Initial Value Interpolation of Sensor Data from Natural Surfaces

Dr. Andrew W. Harrell and Dr John Peters
Engineering Research and Development Center (ERDC)
Vicksburg, MS 39180

Abstract

This paper discusses ways to interpolate data from information on sensor surfaces using 2 dimensional fractal generation programs. MATLAB programs were written to plot two dimensional Brownian motion surfaces from their Hurst exponents using inverse Fourier transforms. Histograms of the power spectrum of the original data and the data from the interpolated surface were compared. The effects of using different types of Gaussian noise functions and a white noise function in the subroutine that generates the Brownian motion data were investigated. Programs in MATLAB to do mid-point interpolation algorithms were written and the results compared, in terms of the histograms of the power spectrums, and the inverse Fourier transform. Also, programs that use multi-fractals to do the interpolation were written and the effect of characterizing the fractal with a lacunarity parameter was looked at. The results were an improved (that is, over previous methods) goodness of fit of interpolated surface with the original sensor data

Introduction and Purpose

Simulation of the natural environment is an important component of analysis systems for remote sensing. An important element of these simulations is capturing the character of surfaces accurately. Not only is the surface representation important for achieving realistic visual appearance but is also critical to the response of many thermal and radar based systems. For visual and thermal systems, the radiance emitted to the sensor depends on the direction of surface normals. Surfaces that are too smooth can over predict the spectral component of radiation received by the sensor. The degree of scatter of electromagnetic energy in radar systems depends intimately on the roughness of the surface.

The most straightforward approach to creating a natural surface is to sample actual surfaces in the type of terrain to be simulated. Current technology enables high resolution scans covering several square meters of area at resolutions on the order of millimeters. These surfaces are described as point values which can be triangulated to create surface meshes. These surface meshes can be incorporated into finite element meshes for simulation of thermal or electromagnetic response. Limitations in computing capacity often require a coarsening of the surface representation. In many cases such a coarsening is of little practical consequence due to limitations in the resolution of the sensing system. However, the scheme for surface representation requires a multi-resolution description to give the best representation for a given level of resolution.

Visualization of natural surfaces can be complicated by incomplete imaging data. Often areas of the surface might be obscured of obstacles in the measured scene that are not to appear in the simulated scene. Removing those objects leaves gaps in the surface data. It is possible with available techniques to create a random field that matches the texture of the natural surface, but to serve as a patch, it is necessary that the random field is conditioned such that the values on the edges of the patch match those of the measured data. For example, Figure 1 is a surface created from a laser scan taken in an unpaved road. The surface is to be incorporated into simulation where the vegetation is to be placed randomly within the scene. The smooth flat areas correspond to regions where grass clumps were growing in the road. The laser scan of the grass clumps was not accurate and therefore not indicative of either the natural ground surface or the grass. The problem is to reconstruct the surface in the areas such that the texture of the natural surface is reproduced while maintaining continuity with the rest of the surface.

The problem with reconstructing a surface patch has two components. First the boundaries of the patch must match. While a scheme could be devised based on krieging (see eg Cressie [1993]), that honors all known data, the resulting surfaces lack the characteristics of the natural surface. Thus, the second criterion is that the reconstructed surface should match the statistical character of the measured surface. The second criterion requires determining the fractal character of the sampled surface and imparting those characteristics to the interpolated patch. In this paper both problem are addressed. It is shown that the fractal character of the natural surface is matched well by fractal Brownian motion statistics, which are parameterized by the fractal dimension (lacunarity) and the Hurst exponent. The matching of the boundary condition for the patch is achieved using a procedure by Pietgen and Saupe [1988]. The effectiveness of the technique is demonstrated by an example.

Scope

This paper explains the computer programs required to average the data and plot it in 3 dimensions. It discusses the mathematical methodology required to compute 2-dimensional multifractal and fractal interpolated surfaces which honor initial data on the boundary and in the interior of sensor blank areas of the measured natural data. It does not explain the mathematics and computer programs required to estimate the multifractal dimensions and lacunarities which serve as parameters to generate the interpolated sensor data surfaces. The references at the end of the paper contain chapters which explain fractal and multifractals and give information on various methods that use wavelets and other mathematical techniques to estimate these parameters. Also, we do not attempt to match up 3-D plots of the resulting interpolated surfaces, but visual comparisons of histograms of data point values and Fourier transforms of the consistency of the interpolated data at various scales and resolutions are used for this.

Data Gathered from Experiments

Shown below is some example data (Figure 1), which was measured by a scanning laser, consists of 130870 lines of x,y,z coordinates (the z coordinate being the scanned pixel intensity). Often, a portion of the sensor data returned from the scan will be missing. Because the laser scans the terrain intermittently and at a varying angle, the x,y,z coordinates are not sorted in simultaneously increasing values. This complicates the process of interpolating the data in missing areas of the terrain in order to plot, display, and interpret it. A 3-D image for Figure 1 can be created from 2-D contour plots. But, it is possible that the mathematical process for this may lose 3-D natural texture information regarding its fractal dimension. Figure 2 shows the resulting 3-D surface image created by a Unix surface plotting program.Figure 3 shows the resulting 3-D surface image created by the MATLAB plotting software written for this project..

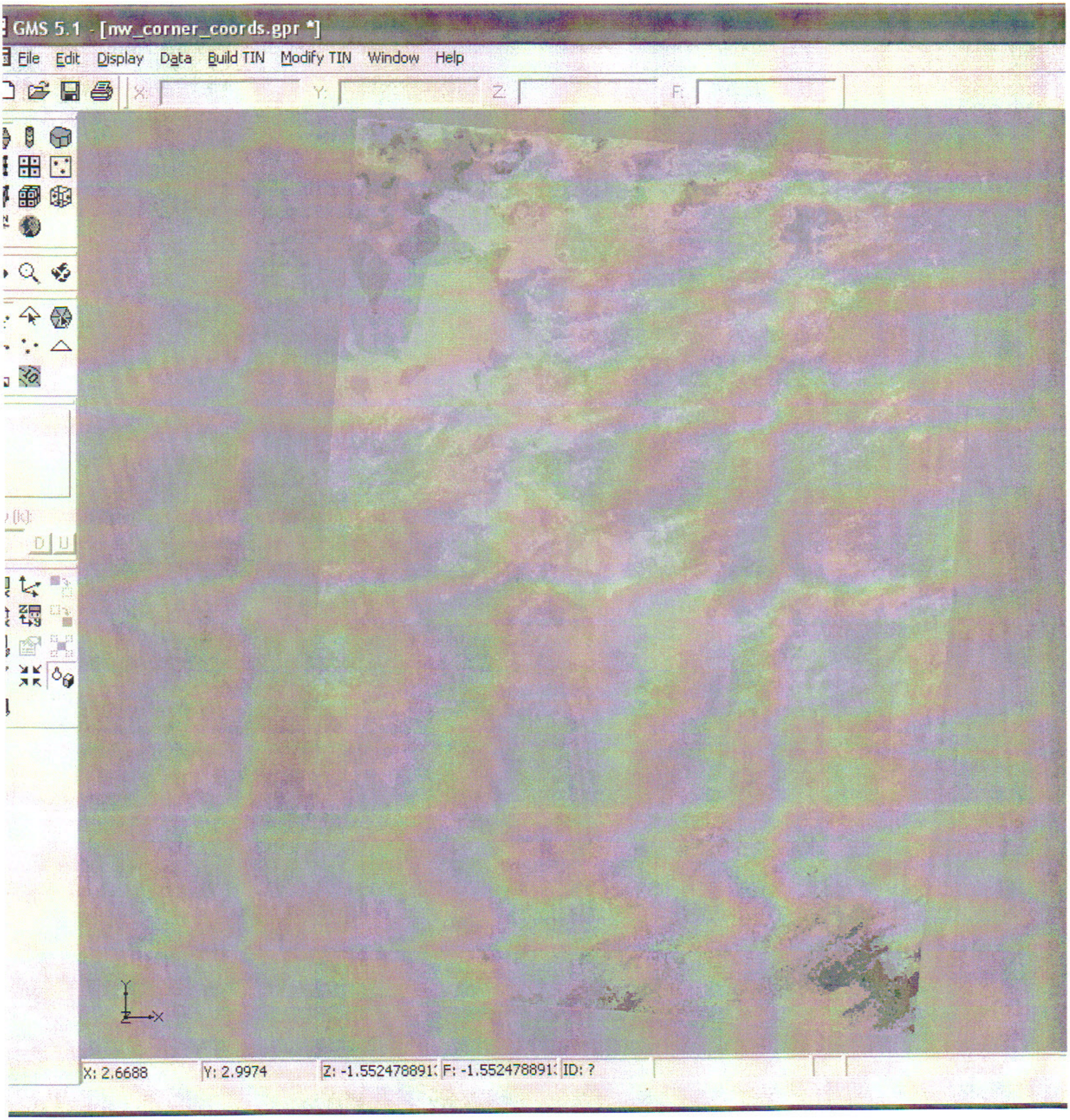

Figure 1 Example Surface Data from Laser Scan of an Unpaved Road.

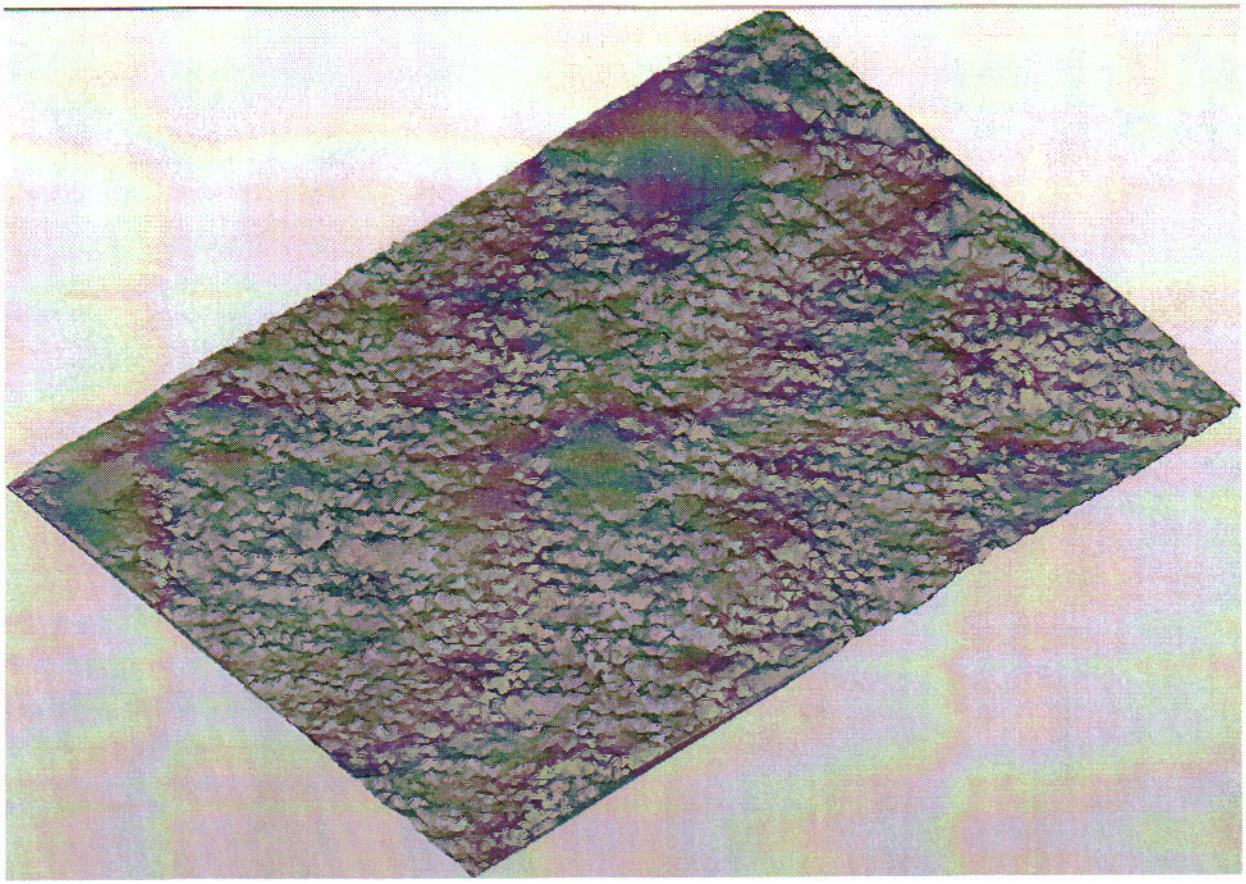

Figure 2. Surface created from laser scan of an unpaved road. Smooth areas correspond to areas covered by grass clumps that gave inaccurate surface representation.

Standard mathematical interpolation formulas in two dimensions assume the x,y coordinates are sorted in x and y values and equally spaced, or at least sorted in x and y values. This is because the interpolation formulas for two dimensions is the tensor product (coordinate by coordinate product) of two formulas for 1 dimension (one for the x interpolation and one for the y).

The sampling program in Appendix A first extracts a certain percent of the rows to plot. It then sorts the data separately, two times, in terms of x and y values. A uniform grid is computed to the resolution that the spacing of the data and the number of points allows. The x,y coordinate values of this grid will be sorted in both x and y and thus allow ploting in MATLAB (the MATLAB griddata function is used for this).

For the MATLAB code, see Appendix A, plot22int.m and sample32D.m.

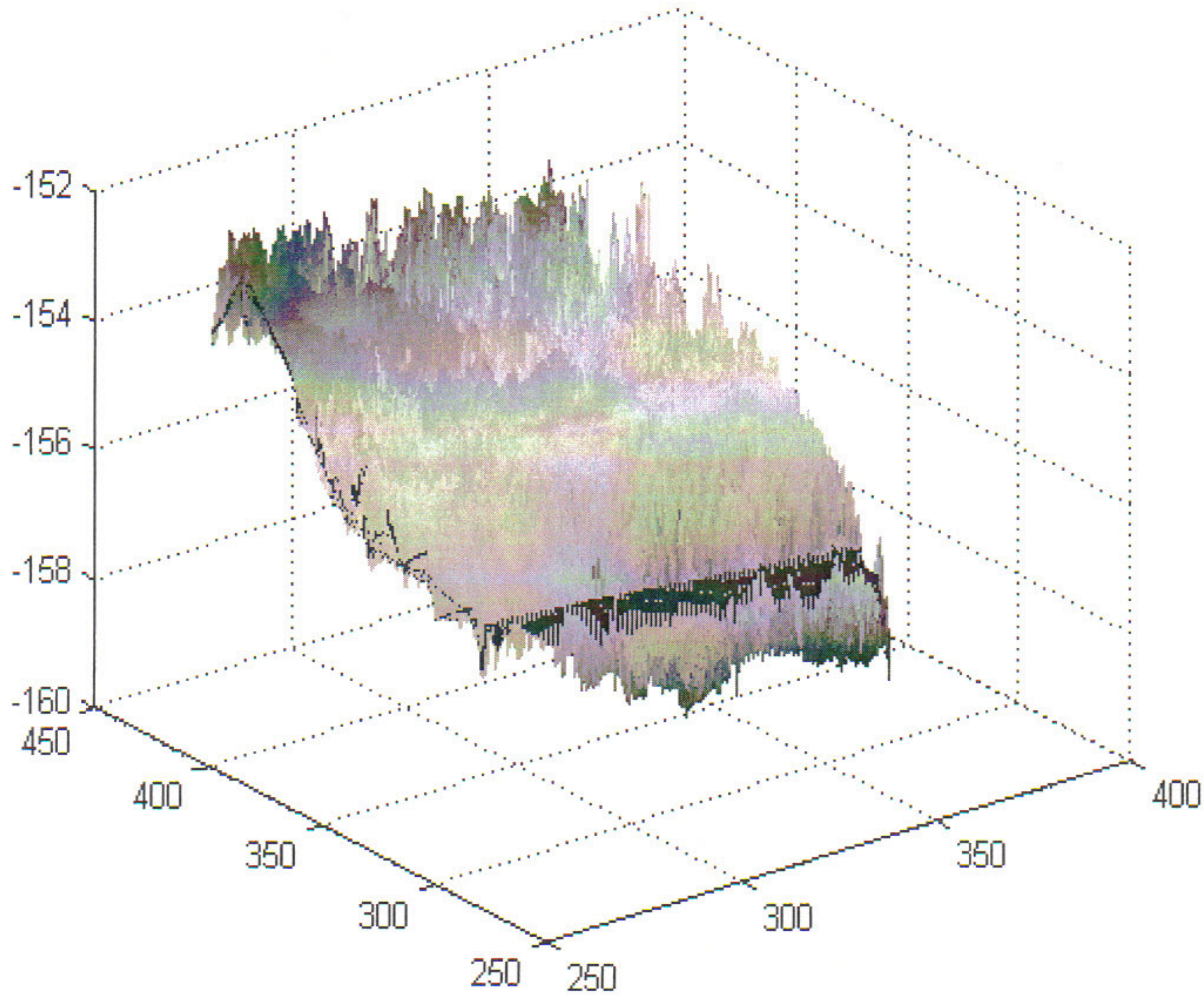

Figure 3 Data from the laser scans in figures 1,2. Coordinate values are multiplied by 100 and plotted in colors by the MATLAB griddata function, black areas highlight missing data.

Method of Mathematical Analysis

Once we have determined how to plot the raw unsorted data it is now possible to compare the accuracy of fit, visually, to different
simulated terrain surfaces of different fractal dimensions and lacunarity. Fractal surfaces can be generated and fitted to given boundary values in various ways. The book by Pietgen and Saupe [1988] contains various algorithms and computer generated visual images of these surfaces. Some of these algorithms use fractional Brownian motion statistical processes by taking the vector product of two statistically generated one-dimensional time series of this type.

The idea of a fractal was defined by Dr. Benoit Mandelbroit in Chapter 3 of his book, The Fractal Geometry of Nature [1977 page 14]. "Mathematicians recognized during their 1875-1925 crisis that a proper understanding of irregularity or fragmentation (as of regularity and connectedness) cannot be satisfied with defining dimension as a number of coordinates." He then goes on to define a fractal. "A **fractal** is by definition a set for which the **Hausdorff Besicovitch dimension** strictly exceeds the **topological dimension**." But, this statement needs further explanation if one doesn't already understand what the Hausdorff Besicovitch dimension and the topological dimension are. The book by David Ruelle [1989, Appendix A] contains explanations and definitions of the above two terms, **topological dimension** and **Hausdorff** dimension (see the references, Harte 1991 and Falconer 1989 for a fuller discussion of the mathematics involved). "A set **E** acquires a topology i.e, becomes a topological space, if a set of subsets of **E,** called open subsets, has been chosen. It is required that **E** and the empty set be open, and that a finite intersection and arbitrary union of open sets be open."... "For each finite covering (X_α) of the topological space **E** by open sets suppose that we can find a new covering (Y_α), with open $Y_\alpha \subset X_\alpha$, such that every intersection of n + 2 sets Y_α is empty. We write then dim **E** <= n. If we have dim **E** <= n but not dim **E** <= n - 1, we say that **E** has **topological dimension n**." "A topological space is **compact** if, for
each every covering (X_α) by open sets, there is a finite subfamily $(X_{\alpha i})$ that is still a covering of **E**." "A **metric** on a set **E** is defined by a distance function d: $ExE \to R$ that satisfies $d(x,x)=0, d(x,y)=d(y,x)\succ 0$ if $x \neq y$, and $d(x,y)+d(y,z)\geq d(x,z)$" Given a **compact metric** space $E \neq 0$, and a real number r > 0, we denote by $\sigma=(\sigma_k)$ a countable covering of E by sets σ_k with diameters $d_k = diam\sigma_k \leq r$. For every $\alpha \succ 0$ we write $m_r^\alpha(E)=\inf_\sigma \sum_k (d_k)^\alpha$. When r decreases to 0, $m_r^\alpha(E)$ increases to a (possibly infinite) limit $m^\alpha(E)$ called the Hausdorff measure of E in dimension α. We write $\dim_H E = \sup\{\alpha : m^\alpha(E)\succ 0\}$. We call this quantity the **Hausdorff dimension** or the **Hausdorff Besicovitch dimension** of **E**. The book by Falconer[1990] gives several others ways to define this dimension. He also shows there that this dimension is not always the same as the more frequently used and easier calculated box-counting dimension (cf. Pietgen, Jurgen, Saupe [1992] for the definition and examples of how to calculate this dimension). Detailed mathematical theorems about inequalities satisfied by the first two types of dimensions and the conditions that need to be satisfied for them to be equal can also be found in the reference by Falconer [1990].

A one dimensional fractional Brownian motion is a continuous time Gaussian[1] process depending on one parameter 0< H <1 (called the Hurst parameter). The variance (square of the root mean squared standard deviation) is given by:
variance(fracBrnMtn(t)-fracBrnMtn(s)) = constant * | t –s | ^ 2H.

he copyright MATLAB program in their Wavelab toolbox uses a random noise generator to create a vector of the required length, then takes its fast fourier transform, raise each coordinate value to the power of the Hurst exponent that is given, and finally computes the inverse fast fourier transform of this spectral density function. Mathematical theorems and the details of the proofs of the properties of the statistical time series associated with this procedure can be found in the book on Wavelets by Mallat [1998] An example of the results of the calculations for one-dimensional profiles is shown in figures 4 and 5 below.

[1] That is, whose probability distribution is of the Gaussian form.

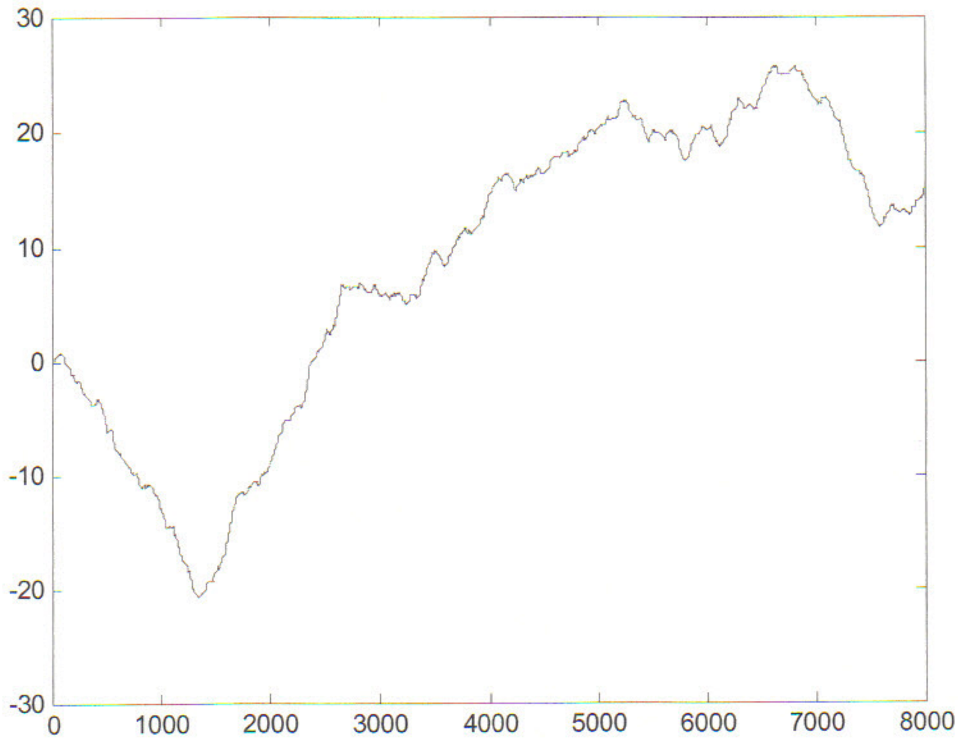

Figure 4   Fractional Brownian Noise, Hurst exponent .9

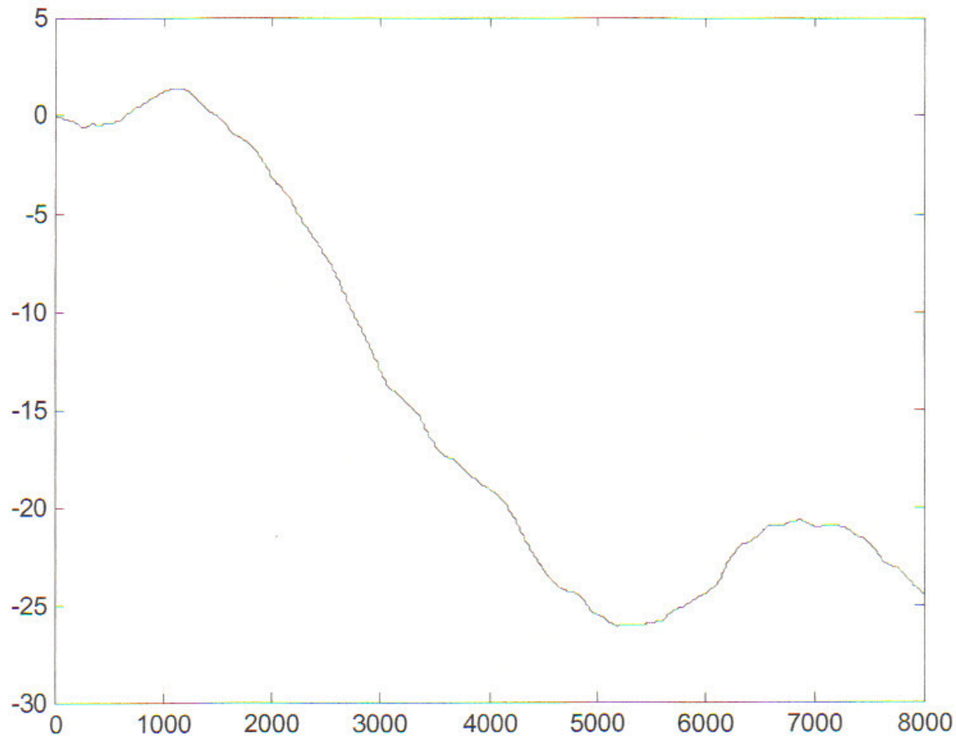

Figure 5 Fractional Brownian Noise, Hurst exponent 1.3

Harrell, in the technical report [TR-01-16, 2001] and papers [2004][2005] has investigated ways of characterizing vehicle test courses by the inverse power that determines the power spectral density functions. In the technical report TR-01-16 by Harrell, Willoughby, and Grey wavelet functions were used to analyze and characterize the power spectrums. In the papers Harrell[2004] and Harrell [2005] a scaled averaging which a series of detrending plots were used to compute inverse powers of the power spectrum. These fractional powers of the power spectrum going one direction on the test courses are similar to the fractal dimensions of the Brownian noise shown above, (see the reference by Mallat for the mathematical theorems and an explanation of this correspondence). In the approach of the books by Barnsley [1988] and Massapust[1994] a recurrent iterative sequence of linear mappings or a series of 2-D wavelet interpolative functions are used to generate fractal surfaces from a pre-encoded starting seed points or from wavelet coefficients. For this paper, the surfaces were first considered to be of fractal nature in two independent directions and the functions in the MATLAB wavelet toolbox and wavelet software from the book by Mallat were used to compute surface fractal dimensions by wavelets. We wrote in MATLAB code and tested out various Fourier transform 2-D surface generation programs, such as those found in the book The Science of Fractals by Pietgen and Saupe. These programs generate fractal surfaces of a given Hurst exponent for both coordinates or different exponents depending on the orientation of the x, y axes. It seems difficult to use this approach to take given boundary and interior values as given from the data. And, it is hard to understand how to use them to generate multifractal surfaces of a given Hurst exponent and lacunarity parameter.

In order to define the term **multifractal** we refer to the books by Harte [2001 chapter 1] and Pietgen and Saupe [1988 pages 66 and 67]. Harte defines what he terms a multifractal measure to be an extremely irregular measure defined on the Borel subsets of a probability space in real multidimensions such that its distribution is not differentiable, with singularities of possibly many different orders [see Harte for the definition of a probability space, distribution, and Borel subsets. The reader may want to refer to this book also for many examples of multifractals that occur in the application areas of differential dynamical systems, turbulence theory, rainfall fields ,earthquake modeling and simulations, power spectrum laws in hydrological studies and the theory of Brownian motion stochastic processes] Pietgen and Saupe have formulated the following somewhat more easily testable and computationally precise definition of a multifractal that comes from a paper by Evertsz and Mandelbrot[1992 J. Phys. A and Phys. Rev. A. see also the reference by Falconer, Chapter 17 for other mathematical theorems that are related to multifractal measures]. Let our space of interest now be called S. A spatial arrangement of points determines a function $P(m,L)$. " $P(m,L)$ is the probability that there are m points within and E-cube (or sphere) of size L centered about an arbitrary point in S. $P(m,L)$ is normalized so that $\sum_{m=1}^{N} P(m,L)=1$ for all L. … The usual quantities of interest are derived from the moments of $P(m,L)$. The mass dimension, $M(L)=\sum_{m=1}^{N} mP(m,L)$ and the number of boxes of size L needed to cover S. $N_{box}(L)=\sum_{m=1}^{N}(1/m)P(m,L)$ and the configurational entropy when space is divided into cubes of size L $S(L)=\sum_{m=1}^{N}\log mP(m,L)$. For a fractal set $M(L)\propto L^D, N_{box}\propto 1/L^D$, and $e^{S(L)}\propto L^D$. In fact, one can define all moments $M^q(L)=\sum_{m=1}^{N} m^q P(m,L)$ for $q\neq 0$ (the q = 0 case is given by the equation above) and one can then estimate D from the logarithmic derivatives $D_q = 1/q < \dfrac{\partial \log M^q(L)}{\partial \log L} > q\neq 0$ and $D=1/q<\dfrac{\partial S(L)}{\partial \log L}>$ for q =0… For a uniform fractal (fractal set) as the number of points examined, $N\to\infty$ the distribution is expected to take the scaling form $P(m,L)\to f(\dfrac{m}{L^D})$ and all moments give the same D. For a non-uniform fractal (a fractal map such as a Poincare map) the moments may take different values. Shapes and measures requiring more than one D are known as **multi-fractals**." Here the term Poincare map refers a map in dynamical systems theory identifying the periodic return of system coordinates in variable configuration space. This is explained more fully in the references by Ruelle [1989] and Harte [Chapter 10 2004] quoted previously. The references by Pietgen, Jurgen, and Saupe [1991 and 2001] and also by Mallat[1998] contain a discussion of how to define a multi-fractal measure on a set and computer algorithms which can be used to estimate and construct a graph of it. The notion of **lacunarity** is defined and explained in Chapter 34 of Dr. Mandelbrot's book [ibid page 310]. "the basic observations about galaxies that led me to distinguish two aspects of texture, calling them lacunarity and succolarity. Lacuna (related to lake) is Latin for gap, hence a fractal is to be called **lacunar** if its gaps tend to be large, in the sense that they include large intervals (discs, or balls). And, a succolating fractal is one that "nearly" includes the filaments that would have allowed percolation; since percolare means :"to flow through" in Latin, succolare (sub-colare) seems the proper neo-Latin for "to almost flow through".

The best way to generate fractal surfaces with arbitrary Hurst exponent and lacunarity parameter, which also allow interpolation from boundary values that we are familiar with is the midpoint displacement algorithm which uses successive random additions. Pseudo code for this algorithm is given in Pietgen and Saupe, chapter 2. The algorithm performs a series of recursive, scaled interpolations starting from the boundary values for the surface which are given as input. It works on a scaled square lattice of points. If there is a mesh size which denotes the resolution of such a grid **delta**, the algorithm generates another square grid of resolution **delta**/ $\sqrt{2}$ by adding the midpoints of all squares. The new square lattice is rotated 45 degrees. By adding the midpoints again we have a the next square lattice which is oriented the same as the first one. Thus we scale the resolution with a factor of r = **1**/ $\sqrt{2}$ and we add a random displacement using a variance which is r^{2H} times the variance of the previous stage. Here **H** is the Hurst exponent. There is an input parameter in the program which determines the maximal level of recursion. This can be set to make the output more detailed and at a higher resolution. The parameter **delta** which sets the standard deviation of the Gaussian noise is scaled according to the Hurst exponent and is used as an offset in the interpolation formulas. The parameter **sigma** in the formulas, which is also written as the standard deviation of a random noise parameter determines the lacunarity of the fractal surface. The algorithm was written in terms of MATLAB code, which is included in Appendix A. The program in appendix A (surfaceplt3.m) also plots a histogram of the x,y coordinate values and the z values of the fast fourier transform of the surface which is generated. This has been included in order to be able to compare the goodness of fit of the surface which is generated with the sensor data that the interpolation scheme is replacing the blank areas with.

Results and Conclusions

If one runs the function with parameters surfaceplt3(6,0,.7,.2,0,0,0) using the MATLAB code in Appendix A.
A 2^6 x 2^6 intepolated fractional Brownian motion surface with Hurst exponent H=.7 and lacunarity parameter sigma=.2 is generated as shown in the Figure 6 below: The computer generated coordinate values (in inches) are plotted on the x,y, and z axis. Because the programs needed to match all the coordinates of the blank areas in the experiment with the generated fractal surface, these are not the measured terrain values from the experiment. The algorithm as given in the appendix generates a surface of extent 2^6 x 2^6 and it is plotted using this range for this example. Figure 7 shows the histograms that the program computed which characterize the surface's coordinate values and their fast Fourier transforms. The histogram plots of the surfaces show each row or column of the surface in different colors. The x-axis of the first histogram plots frequency in cycles/inch. Its y-axis is the square of the magnitude of the occurance of these frequencies plotted in $(\text{cycles/inch})^2$. The algorithm computes the interpolated surface values as a set of complex numbers then takes the