Estimating Fractal Dimension from Range Images of Natural Terrain

Kenichi Arakawa and Eric Krotkov

July 1991
CMU-CS-91-156$_2$

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

This research was sponsored by NASA under Grant NAGW-1175. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of NASA or the U.S. government.

Keywords: Vision and Scene Understanding, Fractals, Range Data

Abstract

We investigate two published approaches to the problem of estimating fractal dimension: the box-counting approach, and the fractal Brownian function approach. In experiments with synthetic images, we find the fractal Brownian function methods proposed by Pentland and Yokoya to be superior to the box-counting approaches described by Voss, because the former do not require data sampled at equal intervals, and are more robust to Gaussian noise. For experiments with real images, we extend the fractal Brownian function methods to accommodate irregularly sampled data supplied by a scanning laser rangefinder. Applying the extended methods to noisy range imagery of natural terrain (sand and rocks), we find (1) that the resulting estimates of fractal dimension correlate closely to the human perception of the roughness of the terrain, (2) that it is appropriate to model the natural terrain studied as a fractal Brownian function, and (3) that the fractal dimension of the sensed point set is a practical and effective measure of the roughness of natural terrain.

Contents

List of Figures

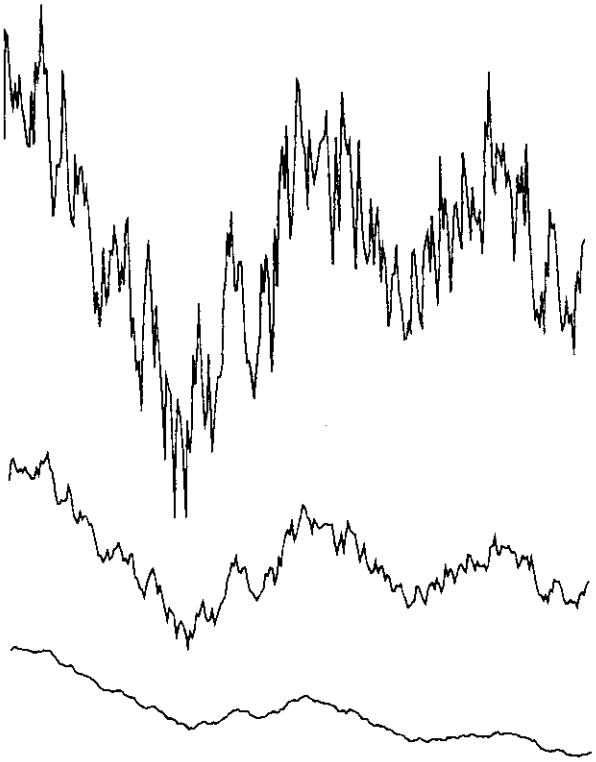


Figure 1: Synthesized one-dimensional fractal patterns
The figure shows three fractal patterns exhibiting fractional Brownian motion. For the upper pattern, the fractal dimension is $D = 1.8$. For the middle, $D = 1.5$, and for the lower, $D = 1.2$.

1 Introduction

Many problems in the analysis of natural surface shapes and the construction of terrain maps to model them remain unsolved. One reason is that the familiar Euclidean geometry of regular shapes, such as surfaces of revolution, does not capture well the irregular and less structured shapes found in nature, such as a boulder field, or surf washing onto a beach.

Mandelbrot [12, 13, 14] proposed fractals as a family of mathematical functions to describe natural phenomena such as coastlines, mountains, branching patterns of trees and rivers, clouds, and earthquakes. Figure 1 illustrates three fractal patterns with different fractal dimensions. Since Mandelbrot introduced them, fractal sets and functions have been found to describe many other environmental properties [1], and have received a great deal of attention from scientists, artists, and others.

Researchers in computer graphics and image understanding have applied fractal theory to a variety of problems. In computer graphics, fractals have been used to synthesize models of complex natural objects such as trees, mountain ranges, and clouds [20]. The rendered images and models exhibit a high level of realism compared to previous efforts. In image understanding, fractals have been employed to analyze projections of natural scenes. Since our research concentrates on *analysis* rather than synthesis, in the following we review briefly the image understanding literature

on fractal analysis, which addresses modeling the shapes of natural objects, segmenting textured surfaces, and interpolating natural surfaces.

For modeling shapes of natural objects, Pentland [18] presents a method to estimate fractal dimension from the "second-order statistics" of image intensities. He goes on to show that measurements of fractal dimension may be used to measure perspective gradient, thus providing an independent check on estimates of surface orientation derived from analysis of foreshortening. Kube and Pentland [9] report that, given certain assumptions (Lambertian reflectance, modest surface slopes, and the absence of occlusions and self-shadowing), a fractal surface with power spectrum proportional to $f^{-\beta}$ produces an image with power spectrum proportional to $f^{2-\beta}$.

Keller et al. [8] estimate the fractal dimension using a least-square linear fit, and use it to distinguish silhouettes of trees from silhouettes of mountains. Dubuc [2] and Dubuc et al. [3] develop a method, called the variation method, for estimating the fractal dimension of both curves and surfaces. Maragos and Sun [15] use morphological operations with varying structuring elements to evaluate the fractal dimension of arbitrary fractal sets. They develop an iterative optimization method that converges to the true fractal dimension.

For segmenting textured surfaces, Pentland [18] computes a self-similarity parameter and uses it to segment real images of scenes of urban areas, mountains, and deserts. He reports classification accuracies of 84 and 88 percent on Brodatz texture patterns. Peleg et al. [17] derive a set of 48 features based on the ϵ-blanket method of estimating fractal dimension suggested by Mandelbrot [13]. They use these features as global characteristics to recognize 128×128 patches of natural textures. Medioni and Yasumoto [16] report that fractal dimension alone does not provide sufficient discriminatory power to classify natural textures. Their results are confirmed by Keller et al. [7], who use estimates of fractal dimension together with features based on the concept of lacunarity as global signatures of texture and as local measurements of texture for segmentation purposes, They present segmentation results for 8 Brodatz texture patterns, an outdoor scene, and a variety of artificial patterns.

Kaneko [6] extends the concept of fractal dimension (as described and applied to texture analysis in his earlier work [5]) to a fractal matrix. This is a multi-dimensional generalization of the fractal dimension, which is a scalar. He applies the fractal matrix model to analyze and classify 13 Brodatz patterns.

For interpolating natural surfaces, Yokoya et al. [22] adopt a recursive midpoint displacement scheme using four neighbors to interpolate natural surfaces. They compute two features—a self-similarity parameter and the standard deviation of the distribution function—and use them for stochastic interpolation in order to preserve the statistical characteristics of the true surface.

The work on modeling natural shapes, segmenting textured surfaces, and interpolating natural surfaces represents significant advances. In this paper, we extend the work on estimating fractal dimension to patterns of natural terrain acquired by a laser rangefinder sensor [11]. Using the sensor, we can only acquire fairly noisy depth data and the observed points are not spaced regularly in Cartesian system. Therefore, we first refer to some proposed techniques for estimating fractal dimension and address their applicability to our purpose. In Section 2, we review some of the techniques proposed for estimating fractal dimension from digitized data. Next, we investigate and compare experimentally two of the techniques, paying attention to the issues of robustness against

sensor noise, and applicability to irregularly sampled data. Finally, we discuss the results, and conclude that the fractal dimension estimates can be used as a reasonable measure of the roughness of natural terrain.

2 Estimating Fractal Dimension

Of the many methods proposed to estimate fractal dimension, we concentrate on two: box-counting approaches, and fractal Brownian function approaches. Because others have already surveyed the related literature [2], we do not consider other important methods, for instance power spectrum methods [18].

In this section, we first define some terms related to self-similarity and self-affinity. Then, we state the two approaches, as reported and developed in the literature, and discuss their strengths and limitations.

2.1 Definitions

In a Euclidean space of dimension E, consider a set S of points $\mathbf{x} = (x_1 \ldots . x_E)$. After scaling by r, $0 < r$, the set S becomes rS, with points $r\mathbf{x} = (rx_1 \ldots . rx_E)$.

The set S is *self-similar* when S is the union of N distinct (non-overlapping) subsets, each of which is identical, up to translation and rotation, to rS. The fractal dimension D of self-similar S then satisfies

$$1 = Nr^D \text{ or } D = -\log N / \log r. \tag{1}$$

The set S is *statistically self-similar* if it is composed of N distinct subsets, each of which is scaled by ratio r from the original, and is identical in all statistical respects to rS. The fractal dimension of statistically self-similar S is given by (1).

A collection of real scaling factors $\mathbf{r} = (r_1 \ldots . r_E)$, with $r_j > 0$, determines an *affinity* Ψ, where Ψ transforms $\mathbf{x} \in S$ into $\Psi(\mathbf{x}) = (r_1 x_1 \ldots . r_E x_E)$. This operation transforms S into $\Psi(S)$ by scaling different coordinates by different amounts. The set S is *self-affine* when S is the union of N distinct subsets, each of which is identical to the sets transformed by affine Ψ. If the condition of invariance under non-uniform scaling is satisfied statistically, the set is *statistically self-affine* (cf. the definition of statistical self-similarity).

In (1) we defined fractal dimension in terms of the self-similar set S. We can also define it in terms of the self-affine set S [21], but for the purposes of this paper, we will define it in terms of one particular class of self-affine shapes (ff. 2.3).

2.2 Box-Counting Approach

Counting the number of boxes of various sizes which cover a fractal pattern is one method for estimating fractal dimension. We explain this box-counting approach following Voss [21].

Consider a one-dimensional fractal pattern, for instance, a coastline. Mandelbrot [12] observes that the apparent length of the pattern varies with the size of the measuring ruler. To see this, assume that the self-similar pattern is of maximum size L_{max}. For a smaller scale $r < 1$, the size is a fraction of the maximum size $L = rL_{max}$, and the pattern consists of $N = 1/r^D$ segments of length L. Thus,

$$Length = LN = L\frac{1}{r^D} = L\left(\frac{L_{max}}{L}\right)^D = \frac{L_{max}^D}{L^{D-1}} = L_{max}^D L^{(1-D)}.$$

This expression shows clearly that the measured length varies inversely with L; the shorter is the ruler, the longer is the measured length of the pattern. Taking logs, the expression for length can be rewritten as

$$\log(LN) = (1 - D)\log L + K .$$

where $K = D \log L_{max}$ is a constant. The plot of this equation on log-log axes is linear; the slope of the line is $(1 - D)$. Therefore, by measuring N segments with different rulers of size L, and identifying the line that best fits the measurements, we can determine the fractal dimension from the slope of the line. This does not require knowing L_{max} or r.

The fractal dimension D characterizes the *covering* of the set S by E-dimensional "boxes" of linear size L. If all of S is contained within one box of size L_{max}, then each of the $N = 1/r^D$ subsets will fall within one box of size $L = rL_{max}$. Thus, the number of boxes of size L, $N_{box}(L)$, needed to cover S, is

$$N_{box}(L) = \left(\frac{L_{max}}{L}\right)^D . \tag{2}$$

Taking logs and rearranging yields

$$\log N_{box}(L) = -D \log L + K .$$

If these functions are differentiable, the fractal dimension D is

$$D = -\frac{\partial \log N_{box}(L)}{\partial \log L} . \tag{3}$$

so as above, we can estimate D as the slope of the line that best fits the points $(\log L. - \log N_{box}(L))$. If the function $\log N_{box}(L)$ is not differentiable, we can estimate the slope by the ensemble average of many pair-wise slopes

$$\frac{\log N_{box}(L_i) - \log N_{box}(L_j)}{\log L_i - \log L_j} .$$

Voss proposed a method for measuring fractal dimension using *Mandelbrot measures* [21]. The method is based on a statistical expansion of fractal dimension, and it is more robust against noise in the samples than is the previous method.

Now consider an object, defined by a set S of points, that is statistically self-similar. We assume all of the points to be equivalent. The spatial arrangement of these points determines $P(m.L)$, which is the probability that there are m points within an E-dimensional box (or sphere) of size L, centered on an arbitrary point in S. $P(m.L)$ is normalized such that, for all L,

$$\sum_{m=1}^{N} P(m.L) = 1 .$$

The number of boxes of size L needed to cover S is

$$N_{box}(L) = \sum_{m=1}^{N} \frac{1}{m} P(m.L) .$$

As above, the fractal dimension is computed as the slope of a line that best fits the points $(\log L. - \log N_{box}(L))$.

For E-dimensional patterns of N data points, the computational complexity of the box-counting method is $O(NL_{search})$, and that of Voss' method is $O(NL_{search}^{E+1})$, where the maximum search size is L_{search}. Thus, the complexity of Voss' method is significantly larger than the box-counting method.

Keller et al. [7] suggest that the fractal dimension estimated by Voss' method saturates as it approaches 3.0, even on synthesized two-dimensional fractal patterns. For small L, and fractal dimension approaching 3.0, some cubes will contain only the point about which the cube is centered, because the points are widely spaced. We can increase L up to the size of the image, but no more. Therefore, we cannot count accurately the number of points in many cubes, which effectively decreases the estimated dimension. Keller et al. proposed a refinement for the purpose of normalization of each scale, in which the two-dimensional pattern between the center point of a cube and each of its adjacent points is approximated by linear interpolation. This technique distorts (smooths) the original pattern, which may itself decrease the estimated fractal dimension.

The box-counting methods apply to regularly sampled patterns, such as images. They do not apply to irregularly sampled patterns, such as terrain maps constructed from sparse sensor data. Modifying the box-counting approach to account for irregularity would require deforming the pattern, for example, by interpolation.

2.3 Fractal Brownian Function Approach

One class of fractal patterns is created by a process with fractional Brownian motion. The *fractal Brownian function* approach applies to this class of fractal patterns. In this section, we explain fractional Brownian motion, define fractal Brownian functions, and discuss methods proposed for using them to estimate fractal dimension.

Brownian motion $B(t)$ is a stochastic process defined as follows.

1. $B(0)$ = constant.
2. $B(b) - B(a) \sim N(0.(b-a)\sigma^2)$, for $a \leq b$.
3. $B(b) - B(a)$ and $B(c) - B(b)$ are statistically independent, for $a \leq b \leq c$.

This can be rewritten as

$$B(rt) = r^{1/2}B(t).$$

A trace of $B(t)$ requires different scaling factors in the two coordinates: r for t, but $r^{1/2}$ for $B(t)$. Therefore, it is self-affine.

Fractional Brownian motion $B_H(t)$ generalizes Brownian motion, and is defined as follows.

1. $B_H(0)$ = constant.
2. For constant C,

$$B_H(t) - B_H(0) = C\left[\int_{-\infty}^{0}\left\{(t-s)^{H-1/2} - (-s)^{H-1/2}\right\}dB(s) + \int_0^t (t-s)^{H-1/2}dB(s)\right].$$

A trace of $B_H(t)$ exhibits a statistical scaling behavior. If the scale t is changed by the factor r, then the increments $\Delta B_H(t)$ change by a factor r^H:

$$\Delta B_H(rt) \propto r^H \Delta B_H(t) .$$

Traces of fractional Brownian motion are one class of fractal patterns.

Pentland [18] defined a *fractal Brownian function* as an extension of statistical self-affinity that characterizes self-affine processes, including fractional Brownian motion. A random function $f(t)$ is a fractal Brownian function if, for all t and Δt, there exists H in

$$Pr\left\{\frac{f(t+\Delta t) - f(t)}{\Delta t^H} < x\right\} = g(x) \tag{4}$$

which is independent of Δt, where $g(x)$ is a cumulative distribution function. In this definition, $\Delta f_{\Delta t} = f(t+\Delta t) - f(t)$ is statistically self-affine, and H is a self-affinity parameter, related to the fractal dimension D of $f(t)$ by $D = 2 - H$. If $g(x)$ is a zero-mean Gaussian with unit variance, then $f(t)$ represents fractional Brownian motion $B_H(t)$. If, in addition, $H = 1/2$, then $f(t)$ represents classical Brownian motion $B(t)$.

Interpreting t as a vector quantity $\mathbf{t}$ extends this definition to higher topological dimensions. In this case, the Δt appearing in the denominator of (4) must be re-written as the norm $\|\Delta \mathbf{t}\|$. If $f(\mathbf{t})$ is a pattern in E-dimensional Euclidean space, then $D = E + 1 - H$ [21]. For instance, if we analyze fractal dimension of natural terrain, we can express the terrain as an elevation map $f(\mathbf{t})$ on a horizontal plane $\mathbf{t} = (x, y)$ and the fractal dimension can be estimated by $D = 3 - H$.

Pentland [18] proves that under certain conditions (constant illumination, constant albedo, and a Lambertian surface reflectance function), a three-dimensional surface with a spatially isotropic fractal Brownian shape produces an image (i) whose intensity surface is fractal Brownian, and (ii) whose fractal dimension is identical to that of the components of the surface normal. He also shows that the definition of a fractal Brownian function on intensity $I(\mathbf{t})$–instead of $f(t)$ in (4)—can be rewritten as

$$E(\Delta I_{\|\Delta \mathbf{t}\|}) \|\Delta \mathbf{t}\|^{-H} = E(\Delta I_{\|\Delta \mathbf{t}\|=1}) . \tag{5}$$

where $E(\Delta I_{\|\Delta \mathbf{t}\|})$ is the expected value of the change in intensity $I(\mathbf{t})$ over distance $\|\Delta \mathbf{t}\|$.

To evaluate the suitability of this fractal model for images of natural surfaces, he observed the empirical distributions of intensity differences $\Delta I_{\|\Delta \mathbf{t}\|}$ for different distances $\|\Delta \mathbf{t}\|$. He observed the distributions to be approximately Gaussian. Moreover, he computed the standard deviation $S(\Delta I_{\|\Delta \mathbf{t}\|})$ of each distribution, and found the points $(\log \|\Delta \mathbf{t}\|, \log S(\Delta I_{\|\Delta \mathbf{t}\|}))$ to lie on a line. From this line in log-log space, he estimated the slope H, which is

$$H = \frac{\partial \log S(\Delta I_{\|\Delta \mathbf{t}\|})}{\partial \log \|\Delta \mathbf{t}\|} . \tag{6}$$

Given H, the fractal dimension of the two-dimensional pattern is $D = 3 - H$.

Yokoya [22] also assumed that intensity in images is distributed by a fractal Brownian function, and that $g(x) \sim N(0, \sigma^2)$. He developed a method for estimating fractal dimension similar to

Pentland's. Instead of the standard deviation in (6), he used the expected value $E(\Delta I_{\|\Delta \mathbf{t}\|})$:

$$H = \frac{\partial \log E(\Delta I_{\|\Delta \mathbf{t}\|})}{\partial \log \|\Delta \mathbf{t}\|} . \tag{7}$$

Both methods are reasonably robust against noisy data, because they use statistics computed from a large number of data points. Yokoya's method, in particular, tolerates zero-mean normally distributed sensor noise, because the method implicitly performs an averaging operation.

The computational complexity of both methods is $O(N\|\Delta \mathbf{t}_{search}\|)$ for regularly sampled data, where N is the number of data points, and $\|\Delta \mathbf{t}_{search}\|$ is the maximum search size. Because Pentland's method computes the standard deviation, it requires slightly more computation than Yokoya's method, which computes the first moment.

3 Experiments

We present the experimental methods and results in three parts. First, we apply Voss' method to synthetic fractal images. Then, we apply Pentland's method and Yokoya's method to the same images. Finally, we extend the latter two methods to handle irregularly spaced data, and apply them to range images of rough, natural terrain.

3.1 Box-Counting Approach on Synthetic Patterns

We synthesized synthetic two-dimensional images using the midpoint displacement algorithm given by Saupe [19]. This algorithm synthesize traces of fractional Brownian motion. Figure 2 illustrates isometric plots of two examples, with $D = 2.2$ and 2.5. The size of all synthesized images studeid here is 256×256 8 bit pixels.

We implemented Voss' method following (3), and in order to confirm its effectiveness we used it to estimate the fractal dimension of the synthetic fractal images. Figure 3 shows the result of plotting the Mandelbrot measures computed by Voss' method when $L_{search} = 50$ pixels. The estimated fractal dimensions are 2.204 and 2.460, and the least-square line fit errors, normalized by $\log N_{box}(L)$, are 0.007 and 0.013. The plotted points $(\log L. - \log N_{box}(L))$ are distributed linearly, so the errors are relatively small, as illustrated. The computation time is about 10^4 s on a Sun4/40 with 24 MB of physical memory.

Figure 3 also compares the synthetic and computed fractal dimensions. We observe fairly good correspondence between $D = 2.1$ and 2.5. We observe saturation for $D \geq 2.6$, as noted by Keller et al. [7]. Figure 3 also shows the correspondence between the fractal dimension for synthesis and the estimated fractal dimension on patterns with Gaussian noise $N(0. \sigma^2)$. In this case, we normalized the range of the pattern to lie between 0 and 1. According to the graph, if σ exceeds 10^{-2}, we cannot expect Voss' method to compute a reasonable estimate of the fractal dimension.

Voss' method is not vulnerable to Gaussian noise if we have a sufficiently large number of data points. However, the fractal dimension estimated by Voss' method saturates if the real dimension exceeds 2.6. Therefore, this method would not be effective for estimating the fractal dimension of "rough" sets, such as rugged, natural terrain. Moreover, as pointed out in Section 2, box-counting approaches in general (including Voss' method) are not applicable to irregularly sampled patterns.

3.2 Fractal Brownian Function Approach on Synthetic Patterns

We implemented the methods proposed by Pentland and Yokoya, and applied them to the same synthetic fractal images (similar to those in Figure 2). The synthetic images are isotropic fractal patterns, therefore, we need only take account of differences of pixel values $\Delta I_{||\Delta \mathbf{t}||}$ along rows and columns of the images.

Figure 4 shows experimental results from Pentland's method on synthetic fractal images with $D = 2.2$ and 2.5. We fit lines to the sets of points $(\log ||\Delta \mathbf{t}||. \log S(\Delta I_{||\Delta \mathbf{t}||}))$ with errors 0.015 and 0.014 (these errors are normalized by $\log S(\Delta I_{||\Delta \mathbf{t}||})$). The slopes of the lines corresponding to H are 0.714 and 0.464. Therefore, the estimated fractal dimensions are 2.286 and 2.536.

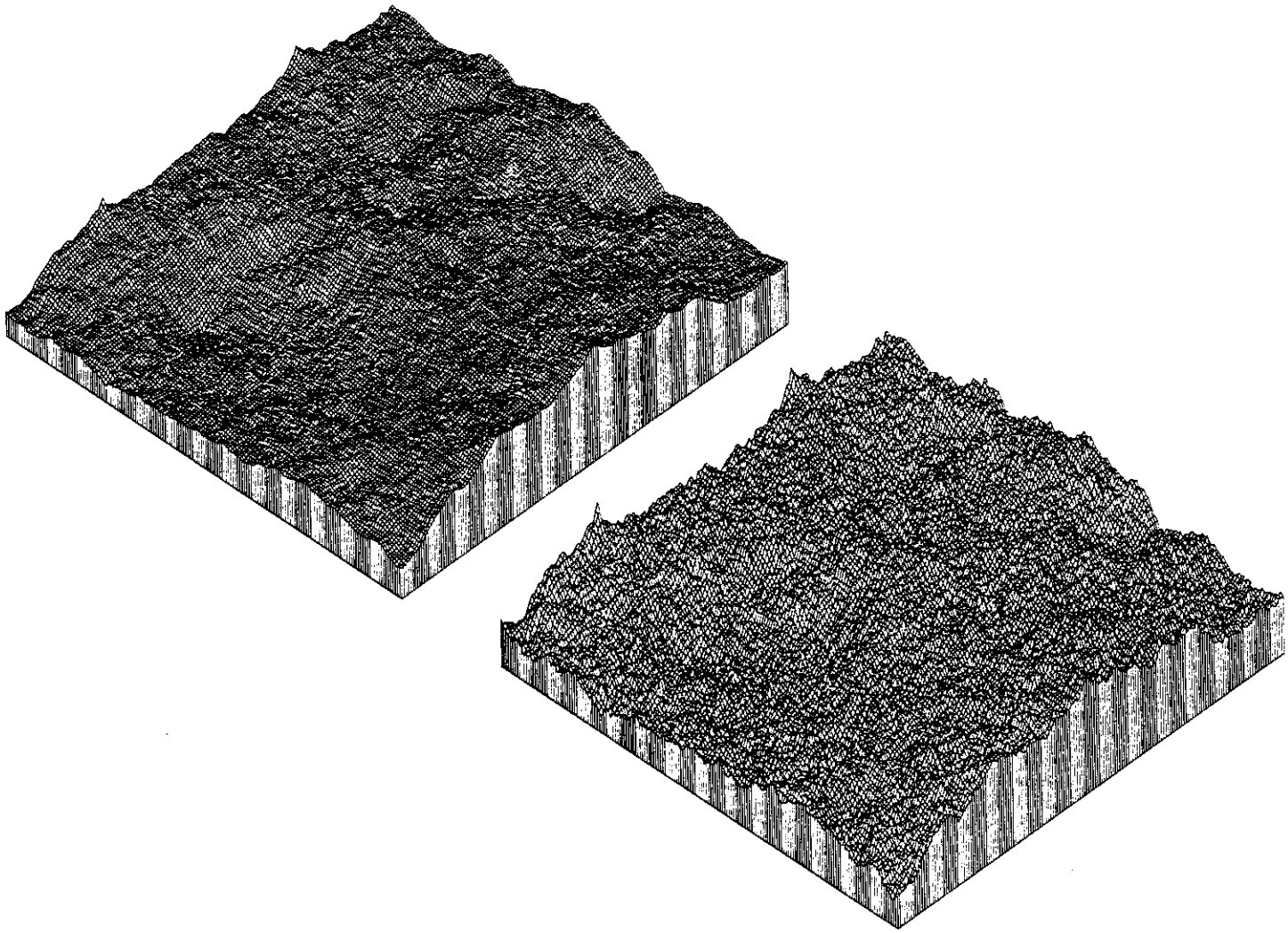


Figure 2: Synthesized fractal images

The figure shows isometric plots of two fractal images. For the image on the upper left, the fractal dimension is $D = 2.2$. For the lower right, $D = 2.5$.

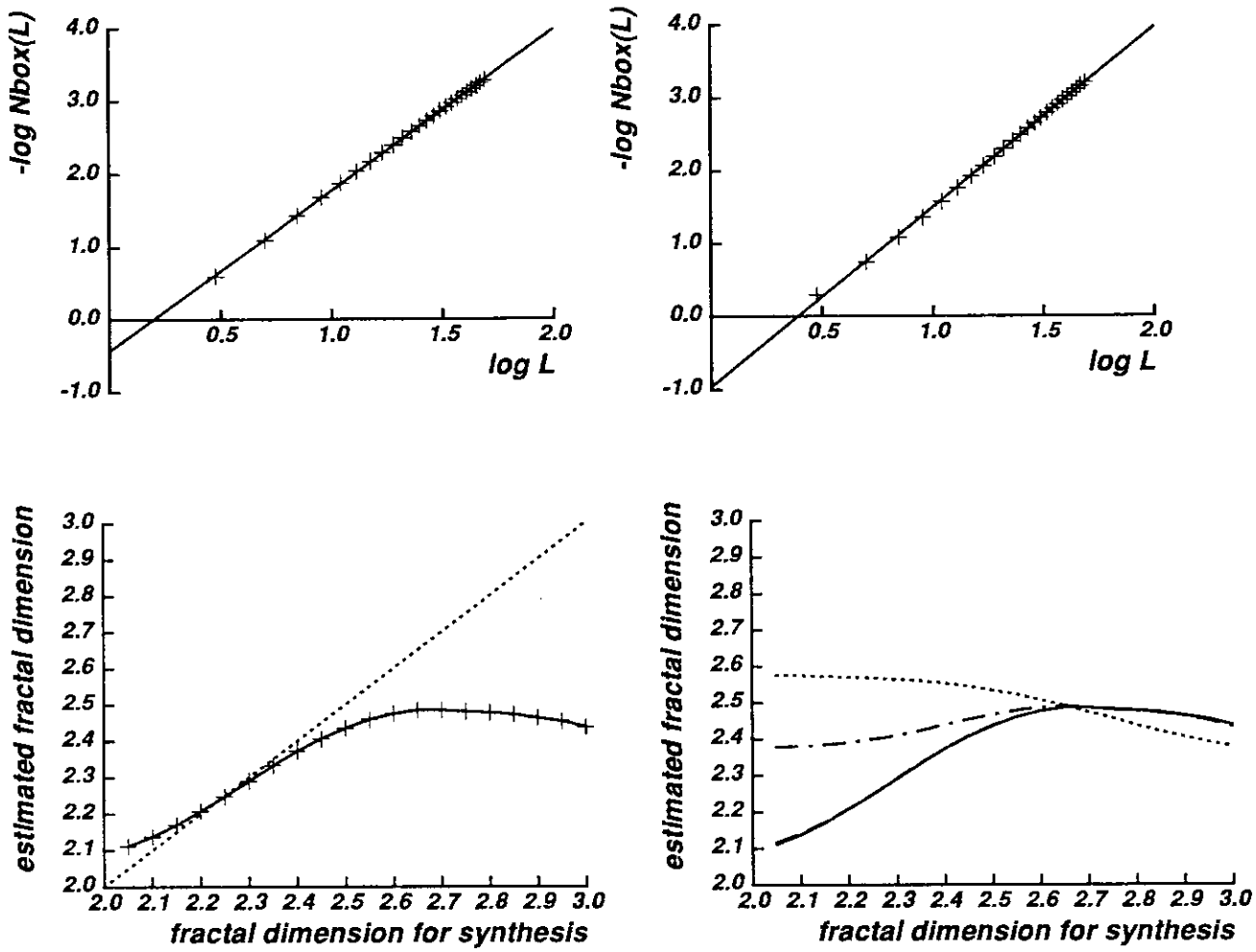


Figure 3: Results and errors with Voss' method

The upper graphs show Mandelbrot measures and a line fitted to them. For the upper left, the fractal dimension is $D = 2.2$, and for the upper right, $D = 2.5$.

The graph on the lower left plots the real fractal dimension against the dimension estimated by Voss' method, with no noise. The dotted line illustrates the ideal relation. The graph on the lower right illustrates the effect of Gaussian noise. The solid line represents estimates from data without noise. The dotted, dot-dashed and dashed lines represent estimates from data with noise with $\sigma = 10^{-1}.10^{-2}$, and 10^{-3} respectively. The line with $\sigma = 10^{-3}$ almost overlays the line without noise.

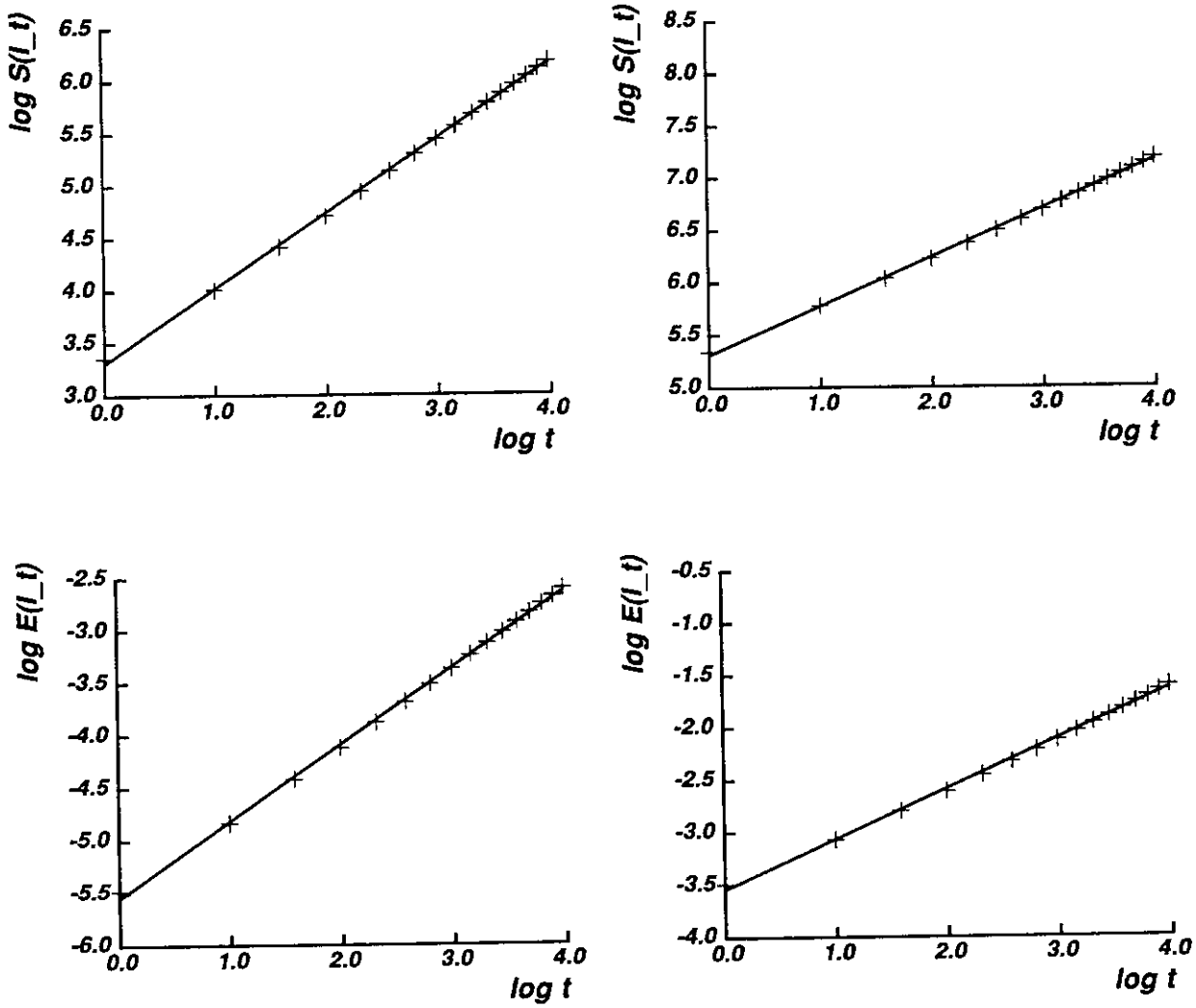


Figure 4: Results with Pentland's and Yokoya's methods

The two upper graphs show points $(\log ||\Delta \mathbf{t}||, \log S(\Delta I_{||\Delta \mathbf{t}||}))$ computed by Pentland's method. The two lower graphs plot points $(\log ||\Delta \mathbf{t}||, \log E(\Delta I_{||\Delta \mathbf{t}||}))$ computed by Yokoya's method. The straight line is fitted to the plotted points. For the graphs on the left, the fractal dimension is $D = 2.2$, and for the graphs on the right, $D = 2.5$.

Figure 4 also shows experimental results with Yokoya's method on the same images. We fit lines to the sets of points $(\log \|\Delta\mathbf{t}\|, \log E(\Delta I_{\|\Delta\mathbf{t}\|}))$ with errors 0.019 and 0.017. The slopes are 0.730 and 0.477. Therefore, the estimated fractal dimensions are 2.270 and 2.523. The computation time for each result is about 60 s when the maximum value of d is 50 pixels.

Figure 5 plots fractal dimension for synthesis against fractal dimension estimated using Pentland's and Yokoya's methods. The estimated values increase monotonically with the fractal dimension for synthesis, and there is no saturation. The fractal dimensions estimated by Pentland's method are slightly larger (by about 0.02) than those estimated by Yokoya's method. Overall, we do not observe any significant difference between the results, and conclude that both methods compute reasonable estimates of fractal dimension over a wide range of scales.

In order to determine the robustness of the methods against Gaussian noise, we repeated these trials on synthetic images with additive noise distributed $N(0, \sigma^2)$. Figure 5 illustrates the results, which do not reveal significant differences between the two. We conclude that if the standard deviation of noise does not exceed 10^{-2}, then both methods compute reasonable estimates of the fractal dimension. This compares favorably to Voss' method (cf. Figure 3).

Based on the above results, we conclude that these two methods are superior to Voss' method because, in practice, they estimate fractal dimension over a wider range of scales, they are as robust in the presence of Gaussian noise, they have lower computational complexity, and they are more easily extensible to irregularly spaced data. Comparing Pentland's and Yokoya's methods, we conclude that there are no significant differences between the results attained, and that there is a minor difference in computation time (Pentland's method requires more computation to find standard deviations than Yokoya's method requires to find expected values).

3.3 Fractal Brownian Function Approach on Range Images

In this section, we apply the fractal Brownian function approach to range images acquired with a scanning laser rangefinder manufactured by Perceptron. The data is elevation rather than image intensity, so we change our notation to use $z(\mathbf{d})$ (with $\mathbf{d} = (x, y)$) instead of $I(\mathbf{t})$.

As we showed earlier, the proposed methods for estimating fractal dimension require regularly sampled data. The Perceptron sensor acquires range images with respect to a spherical-polar coordinate system. Equal sampling intervals in this coordinate system become unequal and irregular when mapped into a Cartesian system. Thus, the Perceptron data points are not equally spaced when expressed in Cartesian coordinates. In this section, we extend the fractal Brownian function approach to accommodate this irregularity.

Procedure

The procedure for estimating the fractal dimension from an input Perceptron range image is stated in the five following steps.

1. Delete corrupted pixels.

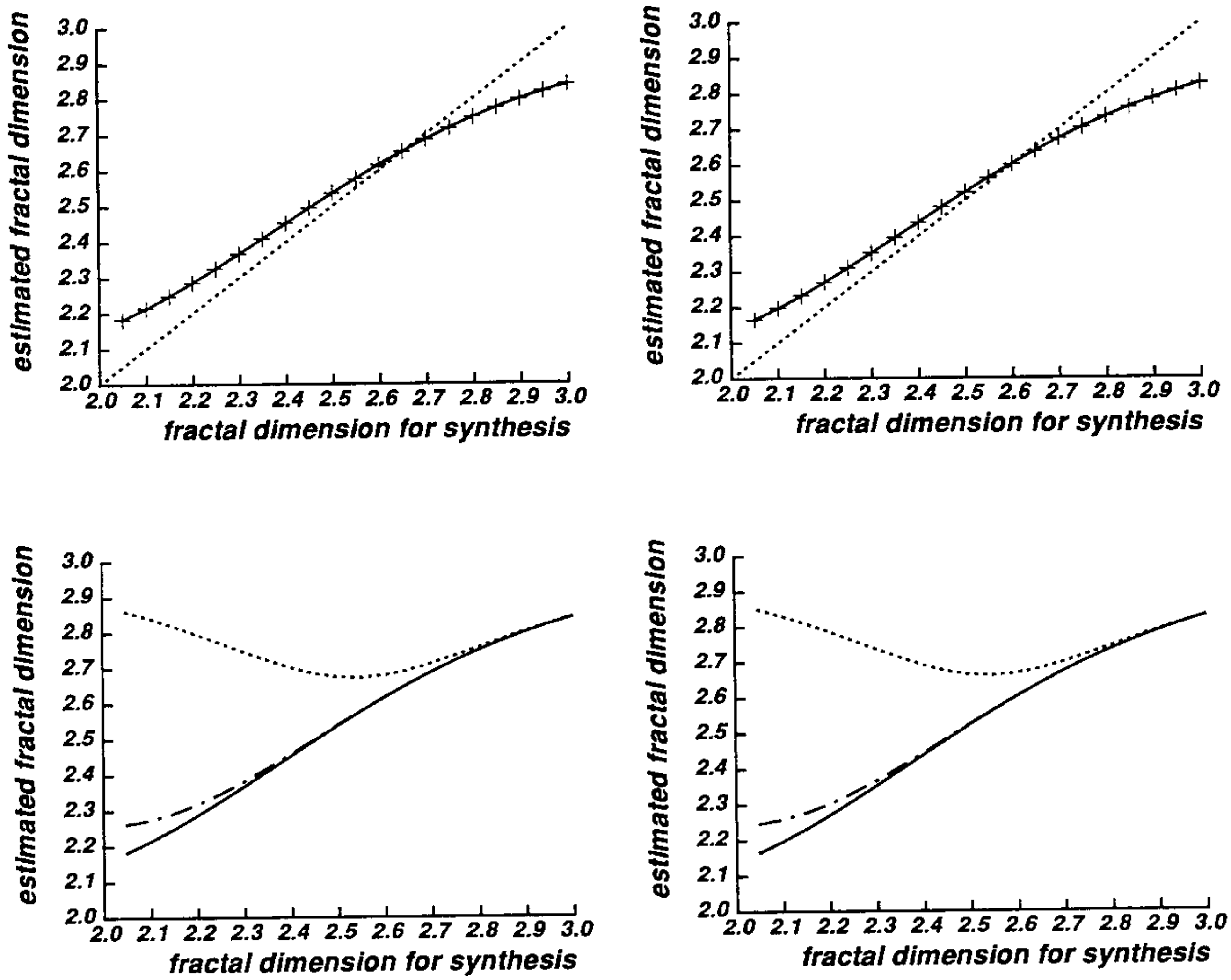


Figure 5: Errors with Pentland's and Yokoya's methods

The graph on the upper left-hand side plots the real fractal dimension against the dimension estimated by Pentland's method. The dotted line illustrates the ideal relation. The graph on the upper right-hand side shows the same for Yokoya's method.

The graph on the lower left-hand side illustrates the effect of Gaussian noise on Pentland's method. The solid line represents estimates from data without noise. The dotted, dot-dashed and dashed lines represent estimates from data with noise with $\sigma = 10^{-1}, 10^{-2}$, and 10^{-3} respectively. The line with $\sigma = 10^{-3}$ overlays the line without noise. The lower right-hand side shows the same for Yokoya's method.

Known problems with the Perceptron sensor, such as internal reflections and vignette effects, cause some range image pixels to have incorrect or invalid depth values. We remove these pixels by thresholding specified image regions [11].

2. Transform range image measurements from sensor-centered polar coordinates into sensor-centered Cartesian coordinates.

   Given the polar coordinates $(r.c.p_{r,c})$ of a pixel located in the row r and column c of the range image, we find coordinates $(x.y.z)$ by

$$x = \rho \sin\theta \quad . \quad y = \rho \cos\circ \sin\theta \quad . \quad z = \rho \sin\circ \sin\theta \quad .$$

$$\rho = qp_{r,c} + o \quad . \quad \circ = \frac{r\circ_{fov}}{N_{rows} - 1} + \circ_{start} \quad . \quad \theta = \frac{c\theta_{fov}}{N_{rows} - 1} + \theta_{start} \quad .$$

   where q and o denote the quantization error (meters per grey-level) and standoff distance (m) of the sensor, $\circ_{fov}$ and $\circ_{start}$ denote the horizontal field of view and the horizontal start angle, θ_{fov} and θ_{start} denote the vertical field of view and the vertical start angle, and N_{rows} and N_{cols} denote the numbers of rows and columns in the range image.

3. Compute statistics of $|z_{x,y} - z_{x+dx,y+dy}|$.

   In the sensor frame, consider two points on the xy plane: $(x.y)$ and $(x + dx.y + dy)$. The Euclidean distance between them is $\triangle d = \sqrt{dx^2 + dy^2}$. We are interested in statistical variations in the absolute value of the difference in elevation between these two points: $\triangle z_{\triangle d} = |z_{x,y} - z_{x+dx,y+dy}|$. Pentland's method requires the standard deviation of the distribution of elevation differences; Yokoya's method requires the expected value (first moment) of the distribution of elevation differences.

   Because the data points are distributed irregularly on the xy plane in the sensor frame, we must extend the original methods, which assume the data is distributed regularly (i.e., that the sampling interval is constant). For $i = 0.1.\ldots.m$, and $\triangle d_k < \triangle d_{k+1}$, we prepare counters A_i, B_i and C_i to correspond to distance $\triangle d_i$. Let ξ be a small distance that satisfies $0 < \xi < \triangle d_i$, for any i. This parameter represents the width of a circular permissible area including a circle of radius $\triangle d_j$ (Figure 6). Suppose there is a data point at $(x + dx.y + dy)$ with elevation z'. If $|\triangle d_j - \triangle d|$ is less than ξ, then the point lies in the permissible area, and we update the counters as follows:

$$A_j \leftarrow A_j + |z - z'| \quad . \quad B_j \leftarrow A_j + (z - z')^2 \quad . \quad C_j \leftarrow C_j + 1 \quad .$$

   After considering all pairs of data points, we ensure that C_i is larger than a threshold of number of pairs. If C_i is small, then we question whether the number of samples was sufficient to compute reliable statistics, and discard this data. Otherwise, we compute the sample standard deviation for Pentland's method by

$$S_{\triangle d_i} = S[|z_{x,y} - z_{x+dx,y+dy}|] = \sqrt{\frac{B_i - (A_i^2/C_i)}{C_i - 1}} \quad .$$

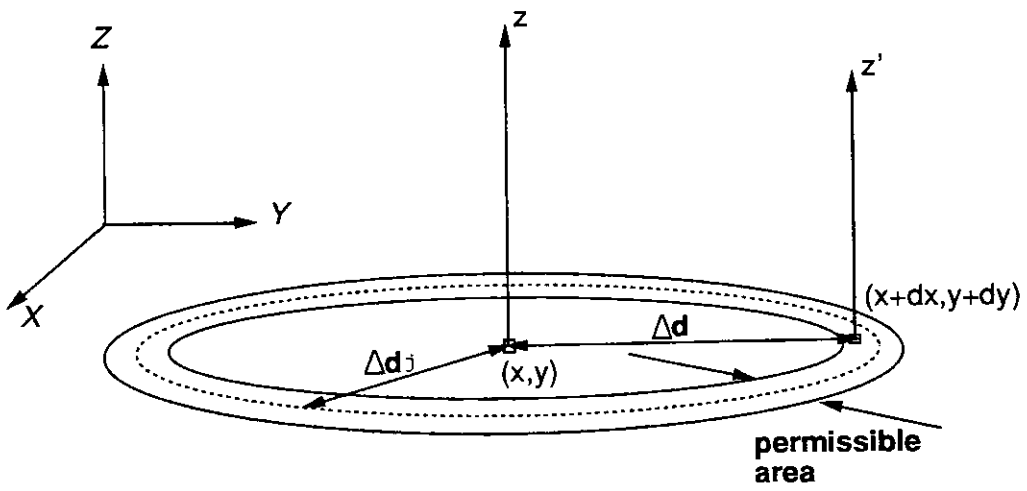


Figure 6: Accommodating irregular sampling intervals

Because of irregular sampling, we cannot detect a pair of points located a certain distance Δd_j away on the xy plane. As a permissible area, we set a circle whose width is 2ξ around a point (x, y) with z. If the point $(x + dx, y + dy)$ exists in the area, its elevation z' is used to compute A_j and B_j.

and the sample mean for Yokoya's method by

$$E_{\Delta d_i} = E[|z_{x,y} - z_{x+dx,y+dy}|] = A_i/C_i$$

4. Plot the points in log-log space and identify linear segments.

   For Pentland's method, the point coordinates are $(\log \Delta d_i, \log S_{\Delta d_i})$. For Yokoya's method, the point coordinates are $(\log \Delta d_i, \log E_{\Delta d_i})$.

   Because most natural patterns exhibit self-similarity only over certain scales, and not over all scales, it is necessary to segment sets of points that are linear. We investigated two approaches to this segmentation problem.

   The first approach is polyline fitting using the minimax method, as proposed by Kurozumi [10]. In the field of document image processing, this technique is frequently used to detect line segments. The technique segments the given points into several sets of points which distribute within narrow rectangles, i.e., nearly along lines. The width of the rectangle must be specified as a threshold. The cardinality of the sets is a natural criterion for determining which should be used to estimate the fractal dimension. However, the cardinality is fairly sensitive to the rectangle width, thus making it difficult to select the proper threshold for this segmentation technique.

   The second approach employs iterative least-square line-fitting (Appendix A). Using this technique, we can construct a set of points that lie within a specified distance of the line. This technique is not sensitive to changes of the threshold. However, the technique selects only the first linear part satisfying the criterion. In the case where several fractal patterns

exist, each with a different fractal dimension, multiple linear segments will appear in the log-log plot. Simply selecting the first will preclude consideration of the others. In these experiments, we use one or the other method; a future topic of research is to combine them to identify all of the fractal patterns.

5. Estimate fractal dimension from the slope of linear segments.

   When the points lie on a line in the log-log space, we can estimate the fractal dimension of the pattern by the difference between the Euclidean dimension of the pattern and the slope of the line formed by the points.

Experimental Results

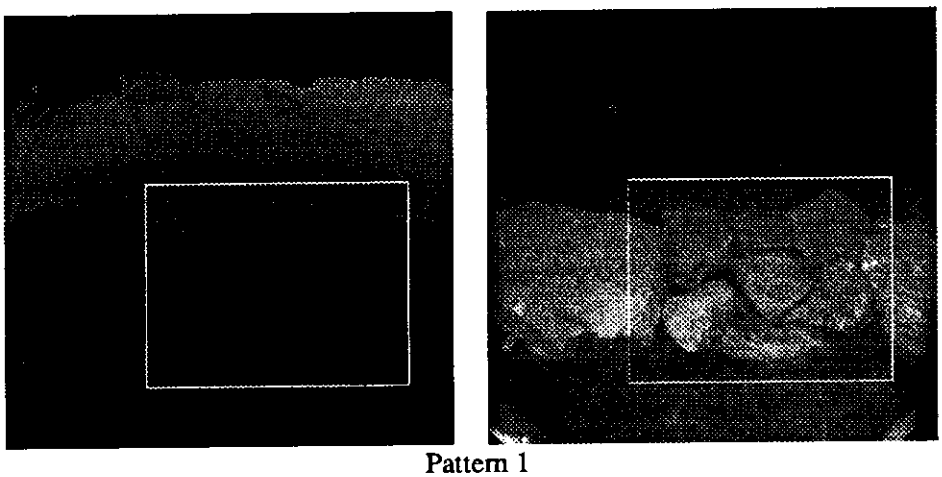
We selected eight patterns from range images acquired with the Perceptron rangefinder. Figures 7, 8 and 9 illustrate the images, which are presented in order of increasing roughness, as determined subjectively by the authors. We estimated the fractal dimension of the regions indicated by white rectangles.

Before applying the procedures to estimate fractal dimension, we checked whether the data acquired by the Perceptron satisfies the conditions on fractal Brownian functions stated in (4). Figure 10 histograms $z_{x,y} - z_{x+dx,y+dy}$ for Pattern 2, with Δd =0.4, 0.6 and 0.8 m. From the figure it is clear that the condition in (4), that $g(x)$ be a cumulative distribution function, is satisfied. We conclude that the patterns are fractal Brownian functions.

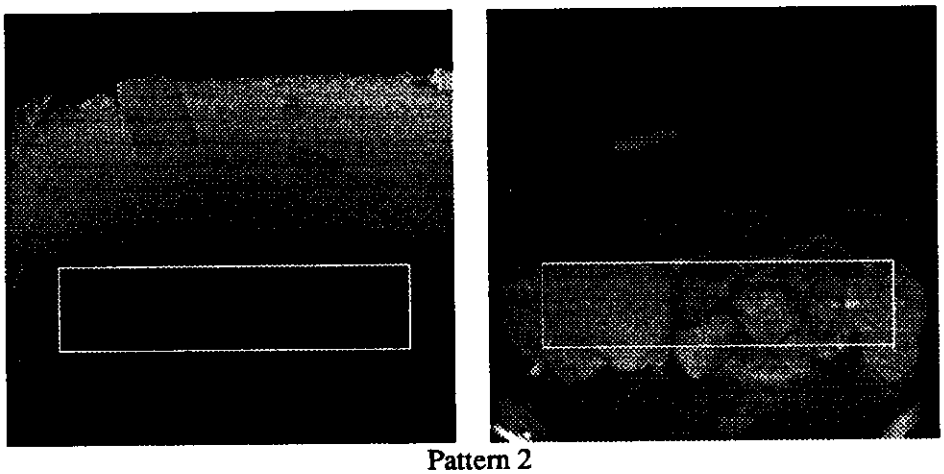
A further condition on the distributions, imposed by Yokoya's method, is that they are normal. We conducted χ^2 tests for Gaussianity and observed negative results, i.e., that the probabilities of the data being normally distributed were quite low. This suggests that it is not probable that the points were created by a fractional Brownian motion process (which is a special case of a fractal Brownian function). However, to the extent that the distributions are symmetric, and exhibit a central tendency, there is some justification in proceeding to apply Yokoya's method, despite the negative χ^2 test results.

Figure 11 shows the result of applying Pentland's method: the points $(\log \Delta d, \log S_{\Delta d})$ segmented by iterative least-square fitting. The points distribute linearly, therefore, we observe self-similarity in all of these natural terrain patterns. Figure 12 shows the result of applying Yokoya's method to the range image regions: the points $(\log \Delta d, \log E_{\Delta d})$. The results of the two methods are similar, as they were with synthesized data. On some patterns, plotted points distribute in several sets of linear parts. If we intend to estimate all fractal dimension in such distribution, segmentation by polyline fitting is useful. However, we must set a parameter of allowable error for fitting, because segmented results are very sensitive to the parameter.

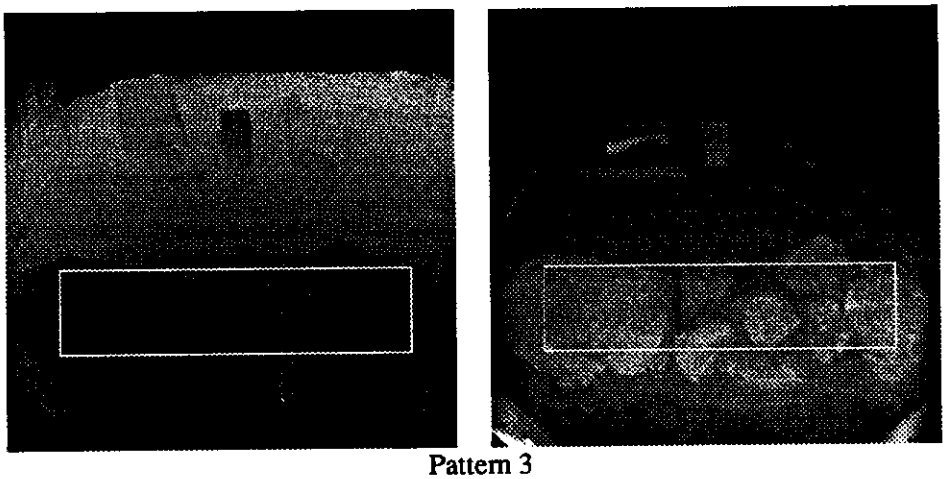
Table 1 lists the fractal dimensions estimated by both methods, with both segmentation techniques. Some of the results for segmentation by polyline fitting required carefully selecting the amount of allowable error. We also illustrate fitting error normalized with $\log E(\Delta z_{\Delta d})$ or $\log S(\Delta z_{\Delta d})$. All errors are small enough to determine that the patterns are fractal. Moreover, the rows are ordered by roughness, as perceived by the authors. The order of estimated fractal dimension correlate strongly to the intuitive order (the last three patterns—sandy flat floors—are



Pattern 1



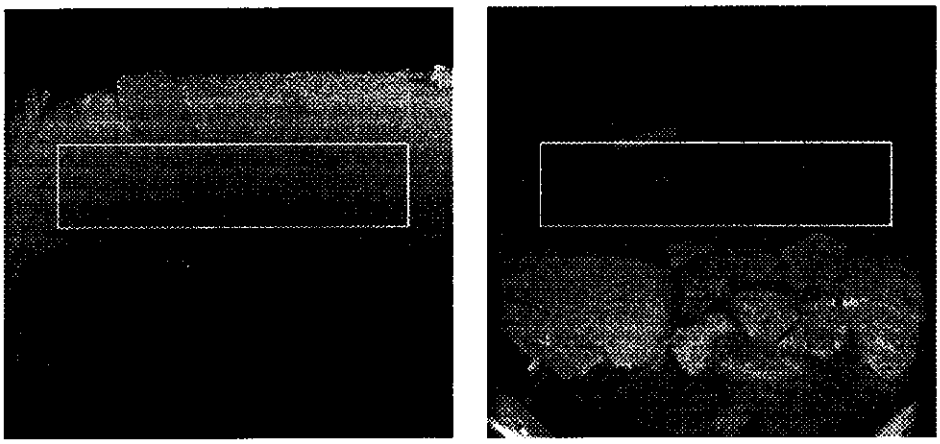
Pattern 2



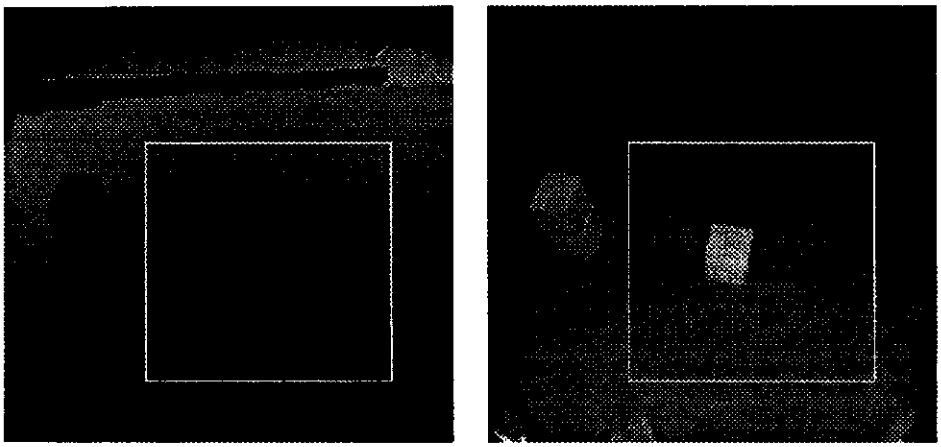
Pattern 3

Figure 7: Rangefinder images (1)

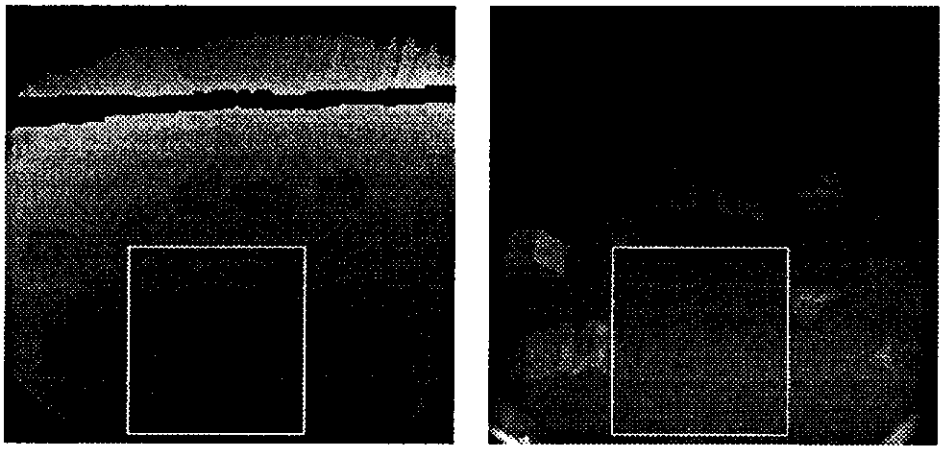
The figure illustrates Perceptron image pairs: processed range (left) and raw reflectance. Only the range images are used to estimate fractal dimension.



Pattern 4

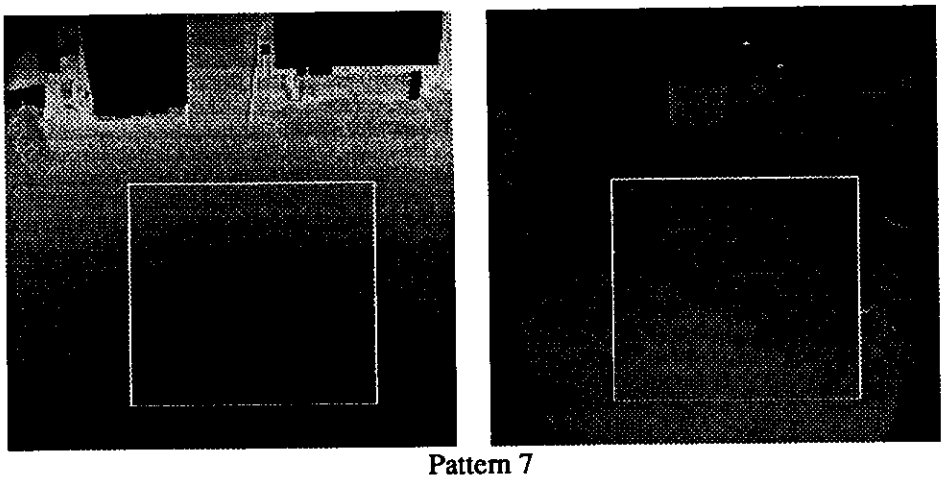


Pattern 5

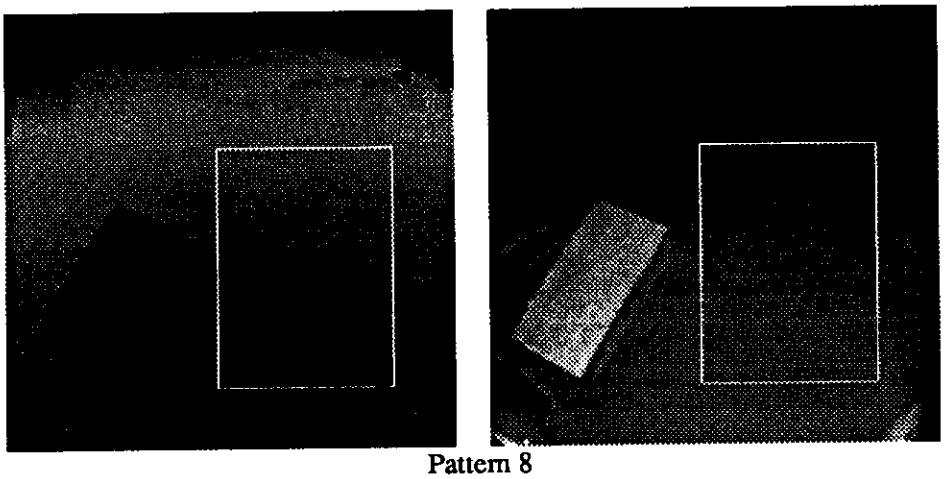


Pattern 6

Figure 8: Rangefinder images (2)

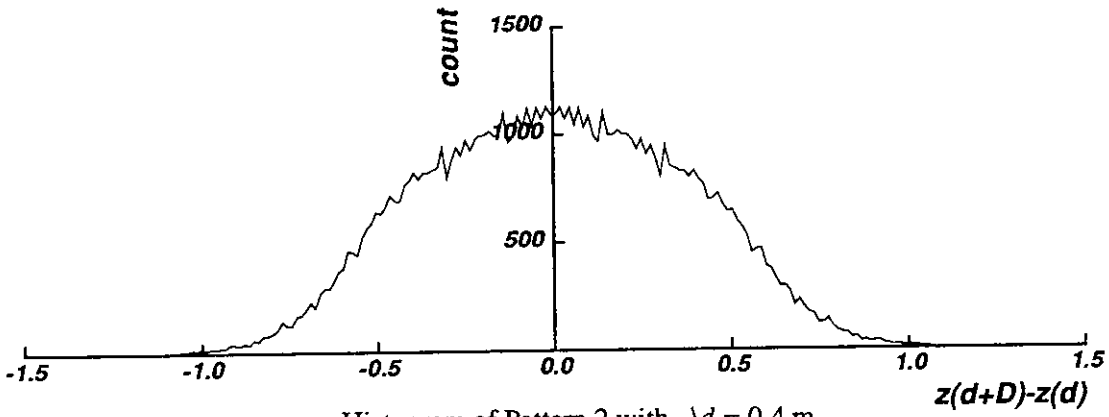


Pattern 7

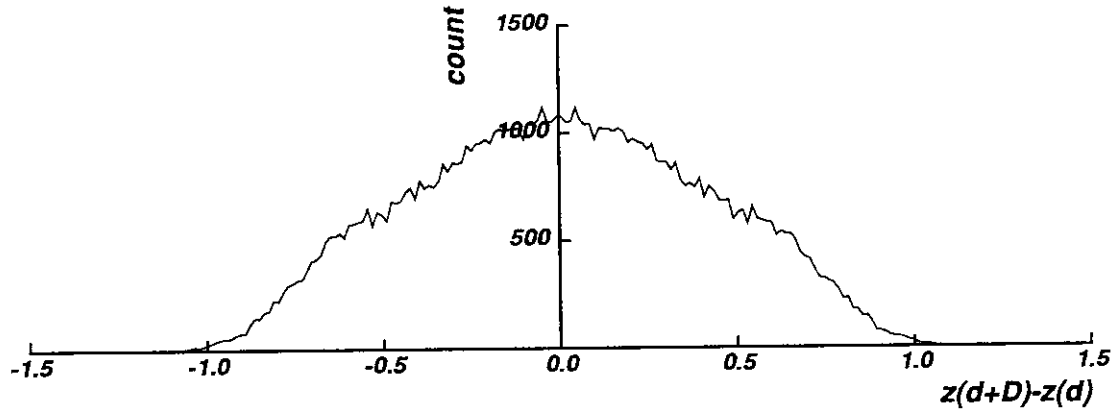


Pattern 8

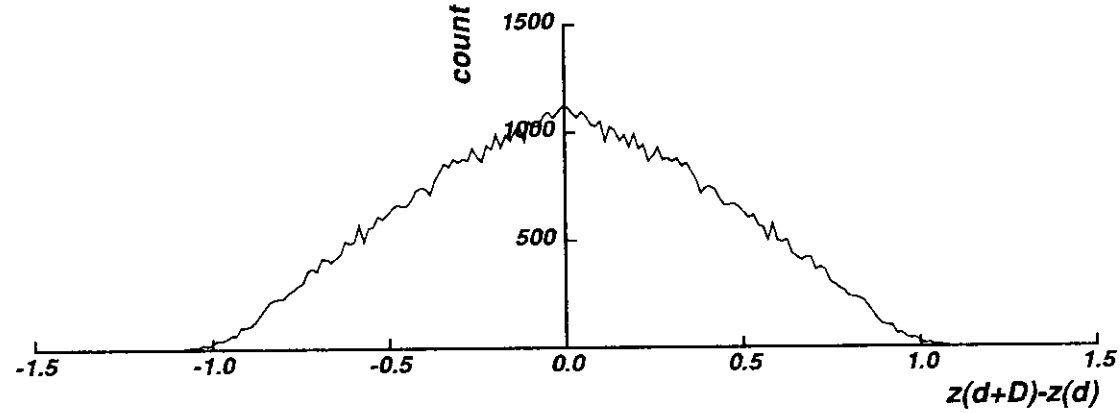
Figure 9: Rangefinder images (3)



Histogram of Pattern 2 with $\Delta d = 0.4$ m



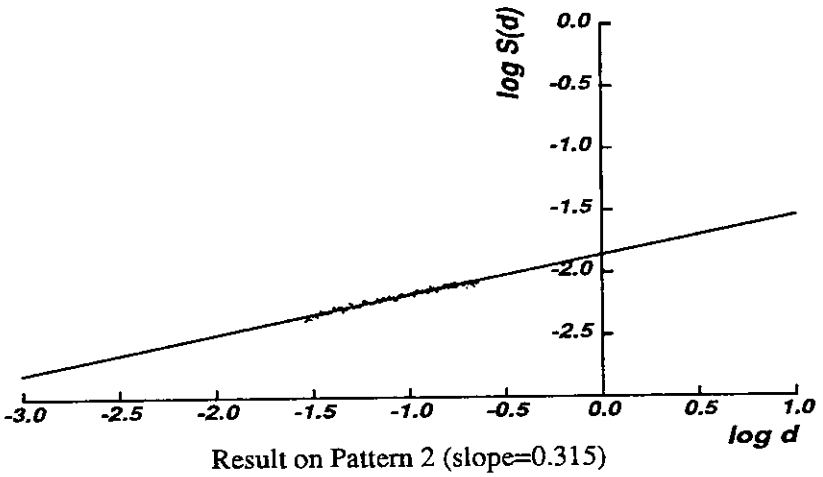
Histogram of Pattern 2 with $\Delta d = 0.6$ m



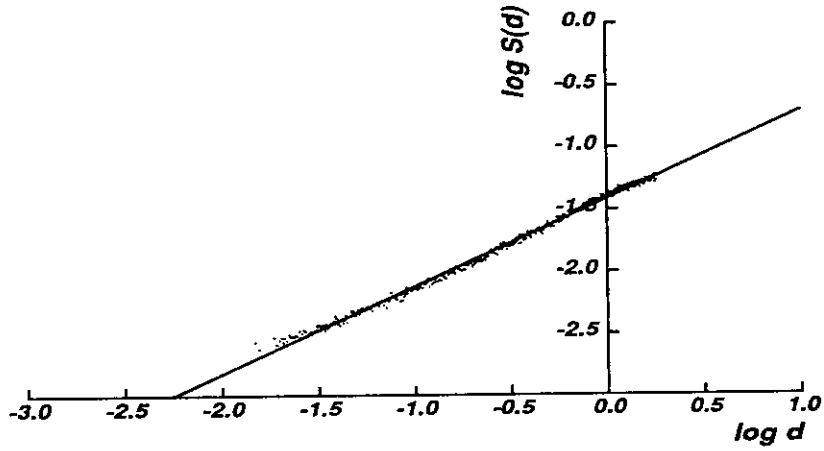
Histogram of Pattern 2 with $\Delta d = 0.8$ m

Figure 10: Empirical distribution functions

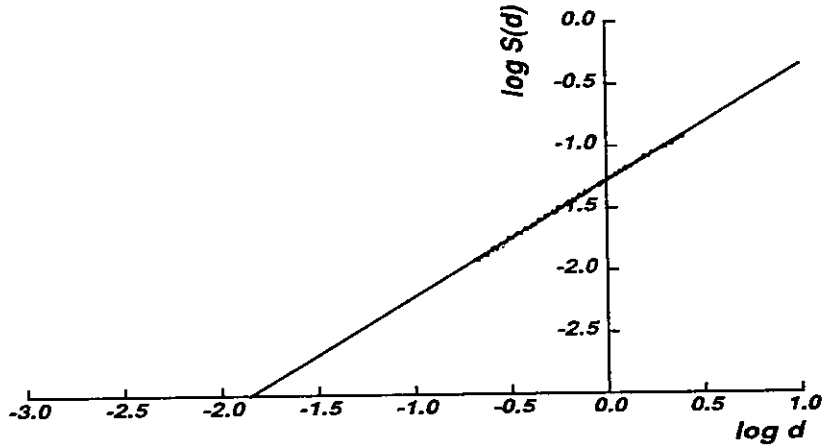
These graphs illustrate histograms of $z_{x,y} - z_{x+dx,y+dy}$ with $\Delta d = \sqrt{dx^2 + dy^2}$ on Pattern 2.



Result on Pattern 2 (slope=0.315)



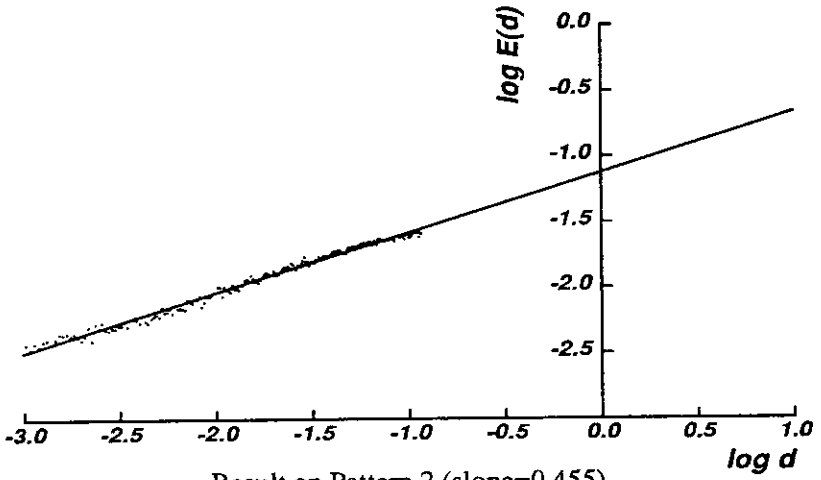
Result on Pattern 4 (slope=0.705)



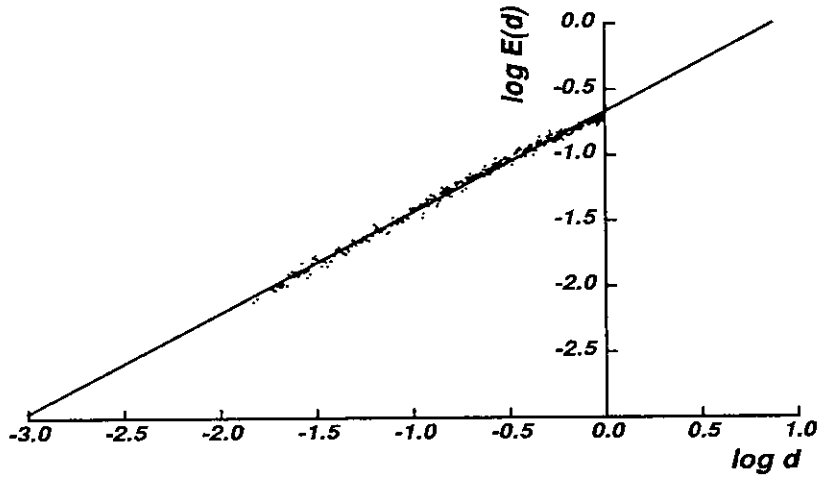
Result on Pattern 6 (slope=0.930)

Figure 11: Results with Pentland's method on Perceptron images

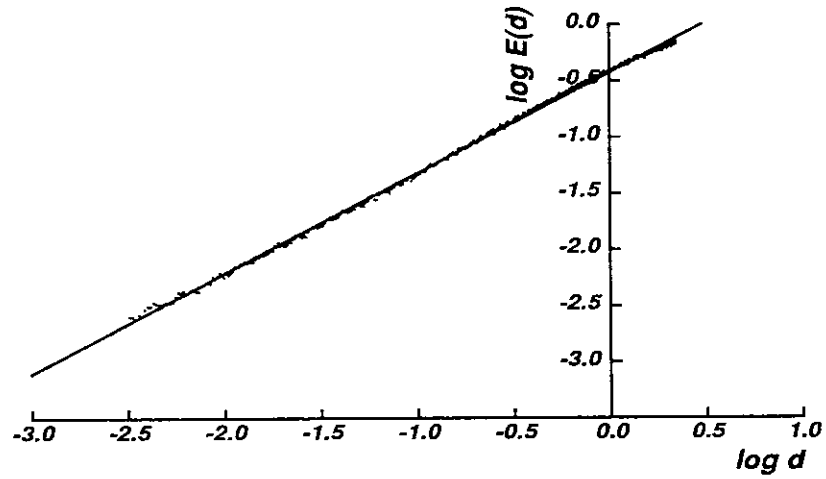
These graphs illustrate the segmented points and the lines fitted to them. The label ($\log d, \log S(d)$) corresponds to ($\log \Delta d, \log S_{\Delta d}$) in the text.



Result on Pattern 2 (slope=0.455)



Result on Pattern 4 (slope=0.766)



Result on Pattern 6 (slope=0.889)

Figure 12: Results with Yokoya's method on Perceptron images

These graphs illustrate the segmented points and the lines fitted to them. The label ($\log d . \log E(d)$) corresponds to ($\log \triangle d . \log E_{\triangle d}$) in the text.

Pattern	D (error) Pentland, LSF	D (error) Pentland, polyline	D (error) Yokoya, LSF	D (error) Yokoya, polyline
1	2.793 (0.011)	2.754 (0.013)	2.691 (0.012)	2.661 (0.025)
2	2.685 (0.012)	2.703 (0.016)	2.545 (0.021)	2.512 (0.016)
3	2.621 (0.014)	2.657 (0.018)	2.498 (0.016)	2.486 (0.014)
4	2.295 (0.020)	2.304 (0.021)	2.234 (0.024)	2.239 (0.025)
5	2.198 (0.009)	2.206 (0.008)	2.210 (0.009)	2.196 (0.010)
6	2.070 (0.008)	2.100 (0.018)	2.111 (0.018)	2.118 (0.021)
7	2.090 (0.009)	2.114 (0.006)	2.090 (0.009)	2.102 (0.014)
8	2.101 (0.019)	2.126 (0.012)	2.038 (0.018)	2.010 (0.022)

Table 1: Fractal dimensions estimated by Pentland's and Yokoya's methods

almost identical). These results suggest that the fractal dimension estimated using these methods can be utilized as a measure of roughness of natural terrain.

The computational complexity of the method utilized here is $O(N^2)$, where N is the number of pixels, because it is necessary to calculate distances between all pairs of pixels in order to determine which pairs lie in the permissible area. On a Sun4/40 with 24 MB of physical memory, estimating the fractal dimension for Pattern 2 (10000 pixels) requires 1.2×10^3 s, and Pattern 5 (19600 pixels) requires 5.5×10^3 s.

4 Discussion

Roughness is an ambiguous property. Geometrically, the Hausdorff-Besicovich dimension is a measure of elasticity, which is one of the main factors contributing to the notion of roughness [4]. The Hausdorff-Besicovich dimension is defined on general geometrical shapes, but it is equal to the fractal dimension only on fractal shapes. Natural terrain typically has a fractal shape (over some range of scales). Therefore, its fractal dimension can be used to express its roughness properly.

In this report, we investigated two published approaches to the problem of estimating fractal dimension: the box-counting approach, and the fractal Brownian function approach. We implemented the published algorithms and applied them to synthetic images and real range images.

In the experiments with synthetic images, we found the fractal Brownian function methods proposed by Pentland and Yokoya to be superior to the box-counting approaches described by Voss. One reason is that the box-counting approaches require data sampled at equal intervals, while the fractal Brownian function approaches do not. Another reason is that the box-counting methods are significantly less robust to data corrupted by Gaussian noise (which has been widely used as a model of sensor noise).

For the experiments with real images, we extended the fractal Brownian function methods to accommodate irregularly sampled data supplied by a scanning laser rangefinder. The extension involves considering neighbors that lie approximately, but not exactly, within a given distance from a data point. This requires searching a permissible area within the given distance. The extension also involves analysis of the distribution of points in a log-log space in order to determine which points to analyze further and which points to ignore (most real patterns are fractal only over some range of scales, not at all scales). We conduct this analysis by applying two segmentation methods—polyline fitting and iterative least-square fitting—to the points in log-log space.

We applied the extended methods to noisy range imagery of natural terrain (sand and rocks) acquired with the Perceptron scanning laser rangefinder. The resulting estimates of fractal dimension correlate closely to the human perception of the roughness of the terrain. We conclude that it is reasonable to model natural terrain as a fractal pattern, and that the fractal dimension is a reasonable measure of roughness of terrain.

Remaining problems in this work that we do not intend to pursue immeniately include determining the region in which to conduct fractal analysis, identifying which linear parts of the log-log curves are most significant, and segmenting multi-fractal patterns. We may need further simulations using different noise models (e.g., non-Gaussian additive noise, quantization and truncation) in order to study sensitivity of methods estimate fractal dimension and to determine how appropriate the estimation is. In the future work, we will explore further the utility and limits of fractal dimension as a descriptor of natural terrain, concentrating on the problem of fractal interpolation.

A Iterative Least-Square Line Fitting

This appendix describes the method for iterative least-square line fitting for input data from Pentland's method. To apply it to data from Yokoya's method, simply exchange $E_{\Delta d}$ for $S_{\Delta d}$.

Let K represent the number of points fitted, and N represent the number of shifted points. The variable e_{th} is a threshold on the initial fitting error, and r_{th} is a threshold on the ratio of current to initial fitting error.

1. $s \leftarrow 0$
2. Fit a line to the input points $(\log \Delta d_i, \log S_{\Delta d_i})$, for $i = s, \ldots, s+K$, using the method of least-squares. If the fitting error e_{ini} is less than a threshold e_{th}, go to 4.
3. $s \leftarrow s+N$

   If $s > m$, then halt (segmentation fails). Otherwise, go to 2.
4. $t \leftarrow s+K$

   If $t > m$, then output $(s, s+K)$ as the segmented linear part, and halt.
5. $u \leftarrow t+K$

   If $u > m$, then $u \leftarrow m$
6. Compute the average distance e between the fitted line and the points $(\log \Delta d_i, \log S_{\Delta d_i})$, for $i = t, \ldots, u$.

   If e/e_{ini} is less than a threshold r_{th}, then fit a line to the points $(\log \Delta d_i, \log S_{\Delta d_i})$. for $i = s, \ldots, u$ using the method of least-squares, and re-calculate e_{ini}. Otherwise, output (s, u), and halt.
7. if $u = m$, output (s, m), and halt.

   Otherwise, $t \leftarrow u$ and go to 5.

References

[1] P. Burrough. Fractal Dimensions of Landscapes and Other Environmental Data. *Nature*, 294:240–242, 1981.

[2] B. Dubuc. On Evaluating Fractal Dimension. Master's thesis, McGill University, April 1988. Available as Technical Report TR-CIM-88-13.

[3] B. Dubuc, J. Quiniou, C. Roques-Carmes, C. Tricot, and S. Zucker. Evaluating the Fractal Dimension of Profiles. *Phys. Rev. A*, 39:1500–1512, February 1989.

[4] H. Kaneko. Fractal and Its Application to Image Analysis. *Journal of the Institute of Television Engineers of Japan*, 41(4):359–366, 1987. In Japanese.

[5] H. Kaneko. Fractal Feature and Texture Analysis. *Trans. of the IEICE*, J70-D(5), May 1987. In Japanese.

[6] H. Kaneko. Fractal Matrix Model and Its Application to Texture Analysis. *Trans. of the IEICE*, E71(12):1221–1228, December 1988. Special Issue on CAS Karuizawa Workshop. In Japanese.

[7] J. Keller, S. Chen, and R. Crownover. Texture Description and Segmentation through Fractal Geometry. *Computer Vision, Graphics, and Image Processing*, 45(2):150–166, February 1989.

[8] J. Keller, R. Crownover, and R. Chen. Characteristics of Natural Scenes Related to the Fractal Dimension. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-9(5):621–627, September 1987.

[9] P. Kube and A. Pentland. On the Imaging of Fractal Surfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-10(5):704–707, September 1988.

[10] Y. Kurozumi and W. Davis. Polygonal Approximation by the Minimax Method. *Computer Graphics and Image Processing*, 19:248–264, 1982.

[11] I. Kweon, R. Hoffman, and E. Krotkov. Experimental Characterization of the Perceptron Laser Rangefinder. Technical Report CMU-RI-TR-91-1, Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, January 1991.

[12] B. Mandelbrot. *Fractals, Form, Chance, and Dimension*. Freeman, San Francisco, California, 1977.

[13] B. Mandelbrot. *The Fractal Geometry of Nature*. Freeman, San Francisco, California, 1982.

[14] B. Mandelbrot and J. Van Ness. Fractional Brownian Motion, Fractional Noises and Applications. *SIAM Review*, 10(4):422–437, October 1986.

[15] P. Maragos and F. Sun. Measuring Fractal Dimension: Morphological Estimates and Iterative Optimization. Technical Report CICS-P-198, Center For Intelligent Control Systems, Cambridge, Massachusetts, March 1990.

[16] G. Medioni and Y. Yasumoto. A Note on Using the Fractal Dimension for Segmentation. In *Proc. IEEE Computer Vision Workshop*, pages 25–30, Annapolis, Maryland, 1984.

[17] S. Peleg, J. Naor, R. Hartley, and D. Avnir. Multiple Resolution Texture Analysis and Classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(4):518–523, July 1984.

[18] A. Pentland. Fractal-Based Description of Natural Scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6):661–674, 1984.

[19] D. Saupe. Algorithms for Random Fractals. In H.-O. Peitgen and D. Saupe, editors, *The Science of Fractal Images*. Springer-Verlag, 1988.

[20] R. Voss. Random Fractal Forgeries. In R. Earnshaw, editor, *Fundamental Algorithms for Computer Graphics*, pages 805–835, Berlin, 1985. Springer-Verlag.

[21] R. Voss. Fractals in Nature: Characterization, Measurement, and Simulation. In H.-O. Peitgen and D. Saupe, editors, *The Science of Fractal Images*. Springer-Verlag, 1988.

[22] N. Yokoya, K. Yamamoto, and N. Funakubo. Fractal-Based Analysis and Interpolation of 3D Natural Surface Shapes and Their Application to Terrain Modeling. Technical Report McRCIM-TR-CIM 87-9, McGill University, Montreal, Quebec, July 1987.