Fractal Image Compression

Maaruf Ali & Prof. Trevor G. Clarkson

Communications Research Group

Dept of Electrical & Electronic Engineering

King's College London (Univ. of London)

Strand, London WC2R 2LS, U.K.

Email: m.ali or tgc @kcl.ac.uk

Fax: +44-171-836-4781

Tel: +44-171-836-2367

Abstract

Standard graphics systems encode pictures by assigning an address and colour attribute for each point of the object resulting in a long list of addresses and attributes. Fractal geometry enables a newer class of geometrical shapes to be used to encode whole objects, thus image compression is achieved. Compression ratios of 10,000:1 have been claimed by researchers[1] in this field. The fractal equations describing these shapes are very simple equations. Specifically, **iterated function system (IFS)** codes are investigated.

The difficult inverse problem of finding a suitable IFS code whose fractal image is to represent the real image and hence achieve compression is investigated through the use of:

a) a library of IFS codes and **complex moments**,

b) the method of **simulated annealing**, for solving non-linear equations of many parameters.

Image Compression

Image compression is reducing the number of bits required to represent an image in such a way that either an exact replica of the image (**lossless compression**) or an approximate replica (**lossy compression**) of the image can be retrieved.

---

[1] M.F. BARNSLEY, A.D. SLOAN, "A better way to compress images", **BYTE**, Jan 1988, p.215-223.

Canonical Representation of Digital Images

A digital picture consists of an $n \times m$ array of integer numbers or picture elements (pels), see Fig.1.

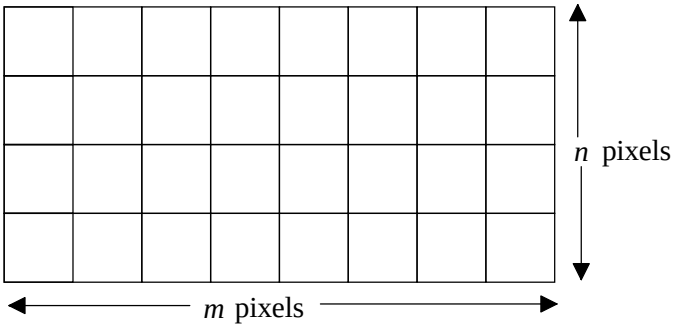


Fig.1 Canonical Representation of Digital Images.

If it takes B bits to encode each pel, then: $\boldsymbol{n \times m \times B}$ **bits** are required to represent the picture digitally. Thus for a 512 × 512 raster with 8 bits/pel: 512 × 512 × 8 = 2,097,152 bits. (A large number!)

Reasons for Compressing Images

1) To reduce the speed of transmission of data files. The larger the file, the longer it takes to transmit.

2) To reduce storage space.

3) Compressed images cost less to transmit, archive, process, retrieve, than uncompressed images.

4) Faster processing of image data which is important in knowledge based systems computing.

Applications

1) Transmission of medical images by telephone.

2) Video modem: standard telephone speed: 9600 bits/s

single video image requires: 128 × 128 × 8 = 131,072 bits

@ 25 frames/s = 3,276,800 bits

Thus required compression ratio, $C_r = 341$.

3) Movie channel over Integrated Services Digital Network (ISDN) 64 kbit/s telephone lines.

4) To achieve High Definition Television (HDTV) and Advanced Television (ATV) in a reduced bandwidth medium.

5) DVI - digital video interaction.

6) FAX: presently 4 pages/min, if $C_r = 10$, then 40 pages/min could be achieved.

Advantages of using fractal transforms

". . . a highly complex structure is generated from a simple concise kernel of data which is easy to produce. (Such large **database amplification** is a primary advantage of fractal techniques in general.)"[2]

Database amplification - the generation of complex images from small databases.

Fractals - a brief introduction

Coastlines, mountains and clouds are not easily described by traditional Euclidean geometry. The natural objects may be described and mathematically modelled by Mandelbrot's fractal geometry. This is another reason why image compression using fractal transforms are investigated. The word fractal was first coined by Mandelbrot in 1975.

Properties of fractals

1) The defining characteristic of a fractal is that it has a **fractional dimension**, from which the word **fractal** is derived.

2) The property of **self-similarity** or scaling is one of the central concepts of fractal geometry.

Self-similarity

The property of objects whereby magnified subsets appear similar or identical to the whole and to each other is known as **self-similarity**. It is a characteristic of fractals and sets them apart from Euclidean shapes which generally become smoother. Thus fractal shapes are self-similar and independent of scale or scaling and possess no characteristic size.

---

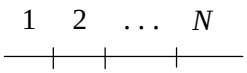
[2] P.E. OPPENHEIMER, "Real Time Design and Animation of Fractal Plants and Trees", **Computer Graphics** (Siggraph '86), Vol.20, No.4, 1986, p.55-64.

Euclidean shapes may be described by a simple algebraic formula whereas fractals are generally constructed using a recursive algorithm suited to computers.

Self-similarity and dimension

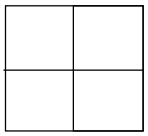
The fractal dimension need not be an integer, it may have fractional values, unlike the Euclidean dimension.

Interpretation of standard integer dimension figures in terms of exact self-similarity and extension to non-integer dimensioned fractals[3]



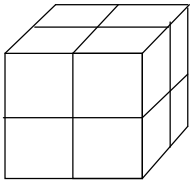
1-D N parts, scaled by ratio $r = \frac{1}{N}$

$N r^1 = 1$



2-D N parts, scaled by ratio $r = \frac{1}{N^{1/2}}$

$N r^2 = 1$



3-D N parts, scaled by ratio $r = \frac{1}{N^{1/3}}$

$N r^3 = 1$

Generalise for an object of N parts, each scaled down by a ratio r from the whole:

$$\boldsymbol{N\,r^{D}} = 1$$

defines the fractal (similarity) dimension **D**:

$$D = \frac{\log N}{\log \frac{1}{r}}$$

Self-affinity

The non-uniform scaling, where shapes are (statistically) invariant under transformations that scale different coordinates by different amounts, is known as **self-affinity**.

---

[3] H.-O. PEITGEN AND D. SAUPE, *The Science of Fractal Image*, Springer Verlag, New York, U.S.A., 1988, Fig.1.4, p.29.

The computer description of natural objects

The natural graphics system encodes pictures by assigning an address and colour attribute for **each** point of the object - resulting in a long list of addresses and attributes. The problem is alleviated by using a newer class of geometrical shapes which are both flexible and controllable. These geometrical shapes posses the property that they can be made to conform to clouds, feathers, leaves and other natural objects. These shapes may be found in the domain of fractal geometry.

Iterated Function Systems (IFS) and their use in Image Compression

Using fractals to simulate natural effects is not new. The innovation is to start with an actual image and **find** the fractals that will imitate it to the required degree of accuracy. Since these fractals are represented in a compact way, the whole image will be represented by a highly compressed data set, thus data compression is achieved.

Iterated Function System (**IFS**) codes are used to represent the fractal transforms. IFS codes use affine transformations to express relations between parts of an image. They are able to define and convey intricate details of a picture.

Fractal compression is a lossy compression technique. The high compression ratio may be increased further by applying the best lossless compression algorithm currently available to the IFS codes itself.

Affine Transformations

These are combinations of rotations, scalings and translations of the co-ordinate axes in *n*-dimensional space. Fig. 2 shows an example of an affine transformation, *W*, operated on a smiling face, *F*, lying on the *xy* plane, moving it to a new face, *W*(*F*). *W* always moves points closer together - it is contractive.

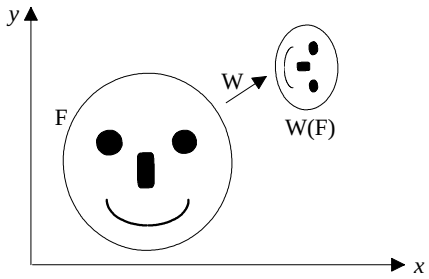


Fig.2. An example of an affine transformation W.

General form for an affine transformation

$$W\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} ax+by+e \\ cx+dy+f \end{pmatrix}$$

If the translations, rotations, and scalings that make up *W* are known in advance, then the coefficients may be calculated by:

$$a = r\cos\theta \qquad b = -\sin\phi$$
$$c = r\sin\theta \qquad d = s\cos\phi$$

where r = scaling factor on x, s = scaling factor on y

θ = angle of rotation on x, ϕ = angle of rotation on y

e = translation on x, f = translation on y

What is an IFS?

An IFS is a collection of contractive affine transformations which express relations between parts of an image. The relations are able to define and convey intricate details of a picture. An example of an IFS code for the generation of a fractal fern leaf is given below, consisting of four affine transformations, in matrix form:

W	***a***	***b***	***c***	***d***	***e***	***f***	***p***
1	0	0	0	0.16	0	0	0.01
2	0.2	-0.26	0.23	0.22	0	1.6	0.07
3	-0.15	0.28	0.26	0.24	0	0.44	0.07
4	0.85	0.04	-0.04	0.85	0	1.6	0.85

In general, an IFS consists of m affine transformations, W_1, W_2, . . ., W_m, with an associated probability each. The probabilities affect the rate of filling-in of the various regions and attributes of the image.

Complex form of IFSs

The points (x, y) in the real 2-D space may be thought of as points z in the complex plane **C**, then the affine transformation ω_i can be expressed in the complex form $\omega_i(z)$ where:

$$z = x + \mathbf{i}\, y \text{ and}$$

$$\omega_i(z) = c_i\, z + (d_i\, z^*) + b_i, \quad i = 1, \ldots, N, \quad \text{where } z^* = x - \mathbf{i}\, y.$$

If this is compared with the polar form of the affine transformation then the complex variables of the affine transform are found to be:

$$c_i^r = \tfrac{1}{2}(r\cos\theta + s\cos\phi) \qquad r^2 = (c_i^r + d_i^r)^2 + (c_i^c + d_i^c)^2$$

$$c_i^c = \tfrac{1}{2}(r\sin\theta + s\sin\phi) \qquad \tan\theta = (c_i^c - d_i^c)/(c_i^r + d_i^r)$$

$$d_i^r = \tfrac{1}{2}(r\cos\theta - s\cos\phi) \qquad s^2 = (c_i^r - d_i^r)^2 + (c_i^c + d_i^c)^2$$

$$d_i^c = \tfrac{1}{2}(-r\sin\theta + s\sin\phi) \qquad \tan\theta = (c_i^c + d_i^c)/(c_i^r - d_i^r)$$

$$b_i^r = e$$

$$b_i^c = f$$

The probabilities stay the same in either case.

Relationship between Complex Moment and Complex form of IFS

The moment of an IFS is defined[4] by:

$$M_{nb} = \int_k z^n d\mu(z) \qquad n = 0, 1, 2, \ldots$$

where z = points generated by affine transformation ω_i.

Consider our fractal image to be made up of m points z_k, then moment M_n is:

---

[4] M.F. BARNSLEY, V. ERVIN, D. HARDIN, J. LANCASTER, "Solution of an inverse problem for fractals and other sets", **Proc. Natl. Acad. Sci.**, USA, Vol. 83, April 1989, p.1975-77.

$$M_n = \sum_{k=1}^{m} (z_k)^{\frac{n}{m}}$$

For an IFS with $r = s$ and $\theta = \phi$ (e.g. of the form $\omega_i(z) = a_i\, z + b_i$, $i = 1, \ldots, N$) then:

$$M_n = \sum_{i=1}^{N} p_i \int_k (a_i z + b_i)^n d\mu(z)$$

expanding using the binomial theorem:

$$M_n = \sum_{i=1}^{N} p_i \sum_{j=0}^{n} ({}^nC_j) \int_k (a_i z)^j b_i^{n-j} d\mu(z)$$
$$= \sum_{i=1}^{N} p_i \sum_{j=0}^{n} ({}^nC_j) a_i^j b_i^{n-j} \int_k z^j \, d\mu(z)$$

where ${}^nC_j = [n(n\text{-}1), \ldots, (n\text{-}j\text{+}1)] \,/\, j!$ and $j! = j(j\text{-}1)(j\text{-}2) \ldots 1$

Knowing that $\int_k z^j d\mu(z) = M_j$, M_n can be simplified to:

$$M_n = \sum_{i=1}^{N} p_i \sum_{j=0}^{n} ({}^nC_j) a_i^j b_i^{n-j} M_j$$
$$= \sum_{i=1}^{N} p_i \sum_{j=0}^{n-1} ({}^nC_j) a_i^j b_i^{n-j} M_j + \sum_{i=1}^{N} p_i a_i^n M_n$$

An equation is obtained giving M_n in terms of the previous moments M_j, $j = 0,..., n\text{-}1$ and the form of the affine transformation a_i, b_i, and p_i, $i = 1, \ldots, N$

Since $M_0 = 1$, the rest of the moments may be calculated, without the need to generate the points z_k. Thus saving valuable computation time where 10,000 or more points may need to be generated to obtain accurate moments.

Thus an IFS that describes an image may be found by attempting to make the moments of the IFS as close to the moments of the image. This only holds for the case $r = s$ and $\theta = \phi$.

For the more general case, the general moment definition has to be used:

$$M_{jk} = \int_k z^j z^{*k} \, d\mu(z) \qquad j, k = 0, 1, 2 \ldots$$

In this case a matrix equation for the moments M_{jk} with $j+k = n$ has to be solved, this is in the form of[5]:

$$-[C] = ([A] - [I])\,[M_{array}]$$

---

[5] D. WILSON, "Fractal Image Compression", M.Sc. Thesis, Sept 1988, Dept. of Comp. Sci., Imperial College of Science, Technology & Medicine, Univ. of London, U.K.

where $[M_{array}]$ = vector $(M_{0n}, M_{1(n-1)}, M_{2(n-2)},...,M_{n0})^T$,

$[A]$ = matrix of dimension $(n+1) \times (n+1)$ whose elements are IFS parameter dependant,

$[I]$ = Identity matrix, I_{ii} 1, $I_{ij} = 0$ if $i \neq j$,

$[C]$ = IFS parameters and moments M_{jk} (where $j+k < n$) dependant vector.

Thus using $M_{00} = 1$ and the IFS code, the moment M_{jk} can be solved.

1) The moment library search method

The moments have to be normalised so that fractal images that are the same except for a global scaling may be compared. By having a large database of IFS codes and its associated normalised moments, this library may be used to search for an IFS code whose moment is closest to the normalised moment of an image to be encoded.

This IFS code may then be retained and passed as the fractal transform of that image segment to use directly to compress that segment or instead the IFS code obtained may be used as a starting point to some non-linear solution method to find a closer IFS code to that segment.

2) Newton's Method to find an IFS code close to an image

Newton's method can be used to solve an equation of the form: $f\left(\vec{x}\right) = 0$. The problem is essentially:

$$f\left(\vec{x}\right) = f_1\left(\vec{x}\right) - f_1\left(\vec{x}_{\text{image segment}}\right)$$

where $f_1\left(\vec{x}\right)$ = normalised moments function of an IFS code

$f_1\left(\vec{x}_{\text{image segment}}\right)$ = normalised moments of an image calculated explicitly from the points of the image.

When $f\left(\vec{x}\right) = 0$, then the vector form of the IFS code $\vec{x}$, has been found.

3) Simulated Annealing[6] method to find an IFS code close to an image

This is a better method of minimising functions of many variables as it will not go immediately to the local minima of a function - a problem inherent with the Newton method.

The problem concerns the thermodynamics of metal cooling and annealing given by the equation:

$$\text{Prob}(E) \approx \exp\left(\frac{-E}{kT}\right)$$

where E = energy of system

T = temperature (Kelvin)

k = Boltzmann's constant.

The method requires parameters that are analogues of T whose value is lowered as the method gets closer to the minima, and energy E, where energy is the value of the system to be minimised.

At initial higher Ts, changes to higher energy states are much more likely to be accepted and it is this feature of the method that allows the algorithm to find the global minima of a function rather than one of many local minima.

The method may be used to find an IFS whose moments are close to a given set of moments. The following are required:

1) Description of system - use vector $\vec{x}$ of size N_{ofx}, where N_{ofx} is the number of points in the image segment

$$\vec{x} = \left(c_1^r, c_1^c, d_{1,}^r d_1^c, b_1^r, b_1^c, \ldots, d_{N_{Affine}}^r, d_{N_{Affine}}^c, b_{N_{Affine}}^r, b_{N_{Affine}}^c\right)^T$$

2) A random system change generator. Accomplished by random vector $\vec{dx}$, a variable of length randomly chosen between 0 and the given length δ_l. δ_l and T were lowered simultaneously creating a new vector $\vec{x}_{new}$:

$$\vec{x}_{new} = \vec{x}_{old} + \vec{dx}$$

The vector was checked for valid IFS code production, if the code obtained was invalid then a new $\vec{dx}$ was generated.

---

[6] S. KIRKPATRICK, C.D. GELATT JR., M.P. VECCHI, "Optimisation by simulated annealing", **Science**, Vol. 220, No.4598, 13 May 1983, p.671-679.

3) The energy of the system, *E*, whose minimisation was required, was substituted by the function:

$$E = \left| f(\overrightarrow{x}) \right|^2 = \sqrt{\sum_{i=1}^{N_{ofx}} f_i(\overrightarrow{x})^2}$$

where $f\left(\overrightarrow{x}\right) = f_1\left(\overrightarrow{x}\right) - f_1\left(\overrightarrow{image}\right)$. Unnormalised moments were used for $f_1\left(\overrightarrow{x}\right)$ and $f_1\left(\overrightarrow{image}\right)$.

4) The parameter *T* and a method of lowering *T*. *T* governs the changes in the function *E*. The value of *E* should be considered carefully as it affects the energy changes, ΔE, that can be acceptable.

How to decode an IFS code - the Random Iteration Algorithm[7]

1) Initialisation: $x = 0$, $y = 0$.

2) For $n = 1$ to Number_of_points_ in_image, do steps (3) and (4).

3) Choose k to be one of the numbers 1,2, . . ., m, with probability p_k.

4) Apply the transformation W_k to the point (x,y) to obtain (x', y').

5) Set (x,y) equal to the new point: $x = x'$, $y = y'$.

6) If $n >$ number_points_required_to_obtain_attractor, then plot (x',y')

7) Loop.

More points are added to an image by increasing the variable Number_of_points_ in_image. This may be required to obtain an image with a greater resolution. Zooming may also be achieved by using an increased scale factor. The variable of (6) is around 100, but may be minimised by empirical methods.

The ability of the random iteration algorithm to produce the same image time and time again independent of the random sequence of events chosen has been proven in two ways:

1) By carrying out computer-graphical maths experiments;

2) By rigorous theoretical foundation of the mathematician John Elton of Georgia Institute of Technology (Atlanta, USA).

---

7 BARNSLEY AND SLOAN, *op. cit.*

Acknowledgements

I would like to thank Dr. T.G. Clarkson (King's College London), Dr D.M. Monro (formerly with Imperial College of Science, Technology & Medicine, London, now Bath Univ., UK.), Dr. M. Gennert (Worcester Polytechnic Institute, MA, USA), Thomas Landgraf (Gesamthochschule Kassel, Germany).

References

(i)BARNSLEY, M.F., ERVIN, V., HARDIN, D., AND LANCASTER, J., "Solution of an inverse problem for fractals and other sets", **Proc. Natl. Acad. Sci.**, USA, Vol. 83, April 1989, p.1975-77.

(ii)BARNSLEY, M.F., AND SLOAN, A.D., "A better way to compress images", **BYTE**, Jan 1988, p.215-223.

(iii)KIRKPATRICK, S., GELATT JR., C.D., AND VECCHI, M.P., "Optimisation by simulated annealing", **Science**, Vol. 220, No.4598, 13 May 1983, p.671-679.

(iv)OPPENHEIMER, P.E., "Real Time Design and Animation of Fractal Plants and Trees", **Computer Graphics** (Siggraph '86), Vol.20, No.4, 1986, p.55-64.

(v)PEITGEN, H.-O., AND SAUPE, D., *The Science of Fractal Image*, Springer Verlag, New York, U.S.A., 1988, Fig.1.4, p.29.

(vi)WILSON, D., "Fractal Image Compression", M.Sc. Thesis, Sept 1988, Dept. of Comp. Sci., Imperial College of Science, Technology & Medicine, Univ. of London, U.K.

(vii)ZORPETTE, G., "Fractals: not just another pretty picture", **IEEE Spectrum**, Oct 1988, p.29-31.