. Both compressed images are subject to a block-loss rate of 5%. (a) DCT, (b) ELT-2.

V. Conclusions

The finite-length implementations, presented here, allow the ELT-2 to be efficiently computed even near the borders. The ELT-2 has proven to be very robust against cell losses and it can be implemented with a minimal increase in computation compared to the LOT. Actually, the ELT-2 masks so well the block-losses that we believe that when the data is lost, the AC coefficients in the block can be subject of very simple reconstruction procedures or simply ignored. This approach led to very good results and may simplify error correction and reconstruction procedures for packetized transmission of images. This transform reveals itself to be a very

attractive alternative for image coding, replacing the DCT or the LOT. While the DCT is computationally much simpler, ELT-2 is slightly more complex compared to the LOT. ELT-2, however, substantially improves the subjective quality for low bit-rates and/or when data is lost due to channel errors.

REFERENCES

[1] H. S. Malvar and D. H. Staelin, "The LOT: Transform coding without blocking effects," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 37, pp. 553–559, Apr. 1989.

[2] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications.* San Diego, CA: Academic, 1990.

[3] H. S. Malvar, *Signal Processing with Lapped Transforms.* Norwood, MA: Artech House, 1992.

[4] P. P. Vaidyanathan, *Multirate Systems and Filter Banks.* Englewood Cliffs, NJ: Prentice-Hall, 1993.

[5] H. S. Malvar, "Lapped transforms for efficient transform/subband coding," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 38, pp. 969–978, June 1990.

[6] H. S. Malvar, "Extended lapped transform: Fast algorithms and applications," in *Proc. Intl. Conf. Acoust., Speech, Signal Processing,* Toronto, Canada, 1991, pp. 1797–1800.

[7] R. D. Koilpillai and P. P. Vaidyanathan, "Cosine modulated FIR filter banks satisfying perfect reconstruction," *IEEE Trans. Signal Processing,* vol. 40, pp. 770–783, April 1992.

[8] N. S. Jayant and P. Noll, *Digital Coding of Waveforms* . Englewood Cliffs, NJ: Prentice-Hall, 1984.

[9] M. J. T. Smith and T. P. Barnwell, III, "Exact reconstruction techniques for tree-structured subband coders," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-34, pp. 434–441, June 1986.

[10] J. D. Johnston, "A filter family designed for use in quadrature mirror filter banks," in *Proc. Intl. Conf. Acoust., Speech, Signal Processing,* Denver, CO, 1980, pp. 291–294.

[11] H. Jozawa and H. Watanabe, "Intrafield/interfield adaptive lapped transform for compatible HDTV coding," in *4th Int. Workshop on HDTV and Beyond,* Torino, Italy, Sept. 4–6, 1991.

[12] R. Bamberger, S. L. Eddins, and V. Nuri, "Generalized symmetric extensions for size-limited multirate filter banks," *IEEE Trans. on Image Processing,* vol. 3, pp. 82–87, Jan. 1994.

[13] R. L. de Queiroz, "Subband processing of finite length signals without border distortions," in *Proc. Int. Conf. Acoust., Speech, Signal Processing,* San Francisco, CA, 1992, vol. IV, pp. 613–616.

[14] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Compression Standard.* New York, NY: Van Nostrand Reinhold, 1993.

[15] P. Haskel and D. Messerschmidt, "Reconstructing lost video data in a lapped orthogonal transform based coder," in *Proc. Intl. Conf. Acoust., Speech, Signal Processing,* Albuquerque, NM, 1990, pp. 1985–1988.

[16] R. L. de Queiroz, T. Q. Nguyen, and K. R. Rao, "The generalized lapped orthogonal transforms," *Electronics Letters,* vol. 30, pp. 107–108, Jan. 20, 1994.

Region-Based Fractal Image Compression Using Heuristic Search

Lester Thomas and Farzin Deravi

***Abstract*—This paper presents work carried out on fractal (or attractor) image compression. The approach relies on the assumption that image redundancy can be efficiently exploited through self-transformability. The algorithms described in this paper utilize a novel region-based partition of the image that greatly increases the compression ratios achieved over traditional block-based partitionings. Due to the large search spaces involved, heuristic algorithms are used to construct these region-based transformations. Results for three different heuristic algorithms are given. The results show that the region-based system achieves almost double the compression ratio of the simple block-based system at a similar decompressed image quality. For the Lena image, compression ratios of 41:1 can be achieved at a PSNR of 26.56 dB.**

I. INTRODUCTION

Fractal coding is a very young science. The term fractal was first coined in 1982 by Mandelbrot in his book *The Fractal Geometry of Nature* [1]. In 1988, Barnsley and Sloan proposed using fractals generated by iterated function systems (IFS) to encode and compress images [2]–[7].

The first practical fractal compression system to appear in the literature was by Jacquin, who used transformations defined on image blocks to encode image data [11]–[13]. Since then, fractal (or attractor) image coders have received considerable attention. Most of the fractal compression systems to appear in the literature are based on Jacquin's block-based compression system [15]–[25]. This correspondence is based on experiments carried out to improve the performance of block-based fractal compression systems [8]–[10]. A novel algorithm is presented that extends the traditional block-based fractal coders by allowing transformations that act on image *regions*. A search algorithm for finding these region transformations is also given; it may be described as a transform space pruning algorithm.

The rest of this section gives an introduction to fractal image compression using a simple block-based example. For a thorough presentation of the theory of ITF's, the reader is referred to [11]. Section II describes a new approach for fractal image compression based on allowing a class of region transformations. Section III presents heuristic search algorithms for constructing these transformations. This correspondence is concluded with a discussion of the merits of this new approach and possible future work.

A. Block-Based Fractal Image Compression

This section will introduce fractal block coding using a simple example. The algorithm (called FIC-B) is a simplification of Jacquin's original algorithm [11]–[13].

The FIC-B algorithm partitions the image into two basic elements: range blocks and domain blocks. The range blocks form a tiling of the image and are 8×8 pixels (picture elements). The domain blocks are twice the size of the range blocks and overlap such that a new domain block starts every pixel. Determining the global transformation for

Manuscript received July 31, 1993; revised June 7, 1994. The associate editor coordinating the review of this paper and approving it for publication was Prof. Nasser M. Nasrabadi.

The authors are with the Department of Electrical and Electronic Engineering, University of Wales, Swansea, Wales, UK.

IEEE Log Number 9411130.