3

Simulating Cities as Fractal Picturescapes

Suppose I want to understand the 'structure' of something. Just what exactly does this mean? It means, of course, that I want to make a simple picture of it, which lets me grasp it as a whole. And it means, too, that as far as possible, I want to paint this picture out of as few elements as possible. The fewer the elements there are, the richer the relationships between them, and the more of the picture lies in the 'structure' of these relationships. (Alexander, 1979, p. 34.)

3.1 The Quest for Visual Realism

The new geometry finds its most obvious expression in the natural world with examples of fractals all around us. Yet as Mandelbrot (1982) himself has argued, fractals are equally applicable to systems other than those portrayed in nature. Any system in which the whole is composed of parts arranged hierarchically in some self-similar order is fractal, and in fact, the most serious candidates may ultimately turn out to be artificial systems ranging from silicon chips to business organizations. Man-made structures such as cities, we will argue, display all the characteristics we have associated so far with fractals. In this spirit then, this chapter pursues two goals. First, we will begin to establish the applicability of fractal methods for describing and modeling cities in terms of the way their form reflects their function, although this will also be our longer term goal throughout subsequent chapters. A second and more immediate goal is to illustrate how fractal geometry combined with state-of-the-art computer graphics, can be used to produce highly realistic but minimalist pictures as Alexander (1979) implies above, pictures which have more than just superficial meaning when applied to city systems.

In a sense, we anticipated this at the end of the previous chapter. But to make real progress, we need to relax our approach to fractals which so far has been mainly based on strictly self-similar forms, completely and precisely determined by their initiators and generating rules. The fractals of Chapter 2 are hardly natural in any case, although as artifacts, they are products of our mathematical imagination, notwithstanding their occasional resemblance to real physical systems. To make progress here therefore, we need to consider fractals whose self-similarity across a range of scales can be described by statistics of randomly distributed variables

through various forms of deviation or variance around the mean. These we will refer to as 'statistical or random' fractals whose form can be self-similar or self-affine but only in terms of averages measured across several scales.

Statistical fractals are obviously necessary if we are to generate realistic natural scenes where randomness of form exists within well-articulated structure. The convergence of fractals and computer graphics is important too and we will start our discussion of realism with a little of its history. It is widely recognized that fractal geometry would not have established itself so firmly in so many sciences without the use of computer graphics in generating pictures. Mandelbrot (1983) describes how the mathematics of deterministic fractals remained largely inaccessible to generations of mathematicians because there was no way of illustrating its import in less esoteric and abstract terms. In fact it was Mandelbrot (1975) who first used computer graphics to illustrate ideas about the modeling of natural terrain using Brownian motion. His ideas in this realm were first formed when he noted the coincidence of the frequency distribution of random coin tossing illustrated in Feller's (1950) famous book on probability with typical cross-sections of terrain.

These early graphics were picked up quickly by a number of researchers. Carpenter (1980) used the ideas to generate computer graphic backcloths for flight simulators while Goodchild (1980) showed how these models might represent real terrain. Smith (1982) showed how they were used in the movie *Star Trek II* to generate a living planet, and Fournier, Fussell and Carpenter (1982) generalized this usage further, producing various types of fictional terrain. However, the pictures which accompanied Mandelbrot's (1983) second English edition of his book *The Fractal Geometry of Nature*, particularly those by his colleague Voss (1985), have gained the greatest recognition and have done most to popularize the subject. Stunning pictures of fractal mountainscapes at different fractal dimensions and their aggregation to the terrain and seas of planet-like worlds have been produced. Most recently, these landscapes such as those generated by Musgrave and his colleagues (Musgrave, Kolb and Mace, 1989; Mandelbrot, 1990) have become so realistic that they are hard to tell apart from natural scenes. This suggests that geomorphic and geologic processes of weathering and erosion are bound to generate fractal forms, thus giving further weight to the long-standing notion that 'form follows function'. Moreover, Mandelbrot's (1982) view that "... the basic proof of a stochastic model of nature is in the seeing: numerical comparisons must come second" has gained much credence through such demonstrations.

There have been other powerful demonstrations of fractal geometry using computer graphics and these have revolved around the idea of illustrating the fractal structure of mathematical space. Although this book is not concerned with these types of fractal, much of the glamour of the subject and not a little of its appeal has come from these geometries. In essence, the geometry of mathematical space is usually geometry which shows the properties of the mathematics involved, particularly the solutions to equations. For example, consider the iterative equation $z_{t+1} = z_t^2 + c$ which is the discrete equation for the logistic growth of a variable z such as population. Then if we consider that the solutions to such equations are complex numbers in that they have real and imaginary parts, and if we plot these on

x-y coordinates when z converges to a finite value, then the geometry of the solution is fractal in the following sense. If we start with $c = 0$, then the resulting solutions with different starting values for z_0 based on complex numbers, form what are called 'Julia sets', while the map of real and imaginary values which we get when we start with c as a complex number and $z_0 = 0$, define the 'Mandelbrot set'. When we plot both types of set and color the map systematically, the boundaries between solutions to the equations and the areas where the z_t values diverge towards infinity are fractal; as we zoom in on these boundaries, detail is magnified and shows the same form, however deep we zoom. These are remarkable results from such simple equations whose form can only be revealed directly through the power of computer graphics.

These sets have been beautifully rendered by Peitgen's group from Bremen (Peitgen and Richter, 1986) but on a more fundamental level, formal relationships between mathematical and physical fractals are being pursued through the idea of fractal attractors. We came across this idea in Chapter 2 when we summarized Barnsley's (1988) work. In essence, what can now be shown is that fractals in mathematical space such as Julia sets can be transformed into fractals in physical space. For example, it is easy, using changes in the transformation rules, to show how the Koch island can emerge from the Julia set and vice versa, the Koch island and the Julia set both being attractors in two dimensions. Finally, perhaps with an even greater sense of mystery, the solutions to many chaotic systems have been shown to have an underlying order which is fractal; and the visualization of chaotic solutions has again only been made possible through recent advances in computer graphics (Devaney, 1990).

Computer graphics is fast becoming a new medium for simulation throughout the sciences as well as in the arts and design. Clearly through the desire to simulate the 'fictional realism' of scenes which look realistic but are figments of the designer's imagination, there come useful ways of rendering backcloths in movies and the graphic arts. But the use of graphics to see what has not been seen before, to explore the whole question of scale and limits, and to render scientific predictions in ways in which the data have not been visualized hitherto, are central to the way fractals have been pioneered and are applicable. This is especially true in fields where data are extensive and have hitherto not been easy to visualize, and it is nowhere more appropriate than in the spatial sciences such as those dealing with both natural and artificial, physical and social systems, especially with urban phenomena in the form of cities, our focus here.

Mathematical models of city systems implemented on computers were first developed 30 years or more ago, but the theories of spatial organization and location used therein originated in economic theory from the early 19th century onwards. The typical urban models proposed so far have thus concentrated upon the location of and interaction between economic activities such as employment, population and transportation at the macro-spatial level where cities are divided into large zones such as census tracts (Batty, 1976) or at the micro-level of the individual or firm (Anas, 1982). Models which take the level of analysis down to the physical form of the city or to the relationship between urban activities, land uses and their physical form have rarely been developed. This is possibly because it is the activity

level which is the most appropriate for simulation, for it is here that economic theory can be brought to bear on model design, and thus there is the implicit view that the translation of spatial activity into physical land use is a fairly trivial task or at least, does not matter. It is more likely that the dearth of modeling the physical configuration of land use *per se* is largely due to unconscious neglect by those who have found it easier to begin with activity simulation and whose disciplinary biases have constrained their interest in the physical form of cities.

However, a major problem has begun to emerge in conventional urban modeling which relates to the meaning of spatial data and predictions. For a long time it has been known that when model outputs in terms of activities are mapped spatially in aggregate zones or as individual point patterns, their form often looks 'wrong' in some indefinable physical sense. Exceptionally good fits in terms of numerical indicators can be obtained, and such models may manifest robust and causally acceptable structures, but when their predictions are mapped, the whole does not seem to add up to the sum of the parts; systematic biases appear and the patterns often look physically imbalanced. In macro-modeling, such biases can often be corrected, or at least there are strategies which enable under- and over-prediction to be handled consistently, but with models based on individual discrete predictions, these problems are rarely addressed because the outputs are hardly ever mapped spatially. There are thus few checks on whether or not such models generate spatially acceptable predictions. In short, whatever type of model is used, their data and predictions have been difficult to assess spatially for computer graphics in this field is in its infancy.

This problem of visualizing spatial data and urban model predictions has only just begun to be tackled in terms of the development of appropriate computer graphics. It is already clear that a school of thought is fast emerging that the ultimate test of any model is that 'it must look right'. In one sense, this school represents a 'back-to-basics' movement which is not only borne of a dissatisfaction with the structure and focus of contemporary models. It is also based on the fact that as powerful computers are now available which make graphics easy to employ, visual reality would seem to be more important than statistical reality. In this, any models which attempt some physical simulation are likely to produce more reasonable-looking spatial patterns than those which are highly abstracted as points and networks. In fact, in this chapter we will show some examples of urban simulations which look distinctly 'uncity-like', thus demonstrating both the power and limits of simulation and the potential of fractal geometry as an organizing mechanism for such simulation. In one sense, although this entire chapter will be focussed on simulating 'right-looking' cities, it will also show how limited our best known theories and models of urban structure are in simulating the physical structure of land uses and urban activities. Before we can do this, however, we need to continue to relax fractal geometry by presenting the statistical view.

3.2 Randomness and Self-Similarity

In Chapter 2, we introduced several deterministic fractals from the strictly self-similar to self-affine but in each case, their generators produced the same forms or attractors each time they were initiated. Thus there is no uncertainty in the geometry of the resulting structures, and the rules for adding, taking away, displacing and transforming the initiator always produce a form whose ultimate attractor is unique. Such fractals however only exist in the world of mathematics for in nature, there is always chance. Objects may be similar but they are rarely identical, or if they are, their identity is only to the resolution of the measuring device, and there is always uncertainty beyond this. The fact that chance plays so dominant a role in the natural world is underscored by theories of evolution whose basis in selective mutation is now well established (Dawkins, 1986), while in the physical world, the repercussions of quantum theory are still reverberating throughout physical theory. In a less abstract realm, natural scenes composed of terrain, vegetation and particular climatic regimes are subject to all the physical and natural forces which enable change to take place in the landscape. It is clear that for any circumstance, although the processes which act as functions of form might be known, their operation is, to all intents and purposes, beyond our ability to observe, and we must be content in estimating their meaning statistically.

The intensity of the processes involved as well as the degree to which they interact within one another are also complemented by various constraints on the operation of the processes in question. In an urban context, such constraints are physical and artificial, ranging from areas of land upon which urban development is virtually impossible within given technological limits to institutional processes which constrain physical development in diverse ways. In short, processes which form cities operate under a variety of constraints which distort and transform the structure in general, and which thus have to modeled statistically. To demonstrate this we will begin with our basic fractal model, the Koch curve which we portrayed at the end of Chapter 2 as an idealized city form or boundary. We will begin relaxing this strictly self-similar deterministic fractal by introducing some elements of chance into its generation, and it is appropriate that we begin with the curve and its form as an island shown in Figures 2.1 and 2.2 respectively. Peitgen, Jurgens and Saupe (1992) provide similar demonstrations.

To introduce the element of chance, consider the way in which the generator is applied to the initiator in the traditional Koch curve which forms each side of the island shown in Figure 3.1(a). The generator is based on the regular midpoint displacement of the line into a line 4/3 times the length with each of the four segments of the line being 1/4 the length of the original line. The Koch curve is obtained by using this generator with the *same orientation* each time it is applied. However, we can introduce an element of chance by letting this orientation be chosen randomly on either side of the line which, when applied to the island, enables the boundary to be enhanced by adding or subtracting to obtain the new detail. The new island is shown in Figure 3.1(b) where the orientation either side of the line

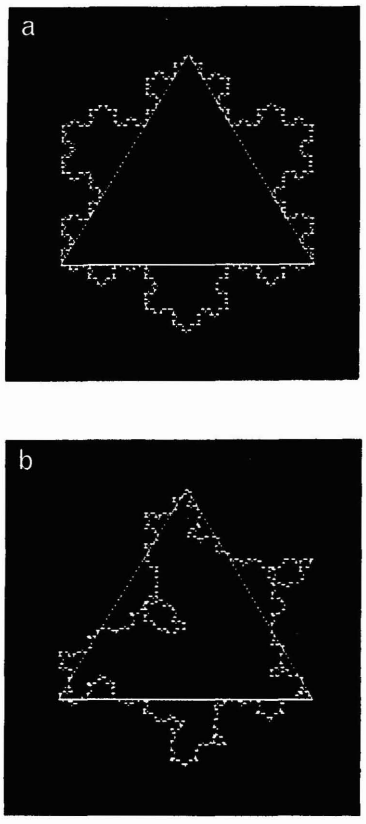


Figure 3.1. Regular and random Koch islands with identical fractal dimensions.

has been chosen randomly at each iteration of the generation. As Figure 3.1(b) shows, it is quite remarkable how the Koch island becomes irregular by introducing this simple chance effect. It is much closer to a natural coastline than the original island, although as the same number of lines with the same length are generated on each iteration, the fractal dimensions of each figure are the same, that is $D = \log(4)/\log(3) \cong 1.262$. This is perhaps the most remarkable aspect of randomization in generating fractals, and it not only shows that very different looking forms can have the same dimension and virtually identical functions (generators), but that fractal dimension says little about the orientation and overall shape of the ultimate figure. We will leave the reader to ponder this further, for it is an important issue throughout this book. However, before we leave the Koch island, we will show how even more irregular coastlines might be generated.

The generator of the Koch curve contains three parameters which might be manipulated or chosen randomly to form different curves, and we have already seen one way of doing this in Chapter 2 where we altered the midpoint and size of vertical displacement to form the Koch forest shown in Figure 2.3 and Plate 2.2. Thus we can alter not only the height of the displacement but also the position of the perturbation of the line given by the two points which define its location relative to the midpoint. These

values, called H, W_1 and W_2 respectively, are shown in Figure 3.2(a). We can now generate Koch islands with the values of H chosen randomly between 0 and say, the length of the initiating line, with the offset values W_1 and W_2 set between 0 and 0.5. In Figure 3.2(b), we show six Koch islands generated in this way where the three parameters are chosen randomly at the beginning of the generation process, but with the orientation chosen randomly at each iteration as in Figure 3.1(b).

The fractal dimensions of the resulting curves are not equal to 1.262, and to compute these, we have used equations (2.25) and (2.26) which are repeated here:

$$L(r) = N(r)r = Kr^{(1-D)}. \tag{2.25}$$

$N(r)$ and $L(r)$ are the number of parts and the length of the line respectively at scale r, and K is a constant of proportionality. The log transform of equation (2.25) yields

$$\log L(r) = \log K + (1-D) \log r, \tag{2.26}$$

where $(1-D)$ is the slope of the regression of $\log L(r)$ on $\log r$ from which the dimension D can be derived directly. For the six islands in Figure 3.2(b) we have calculated the perimeter $L(r)$ over six orders of magnitude and the estimated dimensions and the coefficients of determination associated with these estimates are also shown in this figure. We have not yet formally introduced regression to determine fractal dimensions, and we will postpone further discussion of this until Chapter 5 where we will build on the method first introduced by Richardson (1961).

In fact we will avoid such measurement until then, but from these results it is immediately clear that as the curve becomes more irregular and in this sense fills more of the two-dimensional space available, the fractal dimension increases. The consequent interpretation is that more rugged ria-like coastlines have higher fractal dimensions than smoother lines, and an obvious interpretation is that the value of the fractal dimension has strong implications for the underlying processes of weathering and erosion which lead to such forms. The great appeal of the Koch curve is that its fractal dimension of 1.262 is close to that estimated for the west coast of Britain by Richardson (1961) and Mandelbrot (1967). This is in contrast to the coastline of Australia with a dimension of 1.13 and of South Africa with 1.02. Ria coastlines have higher dimensions, but these are seldom more than 1.5, for above that value, the curve would have to considerably distorted in a rather systematic fashion for it to avoid self-intersection which, of course, is a physical necessity in terms of coastlines.

To illustrate the use of these ideas further in terms of generating realistic curves, we will take a memorable shape and perturb its straightline segments using degrees of perturbation which imply different fractal dimensions. Mainland Australia has been chosen, and it is easily described by the upper left hand shape in Figure 3.3 which consists of 10 straightline segments. Each straightline segment in itself has a dimension of 1, and we can see how close this shape is to the 'real' Australia by simulating different degrees of ruggedness. In Figure 3.3, we show what happens as the degree of ruggedness increases – as the fractal dimension increases in stages from $D \approx 1$ to $D \approx 2$. We must be clear about what we are doing here. If we

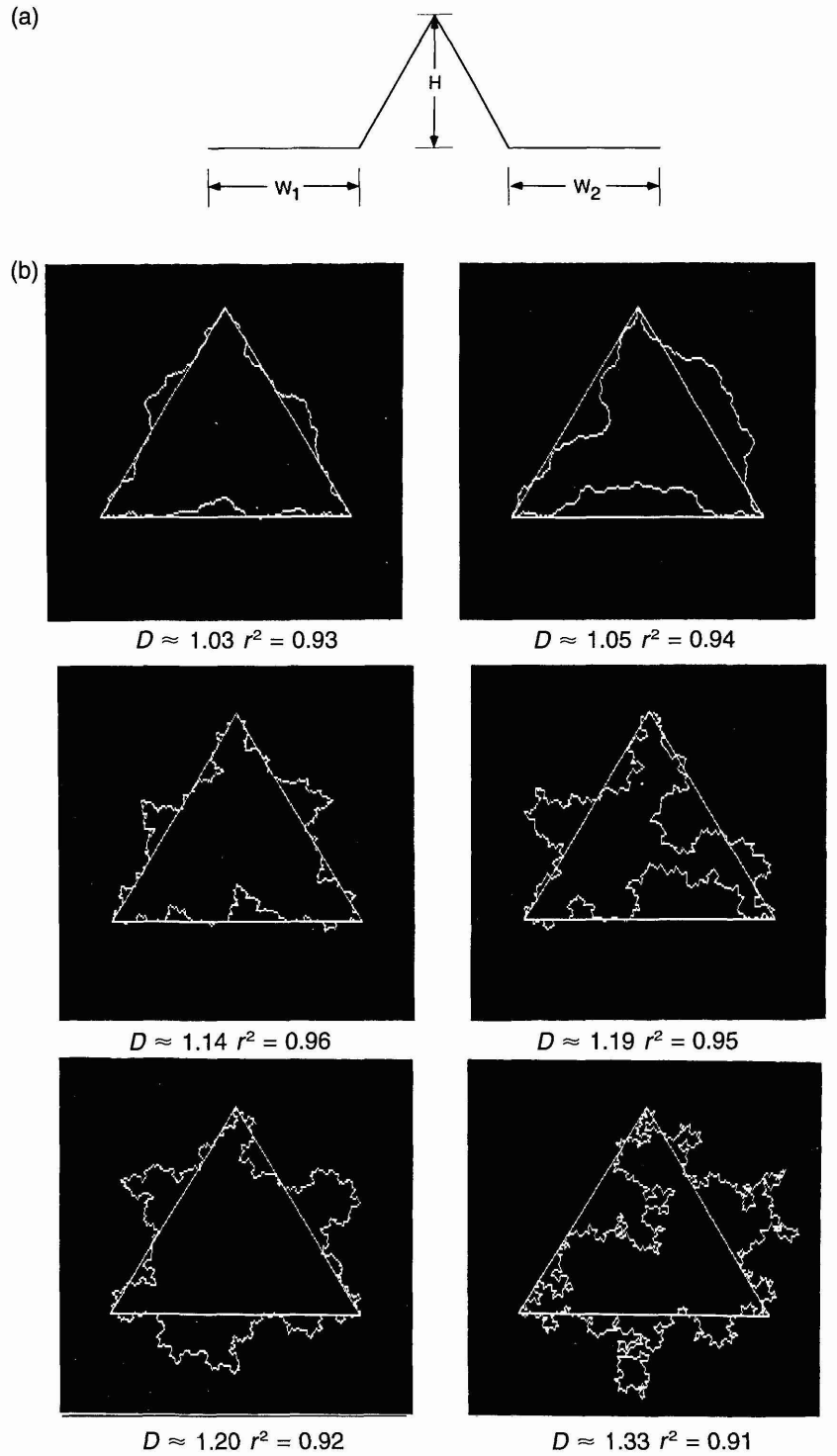


Figure 3.2. Random Koch islands with different fractal dimensions.

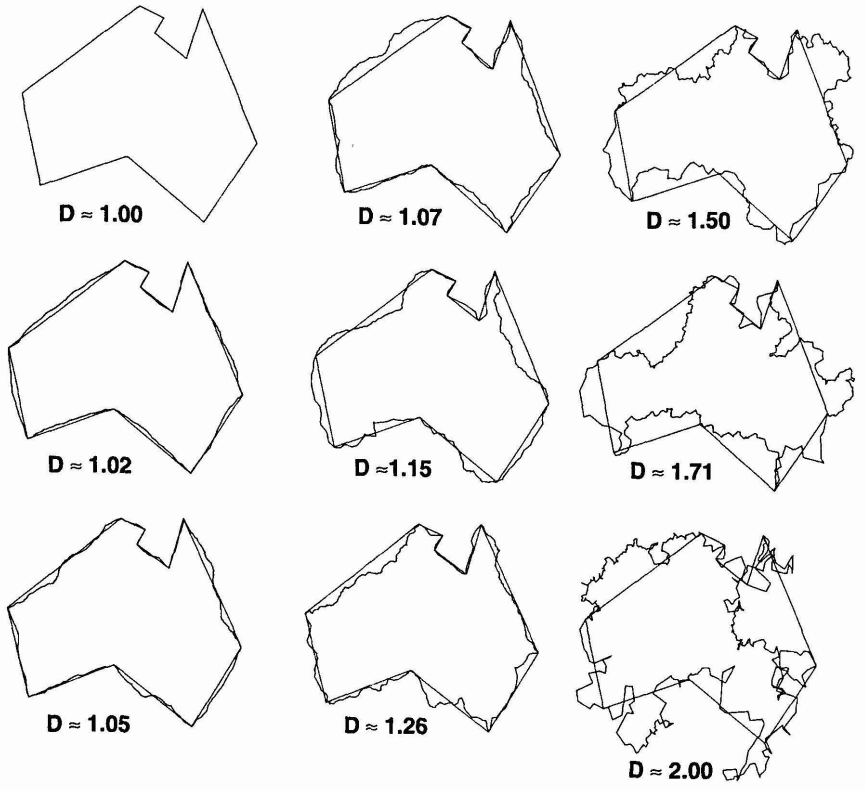


Figure 3.3. Simulating the coastline of Australia.

measured the fractal dimension of each 'Australia' in Figure 3.3, we would get values different from those which we have shown and have used to generate the shape. This is because our overall shape has already been fixed or constrained, and we are just simulating irregularity about each of its parts. Moreover, as the algorithms we use to generate such displacement imply the operation of chance, the values shown are those used to constrain this chance, but do not imply that the level of chance is fixed to those values we input. Nevertheless, the simulations do give us some feel for how the degree of irregularity increases as we increase dimension in the same way we did for the Koch curve in Figure 3.2(b).

Up to $D \approx 1.15$ which is near the accepted dimension of mainland Australia, the simulations clearly increase the realism of the coastline, but after this the coast becomes too rugged. By the time it reaches a ria-like level of 1.5, the only thing in common with Australia is the fact that the simulated coast passes through the eleven points which define the initial map. $D \approx 1.71$ is the dimension of many crystals (and cities as we will see from Chapter 7 onwards) while the map where $D \approx 2$ is much more reminiscent of a random walk across space. As such, Figure 3.3 provides a useful template for assessing approximate fractal dimensions (as Figure 3.2(b) does too). The real point of this example is not simply to show the range of irregularity. It is to emphasize the point that once the degree of irregularity is chosen in terms of a fractal dimension, it is possible to simulate such curves

using computer graphics. In fact, Figure 3.3 shows that we are able to simulate more realistic maps of Australia than the straightline map we started with, but the success of the simulation depends intrinsically upon this starting point, that is upon the initiator. For the pictures of planets, mountains and cities we will show in this chapter, all depend upon choosing initiators which are planet-, mountain- or city-like and upon the use of fractal rendering to make them realistic. There are a number of similar applications where fractals have been used to enhance cartographic detail in cases where the shape is too complex to describe in all its available detail, but where it can be approximated using fractal rendering (Dutton, 1981; Hill and Walker, 1982). The classic example is Australia as we have shown, and other examples where this map has been used to illustrate similar ideas are given in Fournier, Fussell and Carpenter (1982) and in Dell'Ocro and Ghiron (1983).

Before we conclude our introduction to statistical fractals generated from the occurrence of random events, we will examine the other classic fractal of Chapter 2, the Sierpinski gasket. Consider Figure 3.4(a) and note that the gasket can be seen as a process whereby an original equilateral triangle is tiled with three copies of itself which cover only three-quarters of the initiator. The number of units used to cover the shape is three and the scaling is 1/2, in that each side of the original triangle divides into two which form two of the sides of two new triangles. The fractal dimension is thus $D = \log(3)/\log(2) \approx 1.585$. First we will relax the scaling in that instead of choosing the midpoint of each side of the initiator which divides the side into those of two new triangles, we let this value be chosen randomly as any point on the side. This generates the random gasket shown in Figure 3.4(b) whose fractal dimension must be computed using one of the methods such as cell counting introduced in later chapters. The dimension is not important for Figure 3.4(b); this is only one step along the road to a completely random Sierpinski gasket which we will be using as the basis for generating terrain later in this chapter. Imagine now that our initi-

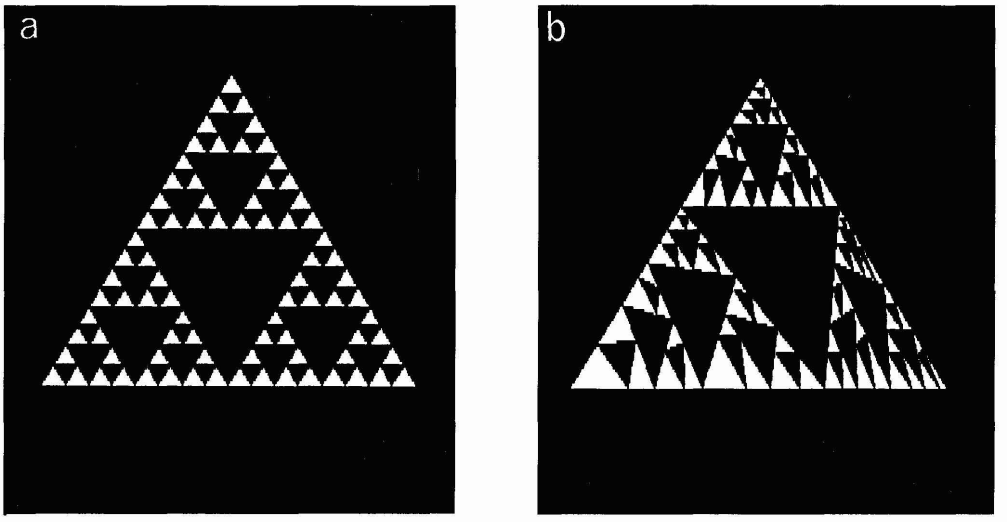


Figure 3.4. The Sierpinski gasket: (a) without, and (b) with random 'midpoint' displacement.

ating triangle is no longer equilateral: it may take on any shape. Then instead of constraining the subdivision of each of the triangle's sides to be somewhere on each of these sides, let this point be chosen randomly somewhere within a circle centered on the midpoints, as in Figure 3.5 which is taken from McGuire (1991). The three new triangles distort the original shape, and further subdivision in the same way continues the distortion. In Figure 3.5, the shape after 10 iterations is shown where the resemblance to terrain is clear. Of course, the Sierpinski gasket is not a particularly good model of a mountain, although the fact that it is a triangle is perhaps close enough for the point to be made. As we did for Australia, we can control the degree of displacement or the fractal dimension in this case by setting the radius of the circle in which the displacement takes place, although we have not pursued this in any formal sense in terms of this example. The way we have generated fractals in this and in the last chapter really depends upon the process of defining a generator which is consistently and persistently applied to an originating or initiating object. At this point, we must step back a little and say something about the possibility that there may be underlying mathematical models of fractals which will help us in

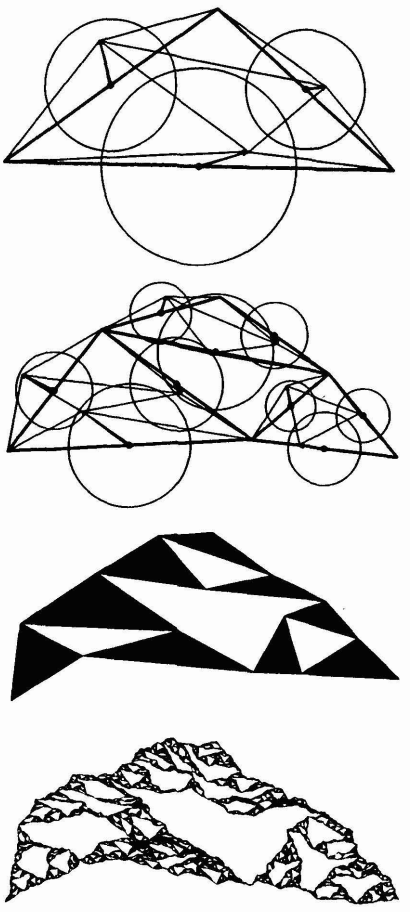


Figure 3.5. Using the random Sierpinski gasket to simulate terrain (from McGuire, 1991).

the quest to generate realistic objects. To this end, we will now introduce Brownian motion, one of the central ideas of this chapter.

3.3 Fractional Brownian Motion

The search for underlying generating functions which give rise to fractal geometries is in some sense a fruitless quest. The generating functions we have used so far can all be specified geometrically, and as Barnsley (1988) has so persuasively shown, a slight change in emphasis based on their treatment as classic transformations yields further insights into their form. In fact, we began our introduction to fractals in Chapter 2 by implying with Mandelbrot (1983) that a shift from continuous functions to discrete represented the obvious way to deal with irregular shapes based on curves which might be continuous everywhere but have no derivatives. Such of course is the Koch curve and its generalization to the coastline. However, the search for an underlying fractal model in one sense throws us back to the very mathematics which fractal geometry has released us from. Not quite perhaps. Although our purpose is not to develop a strict mathematical treatment of fractals in this book, we must indicate that there exist highly formalized models of fractal order which can be approximated by the techniques of continuous mathematics, in particular by infinite series such as Fourier transforms and related functions.

Our starting point in this is 'Brownian motion' or Bm as it is sometimes called. In 1828, the Scottish biologist Robert Brown first made known his observations of the motion of dust particles which appeared to move at whatever scale they were examined. In short, their motion appeared to be fractal. We have almost provided an example of two-dimensional Brownian motion in the last section where our simulation of the coastline of Australia with the dimension near to 2 shows a self-intersecting curve whose lengths and orientations are entirely random. If we were to relax the constraint that the walk should pass through the 11 points of the Australian coastline, then the walk would represent true Brownian motion. However, to get a better sense of this motion, it is worth developing the analysis taking the example of a time-varying phenomena, and then generalizing the analysis to the cases of coastlines and terrain in the two and three dimensions of physical space respectively. Our exposition will closely follow the way the subject is treated in the fractal literature and in this we will closely follow Saupe (1988, 1991) and Voss (1988).

Consider a variable $V(t)$ which is the value of some phenomena at time t and define the change in this variable ΔV as

$$\Delta V = V(t_1) - V(t_0),$$

where the time interval Δt is also defined as

$$\Delta t = t_1 - t_0.$$

It is the change in the variable ΔV which is of major interest in that we will assume that it is this variable which is randomly distributed; thus over

any time period Δt, the value of ΔV would be that which is taken from a normal or Gaussian distribution of the variable. However, because the variable is a fractal, it is not possible to determine any limiting value of dV/dt and thus the value of ΔV must be proportional in some way to the length of the time interval Δt. In fact, we assume that it is the variance of the variable called var(ΔV) which is directly proportional to time, that is

$$\text{var}(\Delta V) = \langle [V(t_1) - V(t_0)]^2 \rangle = (t_1 - t_0)\sigma^2. \tag{3.1}$$

Without loss of generality we will assume that the variance σ^2 can be normalized to 1, and thus in the following exposition, we will only include it explicitly where it is important to do so. Thus equation (3.1) can be written as

$$\text{var}(\Delta V) = \Delta t. \tag{3.2}$$

The implication of equations (3.1) and (3.2) is that the variable ΔV is thus proportional to the square root of Δt, that is

$$\Delta V \propto \Delta t^{1/2}. \tag{3.3}$$

This means that the scaling between ΔV and Δt is one where if time changes by four units, then the value of the variable will only increase by two. In short, the relationship over different time scales is self-affine, not strictly self-similar in the language of Chapter 2.

Before we generalize these equations to a wider class of Brownian motion, we will formally examine this scaling. If we assume that the time changes by a factor r, then the appropriate change variables can be written as

$$\Delta V' = V(rt_1) - V(rt_0),$$

and

$$\Delta t' = rt_1 - rt_0 = r\Delta t.$$

Now the variance of $\Delta V'$ can be written as

$$\begin{aligned} \text{var}(\Delta V') &= \Delta t'\, \sigma^2 = r\Delta t\sigma^2 \\ &= r\ \text{var}(\Delta V), \end{aligned} \tag{3.4}$$

from which it is clear that the value of the change variable $\Delta V'$ is scaled by the square root of r, that is

$$\Delta V' = r^{1/2}\Delta V \propto (r\Delta t)^{1/2}. \tag{3.5}$$

We can now generalize this formalism to fractional Brownian motion (fBm) where we introduce the exponent H from which, as we will show below, the fractal dimension D can be derived. Following equation (3.1), the variance of ΔV can now be stated as

$$\text{var}(\Delta V) = \Delta t^{2H}\sigma^2, \tag{3.6}$$

where H is the Hurst exponent (named after the researcher who first used this equation in measuring the discharge rate of rivers, see Mandelbrot

(1983)), and σ^2 the variance which we can normalize as 1. The variable ΔV can be written as

$$\Delta V \propto \Delta t^H \sigma, \tag{3.7}$$

and it is easy to show that the scaling of time by r following equation (3.5) is given as

$$\Delta V' = r^H \Delta V \propto r^H \Delta t^H \sigma. \tag{3.8}$$

Thus a change in scale of r units in time leads to a change of r^H in ΔV, and it is clear that the case of pure Brownian motion is given when $H = 1/2$. We will assume that H varies in the range from 0 to 1 in this particular example. The last point we should make is that when equations (3.1) to (3.8) apply to any and every time interval Δt, we say that the variable ΔV shows stationarity.

It is fairly straightforward to determine the fractal dimension of fBm. Let us assume that the variable ΔV is examined over N time periods and that for each equal time interval, $\Delta t = 1/N$. Thus for every change in scale of $1/N$, the variable ΔV changes in proportion to $(1/N)^H$. If we consider that for each time interval of length $1/N$, we place a 'box' of length $1/N$ times $(1/N)^H$ over the change in frequency of the variable, then we have to multiply this by all N boxes to get a total coverage of the change in the variable. Formally, we have the change in ΔV with Δt as

$$\frac{\Delta V}{\Delta t} = \frac{N}{N^{H'}} \tag{3.9}$$

and as there are N time periods, the number of square 'boxes' of size $(1/N)^2$ called $N(\Delta t)$ is given as

$$N(\Delta t) = N\frac{\Delta V}{\Delta t} = N^{2-H}$$

$$= \left(\frac{1}{\Delta t}\right)^{2-H} = \Delta t^{H-2}. \tag{3.10}$$

Now from equation (2.17) which counts the number of equal elements which approximate a fractal line, it is clear that the number of segments is

$$N(\Delta t) = \Delta t^{-D}. \tag{3.11}$$

A comparison of equations (3.10) and (3.11) shows that the exponent in both must be equal; that is

$$N(\Delta t) = \Delta t^{H-2} = \Delta t^{-D}, \tag{3.12}$$

from which it is clear that $D = 2 - H$. We can now write our equations of fractional Brownian motion given above in (3.6) and (3.7) as

$$\text{var}(\Delta V) = \Delta t^{2H} \sigma^2 = \Delta t^{4-2D} \sigma^2, \tag{3.13}$$

$$\Delta V \propto \Delta t^H \sigma = \Delta t^{2-D} \sigma. \tag{3.14}$$

To complete this section, we need to clarify the meaning of different values of the Hurst exponent H and the fractal dimension. Clearly for the

case of pure Brownian motion, $H = 1/2$ and thus $D = 1.5$ and in one sense, this represents the baseline. In the case where $H = 1$, then D also equals 1 and this represents a completely smooth function. The change in ΔV is simply a function of time as equation (3.14) indicates. When $H = 0$, then $D = 2$, and this means that at whatever scale the variation in the function is examined, the motion or change is the same. This is characteristic of very spiky-looking functions which 'fill the space available'. We will generalize these ideas to landscapes in the next two sections, but it is worth anticipating what the values of H and D are with respect to terrain. Smoothly varying terrain has both fractal and Hurst dimensions equal to 1, while at the other extreme, very rugged and cavernous terrain has a D near to 2 and a Hurst exponent near to 0. In the next section we will show how these ideas can be developed for simulating landscapes, but if readers wish a more complete exposition, then the chapters by Voss (1988) and by Saupe (1988, 1991) are worth reading as are the relevant sections in Mandelbrot (1983).

3.4 Fractal Planetscapes and Terrain

The equations describing the variance properties of fBm given in (3.1) to (3.14) above only illustrate the properties of these processes and give little insight into the way they might be computed. In fact for pure Brownian motion in the plane this is straightforward and it can be implemented as follows. Defining the variable $V(t)$ now as the total distance traveled so far by a point tracing out a random walk in the plane, we define appropriate units of time; in each equal time interval, we select a pair of x-y coordinates in the plane by drawing random numbers from a Gaussian distribution, appropriately normalized to represent the physical distance-scale properties of the problem. A change in the distance $\Delta\tau(x_{n+1}, y_{n+1})$ can then be computed from $(u^2 + v^2)^{1/2}$ where $u = x_{n+1} - x_n$ and $v = y_{n+1} - y_n$. The total distance traveled at time t_{n+1} is

$$\tau(x_{n+1}, y_{n+1}) = \tau(x_n, y_n) + \Delta\tau(x_{n+1}, y_{n+1}). \tag{3.15}$$

where $n + 1$ acts both as an index of time and space. A plot of such motion is shown in Figure 3.6, while a graph of the change in distance at each time step $\Delta\tau(x_{n+1}, y_{n+1})$ and the total distance traveled $\tau(x_{n+1}, y_{n+1})$ from equation (3.15) is drawn in Figure 3.7. It is clear from these figures that the motion is Brownian, and in particular that changes in the total distance traveled over an arbitrarily chosen time period are proportional in some way to the length of that time period. Detailed measurement of the variation in this function indicates that it is consistent with equations (3.1) and (3.2).

These ideas can easily be generalized to a system with any number of dimensions for fBm in terms of equations (3.13) and (3.14), but their application is somewhat more difficult. There are two broad classes of algorithm which can be used in their implementation, and before we present the original and perhaps most consistent method, we will outline these. The first class of methods is based on *recursive* algorithms, in that its application involves ever more detailed approximation to the limiting fractal function

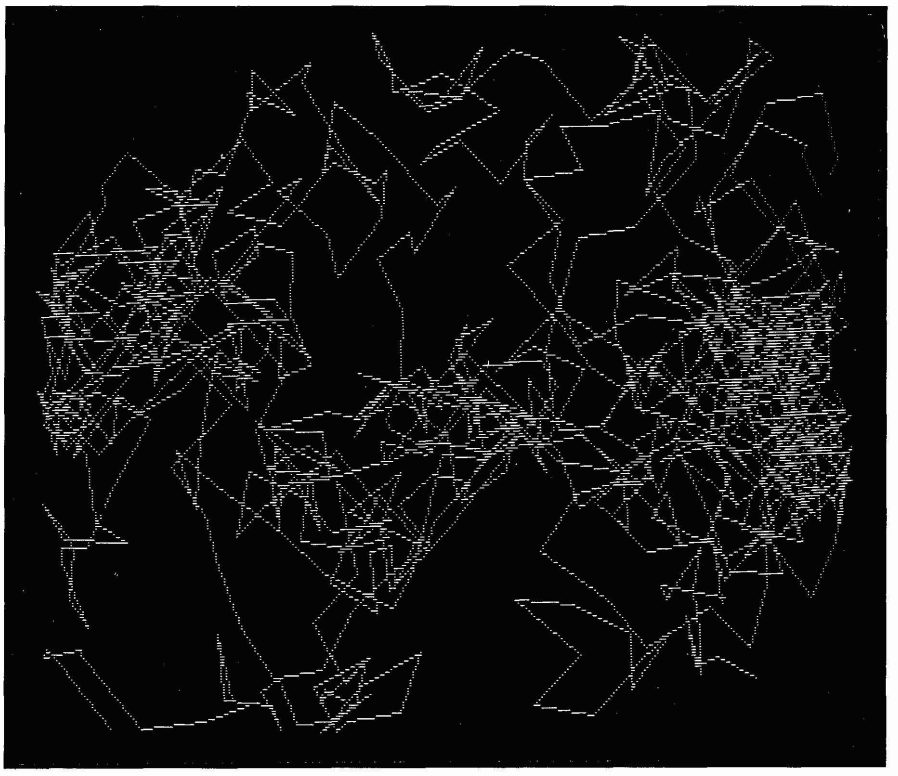


Figure 3.6. Pure Brownian motion in the plane.

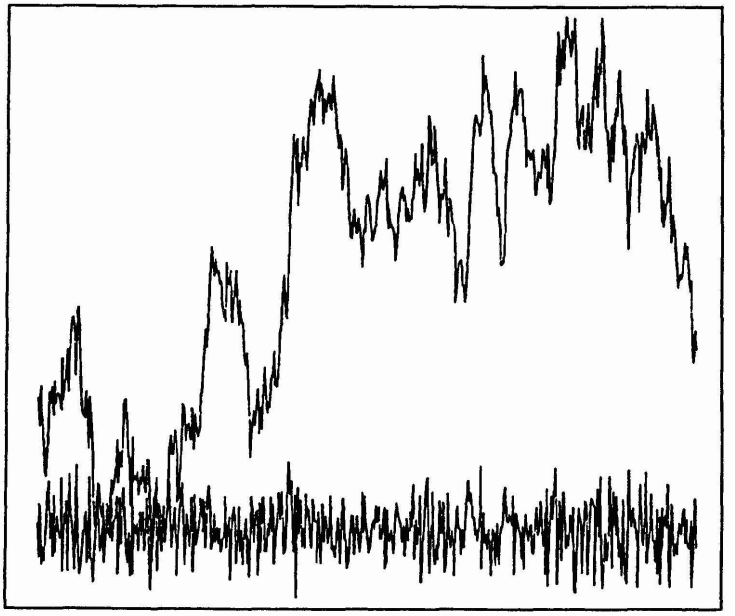


Figure 3.7. Profiles of pure Brownian motion.

and the process of approximation involves algorithms which are iteratively applied at each scale of resolution. In contrast, the second class involves *fixed resolution* algorithms which approximate the function at a prespecified level of detail and thus have to be computed afresh if different levels of detail apply. The computational properties of these algorithms are such that the recursive methods are usually more efficient and enable computation to stop on the basis of what has been computed so far, whereas the second class requires complete computation before the appropriateness of its application can be evaluated.

The best known recursive method is midpoint displacement of the variety we have been using so far in this book and which is best illustrated by the generation of the Koch curve. There are several variants on this process. These involve adding noise and variation after the computation has taken place at each level of resolution in order to resolve the key problem with such methods that the functions generated are not stationary. The second recursive method is more involved although the functions it generates are stationary. This is the random cuts method which is based on the idea of increasing the scale of resolution by taking random cuts across the function, computing its displacement randomly to meet the variance constraints, and continuing this process until a fixed number of cuts are generated. Because the cuts are randomly positioned, it is not possible to ensure that a level of detail is reached by a particular iteration, although the resolution does increase as the method proceeds.

Fixed resolution methods depend upon approximating fBm at a prespecified level of resolution, and these methods are in general based upon approximating the function using various forms of series. The most well-known are based on Fourier transforms, although the general problem with these methods is that they tend to be periodic, in that the functions repeat themselves on a cycle of 2π. Such problems have been dealt with by keeping the transformations well-within the period range, although in general, a major problem remains in that such functions generate intensive demands for computation time. There are also a variety of new methods based on modified midpoint displacement outlined by Mandelbrot (1988), some of which have been implemented by Musgrave, Kolb and Mace (1989). In the sequel, we will not use the fixed resolution methods because the recursive methods are deemed more appropriate for the exploratory ideas developed here. However, there remains the challenge not only to develop new and better methods, but also to provide more definitive comparisons. Useful surveys of the methods and their algorithms are presented by Voss (1988) and Saupe (1988, 1991).

We will begin by outlining how the random cuts method has been applied while in the next section we will deal with midpoint displacement. As indicated earlier, the random cuts method produces functions which exhibit stationarity in their variances. Generalizing fBm to three-dimensional space, for any two-dimensional measure of distance τ_{uv} computed as $(u^2 + v^2)^{1/2}$, the variance must satisfy

$$\langle [z(x+u, y+v) - z(x, y)]^2 \rangle = (\tau_{uv})^{2H}\sigma^2, \tag{3.16}$$

where $z(x, y)$ is the elevation of the terrain at coordinate x, y. In this case we can also assume that σ^2 is normalized to unity. The major change when

one moves from two to three dimensions, from functions in the plane to those in the volume, is that the fractal dimension is now given as $D = 3 - H$. Note also that the intersection of a plane with the fBm surface yields a profile with a fractal dimension of $D = 2 - H$, and these results can be easily generalized. In the context of terrain, this implies that if the fractal dimension of a coastline is determined, then the dimension of its relevant surface is $D + 1$, while if the dimension of the surface is calculated, then the dimension of the plane which cuts the surface as a coastline is $D - 1$. This can provide a cross check in the computation of such dimensions.

The best way to illustrate the idea of the random cuts method is to consider displacement on a sphere or a circle. On the circle, a randomly chosen line which intersects the circle in two places is chosen, and a displacement consistent with the fractal dimension adopted is made. Another cutting line is then chosen which is independent of the first line, a displacement of appropriate proportions is made and so on. This process continues until a level of accuracy required is reached, but unlike midpoint displacement, this is not known in advance. The process stops when all points defining the circle reach the appropriate level of resolution, but it is likely that more than half of these points will be at a level of detail greater than that specified in the stopping rule. This method was originally used by Voss (1985) for pure Brownian motion, for $H = 1/2$, although in later applications, the method has been generalized to fBm.

The method is beautifully illustrated by Voss's (1985) construction of Mandelbrot's famous planetscape *Planetrise over Labelgraph Hill* which is reproduced on the back cover of Mandelbrot's (1983) book. In another context, we illustrate a much simplified reduction of this in Plate 3.1 (see color section). The method clearly demonstrates how the original sphere is cut and then projected onto the flat plane. This picture was based on the random cuts method simulating pure Brownian motion, but since then, various renditions of similar planetscapes have been made using a modified form of the method consistent with $H \neq 1/2$. Voss (1985) and Mandelbrot (1983) both imply that by zooming in on the planet, it is possible to generate mountain and valley landscapes for the fact that the displacement is based on a sphere means that three-dimensional terrain is actually being simulated. Voss has also produced the terrain for this application, and these too are illustrated in Mandelbrot's (1983) book.

Before looking at these pictures, it is worth noting that little work has been done on calculating the actual fractal dimensions of terrain. This has not yet caught the interest of those concerned with computer graphic simulations, although there has been a good deal of discussion concerning ways to increase the realism of such scenes by varying such dimensions. An exception to this is in the work of Goodchild (1980) who has generated several hypothetical fBm terrains using the cutting plane method and who has explored their geomorphologic properties. Goodchild's hypothetical terrains are shown in Figure 3.8 where it is immediately clear that as the fractal dimension of these scenes increases towards 3, the landscapes become a jumble of spikes like stalagmites and stalactites and bear little resemblance to real surface landscapes (Goodchild, 1982; Goodchild and Mark, 1987). In fact, it is at the lower dimensions that these landscapes look more realistic. Mark and Aronson (1984) have fitted fBm surfaces to 17 sets

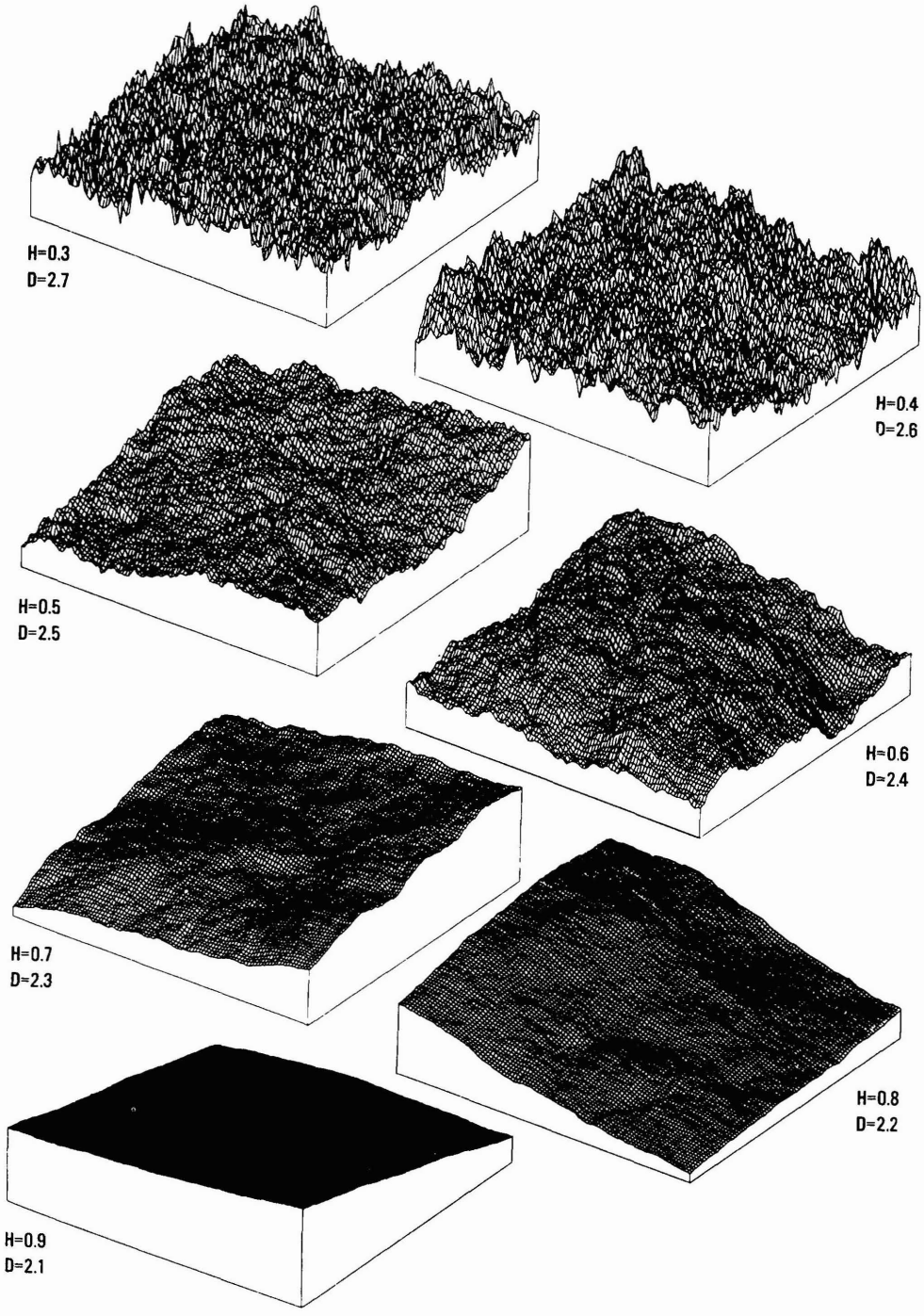


Figure 3.8. Simulated terrain with different fractal dimensions (from Goodchild and Mark, 1987).

of digital elevation data and found that although the fractal function provided rather good fits for spacing intervals less than 0.5 km with a dimension of around 2.25, above this spacing there was a clear break in the slope of the related variogram, suggesting a dimension of 2.75 for 0.5 to 5 km spacing. Over 5 km, there was no correlation with the fractal function.

These points have been recognized by those developing simulations of terrain. Voss (1985, 1988), for example, indicates that several of his landscapes have been made more realistic by scaling elevations to make them smoother through post-processing of the outputs from the cutting plane method. Other *ad hoc* techniques have been used. For example, some have varied fractal dimension directly with respect to elevation, with higher dimensions at higher elevations. As we have indicated earlier, Musgrave, Kolb and Mace (1989) have developed such simulations by including hydraulic erosion and thermal weathering processes directly into such simulations with striking effect. There is clearly much that can be done to extend these models, but before we show how they can be applied to city systems, we will introduce the technique of midpoint displacement which has been used more widely than the method just described.

3.5 Simulating Brownian Motion by Midpoint Displacement

There are several reasons why the technique of midpoint displacement, although less consistent than the random cuts method, might be preferred. First it allows direct control over the level of detail simulated. That is, in advance, one has some idea of how the landscape might look and this is important if the goal is simply realistic-looking terrain rather than terrain which accurately reproduces some reality. It is of course easier to implement and perhaps easier to analyze, and it relates to the ideas we have already introduced in our study of fractals. Its basic problem is that it does not completely produce the required stationary variances; that is, the variances produced are stationary, but only with respect to those displacements that reflect the hierarchical structure of the way the function is computed.

To illustrate the process, we will revert to our two-dimensional function which relates the variable $V(t)$ to time t. To fix ideas, we might think of this as a line whose coordinates are $V(t)$ and t and to illustrate the method, we show this line in Figure 3.9. We will first outline the method for the case of pure Brownian motion using equation (3.1) based on the interval $\Delta t = t_1 - t_0 = 1$ where $t_1 = 1$ and $t_0 = 0$. First we restate equation (3.1)

$$\text{var}(\Delta V) = \langle [V(t_1) - V(t_0)]^2 \rangle = (t_1 - t_0)\sigma^2, \tag{3.1}$$

where using the unit interval, this becomes

$$\text{var}(\Delta V) = (1 - 0)\sigma^2 = \sigma^2. \tag{3.17}$$

We will now begin the midpoint displacement. In the first step, we choose

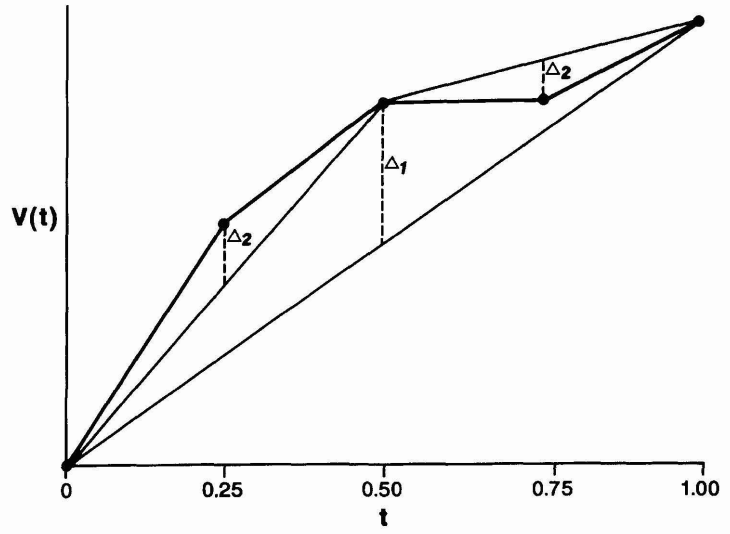


Figure 3.9. Brownian motion as midpoint displacement.

the variance of $V(1/2)$ as the midpoint $1/2$ by adding a variance displacement Δ_1^2 to half the variance of the entire original interval. The variances $\langle[V(1/2) - V(0)]^2\rangle$ and $\langle[V(1/2) - V(1)]^2\rangle$ are equal for one of these, then

$$V(1/2) - V(0) = \frac{1}{2}[V(1) - V(0)] + \Delta_1, \tag{3.18}$$

$$\langle(V(1/2) - V(0))^2\rangle = \frac{1}{4}\langle[V(1) - V(0)]^2\rangle + \Delta_1^2 = \frac{1}{2}\sigma^2$$

$$= \frac{1}{4}\sigma^2 + \Delta_1^2 = \frac{1}{2}\sigma^2. \tag{3.19}$$

It is easy to see that the variance in the displacement and the displacement itself from equation (3.19) are calculated as

$$\Delta_1^2 = \frac{1}{4}\sigma^2 \text{ and } \Delta_1 = \frac{1}{2}\sigma.$$

The second step proceeds in like manner. The variances $\langle[V(1/4) - V(0)]^2\rangle$ and $\langle[V(3/4) - V(1)]^2\rangle$ are equal and taking one of these, the new displacement values Δ_2^2 and Δ_2 are calculated from

$$V(1/4) - V(0) = \frac{1}{2}[V(1/2) - V(0)] + \Delta_2, \tag{3.20}$$

$$\langle[V(1/4) - V(0)]^2\rangle = \frac{1}{4}\langle[V(1/2) - V(0)]^2\rangle + \Delta_2^2 = \frac{1}{4}\sigma^2$$

$$= \frac{1}{8}\sigma^2 + \Delta_2^2 = \frac{1}{4}\sigma^2. \tag{3.21}$$

The displacements are thus

$$\Delta_2^2 = \frac{1}{8}\sigma^2 \text{ and } \Delta_2 = \left(\frac{1}{8}\right)^{1/2}\sigma.$$

Continuing this process and noting the equality of the variance displacements for subdivisions of the intervals at the same level, it is easy to show that on iteration k, the mean squared displacement or variance of the displacement is given as

$$\Delta_k^2 = \frac{1}{2^{k+1}}\sigma^2,$$

from which the actual displacement is the square root. The logic of this subdivision is shown in Figure 3.9 where it is clear that equations (3.18) to (3.21) apply to all subdivisions at the appropriate level and not just those intervals that are given above.

This method although applied in its pure form several times (Carpenter, 1980; Fournier, Fussell and Carpenter, 1982), can easily be generalized to fBm. We follow exactly the same steps, but note now that there is an exponent of $2H$ on the time interval associated with the variance. Restating equation (3.6)

$$\text{var}(\Delta V) = \langle [V(t_1) - V(t_0)]^2 \rangle = (t_1 - t_0)^{2H}\sigma^2, \tag{3.6}$$

and using the unit interval as in equation (3.17)

$$\text{var}(\Delta V) = (1 - 0)^{2H}\sigma^2 = \sigma^2, \tag{3.22}$$

we follow an identical sequence to the pure case above. In the first step,

$$V(1/2) - V(0) = \frac{1}{2}[V(1) - V(0)] + \Delta_1, \tag{3.23}$$

$$\langle [V(1/2) - V(0)]^2 \rangle = \frac{1}{4}\langle [(V1) - V(0)]^2 \rangle + \Delta_1^2 = \left(\frac{1}{2}\right)^{2H}\sigma^2$$

$$= \frac{1}{4}\sigma^2 + \Delta_1^2 = \left(\frac{1}{2}\right)^{2H}\sigma^2, \tag{3.24}$$

with the displacement calculated as

$$\Delta_1^2 = \left[\frac{1}{4} - \left(\frac{1}{2}\right)^{2H}\right]\sigma^2.$$

The second step proceeds in like manner. The variances $\langle [V(1/4) - V(0)]^2 \rangle$ and $\langle [V(3/4) - V(1)]^2 \rangle$ are equal and the new displacement values Δ_2^2 and Δ_2 are calculated from

$$V(1/4) - V(0) = \frac{1}{2}[V(1/2) - V(0)] + \Delta_2, \tag{3.25}$$

$$\langle [V(1/4) - V(0)]^2 \rangle = \frac{1}{4}\langle [V(1/2) - V(0)]^2 \rangle + \Delta_2^2 = \left(\frac{1}{4}\right)^{2H}\sigma^2$$

$$= \frac{1}{4}\left(\frac{1}{2}\right)^{2H}\sigma^2 + \Delta_2^2 = \left(\frac{1}{4}\right)^{2H}\sigma^2. \tag{3.26}$$

A little rearrangement of equation (3.26) shows that the variance of the displacements is

$$\Delta_2^2 = \frac{\sigma^2}{(2^2)^H}(1 - 2^{2H-2}),$$

and in general

$$\Delta_k^2 = \frac{\sigma^2}{(2^k)^H}(1 - 2^{2H-2}).$$

In Figure 3.10, we show an example of the application of midpoint displacement for the case of a fractal line whose details at successive levels of resolution have been generated using pure Brownian motion (with $H = 1/2$). This illustrates how the profile for each level of resolution provides the initiator for the generation of detail at the next level down.

3.6 Fractal Terrain Using the Midpoint Displacement: the 'Earthrise' Sequence

We have already seen how we might generate fairly realistic terrain by tiling the plane with triangles whose coordinates are chosen randomly but within the logic of hierarchical midpoint displacement. The sequence of

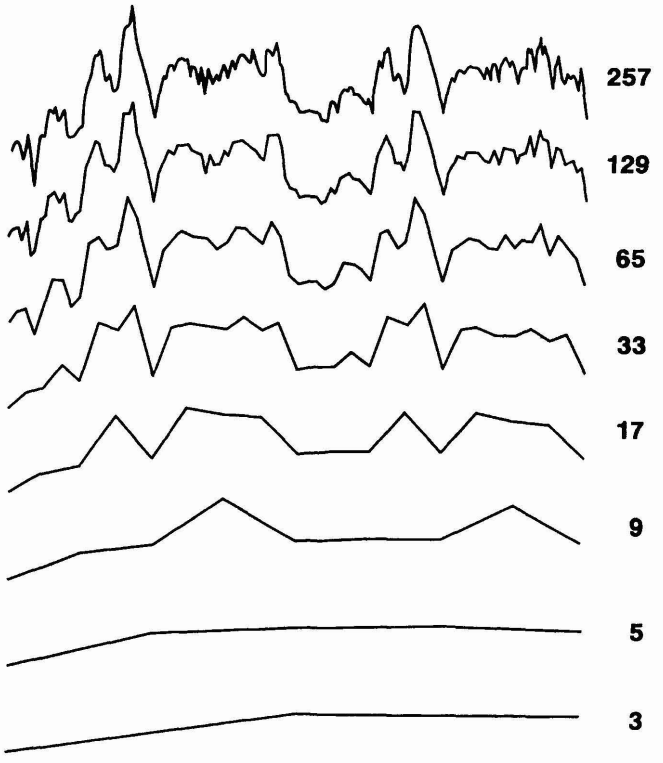


Figure 3.10. Brownian motion computed by midpoint displacement across several scales.

distorting the Sierpinski gasket used by McGuire (1991) presented in Figure 3.5 illustrates a more general approach which we will use in generating landscape and cityscape scenes in the rest of this chapter. A particularly useful demonstration of this method is given as van Dam (1984) which involves replacing each triangle with four, not three, copies of itself is shown in Plate 3.3. There is however a problem in using midpoint displacement in that the nonstationarity of the process sometimes leads to creasing in the landscape (or in the form of whatever object is being rendered). This is due to the fact that at the higher and earlier levels of recursion, the points and lines generated are not subject to any further randomization, thus implying greater degrees of nonstationarity when compared with points which are generated later in the recursion.

We can show this formally in terms of equations (3.17) to (3.26) which we used in the last section to generate fBm. First note that the variances for the intervals [1/4, 0] and [3/4, 1] must be the same, that is

$$\langle [V(1/4) - V(0)]^2 \rangle = \langle [V(3/4) - V(1)]^2 \rangle = \left(\frac{1}{4}\right)^{2H} \sigma^2. \tag{3.27}$$

Now if we add these two variances we would expect them to equal the variance of the interval [1/2, 0] or [1, 0]. Equating these two variances, we get

$$\langle [V(1/2) - V(0)]^2 \rangle = \langle [V(1/4) - V(0)]^2 \rangle + \langle [V(3/4) - V(1)]^2 \rangle$$

which from equations (3.24) and (3.27) implies that

$$\left(\frac{1}{2}\right)^{2H} \sigma^2 = 2 \left(\frac{1}{4}\right)^{2H} \sigma^2, \tag{3.28}$$

which is only the case when $H = 1/2$, the case of pure Brownian motion. This is the main reason why those using the midpoint displacement algorithm usually introduce some form of additional random generator either during the process of iteration or after the output at the required level of resolution has been computed. However, the use of other tessellations in the plane can help resolve this, such as the choice of a square grid as initiator. Mandelbrot (1988) has used nested hexagons to develop the method more recently, although later in this chapter, we will demonstrate the importance of choosing the correct initiator by adopting a square grid for the generation of cityscapes.

We will use the triangular net to first show how it is possible to construct a planetscape and then a mountainous terrain by midpoint displacement for the case of pure Brownian motion. Our method is extremely fast and involves very short computer programs which assume that some overall shape of the object in question is input to the program in the first place. In Figure 3.11, we show how a mountainous landscape can be generated by inputting the basic structure of the landscape in terms of its overall form. In this case, the inputs are large overlapping triangles which are then used in the rendering of fractal detail. Each triangle is rendered separately using the type of triangle displacement shown in Figure 3.11. The colors of the scene are chosen so that the nearer the top of each mountain, the more likely the mountain is to be snow-covered. The technique we use examines

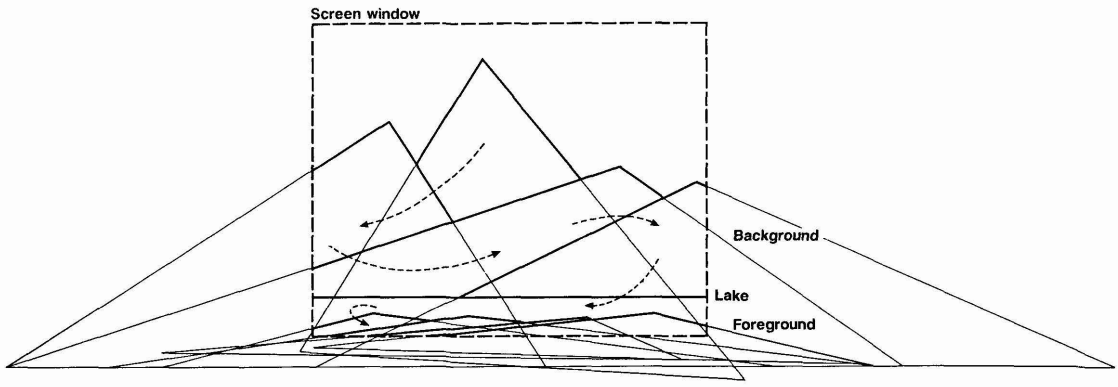


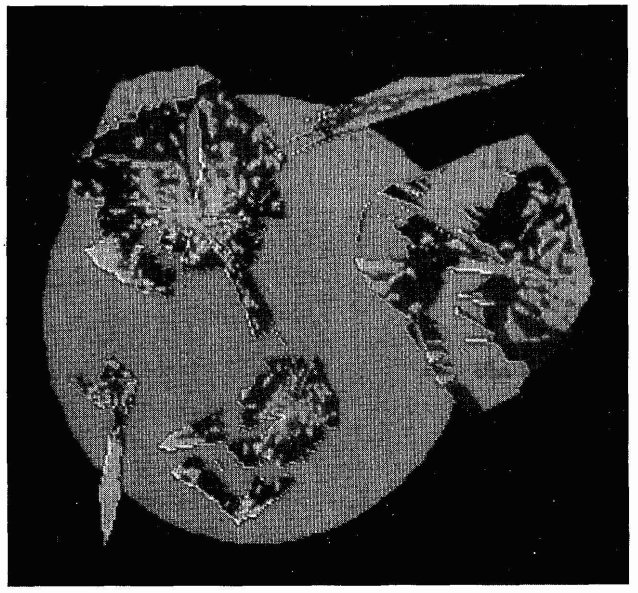
Figure 3.11. Simulating an Alpine scene using triangular midpoint displacement.

each of the most detailed triangles generated, computes the distance from its centroid to the top of the relevant mountain and then chooses the color randomly but within the limits imposed by the value of this distance. The effect of creasing, however, is quite clear in this picture although this can clearly be used to advantage in that real mountainous terrain often shows this type of creasing due to differences in underlying geological structure.

We have used the same method to copy the Mandelbrot–Voss planetrise picture shown earlier in Plate 3.1. In this case, we use a solid blue circle on which a triangular continental land mass is placed. This land mass is then rendered using triangular midpoint displacement. The colors are chosen in the same way as those determined in the Alpine scene in Plate 3.2. The centroid of the basic land mass is computed and the further away the centroid of each individual triangle at the most detailed level is, the more likely the triangle is to be colored green, the less likely to be colored yellow. This generates reasonably realistic continental land masses. At the same time, islands are spawned from this, for the choice of color is also extended to the generation of blue sea in the peripheral areas of the land masses. However, our pictures are very much in the spirit of Voss's (1985) fractal forgeries in that to generate the planet in the plane, we let the continents overlap the edge of the circle and simply clean them off once the detail of the planet's terrain is complete. This is shown in Figure 3.12(a).

We have also used two other elements to generate the illusion that the picture is a true three-dimensional rendering when it is only two. First we have constructed a lunar-like landscape by triangular displacement and into this we have introduced some oval shaped craters. The colors chosen for this part of the landscape – black and yellow – give high contrast to the picture as the planet is based on blue, yellow and green, typical of the colors of the earth seen from space. Finally we have introduced a light source which, like the sun, is a long distance away from the planet. This means that one side of the planet is dark. In Plate 3.3, we show the planet and its final rendering through four stages of construction. In fact, the picture is sufficiently realistic on the fourth iteration for no further rendering to be necessary, although this is because the scene has been generated on a small computer with a low resolution screen of the order of 320×256 pixels. Note that in both these pictures – the mountainscape and the planet-

(a)



(b)

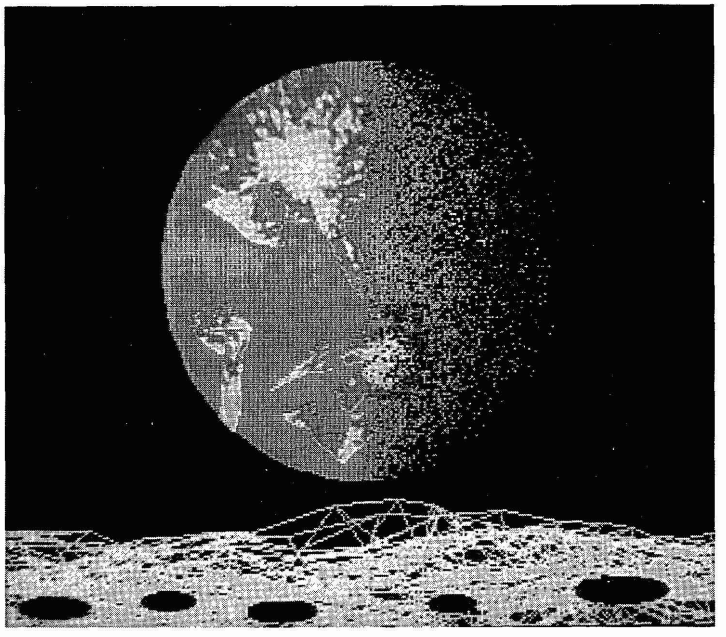


Figure 3.12. A simple planetrise: (a) construction; (b) final rendering.

scape – only four colors were employed, but even these can still be used to realistic effect.

Saupe (1991) says: "In order to generate a fractal, one does not have to be an expert in some involved theory. More importantly, the complexity of a fractal, when measured in terms of the length of the shortest computer program that can generate it, is very small". This statement can be borne out in the applications which are featured throughout this book, but it is particularly pertinent to the examples of this section. To generate the planetscape in Figure 3.12 and Plate 3.3 requires 165 statements in BASIC with an additional 15 relating to the input data. With some optimization of this code, this can be reduced to around 120 statements. What is so remarkable about fractals is that their realism increases dramatically, perhaps exponentially, as their generation at lower levels proceeds. This is very clear in Plate 3.3 where four levels of successive resolution are illustrated. In running the programs associated with this planetscape, the emergence of realism is almost magical as it is observed on the computer screen, although readers must be warned that such realism is in the eye of the beholder who is viewing the picture from a fixed human scale. What might appear realistic would not be so if its scale where enlarged accordingly. For example, zooming in on the fourth level of recursion which demonstrates fractal detail as in Plate 3.3 and scaling this back up to the base scale of the observer, the detail would then look crude and unrealistic. However, if the fractal generation were to continue to orders of magnitude well below the resolution of the computer screen, scaling back up would give sufficient detail to retain the realism.

3.7 Elementary Models of Urban Structure

When we come to apply these ideas to cities and urban systems generally, we require much more elaborate models than those which lie behind the planetscapes and terrain simulated above. These models are simplistic in the extreme, based on common observations of how landscapes look and even in these contexts, to increase the realism further requires models of erosion and weathering which build on more formal ideas in geomorphology. In developing fractal geometry in city simulation, some rudimentary theory about what activities and land uses are located where, must be used, and this means that theories of location and urban structure which form the basis of urban economics, transportation and human geography, are required. In this section, we will introduce the most elementary of such theories and use the resulting model to determine what activity or land use is to be located in each of our zones or sites of the city, and where in turn these sites are generated, using hierarchical triangular midpoint displacement.

Cities and their activities and land uses clearly manifest forms which are self-similar as we demonstrated in Chapter 1 and loosely alluded to in terms of fractals in Chapter 2. At higher levels of spatial aggregation, for example at the regional level, self-similarity is directly invoked in central

place theory (Woldenberg and Berry, 1967). The idea of modeling self-similarity at different scales involves finding an appropriate generating function which can be applied to each scale in a recursive manner. A simple example might be based on central place theory or on theories of the disposition of neighborhoods and district centers within cities: such a rule might involve market area, range of good, population served, the variety of services and goods available at different hierarchical levels and so on. Such a function would be applied first to the largest center, and then follow the rank–size distribution through lower order centers. We are able to use any method which subdivides the original space into regular numbers of subspaces, quadrants, whatever, at each level of the hierarchy and the generation would continue until the lowest order of center is reached. Clearly the recursive rule involves locating lower and lower orders of non-overlapping subdivision. The method we will use will begin with one or four spaces and continually subdivide these by four at lower levels, leading to an hierarchy of locations ordered from 1 to 4, 4 to 16, 16 to 64, and in general for any iteration k, 2^{k-1} to 2^k. We will explain the hierarchical nesting in detail in the next section.

The hierarchy we have just alluded to might equally well be an artifact of the method as it clearly is in the landscape examples given earlier. It does not have to have substantive meaning at each level for it to generate realistic scenes or locations. However, in the examples of cities, we will attempt to give the hierarchy more substantive meaning in terms of location theory and the perception of space at different scales. Central place theory and neighborhood hierarchies have already been mentioned, but there are also hierarchies of traffic routes, public and private services, firms in terms of their spatial organization from regions to the local level and so on, as we implied in Chapter 1. As we also noted there, treating cities as hierarchies is somewhat controversial for a number of studies, notably that by Alexander (1965), argue that hierarchy is too simplistic an ordering device, that activities and land uses in cities are composed of overlapping areas whose order is more lattice- than tree-like. However, this takes us to questions of the rationale for such hierarchies, and we will postpone this until the next section.

We have already introduced the notion that appropriate models of urban activity are to be used to predict the land use/activity type at each level of fractal detail, thus forming a basis for rendering. In this first application, however, we will only use such models to predict land use at the lowest level, not at intermediate levels which would imply that the hierarchy used in simulation has substantive meaning. Thus once a lowest branch in the hierarchy is reached, the model is invoked to enable activity types to be determined. Here we have assumed that there are three key urban activities in one-to- one correspondence to land uses: these are commercial–industrial ($u = 1$), residential–housing ($u = 2$), and open space–recreational ($u = 3$) where the index u defines the particular land use–activity in question. The model for these activities is based on a simple distance relationship to the central business district (CBD) where the profiles of land use type imply that different land uses dominate different concentric rings. These are the so-called von Thunen rings which characterize the organization of land use in strongly monocentric cities. In general these profiles are structured so

that commercial–industrial land uses dominate the core and inner areas of the city, residential housing the peripheral areas of the city and the inner suburbs, with open space more randomly configured throughout the city. These patterns have been central to theories of urban structure and location from urban ecology in the mid-1920s to contemporary urban economics which began with Alonso (1964).

The general form of the model predicts a probability $p^u(r)$ which is a function of the distance r from the CBD specific to each land use u. This is given as

$$p^u(r) = a^u + b^u\ (r - R^u),\ u = 1, 2, 3, \tag{3.29}$$

where a^u, b^u and R^u are parameters whose magnitude and sign control the profile of the probability distribution with respect to distance from the CBD. The precise forms of these equations for the simulation which will follow can now be stated. For the commercial–industrial activity $u = 1$, equation (3.29) can be written

$$p^1(r \leq 400) = 1.38 - 0.0074r$$

where the probability declines inversely with distance, and is near to 0 when $r = 186$. When the distance is greater than 400, the probability is set at a threshold value of

$$p^1(r > 400) = 0.002$$

reflecting a minimum threshold on the existence of this activity. It is quite clear, however, if only these equations were to be used, that there would be a break in the profile from $r = 186$ to $r = 400$ where the probability would be 0. To control for this, an additional equation is also applied which is set up as the conditional that

$$\text{if } p^1(r) < 0.04, \text{ then } p^1(r \leq 400) = 0.04.$$

The combined effect of these equations generates the commercial–industrial profile shown in Figure 3.13. Note that the values used are arbitrary and only of relative meaning for they reflect the coordinates for plotting on the particular display used.

Residential land use ($u = 2$) is controlled by a similar set of equations which reflect both positive and inverse distance relations. Then

$$p^2(r \leq 315) = 0.20 + 0.0024\ (r - 30)$$

and

$$p^2(r > 315) = 0.88 - 0.0035\ (r - 315).$$

The effect of these equations is to generate an increasing function of distance from $p^2(0) = 0.128$ to a maximum of $p^2(315) = 0.88$ which then declines to $p^2(516) = 0$. To ensure a minimum value of residential activity, the conditional is

$$\text{if } p^2(r) < 0.05, \text{ then } p^2(r) = 0.05.$$

Finally for open space $u = 3$, the relationship is simply one of inverse distance

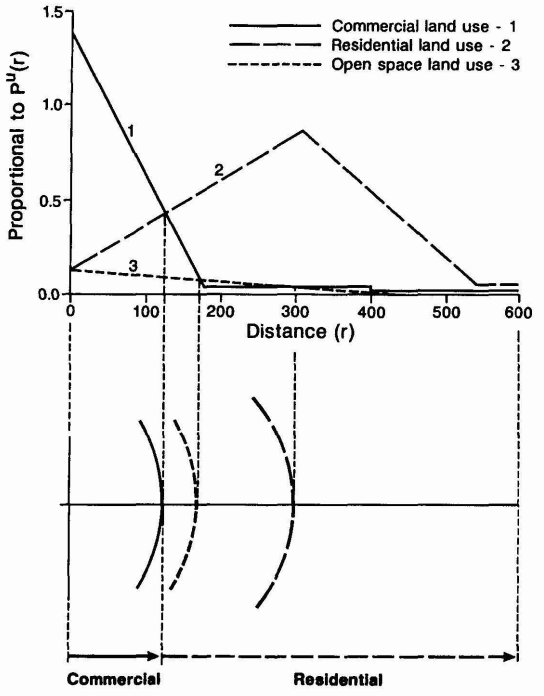


Figure 3.13. Land use profiles and von Thunen rings in the monocentric city.

$$p^3(r) = 0.12 - 0.0002r$$

where the probability declines from $p^3(0) = 0.12$ to $p^3(480) = 0$. To ensure that this function does not predict negative values, the conditional

$$\text{if } p^3(r) < 0, \text{ then } p^3(r) = 0$$

is invoked. These three profiles are shown in Figure 3.13.

Examining these probabilities, it is clear that they are nowhere normalized to exactly sum to 1. We have done this so that when $\Sigma_u\, p^u(r) < 1$, the residual probability is regarded as the probability of vacant land occurring. The overall probability of vacant land occurring is best seen by visually aggregating the profiles in Figure 3.13 and this implies that as distance increases away from the CBD, the probability of vacant land also increases. The other point is that in the vicinity of the CBD, the probabilities sum to greater than 1, that is $\Sigma_u\, p^u(r) > 1$. This does not constitute a problem because the order in which the activities are considered in the simulation means that commercial–industrial are always allocated first, then residential and finally open space. This achieves the following effects.

The probability structure is first set up in the order of importance of these activities. A range of probability is fixed for each activity as: $rN_0 = 1$, $rN_1 = 1000p^1(r)$, $rN_2 = 1000\ [p^1(r) + p^2(r)]$, and $rN_3 = 1000\ [p^1(r) + p^2(r) + p^3(r)]$. An activity type is allocated by drawing a random number between 1 and 1000. If the sum of the probabilities is greater than 1, then the commercial–industrial land use takes priority, then the residential and finally open space. In fact when $r = 0$, then $rN_1 = 1000 \times 1.38$ and thus the activity will always be

commercial–industrial at the CBD. Only when $r > 50$ will other activities be 'competing' for allocation. However, when $r > 550$, $rN_3 = 6$ and effectively all the activity will be vacant land. In essence, this marks the boundary of the city. These equations thus control many dimensions of urban activity allocation and physical form. The shape of the city can be quite radically altered by changing the parameters a^u, b^u and R^u. The values used were fixed by a process of trial and error simulation as well as being judged consistent with simple urban bid-rent and population density theory which we refer to in later chapters.

We have thus defined a simple model of urban land use location which operates through functions which imply the importance and dominance of each land use at different distances from the CBD. Such a monocentric model is of course a gross simplification. It is not unlike the 'model' we used to render the slopes of the mountainscape in the last section. Nevertheless, it does provide a useful rationale for urban location and much of the theoretical edifice of urban economics and human geography is built upon these basic ideas. However, our focus here is not upon developing the best model but upon using a rudimentary model of urban structure to provide a rationale for 'coloring' the city using triangular midpoint displacement. To this we now turn.

3.8 Fractal Cityscapes: The 'London' Sequence

As we implied above, we will now operationalize the model within the context of triangular midpoint displacement for an urban system with the broad dimensions of a world city such as London or Tokyo. A justification for fractal rendering of the sites of the city at its lowest level is based both on our casual and more formal observations that cities display such irregular patterns. Such patterns are formed from individual sites and parcels whose irregularity is conditioned by a myriad of historical, social and physical characteristics. Such patterns are impossible to describe in detail, and defy conventional modeling over a range of scales, although we do know the general principles and reasons as to how and why such patterns are formed. The patterns do in fact appear to be fractal, and thus a first attempt in unraveling their structure can be based on fractal simulation. This is an important point which we cannot stress too much. This chapter is about using fractals to generate a perceived realism in which traditional urban models might be embedded. This is a much more modest goal than designing a complete fractal model, although our models will become more complete as the chapters unfold.

Here we not only acknowledge that fractals are useful in identifying the basic processes at work in cities, but that they are useful in more superficial ways – for rendering the forms produced by traditional models, thus making their outputs more visually acceptable. Such a goal is important in communicating problems, plans and policies in ways in which decision-makers best understand. As we continue, our focus will begin to change as we move towards models based on better founded urban theory, but

we will still retain an emphasis on their visualization using state-of-the-art computer graphics. The method we have developed begins by dividing the urban space, which we define as a circle centered on the CBD, into 10 identical triangular wedges or sectors. Each sector is then subjected in turn to hierarchical subdivision, and once the appropriate level of fractal detail has been reached in any sector, the simulation moves to an adjacent one and begins afresh. The process starts with the due eastern sector and rotates in counterclockwise fashion until all the sectors have been treated. We consider that the use of the triangular lattice rather than a square grid is possibly more appropriate to highly polarized cities where the development has occurred historically from the CBD to the periphery, although the lattice used should make little difference to the simulation.

Let us first define the spatial units or zones in question. The original circular space is subdivided into 10 sectors, each sector referred to as Z_θ where θ is an index reflecting the angular orientation of the sector in question. Within each sector, the zones are referred to by $Z_k(s)$ where k is the zone in question and s is the hierarchical or recursive level. From each branching of the hierarchy, there are K zones, $k = 1, 2, \ldots, K$. Over the levels of the hierarchy given by recursive levels $s = 1, 2, \ldots, S$, particular zones are referred to in the sequence $i, j, k, \ldots$, where i is a typical zone on the $(s - 2)$th level, j is a zone on the $(s - 1)$th, k is a zone on the sth level and so on. The generating rule used to subdivide zones from one level of the hierarchy to the next is given as

$$Z_k(s) = G_k[Z_j(s - 1)], \quad j, k = 1, 2, \ldots K, s \geq 1, \tag{3.30}$$

where j is the zone being subdivided on level $s - 1$ and G_k is the subdivision operator. A particular sequence of zones can now be generated in the following way. The process is begun by applying the rule in equation (3.30) to the original sector Z_θ

$$Z_i(0) = G_i [Z_\theta], \quad \theta = 2\pi/10, 4\pi/10, \ldots, 2\pi, i = 1, \ldots, K. \tag{3.31}$$

Recursion on equation (3.30) using equation (3.31) leads to the sequence

$$Z_n(s) = G_n [G_m [\ldots G_j [G_i [Z_\theta]] \ldots]]. \tag{3.32}$$

Because K zones are generated from each branch in the hierarchy, it is easy to show that at the sth level down the hierarchy, there a total of K^s zones. There is also need for a stopping rule to end the recursion.

In our case, we are subdividing space to form a triangular mesh. The original segment Z_θ is divided into four triangles in the manner shown in Figure 3.14 where $K = 4$. From this diagram, it is clear that at recursive level $s = 1$, there are four sectors in the original segment, at level $s = 2$, 16; at $s = 3$, 64, and so on. The stopping rule is based on the level of resolution below which further spatial detail is not required. In this case, this is the level of pixel resolution of the display (which is 320×256 pixels). A quick calculation shows that with 10 sectors, when $s = 6$ we are below the level of resolution of the screen, and thus in the sequel we will find that fractal detail can be most clearly articulated at levels $s = 4$ and $s = 5$, not greater. We have chosen G_k to reflect the subdivision of triangular space into four triangles in the manner shown in Figure 3.14. This involves midpoint displacement of each side in a constrained random fashion, the degree of con-

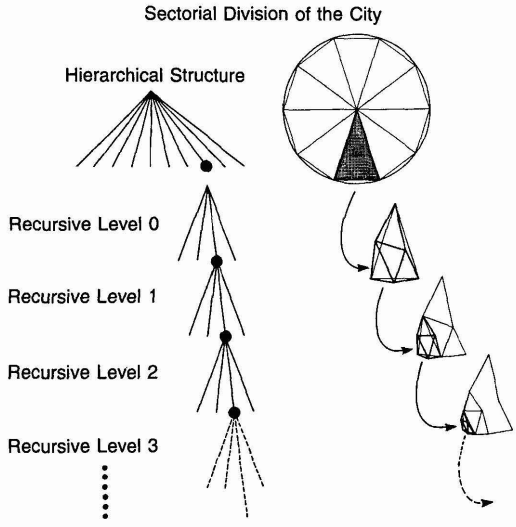


Figure 3.14. Fractal rendering of the monocentric city.

straint reflecting the degree of irregularity, hence the fractal dimension of the resulting surface as described earlier in this chapter. The algorithm used to effect the displacement uses simple trigonometric functions to compute the associated coordinate pairs which define the triangular mesh. The degree of randomness introduced is difficult to quantify in any simple way, but it is reflected in the displacements in Figure 3.14.

The fractal simulations involve a straightforward concatenation of the recursive generating process (in Figure 3.14 and equations (3.30) to (3.32)) with the general model structure (in Figure 3.13 and equation (3.29)). To demonstrate the dependence of pattern and shape on the level of recursion, we have run the model with distances and scale similar to those of Greater London (GLC, 1985) for levels of recursion $1 \leq s \leq 5$. This produces five simulations which are presented in Plate 3.4 where the colors blue, red and green represent commercial–industrial, residential, and open space–recreational land uses respectively. These show quite different patterns. Up to level $s = 2$, the pictures reveal the coarseness of the triangular mesh used to generate shapes of land use activity. Moreover, not enough zones are generated to achieve a reasonable distribution of activity types. However, for $s > 2$, the pattern becomes much more acceptable; but when $s > 5$, which touches the level of pixel resolution, the pattern looks more like a pointillist painting than a city. The most appropriate-looking images are thus generated for $s = 3$ and $s = 5$. This is an important point in the simulation of visual realism, and it also suggests that the probability structure of the underlying model is not invariant to scale, an issue which in some senses is obvious, but one which has rarely been explored in the mainstream of research.

These types of simulation do, however, reveal the inadequacies of conventional urban models in terms of their spatial patterns and visual realism. The images shown in Figure 3.14 are too compact in that one might expect a much greater spread of development unconnected to the main city but indicative of the way development hops around on the edge of a large city.

Despite the preset wedge-sector geometry which provides the template for the city, these patterns do not display the classic corridor effects which characterize the typical radial–concentric city. Compare these, especially the images for $s = 3$, 4 and 5 in Plate 3.4, to those in Chapter 1 – Figures 1.15 and 1.16, and Chapter 7 – Figures 7.2 to 7.5 and Plate 7.1, which illustrate real urban agglomerations. The advantage of fractal simulation thus becomes clear. Spatial effects in models are immediately clarified, and systematic biases can be detected and corrected. Only large-scale simulations can achieve this, and the pictures in Plate 3.6 speak for themselves.

Finally, although the broad shape of our simulations reflect those of London, these simulations are as much 'London' as are the Mandelbrot–Voss planetrise pictures shown earlier which are implied to be the 'Earth' as seen from the 'Moon'. This is a very important issue in fractal graphics for in this case, it suggests the sorts of elements required in order to generate minimal city forms. The whole feel to the images for $s \geq 3$ is that of a large monocentric city like London. In fact, we have cheated slightly by adding the distinctive River Thames to the images after they have been generated. This is a strong perceptual clue to any picture but even without it, the images for $s \geq 3$ reflect a large city like London. In our fully-fledged simulations which we will develop in the next chapter, we will in fact omit the river for in these simulations which will actually be of London; the shape of the city will be encoded in the input data which reflect the built-up area and the Greater London County boundary.

The examples we have ended with in this chapter constitute a good basis for experimentation, in terms of the mechanisms of developing cities in physical terms, of exploring model structures through their causal chains, and of judging visual realism. A particularly important issue is to find out the way parameters might combine with one another to generate realistic and unrealistic morphologies, and the 'London' sequence developed here provides a firm basis for this. In the next chapter, we will extend our hypothetical model by setting up a computer laboratory to generate a variety of experiments in visualizing urban form. This, however, will be but an initial foray into this kind of experimentation, and as such represents a powerful line of inquiry which we will leave for future research. We will also progress our simulations forward by developing more realistic models which we render with fractal midpoint displacement, and the emphasis will turn to explicitly fitting these models to data. The great strength of fractal models which generate picturescapes is that they provide a way of making our theories more real and of communicating more meaning to our analyses. For the first time we can move away from but still retain the logic of our theoretical models which hitherto have usually been regarded as extreme cases; with a little imagination, we can render these more realistically without losing the need for high theory. Fractal rendering represents a powerful way of achieving this, and in the next chapter, we will demonstrate how this is possible in the real as well as in the imaginary world.