Low bit rate video compression using relative fractal coding

Jayanta Mukhopadhyay and S.K. Ghosh
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur, India - 721 302.

ABSTRACT

A new approach of relative fractal coding has been presented in this paper. In this technique, given the fractal code of a reference image, one can generate a relative fractal code of any other image of same size. The convergence of the relative code is also guaranteed. This method is found to be well suited for video compression as there is high temporal redundancy between the two successive frames.

1 Introduction

The promising performance provided by Jacquin's [1] algorithm for fractal based image compression and subsequent development of other techniques based on it, led to the application of fractal compression techniques to video compression. The major challenge in compressing video using fractal methods is to guarantee the convergence of the decoder. In case of the conventional fractal coding, the convergence of the decoder depends on the total fractal code of the image and thus any local change in the code may not guarantee the convergence of the decoder.

The research in the fractal video compression mostly concentrated in the following two approaches -

- Intra-frame coding using fractal technique and inter-frame coding using other schemes like vector quantization techniques [2, 3, 4, 5, 6].
- Extension of the conventional 2-D fractal scheme to three-dimensional blocks where the third coordinate represents the temporal axis [2, 7, 8].

It may be noted that in the first approach the codes of the individual frames are not solely fractal codes and they do not converge independently. In the second case the video sequence is modeled as a 3-D block with the third-dimension as the temporal axis. As a result the temporal redundancies, which are usually localized, are not exploited efficiently.

In this paper a new approach for fractal video coding based on an innovative concept of *relative fractal coding* has been proposed. The major objective of this work is to present the principle of *relative fractal coding* and its application in video compression. We do not intend to compete with other efficient video coding techniques based on DCT or wavelet transforms. The proposed scheme produces fractal code where a portion of the code is taken from the reference (i.e. the previous frame) using the local inter-frame temporal changes. The convergence for individual frames are also taken care of by additional measures.

2 Overview of Fractal Image Encoding and Decoding

In fractal compression an image is encoded as the attractor of an iterated function system. It is based on the observation that natural images are partially self-transformable [1]. They contain 'affine redundancy' in the sense that a block in the image (called *range*) can be derived from another block of the same image(called *domain*) by some affine transformation. In the fractal encoding method, following Jacquin's approach, the encoding process starts with partitioning of the image into a set of non-overlapping segments(*range blocks*) and then for each range block an image block (*domain block*) with different resolution is searched that gives the best affine mapping to the range segment. Compression is achieved by encoding the location of the domain blocks and the affine transformation for each range segment. The algorithm is described in the next section.

2.1 The Conventional Algorithm

The conventional algorithm as proposed by Jacquin [1] can be summarized as follows. An image I is defined as the mapping of points in discrete 2D space $Z \times Z$ to gray level values belonging to real set R such that $I : Z \times Z \to R$. The input image I is partitioned into non-overlapping *range blocks*. For every range block R_i in I, a suitable *domain* block D_j in I is located by exhaustively searching the image and an associated affine transformation W such that R_i can be reconstructed (at least approximately) as $W(sh(iso(D_j)))$, where $iso(.)$ denotes isometric transformation on D (say rotation) and $sh(.)$ is a shrinking operation (say averaging or decimation operations). The criteria for selecting the domain D_j for range R_i is that the error between R_i and $W(sh(iso(D_j))$ (i.e. the *Root_Mean_Square(RMS) error* is less than a threshold value. It may be noted that this threshold is termed in this work as the *qualifying block threshold* (QBT).

In the compressed image, the $< D, sh, iso, W >$ information is stored for every range block. As the shrinking function sh will be uniformly applied to each domain, it is sufficient to store only $< D, iso, W >$.

2.2 Fractal decoding and convergence of the decoder

The decoding process is considerably simple. Given the codes for a range R_i in the form of $< D_j, iso_i, W_i >$ one has to iteratively apply $W_i((sh(iso_i(D_j))))$ for every ith range block. The convergence of the decoder is guaranteed when the magnitudes of the *scale factors* of the affine transforms are kept less than *one*.

It has been shown in [9] that the fractal encoded image can be modeled as a *flow graph*, where the directed edges from a pixel q to p exists if the brightness value of the pixel p is determined from q. It is proved that *if for every pixel there exists one and only one pixel from which the brightness values are computed, the image space is partitioned into a set of* circular plants. The structure of a circular plant consists of a circular chain, called as its *limit cycle* and several chains of pixels coming out of the *limit cycle*. Each circular plant converge independently. The convergence of the encoder under this circumstance depends upon the convergence of these *limit cycles*. The example of a *circular plant* is shown in figure 1. The detailed discussion and analysis of this partitioning could be found in [9]. The concept also leads to the development of a linear time fast decoding algorithm.

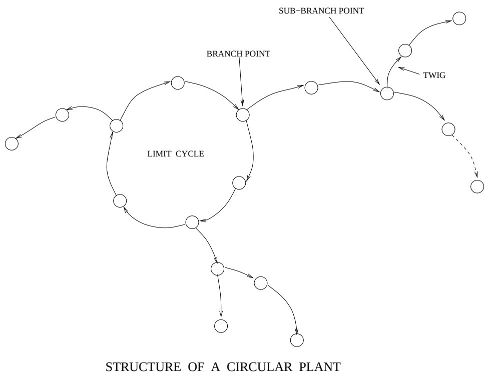

Figure 1: A *circular* plant

3 Relative fractal coding

In this section we will introduce the concept of *relative fractal coding*. In this encoding technique, given the fractal code of a reference image, one can generate a relative fractal code of any other image of the same size. This relative fractal code combined with the code of the reference image, produces the complete fractal code of the target image. More the similarity in the images, less is the size of the relative code. In the relative code, the same range-domain mappings of the reference image are used. Only the transformation of brightness values are changed, if required.

The relative fractal coding is basically meant for images having considerable similarity with the reference image (like video frames or components of an color image). As it is expected that many of the range blocks in an image remain (almost) unchanged with respect to the reference image, the associated affine transforms of the reference image also could be used in the relative codes. Hence in this case only a few number of affine transforms are required to be stored with an additional overhead of storing the associated range information. Hence the relative code structure consists of the header information storing the range identities for which the affine transforms are stored, the sequence of affine transforms and the list of brightness values for the sequence of limit cycle points. This algorithm for the relative fractal coding is presented below. Here the affine transforms of the reference image are used in the relative codes if RMS error is within a particular error threshold, termed as *Relative_error_threshold*.

```
Algorithm Relative_Fractal_Coding
Input : An image $I$, Reference fractal code, Relative_error_threshold.
Output : The relative code of the image $I$.

1. For every range block $R$ do {
    1.1. Get the reference code for the range as $< D_R, iso_R, W_R >$.
    1.2. Find the affine transform $W'_R$ which minimizes $E(I(R), W'_R(I(sh(iso_R(D_R)))))$.
    1.3. Calculate error difference , $Err\_diff = E(I(R), W'_R(I(sh(iso_R(D_R))))) - E(I(R), W_R(I(sh(iso_R(D_R)))))$
    1.4. If ($Err\_diff < Relative\_error\_threshold$) {
            Output the range identity and $< W'_R >$.
         }
   }
2. Compute the sequence of limit cycle points and output their brightness values in $I$.

End Relative_Fractal_Coding
```

3.1 Convergence of the relative code

As the convergence of the fractal decoder solely depends on the convergence of the limit cycles (refer section 2.2), given the brightness values of the limit cycle points, decoding process of the fractal codes will always converge. It may be noted that as the number of limit cycle points is very small, this will add an insignificant amount of overhead on the fractal code size. Hence the respective brightness values of the points of the limit cycles are appended with the relative code (refer step 2 of the algorithm *Relative_Fractal_Coding* of the previous section).

4 Video compression using *relative fractal coding*

Relative fractal coding is found to be useful in compressing video sequences as the successive frames are expected to be highly correlated. In a simple strategy, one may define a group of frames. The first frame of the group acts as the reference image and the others are coded with respect to its fractal code.

Further, the performance is greatly improved if the reference frame is also shifted to the previous relatively coded frame in the compression scheme and can be termed as *Progressive Relative Fractal Coding* (PRFC) technique. This not only improves the compression ratio, but also enhances the quality of reconstruction. In this case for relatively coding a video frame, first the full fractal code of its previous frame is computed and then the code is used as the reference code.

4.1 Performance of the algorithm

Experimentations are carried out for a set of video sequences using *Progressive Relative Fractal Coding* (PRFC) scheme. The reference image is fractally coded with both fixed range size and quadtree partitioning scheme.

Experimental results of a set of video sequences are shown in table 1. The video sequences are sampled at 10 fps (frames per second). Each component of the color frames [one luminance (Y) and two chromatic components (U,V)] is independently encoded using this scheme. A typical example of variations of PSNR and BPP with the frame numbers for the sequence *Akiyo* is demonstrated in figure 2. It is interesting to note that the frame number 1 has high BPP values (around 0.5) in all Y, U and V components whereas in the subsequent frames the values are negligible. It may also be noted that the encoding time of the first frame is quite high (typically 309.77 seconds), while that of subsequent frames are negligible (typically, ranging between 0.04 to 0.06 seconds).

The frames (frame number 1, 2, 10, 25) of the reconstructed video sequence of *Akiyo* are shown in figure 3. In table 2 the results for *Akiyo* sequence for various frame rates are presented. It has been observed that the comparable results with respect to *wavelet* and *DCT* based schemes are obtained. Talluri *et al.* [10] have presented some results for wavelet and DCT based schemes (refer table I of [10]). From [10] it can be observed that for *Akiyo* sequence, in *wavelet* based scheme a bit rate of 10 Kbps (for a frame rate of 5 fps in QCIF resolution) with PSNR values of 32.3 dB, 35.4 dB and 38.4 dB (in Y, U and V components) has been achieved. On the other hand, for *DCT* based approach a bit rate of 48 Kbps (for 10 fps in CIF resolution) with PSNR values of 34.0 dB, 40.9 dB and 43.4 dB (in Y, U and V components) is obtained. In the proposed scheme for 5 fps frame rate in QCIF bit rate of 5.92 Kbps is achieved (with average PSNR values of 26.99 dB, 32.72 dB and 37.67 dB in Y,U and V components), and for 10 fps frame rate in CIF a bit rate of 42.17 Kbps is obtained (with average PSNR values of 29.34 dB, 35.35 dB and 38.70 dB in Y,U and V components). However, it must be mentioned, though the data rate is found to be much smaller in this case, the PSNR values are also less for the proposed scheme. The proposed scheme should not be seen as a competitive scheme of the existing efficient DCT or Wavelet based scheme, rather it is a demonstration of the concept of relative fractal coding.

Video Sequence	Avg PSNR Y(dB)	Avg PSNR U(dB)	Avg PSNR V(dB)	Data Rate (Kbps)
Akiyo	27.42	32.74	37.29	8.46
Claire	26.96	33.91	37.04	13.45
Foreman	16.45	34.95	33.91	23.24
MissUsa	29.43	35.77	30.61	17.21
News	21.95	30.92	34.86	10.95

Table 1: Relative fractal coding for video sequences

Figure 3: Reconstructed frames of *Akiyo* video sequence

5 Conclusion

A new method for fractal video coding has been proposed in this paper. It is based on an innovative concept of *relative fractal coding*, used for reflecting the lo-

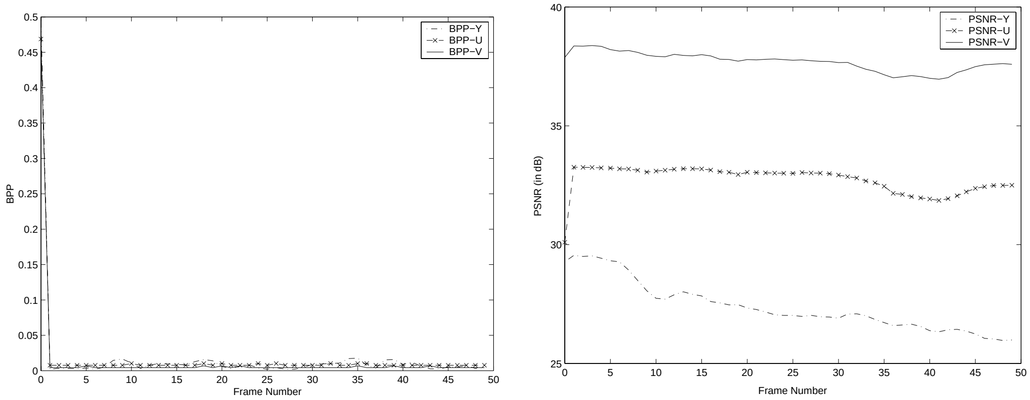

Figure 2: *BPP vs Frame Number* and *PSNR vs Frame Number*

Frame Rate (fps)	Type	Bit Rate (Kbps)	Avg PSNR Y(dB)	Avg PSNR U(dB)	Avg PSNR V(dB)
5	CIF	24.60	27.88	34.92	38.33
5	QCIF	5.92	26.99	32.72	37.67
7.5	CIF	42.09	28.55	35.17	38.43
7.5	QCIF	8.10	27.82	32.94	37.60
10	CIF	42.17	29.34	35.35	38.70
10	QCIF	10.05	28.18	33.01	38.00

Table 2: Coding results for *Akiyo* video sequence

cal changes in a sequence of video frames. The major problem in fractal video coding is to ensure the convergence of the decoder. In this technique the convergence of the decoder has been ensured by computing the *limit cycle points* and transferring those points in the successive frames. The proposed *relative fractal coding* scheme is found to be efficient in coding low bit rate video. It is interesting to note that the BPP and encoding time of the relatively coded frames are extremely low.

References

[1] A. E. Jacquin. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Transaction on Image Processing*, 1, January 1992.

[2] J.M. Beaumount. Image data compression using fractal techniques. *Br. Telecommunication Technology Journal*, 9(4):93–109, October 1991.

[3] L.Hurd, M.F. Barnsley, and M. Gustavus. Fractal video compression. *Proceedings of 37th IEEE Computer Society International Conference*, 37:41–42, 1992.

[4] B. Hurtgen and P. Buttgen. Fractal approaches to low rate video coding. *Proc. SPIE Conf. Visual Communications and Image Processing (VCIP)*, 1:120–131, November 1993.

[5] Y.Fisher, D.N. Rogovin, and T.P.J. Shen. Fractal (self-vq) encoding video sequences. *Visual Communication and Image Processing, SPIE Proceeding*, September 1994.

[6] D.L. Wilson, J.A. Nicholls, and D.M. Monro. Rate buffered fractal video. *Proc. of ICASSP*, V:505–508, 1994.

[7] H.Li, M. Novak, and R. Forchheimer. Fractal based image sequence compression scheme. *Optical Engineering*, 32(7):1588–1595, July 1993.

[8] K.U. Barthel and T. Voye. Three-dimentional fractal video coding. *International Conference on Image Processing*, 3, October 1995.

[9] J. Mukherjee, P. Kumar, and S.K. Ghosh. A graph-theoretic approach for studying the convergence of fractal encoding algorithm. *IEEE Transaction on Image Processing*, 9, March 2000.

[10] R. Talluri, K. Oehler, T. Bannon, J.D. Courtney, A. Das, and J.Liao. A robust, scalable, object-based video compression technique for very low bit-rate coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1), February 1997.