real coordinates of the set of values as the surface to plot. The surface profiles themselves are displayed as color coded values after the coordinate values are transformed by the root mean square function. For each x coordinate row value (in inches) on the abscissa, the histogram function plots different colors that represent the number of different points with that y value as a vertical column. In the MATLAB plot program of Appendix A, surfaceplt3, after the interpolated surface is plotted the fast Fourier transform of the coordinates are taken and the root mean square of their values computed. This is what is plotted in the first histogram on the left in figure 7. Taking the root mean square enables the computation of the histogram function, as it is written in MATLAB code to sort the values correctly into bins. The second histogram on the right of figure 7 is a plot of the x,y coordinates of the the interpolated surface before their Fourier transform and root mean square coordinate values are computed. This plot visually shows a representation of the distribution of the surface coordinate values that the interpolation, at that degree of roughness and lacunarity, has generated.

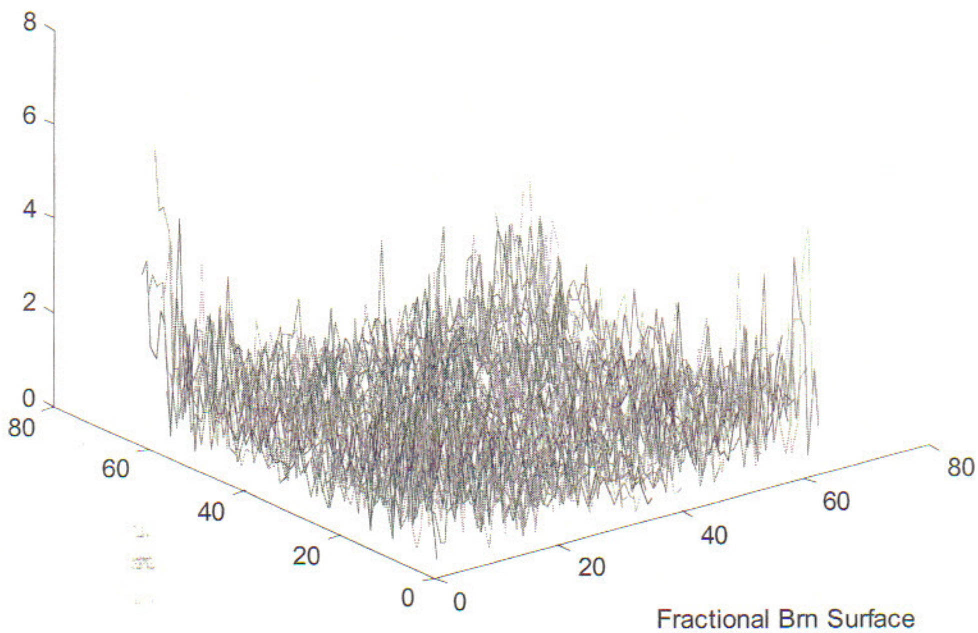

Figure 6 Interpolated surfaces using 64 levels, 4 randomly generated points on the boundary, Hurst exponent .7 and lacunarity parameter .2

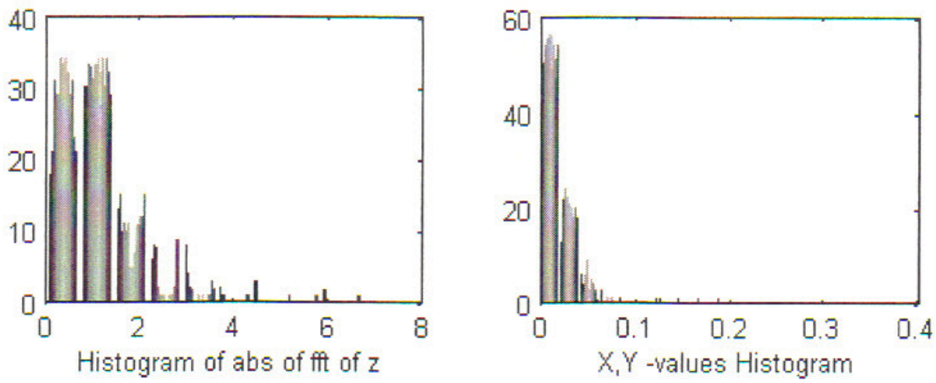

Figure 7 Histograms which characterize the coordinate values and spectrums of the interpolated surface from Figure 6

Figures 8 and 9 show the surface and corresponding histograms generated using the same parameters, but using 32 levels of resolved interpolation. The spatial extent for the coordinates of the surface at this level of resolution is 2^5 x 2^5 and it is plotted over this range. In the figure. Notice, that, except for a scaling factor, there is very little difference in the distribution of coordinate values of the histograms of two surfaces generated with the same fractal parameters, but calculated to different levels of interpolated resolution. Fractal dimensions can be calculated using a renormalized box-counting scaling process (cf. chapter 4 Pietgen and Saupe, 1992 and 2001). Thus, this observation indicates that changing the resolution in process will not change the dimension of the surface's power spectral density which we are trying to match up.

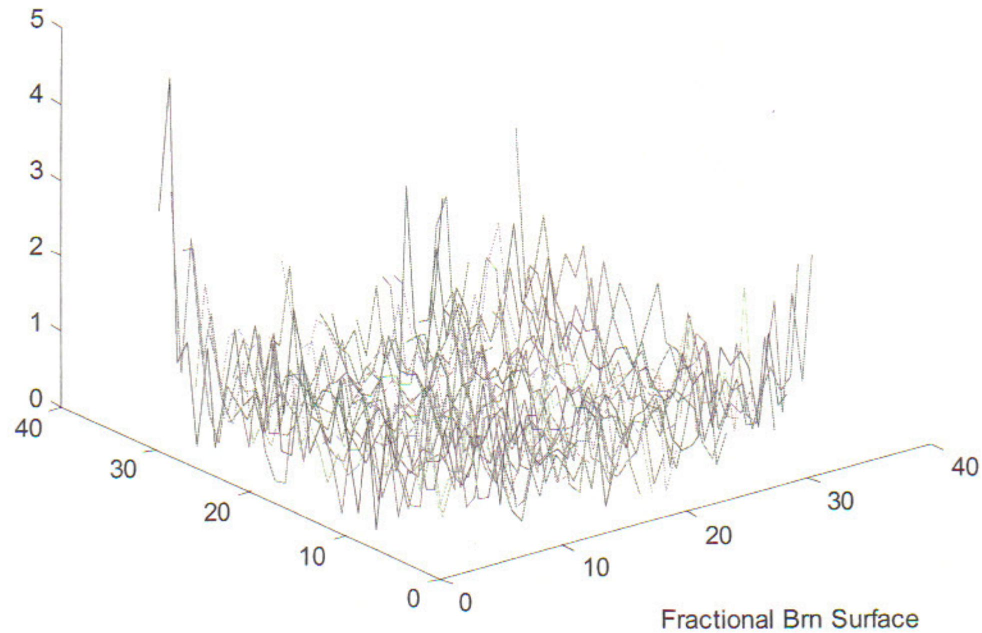

Figure 8 **Surface in figure 6 with 32 levels of interpolated resolution**

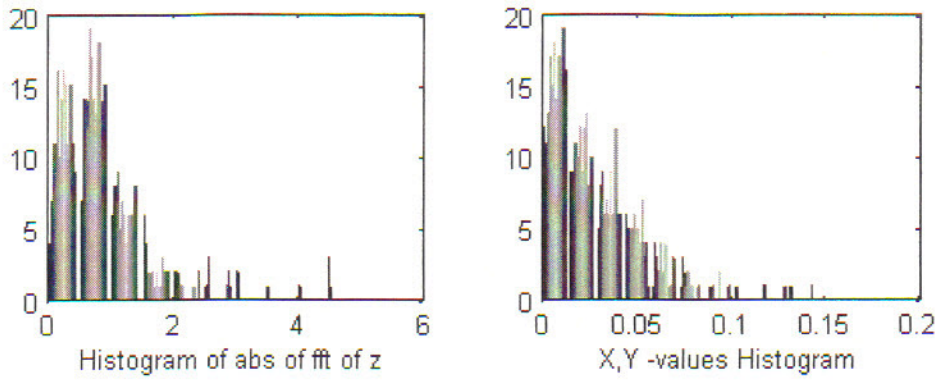

Figure 9 Histogram values for surface in figure 7

In conclusion it is our opinion that the mid-point interpolation algorithm is suitable to use for filling in the blank areas of sensor data. Examination visually of the histograms of different levels of resolution in the output of the algorithm shows it to be robust in this sense. This approach is preferable to a using vector product of two one-dimensional interpolations because we don't want to assume the information in the blank areas is necessary symmetric in that sense. As stated in the scope section of the paper in the introduction, we do not attempt to match up 3-D plots of the resulting interpolated surfaces. Visual comparisons of histograms of data point values and Fourier transforms of the consistency of the interpolated data at various scales and resolutions have been used to determine results. Future work that we do in this area should concentrate on developing good workable programs to estimate by wavelets the fractal surface dimensions that occur in the ground truth of various types of terrain that the sensors will be looking at. After the range of dimensions that we want to consider is determined a measure of effectiveness should be constructed to match up the residuals errors between the rest of the surface and the blank area with the rest of the surface and the output of the algorithm replacing the blank area. The parameters in the algorithm that affect lacunarity and texture by generating normal Gaussian errors can then be varied in order to investigate how robust the algorithm's output is in a more precise mathematical sense.

Acknowledgements: The definition of the problem, research direction, funding, and introduction to this paper was provided and written by Dr. John Peters, Engineering Systems and Materials, GSL, ERDC. Stacy Howington of Engineering Systems and Materials, GSL, ERDC supplied the sensor data along with the 2-D contour plots. The mathematical research, computer programs and the scope, data gathered from experiments, mathematical methods, and conclusions and results sections of the paper were written by Dr. Andrew Harrell of the Mobility Systems Branch, Engineering Systems and Materials, GSL.

References:

Abry P., Gonzalves P., Flandrin, P. 1995. Wavelets Spectrum Analysis and 1/f Processes" pp. 15-31 in Wavelets and Statistics. Lecture Notes in Statistics # 103, Springer Verlag, New York, NY.

Apostol, T.M. 1974. Mathematical Analysis, 2nd ed. Addison Wesley, 492 pp.

Barnsley, Michael. 1988. Fractals Everywhere. Academic Press. Orlando Florida. 394 pp.

Cressie, Noel A.C., 1993. Statistics for Spatial Data. John Wiley, 900 pp.

Daubechies, I. 1992. Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics (SIAM). Philadelphia, PA. 357 pp

Evertsz, Carl J.G. and Mandelbrot, Benoit B. 1992. Harmonic Measure Around a Linearly Self-Similar Tree. J. Phys. A. 25: 1781-1797.

Evertsz, Carl J.G. and Mandelbrot, Benoit B. 1992 Variability of the From and of the Harmonic Measure for Small Off-Lattice Diffusion Limited Aggregates. Phys.Rev.A. 45:5798.

Falconer Kenneth. 1990. Fractal Geometry, Mathematical Foundations and Applications. John Wiley and Sons. Sussex, England.288 pp.

Harrell Andrew W., Grey Cliff, Willoughby William. 2001. Application of Wavelet Analysis to Logistics Test Vehicle Ride Experimernt Results. Geotechnical and Structures Laboratory, U.S. Army Corps of Engineers Engineering Research and Development Laboratory. Rpt. ERDC/GSL TR-01-16.Vicksburg, MS 39180.43 pp.

Harrell Andrew W.. 2004, Characterizing Vehicle Test Courses, Journal of the MS Academy of Sciences.49:135-166.

Harrell Andrew W.. 2005, Characterizing Urban Test Courses, Transactions of the Mathematics, Computer Science, Statistics Div. of the MS Academy of Sciences.2005:1-56

Harte, D. 2001. Multifractals, Theory and Applications. Chapman and Hall. Boca Raton, FL. 248 pp.

Kay S. M. 1988. Modern Spectral Estimation. Prentice Hall. Englewood Cliffs, NJ.543 pp.

Mallat Stephane. 1998. Wavelet Tour of Signal Processing. Academic Press. San Diego CA. 577 pp.

Mandelbrot, Benoit B. 1977. The Fractal Geometry of Nature. W.H. Freeman and Company, New York. 468 pp.

Massopust, Peter R.. 1994. Fractal Functions, Fractal Surfaces, and Wavelets. Academic Press. San Diego, CA..383 pp.

Meyer, Yves. , Robert D. Ryan. 1993.Wavelets, Algorithms and Applications. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.,133pp.

Misiti, M., Misiti, Yves, Oppenheim, George, Jean-Michel Poggi. 1996. Wavelet Toolbox. The Math Works Inc. Natick MA.624 pp.

Pietgen, Heinz Otto, Jurgens, Hartmut, Saupe Dietmar, 1992 and 2001, Chaos and Fractals, New Frontiers of Science, Springer-Verlag, New York.984 pp.

Pietgen, Heinz Otto and Saupe, Dietmar. 1988. The Science of Fractals Images. Springer-Verlag. New York. 312 pp.

Press, W. H., Vetterling, W.T., Teukolsky, S. A., Flannery, B.P.. 1992. Numerical Recipes in C, 2nd ed. Cambridge U. Press, New York, N.Y., 996 pp.

Ruelle, David. 1989.Elements of Differentiable Dynamics and Bifurcation Theory. Academic Press, San Diego, CA. 187 pp.

Turcotte, Donald L. 1992. Fractals and Chaos in Geology and Geophysics. Cambridge University Press. Cambridge, UK , 398 pp.

Walter, G.C. and Shen, Xioping. 1998. Positive Estimation with Wavelets. Contemporary Mathematics.216:63-81.AMS. Providence, RI..

Appendix A

```
function plotint(filename,outf,percent)
% function to sample and plot
% a datafile of x,y and z coord
% values of pts. percent is the
% decimal amount of rows in the original
% file to plot
A=dlmread(filename,' ');
C=A(:,1:3);   % extract first three cols.
[n,m]=size(C);
szz=n;
subrows=fix(percent*szz);
C1=C(1:subrows,:);
C1=C1*10^2;
B2=sort(C1,1);
D2=sample32D(C1,1);
dlmwrite([outf,'x'],D2(:,:,1),' ');
dlmwrite([outf,'y'],D2(:,:,2),' ');
dlmwrite([outf,'z'],D2(:,:,3),' ');
XI=D2(:,:,1);
YI=D2(:,:,2);
```

```
ZI=D2(:,:,3);
j=1;
mesh(XI,YI,ZI), hold
for i=1:j:szz
plot3(C1(i,1),C1(i,2),C1(i,3),'.');
end
end
```

```
function Aest=sample32D(A,detailfctr)
% function to sort the coordinates and
% plot a certain percent the sensor data
Bx=sort(A,1);
By=sort(A,2);
[n,m]=size(Bx)
sz=n
mnx=Bx(1,1);
mxx=max(Bx(:,1));
mny=By(1,2)
mxy=max(By(:,2));
intn=fix(sqrt(sz));
%intn=intn*2
intn=intn*fix(sqrt(10^detailfctr))
xcellsize=(mxx-mnx)/intn
ycellsize=(mxy-mny)/intn
for i=1:intn
    for j=1:intn
   Aest(i,j,1)=mnx+i*xcellsize;
   Aest(i,j,2)=mny+j*ycellsize;
    Aest(i,j,3)=0;
    end
end
Aest(:,:,3)=griddata(Bx(:,1),Bx(:,2),Bx(:,3),Aest(:,:,1),Aest(:,:,2));
end
```

```
function surfaceplt3(maxlevel,npts,H1,r,addition,fileread,filesave)
% surfaceplt3 -- Plot 2-d Fractional Brownian Surfaces
% along with histograms of their profile elevations
% and fast Fourier spectral densities.
%
% note: the histogram plots of surfaces show each
% row or column of the surface in different colors.
% the surface profiles themselves are displayed in
% as color coded values in root mean squared
% form. Taking the root mean square, enables
```

```
%   the histogram function, as it is written in MATLAB, to
% sort the values correctly into bins
%  the fileread filesave boolean values determine whether
% the boundary initial values are read from a file
% and saved .
% maxlevel is the dim of the input value matrix
% in terms of a power of two.
% npts is the number of initial value pts
% to be specified
A=intsurfaceplt(maxlevel,r,H1,0,fileread,filesave,npts);
n=2^maxlevel;
% A is defined as a complex matrix here to allow
% the fast fourier transform function to be called
x=real(A);y=imag(A);
invx=fft2(A);
xi=real(invx);yi=imag(invx);
x1=abs(x);
for k=1:n-1
   for j=1:n-1
       z(k,j)=sqrt(xi(k,j)^2+yi(k,j)^2);
   end
end
ti=1:n-1;
[XI,YI]=meshgrid(ti,ti);
plot3(XI,YI,z);
xlabel('Fractional Brn Surface');
figure
subplot(2,2,1);
hist(z);
xlabel('Histogram of abs of fft of z');
subplot(2,2,2)
hist(x1);
xlabel('X,Y -values Histogram');



function Y=intsurfaceplt(maxlevel,sigma,H,addition,bdvalues,filesave,npts)
%   fractal surface generator from Pietgen, Saupe Chap. 2
%  maxlevel=maximal number of recursions, N=2^maxlevel
%  sigma=initial standard deviation
%  H=fractal dimension D=3-H
%  addition =boolean variable to turn random
%         additions on and off
%  bdvalues =boolean variable to turn
%  the specifcation of initial boundary values on and off
%    f3 and f4 functions to average surrounding pts
N=2^maxlevel;
delta=sigma;
% set the initial random corners
if bdvalues == 0
% set the initial random corners
X(1,1)=delta*WhiteNoise();
X(1,N+1)=delta*WhiteNoise();
X(N+1,1)=delta*WhiteNoise();
X(N+1,N+1)=delta*WhiteNoise();
else
  X=getinitvalues(maxlevel,npts,'init.dat',bdvalues);
end
if filesave
   dlmwrite('init.dat',X,' ');
```

```
end
D=N;d=N/2;
for stage=1:maxlevel
   % rotating from grid type to type at 45 degrees
   delta=delta*(.5)^(.5*H);
   for x=d:d:N+1-d
      for y=d:d:N+1-d
         if x+1<N+1 & y+1<N+1
         X(x+1,y+1)=f4(delta,X(x+d+1,y+d+1),X(x+d+1,y-d+1),X(x-d+1,y+d+1),X(x-d+1,y-d+1));
         end
      end
   end
   % displace other points if needed
   if addition==1,
      for x=1:d:N+1
         for y=1:d:N+1
      if x+1<N+1 & y+1<N+1
            X(x+1,y+1)=X(x+1,y+1)+delta*WhiteNoise();
      end
         end
      end
   end
   % rotating grid again
   delta=delta*(.5)^(.5*H);
    for x=d:d:N+1-d
        if x+1<N+1
  X(x+1,1)=f3(delta,X(x+d+1,1),X(x-d+1,1),X(x+1,d+1));
  X(x+1,N+1)=f3(delta,X(x+d+1,N),X(x-d+1,N+1),X(x+1,N-d+1));
X(1,x+1)=f3(delta,X(1,x+d+1),X(1,x-d+1),X(d+1,x+1));
 X(N+1,1)=f3(delta,X(N,x+d+1),X(N,x-d+1),X(N-d+1,x+1));
        end
    end
    % interpolate and offset interior grid pts.
     for x=d:d:N+1-d
      for y=d:d:N+1-d
  if x+1<N+1 & y+1<N+1
         X(x+1,y+1)=f4(delta,X(x+1,y+d+1),X(x+1,y-d+1),X(x+d+1,y+1),X(x-d+1,y+1));
  end
      end
     end
     for x=d:d:N+1-d
      for y=d:d:N+1-d
          if x+1<N+1 & y+1<N+1
         X(x+1,y+1)=f4(delta,X(x+1,y+d+1),X(x+1,y-d+1),X(x+d+1,y+1),X(x-d+1,y+1));
          end
      end
     end

     % displace other pts if needed
    if addition==1,
      for x=1:d:N+1
         for y=1:d:N+1
            if x+1<N+1 & y+1<N+1
            X(x+1,y+1)=X(x+1,y+1)+delta*WhiteNoise();
            end
         end
      end
    end

      for x=d:d:N+1-d
```

```
            for y=d:d:N+1-d
                if x+1<N+1 & y+1<N+1
               X(x+1,y+1)=X(x+1,y+1)+delta*WhiteNoise();
                end
            end
        end

 d=d/2;
end
Y=X;
end




function y = f3(delta,x0,x1,x2)

    y =(x0+x1+x2)/3+delta*WhiteNoise();
end




function y = f4(delta,x0,x1,x2,x3)

    y =(x0+x1+x2+x3)/4+delta*WhiteNoise();
end


function A=getinitvalues(maxlevel,npts,filename,fileread)
% function to get initial boundary values
% to determine the solution of the p.d.e.
% If all the values are to be read from a
% file set npts=0, else these values
% are input one at a time from the console
for i=1:2^maxlevel
    for j=1:2^maxlevel
        A(i,j)=0;
    end
end
if fileread==1
A=dlmread(filename,' ');
end
for i=1:npts
    k=input('row= ');
    l=input('column= ');
    x=input('value= ');
    A(k,l)=x;
end

end
```