

Fractal Image Compression with Region-Based Functionality

Kamel Belloulata and Janusz Konrad, *Senior Member, IEEE*

Abstract—Region-based functionality offered by the MPEG-4 video compression standard is also appealing for still images, for example to permit object-based queries of a still-image database. A popular method for still-image compression is fractal coding. However, traditional fractal image coding uses rectangular range and domain blocks. Although new schemes have been proposed that merge small blocks into irregular shapes, the merging process does not, in general, produce semantically-meaningful regions. We propose a new approach to fractal image coding that permits region-based functionalities; images are coded region by region according to a previously-computed segmentation map. We use rectangular range and domain blocks, but divide boundary blocks into segments belonging to different regions. Since this prevents the use of standard dissimilarity measure, we propose a new measure adapted to segment shape. We propose two approaches: one in the spatial and one in the transform domain. While providing additional functionality, the proposed methods perform similarly to other tested methods in terms of PSNR but often result in images that are subjectively better. Due to the limited domain-block codebook size, the new methods are faster than other fractal coding methods tested. The results are very encouraging and show the potential of this approach for various internet and still-image database applications.

Index Terms—Still image coding, fractal image coding, region-based coding, region-based functionalities

I. INTRODUCTION

Region-based functionality has been extensively explored in the context of video sequences and today is one of the more appealing features of MPEG-4, the new video compression standard [1]. Although much less explored with respect to still images, we believe region-based functionality has its merits there as well, for example in object-based querying of still-image databases. This is especially appealing when applied to semantically-meaningful regions, that is regions that correspond to projections of 3-D objects. Since an automatic extraction of semantically-meaningful regions is computationally challenging, it is usually performed off-line, prior to the encoding, by means of sophisticated, CPU-intensive algorithms. Clearly, of interest are image compression methods that can encode regions independently of one another according to pre-computed segmentation maps, and subsequently preserve independence of the encoded regions during transmission. Algorithms that satisfy the above constraints belong to *region-*

based image coding methods that form an alternative to standard, block-based coding; instead of block-by-block treatment, an image is compressed and transmitted/stored region by region. The decoder is capable of receiving and interpreting each region's data (e.g., shape and texture) regardless of whether other regions are transmitted or not. Such an approach has two benefits. First, it permits new functionalities at the receiver, such as advanced object-based queries, selective region transmission (e.g., most interesting regions first), or object replacement. Some of such operations are possible in a block-based system if image segmentation is performed at the decoder. The advantage of region-based coding is that the segmentation is performed either at or prior to the encoding, where computing resources are often more abundant. Secondly, for a given rate, region-based coding often achieves better image quality than standard coding since region boundaries usually coincide with intensity edges that are difficult to encode.

A popular method for still-image compression is fractal coding. Traditional fractal image coding is essentially block-based [2]; the image domain is decomposed into square range and domain blocks and a contraction mapping is found that best maps domain blocks into range blocks [3]. This contraction mapping defines a fractal code of the image. Similar principle can be applied in the spectral domain, for example by means of the discrete cosine transform (DCT) applied to each range and domain block prior to finding the contraction mapping [4], [5]. Other hybrid schemes using wavelets have been recently proposed as well [6], [7], [8], [9]. Attempts have also been made to extend fractal coding beyond uniform square blocks in order to adapt the coding to local image characteristics and consequently increase coding performance. This has led to either square or rectangular non-uniform partition schemes such as quadtree [10], [11] and horizontal-vertical partitioning [12]; the usual coarse-to-fine approach starts from a maximum-size range block and performs recursive partitioning of blocks according to a quality metric. In fine-to-coarse quadtree partitioning [13] small blocks are merged into larger blocks within a quadtree. Alternatively, neighboring range blocks can be merged without the quadtree restriction; right-angled *irregularly-shaped* (IS) range blocks result [14], [15], [16], [17], [18]. Although the range, and consequently domain blocks, can take on a variety of shapes, this method creates partitions based on thresholding a simple local metric and cannot, in general, produce semantically-meaningful regions. In this sense, the method is texture-adaptive rather than region-based; semantically-meaningful regions are neither encoded nor transmitted independently of each other.

In this paper, we explore fractal image coding in the context

K. Belloulata is with Université de Moncton, Ecole de génie, Moncton, New Brunswick, Canada, E1A 3E9 (bellouk@umoncton.ca). J. Konrad is with Boston University, Department of Electrical and Computer Engineering, Boston MA 02215, USA (jkonrad@bu.edu).

This work was partially supported by the Natural Sciences and Engineering Research Council of Canada under grant RGPIN121619 and was performed by the authors while both were with INRS-Télécommunications, Montréal, Canada.

of region-based functionality. We propose two region-based fractal coding schemes implemented in spatial and transform domains, respectively. In both approaches regions are defined by a prior segmentation map, and are fractal-encoded independently of each other. A new dissimilarity measure is proposed that is limited to single-region pixels of the range block. Numerous experimental results are shown.

In Sections II and III, we introduce the notation and briefly review fractal image coding with regularly- and irregularly-shaped blocks. The proposed spatial- and transform-domain region-based fractal coding methods are described in Sections IV and V, respectively. In Section VI numerous experimental results are shown whereas in Section VII we draw conclusions.

II. FRACTAL IMAGE CODING WITH REGULARLY-SHAPED RANGE AND DOMAIN BLOCKS

Let $I(\mathbf{x})$ be image intensity of a pixel at position $\mathbf{x} = (x, y)$, and let $\{r_1, \dots, r_N\}$ be the set of N non-overlapping range blocks (i.e., collections of pixel coordinates) partitioning the image. Similarly, let $\{d_1, \dots, d_M\}$ be the set of M , possibly overlapping, domain blocks covering the image. Finally, let $I_{r_i} = \{I(\mathbf{x}) : \mathbf{x} \in r_i\}$ and $I_{d_j} = \{I(\mathbf{x}) : \mathbf{x} \in d_j\}$.

For each range block r_i the goal is to find a domain block d_j and a contractive mapping w_i that jointly minimize a *dissimilarity* (distortion) criterion ε . The contractive affine mapping w_i consists of three sub-mappings:

1. *contraction* $\sigma(I, \mathbf{x})$: usually preceded by lowpass anti-alias filtering; e.g., K -fold contraction with four-neighbor averaging: $I(\mathbf{x}) \stackrel{\sigma(I, \mathbf{x})}{\longleftarrow} \frac{1}{4} \sum_{\mathbf{y} \in \eta(K\mathbf{x})} I(\mathbf{y})$, where η is the first-order neighborhood (N-E-W-S),
2. *photometric transformation* to account for different dynamic ranges of pixels in the range and domain blocks: $\gamma(I) \triangleq sI + o$, where s is a scaling factor (gain) and o is an offset,
3. *geometric transformation* ξ (inverse mapping: range \rightarrow domain): $\xi(\mathbf{x}) \triangleq \mathbf{A}\mathbf{x} + \mathbf{b}$, where $\mathbf{x} \in r_i$, \mathbf{A} is a 2×2 matrix and \mathbf{b} is a translation vector (this mapping must be 1-to-1 between pixels of the range and domain blocks).

The overall transformation w_i that maps a domain-block pixel into the range-block pixel at \mathbf{x} is:

$$w_i(I, \mathbf{x}) \triangleq \gamma \circ \sigma(I, \xi(\mathbf{x})), \quad \mathbf{x} \in r_i, \quad (1)$$

where \circ is the composition operator. The above general expression can be simplified by constraining the transformation \mathbf{A} to eight cases: 4 rotations ($0^\circ, 90^\circ, -90^\circ, 180^\circ$) and 4 mirror reflections (mid-horizontal, mid-vertical, first diagonal, second diagonal) [3], [19]. We denote the set of possible transformations \mathbf{A} by $\{\zeta^p\}_{p=1}^8$. Furthermore, by expressing \mathbf{b}_i implicitly as the index of the domain block, i.e., j in d_j , we can write w_i as follows:

$$w_i(I_{d_j}, \mathbf{x}) \triangleq s_i \cdot \sigma(I_{d_j}, \zeta_i^p(\mathbf{x})) + o_i, \quad \mathbf{x} \in r_i, p \in \{1, \dots, 8\}. \quad (2)$$

In order to encode range block r_i , a search for index j (domain block d_j) and for an isometry ζ_i^p must be executed, jointly with the computation of photometric parameters s_i and o_i . This can be performed by minimizing the following mean-squared error

$$\varepsilon(I_{r_i}, I_{d_j}, w_i) = \frac{1}{|r_i|} \sum_{\mathbf{x} \in r_i} [I_{r_i}(\mathbf{x}) - w_i(I_{d_j}, \mathbf{x})]^2, \quad (3)$$

where $|r_i| = \text{Card}(r_i)$. While the isometry ζ_i^p and index j (equivalent to translation \mathbf{b}) are usually found by exhaustive search, the scaling s_i and offset o_i are computed as follows

$$s_i = \frac{\sum_{\mathbf{x} \in r_i} [\sigma(I_{d_j}, \zeta_i^p(\mathbf{x})) - m_{d_j}] [I_{r_i}(\mathbf{x}) - m_{r_i}]}{\sum_{\mathbf{x} \in r_i} [\sigma(I_{d_j}, \zeta_i^p(\mathbf{x})) - m_{d_j}]^2}, \quad (4)$$

$$o_i = m_{r_i} - s_i \cdot m_{d_j}, \quad (5)$$

where m_{r_i} and m_{d_j} are the mean intensity values in the range and domain blocks, respectively. Instead of transmitting the photometric offset o_i (in addition to j , ζ_i^p and s_i), mean value m_{r_i} of the range block r_i can be transmitted. This permits a precise representation of the mean local intensity but to assure convergence at the decoder, without a constraint on the intensity scaling coefficients [10], requires a modification of the photometric transformation. This can be considered as orthogonalization with respect to the constant blocks and has been treated in detail in [20].

In DCT-based methods, first all range and domain blocks are transformed *via* DCT: $\hat{I}_{r_i} = \text{DCT}(I_{r_i})$ and $\hat{I}_{d_j} = \text{DCT}(I_{d_j})$. Let $\mathbf{u} = (u, v)$ be a 2-D frequency of a DCT coefficient and let ζ_i^p be one of the 8 isometries discussed before but adapted to the DCT coefficient domain [21], [4]. The contraction mapping σ must be redefined as well to account for properties of the frequency \mathbf{u} . Rather than performing spectral subsampling, the mapping σ can be, for example, defined as follows: a $K \cdot Q \times K \cdot Q$ domain block \hat{I}_{d_j} is mapped onto a $Q \times Q$ range block \hat{I}_{r_i} while retaining only the low-frequency part of the spectrum of \hat{I}_{d_j} [22]. The local affine transformation w_i is then computed by minimizing the following dissimilarity measure

$$\varepsilon(\hat{I}_{r_i}, \hat{I}_{d_j}, w_i) = \frac{1}{|v_{r_i}|} \sum_{\mathbf{u} \in v_{r_i}, \mathbf{u} \neq \mathbf{0}} [\hat{I}_{r_i}(\mathbf{u}) - w_i(\hat{I}_{d_j}, \mathbf{u})]^2, \quad (6)$$

where the summation is performed over all frequencies in the DCT-transformed range block except $\mathbf{u} = \mathbf{0}$. Above, v_{r_i} denotes the set of frequencies (after DCT) for block r_i ; v_{r_i} spatially coincides with r_i but its members are discrete frequencies ($|v_{r_i}| = \text{Card}(v_{r_i})$). The exclusion of $\mathbf{u} = \mathbf{0}$ is due to the fact that usually the mean intensity of the range block r_i (equal to $\hat{I}_{r_i}(\mathbf{0})$) is transmitted instead of the offset o_i .

For the simplified transformation (2), the domain-block index j , the isometry ζ_i^p and the scaling factor s_i are found by minimizing ε above. Thus, to encode the block r_i , for each combination of j and ζ_i^p , the scaling factor is computed as follows:

$$s_i = \frac{\sum_{\mathbf{u} \in v_{r_i}, \mathbf{u} \neq \mathbf{0}} \sigma(\hat{I}_{d_j}, \zeta_i^p(\mathbf{u})) \hat{I}_{r_i}(\mathbf{u})}{\sum_{\mathbf{u} \in v_{r_i}, \mathbf{u} \neq \mathbf{0}} [\sigma(\hat{I}_{d_j}, \zeta_i^p(\mathbf{u}))]^2},$$

and the best combination of the three parameters is selected, i.e., the one that gives the lowest ε .

III. FRACTAL IMAGE CODING WITH IRREGULARLY-SHAPED RANGE AND DOMAIN BLOCKS

The basic unit on which a typical fractal coder operates is a rectangular block, whether range or domain (Fig. III(a)). In order to allow operation on non-rectangular range blocks, merging of blocks into composite range blocks has been proposed in the past [23], [14]. First, one of the range blocks is chosen as a seed and a suitable domain-to-range block transformation is found. Then, four neighboring range blocks nearest from the seed as well as four neighboring domain blocks nearest from the domain of the seed are examined (Fig. III(b)). The seed transform is extended to cover the seed and one of its neighbors, and a threshold on the dissimilarity measure determines which extensions are acceptable. If a particular extension is accepted, the transformation remains extended and the neighbor block is merged with the seed. The process is recursively applied to all neighbors of the extended seed transformation. When the seed transformation has been enlarged as much as possible then a new seed is used from the next uncoded range block and the process is repeated. The goal is thus to map as large groups of contiguous domain blocks as possible into groups of contiguous range blocks, all with the same transformation (Fig. III(c)). If the transformation used is an isometry, then the shapes of the range and domain composite blocks are identical.

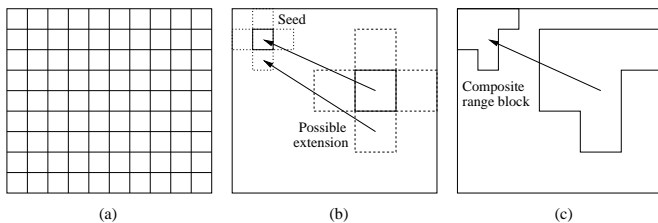


Fig. 1. Illustration of fractal image coding with irregularly-shaped range and domain blocks [14]: (a) regular partitioning (atomic blocks), (b) adding an extension to the seed transformation, (c) irregularly-shaped range and domain blocks.

To find optimal transformations (irregularly-shaped range blocks and corresponding transformations) for a given distortion, an exhaustive search of the space of blocks needs to be performed. Since this is hardly practical, suboptimal but computationally-efficient approaches have been proposed, such as heuristic algorithm [14], evolutionary algorithm [15], [16] and deterministic search [17].

Note, that the described method creates image partitioning during encoding and does not use any prior partitioning, unlike in MPEG-4. In this respect, the method is texture-adaptive and not region-based. The computed irregular partitions are not semantically-meaningful, although, coincidentally, some range-block boundaries may coincide with semantic-region boundaries.

IV. REGION-BASED FRACTAL IMAGE CODING IN THE SPATIAL DOMAIN

In a typical fractal image coder (Section II), range blocks are defined independently of image content and may overlap two or more objects (regions) with quite different intensity characteristics (Fig. IV-A); the search for a good domain-range correspondence may be difficult. This difficulty is also present,

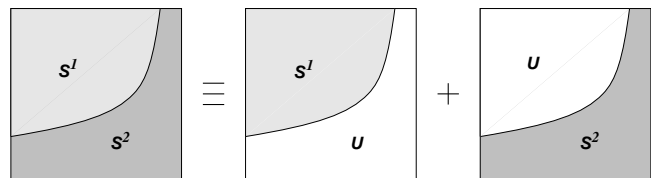


Fig. 2. Block with two segments S^1 and S^2 , belonging to different regions, and its decomposition into two blocks with defined pixels (grey), either in S^1 or in S^2 , and undefined pixels in U (white).

although to a lesser degree, in texture-adaptive fractal coding presented in Section III. To alleviate this problem, we propose a new approach to fractal image coding that accounts for *prior* image segmentation so as to encode regions independently of each other. This allows independent transmission/storage and, therefore, decoding of individual regions, thus permitting new functionalities.

A. Proposed approach

Segmentation map associates a label with each image point; same-label points form a region. We propose to use square range and domain blocks as in the standard case, but to restrict the distortion measure ε (3) to a subset of pixel locations in the range block that are associated with one region only. We call this subset a segment; an example of block with two segments is shown in Fig. IV-A).

Let $S_{d_j}^n$ be the n -th segment in the domain block d_j (a block may consist of more than 2 segments). Similarly, let $S_{r_i}^m$ be the m -th segment in the range block r_i . Let \tilde{I}_{d_j} be a padded (extrapolated) version of I_{d_j} defined as follows:

$$\tilde{I}_{d_j}(\mathbf{x}) = \begin{cases} I_{d_j}(\mathbf{x}) & \text{if } \mathbf{x} \in S_{d_j}^n, \\ v & \text{otherwise,} \end{cases} \quad (7)$$

where v is a padding value, typically zero, mean local intensity in $S_{d_j}^n$ or intensity value at the nearest pixel within $S_{d_j}^n$. We define a new distortion measure ε as follows:

$$\varepsilon(I_{r_i}, \tilde{I}_{d_j}, w_i^m) = \frac{1}{|S_{r_i}^m|} \sum_{\mathbf{x} \in S_{r_i}^m} [I_{r_i}(\mathbf{x}) - w_i^m(\tilde{I}_{d_j}, \mathbf{x})]^2, \quad (8)$$

where w_i^m denotes an affine transformation for segment $S_{r_i}^m$. Note, that compared to (3), the above distortion is evaluated only at pixel positions within a single range block segment $S_{r_i}^m$. Clearly, when the shapes of $S_{r_i}^m$ and of the contracted domain segment $S_{d_j}^n$ do not match, an extrapolation of some pixels in the domain block is required, hence the padding (7). Also note, that when the range and domain blocks examined happen to both lie in the interior of the region being encoded, $S_{r_i}^m$ and $S_{d_j}^n$ cover the whole range and domain blocks, respectively, and the distortion (8) simplifies to the standard case (3).

Similarly to the standard fractal coding, for every range block segment $S_{r_i}^m$ we search for a domain block segment $S_{d_j}^n$ and a mapping w_i^m that jointly minimize ε . The domain-block index j and the isometry ζ_i^p are found by exhaustive search, whereas the parameters s_i and o_i are computed similarly to (4) and (5), respectively, but with the summations restricted to $S_{r_i}^m$. The image is reconstructed at the decoder by iterative application

of the mapping w_i^m restricted to $\mathcal{S}_{r_i}^m$. A proof of convergence of this algorithm has been given by Mansouri [24] and can be found in the Appendix.

B. Choice of search space

The computational complexity of encoding an image using the proposed method is directly related to the size of search space over which the distortion (8) is minimized; the number of permissible domain blocks plays the dominant role. The most demanding case is when *each* segment of *every* domain block of the image is considered; the domain-block codebook is built from the whole image. This exhaustive procedure is theoretically optimal but extremely involved computationally. Moreover, it does not allow for independent decoding of regions.

In order to assure region-by-region encoding/decoding both range and domain blocks must be located within the same region \mathcal{R} . Therefore, while minimizing the distortion (8), four cases regarding the location of blocks r_i and d_j with respect to the region \mathcal{R} may arise:

1. r_i and d_j are both interior blocks (*int/int*),
2. r_i and d_j are both boundary blocks (*bnd/bnd*),
3. r_i is an interior block whereas d_j is a boundary block (*int/bnd*),
4. r_i is a boundary block whereas d_j is an interior block (*bnd/int*).

While in the first case standard full-block search is executed among region's interior blocks (3), in the second case only partial matching is performed using dissimilarity measure (8) as both range and domain blocks are on region's boundary. Clearly, intensity padding, to be further discussed in Section IV-C, is needed when contracted $\mathcal{S}_{d_j}^n$ does not fully enclose $\mathcal{S}_{r_i}^m$. In the third case, intensity extrapolation (7) of the domain block is always needed since r_i is an interior block whereas d_j is a boundary block. Although usually a better solution than using intensities from a neighboring region (standard fractal coding), the padding is likely to result in sub-optimal intensities (padding inaccuracy) as compared with domain blocks fully-enclosed in a region. In the fourth case, no padding is needed; r_i is a boundary block while d_j is an interior block. Although a feasible scenario, this case requires a costly examination of all interior domain blocks for each boundary range block; typically there are many more interior than boundary blocks.

In the most complex scenario that assures region-by-region functionality, a 1+2+3+4 search is performed (all 4 cases considered). We expect that the third (*int/bnd*) and the fourth (*bnd/int*) case will contribute little to the PSNR performance of the compression. By skipping the third case, a 1+2+4 search can be performed with the additional benefit of reduced complexity. Similarly, by skipping the fourth case a 1+2+3 search can reduce the complexity by not considering interior domain blocks for each boundary range block. By skipping both cases, a 1+2 search results in additional computational savings by mapping boundary blocks only among themselves and mapping interior blocks onto interior blocks only. A schematic representation of the 1+2 search is shown in Fig. IV-B.

To evaluate the performance/complexity tradeoff with respect to various search scenarios, we have encoded independently the foregrounds and the backgrounds of four MPEG-4 test images (see Section VI for details). We have tested four search scenarios: 1+2+3+4, 1+2+3, 1+2+4 and 1+2, all with LPE padding (Section IV-C), but we are presenting results only for the most (1+2+3+4) and least (1+2) complex cases. Table IV-B shows PSNR, rate per pixel and CPU time (for a 360MHz Ultra-5 Sun workstation) for the foregrounds and backgrounds, respectively, of the four test images. Note the reduction of computation time by 25-51% while suffering only 0.04-0.46dB performance penalty. The complexity and performance of the 1+2+3 and 1+2+4 search scenarios stayed between those of the 1+2+3+4 and 1+2 scenarios. Clearly, the 1+2 search (*int/int*, *bnd/bnd*) is attractive computationally while at the same time achieves very good performance. In the remainder of this paper we will use the 1+2 search only.

C. Choice of padding for boundary domain blocks

In the 1+2 search scenario (*int/int* + *bnd/bnd*) selected for further experiments, boundary domain blocks need, in general, to be extrapolated (7). This situation arises when the (contracted) domain-block segment $\mathcal{S}_{d_j}^m$ does not fully enclose the range-block segment $\mathcal{S}_{r_i}^m$ under consideration.

We examine two padding schemes adopted in MPEG-4: lowpass extrapolation (LPE) padding used in the intra-frame mode and zero-value (ZERO) padding used in the inter-frame mode. In the MPEG-4 LPE padding, the unknown pixels of the domain block (\mathcal{U} in Fig. IV-A) are filled with the average value of pixels within the segment $\mathcal{S}_{d_j}^n$:

$$m_{d_j} = \frac{1}{|\mathcal{S}_{d_j}^n|} \sum_{\mathbf{x} \in \mathcal{S}_{d_j}^n} I_{d_j}(\mathbf{x}), \quad (9)$$

and then are lowpass filtered. We do not perform the lowpass filtering explicitly, but we rely on the lowpass properties of the spatial averaging executed within the contraction operator σ . The ZERO padding was proposed in MPEG-4 for zero-mean blocks such as those resulting from inter-frame motion compensation. We examine this padding mode, as a potential alternative to the LPE padding, due to its lower computational complexity.

In Table IV-B the two rightmost columns compare LPE and ZERO padding for the 1+2 search in terms of PSNR, bit rate and CPU time. Note that while a loss of performance (0.01 to 0.57dB) was expected, a minimal reduction in CPU time is a bit surprising; padding seems to contribute minimally to the computational complexity. Since the LPE padding performs better with insignificant CPU penalty, it will be used in the remainder of this paper.

D. Encoding of parameters

The following parameters of each range block segment $\mathcal{S}_{r_i}^m$ need to be encoded for subsequent transmission or storage: photometric gain s_i and mean intensity m_{r_i} , translation vector \mathbf{b} (expressed in terms of the relative position of $\mathcal{S}_{d_j}^n$ with

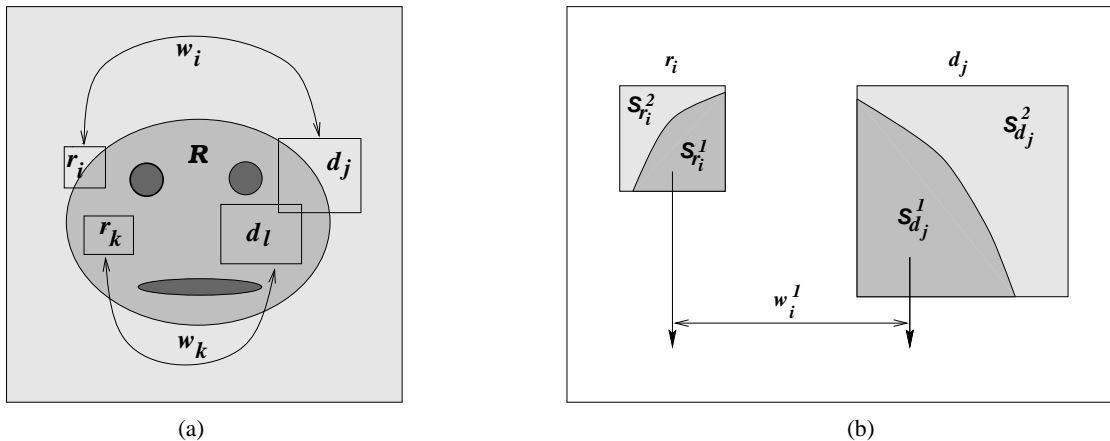


Fig. 3. Schematic illustration of the proposed region-based spatial-domain fractal coding scheme. (a) Interior range blocks are matched against interior domain blocks of the same region and boundary range blocks are partially matched (8) against same-region domain blocks. (b) Segments of boundary blocks are processed independently: segment $S_{r_i}^1$ is mapped onto the domain block d_j by affine transformation w_i^1 , while segment $S_{r_i}^2$ of the same range block r_i may be mapped onto a segment of another domain block by a different transformation.

TABLE I

PERFORMANCE COMPARISON OF REGION-BY-REGION SPATIAL-DOMAIN FRACTAL CODING IN TWO SEARCH SCENARIOS (1+2+3+4 AND 1+2) AND TWO PADDING SCENARIOS (LPE AND ZERO). SEE TEXT FOR DETAILS.

	1+2+3+4 search (LPE)			1+2 search (LPE)			1+2 search (ZERO)		
	PSNR [dB]	Rate [bpp]	CPU [s]	PSNR [dB]	Rate [bpp]	CPU [s]	PSNR [dB]	Rate [bpp]	CPU [s]
Foregrounds									
<i>Cyclamen</i>	32.88	0.30	128	32.42	0.29	62	32.11	0.28	61
<i>News</i>	30.47	0.25	108	30.10	0.24	56	30.00	0.24	55
<i>Foreman</i>	39.11	0.23	96	39.00	0.23	61	38.86	0.23	61
<i>Children</i>	28.34	0.13	32	27.92	0.13	17	27.88	0.13	16
Backgrounds									
<i>Cyclamen</i>	40.28	0.40	208	40.07	0.40	110	39.50	0.40	114
<i>News</i>	33.66	0.40	316	33.38	0.40	205	33.22	0.40	204
<i>Foreman</i>	29.40	0.40	277	29.36	0.39	208	29.35	0.38	206
<i>Children</i>	31.88	0.51	480	31.70	0.51	352	31.67	0.51	349

respect to $S_{r_i}^m$, and affine transformation A reduced to 4 rotations and 4 mirror reflections. Since s_i and m_{r_i} have, in general, non-uniform distributions, entropy coding usually proves beneficial. We chose 5-bit quantization of s_i and 7-bit quantization of m_{r_i} , reported in the literature to give good performance [10], followed by Huffman coding. To ensure that both the encoder and the decoder use the same s_i and m_{r_i} values, we quantize both during the minimization of the dissimilarity measure (3), i.e., prior to each evaluation of ε . We encode the vector b as a relative position of $S_{d_j}^n$ with respect to $S_{r_i}^m$ using fixed-length codewords (determined by image size). We use 3-bit codewords for the 8 possible rotations/reflections.

The segmentation maps also need to be transmitted in order that during the decoding suitable segments be cut out from the domain blocks and mapping w_i^m be properly applied. A discussion of the segmentation map encoding is beyond the scope of this paper; the reader is referred to recent literature (e.g.,

[25]). We note, however, that state-of-the-art lossless methods allow transmission of typical segmentation maps (MPEG-4 test sequences) at about 0.01-0.03bpp [26] with lossy intra-frame methods reducing this rate even further.

V. REGION-BASED FRACTAL IMAGE CODING IN THE TRANSFORM DOMAIN

In DCT-based fractal coding (Section II), boundary range blocks contain pixels from two or more objects. Thus, similarly to the spatial-domain case, independent decoding of objects is not possible. Also, the coding quality may suffer since pixels on different sides of the boundary may have different characteristics; by applying the standard DCT to such a block, spectral properties of these pixels are mixed up making the search for a good range-domain correspondence unreliable. In particular, a sharp intensity transition may cause significant spectral oscillations.

A. Proposed approach

In order to alleviate the above deficiencies, we propose to apply shape-adaptive DCT (SA-DCT) [27] to each segment S of the boundary range and domain blocks. The basic concept of the SA-DCT is to perform vertical 1-D DCTs on the defined pixels first (Fig. IV-A), and then to apply horizontal DCTs to the vertical DCT coefficients with the same vertical frequency index. Fig. V-A illustrates this idea. The final coefficients of the SA-DCT are located in the upper-left corner of each block. The number of the SA-DCT coefficients is identical to the number of defined pixels. Since the shape of each segment is transmitted, the decoder can perform the inverse SA-DCT. The most important benefit of SA-DCT is its capability to adapt to arbitrarily-shaped regions; the method simplifies to the standard DCT for rectangular segments.

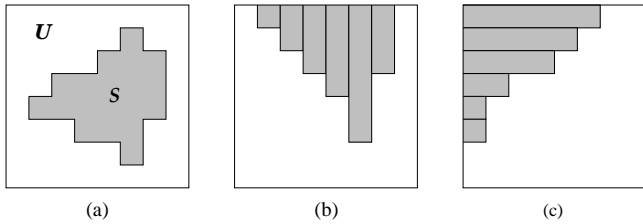


Fig. 4. Illustration of SA-DCT: (a) arbitrarily-shaped region; (b) vertical alignment followed by vertical 1-D DCTs; (c) horizontal alignment followed by horizontal 1-D DCTs.

Due to its good performance, the SA-DCT algorithm has become a common tool for coding of arbitrarily-shaped image regions and, in particular, has been incorporated into MPEG-4. We use a variant of SA-DCT, called Δ DC-SA-DCT [28]. It improves the performance of the SA-DCT by means of two additional processing steps: extraction of the DC component from the segment S before performing forward SA-DCT and Δ DC correction carried out during the inverse SA-DCT.

Similarly to the spatial-domain coding (Section IV) we limit the search space of domain blocks to allow region-based functionality: either each interior range block is matched against interior domain blocks of the same object (standard fractal/DCT coding of Section II applies), or each boundary range block is matched against domain blocks from the boundary of the same object. We process each segment of the range block r_i independently. First, we apply the SA-DCT to each segment $S_{r_i}^m$ of the block r_i (Fig. V-A) and then we find parameters of the transformation w_i^m for each such segment.

Recall that \hat{I}_{r_i} and \hat{I}_{d_j} are DCT-transformed blocks of intensities I_{r_i} and I_{d_j} . Let $\mathcal{P}_{r_i}^m$ (Fig. V-A) be the segment $S_{r_i}^m$ after SA-DCT. Note that the shape of $\mathcal{P}_{r_i}^m$ is different from that of $S_{r_i}^m$ due to the executed vertical and horizontal shifts, but that the number of defined pixels is unchanged. Also, let $\hat{I}_{r_i}^m(\mathbf{u})$ be an SA-DCT coefficient in $\mathcal{P}_{r_i}^m$ at frequency \mathbf{u} . To find the best domain block d_j for a given range block r_i , we propose the following distortion measure in the transform domain:

$$\varepsilon = \frac{1}{|\mathcal{P}_{r_i}^m|} \sum_{\mathbf{u} \in \mathcal{P}_{r_i}^m, \mathbf{u} \neq \mathbf{0}} [\hat{I}_{r_i}^m(\mathbf{u}) - w_i^m(\tilde{I}_{d_j}^n(\mathbf{u}))]^2, \quad (10)$$

where $\tilde{I}_{d_j}^n$ is an extrapolated n -th segment of the SA-DCT-

transformed domain-block intensity:

$$\tilde{I}_{d_j}^n(\mathbf{u}) = \begin{cases} \hat{I}_{d_j}^n(\mathbf{u}) & \text{if } \mathbf{u} \in \mathcal{P}_{d_j}^n, \\ v & \text{otherwise.} \end{cases} \quad (11)$$

Although various v values could be used, the to-be-padded coefficients are at higher frequencies and therefore a logical choice, that we adopt here, is to set v to zero (ZERO padding). Clearly, contributions to ε are only made at indices within

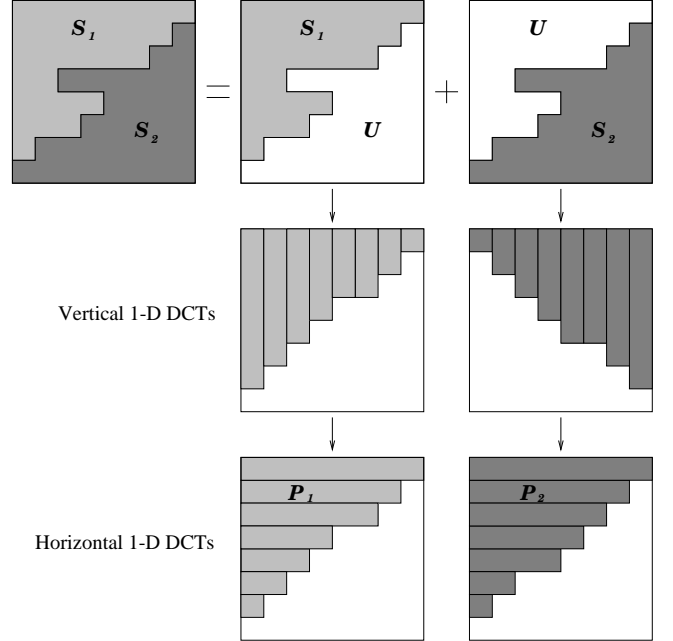


Fig. 6. Example of application of the SA-DCT to a boundary block with two segments. Note that due to horizontal and vertical shifts the shape \mathcal{P}_i is different from that of S_i .

range-block segment $\mathcal{P}_{r_i}^m$. If at frequency \mathbf{u} , a coefficient in $\mathcal{P}_{r_i}^m$ exists while it does not exist in the segment $\mathcal{P}_{d_j}^n$ of the domain block under consideration, then a significant contribution to ε is made. Thus, good matches should be found between similarly-textured range and domain segments that have either similar shapes or little spectral content at range coefficients with no domain-segment correspondence, either an acceptable solution. In the case a domain-block segment $\mathcal{P}_{d_j}^n$ fully encloses $\mathcal{P}_{r_i}^m$, it is likely to be accepted if same-index coefficients are similar in $\mathcal{P}_{d_j}^n$ and $\mathcal{P}_{r_i}^m$. Unfortunately, if the segments' shapes or sizes are very different, the basis functions for $\mathcal{P}_{d_j}^n$ and $\mathcal{P}_{r_i}^m$ may differ substantially; despite similar coefficient values, the spatial-domain intensity patterns may be quite different in both cases thus degrading performance. This ambiguity, as we will see later, prevents the SA-DCT-based method from attaining significantly higher performance than that of the proposed spatial method (Section IV).

B. Choice of search space

As shown in Fig. V-A the search for parameters of the transformation w_i is executed independently for each segment of a range block. However, a significant reduction of computational complexity is possible if common domain-block position j (\mathbf{b})

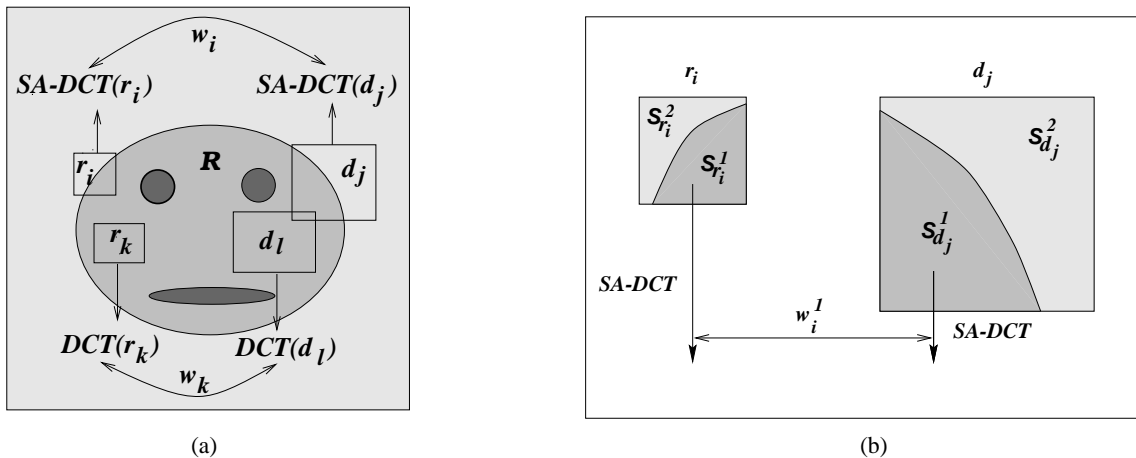


Fig. 5. Schematic illustration of the proposed region-based transform-domain fractal coding scheme. (a) Interior range blocks are matched against interior domain blocks of the same region and boundary range blocks are partially matched (10) against same-region domain blocks. (b) Segments of boundary blocks are processed independently: the segment $S_{r_i}^1$ is mapped onto a segment of the domain block d_j by affine transformation w_i^1 , while the segment $S_{r_i}^2$ of the same range block r_i may be mapped onto a segment of another domain block by a different transformation. Interior blocks are transformed by DCT while boundary-block segments are transformed using SA-DCT.

and isometry ζ_i^p are used for all the segments, but s_i and m_{r_i} are computed independently for each segment in order to allow precise matching of textures. We have experimented with this approach and we have observed a performance degradation of less than 0.1dB while reaching 20-30% reduction of the CPU coding time. Some results using this approach have been presented by us earlier [29]. For the remainder of this paper, however, we will use the independent search for all segments of a range block.

C. Encoding of parameters

The parameters s_i , m_{r_i} , \mathbf{b} , and \mathbf{A} of each segment are encoded exactly in the same fashion as in the spatial-domain approach (Section IV-D).

VI. EXPERIMENTAL RESULTS

We have compared the proposed region-based fractal coding methods with other algorithms on a set of still images extracted from MPEG-4 test sequences *Cyclamen* (SIF), *News*¹ (CIF), *Foreman* (CIF) and *Children* (CIF). The segmentation maps used are those provided within MPEG-4 except for *Foreman* which we segmented manually. Only the luminance component of each image was used. Note that out of the six compression methods compared only the methods proposed in this article permit region-based functionality:

1. standard fractal coding in the spatial domain (Section II) – *Fractal/Spatial*,
2. hybrid fractal/DCT coding (Section II) – *Fractal/DCT*,
3. shape-adaptive fractal coding in the spatial domain (proposed in Section IV) – *SA-Fractal/Spatial*,
4. shape-adaptive hybrid fractal/SA-DCT coding in the transform domain (proposed in Section V) – *Fractal/SA-DCT*,

¹The original MPEG-4 sequence *News* consists of 4 objects (news anchors, “MPEG-4 WORLD” graphics, screen and background). We have combined the original background with the screen and the graphics into a new background, while leaving the two news anchors as the foreground object.

5. evolutionary fractal coding [16] (Section III) – *Evolutionary*,
6. JPEG compression standard - *JPEG*.

Identical coding parameters were used in the first four fractal coders in all experiments, including the same quantizer and Huffman coding of the transformation parameters. Based on our prior research [8], we have applied several improvements to the first four fractal coders listed above. In order to improve the compression performance all non-boundary range and domain blocks (full blocks) are classified according to their intensity variance. This classification permits the parameters of range blocks from different classes to be processed differently. For range blocks with extremely small variance, only the mean value m_{r_i} is encoded, whereas the other transformation parameters are skipped, thus increasing the compression ratio with no quality penalty [30]. Range blocks with small variance are encoded by transmitting all parameters of transformation w_i but the search is executed only among domain blocks with small variance. Similarly, for range blocks with large variance the search is performed only on domain blocks with large variance. Another improvement that we have applied is quadtree partitioning. We have applied quadtree partitioning to all range blocks in the standard fractal coding methods (*Fractal/Spatial* and *Fractal/DCT*), and only to the non-boundary (full) blocks in the shape-adaptive methods (*SA-Fractal/Spatial* and *Fractal/SA-DCT*). Starting with 8×8 range blocks we have divided them into four 4×4 blocks whenever intensity variance exceeded certain threshold [11]. The resulting range blocks were always matched against twice larger domain blocks. The block splitting based on intensity variance incurs a small performance penalty as compared with the splitting based on dissimilarity measure ε (3) [10], but is much more efficient computationally.

The evolutionary coding [16] was performed using the coder freely available on the internet [31]. While the common set of parameters for all the images was: $-e$ 4 $-s$ 5 $-o$ 7, the remaining parameters ($-b$ $-r$) were: (3, 600) for *Cyclamen*, (4,

720) for *Foreman*, (4, 750) for *News*, and (3, 300) for *Children*. We used the JPEG coder as implemented in the *xv* program (version 3.10a). The experiments were executed on a 360MHz Sun Ultra-5 workstation.

A. Encoding of objects over uniform background

High-contrast objects are difficult to encode due to sharp intensity transitions at object boundaries. Since the available MPEG-4 test sequences have few high-contrast objects, we created them synthetically by replacing the background with uniform intensity. We considered two background intensities: mean intensity of the object and uniformly black color (dramatic object/background transition). In both experiments no quadtree partitioning was applied. Table VI-A shows the distortion (PSNR), bit rate and execution times for all four foreground objects with mean-intensity background. The PSNR is computed over complete images and reflects the performance of each algorithm also within the background close to object boundary.

It is clear from the table that the proposed shape-adaptive methods both in the spatial and transform domains perform similarly or better than their non-shape-adaptive counterparts (-0.10 to 1.58dB gain). In the case of zero-valued backgrounds (not shown in the table), the improvement was more dramatic (0.92 to 6.60dB gain). However, such case of extremely high contrast usually characterizes only small areas in an image and thus has small impact on the overall gain. Note also a significant reduction of the CPU time for some images (*News* and *Foreman*). This is not surprising since the search space is reduced to the *int/int-bnd/bnd* case, while the overhead due to the SA-DCT is minimal due to 1-D DFTs applied to relatively small blocks.

In comparison with the evolutionary algorithm, the proposed methods achieve lower PSNR for all images except *Foreman* for which they show up to 3.39dB gain. Comparing to JPEG the new methods have lower PSNR for *Cyclamen* and *News* (up to -1.52dB) but higher for *Foreman* and *Children* (up to 1.80dB). For zero-valued background the proposed methods performed, as expected, much better than the other 4 schemes. Also, note a significant reduction in coding time for our methods in comparison to the evolutionary algorithm.

In order to visually compare the performance of the proposed algorithms with their non-shape-adaptive counterparts, in Fig. VI-A we show the encoded image *News*. Note that all four algorithms do not use quadtree partitioning. Visually, the proposed algorithms outperform the standard non-shape-adaptive fractal methods, although this is not evidenced by PSNR values. The reason for this is that the concentrated improvement around the "MPEG-4 WORLD" graphics is compensated for by numerous smaller errors throughout the objects. The relatively high PSNR values, despite significant distortions present within the objects, are due to the uniform background that is encoded with very little distortion.

B. Encoding of complete images

To verify the performance of the proposed algorithms in a more realistic scenario, we performed a rate-distortion comparison on complete images, i.e., with textured background.

Considering, as before, the *Fractal/Spatial* algorithm as representative of non-shape-adaptive algorithms (Table VI-A), first we have encoded all four test images as complete frames (no region-based functionality). Then, we have encoded the same images region-by-region using the *SA-Fractal/Spatial* algorithm. In both tests, the same quadtree partitioning as the one described in Section VI-A was used. The rate-distortion curves are shown in Fig. VI-B. Note that the new algorithm performs slightly better for *Cyclamen* and *News* for the whole range of bit rates, while it is slightly worse for *Foreman* and *Children*. Overall, based on PSNR values, it is fair to say that our new algorithms perform similarly to standard fractal coding algorithms. Note that the rate for shape information is not accounted for in Fig. VI-B. However, as we have mentioned before (Section IV-D), and as has been pointed out elsewhere in the literature [27], the shape information rate has negligible impact on the performance since modern compression methods can encode object boundaries at the rate of about 0.01-0.03bpp [26].

A visual confirmation of the above claims can be found in Figs. VII and VII where results for encoding *News* and *Foreman* at 0.72bpp are shown. We have selected coding parameters so as to assure better quality for the foreground than for the background. Note the much less distorted area around the "MPEG-4 WORLD" graphics in *News* for the region-based algorithm, which is reflected in an almost 3dB PSNR gain. The new algorithm is slightly outperformed in PSNR by the *Fractal/Spatial* algorithm for *Foreman*, but the visual differences are insignificant. Recall that the *SA-Fractal/Spatial* algorithm allows image decomposition at the decoder with no need for image analysis, as is demonstrated in Figs. VII(c-d) and VII(c-d). Moreover, all this is done at a much lower computational complexity. While the *Fractal/Spatial* algorithm needed 790 and 757 seconds to encode *News* and *Foreman*, respectively, the *SA-Fractal/Spatial* algorithm needed only 322 and 328 seconds. At no, or very little, performance penalty the proposed algorithms add new functionality to the encoding of still images and do so at a lower computational cost.

VII. CONCLUSION

We have introduced fractal image coding with region-based functionality. While retaining square range and domain blocks we have modified the dissimilarity measure to account for pixels belonging to one region only. Accordingly, we have restricted the search for matching domain blocks to the region of interest. Consequently, the decoder can resolve a region without reference to the information about other regions of the image. This permits interesting operations at the decoder, such as object-based queries, selective object transmission or even object replacement. The proposed approach differs from other region-based fractal coding methods proposed recently [16], [18] in that it does not perform image segmentation during encoding but accounts for a prior segmentation. Since an automatic computation of semantically-meaningful regions is extremely difficult, our approach, by exploiting a prior segmentation, can delegate the segmentation to sophisticated off-line algorithms. As was demonstrated, the proposed algorithms per-

TABLE II

COMPARISON OF CODING RESULTS FOR FOREGROUND OF THE FOUR TEST IMAGES WITH THE BACKGROUND SET TO MEAN INTENSITY OF THE FOREGROUND.

	<i>Cyclamen</i>			<i>News</i>			<i>Foreman</i>			<i>Children</i>		
	PSNR [dB]	Rate [bpp]	CPU [s]	PSNR [dB]	Rate [bpp]	CPU [s]	PSNR [dB]	Rate [bpp]	CPU [s]	PSNR [dB]	Rate [bpp]	CPU [s]
<i>Fractal/Spatial</i>	32.40	0.32	118	30.15	0.25	108	37.65	0.23	96	28.10	0.15	33
<i>SA-Fractal/Spatial</i>	32.42	0.33	109	30.10	0.26	70	39.00	0.24	65	28.00	0.15	28
<i>Fractal/DCT</i>	32.52	0.33	120	30.10	0.24	107	37.82	0.23	98	28.00	0.14	35
<i>Fractal/SA-DCT</i>	32.75	0.33	110	30.31	0.26	78	39.40	0.24	70	28.17	0.15	36
<i>Evolutionary</i>	33.53	0.33	187	31.02	0.26	140	36.01	0.24	204	28.41	0.15	256
<i>JPEG</i>	33.94	0.33	3	31.61	0.26	3	37.60	0.24	3	26.52	0.15	2

(a) *Fractal/Spatial* (30.15dB at 0.25bpp)(c) *Fractal/DCT* (30.10dB at 0.24bpp)(b) *SA-Fractal/Spatial* (30.10dB at 0.26bpp)(d) *Fractal/SA-DCT* (30.31dB at 0.26bpp)Fig. 7. Foreground of *News* test image with mean-valued background encoded using four fractal coding methods.

form similarly numerically but often better visually than standard fractal algorithms. Although we have compared our methods with only three fractal coding algorithms, our approach can

be easily incorporated into other fractal coding schemes. For example, the proposed region-based functionality can be extended to quadtree [10], horizontal-vertical [12], evolutionary

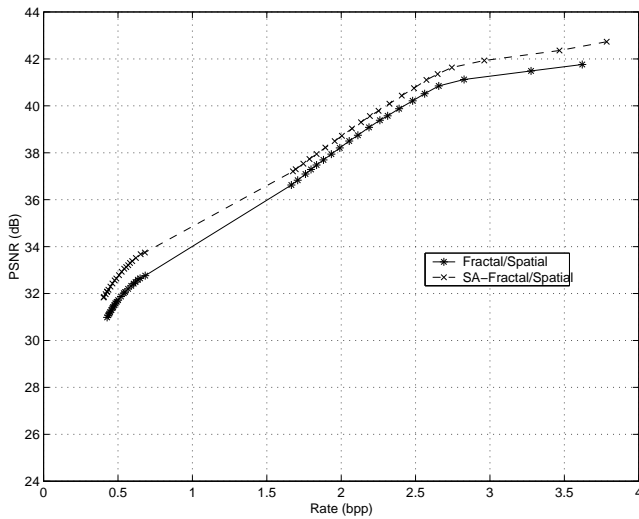
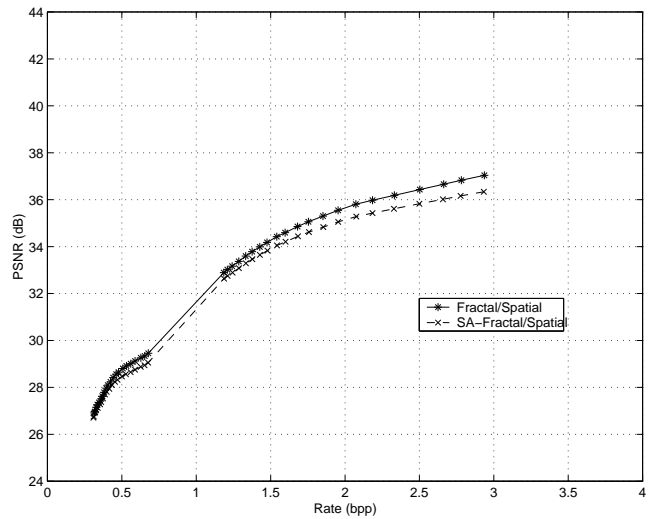
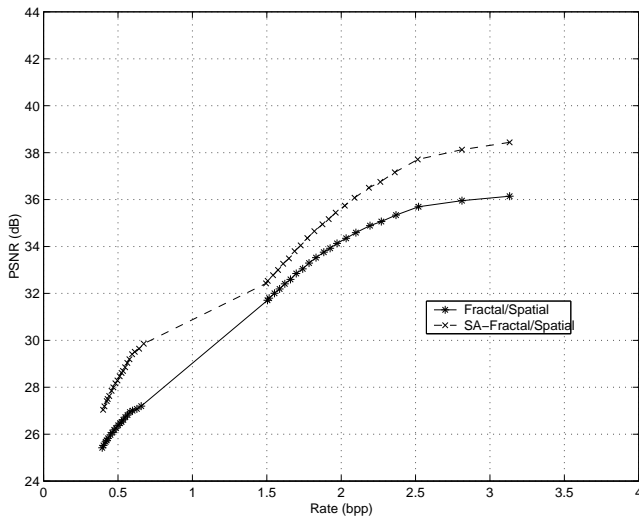
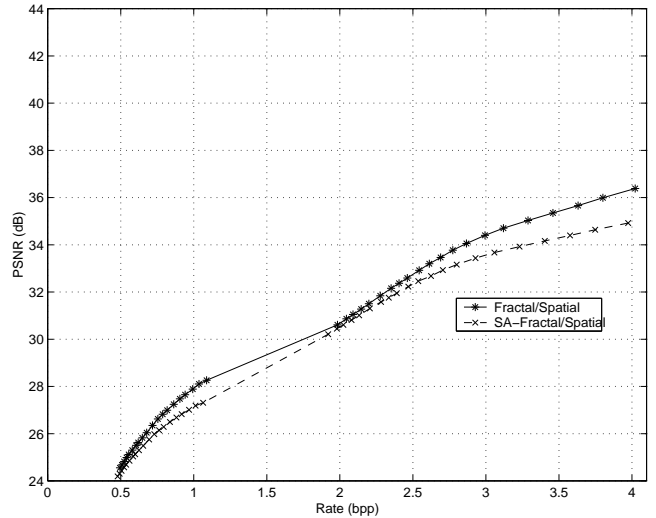
(a) *Cyclamen*(c) *Foreman*(b) *News*(d) *Children*

Fig. 8. Rate-distortion performance of region-based fractal coding (*SA-Fractal/Spatial*) and standard fractal coding (*Fractal/Spatial*) for complete images.

[15], [16], [18], triangular and quadrilateral [32] or the optimal hierarchical partitioning [33] schemes by constraining the dissimilarity measure and the domain-block search to a single region. The performance benefits of those schemes can be maintained while the functionality of the coder can be enhanced.

VIII. ACKNOWLEDGMENTS

The authors would like to thank Mr. A.-R. Mansouri of INRS-Télécommunications, Montréal, Canada for his contributions to this work and for the proof of convergence of the *SA-Fractal/Spatial* algorithm presented in the Appendix. They would also like to thank Prof. R. Stasiński of the Technical University of Poznań, Poland for providing an implementation of the SA-DCT operator. The first author would like to thank Prof. A. Baskurt of the University of Claude Bernard Lyon-1, France, for introducing him to fractal coding.

APPENDIX

The following lemma and its proof are quoted verbatim from [24] and prove the convergence of the proposed *SA-Fractal/Spatial* algorithm with *fixed* padding (zero-valued or mean-valued).

Lemma 1: Let Ω be an open bounded subset of \mathbf{R}^2 (the image domain), let $\Omega_0 \subset \Omega$ be an open subset of \mathbf{R}^2 (the region to be fractal coded), and let $\phi : L^p(\Omega) \rightarrow L^p(\Omega)$, with $p \in [1, \infty]$, be a mapping such that $\|\phi(f) - \phi(g)\|_{L^p(\Omega)} \leq C\|f - g\|_{L^p(\Omega)}$, $\forall f, g \in L^p(\Omega)$, for some $0 < C < 1$ (the contraction mapping). Let $v \in \mathbf{R}$ (the padding value), and define the operator $P : L^p(\Omega) \rightarrow L^p(\Omega)$ (the padding operator) by $P(f)(x) = f(x)$, $\forall x \in \Omega_0$, and equal to v otherwise. Then, the composition $P \circ \phi : L^p(\Omega) \rightarrow L^p(\Omega)$ is a contraction mapping.

Proof: $\forall f, g \in L^p(\Omega)$,



(a) *Fractal/Spatial* (27.33 dB at 0.73 bpp)



(c) *SA-Fractal/Spatial* (32.63 dB at 0.40 bpp)



(b) *SA-Fractal/Spatial* (30.12 dB at 0.72 bpp)



(d) *SA-Fractal/Spatial* (30.10 dB at 0.32 bpp)

Fig. 9. Test image *News* encoded using: (a) standard *Fractal/Spatial* algorithm, and (b) proposed *SA-Fractal/Spatial* algorithm; and its decomposition into (c) foreground and (d) background.

$$\begin{aligned}
 \|P \circ \phi(f) - P \circ \phi(g)\|_{L^p(\Omega)} &= \|\phi(f) - \phi(g)\|_{L^p(\Omega_0)} \\
 &\leq \|\phi(f) - \phi(g)\|_{L^p(\Omega)} \\
 &\leq C\|f - g\|_{L^p(\Omega)} \quad \text{Q.E.D.}
 \end{aligned}$$

REFERENCES

- [1] Special issue on MPEG-4, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, Feb. 1997.
- [2] B. Wohlberg and G. de Jager, "A review of the fractal image coding literature," *IEEE Trans. Image Process.*, vol. 8, pp. 1716–1729, Dec. 1999.
- [3] A. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Trans. Image Process.*, vol. 1, pp. 18–30, Jan. 1992.
- [4] Y. Zhao and B. Yuan, "A hybrid image compression scheme combining block-based fractal coding and DCT," *Signal Process., Image Commun.*, vol. 8, pp. 73–78, Mar. 1996.
- [5] K. Barthel, J. Schuttmeier, T. Voyer, and P. Noll, "A new image coding technique unifying fractal and transform coding," in *Proc. IEEE Int. Conf. Image Processing*, vol. III, pp. 112–116, Oct. 1994.
- [6] G. M. Davis, "A wavelet-based analysis of fractal image compression," *IEEE Trans. Image Process.*, vol. 7, no. 2, pp. 141–154, 1998.
- [7] H. Krupnik, D. Malah, and E. Karnin, "Fractal representation of images via the discrete wavelet transform," in *Proceedings of the 18th IEEE Convention of Electrical and Electronics Engineers in Israel*, vol. 2.2.2, pp. 1–5, Mar. 1995.
- [8] K. Belloulata, A. Baskurt, H. Benoit-Cattin, and R. Prost, "Fractal coding of subbands with an oriented partition," *Signal Process., Image Commun.*, vol. 12, pp. 243–252, June 1998.
- [9] K. Belloulata, *Compression d'images par fractals: étude sur la mesure et le domaine de recherche de l'auto-similarité et sur l'accélération de la génération du modèle fractales*. PhD thesis, Institut National de Sciences Appliquées, Lyon, France, 1998.
- [10] Y. Fisher, "Fractal encoding with quadrees," in *Fractal Image Compression: Theory and Applications to Digital Images* (Y. Fisher, ed.), pp. 55–77, Springer-Verlag, 1995.
- [11] D. Saupe and S. Jacobs, "Variance-based quadrees in fractal image compression," *Electron. Lett.*, vol. 31, no. 1, pp. 46–48, 1997.
- [12] Y. Fisher and S. Menlove, "Fractal encoding with HV partitions," in *Fractal Image Compression: Theory and Applications to Digital Images* (Y. Fisher, ed.), pp. 119–136, Springer-Verlag, 1995.
- [13] N. Lu, *Fractal Imaging*. New York: Academic Press, 1997.
- [14] L. Thomas and F. Deravi, "Region-based fractal image compression using heuristic search," *IEEE Trans. Image Process.*, vol. 4, no. 6, pp. 823–838, 1995.
- [15] D. Saupe and M. Ruhl, "Evolutionary fractal image compression," in *Proc. IEEE Int. Conf. Image Processing*, vol. III, pp. 129–132, Oct. 1996.
- [16] M. Ruhl, H. Hartenstein, and D. Saupe, "Adaptive partitionings for fractal

(a) *Fractal/Spatial* (29.50 dB at 0.72 bpp)(c) *SA-Fractal/Spatial* (40.01 dB at 0.40 bpp)(b) *SA-Fractal/Spatial* (29.26 dB at 0.73 bpp)(d) *SA-Fractal/Spatial* (29.15 dB at 0.33 bpp)

Fig. 10. Test image *Foreman* encoded using: (a) standard *Fractal/Spatial* algorithm, and (b) proposed *SA-Fractal/Spatial* algorithm; and its decomposition into (c) foreground and (d) background.

- image compression," in *Proc. IEEE Int. Conf. Image Processing*, vol. III, pp. 310–313, Oct. 1997.
- [17] M. Breazu and G. Todorean, "Region-based fractal image compression using deterministic search," in *Proc. IEEE Int. Conf. Image Processing*, vol. III, pp. 742–746, Oct. 1998.
- [18] H. Hartenstein, M. Ruhl, and D. Saupe, "Region-based fractal image compression," *IEEE Trans. Image Process.*, vol. 9, pp. 1171–1184, July 2000.
- [19] A. Jacquin, "Fractal image coding: A review," *Proc. IEEE*, vol. 10, pp. 1451–1465, Jan. 1993.
- [20] G. Oien and S. Lepsoy, "Fractal-based coding with fast decoder convergence," *Signal Process.*, vol. 3, pp. 105–117, June 1994.
- [21] R. Bracewell, K. Chang, A. Wang, and Y. Wang, "Affine theory for two-dimensional Fourier transform," *Electron. Lett.*, vol. 29, no. 3, p. 304, 1993.
- [22] Y. Zhao and B. Yuan, "Image compression using fractals and discrete cosine transform," *Electron. Lett.*, vol. 30, no. 6, pp. 474–475, 1994.
- [23] L. Thomas and F. Deravi, "Pruning of the transform space in block-based fractal image compression," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. V, pp. 341–344, Apr. 1993.
- [24] A.-R. Mansouri, "Some results in fractal coding," INRS-Telecommunications, 1999.
- [25] Special issue on representation and coding of images and video II, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, Feb. 1999.
- [26] L. Labelle, D. Lauzon, J. Konrad, and E. Dubois, "Arithmetic coding of a lossless contour-based representation of label images," in *Proc. IEEE Int. Conf. Image Processing*, vol. I, pp. 261–265, Oct. 1998.
- [27] T. Sikora and B. Makai, "Shape-adaptive DCT for generic coding of video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 59–62, Feb. 1995.
- [28] P. Kauff and K. Schüür, "An extension of shape-adaptive DCT (SA-DCT) towards DC separation and Δ DC correction," in *1997 Picture Coding Symposium*, pp. 647–652, Sept. 1997.
- [29] K. Belloulata, R. Stasiński, and J. Konrad, "Region-based image compression using fractals and shape-adaptive DCT," in *Proc. IEEE Int. Conf. Image Processing*, vol. II, pp. 815–819, Oct. 1999.
- [30] K. Belloulata, A. Baskurt, H. Benoit-Cattin, and R. Prost, "Fractal coding of medical images," in *SPIE Proceedings on Medical Imaging X: Image Display, Yongmin Kim; Ed.*, vol. 2707, pp. 598–609, Feb. 1996.
- [31] <ftp://shear.informatik.uni-leipzig.de/pub/Fractal/software/frap/frap.tar.gz>
- [32] F. Davoine, J. Svensson, and J. M. Chassery, "A mixed triangular and quadrilateral partition for fractal image coding," in *Proc. IEEE Int. Conf. Image Processing*, vol. II, pp. 284–287, Oct. 1995.
- [33] D. Saupe, M. Ruhl, R. Hamzaoui, L. Grandi, and D. Martini, "Optimal hierarchical partitions for fractal image compression," in *Proc. IEEE Int. Conf. Image Processing*, vol. III, pp. 737–741, Oct. 1998.