FRACTAL CODING OF VIDEO SEQUENCE USING CPM AND NCIM

Chang-Su Kim, Rin-Chul Kim and Sang-Uk Lee

Signal Processing Lab., School of Electrical Eng., Seoul National University, Seoul, KOREA.

ABSTRACT

We propose a novel algorithm for fractal video sequence coding, based on the circular prediction mapping (CPM), in which each range block is approximated by a domain block in the circularly previous frame. In our approach, the size of the domain block is set to be same as that of the range block for exploiting the high temporal correlation between the adjacent frames, while most other fractal coders use the domain block larger than the range block. Therefore the domain-range mapping in the CPM is similar to the block matching algorithm in the motion compensation techniques, and the advantages of this similarity are discussed. Also we show that the CPM can be combined with non-contractive inter-frame mapping (NCIM), improving the performance of the fractal sequence coder further. The computer simulation results on real image sequences demonstrate that the proposed algorithm provides very promising performance at low bit-rate, ranging from 40 Kbps to 250 Kbps.

1. INTRODUCTION

Fractal compression, which is based on the IFS (iterated function system) proposed by Barnsley[1], is a new approach to image coding recently. The basic notion of the fractal image compression is to find a contraction mapping whose unique attractor approximates the source image. In the decoder, the mapping is applied iteratively to an arbitrary image to reconstruct the attractor. If the mapping can be represented with less bits than the source image, a coding gain is obtained.

After Jacquin[2] proposed the first automatic algorithm for fractal coding of still images, much effort[3] has been made to the fractal still image coding techniques. However, little work has been reported on the fractal video sequence coding techniques. Lazar[5] and Li[6], respectively, extended the still image coding techniques straightforwardly to the video sequence coding, by employing 3-D domain blocks and range blocks. But these algorithms are very complicated to implement, and severe 3-D blocking artifacts are observed in the reconstructed images in many cases. Alternative approach, which encodes each frame using the previous frame as a domain pool, was proposed by Fisher[7]. The main advantage of [7] is that fast decoding is possible, since it does not require iteration at the decoder. But, the temporal correlation between the frames may not be effectively exploited, since the size of the domain block is larger than that of the range block.

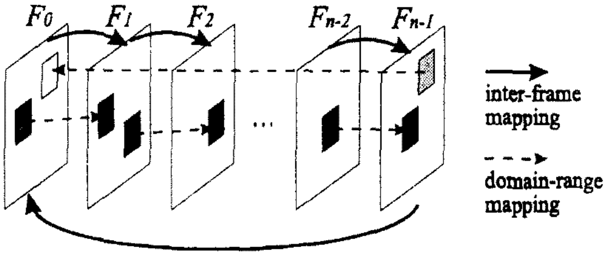

Figure 1: The structure of the CPM

In this paper, a novel approach, called the Circular Prediction Mapping (CPM), is proposed to combine the fractal sequence coder with the well-known motion estimation/motion compensation (ME/MC) techniques, so that we can exploit high temporal correlation between the frames. In the CPM, n frames are encoded as a group, and each range block is motion-compensated by a domain block in the n-circularly previous frames, which is of the same size as the range block. By selecting appropriate parameters in the domain-range mappings, the CPM becomes a contraction mapping. In the decoder, the CPM is applied iteratively to arbitrary n frames to reconstruct the attractor frames. In addition to presenting the performance of the CPM fractal coder, we shall show that the CPM can be combined with Non-Contractive Inter-frame Mapping (NCIM) to further exploit the temporal correlation between the frames, without affecting the convergence of the decoding process. The computer simulation results on various real image sequences demonstrate that the hybrid coder of CPM and NCIM provides very promising performance at low bit-rate, ranging from 40 Kbps to 250 Kbps.

2. CIRCULAR PREDICTION MAPPING (CPM)

The CPM is a suitable contraction mapping for encoding and decoding of moving image sequences[4], in which each frame is predicted blockwise from the n-circularly previous frame, as shown in Figure 1. The k-th frame F_k is partitioned into the range blocks, and each range block in F_k is predicted or approximated by a domain block in $F_{[k-1]_n}$, where $[k]_n$ denotes (k *modulo* n).

To the end of this section, we assume without loss of generality that 4 frames are encoded as a group, *i.e.*, the length of a coding group n is 4. Then the CPM is composed of 4 inter-frame mappings, and each inter-frame mapping is sum of domain-range mappings.

2.1. Domain-Range Mapping in the CPM

Each range block R_i in the k-th frame F_k is approximated by a domain block $D_{a(i)}$ in the 4-circularly previous frame $F_{[k-1]_4}$, which is of the same size as the range block. The approximation of R_i is given by

$$R_i \cong \widetilde{R_i} = s_i \cdot \mathcal{O}(D_{a(i)}) + o_i \cdot C, \qquad (1)$$

where $a(i)$ denotes the location of the optimal domain block, and s_i, o_i are real coefficients, respectively. The C is a constant block whose all pixel values are 1, and $\mathcal{O}$ is the orthogonalization operator, proposed by Øien[8] for fractal still image coder. Notice that the s_i coefficient determines the contrast scaling in the mapping, and the o_i coefficient represents the DC component of the range block R_i. By constraining the contrast scaling coefficients s_i to be quantized between -1 and 1, the CPM becomes a contraction mapping. In the decoder, the CPM is applied iteratively to arbitrary 4 frames to reconstruct the attractor frames.

Since the size of the domain block is set to be same as that of the range block, the proposed domain-range mapping can be interpreted as a kind of motion compensation techniques. Then, the $a(i)$ describes the translational motion of block, *i.e.*, the $a(i)$ is the motion vector. Besides the translational motion, the changes in contrast and overall brightness of block are compensated by the s_i and o_i coefficients, respectively. In this context, the main advantage of the proposed domain-range mapping can be discussed as follows. In real moving image sequences, small motion vector is more probable than large motion vector. Therefore, the search region for the motion vector $a(i)$ can be localized to the area near the location of the range block, yielding a significant saving in the computational burden of the encoder. Notice that most other fractal coders search over much larger region to find a good domain-range mapping. In addition, the $a(i)$ can be coded with less bits.

2.2. Decoding Algorithm

The attractor sequence can be reconstructed by iteratively applying the CPM to an initial arbitrary sequence. In general, the convergence speed is dependent on the ratio of the size of the domain block and the size of the range block. The larger is the domain block as compared to the range block, the faster the decoded sequences converge. Thus, the convergence speed is relatively slow in the CPM, since the size of the domain block is set to be same as that of the range block to exploit the temporal correlation.

But, this disadvantage is fully compensated by the advantage of the CPM that one iteration of the CPM is equivalent to 4 (= the length of a coding group) iterations in other fractal coders. This merit stems from the fact that the search region for the domain block is confined to 4-circularly previous frame. Let F_k^i denote the k-th decoded frame at the i-th iteration. At the first iteration, F_0^1 is decoded by applying (L_0, T_0) to an arbitrary frame, and F_k^1 $(1 \leq k < 4)$ is subsequentially decoded by applying (L_k, T_k) to the previously decoded frame F_{k-1}^1, respectively. Note that F_3^1 is more closer to the attractor sequence than F_0^1, since F_3^1 is actually the result of 4 iterations. At the second iteration, F_0^2 is decoded using F_3^1 of the first iteration, and so on. This process is repeated until the difference between the outputs from successive iterations becomes sufficiently small.

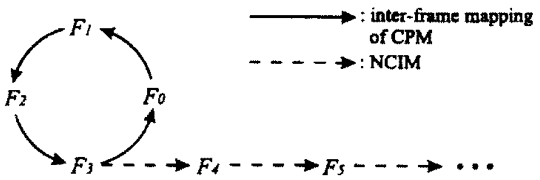

Figure 2: The hybrid structure of CPM and NCIM

2.3. CPM with Non-Contractive Inter-frame Mapping (NCIM)

The CPM is composed of 4 inter-frame mappings and each inter-frame mapping should be contractive for the decoder to reconstruct the attractor sequence. Therefore, the contrast scaling coefficients s_i in (1) are constrained to be between -1 and 1. Thus, the CPM cannot effectively exploit the temporal correlation between the frames. In other words, the increased contrast between the frames cannot be depicted by the CPM.

Thus, instead of encoding all the input frames with the CPM, the hybrid structure of CPM and NCIM can be employed in the encoder, as shown in Figure 2. The first four frames F_k $(0 \leq k < 4)$ are encoded by employing the CPM, and the following frames F_k $(k \geq 4)$ are encoded by employing the NCIM's, respectively. The NCIM's are same as the 4 inter-frame mappings which compose the CPM, except that there is no constraint on the contrast scaling coefficients s_i, *i.e.*, the absolute value of s_i could be larger than 1. Therefore, we can exploit the temporal correlation further with the NCIM's, obtaining more coding gain. But if a scene change occurs, then the CPM should be employed again in order to encode the first four frames, without depending on the frames before the scene change. And the following frames are subsequentially encoded with the NCIM's, till the next scene change occurs.

In the decoder, the first four frames F_k $(0 \leq k < 4)$ are reconstructed by applying the CPM iteratively. Then, F_4 can be reconstructed by applying the NCIM to the reconstructed F_3 without requiring iteration, even if the NCIM which predicts F_4 from F_3 is not a contractive mapping. Similarly, F_k $(k \geq 5)$ can be sequentially reconstructed by applying the NCIM to the reconstructed F_{k-1}.

3. DESIGN OF CODER

In this paper, we implement two different fractal coders. The first one is the CPM fractal coder, in which all the input frames are encoded with the CPM. And the second one is the hybrid fractal coder of CPM and NCIM, in which the first four frames are encoded with the CPM and the following frames are encoded with the NCIM's, respectively. This section describes several issues relating to the implementation of these two coders in more detail.

3.1. CPM Fractal Coder

3.1.1. Image Partitioning

Since moving image sequences are unbounded in temporal direction, they should be temporally partitioned before encoding. In our approach, each 4 frames of input sequences is encoded as a coding group ($n = 4$). If the length of a coding group n is large, the backward prediction error of F_0 from F_{n-1} will become larger and there will be a significant time-delay between the encoder and the decoder. On the contrary, if the n is small, the overall bit-rate will become higher, which we shall discuss in 4.1.2 in more detail, and the decoding speed will become slower. Therefore, in this paper, the n is selected to be 4 as a tradeoff.

After the temporal partitioning, each frame is spatially partitioned into the range blocks of maximum size 32×32 and minimum size 4×4, using the quadtree structure. First, 32×32 range block is approximated by a domain block in the 4-circularly previous frame, and if the approximation error is larger than the pre-specified threshold, then it is decomposed further into four smaller 16×16 range blocks. This process is repeated until the approximation error becomes smaller than the threshold or 4×4 range block is generated.

3.1.2. Parameter Quantization and Bit Allocation

For efficient transmission or storage, it is necessary to quantize the coefficients. Let us describe the issue relating to the quantization in more detail.

The compressed data for each range block are composed of the address $a(i)$ and s_i, o_i coefficients. First, the s_i coefficients representing the contrast-scaling in the domain-range mappings are fixed to 0.9, since the coefficients are distributed compactly at the center of 1 and should be quantized between -1 and 1 to ensure the contractivity of the CPM. Therefore, no bit is allocated to the s_i coefficients. Secondly, the o_i coefficient represents the DC component of the range block R_i, and is highly correlated with the DC component of the optimal domain block $D_{a(i)}$. Thus, the o_i in F_k ($1 \leq k < 4$) is predicted from the o_i's in F_{k-1}, and the prediction error is encoded with the Huffman coder. But the o_i in F_0 is uniformly quantized with 8 bits between $0 \sim 255$ for the causality of the system, so the bit-rate for F_0 is usually higher than the bit-rates for the other frames in the same coding group. Lastly, the search region for the optimal domain block $D_{a(i)}$ is the square centered at the location of the range block, and we express the address $a(i)$ with respect to the location of the range block. The possible coordinates of the $a(i)$ in the forward and backward domain-range mappings are constrained to be in $\{-16, -14, \ldots, 12, 14\}$ and $\{-15, -13, \ldots, 13, 15\}$, respectively. The possible address of the optimal domain block in the forward domain-range mappings are intentionally separated from that in the backward domain-range mappings, in order to eliminate the self-mapping effect[4]. Eight bits are needed for representing each $a(i)$ with fixed-length codewords. But the probability distribution of the $a(i)$ is not uniform, since the $a(i)$ is the counterpart of the motion vector in ME/MC. More specifically, the small motion vector is more probable than the large motion vector. This nonuniformity is exploited by the Huffman coder, using the probability distribution function obtained from many test sequences.

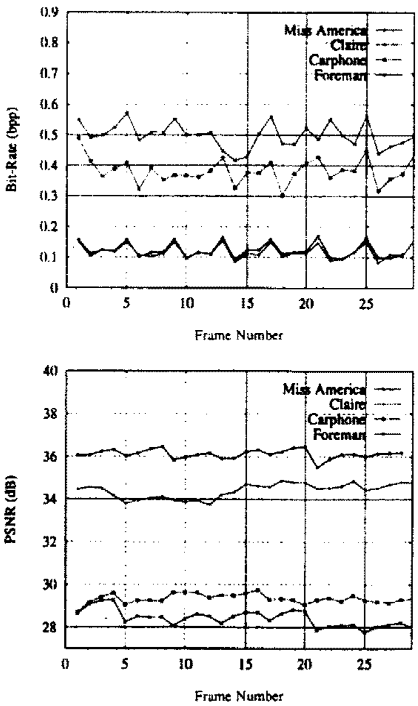

Figure 3: The bit-rate and PSNR performances for the various CIF sequences

3.2. Hybrid Coder of CPM and NCIM

The hybrid coder of CPM and NCIM is implemented as shown in Figure 2. The NCIM is implemented in the same way as the CPM, except that the coefficients s_i are uniformly quantized, instead of being fixed to 0.9. As the size of the quadtree-partitioned range blocks gets larger, the optimal s_i coefficients become more compactly distributed at the center of 1, since it is more probable that the domain-range mapping describes true motion between the frames. Therefore, different uniform-quantizers are employed, according to the size of the range block. It is clear that the uniformly quantized coefficients do not have uniform distribution. Therefore, the Huffman coder is employed to further compress these coefficients.

4. SIMULATION RESULTS

4.1. CPM Fractal Coder

The proposed CPM fractal coding algorithm is tested on real moving image sequences. Figure 3 shows the bit-rate and PSNR performances for various CIF (352×288) gray-level image sequences, whose frame rates are 25 frames/s. For the "Miss America" and "Claire" sequences, the average bit-rates are 0.124 and 0.116 bpp, yielding 64.5 and 68.9 of

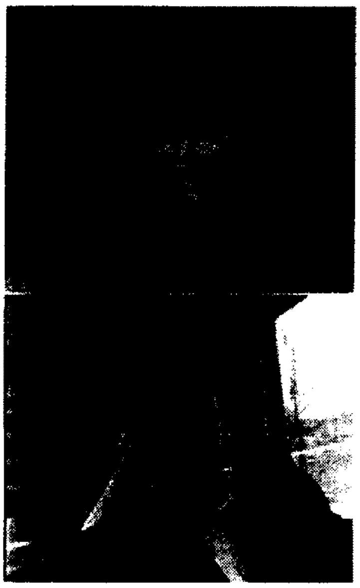

Figure 4: The samples of the reconstructed frames (top: "Miss America" 1st frame, bottom: "Car Phone" 1st frame)

the compression ratios, respectively. In other words, 5 ~ 6 frames can be transmitted in a second at the bandwidth of 64 Kbps. The bit-rates for the "Foreman" and "Car Phone" sequence are higher than those of the "Miss America" and "Claire" sequences. This is inevitable since these sequences are finely-detailed and contain large inter-frame motions.

Figure 4 presents samples of the reconstructed frames. It is seen that the proposed algorithm reconstructs the "Miss America" sequence with good subjective quality. We believe that the subjective quality is sufficient for video conferencing applications, since it does not yield severe blocking artifacts, which are the main defects of the 3-D block approaches[5, 6]. It is also observed that the trees outside the window in the "Car Phone" sequence are reconstructed very faithfully, though they move very fast. This is due to the fact that the domain-range mapping of the CPM is very similar to the ME/MC techniques. In the 3-D block approaches, the domain-range mapping often fails, and the quality of the reconstructed frames is poor in such finely-detailed and fast moving regions. These simulation results indicate that the CPM fractal coder provides much better performance than the conventional 3-D block approaches.

4.2. Hybrid Fractal Coder of CPM and NCIM

In this experiment, the hybrid fractal coder of CPM and NCIM is tested on real moving image sequences. The tested sequences are the CIF (352×288) gray-level image sequences, whose frame rates are originally 25 frames/s. But the frame rates are reduced to 8.33 frames/s to verify the performances at low bit-rate, below 256 Kbps. In other words, every third frame is encoded and the other two frames are skipped.

Figure 5 shows the bit-rate and PSNR performances of the hybrid fractal coder on the "Foreman" and "Claire" sequences. Since both sequences contain no scene change in the 1st ~ 80th frames (originally, 1st ~ 238th frames), the first four frames are encoded with the CPM and the following frames are encoded with the NCIM's, respectively. It is seen that the performances for the NCIM-coded frames are better than those for the CPM-coded frames. This is due to that the NCIM's can exploit the temporal correlation more effectively than the CPM, since there is no constraint on the contrast scaling coefficients s_i in the NCIM's. However, the CPM should be employed at the start of a sequence or after scene changes, in order to encode the first four frames without depending on the previous frames. In addition, the CPM-coded frames provide access points to the coded sequence, where the decoding process can begin. Therefore the CPM-coded frames should be inserted to the coded sequence in some frequency, according to the requirement of random access.

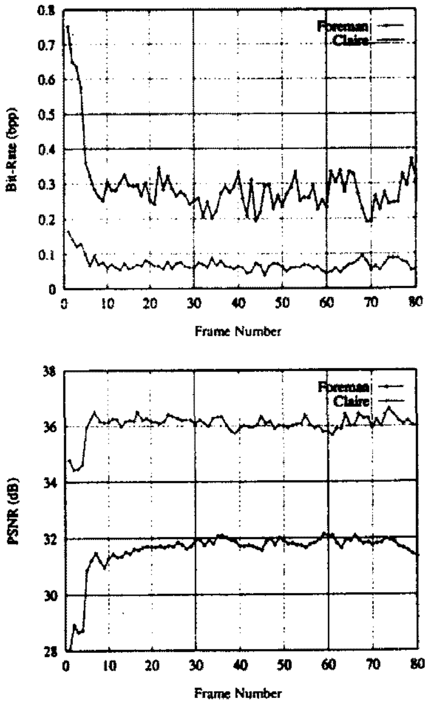

Figure 5: The bit-rate and PSNR performances of the hybrid fractal coder on the "Foreman" and "Claire" sequences

In Figure 6, we present the rate-distortion performances of the hybrid fractal coder on the "Foreman", "Car Phone" and "Claire" sequences. The provided PSNR and bit-rate are averaged over all the frames. Figure 7 shows samples of the reconstructed "Foreman" and "Claire" sequences. It is observed that the proposed algorithm reconstruct "Foreman" and "Claire" sequences with good subjective quality. The bit-rate for the "Foreman" sequence (=92.1 Kbps) is higher than that for the "Claire" sequence (=48.1 Kbps), since there is fast camera panning in the "Foreman" sequence. The "Claire" is a typical image sequence for videophone or video-conference applications. In such head-and-

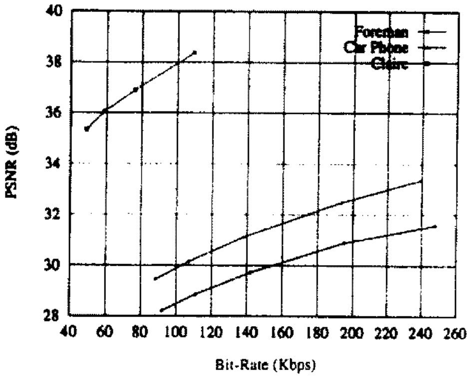

Figure 6: The rate-distortion curves for the "Foreman", "Car Phone" and "Claire" sequences

shoulder image sequences, the hybrid fractal coder provides a good image quality at very low bit-rate, below 64 Kbps. This indicates that the proposed hybrid fractal coder is a promising technique for the image sequence coding at very low bit-rate.

5. CONCLUSION

In this paper, we proposed a novel algorithm for fractal video sequence coding, based on the Circular Prediction Mapping (CPM). In our approach, each range block was approximated by a domain block in 4-circularly previous frame, and the size of the domain block was set to be same as that of the range block, in order to exploit the high temporal correlation in real moving image sequences.

We implemented two CPM-based fractal coders: 1) The first one is the CPM fractal coder, in which all the frames are encoded by employing the CPM. It was demonstrated by the computer simulation on the "Miss America" and "Claire" sequences that the CPM fractal coder provides the average compression ratios ranging from 60 to 70, without observing severe blocking artifacts, which are the main defects of the 3-D block approaches. 2) The second one is the hybrid fractal coder of CPM and Non-Contractive Interframe Mapping (NCIM). It was demonstrated that the hybrid fractal coder provides sufficient image quality for the video conferencing image sequence at very low bit-rate, below 64 Kbps. We believe that the CPM-based fractal coder will be a strong candidate for the very low bit-rate video coding techniques.

6. REFERENCES

[1] M. F. Barnsley, *Fractals Everywhere*, Academic Press, San Diego, 1988.

[2] A. E. Jacquin, "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations," *IEEE Trans. Image Processing*, vol.1, no.1, pp.18-30, Jan. 1992.

[3] Y. Fisher (editor), *Fractal Image Compression-Theory and Application*, Springer-Verlag, New York, 1995.

Figure 7: The samples of the reconstructed frames (top: "Foreman" 7th frame, 28.87 dB, 92.1 Kbps, bottom: "Claire" 7th frame, 35.80 dB, 48.8 Kbps)

[4] C. S. Kim and S. U. Lee, "Fractal Coding of Video Sequence by Circular Prediction Mapping," to be published in the *Proc. of NATO ASI on Fractal Image Encoding and Analysis*, Trondheim, Norway, 1995.

[5] M. S. Lazar and L. T. Bruton, "Fractal Block Coding of Digital Video," *IEEE Trans. Circuits and Systems for Video Technology*, vol.4, no.3, pp.297-308, Jun. 1994.

[6] H. Li, M. Novak and R. Forchheimer, "Fractal-based Image Sequence Compression Scheme," *Optical Engineering*, vol.32, no.7, pp.1588-1595, 1993.

[7] Y. Fisher, T. P. Shen and D. Rogovin, "Fractal (Self-VQ) Encoding of Video Sequences," in *Proc. SPIE:VCIP*, vol.2308, pp.1359-1370, 1994.

[8] G. E. Øien, S. Lepsøy and T. A. Ramstad, "An Inner Product Space Approach to Image Coding by Contractive Transformations," in *Proc. ICASSP*, pp.2773-2776, 1991.