Appendix D Analysis, Generation and Compression of Pavement Distress Images Using Fractals

The paper in this appendix was presented at the British Computer Society's (BCS) International State-of-the-Art Seminar - The Applications of Fractals and Chaos, London, U.K., 19-20 Feb., 1992. The paper appeared both as a proceeding and as a chapter in a book, the full references of which are given below:

Ali, M., Gennert, M. A., and Clarkson, T. G., "Analysis, Generation and Compression of Pavement Distress Images Using Fractals", *Proc. of the British Computer Society's (BCS) International State-of-the-Art Seminar - The Applications of Fractals and Chaos*, London, U.K., 19-20 Feb., 1992.

Ali, M., Gennert, M. A., and Clarkson, T. G., "Analysis, Generation and Compression of Pavement Distress Images Using Fractals", *The Applications of Fractals and Chaos*, eds. A. J. Crilly, R. A. Earnshaw and H. Jones, Springer-Verlag, Berlin, 1993, p.147-169. ISBN 3-540-56492-6, ISBN 0-387-56492-6.

ANALYSIS, GENERATION AND COMPRESSION OF PAVEMENT DISTRESS IMAGES USING FRACTALS

Maaruf Ali, Michael A. Gennert[333] and Trevor G. Clarkson

Abstract

The vast amount of data generated by automated surface distress evaluation equipment far exceeds the storage capabilities of current digital data storage systems. A study using fractals is being carried out to alleviate the data storage problem, since fractal image compression offers the largest compression ratio of the available image compression algorithms. This paper discusses the use of fractals to analyse, compress and generate pavement distress features, i.e., cracks in the road surface. Much of the following is abridged from a paper by LeBlanc [LeBl91]. A method for calculating the fractal dimension of cracks is presented and values for pavement cracks reported. Several methods for fractal image compression are explained, especially the midpoint displacement algorithm to generate pavement distress images and iterated function system codes. The use of fractal techniques to generate standard images for testing an automated surface distress evaluation system is proposed.

Keywords: Fractal Compression, IFS, Pavement Distress Modelling, Fractal Dimension.

[333] The authors are with King's College London, except, Dr. Gennert, who is affiliated with the Computer Science Department, Worcester Polytechnic Institute, 100 Institute Rd., Worcester, Massachusetts 01609-2280, U.S.A.

Introduction

The deterioration of transportation systems in the United States is a problem of major concern to local, state and federal agencies and to the public. Highways in the United States are deteriorating at an alarming rate due to the normal ageing processes, as well as being subject to greater and more severe traffic loads. This problem is compounded by decreases in available funding for restoration of this vital element of the infrastructure. Improvements are not predicted as maintenance and construction costs rise due to inflation in material and labour costs and as revenues decline. The need for a pavement management system to provide accurate assessment of the condition of various types of highway pavement is of critical importance in addressing highway maintenance needs.

Pavement management systems require systematic monitoring of pavement surface to determine preventive and corrective maintenance. The process involves accumulation of large amounts of visual data, typically obtained from site visitation. The pavement surface condition is then correlated to a pavement distress index that is based on a scoring system previously established by the various state Departments of Transportation (DOTs). The scoring system determines if the pavement section requires maintenance, overlay or reconstruction.

A large amount of data is required to establish a pavement distress index. The computation process is currently computerised. However, the raw data are still input manually, a laborious and expensive task. This, coupled with the difficulty and impracticality associated with field measurements of pavement surface cracks in areas

with high traffic density, emphasises the need for an automated visual crack measuring device.

The objective of this project is to develop a system for processing video images of pavements and to identify, quantify, and classify pavement distress. The basic strategy for automated acquisition and analysis of video images for evaluating pavement distress is straightforward: a vehicle (see Figure 1) travels along the road taking pictures of the pavement, which are analysed to evaluate the type, severity and distribution of surface cracks and patches.

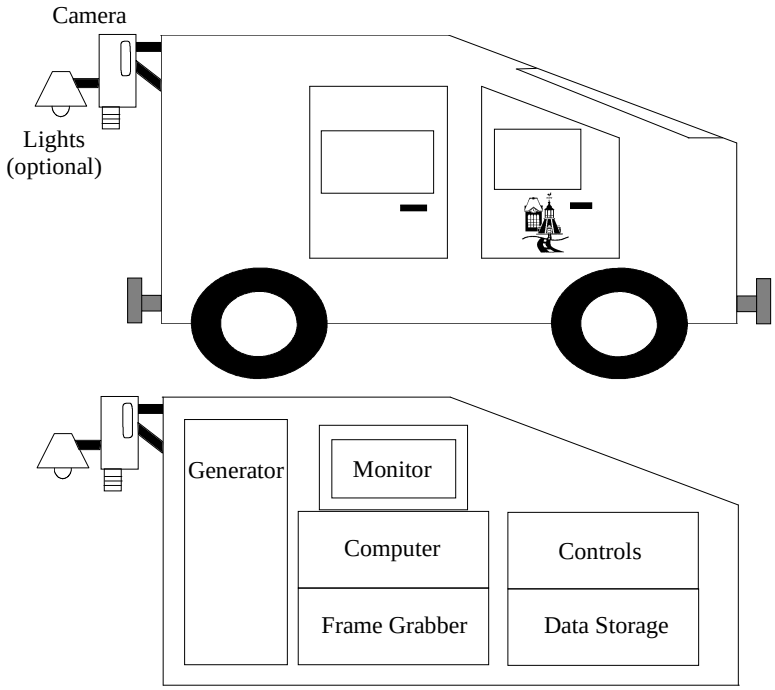


Figure 1. Schematic drawing of a pavement data collection system [LeBl91].

Road images are taken by a moving vehicle where the data are analysed in real time; only summary information about distress severity and extents is recorded (see Table 1). This solves the data rate and storage problem, but since the raw data is discarded many useful functions cannot be carried out by the pavement maintenance engineer. These functions include, for example, evaluating the cause of distress and comparison of records to determine pavement wear. The second approach retains the

raw data by using commercial video cameras and recording equipment for data analysis at a later time. But the major problem of using currently available recording equipment is that its limited bandwidth is incapable of meeting the system specifications, i.e., low resolution images.

A solution elaborated in this paper is to use a special purpose high resolution image acquisition system with data compression to allow recording raw image data at moderate road speeds (45 mph) using commercial equipment. The compressed images are stored and reconstructed at a later time for evaluation by the pavement engineer or can be input directly for image processing by automated machine vision systems.

Fractal coding is chosen because it satisfies the condition of image equivalency and it offers a very high compression ratio. Human to computer evaluation is not considered. Equivalency in terms of this project is defined to be:

> the reconstructed image is equivalent to the original uncompressed image as judged by humans;
> machine vision image processing determines that both the uncompressed and compressed images show the same distress, distress extent and severity.

This paper reports ongoing research being conducted by Worcester Polytechnic Institute, Massachusetts, USA, and King's College London, University of London, England, in using fractals to compress images. Initially, simple distress features are being modelled, namely simple longitudinal, transverse and diagonal cracks. Complete image reconstruction accuracy is not the objective of the work. Sufficient data must be retained to allow accurate assessment of severity and extent of pavement distress by both humans and computers.

Data Reduction

THE NEED FOR DATA REDUCTION

The volume of data to be processed is enormous. If P is the number of points per square inch to be analysed and assuming a 14-foot lane width, 10,644,000 × *P* points per mile of highway are needed. If 10,000 miles of highway per year are to be covered (not unreasonable for many states) at a resolution of 1/16 inch (to guarantee detection of cracks 1/16 inch wide) with *P* = 256, 27.1 trillion [10^{12}] points per year need to be processed. To acquire this data at 45 miles per hour would need 222 hours of road time; thus, data acquisition can be safely restricted to periods of favourable weather. Data would need to be acquired at a rate of 33.9 million points per second. This can be processed off-line at a much lower rate, as computers can run 365 days a year, 24 hours per day. Thus, 8760 hours are available to process data that required 222 hours to acquire, for an effective rate of 0.86 million points per second.

Table 1. Simplified Inspection System Technical Specifications

Maximum Inspection speed:	45mph (72km/h)
Inspection width:	14 feet (4.3m)
Road coverage:	continuous (100% inspected within a lane)
Minimum crack width:	1/16 inch (1.6mm)
Maximum joint width:	1 inch (25mm)
Pavement Surface:	bituminous or portland cement concrete
Pavement condition:	new to severely worn
Aggregate:	any material, not worn shiny-smooth
Lighting conditions:	any natural combination of sun and skylight

DATA REDUCTION METHODS

Ignoring Other Than Pavement Distress Features

One type of information that does not contribute to pavement distress evaluation is image texture due to aggregate. Aggregate has a signature that is easy to characterise: it has a roughly circular texture at a regular frequency. Methods for detecting and eliminating most, if not all, of the signal directly attributable to aggregate are explored. This enables information storage requirements to be reduced to four bits per pixel or less, thus halving storage requirements at the cost of some decoding prior to processing.

Further compression of the aggregate-free images is possible using the simple but powerful method of image encoding known as Run Length Encoding. In Run Length Encoding, consecutive runs of pixels with identical values are efficiently represented by storing the intensity value only once, with a repetition count telling how many times the given value is to be repeated. On a typical image this results in a compression ratio of 10 to 100 over the original image, and with filtering provides even greater compression.

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.

Fractals

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 that are both flexible and controllable. These geometrical shapes can be made to conform to clouds, feathers, leaves and other natural objects and are found in the domain of fractal geometry.

ADVANTAGES OF USING FRACTAL TRANSFORMS

A Fractals is described as ". . . a highly complex structure . . . 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.)" [Oppe86]

FRACTALS — A BRIEF INTRODUCTION

Coastlines, mountains and clouds are not easily described by traditional Euclidean geometry. We use Mandelbrot's fractal geometry to describe and mathematically model the natural objects. This is another reason image compression using fractal transforms is studied. Mandelbrot first coined the word fractal in 1975 [Mand75].

PROPERTIES OF FRACTALS

The property of self-similarity, or scaling, is one of the central concepts of fractal geometry. Also, a fractal has a fractional dimension, from which the word fractal is derived; Euclidean shapes have integer dimensions only.

Self-similarity

The property of objects whereby magnified subsets appears similar or identical to the whole and to each other is known as self-similarity. It sets fractals apart from Euclidean shapes which generally become smoother. Thus, fractal shapes are self-similar, independent of scale or scaling and possess no characteristic size. This describes a pavement distress image: a magnified segment of a crack appears the same as the unmagnified segment. Thus cracks are effectively represented by fractals.

Fractal Dimension

The classic example of a self-similar curve is that of a coastline, the length of which increases as the dimension of the ruler used to measure it decreases. Thus, unlike a straight line a fractal's measured length depends on the ruler's length. This dependence is described by the fractal dimension, *D*. Other fractal dimension measures exist, such as the similarity dimension. A detailed discussion of fractal dimension is beyond the scope of this paper. For pavement analysis purposes, a method for calculating the fractal dimension of cracks is required and presented, and conversely, the generation of cracks given a fractal dimension.

The distance or the number of rulers, *N*, obtained by measuring a straight line, *L*, using a ruler of length *K* is

$$N = \frac{L}{K}$$

When the curve is not straight but convoluted, *N*, the number of rulers needed to fit the curve, grows by an exponential factor, *D*, as

$$N = \left[\frac{L}{K}\right]^{D}$$

For a straight line, $D = 1$. D must be less than 2, otherwise the line has space and contradict the Euclidean definition of a line.

Measuring the Fractal Dimension of a Crack

We measure the fractal dimension of a crack using a modified Calliper method, which has produced the most consistent results and is most suitable for measuring the fractal dimension of linear structures.

First, segment the distress features from the pavement image. An example of a Portland Cement Concrete (PCC) pavement image is shown in Figure 2. Figure 3 shows the segmented image of the distress. Here the pavement's grey level is altered to white, whilst those pixels associated with the distress itself are black. Segmentation is a major problem in itself; ongoing research is being conducted at Worcester Polytechnic Institute to segment an image more efficiently.

Figure 2. Image of a distressed PCC pavement [LeBl91].

Next, skeletonize the segmented image, that is, reduce the width of the pixels to a single pixel. This is carried out in order to determine the morphology of the distress.

Figure 4a shows a segmented transverse crack taken from Figure 3, Figure 4b show the same crack after skeletonization.

Figure 3. Segmented image of the distress in Figure 2 [LeBl91].

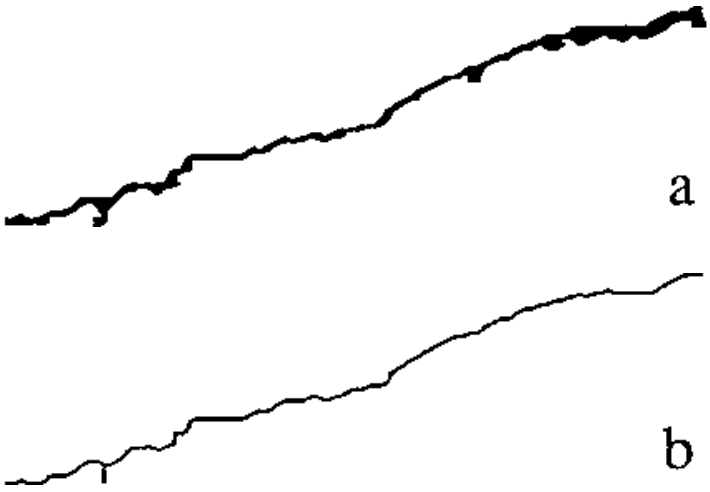


Figure 4. Transverse cracks (a) from Figure 3; (b) same crack skeletonized [LeBl91].

Then, the measured length, *M*, in pixels of the skeletonized crack is determined using a variety of ruler lengths, *K*, also in pixels. Following Smith [Smit89], a log-log plot of *M* (log curve length) as a function of *K* (log ruler length) is plotted, as shown in Figure 5.

Finally, the slope S is computed, to give the fractal dimension D, from the expression

$$D = 1 - S$$

The value obtained for this example is $D = 1.08$.

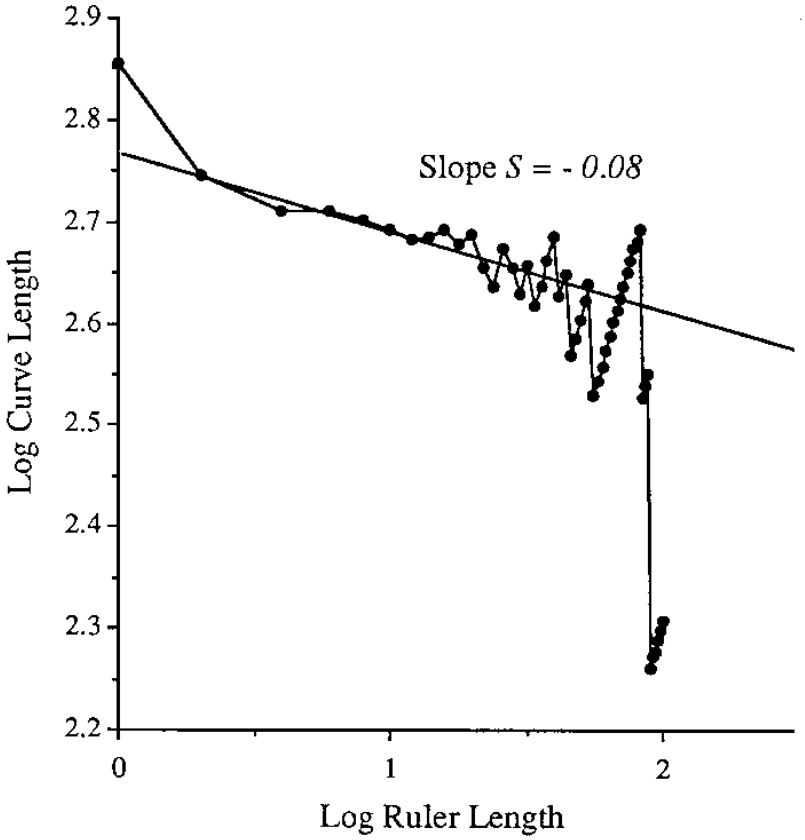


Figure 5. Log-log plot of curve length versus ruler length from Figure 4b [LeBl91].

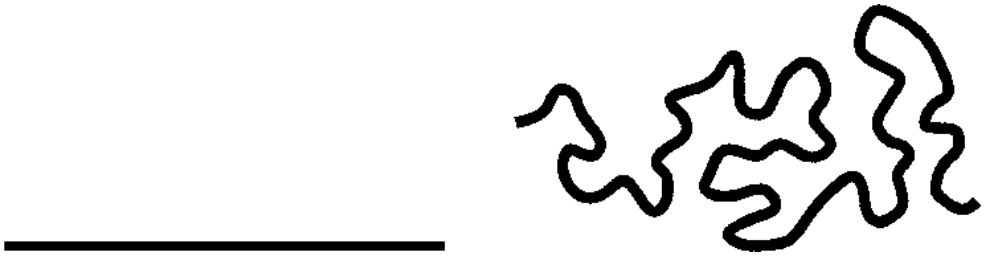
For small K values, the discreteness of the digital image means the left portion of the curve is generally flat, since pixels are larger than points and self-similarity breaks down at the pixel level. For large values of K the behaviour of the curve is explained by quantization error because the length is not an integral number of ruler lengths.

Point selection for determining the slope was arbitrary; ruler lengths of 10 to 100 pixels give the most repeatability. A more rigorous slope measurement method must be developed to fully automate fractal dimension measurements.

From the study of both PCC and asphalt cement (AC) pavements, the fractal dimension was found to lie in the range

$$D = 1.10 \pm 0.05$$

Joints in PCC pavements have fractal dimensions below this range. Curves with fractal dimensions above this range are too convoluted. See Figure 6.



$D = 1.0$ $D = 1.3$

Figure 6. Curves with fractal dimensions outside the range $D = 1.1 \pm 0.05$ [LeBl91].

Due to the difficulty of segmenting distress features in digitised pavement images (about 20), the population of the data is not large enough to derive statistical information concerning the fractal dimension, such as the standard deviation, nor to claim that the fractal dimension range is universal. No theories for this observation have yet been formulated.

The Simulation of Pavement Surface Distress Using Fractals

The first technique to be described generates fractals by iterative applications of ‘generator’ functions to Euclidean objects. In the case for the simulation of pavement distress, a straight line, the Euclidean object, is dissected into smaller line segments

by the generator. The resulting line segments are further dissected by the same generator, until the dissections are smaller than the screen or output device resolution. No general theory exists in designing or selecting the required optimal fractal generator to simulate pavement distress. The approach undertaken was to search for the generator and build up a library of generators that produce the different types of distress by subjective comparison of the results with the real distress.

A technique described later in this paper attempts to find the appropriate iterated function system (IFS) code to generate the distress by the moment matching method.

Fractal Generating Functions

DETERMINISTIC FRACTAL GENERATING FUNCTIONS

A Koch snowflake is a classic example of a fractal produced by deterministic fractal generating functions. The fractal is brought into existence by applying the generator shown in Figure 7b at each iteration. The generator trisects a line segment, as shown in Figure 7a, replacing the central segment with two segments of equal length after the application of the generator G times. The curve produced for $G = 3$ is shown in Figure 7c. The resulting curve has 3^G segments. The curve is only a fractal as G approaches infinity. For practical purposes G must be larger than the logarithm, base 2, of the image size in pixels. If the image size is 2^8 pixels wide, then G need only be equal to or greater than 9.

What most people do not understand is that the human eye is incapable of ever seeing a fractal. It cannot resolve the infinite detail present in a fractal, only an approximation of the fractal is sensed.

The Koch snowflake does not resemble a pavement distress in any way. Two reasons for this observation are:

The fractal dimension of the Koch snowflake is greater than the measured range of pavement distress dimensions and is given as (see [Voss88])

$$D = \frac{\log[N]}{\log\left[\frac{1}{R}\right]}$$

where N is the number of segments into which the line is dissected, and R is the ratio of the new line segment length to the original line length. For the Koch snowflake $N = 4$, $R = 1/3$ when $D = 1{\cdot}262$ (3 d.p.), which is outside the distress dimension range.

The Koch snowflake is too regular.

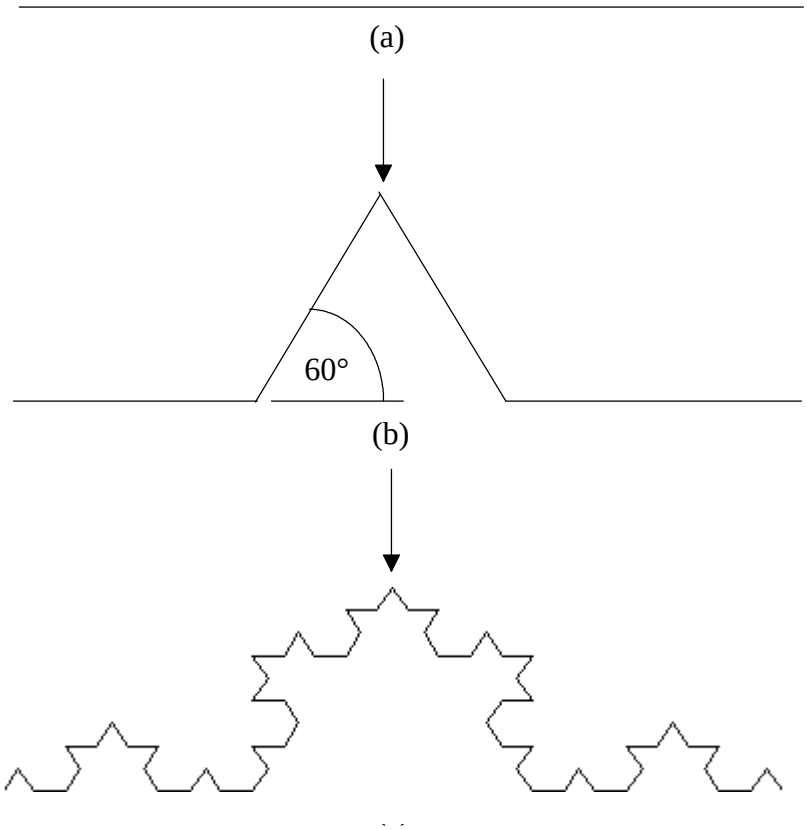


Figure 7. An example of a deterministic fractal generating function. (a) Line segment used for generating a fractal; (b) Koch snowflake generator; (c) curve after three iterations of the generator [LeBl91].

From these observations we deduce that control of the fractal dimension is required in order to produce more irregular fractals and fractals whose fractal dimension lies in the range where distress fractals exist.

STOCHASTIC FRACTAL GENERATING FUNCTIONS — STOCHASTIC MIDPOINT DISPLACEMENT ALGORITHM

Irregular fractals are produced by randomly varying the lengths and orientations of the basic generator function at each iteration. In the deterministic case, the generator bisects the line segment by displacing the midpoint to produce two new line segments. The displacement is always the same, as shown in Figure 8a. This generator function displaces the midpoint in an orthogonal direction by a quarter of the original line length. Figure 8b shows the case for the stochastic midpoint displacement generator, where both the magnitude, r, and the direction, θ, of the displacement are selected randomly. The midpoint is only displaced uniformly within the closed circle of radius p. The displacement angle, θ, is therefore uniformly distributed within the circle with the probability density of the angle, θ, given by

$$P_\theta(\theta) = \frac{1}{2\pi}$$

To obtain a uniform distribution, the probability density of any displacement magnitude, r, is

$$P_r(r) = \frac{2r}{p^2}$$

The equation used to calculate the Koch snowflake is still used to calculate the fractal dimension. In this case, where $N = 2$ and $R = E\{l_1\}$ the average or expected value of l_1. l_1 is the length of the line P_1M', the line connecting an endpoint to the

displaced midpoint in Figure 8b. The other line segment, P_2M', obeys the same statistics; the expected lengths are equal, and this is given by

$$E\{l_1\} = E\{l_2\} = \int_0^P \int_0^{2\pi} P_r(r) P_\theta(\theta) l_1(r,\theta)\, dr\, d\theta$$

From the geometry of Figure 8b l_1 is

$$l_1(r,\theta) = \sqrt{0.25 + r^2 + r\cos\theta}$$

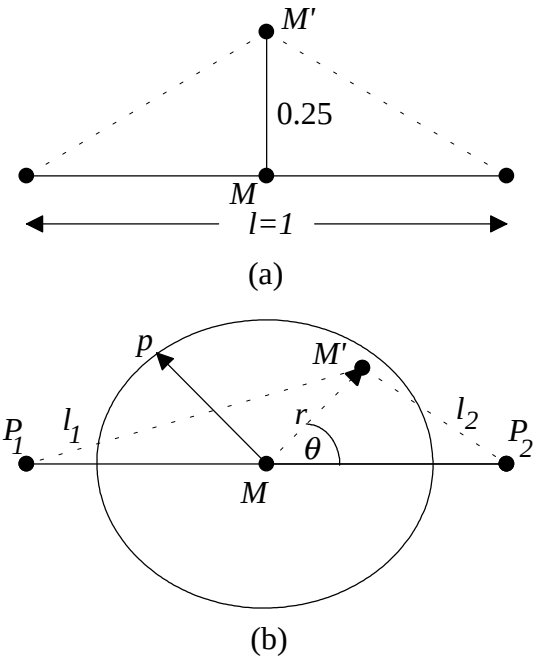


Figure 8. Midpoint displacement generators: (a) Deterministic, (b) stochastic [LeBl91]

A plot of the fractal dimension, *D*, against the parameter *p* is given in Figure 9.

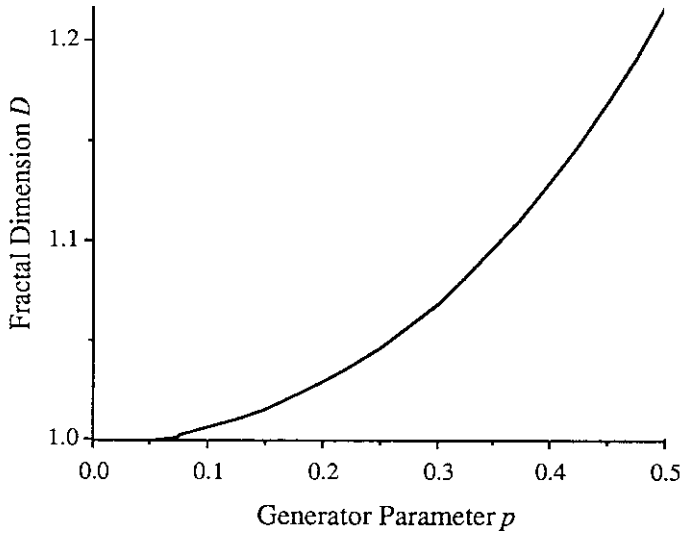
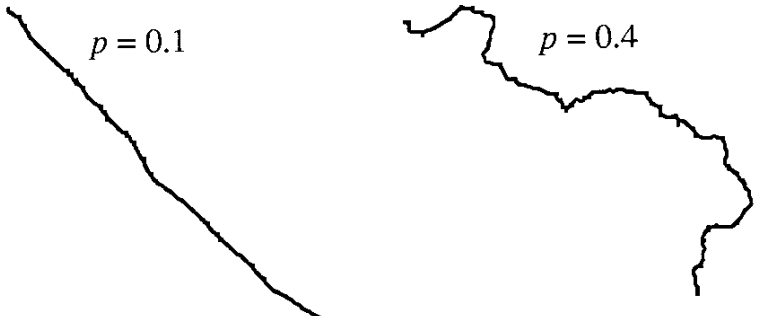


Figure 9. Fractal dimension as a function of the generator parameter [LeBl91].

The values of p, found to lie in the range: $0.25 \leq p \leq 0.425$, produce cracks whose fractal dimensions lie in the range of pavement distress. For p values greater than 0.5, the curves are too convoluted to resemble pavement distress; above 0.85, the fractal dimension exceeds two, and the curves are area-filling in nature. Conversely, for p values less than 0.25, the curves are too straight. Some curves obtained by using this algorithm are given in Figure 10.



(a) (b)

Fig. 10. Fractals produced by a stochastic midpoint displacement generator [LeBl91].

Crack Width Algorithm

The fractal curve produced above had no width. To render the curve it was necessary to assign a pixel width of one or two. With this the curve had constant width, which is contrary to real pavement distress. To obtain a more realistic distress, a crack width algorithm is implemented. The input parameters for this are the width at one end of the crack and the minimum and maximum widths. Points are constructed at both segment ends a constant perpendicular distance from the segment by defining a constant width box around the segment. The width of the box is randomly generated and is constrained by the upper and lower width boundaries. Too, it cannot vary by more than a pixel from the previous line segment of the fractal curve. To make the curve appear continuous and with no abrupt width changes, the box endpoints are

connected end to end by shaped polygons. The final stage consists of replacing the line segments by the boxes and polygons to render the final crack. Figure 11 shows the crack of Figure 10b after application of the crack width algorithm. No theoretical foundation exists for this algorithm; its use is purely for the purpose of generating realistically looking cracks.

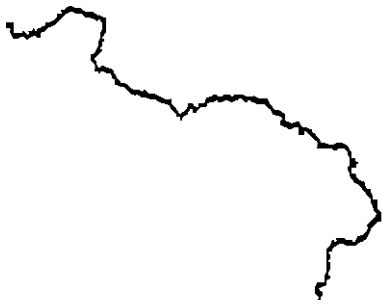


Figure 11. A fractal curve with algorithmically produced width [LeBl91].

Generating Pavement Images and Image Compression

Real distress features not only have a nonconstant width but also a nonconstant grey level with colour variation as well. The grey level of the pavement depends upon, for example, the material, lighting conditions and the geometry of the distress. To simulate new pavement images to reflect lighting or material, other pavement images can be combined and processed with the application of computer models to compute the correct grey levels. To construct an image this way (see Figure 12) we first use the segmented image of Figure 3 as a mask to remove the distress from the image of Figure 2. Then we superimpose the computed distress of Figure 11 onto Figure 2, adjusting the grey levels. Only 13 parameters are required to achieve this:

the crack endpoint co-ordinates (xstart, ystart, xstop, ystop — four values);
the fractal dimension;
the minimum crack width;

the maximum crack width;
initial crack width;
the assumed crack depth;
the average pavement surface reflectivity;
two illumination angles;
the ratio of directed to ambient illumination.

Figure 12. Simulated image of a distressed PCC pavement [LeBl91].

Note that the last six parameters are only needed for synthesising crack images and might not be needed for crack analysis (and might be very difficult to recover). Thus one normally only need the first seven parameters for analysis. This holds only for simple cracks. For more complicated cracking patterns consisting of N crack segments, less than $7N$ parameters are needed, since crack endpoints are necessarily shared among multiple cracks. The data used is 1/500th of the data in the original image, thus an effective compression ratio of 500:1 is achieved.

Iterated Function Systems (IFS)

IFS: 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 imitate it to the required degree of accuracy (see [Barn88; Zorp88]). Since these fractals are represented in a compact way, the whole image is 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 which express relations between parts of an image. They define and convey intricate details of a picture.

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

AFFINE TRANSFORMATIONS

Affine transformations are combinations of rotations, scalings and translations of the co-ordinate axes in *n*-dimensional space. Figure 13 shows an example of a contractive affine transformation, *W*, operated on a smiling face, *F*, lying in the *xy* plane and moving it to a new face, *W*(*F*). *W* always moves points closer together — it is contractive. The general form of an affine transformation is

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

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

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

$$c = r\sin\theta \qquad d = \;\; s\cos\phi$$

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

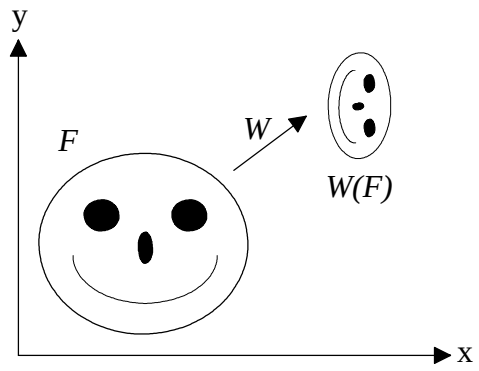


Figure 13. An example of a contractive affine transformation, *W*.

WHAT IS AN IFS?

An IFS is a collection of contractive affine transformations that express relations between parts of an image. The relations define and convey intricate details of a picture. An IFS code for the generation of a fractal fern leaf consisting of four affine transformations in matrix form is

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

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

THE COMPLEX FORM OF IFSs

The points (x, y) in the real 2D space can be seen as points z in the complex plane. Then the affine transformation, ω_i, expressed in the complex form $\omega_i(z)$ is

$$z = x + \boldsymbol{i}y$$

$$\omega_i(z) = c_i\, z + (d_i\, z)^* + b_i \qquad i = 1, \ldots, N$$

where $z^* = x - \boldsymbol{i}y$

Comparing this with the polar form of the affine transformation, the complex variables of the affine transform are

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

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

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

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

$$b_i^r = e$$

$$b_i^c = f$$

The probabilities stay the same in either case.

RELATIONSHIP BETWEEN THE COMPLEX MOMENT AND THE COMPLEX FORM OF IFS

The moment of an IFS is defined as [Barn86]

$$M_n = \int_k z^n \, d\mu(z) : n = 0, 1, 2\ldots$$

where z represents the points generated by the affine transformation ω_i.

Consider our fractal image to be made up of m points z_k. Then the moment, M_n, is

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

For an IFS with $r = s$ and $\theta = \phi$ (for example, of the form $\omega_i(z) = a_i\, z + b_i$, $i = 1,...,N$)

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 yields

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

where $${}^nC_j = \frac{n(n-1),\ldots,(n-j+1)}{j!}$$

and $$j! = j(j-1)(j-2)\ldots 1$$

Since $\int_k z^j d\mu(z) = M_j$, M_n is simplified to

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

Taking M_n to the LHS and rearranging yields

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

An equation is obtained giving M_n with the previous moments M_j, $j = 0, \ldots, n - 1$, 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 are calculated without the need to generate the points z_k. Valuable computation time where 10,000 or more points need to be generated to obtain accurate moments is thus saved.

Thus, an IFS that describes an image is found by attempting to make the moments of the IFS as close to the moments of the image as possible. This only holds for the case $r = s$ and $\theta = \phi$. For the more general case, the general moment definition is

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

Here, a matrix equation for the moments M_{jk} with $j + k = n$, must be solved in the form (see [Wils88])

$$-[C] = ([A] - [I])\,[M]$$

where $[M] = \text{vector } [M_{0n}, M_{1(n-1)}, M_{2(n-2)}, \ldots, M_{n0}]^T$;

[A] = matrix of dimension (n + 1) × (n + 1) whose elements are IFS parameter dependant;

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

[C] = vector whose elements depend upon the IFS parameters and moments M_{jk} (where $j + k < n$).

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

The Moment Library Search Method

The moments must be normalised so that fractal images which are the same except for a global scaling can be compared. By having a large database of IFS codes and its associated normalised moments, this library is used to search for an IFS code whose moment is closest to the normalised moment of an image to be encoded. This IFS code is then retained and passed as the fractal transform of that image segment to directly compress that segment; otherwise, the IFS code obtained is used as a starting point to a non-linear solution method to find a closer IFS code for that segment.

Newton's Method to Find an IFS Code Close to an Image

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

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

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

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

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

Simulated Annealing Method to Find an IFS Code Close to an Image

The simulated annealing method (see [Kirk83]) to find an IFS code close to an image is a better method of minimising functions of many variables because it does not immediately go to the local minimum of a function, a problem inherent with the Newton method. The problem concerns the thermodynamics of metal cooling and annealing given by

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

where E = the energy of the system;

T = temperature (Kelvin);

B = Boltzmann's constant.

The method requires parameters that are analogues of *T*, whose value is decreased as the method gets closer to the minimum, and energy, *E*, where *E* 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. It is this feature of the method that allows the algorithm to find the global minimum of a function rather than one of many local minima. The method is used to find an IFS whose moments are close to a given set of moments. The following conditions are required:

description of the system — use the vector $\vec{x}$ of size N_{ofx}, where N_{ofx} is the number of points in the image segment

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

a random system change generator — this is accomplished by the random vector, $\vec{dx}$, a variable of length randomly chosen between 0 and the given length, δ_l. δ_l and *T* were decreased 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;

the energy of the system, *E*, whose minimisation is required, is

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

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

the parameter *T* and a method of decreasing *T*; *T* governs the changes in the function *E*. The value of *E* should be considered carefully, since it affects the energy changes, ΔE, which are acceptable.

HOW TO DECODE AN IFS CODE

The Random Iteration Algorithm

A summary of the random iteration algorithm (see [Barn88]) given in pseudo-code form is:

1. initialise variables, $x = 0$, $y = 0$;
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_of_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 can be required to obtain an image with a greater resolution. Zooming

is also achieved by using an increased scale factor. The variable number_of_points_required_to_obtain_attractor is around 100 but can be minimised by empirical methods. The ability of the random iteration algorithm to produce the same image independent of the random sequence of events chosen is proven in two ways:

by carrying out computer graphic mathematical experiments;
by the rigorous theoretical foundation of the mathematician John Elton of Georgia Institute of Technology, Atlanta, Georgia, USA (see [Elto87]).

Conclusions

The attempts to simulate pavement distress using fractal techniques illustrate the problems of image equivalency. There is no objective test for image equivalency such as there is for image accuracy. The project illustrates that the fractal dimension can be used to assess image accuracy by using the fractal dimension to generate the fractal curve and by comparing the calculated fractal dimension of the same curve.

The paper shows the suitability of using fractal techniques for image compression and generation related to pavement distress, such as the variation of the midpoint displacement algorithm and IFS coding techniques. Pavement test images can also be generated and used for testing and developing automated pavement evaluation systems. The advantage of test images is that their geometric and photometric characteristics are known in advance. Further work is required to improve and integrate this new technology into an operational automated pavement surface distress evaluation system.

A modified method of calculating the fractal dimension (Calliper method) of linear structures has been explained. Table 2 shows the values of the generator, *p*, and its associated fractal dimensions, theoretical and measured. The maximum discrepancy between the measured and the calculated fractal dimension was 5.4%, whereas the least was only 0.2%! This shows the reliability of the method. The major discrepancy in the work can be attributed to the difficulty in determining the slope of the log-log plots, such as seen in Figure 5. This requires further research and development.

The major advantage of using fractal techniques is that they offer a very large image compression ratio. The work being conducted using the fractal dimension has given compression ratios of 500:1. This figure was obtained by comparing the information required to generate the distress with the raw data of the pavement that included the distress and the aggregate — most of the data is required to synthesise the aggregate. Since only distress data is of concern to the pavement maintenance engineer, a library of pavement surface images can be stored on a CD. When a pavement image is reconstructed from the compressed data, the distress is superimposed on the relevant aggregate image selected from the CD; hence, a large compression ratio of 500:1 or even higher is easily attained. A fractal method for synthesising the aggregate from the IFS code is being studied.

Table 2. Comparison Between Theoretical and Measured Fractal Dimension

Image	Generator 'p'	Generator 'D'	Measured 'D'	Percentage Error Between Measured & Calculated D
Figure 10a	0.1	1.01	1.06	+5.4%
Figure 10b	0.4	1.13	1.11	-1.7%
Figure 7c		1.26	1.26	-0.2 %

Using Iterated Function System codes — a type of fractal equation — to compress image segments has been explained. The difficult inverse problem of finding a suitable IFS code whose fractal image is to represent the real image and hence achieve compression is being studied through the use of

a library of IFS codes and complex moments;
the method of simulated annealing for solving non-linear equations of many parameters.

The application of simulated annealing is still under development.

IFS codes are robust, thus they are ideal for transmission through noisy distortion-inducing communication channels, since small deviation of the IFS codes still produce recognisable images with minimal distortion. Figure 14 shows an example of a pavement distress generated entirely from an ASCII file of IFS codes whose size is only 926 bytes.

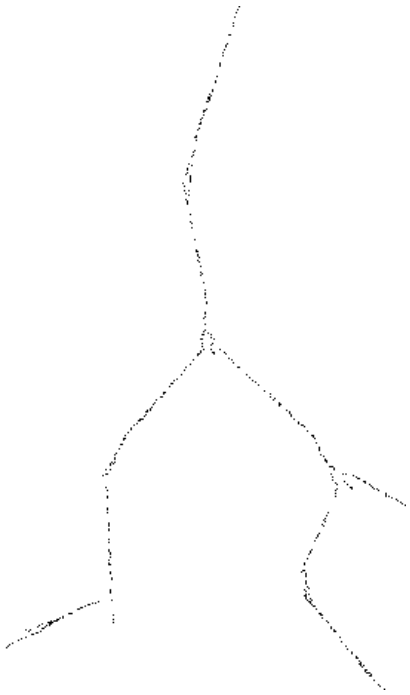


Figure 14. An example of an IFS-generated pavement distress.

Segmentation probably plays the most significant part in fractal image compression. The success of achieving a large compression ratio effectively rests on an efficient segmentation algorithm. Work being conducted at Worcester Polytechnic Institute on a fractal model-based segmentation technique is producing promising results. A basis of the algorithm is to determine the fractal dimension of the segment and compare it with the range of fractal dimensions of, for example, pavement distress features, to see if the segment is that of the object of interest. Block-based coding is also under investigation to avoid the problems associated with segmentation.

Acknowledgements

I thank Drs. N. Wittels, M. Ward and T. El-Korchi for their help and co-operation during my stay at Worcester Polytechnic Institute in Massachusetts, USA. A special thank you to my professor, Dr. Mike Gennert. This paper is an abridged version of [LeBl91]. I am also grateful to Prof. D. M. Monro (Bath University, England) for supplying the material by Wilson [Wils88].

References

[Barn86]

Barnsley, M.F., Ervin, V., Hardin, D., and Lancaster, J., “Solution of an inverse problem for fractals and other sets”, *Proc. Nat. Acad. Sci*., USA, Vol.83, p. 1975-77, April 1986.

[Barn88]

Barnsley, M.F., and Sloan, A.D., “A better way to compress images”, *BYTE*, January 1988, p.215-223.

[Elto87]

Elton, J.H., "An ergodic theorem for iterated maps", *Ergod. Th. & Dynam. Sys*., Vol.7, p.481-488, 1987.

[LeBl91]

LeBlanc, J., Gennert, M.A., Wittels, N., and Gosselin, D., "Analysis and generation of pavement distress images using fractals", *Transportation Res. Rec*., No.1311, p.158-165, 1991.

[Kirk83]

Kirkpatrick, K., Gelatt, C.D., Jr., and Vecchi, M.P., "Optimisation by simulated annealing", *Science*, Vol.220, No.4598, p.671-679, 13 May 1983.

[Mand75]

Mandelbrot, B.B., *Les objects fractals: forme, hasard et dimension*, Paris: Flammarion, 1975.

[Oppe86]

Oppenheimer, P.E., "Real time design and animation of fractal plants and trees", *Computer Graphics*, Vol.20, No.4, p.55-64, 1986 (Siggraph 86).

[Smit89]

Smith, T.G., Jr., Marks, W.B., Lange, G.D., Sheriff, W.H., and Neale, E.A., "A fractal analysis of cell images", *Journal of Neuroscience Methods*, 27, p. 173-180, 1989.

[Voss88]

Voss, R.F., "Fractals in Nature: From characterisation to simulation", *The Science of Fractal Images,* eds. Peitgen, H.-O., and Saupe, D., Springer-Verlag, New York, 1988, Fig.1.4, p.29.

[Wils88]

Wilson, D., “Fractal Image Compression”, computer science M.Sc. project report, September 1988, Imperial College of Science, Technology & Medicine, University of London, England.

[Zorp88]

Zorpette, G., “Fractals: not just another pretty picture”, *IEEE Spectrum*, October 1988, p.29-31.