MONITORING OF THE SEABED USING SIDESCAN SONAR AND FRACTAL PROCESSING

L.M. Linnett, S.J. Clarke, C.St.J. Reid, A.D. Tress.

Dept. of Computing & Electrical Engineering
Heriot - Watt University
Riccarton
Edinburgh, EH14 1AS

1. INTRODUCTION

In recent years there has been an increase in the demand for high quality sidescan sonar imagery for the mapping of sediments on the sea-floor. Much of this interest has come from the oil and gas industries who are interested in the exploitation of mineral resources; the fishing industry who are interested in acquiring data relating to fish stocks such as shell beds; and the defence industry for applications in route survey and object detection. Coupled with this demand for data, has been the development of increasingly sophisticated sonar equipment capable of obtaining high resolution images of the sea-floor. These factors have led to an abundance of high resolution data which currently must be examined visually by trained personnel. This is a subjective and time-consuming process, which relies upon the skill and experience of the interpreter. The challenge has therefore been to develop a fully automatic method of segmenting and classifying sediments on the sea-floor, and detecting and analysing any man-made objects that may be present. As the resolution of the sidescan data increases, the data-size is measured in Gigabytes presenting problems for the storage and manipulation of the data on computer. Hence another challenge is to develop computer algorithms that are capable of processing and displaying such large amounts of data, and also to find a suitable representation for the data which will greatly reduce the storage requirements, whilst retaining the key information.

The visual appearance of a texture contained in a sonar image, is related to the structure of the sea-floor and the sonar reverberation characteristics of the sediment on the sea-floor [1]. With high-quality data the visual appearance of the texture on the sonar image often closely resembles the texture of the sediment as viewed under conventional conditions, which is most apparent for sand-ripple textures. The relationship between the visual appearance of the sonar image and the distribution of the sediments on the sea-floor, makes it possible to map these sediments by classifying the textures contained in the sidescan image.

Before a computer may be used to analyse the textures contained in sidescan sonar data, it is necessary to have a suitable numerical representation of each texture. This task is complicated by the fact that whilst texture may be easily described in qualitative terms, there exists no precise definition which would enable its quantification. This problem has been alleviated to some extent by the advent of the fractal model [2], which has been shown to be useful in modelling naturally occurring signals such as image texture. The fractal model and its application to the analysis of sidescan data is now discussed.

2. THE USE OF FRACTALS IN THE ANALYSIS OF IMAGES

The idea of using fractals for the analysis of texture images was inspired by the impressive results of synthesising images of natural scenes. Some excellent examples of computer generated images of natural scenes using fractals have been produced [3]. Images of mountains, clouds and planets have appeared in the work of Mandelbrot [2]. These ideas have been used and extended by the authors [4] to produce images that resemble the seabed. An example is shown in figure 1.

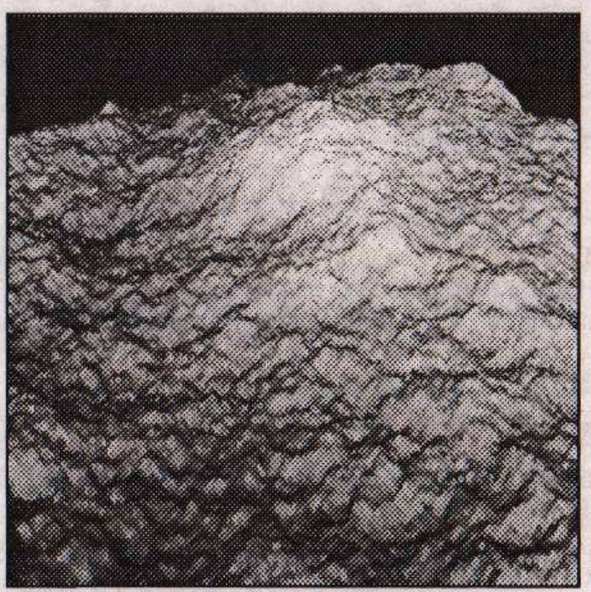


Figure 1 A synthesised fractal "seabed".

Relatively few authors have explored the use of fractals in texture analysis. Keller et al [5] showed its use in segmenting natural scenes, Pentland [6] measured the power spectrum over small image areas and used the fractal dimension as an image descriptor. Medioni [7] found that the fractal dimension was not useful as a texture descriptor and Peleg et al [8] used the change in area with resolution as a measure of fractal dimension to classify textures.

Several techniques have been formulated for measuring the fractal dimension of a texture, namely

a) Perimeter - Area measurements,
b) Box measurements.
c) Change in statistic with resolution.

These are all two dimensional(2D) measurements and none takes directionality into account. It was decided that a technique should be used that allowed directionality to be a feature in the measurement, since it is an essential element of texture description. To this end it was decided to use a set of one dimensional(1D) measurements, each measurement differing in direction. One procedure, that of the "sausage method" [2] looked useful. In this technique the change in length of a curve is measured at differing resolutions. The technique is often used by mathematicians to measure the length of a non-deterministic curve. Mandelbrot has shown that for a fractal line, the sausage method could give a measure of its fractal dimension. Peleg [8] showed that the technique could be used in the 2D case of textures. However, several problems were apparent. While the textures all had differing fractal dimensions they were each measured in isolation. This is an important fact for as Granlund [9] has pointed out ".... *verifying a texture difference when the borders are known is a much simpler task and also one of little practical use....*". Also no account had been taken of directionality. This is important since a texture may be considered as an anisotropic fractal.

Briefly, the background to the measurement scheme is as follows. Mandelbrot found that for many curves, their length could be approximated using the formula,

$$L(\lambda) = k\lambda^{1-D}$$

Where $L(\lambda)$ is the length of the curve measured using a unit of length λ, k is a constant and D is the fractal dimension of the curve and is a constant for that particular curve. One of several methods he proposed for measuring this dimension was to consider points a distance λ from the curve (on either side) so forming a strip 2λ wide. The area of this strip divided by 2λ giving an approximation to the length. As λ decreased, the length increased. Extending this to a surface in two dimensions, i.e. an image, produces an upper level at distance λ above the surface, and a lower level at distance λ below the surface. Computationally, given an image $g(i,j)$, where i, j represents a pixel in the image, then the initial values of the upper and lower images (blankets) are defined as

$$u_0(i,j) = l_0(i,j) = g(i,j)$$

where u and l represent the upper and lower blankets. For any point (i,j), the blankets are computed as

$$u_\varepsilon = \max\left([u_{\varepsilon-1}(i,j)+\lambda],\ [u_{\varepsilon-1}(m,n)]\right) \qquad (1)$$
$$l_\varepsilon = \min\left([l_{\varepsilon-1}(i,j)-\lambda],\ [l_{\varepsilon-1}(m,n)]\right) \qquad (2)$$

where (m,n) is the position of a pixel surrounding (i,j) in an eight connected neighbourhood. λ is the level by which the pixel at (i,j) is raised, and ε is the iteration number.
The volume enclosed between these two layers is

$$V_\varepsilon = \sum \left(u_\varepsilon(i,j) - l_\varepsilon(i,j)\right)$$

The change in volume between iterations ε and ε-1 is

$$\delta V = \frac{V_\varepsilon - V_{\varepsilon-1}}{2\lambda}$$

where the divisor, 2, accounts for the upper and lower layers. Mandelbrot has related the area to the fractal dimension in a similar expression to that for the length, namely

$$A(\lambda) = k\lambda^{2-D}$$

where the exponent has merely increased by 1, ie the dimension has increased by 1.

To illustrate the technique, consider two of the 1D signals from the set of fractal signals shown in figure 2. These two signals represent fractal lines of dimension 1.2 and 1.5. Applying the above ideas to these signals produces the results shown graphically in figure 3, where *log(δA)* has been plotted against *log(iteration(ε))*. δA being the change in area between iterations. The slope of these lines is (1-D) and when measured they give fractal dimensions(D) of 1.22 and 1.51 respectively, a satisfactory approximation. If the value of the level(λ, currently set to 1) is well within the range of the amplitudes of the signal then many different values could be used for λ. If the signal is a true fractal then the result should be independent of the starting value of λ, since true fractals are scale independent.

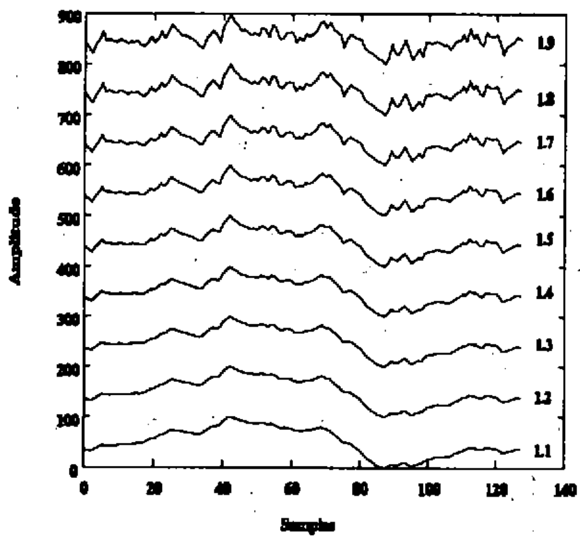


Figure 2 Fractal signals, dimensions 1.1 to 1.9.

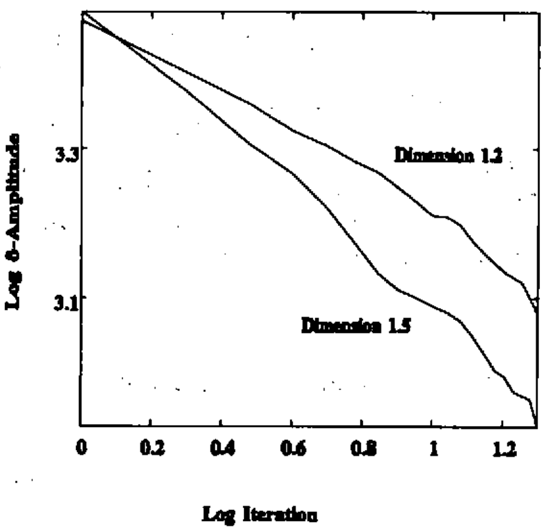


Figure 3 Measurement of fractal dimension.

In order for successive iterations to generate significant features, each feature or iteration must contribute new information to the feature vector. In the frequency domain there appears to be little change in the spectral content with successive iterations after the first. The effect of successive iterations is therefore investigated.

The first aspect to consider is the correlation of the features between successive iterations. If successive iterations are highly correlated then there will be little or no information content to be analysed by a statistical method and therefore little need to include several iterations. To investigate this aspect, consider the fractal signals shown previously in figure 2. Using a λ value of 1 and an operator length of 3, five iterations for producing the difference signal were performed. The correlations between successive difference signals were then computed, for example, iteration 1 with iteration 2, iteration 2 with iteration 3 etc. The results are tabulated in table 1. From this it can be seen that there is a high correlation between successive iterations, the correlations increasing as the number of iterations increases and being large for all signals. The conclusion is that it would not be useful to use successive iterates as additional features for a statistical classifier. It may also be seen that as the fractal dimension and thus the high frequency content of the signal increases, the correlation for any particular iteration decreases.

Fractal Dimension

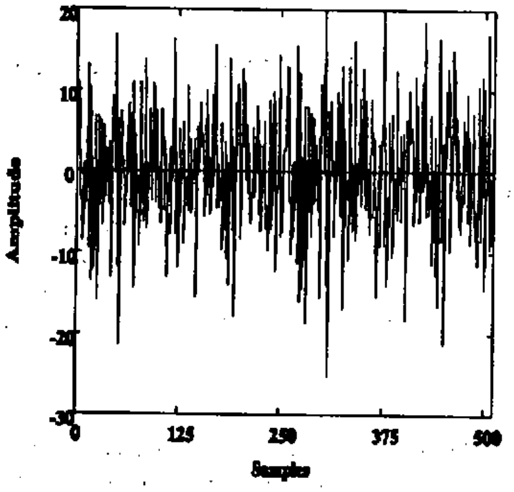
Iterates	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
1,2	0.844	0.808	0.787	0.779	0.762	0.740	0.723	0.710	0.702
2,3	0.923	0.897	0.877	0.864	0.848	0.830	0.811	0.800	0.794
3,4	0.962	0.951	0.940	0.927	0.912	0.891	0.874	0.869	0.862
4,5	0.972	0.964	0.955	0.944	0.929	0.912	0.904	0.902	0.902

Table 1. Correlation between iterates for fractal signals 1.1 to 1.9.

2.1. Experiments on 1-Dimensional Signal Separation.

Four sets of 1D signals were synthesised, two from a Gaussian probability density function with characteristics **N(0,50)** and **N(0,70)**, and two from a Uniform probability density function with characteristics **U(0,50)** and **U(0,70)**.

Both the Uniform and Gaussian sets of signals have very small differences in variances (50 and 70), the same mean, and yet they are capable of being separated. This can be shown graphically as follows: the two Gaussian signals have been put together side by side as one signal, shown in figure 4a. After running three iterations (λ=1) and then performing a sliding *t*-test (window=29) on the result of the third iteration, the results of figure 4b are produced. Also shown are the results of the sliding *t*-test on the original signal. The straight line indicates the 0.1% significance level. Figures 5a,b show the equivalent results for the uniformly distributed signals. In the sliding *t*-test a window is taken either side of a sample point, the *t*-statistic is then computed from the two windows, large absolute values of the statistic being regarded as significant, the windows are then moved along one sample and the *t*-statistic recomputed. It is these *t*-values which are shown plotted in figures 4b and 5b. These show clearly the separation of the two signals at their respective mid-points.



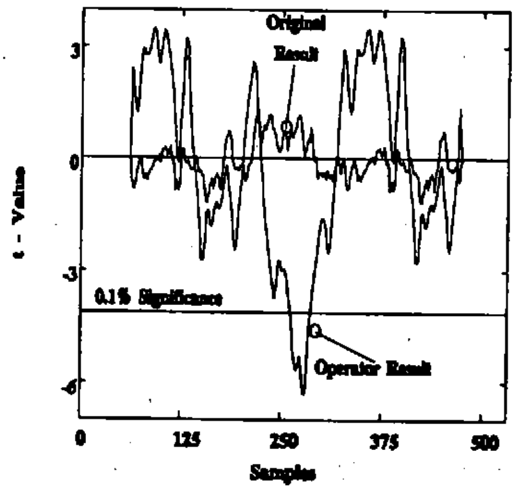


Figure 4 (a) Original Gaussian signals(left) and **(b)** *t*-test results(right).

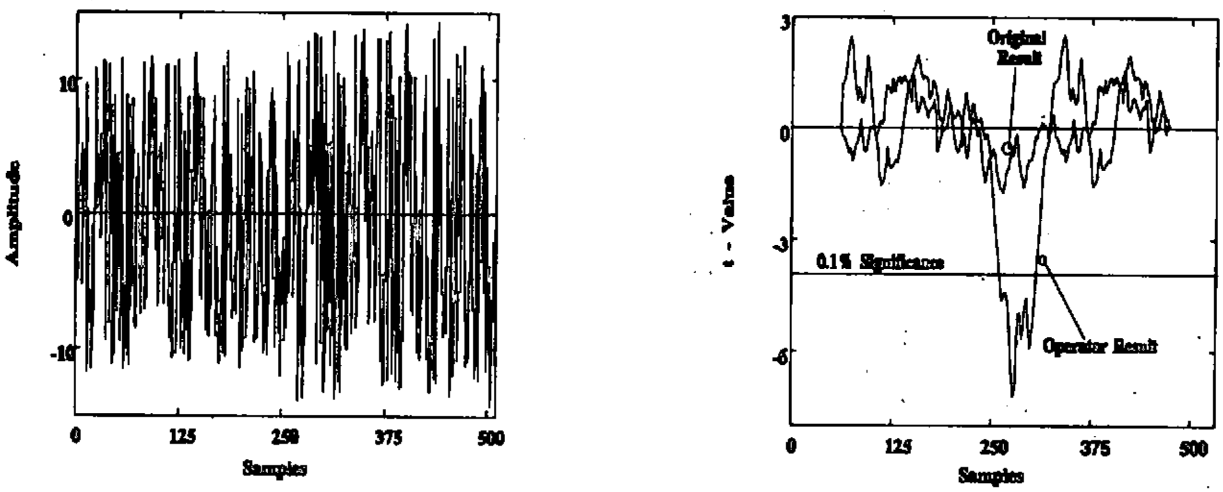


Figure 5 (a) Original uniform signals(left) and **(b)** *t*-test results(right).

The *t*-statistic is used to measure differences in the means of two populations, and so is a suitable test for this example. It is formulated as follows,

$$t = \frac{(\bar{x} - \bar{y})}{s}$$

$$\text{Where} \qquad s^2 = \frac{\sum x_i^2 - n\bar{x}^2 + \sum y_i^2 - n\bar{y}^2}{n(n-1)}$$

$\bar{x}$, $\bar{y}$ are the means of the windows(populations) and n is the size of the window. x_i, y_i represent the i^{th} samples of the windows.

It is concluded that if the iterates of the fractal measures are to be used as features in a statistical classifier, then only a small number of iterations need be used, and that this has the potential to separate signals. It is now speculated that if such a separation can be achieved in one dimension, ie one direction, then significant separation could be achieved by considering many directions and using the resultant set of features as input to a multi-variable statistical classifier. Furthermore, a separation between two one-dimensional signals to indicate the boundaries has been shown, without knowledge of the boundary location and with the two signals appearing as part of the same signal.

2.2 The use of directionality.

Extending the previous sections work to two dimensions, i.e. images, now allows the incorporation of directionality into the measurement scheme. Just as the operator was defined in one dimension using equations (1) and (2) for the lower and upper bounds, this same definition now allows the measurement of these bounds in different directions. Seven different directions based on a 3 * 3 pixel grid are used in extending the operator for the analysis of images. Using directionality the original image has been transformed into several feature images and a pixel at any position has associated with it a feature vector

$\mathbf{X} = \{ x_1, x_2, ..., x_n \}$, where there are n feature images. Within the original image are several textures. This information is used to assign one texture to the pixel under examination given its n dimensional feature vector. The methods used to achieve this are essentially statistical. Further techniques that are not statistical may also be used, e.g. "nearest neighbour" matching of a pixel with its surrounding neighbours employing context.

The key element of the supervised technique is the fact that information about the individual textures is known or obtained before a classification result is achieved. This is usually done by the use of training areas within the original image. These are areas to which the observer has decided to assign specified textures. From these representative samples are derived the statistics of that texture, viz. the mean vector and the covariance matrix, for use in an appropriate model. The most common classification scheme used in image processing is that of Maximum Likelihood Discriminant Analysis. A background to the technique is given in [10],[11]. The basis for most discriminant models is the Gaussian distribution. Using this as the basic model, the feature images are analysed and a probability is given to each pixel in the input image. This probability is a measure of how well the pixel fits to a given class. The classes are the different seabed types. The pixel is then assigned to the class for which it has the highest probability. The output is a classified image based on the seabed types. Figures 6 and 7 show examples of the output, where the boundaries between different classes (textures) have been overlaid on the original images. Figure 6 shows four textures taken from [12], in the form of a collage and showing the separation that can be achieved. Figure 7 is an image of a moderate resolution sidescan sonar image.

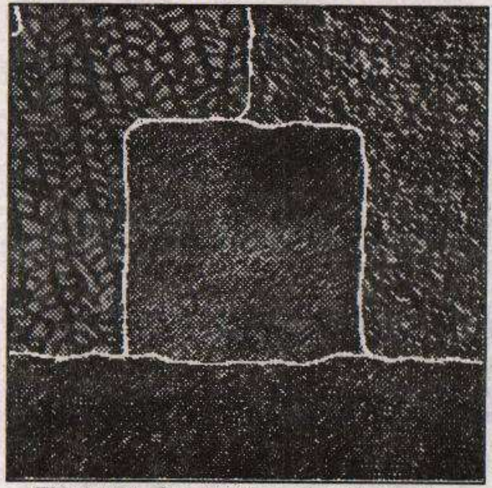


Figure 6. Separation of 4 Textures.

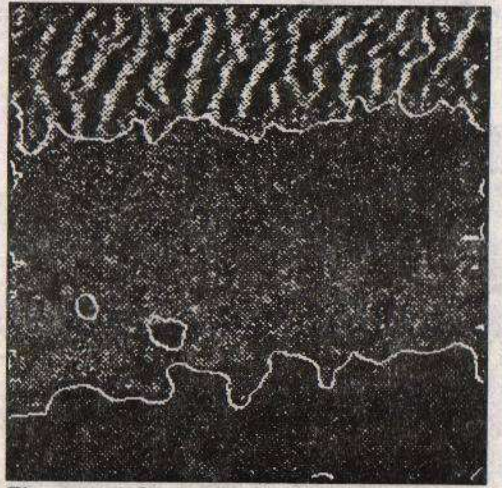


Figure 7 Segmented Sidescan Sonar Image

3. REAL-TIME PROCESSING OF LARGE SIDESCAN DATA SETS

3.1 Large Data Sets

One of the major difficulties associated with the analysis of sidescan sonar information is that very large amounts of data must be processed. Any approach to classifying sidescan information must be designed with this requirement in mind. Practical problems can constrain the type of processing undertaken - for example, if a sonar survey contains hundreds of megabytes of data, it may be impossible to undertake any form of analysis which requires multiple passes through the data set. One important ramification of this limitation is that the techniques used in unsupervised image classification, which tend to require a characterisation of the entire data set using some form of clustering, are unsuitable for this approach to large data set analysis. The authors have therefore concentrated on supervised classification techniques, based on a database of known seabed types. The classification method used in this work is a linear classifier of the type discussed earlier, allowing the extraction of the relative probabilities of each pixel's membership of each seabed (texture) class. This allows the results to be presented either in terms of the relative probabilities (interpreted as the degree of membership of the class), or by using a probability threshold to produce a "don't know" classification - in effect refusing to consign a pixel to any of the available classes, on the grounds of insufficient knowledge. Use of the "don't know" classification is found to improve the accuracy of the classification obtained.

3.2 Texture Databases
Although the use of a database of seabed texture types is in part forced by the bulk of data to be processed, the notion of a limited number of seabed textures is not necessarily a limiting one. Indeed, one of the goals of work is to derive a limited set of generic seabed textures; this set will probably consist of less than 30 seabed types. Work at Heriot-Watt has gone some way towards the goal of distinguishing this number of textures; the feature measures and classification techniques have been dealt with elsewhere. As an illustrative example only, a set of 16 textures from [12] are shown successfully separated in figures 8 and 9; the classification system used a linear classifier and a set of 17 "fractal" features. Although these textures are not synthetic, they are not derived from sidescan sonar, so may not necessarily be representative of the various texture types arising from that source. Therefore, a parallel research effort has gone into distinguishing textures in real sidescan data. In addition to proving the feature extraction and classification processes on authentic sonar data, this has allowed the authors to address the data rate problem posed by sidescan sonar.

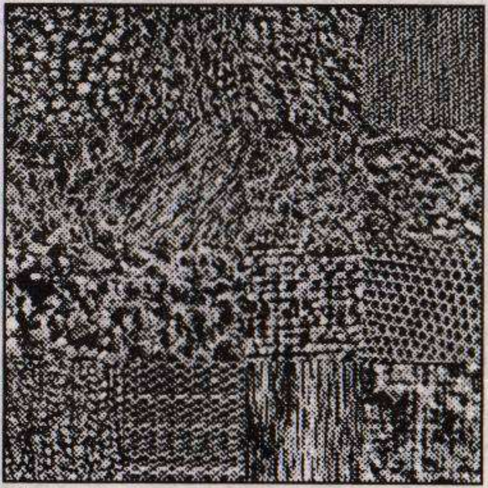


Figure 8 16 textures from Brodatz [12].

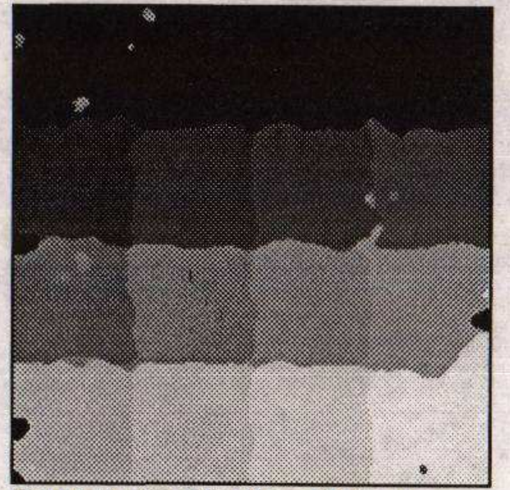


Figure 9 Classification of 16 textures.

3.3 Real-Time Processing
A typical data rate for one channel (port or starboard) of the route survey sidescan sonar used in this work is about 10 lines per second, each line of data containing about 1Kbyte of information. Using relatively unsophisticated equipment (a 50MHz 486 PC running UNIX), it has proved possible to perform feature extraction, classification and display of the results obtained at this rate. A consistent time lag of about 3 seconds is obtained between data input and the display of results; the performance of data processing with

a constant time lag qualifies this as real-time sidescan data processing. A data flow diagram of the processes undertaken is shown in figure 10. The approach to processing large data sets discussed earlier greatly eases the practical problems associated with such a system. Each stage of processing is dependent only on the previous stages, and no feedback is transmitted at any stage. Therefore, the system is implemented as a sequence of filters, which are relatively small bodies of code, allowing them to be written and tested on a piecemeal basis. Figures 11 and 12 show some of the results obtained on a 15000 line (8MByte) sidescan survey of Bigbury Bay, near Plymouth, using a set of seven fractal-based features. Future work will focus on increasing the database of real sidescan seabed textures, increasing the number of features extracted from the data, and extending the performance of the classifier.

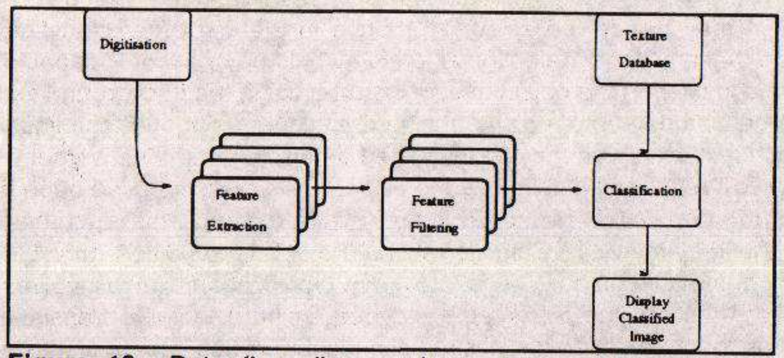


Figure 10 Data flow diagram for real-time sidescan sonar processing

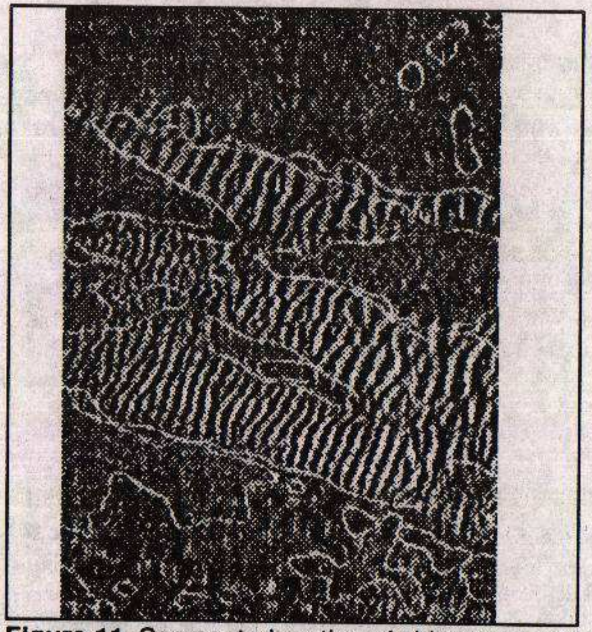


Figure 11 Segmented portion of sidescan sonar data, Bigbury Bay.

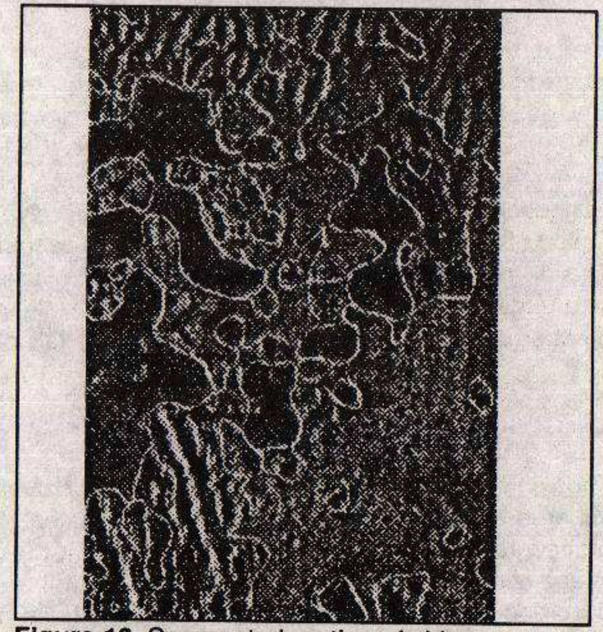


Figure 12 Segmented portion of sidescan sonar data, Bigbury Bay.

4. OBJECT DETECTION

Some applications of sidescan sonar are concerned with the detection of any man-made objects that may be present on the sea-floor, and their subsequent identification. At present this task is performed manually by a trained operator. However, since the presence of such objects is usually a rare occurrence, it is difficult for these operators to maintain their alertness when considering large amounts of data. For this reason a computer algorithm that is capable of reliably performing this task, is of interest.

Figure 13 illustrates a region of seabed containing a man-made object. The object is immediately noticeable to a human observer as a bright reflection associated with a dark shadow. These characteristics indicate that the object may be viewed as a disturbance in the first-order statistics of the background texture. This suggests that a suitable detection algorithm would be one that is capable of identifying these deviations. As we are considering the object to be a disturbance in the background texture, it is necessary to first segment the sidescan data into regions of different texture using the techniques discussed earlier. Once segmented, each texture region may be inspected in turn to determine wether it contains an object. The detection algorithm that has been developed to perform this task, is based upon the inspection of the first-order statistics of the grey-levels contained in the texture region. This algorithm is greatly simplified if the distribution of the grey-levels is Gaussian in nature. This condition may be imposed upon the sidescan data by performing a technique called histogram modification, which involves the computation of the actual grey-level distribution (the histogram) and performing a suitable translation of the grey-level values in such a way as to approximate the desired distribution, in this case a Gaussian distribution. This technique is normally performed globally for the whole image, however in this application it was found that the best results were obtained when the technique was applied in a moving box over the image, with the histogram being modified locally.

Having thus modified the image histogram, a probability threshold is selected. This threshold is used to locate light and dark regions of the image, similar to the human system of finding the light and dark regions caused by shadows and highlights made by sidescan sonar. This probability threshold is used for the lower tail of the density function (dark regions) and the upper tail (light regions). Considering the case of the lower tail (the case for the upper tail is identical). All pixels whose values lie within this probability level are assigned a value of 255 (white), all others are given the value 0 (black). Since there is a homogeneous region of texture, then all white pixels may be assumed to be distributed evenly throughout this new image. This distribution of white pixels can be modelled using a Poisson distribution. Given a subarea of a certain size (e.g. 8 * 8 pixels), the number of white pixels expected within that subarea may be calculated. If this number is x, then the probability that r points fall into a subarea is given by -

$$P(r) = \frac{e^{-x} \cdot x^{r}}{r!}$$

Thus as each subarea within the thresholded probability image is scanned the number of points within that subarea is measured and a probability calculated. In this way a probability image is built up. An identical procedure is used for the upper tail image. If these two images are *ANDED* together a final probability image is obtained. This *ANDING* is used to further enhance the probability of both a dark and light region occurring within a relatively small region. These probabilities result in extremely small values (typically 0.0001%) where a texture abnormality occurs, they are thus best displayed on a logarithmic scale, where white represents the small values (abnormalities) and dark represents the high values (normal texture). Figure 14 shows the resulting probability image alongside the original. Figure 15 shows the original image overlaid onto the probability height image.

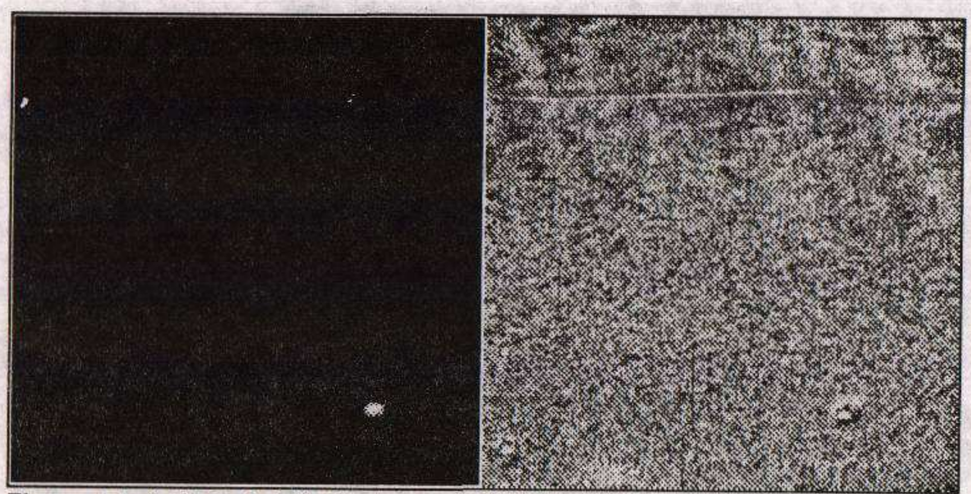


Figure 14 (left) Probability image and (right) original image.

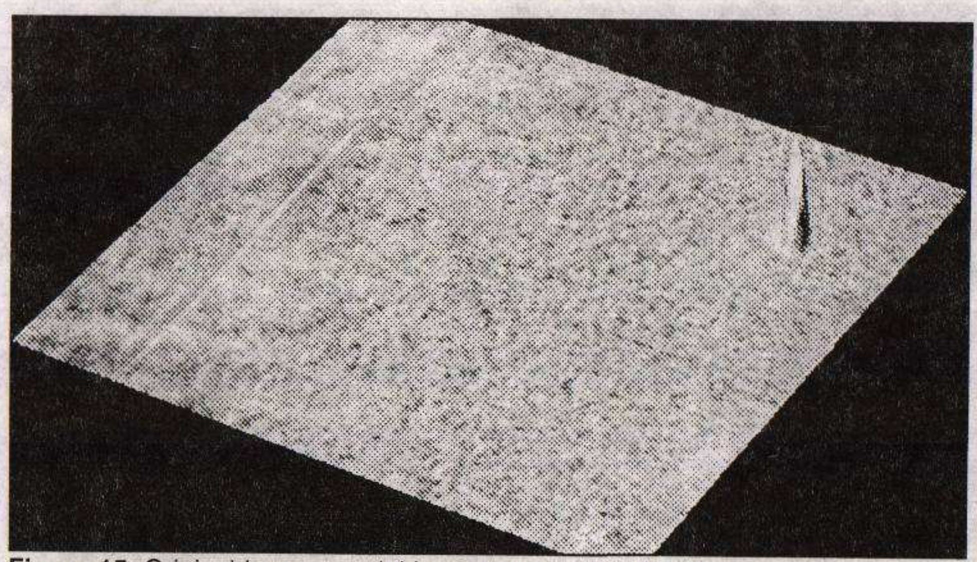


Figure 15. Original Image overlaid on probability height image.

A second example of the technique is illustrated in figures 16 and 17. Figure 16 depicts a region of the seabed containing two textured regions and three objects. This image is first segmented into the two sediment types, and the algorithm is applied to each region. The result is depicted in figure 17, with the original image overlaid on a height map of the probability of an object being present. Clearly all three objects have been successfully detected.

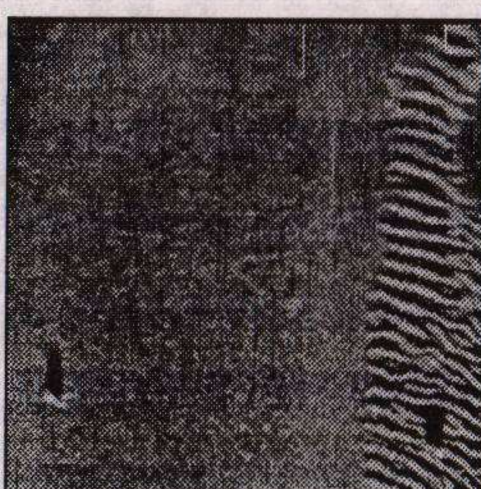


Figure 16 Original image containing three man-made objects.

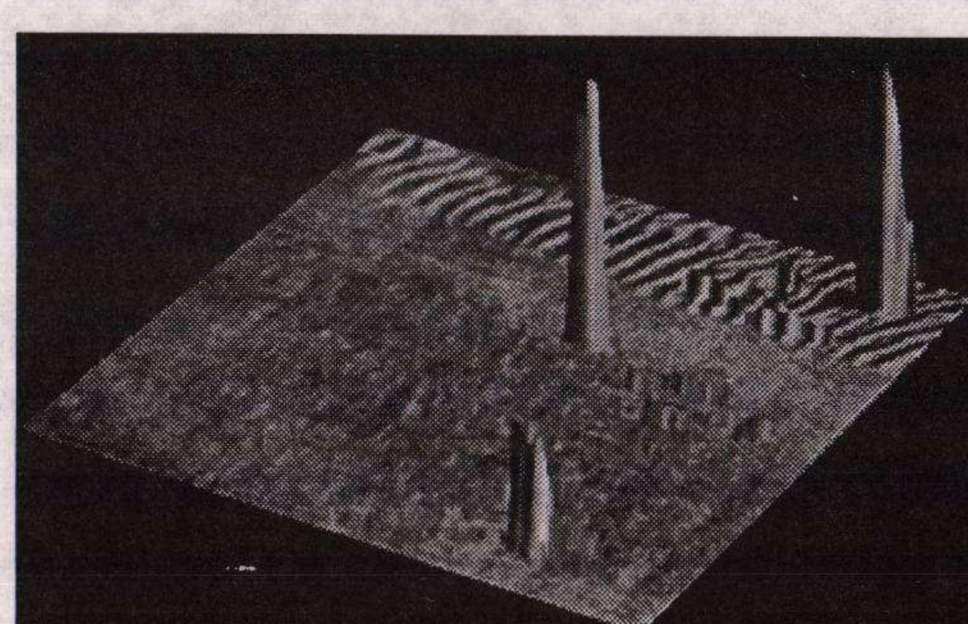


Figure 17 Original image overlaid on probability height image.

5. CODING AND TRANSMISSION OF SIDESCAN DATA

Currently, the majority of sidescan data is obtained by towing a sonar transducer array over the seabed. However, in the future the trend is towards an untethered AUV (Autonomous Underwater Vehicle) performing the sidescan survey. This has the advantage of being able to operate in inaccessible areas such as in the vicinity of oil platforms, and in hazardous applications such as minehunting. One of the technological problems associated with this scheme is concerned with the transmission of the sidescan data from the AUV to the survey vessel. In the absence of a cable the most suitable transmission method is via an acoustic data channel. Unfortunately a reasonable error rate from the acoustic channel requires modest data rates in the region of 1-2 kbits/sec. This data rate is insufficient to transmit high resolution sidescan images in real time, which means that a suitable coding scheme is required to reduce the data-size of the sidescan images, in effect allowing an increase in the bandwidth.

Several image coding techniques [13] have been developed primarily in response to the demand for video conferencing and video telephone transmission over standard telephone networks. These techniques are capable of compression ratios of up to 70:1 for "head and shoulder" images. The redundancy which these coding techniques rely upon, such as homogeneous patches of grey tone (in the background, clothing and face) is not a characteristic of sidescan data. As a consequence, the compression ratios required for real-time transmission over a limited bandwidth channel are much greater than the best achieved using these conventional techniques. However sidescan data contains a special form of redundancy due to the fact that it consists mainly of large areas of homogeneous texture. The approach to coding this type of data described in this paper, is to first segment the sidescan images into homogeneous regions of texture, and then to code the textures and texture boundaries separately. The texture boundaries may be coded using conventional techniques. The subsequent decoding of the image is performed by first decoding the texture boundaries and then filling them with portions of texture synthesised from the appropriate model parameters. This technique is not "lossless" or error free like some conventional techniques, since the exact image is not obtained on reconstruction of the coded data. However, since the recipient of the sidescan data is not normally concerned with each grain of sand contained in the image, the reconstruction of a visually similar image is deemed sufficient for this application. The coded data is transmitted over an acoustic channel and reconstructed at the receiver.

The coding scheme outlined in the introduction consists of two parts:

1. The sidescan image must be classified into homogeneous texture regions.

2. Each texture contained in the image must be coded into a suitable form which will enable the generation of a visually similar texture at the receiver.

The first task is satisfied by the texture classification scheme described earlier. The second task is known as texture synthesis, and may be viewed as the process of estimating certain signal parameters from a sample of a real texture, and the generation of a random signal with the same parameters. Researchers have used parameters from the Markov random field model, the ARMA (autoregressive moving average) model and other stochastic texture models, to synthesise textures. Some of the best synthesis results in the literature have been reported by Gagalowicz et al [14], which are based upon stochastic texture model parameters including the histogram and autocovariance function and second-order statistics amongst others.

The parameters chosen for the texture synthesis procedure adopted in this application are the mean and autocorrelation function of the texture process. This information corresponds to the magnitude response of a stationary and ergodic texture process which is demonstrated by the Wiener-Khinchine relationship [15]. The texture process is assumed to be wide-sense stationary and ergodic, and the mean is estimated from an average of the grey-levels in the texture region, and the autocorrelation function is estimated for several displacements over the texture field.

A technique was developed to generate a stationary texture field with a specified mean and autocorrelation function. This procedure in common with those presented by Gagalowicz, involves two steps. In the first step a random field is generated to possess the mean and variance of the original texture, whilst in the second step the grey-levels of each pixel in the image are adjusted according to a decision criterion which ensures that the autocorrelation function of the synthetic image matches that of the original texture.

To determine the viability of the coding scheme described here, a test was performed at the Western Harbour, Leith Docks, Edinburgh using the example image depicted in figure 18. This image was first classified using the technique described earlier to obtain the classification result depicted in figure 19. The classified image was then used to estimated the model parameters for both textures contained in figure 18. The code required to represent the original image consisted of exactly 1000 bytes (8000 bits), which comprised of 384 bytes of texture model parameters for each texture, and 232 bytes for the coded boundary information. The size of the original image is 262144 bytes, and hence a compression ratio of over 260:1 was achieved using this scheme. The coded data was transmitted a distance of 600m across the Western Harbour using a frequency-hopping modulation scheme [16], at a data rate of 3000 bits/sec. The received data contained no errors, and was used to reconstruct the image shown in figure 20.

A characteristic of this coding scheme, is the fact that the model parameters for a particular texture need only be sent once. Hence if a subsequent image containing the same textures as those in figure 18 was to be coded, only the boundary information need be sent and a much greater compression ratio would be obtained.

The coding scheme could be expanded to include the results of the object detection procedure, in the data to be transmitted. This information would be a sub-image containing the object, extracted from the original data and included at the correct position in the reconstructed image.

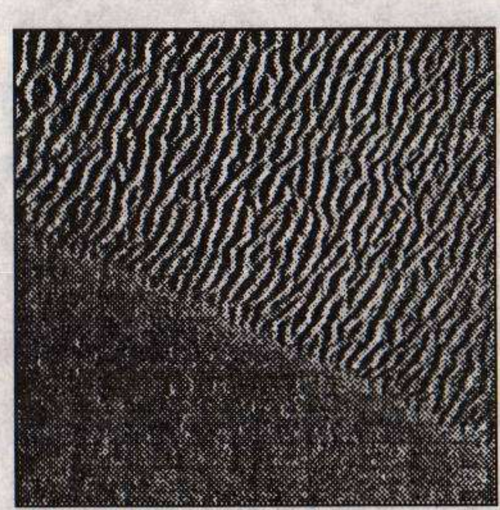


Figure 18 Image used in the coding experiment.

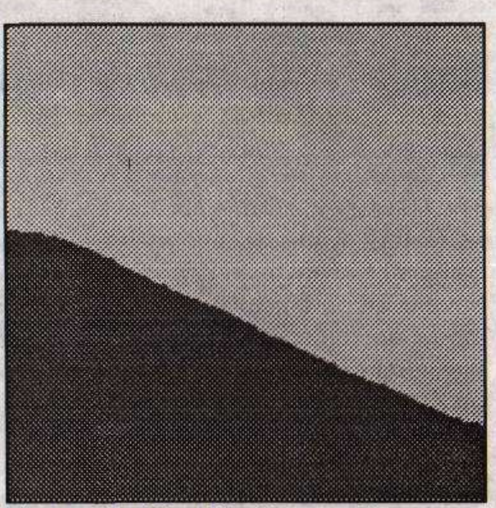


Figure 19 Classification result.

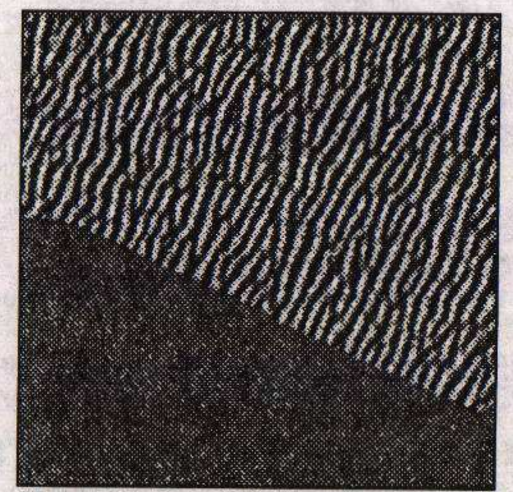


Figure 20 Decoded image.

6. CONCLUSIONS

This paper has attempted to give an overview of some of the current problems associated with the processing of sidescan sonar data. The automatic separation of seabed types on sonar records has been discussed, along with scale invariant processing using fractals. Some indication of future research topics has been addressed, namely efficient data representation and the acquisition of suitable seabed databases.

7. ACKNOWLEDGEMENTS

The authors would like to thank Professor G.T.Russell, Head of Department, Computing & Electrical Engineering, Heriot-Watt University, for his most helpful discussions and enthusiasm, despite extreme pressure on his time.
The authors would also like to acknowledge the encouragement and support from DRA, Bincleaves, in particular, Nick Langhorne, and John Wickenden.
Finally, the help of the rest of the Sonar Processing Research Group at Heriot-Watt University, is gratefully acknowledged.

8. REFERENCES

[1] N.G.Pace, C.M.Dyer, *Machine Classification of Sedimentary Sea Bottoms*, IEEE Transactions on Geoscience Electronics, Vol GE-17, No.3, pp52 - 56, 1979.

[2] B.B.Mandelbrot, *The Fractal Geometry of Nature*, W. H. Freeman, New York, 1983.

[3] R.F.Voss, *Random Fractal Forgeries*, in Fundamental Algorithms For Computer Graphics, R.A.Earnshaw (ed.), Springer-Verlag, Berlin, 1985, pp805-835.

[4] L.M. Linnett, *Multi Texture Image Segmentation*, PhD Thesis, Dept. of Computing & Electrical Engineering, Heriot-Watt University, Edinburgh, March 1991.

[5] J.M.Keller, R.M.Crownover, R.Y.Chen, *Characteristics of Natural Scenes Related to the Fractal Dimension*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.9 No.5, pp621-627, September 1987.

[6] A.P.Pentland, *Fractal based description of Natural Scenes*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, No.6, pp661-674, November 1984.

[7] G.G.Medioni, Y.Yasumoto, *A Note on Using Fractal Dimension For Segmentation*, Proc of the Workshop on Computer Vision, Annapolis, MD, USA, pp25-30, April-May, 1984.

[8] S.Peleg, J.Naor, R.Hartley, D.Avnir, *Multiple Resolution Texture Analysis and Classification*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, No.4, pp518-523, July 1984.

[9] H.Knutsson, G.H.Granlund, *Texture Analysis using Two-Dimensional Quadrature Filters*, IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management-CAPAIDM, pp206-213, 1983.

[10] C.Chatfield, A.J.Collins, *Introduction to Multivariate Analysis*, Chapman-Hall, London, 1980.

[11] W.W.Cooley, P.R.Lohnes, *Multivariate Data Analysis*, John Wiley and Sons, London, 1971.

[12] P.Brodatz, *Textures - A Photographic Album for Artists and Designers*, Dover, New York, 1966.

[13] R.J.Clarke, *Transform Coding of Images*, Academic Press, 1985.

[14] A.Gagalowicz, S.D.Ma, *Sequential Synthesis of Natural Textures*, Computer Vision, Graphics and Image Processing, Vol.30, pp289-315, 1985.

[15] P.A.Lynn, *An Introduction to the Analysis and Processing of Signals*, MacMillan Press, 2nd edition, 1982.

[16] R.M.Dunbar, D.R.Carmichael, *Underwater communication in the presence of high level electromagnetic and acoustic noise*, Final project report to SERC/MTD, ref GR/F01192, Dept. of Computing & Electrical Engineering, Heriot-Watt University, Edinburgh, 1992.