Appendix F Using Linear Fractal Interpolation Functions to Compress Video Images

The paper in this appendix was presented at the Fractals in Engineering '94 Conference which was held in the École Polytechnic, Montréal on 1-4 June 1994. The paper appeared both in the journal, *FRACTALS* and published as a proceeding, the full references of which are given below:

Ali, M. and Clarkson, T.G., “Using Linear Fractal Interpolation Functions to Compress Video Images”, *FRACTALS - An Interdisciplinary Journal on the Complex Geometry of Nature*, (World Scientific Publishing Company), Vol.2, No.3, Sept., 1994, p.417-421, ISSN 0218-348X, MITA(P) No. 195/12/93.

Ali, M., and Clarkson, T.G., “Using Linear Fractal Interpolation Functions to Compress Video Images”, *Fractals in Engineering*, eds. S. Baldo, F. Normant and C. Tricot, World Scientific, Singapore, 1994, p.232-236. ISBN 981-02-1835-4.

Further references on this work may be found in:

Ali, M. and Clarkson, T.G., “Video Image Compression Using Fractal Interpolation Functions and Calculation of its Fractal Dimension”, *Proc. of the Nonlinear Dynamics of Electronic Systems Workshop, NDES’94*, Univ. of Mining and Metallurgy, Kraków, Poland, 29-30 July, 1994, p.33-38.

Ali, M., and Clarkson, T.G., “Video Image Compression Using Fractal Interpolation Functions and Calculation of its Fractal Dimension”. To appear in the *International Journal of Electronics*, special issue on Non-linear Dynamics of Electronic Systems, Publication date: Spring 1995.

Fractals, Vol.2 No.3 (1994) 417-421

USING LINEAR FRACTAL INTERPOLATION FUNCTIONS TO COMPRESS VIDEO IMAGES

MAARUF ALI and TREVOR G. CLARKSON

Communications Research Group, Dept. of Electronic & Electrical Engineering

King's College London, Strand, London WC2R 2LS, U.K.

e-mail: `m.ali@kcl.ac.uk`

Phone: +44-(0)171-836-5454 Ext. 3583

Fax: +44-(0)171-836-4781

This paper describes the novel application of using linear fractal interpolation functions (FIFs) to model video signals represented as single-valued discrete-time sequences to compress video images. The problem is data compression of full-motion broadband television signals. The viability of using FIFs to model video signals is shown by modelling test static image frames. Compression ratios, SNRs and compression-decompression times are reported. Extension of this work to compress motion video is described. Finally the images are analysed by calculating and plotting the fractal dimensions of each line in the frame against the line for that image.

1. LINEAR FRACTAL INTERPOLATION FUNCTIONS (SELF-AFFINE FRACTAL MODEL)

In linear fractal interpolation, a set of points is interpolated with a continuous, single valued function that passes through the given interpolation points[1-3]. The interpolation function is constructed with m affine maps of the form:

$$w_i\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_i & 0 \\ c_i & d_i \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix}, \qquad i = 1, 2, \ldots, m \tag{1}$$

where d_i is constrained to lie in the interval (-1, 1). Each affine map is constrained to map the endpoints of the set of interpolation points to two consecutive interpolation points, that is:

$$w_i\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} x_{i-1} \\ y_{i-1} \end{pmatrix} \quad \text{and} \quad w_i\begin{pmatrix} x_m \\ y_m \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \end{pmatrix}, \qquad i = 1, 2, \ldots, m \tag{2}$$

Controlling the vertical scaling factor, d_i, independently, the four parameters (a_i, c_i, e_i and f_i) are given by:

$$a_i = \frac{(x_i - x_{i-1})}{(x_m - x_0)} \tag{3}$$

$$c_i = \frac{(y_i - y_{i-1})}{(x_m - x_0)} - d_i \frac{(y_m - y_0)}{(x_m - x_0)} \tag{4}$$

$$e_i = \frac{(x_m x_{i-1} - x_0 x_i)}{(x_m - x_0)} \tag{5}$$

$$f_i = \frac{(x_m y_{i-1} - x_0 y_i)}{(x_m - x_0)} - d_i \frac{(x_m y_0 - x_0 y_m)}{(x_m - x_0)} \tag{6}$$

The fractal function is decompressed using the deterministic FIF generation routine.

1.1 Contraction Factor, d_i , Calculation by the Analytic Method

As explained in Fig. 1, the determination of d_i is simply:

$$d_i = \frac{\nu_{max}}{\mu_{max}} \tag{7}$$

If the function H, lies below the straight line for either ν or μ then that term has a negative value.

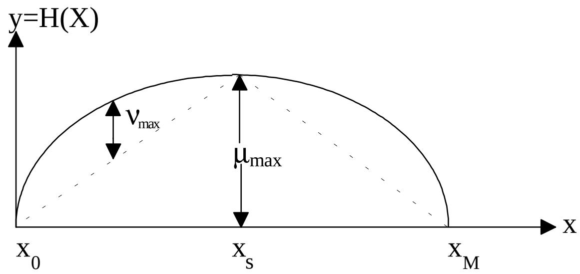


Fig.1 Determination of the affine parameter d_i for a function $H(x)$ to be modelled by fractal interpolation functions.

1.2 Summary of an Inverse Algorithm (Parameter Identification Algorithm)

1. Choose the initial point on the function H as the first interpolation point and the left endpoint of the first section[1].

2. Choose the next point on the function as the next interpolation point and the right endpoint for that section.

3. Compute the contraction factor, d_i, associated with the section defined by the pair of interpolation points, using Eq. (7).

4. If $|d| \geq 1$ go to step two, otherwise go to step five.

5. Compute the affine map parameters a_i, e_i, c_i and f_i using Eqs. (3)-(6) and form the affine map $\boldsymbol{w}_i$ associated with the pair of interpolation points. Apply the map to each point of the function to yield $\boldsymbol{w}_i(H)$.

6. Compute and temporarily store the distance between the original function located between the pair of interpolation points, say H_i, and $\boldsymbol{w}_i(H)$. The Euclidean distance measure is a convenient measure to program.

7. Repeat steps two to six until the end of the function, *H*, is reached.

8. Store the pair of interpolation points and contraction factor which yield the minimum value of $h(H_i, \mathbf{w}_i(H))$ from steps five and six. This step may be speeded up by satisfying some tolerance limit. '*h*' is the Euclidean distance measure in our case.

9. Let the right endpoint of the stored pair of interpolation points be the left endpoint of the next pair of interpolation points

10. Go to step two and continue until the entire function has been searched.

If the function is self-affine, the algorithm will find a set of maps to represent the function, if the map is not self-affine then the IFS maps found will approximate the function.

2. COMPRESSION OF STATIC AND MOTION IMAGES

By representing a whole frame (256 × 256 pixels, 256 grey levels), as a single video line, the FIFs were successfully able to model the images, the results are shown in Fig. 2.

For motion video compression, the changes in the waveform from one frame to the next may be coded using FIFs or the error difference sent if the changes are within some tolerance limit.

Original Test Images | Decompressed Fractal Images

Claire Image

"cl_d"

Left = 10th Pixel

Right = 20th Pixel

Total Samples = 64,680 No. of Blocks = 256 No. of samples in each block = 256

α = 0 Time[1] ≈ 4min Total No. of Maps = 5,164 No. of Tests = 101,186

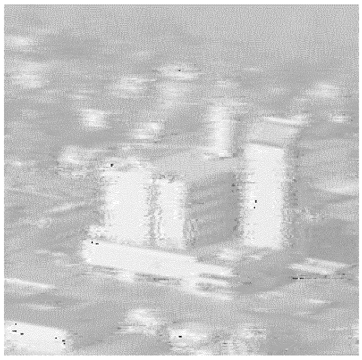
Compression Ratio[2] = 4.18 SNR = 19.7 dB

Hospital Image

"psemr256"

Left = 10th Pixel

Right = 20th Pixel



Total Samples = 65,738 No. of Blocks = 256 No. of samples in each block = 256

α = 0 Time[3] ≈ 9min Total No. of Maps = 5,181 No. of Tests = 101,434

Compression Ratio[2] = 4.17 SNR = 18.0 dB

1. Programs executed on 66MHz 486DX PC. Times are for one compression and decompression cycle.

2. Compression Ratio = (Total Samples × 8 bits/sample)/(Total No. of Maps × 24 bits/map).

3. Programs executed on 33MHz 486DX PC. Times are for one compression and decompression cycle.

Fig. 2 Image Compression Using Fractal Interpolation Functions

3. FRACTAL DIMENSION CALCULATION AND ITS USE

The box dimension[1,2] is used to calculate the fractal dimension of our discrete image data. The details of the algorithm is given in Peitgen[4]. Figure 3 shows the plot of fractal dimension per line of the video waveform for the two test images. The plot shows that the video waveforms had similar fractal dimensions and that the data were similar. The average fractal dimension was found to lie within a narrow range of 1.07 to 1.17.

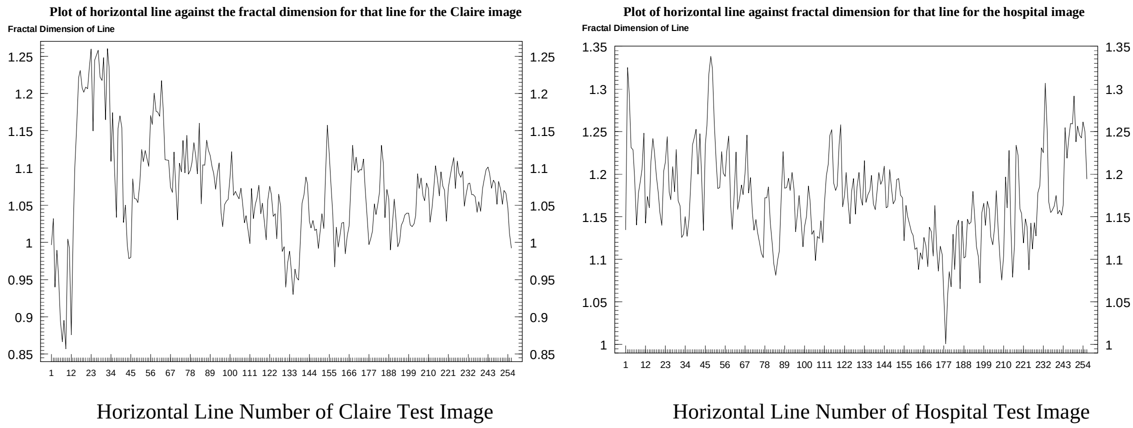


Fig. 3 Plot of Fractal Dimension/Line of the Video Waveform for the Two Test Images where the average fractal dimension is 1.07 for Claire and 1.17 for the Hospital image.

4. CONCLUSIONS

The use of FIFs to represent video signals was shown to be viable. The initial compression ratio of ~4.2:1 can be improved by optimisation of the algorithm and entropy coding of the fractal code. Self-similarity if it does not exist for one line of data, then it surely exists for multiple adjacent lines, and hence video data is suitable for fractal coding. The main advantage with fractal coding is the resolution independence of the code - images may be decoded to higher resolutions and be suitable for high definition televisions. The average fractal dimension of the video data for the whole frame lay within a narrow range of 1.07 to 1.17, this shows the similar nature of image data.

5. ACKNOWLEDGEMENTS

I would like to thank J. Mackenzie and Dr. M. B. Sandler of the Signals, Circuits and Systems Research Group, King's College London, for supplying the FIFs software. This work is funded by The British Council under the British-German Academic Research Collaboration Programme, Project 522, *Application orientated fractal image compression for video data transmission with limited bandwidth*, with the Universität Bremen in Germany. Permission to use the image of the building was kindly provided by McDonnell Douglas Missiles Systems Company, Maryland, U.S.A.

6. REFERENCES

1. D.S. Mazel & M.H. Hayes, “Using Iterated Function Systems to Model Discrete Sequences,” *IEEE Trans. on Sig. Proc.*, Vol.40, July 1992, No.7, p.1724-1734.

2. D. Oliver, *Fractal Vision*, Sams Publishing, 1st Edition 1992.

3. M Barnsley, *Fractals Everywhere*, Academic Press Inc. 1988.

4. H.-O. Peitgen & D. Saupe, eds. *The Science of Fractal Images*, Springer-Verlag 1988.