AD-A117 135 ROYAL AIRCRAFT ESTABLISHMENT FARNBOROUGH (ENGLAND) F/G 20/3
TRANSFORMATION OF DATA INTO A RASTER-SCAN FORMAT AND SOME IMAGE--ETC(U)
DEC 81 P A ROBERTS
UNCLASSIFIED RAE-TR-81152 DRIC-BR-82879 NL

1 OF 1
AD A
117135

END
DATE
FILMED
08-82
DTIC

TR 81152

AD A117135

BR82079

TR 81152

ROYAL AIRCRAFT ESTABLISHMENT

*

Technical Report 81152

December 1981

TRANSFORMATION OF DATA INTO A RASTER-SCAN FORMAT AND SOME IMAGE PROCESSING APPLICATIONS

by

P. A. Roberts

*

Procurement Executive, Ministry of Defence
Farnborough, Hants

UDC 681.3.056 : 551.4 : 681.3

ROYAL AIRCRAFT ESTABLISHMENT

Technical Report 81152

Received for printing 15 December 1981

TRANSFORMATION OF DATA INTO A RASTER-SCAN FORMAT AND SOME IMAGE PROCESSING APPLICATIONS

by

P. A. Roberts

SUMMARY

An account is given of techniques that can be used to process and manipulate data that is in the form of either a series of contours or a set of unconnected points, so that it can be transformed into a digital raster-scan format. Two applications of this work are described; these are the deliberate degradation in the spatial resolution of contour maps and the use of terrain height information in conjunction with Landsat imagery. In the discussion of both these topics reference is made to programs available at the Remote Sensing Unit of Space Department, RAE.

Departmental Reference: Space 610

LIST OF CONTENTS

		Page
1	INTRODUCTION	3
2	SUMMARY OF SOFTWARE	3
	2.1 General	3
	2.2 The format of contour files	4
	2.3 The format of grid files	4
	2.4 Converting contour data into a grid format	5
	2.5 Converting a set of unconnected points into a grid format	7
	2.6 The programs	7
3	DEGRADING THE SPATIAL RESOLUTION OF CONTOUR MAPS	8
	3.1 Background	8
	3.2 Basic theory	9
	3.3 Example	10
	3.4 Further applications	12
4	APPLICATIONS TO TERRAIN DATA	13
	4.1 General remarks	13
	4.2 Producing a terrain height matrix	14
	4.2.1 Generation of a height matrix from contours	14
	4.2.2 Generation of a height matrix from spot heights	15
	4.3 Quantities derived from the height matrix	16
	4.4 Use of terrain in conjunction with imagery	17
	4.4.1 Displaying terrain parameters as grey tone images	17
	4.4.2 Applications	18
5	CONCLUSIONS	18
Appendix A	The SYMAP interpolation algorithm and program CONT.GRIDX	19
Appendix B	Contour processing programs	24
Appendix C	Grid processing programs	28
Appendix D	Image processing programs	31
Appendix E	Derivation of the convolution function when the point spread functions of the measurement and required beams are Gaussian	32
Appendix F	Least squares fitting to a rectangular grid of points	35
Table 1		38
References		39
Illustrations		Figures 1-9
Report documentation page		inside back cover

1 INTRODUCTION

Two-dimensional data can be presented in a variety of different ways, *eg* as a contour map. Some formats are more appropriate than others for performing certain operations on such data. One format that is particularly suited to easy manipulation with a computer, and for performing a wide variety of mathematical operations, is that of an equispaced grid of points. Such a format forms the basis on which most digital satellite imagery is handled and it forms the basis of the file format associated with the image processing software at the Remote Sensing Unit of Space Department, RAE, Farnborough. This file format stores an equispaced grid of points as a raster scan of integer values and is known at the Remote Sensing Unit as the 'image format'. This Report describes a series of programs that has been written by the author to handle data that is in the form of either a series of contours (*eg* height contours on a map) or a series of unconnected points. The objective of these programs is twofold; firstly to convert the data into an equispaced grid of points and secondly to perform certain operations on the data when it is transformed to such a format. Once data has been transformed into a grid representation it is trivial for the data to be written in an image format so that all the facilities available with the image processing software of the Remote Sensing Unit can be utilized.

This Report essentially falls into two parts, the first dealing with the software developed and the algorithms employed and the second with two particular applications of this work. The programs themselves will not be described in great detail as most of them perform operations that should be readily appreciated by readers. In those instances where this is deemed not to be the case, reference will be given as to where adequate descriptions of the algorithms employed can readily be found; in one instance this information is contained in an appendix. The two applications that are discussed are the deliberate degradation in the spatial resolution of contour maps and the use of terrain height information with Landsat imagery. Both these topics are described in conjunction with programs that are available at the Remote Sensing Unit.

2 SUMMARY OF SOFTWARE

2.1 General

One of the primary aims of the work described here was to develop a series of programs which would enable contour data to be transformed into the RAE image format, the format in which all types of digital imagery processed by the Remote Sensing Unit is handled. Once data is transformed into this format it can be handled by routines available in the image processing software library and can be used in conjunction with other imagery. In addition to employing the RAE image format it has proved necessary to establish two further file formats to support this work. These are the contour file and the grid file. These formats are described in more detail in sections 2.2 and 2.3. By far the most important algorithms that are used are those which transform data, from either a series of contours or a set of unconnected points, to an equispaced grid of points. Discussion of the two algorithms adopted for these operations is presented in sections 2.4 and 2.5. Finally, section 2.6 provides an overview of all the programs that have been developed in support of this work.

2.2 The format of contour files

The contour file is a text file organised on a line basis with each line containing no more than 80 characters. The first line contains text describing the contents of the file and is followed by a series of one line headers, each of which is followed by lines containing the x and y co-ordinates of a series of data points defining a contour. The co-ordinates are stored as 5 digit integers as the Tektronix 4014 digitising tablet, used for digitising contour maps, generates integers in the range 0-4000. The first header in the file is always associated with fiducial points, generally, though not necessarily, the four corners of the digitised map. A maximum of four fiducial points is allowed.

The format was dictated by a number of considerations. The data is taken to represent a series of contours and/or points representing a surface which is considered to be smooth and continuous. This means that no two contours can cross or touch each other in any way and no two data points may be coincident. An exception to this rule occurs for closed contours where it is required that the first and last points on the contour must be identical, a condition that the program that controls digitisation ensures is satisfied. These constraints limit the different types of contour that have to be distinguished to three:

(1) closed contours;
(2) open contours;
(3) undefined contours, *ie* one or more unconnected points which have the same value associated with them.

The information defining the type of contour is given in the contour type code in the header. This header contains three other parameters:

(1) The number of the contour in the file. (This is not used by any program in controlling its operation, it is included only for convenience in examining and editing files.)

(2) The value to be associated with the contour, *eg* the height.

(3) The number of points in the contour. (The maximum number allowed is arbitrarily set to 5000.)

The file is a text file since it was deemed highly desirable to examine and edit files directly, a feature that has proved exceedingly useful in developing and testing many of the algorithms employed.

2.3 The format of grid files

The grid file is also a text file organised on a line basis with up to 80 characters per line. The first line contains text describing the contents of the file and the second line gives the x and y dimensions of the grid (expressed as the number of points along each side), the co-ordinates of the lower left corner, the grid spacing (which is the same in both x and y directions) and the grid identification code (explained below). Subsequent lines contain the values of the points in the grid starting with the top line of the grid. Each new grid line starts a new line in the grid file. Data points are stored as real numbers but are restricted to values in the range -999.9 to 9999.9, *ie* a 'real' number with not more than six text characters.

A text file was again adopted because of the desirability of directly examining and editing the contents of the file. Generally, this file acts as an intermediary between contour and image formats but is nevertheless necessary because it is considered essential to have provision for negative numbers, a provision that does not exist with the RAE image format.

The grid identification code indicates the type of data that is contained in the file and is as follows:

Code	Data type
1	Height
2	x direction gradients
3	y direction gradients
4	Slope
5	Aspect
6	Convolution grid
0	Data not covered by codes 1-6

These data types are explained in more detail later in this Report.

2.4 Converting contour data into a grid format

The basic problem is to fit to the contour data a surface which is smooth and continuous and on which every contour lies. The surface can then be sampled at the appropriate points to produce the required grid of points. Two major difficulties are encountered in trying to solve this problem. The first is that the surface has to be constrained such that between contours it remains within the range of values defined by the contours. The second arises from the constraint that the surface must pass through the contour data, a condition that would be considerably easier to satisfy if the data was a series of unconnected points rather than a series of lines.

An earlier Report written by the author[1] describes a technique for solving this problem. Very briefly, the method adopted takes a series of equispaced cuts across the map in the x direction and a similarly spaced set in the y direction, the intersection points between the two sets of cuts being the required grid points. This process transforms the original data to a series of points lying along cuts, these points being where the cuts and the contours intersect. The problem now is to fit a surface to this data. One class of interpolation that has proved very popular in recent years is the spline (more particularly the cubic spline). For this particular problem, however, the adoption of a bicubic spline is complicated by the constraint that between contours the surface must remain within the values assigned to the contours. As the data is arranged as contours rather than just as a series of unconnected points, the use of interpolation algorithms based on local fits is inappropriate, since it will often be found that near contours all the points selected will lie on the same contour unless considerable modification is made to the choice and use of nearby points. The solution adopted does not perform a true two-dimensional fit but rather two one-dimensional local fits are applied along each of the cuts to derive values at grid points lying on them. At the same time weighting values are produced that reflect the confidence in the interpolated values.

152

This procedure results in two grids of points corresponding to the two sets of cuts, together with two grids containing the associated weighting values. These four grids are then used to produce the final grid.

The program described in Ref 1 has subsequently been improved. The main modifications are:

(1) The method of combining the values obtained from the two orthogonal cuts to produce the final grid value has been changed.

(2) The program can generate gradients. This data is output as two grids, one grid comprising the x direction gradients and the other y direction gradients.

These modifications are now considered in more detail.

The method now adopted for combining the values obtained from the two sets of orthogonal cuts is as follows. It is assumed that the closer an interpolated value is to one of the points defining the interval in which an interpolated value is required, the more reliable the interpolated value is expected to be. Thus, if an interpolated value is required at a point x on a cut and this point lies between points x_i and x_{i+1} then the associated weighting value w for the point at x is given by

$$w = w_i \qquad (w_i \geqslant w_{i+1})$$

$$w = w_{i+1} \qquad (w_i < w_{i+1})$$

where

$$w_i = \frac{1}{|x_i - x|^n + \varepsilon} \qquad x_i \leqslant x \leqslant x_{i+1}$$

$$w_{i+1} = \frac{1}{|x_{i+1} - x|^n + \varepsilon} \qquad x_i \leqslant x \leqslant x_{i+1}$$

$$\varepsilon = 0.01|x_{i+1} - x_i|^n$$

$$n > 0 \quad .$$

The parameter 'ε' is to ensure that w is finite when x coincides with either x_i or x_{i+1}, otherwise the value of w under such circumstances is unimportant.

For each grid point the values G_x and G_y, which are produced from one-dimensional local fits along the cuts in the x and y directions, are used in conjunction with their respective weighting values w_x and w_y to derive the final grid value G, viz

$$G = \frac{w_x G_x + w_y G_y}{w_x + w_y} \quad .$$

The value of n was chosen to be 1, a value which empirically gave satisfactory results. Lower exponents tended to give a relatively flat surface except near contours where it was necessary to attain the values of the contours, whereas higher exponents often gave rise to a sharp change in gradient in areas nearly equidistant from two or more contours.

The option for generating gradients was easy to implement, since the interpolation function between two data points is of the form

$$y = ax^3 + bx^2 + cx + d$$

and so the gradient is just

$$\frac{dy}{dx} = 3ax^2 + 2bx + c \quad .$$

Gradients are obtained in the x and y directions because cuts across the contour map are made in those directions. It should be noted that these gradients do not correspond to the gradients associated with the final grid of points, since these are obtained from an average of values derived from the two orthogonal sets of cuts, but are rather internal 'working gradients' associated with interpolation along the cuts taken across the map.

2.5 Converting a set of unconnected points into a grid format

This problem is very similar to the one discussed in the previous section in that it is required to fit to the data a surface that is smooth and continuous and which passes through every point. Again, the surface can be sampled at the appropriate points so as to generate the required grid. Numerous algorithms exist for performing surface fitting and interpolation with data consisting of a set of unconnected points (see for example Ref 2). The method adopted here is based on the interpolation function devised by Shepard[3] and used in the Symap plotting package, which is available at many large computer installations, for producing contour maps. Basically, the method adopts a surface based on a weighted average of the values of local points, the weighting function adopted being proportional to the inverse of the distance squared but also taking account of the direction in which data points lie. A summary of the algorithm and how it is implemented is given in Appendix A. There are however two properties of the Symap algorithm that should be noted. Firstly, as the algorithm only uses a few nearby points the interpolation surface will be most reliable in those areas where the density of points is greatest. Secondly, the algorithm can provide values at points beyond the limits of the data set. For points very far from the data set the interpolated value tends to the average of the four nearest points in the data set.

2.6 The programs

The programs that have been developed can be classified into one of three categories depending upon the format of the input file. These programs, together with a description of their function and applications, are given in Appendices B to D. Appendix B lists those programs that handle contour files, Appendix C lists those that handle grid files and Appendix D lists two programs that have been added to the image processing library. A comprehensive list of the image processing programs and the image format are given in Ref 4 but it should be noted that there are two types of image files, one which will handle picture elements with intensity values in the range 0-255 (1 byte) and another which will handle picture elements within values in the range 0-65536 (2 bytes).

In several of the program descriptions reference is made to average tangents interpolation. This type of interpolation is described in Ref 1. For convenience, in the program descriptions contour data and data consisting of a series of unconnected points will often be referred to collectively as contour data.

3 DEGRADING THE SPATIAL RESOLUTION OF CONTOUR MAPS

3.1 Background

In many fields, *eg* geography, radio astronomy, meteorology, the spatial measurements of a variable are often presented in the form of contour maps. This is done either for clarity, or because the data is considered to contain insufficient detail or is in a form unsuitable for display as a grey tone image. Sometimes, as in the case of height information on an Ordnance Survey map, the measurements represent true values for the variable, *ie* the measurement process has not affected the values measured except for introducing statistical errors. For such data it is possible to calculate the results that would be obtained if the measurement process affected the values recorded, *eg* for terrain height information it would be possible to calculate the results that would be obtained from an airborne altimeter with a known footprint. However, often the data presented on a map has been affected by the measurement process. For remotely sensed data this arises because the finite beam width of the measurement sensor essentially 'averages' the variable over a certain area. Thus the measured data has a certain spatial resolution associated with it. If the shape of the measurement beam is known then it may be possible to calculate what the measurements would have been if the measurement beam had been larger. (It is not possible to do this for the case of a smaller beam.) This is important because it is often necessary to compare different variables each of which has a different spatial resolution. The comparison of these variables can be made considerably easier if the resolution of the different variables can be converted to a common value, *ie* to the poorest resolution. This in turn may allow further parameters to be derived by direct comparison of the variables, *eg* by ratioing.

The procedures outlined above are best performed on a digital computer. The contour maps must therefore be transformed into a format that is suitable for easy manipulation in a computer. The most appropriate format for this is that of an equispaced grid of points and that is the format that has been adopted for the work described here. (The exact file format adopted has been described in section 2.3.) The degradation of resolution can be performed by convolving a suitable 'filter' over the grid of points. A contour map can then be produced at a degraded resolution by using a contour threading routine. Numerous algorithms exist for contour threading and the one in use at the Remote Sensing Unit is described in Ref 5.

3.2 Basic theory

The variable displayed by the contour map ($S(x)$) is the result of the measurement beam ($B(x)$) 'averaging' the source data ($R(x)$), or $S(x)$ is the result of the convolution of the measurement beam and the variable being measured, *ie*

$$S(x) = R(x) * B(x)$$

where '*' denotes convolution.

It is desired to determine the variable $S'(x)$ that would be observed if a larger measurement beam $B'(x)$ were to be used, *ie*

$$S'(x) = B'(x) * R(x) \quad .$$

Now

$$S(v) = R(v)B(v)$$

where 'v' denotes the Fourier transform of a spatial quantity, denoted by 'x'.

Thus

$$S'(v) = \frac{B'(v)S(v)}{B(v)} \quad .$$

Put

$$T(v) = \frac{B'(v)}{B(v)}$$

whence

$$S'(x) = T(x) * S(x) \quad .$$

Thus the desired result can be obtained by convolving the measured data with $T(x)$

where $T(x) = \text{INV FT}\,\dfrac{B'(v)}{B(v)} \quad .$

The problem thus presents itself in two parts:

(1) determining $T(x)$;

(2) convolving $S(x)$ with $T(x)$.

The problem of determining $T(x)$ can either be undertaken analytically or numerically, but it does require knowledge of the shape of the measurement beam. Such information is normally stated in general terms, *eg* approximately a Gaussian of half power width 3.5 minutes of arc, or illustrated on the map by contours. It should be realised that for certain functions for $B(x)$ and $B'(x)$, $T(x)$ may not exist. This will occur if $T(v)$ becomes indeterminate at some v , *eg* if $B(v) = 0$. The condition for $T(x)$ to exist is that $B'(v)/B(v)$ be real and even, which greatly restricts the functions that can be adopted for $B(x)$ and $B'(x)$. (For further details the reader should consult one of the many books on Fourier analysis.) One particularly useful case for which $T(x)$ exists is when both $B(x)$ and $B'(x)$ are Gaussian. This is because it is often found that a Gaussian is a reasonable approximation to an experimentally determined beam shape and because it also has the advantage that it readily lends itself to analytic treatment. Obviously, in cases where the initial measurement process has not affected the data or where the effect of the measurement process can be neglected then $T(x) = B'(x)$.

The convolution process consists of multiplying $R(x)$ by $B(x)$ and integrating the product over all positions of the latter. In the two-dimensional case

$$S(a,b) = \iint R(x,y)B(x + a, y + b)\,dx\,dy \quad .$$

In the problem being considered here the implementation is discrete, thus

$$S(i,j) = \sum_x \sum_y R(x,y)B(x + i, y + j) \qquad x,y,i,j = 0,1,2, \text{ etc.}$$

In practice, the number of points associated with B is generally much smaller than with R so the summation is performed over i and j giving

$$S(x,y) = \sum_i \sum_j R(x,y)B(x + i, y + j) \quad .$$

If R is a matrix of order $(k \times \ell)$ and B is a matrix of order $(m \times n)$, then the amount of computation for $k\ell \gg mn$ is approximately proportional to $k\ell mn$. If the linear sampling of the variables represented by R and B is decreased by a factor f , then the level of computation involved in the convolution is reduced by a factor $1/f^4$. Thus, if the sampling is decreased by a factor 2 the corresponding computation is decreased by a factor of 16.

3.3 Example

Fig 1a shows a plot of data after a contour map has been digitised. The size of the measurement beam is illustrated by the circle in the top right hand corner. This circle represents an area of the full width at half maximum intensity of the measurement beam. As the paper from which the map was obtained gives no other details except that the map represents an area of the sky mapped at 53 μm it is not unreasonable to approximate the beam shape by a two-dimensional Gaussian, an assumption which is often made for simplicity at such wavelengths.

In Fig 1a the outermost contour does not appear on the original map. It is added because the shape of the contour map suggests that the variable tends to a base level away from the main peaks, and as is mentioned in Ref 1 such a contour is required to achieve a satisfactory operation of the interpolation algorithm in the outermost contour interval of the original data. For the map illustrated in Fig 1a the region enclosed by the rectangle was to be represented by a grid of points 280 × 320. The grid spacing was set by the requirement that it be no greater than half the linear size of the smallest feature in the map and that it should also be small enough to describe T(x) with sufficient accuracy. Fig 1b shows the data in Fig 1a after the data has been transformed into a grid format and then a contour threading routine used to reconstruct the original map. Comparison of the two maps in Fig 1 shows that the grid representation is generally a good approximation to the original. The most significant difference between the two occurs for the outermost contour of Fig 1b where part of it has a 'jagged' appearance. The reason for this is that because the values stored using the grid format differ by multiples of 0.1, slopes small in comparison to 0.1 appear 'stepped' rather than smooth as thus by a similar argument do contours. For this particular example the process of generating a contour map from the grid also contributes to the 'jagged' appearance for a similar reason because the data was compressed by 0.8 prior to the contour threading operation being performed. This data compression is related to converting the grid format into an image format, the latter being required for the contour threading program.

As an example, the original data is to be degraded in spatial resolution to 1.1, 1.2, 1.4, 1.6 and 2.0 that of the original data. For each of these cases a Gaussian beam shape is taken for the required beam. The derivation of T(x) for Gaussian measurement and required beams is given in Appendix E. T(x) was evaluated for a rectangular grid of points the size of which was determined by when T(x) decreased to less than 1% of its maximum value. The number of points in the grid representing T(x) thus increases the more it is desired to degrade the resolution of the original data. Fig 2 shows how the shape of the contour map changes as the resolution of the data is degraded. It can be seen that as the resolution is degraded the finer detail disappears and also the peak intensity decreases.

The above procedure is carried out using the following programs:

1	CONT.DIG	This is used to digitise the contour data from a published intensity map.
2	CONT.GRID	This converts the digitised contour data into a grid format.
3	GRID.GAUSS	This is used to produce the convolution grid for the case of Gaussian measurement and required beams. If the number of points in the grid exceeds about 250 then a grid spacing of twice, three times, or more of that of the grid produced in 2 is adopted. This is done in order to maintain computation involved in the convolution (step 5) at a reasonable level.
4	GRID.AVE	This is used if the grid produced in 3 has been made with a grid spacing greater than that of the grid produced in 2. The program is thus used to average the grid produced by 2 over 2 × 2, 3 × 3, etc points so that the grid spacing is the same as that of the convolution grid.
5	GRID.CONV	This convolves the grid produced by 4 (or 2) with that produced by 3.

6 GRID.IM This converts the grid produced by 5 into an image format.

7 IM.CONTOUR This program performs the contour threading operation on the image produced by 6. (See Ref 5.)

8 MAP.HPLOTR This plots the results of IM.CONTOUR. (See Ref 5.)

3.4 Further applications

The technique of transforming a digitised contour map into a rectangular grid of points can be used to generate contour levels at other than those of the original data. This operation can be performed by simply specifying the values of the contours that it is required to thread through the grid of points. Fig 3 shows Fig 1b and also the shape of that map when the contours are chosen at intermediate levels. This operation is performed as follows:

1 CONT.DIG This is used to digitise the contour map.

2 CONT.GRID This transforms the contour data into a grid format.

3 GRID.IM This converts the grid into an image format.

4 IM.CONTOUR This performs the contour threading operation.

5 MAP.HPLOTR This plots the results of IM.CONTOUR.

A contour map c:n also be displayed in the form of a grey tone image. Fig 4 shows the original data displayed as an image and the same data after it was transformed into a grid format and displayed as a grey tone image. These operations are performed as follows:

1 CONT.DIG This is used to digitise the contour map.

2 CONT.EDIT This is used to create a contour file that does not include the contour that was added to the original data.

3 CONT.IM This is used to create an image of the contour file produced by 2. The result of this operation is shown in Fig 4a.

4 CONT.GRID This transforms the data produced by 1 into a grid format.

5 GRID.IM This transforms the data produced by 4 into an image format.

6 CONT.EDIT This is used to extract from the file produced by 1 the outermost contour of the original map data, *ie* contour level 5.

7 CONT.GRID This is used to create a 'mask' whereby points inside the contour obtained by 6 are set to the value 5 and all other points are set to 0. The dimensions of the grid, the origin and the grid spacing must be the same as that of the grid produced by 4.

8 GRID.IM This transforms the grid produced by 7 into an image format, where all points equal to 5 are set to 255 and all others to 0. 152

9 IM.MERGE The image produced by 7 is used in conjunction with that produced by 5 to set all pixels in the image produced by 5 to zero if they lie beyond the outermost contour of the original data. The result of this operation is shown in Fig 4b.

Note: In this particular example, the task of setting the data to zero for points beyond the outermost contour of the original data could be performed more easily by the use of GRID.BASE on the grid produced by step 4. The required image could then have been obtained by using GRID.IM on the grid produced by GRID.BASE.

In Fig 4b certain linear features can be seen in the interpolated data between the contours. The difference in values across these features is typically less than a tenth of the associated contour interval. These artifacts arise from the operation of the interpolation algorithm used by CONT.GRID and are a shortcoming of the algorithm when the grid spacing is considerably less than the 'mean' contour spacing. The algorithm does not suffer from this shortcoming when the grid spacing is considerably greater than, or is of the same order as, the 'mean' contour separation. The artifacts can be 'smeared out' if required without significantly changing the reliability of the data near the original contours, by the use of a small (3 × 3 or 5 × 5) smoothing filter.

The conversion of a contour map into a grid format enables variables represented by two or more contour maps to be directly compared, *eg* by displaying each as an image in a different primary colour on an interactive image processing machine such as the IDP 3000. (A summary of the facilities available on this machine can be found in Ref 6 and a much more detailed and comprehensive description in Ref 7.) Alternatively, it may be possible to derive other quantities, *eg* two contour maps of the same region at different wavelengths could be used to derive a colour temperature map of that region.

4 APPLICATIONS TO TERRAIN DATA

4.1 General remarks

In remote sensing an increasingly important field of investigation involves developing techniques for using non-satellite derived data in conjunction with satellite imagery. Here techniques are described for using terrain height information, obtained from maps, in conjunction with imagery from Landsat satellites. (Landsat is a series of NASA remote sensing satellites which contain multi-spectral scanners which are used to record the intensity of reflected radiation from the ground in four spectral bands in the visible and near infrared.)

The height information on a map may be provided in some or all of the following forms:

(1) height contours (these generally occur at preferred values, *eg* 0 ft, 50 ft, 100 ft, etc);

(2) spot heights;

(3) contours defining areas of constant height (these border areas of constant height such as lakes and as such are not restricted to preferred values, as generally are height contours, but rather can occur at any height);

(4) break lines, indicating features such as cliffs.

On a map virtually all the height information is conveyed by the height contours. Little information is conveyed by spot heights, as will be demonstrated in section 4.2.1, unless there is a very large number of such points. Since the work here is primarily concerned with Landsat imagery neglecting areas of constant height is not particularly serious since such areas generally indicate water which can easily be discerned on Landsat imagery. As the height information is to be transformed into a grid format it is not possible to represent vertical or near vertical slopes, indicated by break lines, in such a format. Further, break lines are not consistent with the assumption that has been adopted for this work that the surface represented is smooth and continuous. Thus, when generating a height matrix from map data only the information conveyed by the height contours is used.

In order to illustrate the techniques that are described, part of a map has been digitised. Fig 5 shows a 5 km square area of Hampshire after digitisation of the contours and spot heights. The data is taken from a 1:25000 scale Ordnance Survey map (sheet SU54), the lower left corner having easting 453000 and northing 140000 metres.

4.2 Producing a terrain height matrix

4.2.1 Generation of a height matrix from contours

When converting a contour map into a grid format it is necessary to define a grid spacing. It should be remembered that the smaller the grid spacing the more accurately the contours can be reconstructed from the height matrix and in any case the spacing should be no greater than half the scale size of the smallest feature which it is required to preserve in the transformation to a grid format. One measure of the distribution in the scale size of features is given by the distribution of the linear separation between adjacent contours. The latter can be obtained by taking two orthogonal sets of closely spaced cuts across the contour map and then along these cuts determining the separation between adjacent intersection points with contours. It is necessary to take two sets of cuts since it may be that some contours run parallel to one of the sets of cuts and thus would be ignored by that set. Contour spacings missed by one set of cuts are therefore obtained by the second set. The method can however lead to an overestimate by $\sqrt{2}$ of the minimum separation between two contours (or loops in the same contour). The value of $\sqrt{2}$ is a worst case value and arises when two contours are both 45° to both sets of cuts. When the two contours are perpendicular to one of the sets of cuts then the minimum separation is correctly estimated. It is not the intention to deal here in detail with the problems associated with this technique (*eg* how the distribution associated with each of the sets of cuts varies when the orientation of the cuts is changed) nor is it necessary to do so providing it is assumed that in order to represent features of a given size (determined by the method described) the grid spacing should be no greater than a third of that size $(\approx \frac{1}{2} \cdot 1/\sqrt{2})$.

The procedure outlined above has been applied to the map shown in Fig 5. The number of cuts taken across the map is 500 in both the x and y directions and the grid spacing is 10 metres. Fig 6 shows a plot of the percentage of contour separations less than or equal to a particular value, against contour separation. It shows that only 20% of the separations are less than 100 metres and that none are less than 10 metres.

The map illustrated in Fig 5 has been converted into a grid 500×500, with a 10 metre grid spacing. The accuracy of the height matrix can be ascertained by comparison with the spot heights for the area, although it should be realised that beyond the outermost contours the interpolation technique is not very reliable. (This is discussed in more detail later in this section.) Values can be derived from the height matrix at each of the spot height locations by using the 16 surrounding matrix values (see Ref 1 for details of the interpolation algorithm). A comparison of the two sets of values is given in Table 1. As is only to be expected, there are no differences greater than the contour interval of 25 ft. The mean difference between the two sets of values is only 0.04 ft, a value that is probably rather fortuitous for a sample of only 21 points. The standard deviation of the differences is 7.1 ft.

In the previous paragraph it was stated that the height values are not particularly reliable beyond the outermost contour. This problem is associated with attempting to interpolate outside the data set. (A detailed discussion of this is given in Ref 1.) As there is not a satisfactory solution to this problem it is best not to use such data. Thus, later in this Report when illustrating techniques using the data in Fig 5, only the central 4 km square area will be considered.

In section 3.5 it was mentioned that certain undesirable artifacts become evident with the algorithm used to convert contour maps into a grid format when the grid spacing is considerably less than the 'mean' contour separation. These artifacts can if required be smoothed out by the use of an appropriate filter. The smoothing of the height data results in the original contour data not being accurately reproducible from the height matrix. In the case of contour data obtained from Ordnance Survey maps this is not important because there are considerable uncertainties in the position of contours. As a very crude guide the vertical accuracy of contours can be taken as being approximately equivalent to a quarter of a contour interval, however for a precise definition of contour accuracy Ref 8 should be consulted.

4.2.2 <u>Generation of a height matrix from spot heights</u>

The spot heights can also be used to generate a height matrix by adopting the algorithm described in section 2.5 and Appendix A. The quality of the height matrix thus produced can be ascertained by generating contours from the matrix and comparing them with those shown in Fig 5. The contour map that is obtained is shown in Fig 7 and it can be seen that the contours are substantially different from those of Fig 5, which is only to be expected when so few points are available. If more points were available then a better agreement would be expected. Obviously, unless there is a model for the data being fitted, then with so few points no technique can be expected to reproduce the map illustrated in Fig 5. However, the comparison of the two maps does illustrate how little information is conveyed by the spot heights in comparison with the contours.

The operation to produce the contour map illustrated in Fig 7 was carried out using the following programs.

1	CONT.DIG	This is used to digitise the spot heights.
2	CONT.GRIDX	This transforms the data into a grid format.
3	GRID.IM	This converts the above grid into an image format.

4 IM.CONTOUR This program performs a contour threading operation on the image produced in 3.

5 MAP.HPLOTR This plots the results of IM.CONTOUR.

4.3 Quantities derived from the height matrix

Using the height matrix it is possible to derive the magnitude and direction of the gradient of the terrain. Henceforth in this discussion the magnitude of the gradient will be referred to as slope and the direction as aspect. The slope and the aspect are obtained from the x and y direction gradients m_x and m_y that are obtained from a least squares fit of a two-dimensional second order polynomial over a small $n \times n$ matrix of points. This least squares fitting is considerably simplified by the fact that the points are an equispaced grid and the point of interest is the one at the centre. This least squares fitting is described in more detail in Appendix F, where it can be seen that the x and y direction slopes are easily evaluated. Because of the possible undesirable effects of the interpolation algorithm when the grid spacing is considerably less than the mean contour spacing, the derivation of the slope and aspect under such a condition are best made using a 5×5 or a 7×7 matrix, unless subsequent averaging or smoothing is to be performed. The height data can also be smoothed using this least squares fitting technique.

The slope and aspect are obtained from the x and y direction gradients m_x and m_y as follows:

$$\text{slope} = \tan^{-1}\left(\sqrt{m_x^2 + m_y^2}\right)$$

$$\text{aspect} = \tan^{-1}\left(\frac{m_x}{m_y}\right) \qquad \text{slope} \neq 0$$

$$= \text{undefined} \qquad \text{slope} = 0 \ .$$

The direct illumination from the Sun at a point on the ground can be obtained from the slope and aspect. It is given by the equation

$$I_s = \left[\cos(\alpha - \gamma) \cos \beta \sin \delta + \sin \beta \cos \delta\right] I_0$$

where α is the Sun azimuth, β the Sun elevation, γ the aspect, δ the slope and I_0 the solar illumination on a unit area perpendicular to the direction of the illumination; α and γ are measured from north. It must be emphasised that this equation only refers to direct illumination and neglects totally the not insignificant contribution that will arise from backscattered radiation. A more useful quantity for investigating the variation in brightness due to the variation in the gradient of the terrain is given by

$$R_d = \frac{\sin \delta \cos(\alpha - \gamma)}{\tan \beta} + \cos \delta \ .$$

This quantity, which henceforth will be referred to as the variation in direct solar illumination, is unity for flat terrain, negative for terrain that is in shadow and greater than unity for terrain inclined towards the Sun direction. Also, the variation in R_d over a given area increases as the Sun elevation decreases. Neither of the equations takes into account the possibility that some parts of the terrain may be shaded from direct illumination by others. Thus, these equations are only valid for all values of the terrain aspect and Sun azimuth angles when the Sun elevation angle is greater than the steepest slope, *ie* $\beta > \delta$.

4.4 Use of terrain in conjunction with imagery

4.4.1 Displaying terrain parameters as grey tone images

Height, slope and the variation in direct solar illumination present no problems in being displayed as grey tone images. Here, the lower values are assigned to lower pixel intensities and higher values to higher pixel intensities. (Display units at the Remote Sensing Unit limit the pixel intensities that can be displayed to the range, 0-255.) Aspect can also be displayed as a grey tone image although the interpretation of such an image is rather more difficult. This is because

(1) the aspect can vary between 0^{o} and 360^{o}, but 0^{o} and 360^{o} represent the same direction;

(2) the aspect for flat terrain is undefined;

(3) the value of the aspect becomes more unreliable as the slope becomes smaller.

One convention that can be adopted for displaying aspect and the one that is adopted here, although there is some ambiguity in the interpretation of the minimum and maximum permissible intensities, is to assign an aspect value of 0^{o} to an image intensity of zero, 360^{o} to the maximum image intensity, and flat terrain to a zero image intensity. Other conventions for displaying aspect can of course be envisaged.

When considering the height matrix as an image it must be remembered that the values are associated with the centre of the pixels and do not represent the average value over a pixel (the linear dimension of which equals the grid spacing - only square pixels are considered here). In the situation where the terrain is changing slowly compared with the size of a pixel then the estimate will be a reasonable approximation to the average value over a pixel. However in the situation where the terrain is changing rapidly compared with the pixel size the height estimate will not necessarily be a good approximation to the mean value over a pixel. In such circumstances the height must first be evaluated using a grid spacing for which the terrain is changing slowly, and then a suitable average taken to obtain an estimate of the mean value for the required pixel size. This technique can also be used to evaluate the mean solar illumination over a pixel but it cannot be applied to the slope and aspect. For these quantities it is necessary to evaluate the mean x and y direction gradients in the manner described and then to use these gradients to derive the required mean slope and aspect.

The central 4 km square area of the map shown in Fig 5 has been converted to a grid format with a grid spacing of 10 metres. This was then used to derive the height, slope, aspect and variation in direct solar illumination corresponding to 70 metre square pixels. A pixel size of 70 metres square has been chosen because it is the approximate size of Landsat pixels. The values associated with the pixels have been obtained by using the 49 values of the matrix covering each pixel. The grey tone image representations of the four terrain parameters are shown in Fig 8. For Fig 8d the Sun is assumed to be at the top of the image at an elevation of 12^{o}. (The maximum slope for any pixel is 9^{o}.)

4.4.2 Applications

The four terrain parameters can be considered as four additional image planes to be used in conjunction with the four image planes associated with the four spectral bands of Landsat images. All eight of these image planes can therefore be analysed together. Thus in classification for example, it would be possible to consider only areas exceeding a specified height or only flat areas.

Classification with Landsat imagery is complicated by the fact that although two areas may have the same cover type and thus in a given spectral band would be expected to have the same intensity, in practice they will have different intensities if the terrain is such that the areas receive different illumination. These differences could be reduced by using terrain data to correct the intensity of Landsat pixels to account for the variation in direct solar illumination. Thus pixels associated with slopes lying away from the Sun would be brightened and those with slopes facing towards the Sun would be darkened. This technique should improve the reliability of multispectral classifications, particularly for scenes taken at low sun elevation angles when the problem is most serious.

In order to use Landsat imagery and terrain data together in the ways described it is obviously necessary for them to be transformed into the same co-ordinate system. It is probably more useful for interpretation if the Landsat data is transformed to the same map projection as the terrain data. A problem that can arise when combining data from different sources in the ways described is one of registration. Misregistration could result in a height estimate associated with a particular pixel not being appropriate to that pixel but rather to another pixel nearby. The errors arising from misregistration will be expected to increase the greater the degree of misregistration and to be more significant in areas where the terrain is changing most rapidly, especially when it is changing on a scale that is similar to or smaller than the pixel size.

5 CONCLUSIONS

This Report has described some techniques that can be used to process data that is in the form of either a contour map or a set of unconnected points. Two applications of this work have been described which help to illustrate the advantages of transforming data into a form that can easily be handled and manipulated by a computer. The importance of this work is that it has made it considerably easier to compare data from different sources and in different formats and thus hopefully will encourage greater use of ground truth data with satellite imagery. One application that should become more important in the future is the use of digital terrain data with satellite imagery of land surfaces, especially if both the quality and availability of suitable topographic data bases continues to improve.

Appendix A

THE SYMAP INTERPOLATION ALGORITHM AND PROGRAM CONT.GRIDX

This algorithm has been devised by Shepard[3] and adopts a surface based on a weighted average of the values at nearby data points. The value at a reference point P on the surface (co-ordinates x, y) is given by

$$F(P) = \frac{\sum_{i=1}^{L} w_i z_i}{\sum_{i=1}^{L} w_i} \quad \text{if P is not coincident with a data point}$$

$$= z_i \quad \text{if P is coincident with a data point}$$

where L is the number of nearby data points (total number of points = N), co-ordinates x_i, y_i and associated value z_i .

The weighting function adopted is inversely proportional to the distance squared and is given by

$$w_i = \frac{1}{d_i^2} = \frac{1}{(x - x_i)^2 + (y - y_i)^2} .$$

It has been found empirically that lower powers of the distance produce a relatively flat surface with very steep gradients near the data points so as to achieve the proper values at the data points. Conversely, higher powers of the distance produce a surface that is relatively flat near all the data points but with very steep gradients over small intervals between data points.

To prevent the amount of computation required becoming excessive, and because the nearby data points are predominant in the interpolation, a maximum number of 10 points is chosen. Initially, a search radius r is established according to the overall density of the data points, such that on average seven points are included in a circle of radius r . This search radius is very important because, as described below, it influences the selection of data points and is itself often used in the interpolation algorithm.

As program CONT.GRIDX is required to generate a rectangular grid of points with sides parallel to the x and y axes, generally using a data set covering a rectangular area, also with sides parallel to the x and y axes, the initial search radius is obtained from

$$r = \sqrt{\frac{7(y_{max} - y_{min})(x_{max} - x_{min})}{\pi N}}$$

where x_{min}, x_{max}, y_{min} and y_{max} are the minimum and maximum x and y co-ordinates of the data points. The author has found this method of determining the search radius produces satisfactory results if, on the scale of the search area, the density of data points does not vary too greatly over the data set. However, if there are large variations in the density of data points, as occurs in Fig 7, then the author has found that it is better arbitrarily set r to a large value so that at most positions within the rectangle defined by x_{min}, x_{max}, y_{min} and y_{max} interpolated values are derived using the maximum of 10 data points. This latter method was used to generate the grid from which the contours shown in Fig 7 were derived, the initial search radius being obtained from

$$r = \sqrt{(y_{max} - y_{min})(x_{max} - x_{min})} \quad .$$

In order to select and weigh data points that are to be used in the interpolation process, a collection C' of data points near P and a final search radius r' are defined as follows. First let $C_p = \{D_i | d_i \leqslant r\}$

and $n(C_p)$ = the number of elements in C_p.

Subscript i_j is defined such that $0 \leqslant d_{i_1} \leqslant d_{i_2} \leqslant \ldots \leqslant d_{i_n}$.

Now define

$$C_p^n = \left\{D_{i_1}, D_{i_2}, \ldots, D_{i_n}\right\} \qquad (n \leqslant N)$$

and

$$r'\left(C_p^n\right) = \min\left\{d_{i_j} | D_{i_j} \notin C_p^n\right\} = d_{i_{n+1}} \quad .$$

Finally let

$$C_p' = \begin{cases} C_p^4 & \text{if } 0 \leqslant n(C_p) \leqslant 4 \\ C_p & \text{if } 4 < n(C_p) \leqslant 10 \\ C_p^{10} & \text{if } 10 < n(C_p) \end{cases}$$

and

$$r_p' = \begin{cases} r'\left(C_p^4\right) & \text{if } n(C) \leqslant 4 \\ r & \text{if } 4 < n(C) \leqslant 10 \\ r'\left(C_p^{10}\right) & \text{if } 10 < n(C) \ . \end{cases}$$

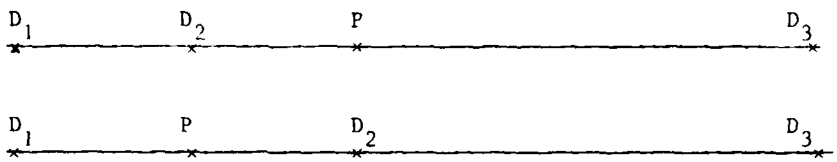
For a reference point P new weighting factors $S_i = S(d_i)$ are defined from the following

$$S(d) = \begin{cases} \dfrac{1}{d} & \text{if } 0 < d \leqslant \dfrac{r'}{3} \\ \dfrac{27}{4r'}\left(\dfrac{d}{r'} - 1\right)^2 & \text{if } \dfrac{r'}{3} < d \leqslant r' \\ 0 & \text{if } r' < d \ . \end{cases}$$

In program CONT.GRIDX the points are evaluated a line at a time from the top of the grid, *ie* as a series of x direction grid lines in descending values of y. In order to select the data points required they are ranked in order of increasing distance from the point under consideration, and the points selected according to the above criteria.

The ranking is performed using a bubble sort which involves taking each pair of data points in turn and determining whether or not they are in the correct order; if they are not the points are interchanged. After passing through the data set, if any pair of points has been interchanged then the process is repeated. Only when no pair of points have been interchanged is the data ranked correctly. This method of sorting is reasonably efficient for the way points are to be evaluated as going along a line changes the ranking of the data points only slowly. Only when a transition is made from the end of one line to the start of the next is a large reordering required.

In the interpolation, direction as well as distance is taken into account. The need for a directional term can be seen from the following two examples which do not yield identical values at P .



This difference arises because an intervening data point should be expected to screen the effect of a more distant one. A directional weighting term for each data point D_i near P is thus defined as follows:

$$t_i = \frac{S_i + \sum_{D_j \in C'} S_j \left[1 - \cos(D_i P D_j)\right]}{\sum_{D_j \in C'} S_j}$$

$$\cos(D_i P D_j) = \frac{(x - x_i)(x - x_j) + (y - y_i)(y - y_j)}{d_i d_j}$$

and the new weighting function is given by $w_i = (S_i)^2(1 - t_i)$.

Again, with this new weighting term points near P are more important than more distant ones.

For P sufficiently close to some point (d_i is small) w_i will vary as d_i^2 and the interpolated surface will have zero gradient at every D_i . To correct this, increments are added to the function values at nearby data points. This is accomplished by first calculating constants A_i and B_i , representing the desired slopes in the x and y directions at each data point D_i . A_i and B_i are weighted averages of divided differences of z about D_i . Set $C''_i = C'_{D_i} - \{D_i\}$, then

$$A_i = \frac{\displaystyle\sum_{D_j \in C''_i} w_j \frac{(z_j - z_i)(x_j - x_i)}{\left(d[D_j, D_i]^2\right)}}{\displaystyle\sum_{D_j \in C''_i} w_j}$$

$$B_i = \frac{\displaystyle\sum_{D_j \in C''_i} w_j \frac{(z_j - z_i)(y_j - y_i)}{\left(d[D_j, D_i]\right)^2}}{\displaystyle\sum_{D_j \in C''_i} w_j} \quad .$$

A further parameter v is defined such that

$$v = \frac{\alpha[\max\{z_i\} - \min\{z_i\}]}{\left[\max\left(A_i^2 + B_i^2\right)\right]^{\frac{1}{2}}} \quad .$$

This parameter limits the maximum effect of the slope terms on the final interpolated value at P. The choice of α is somewhat arbitrary but Ref 3 suggests that a reasonable criterion is to limit the effect of slope to one-half the contour interval, *eg* if it is required that there be 10 contour levels over the range of values of z_i then $\alpha = 0.05$. (It should be remembered that one of the main applications of this algorithm is the generation of contour maps.) Although this concept is not strictly appropriate for program CONT.GRIDX, as the grid of points generated will not necessarily be used to produce a contour map, it is nevertheless used as a basis for establishing a criterion for α. It is assumed that a reasonable value for a contour interval is given by the standard deviation (σ_z) on the mean value of z_i. Thus

$$\alpha = \frac{\sigma_z}{2[\max\{z_i\} - \min\{z_i\}]}$$

where $$\sigma_z = \frac{1}{N}\sqrt{N \sum_{i=1}^{N} z_i^2 - \left(\sum_{i=1}^{N} z_i\right)^2} \quad .$$

To include the effect of slope in deriving an interpolated value at P, an increment Δz_i is computed for each $D_i \in C'_P$ as a function of P such that

$$\Delta z_i = \left[A_i(x - x_i) + B_i(y - y_i)\right]\left[\frac{v}{v + d_i}\right] \quad .$$

Thus the interpolation function is given by

127

$$F(P) = \begin{cases} \dfrac{\displaystyle\sum_{D_i \in C'_i} w_i(z_i + \Delta z_i)}{\displaystyle\sum_{D_i \in C'_i} w_i} & \text{if } d \neq 0 \text{ for all } D_i \in C' \\[2ex] z_i & \text{if } d = 0 \text{ for some } D_i \in C'. \end{cases}$$

Computer rounding and truncation can cause considerable inaccuracy when P is very close to D_i . Since $\lim_{P \to D_i} F(P) = z_i$ the problem is overcome by establishing a distance ε , depending upon the computer's precision, and defining $F(P) = z_i$ within ε of D_i . If several data points occur within this neighbourhood then the average of their values is taken.

Appendix B

CONTOUR PROCESSING PROGRAMS

CONT.DIG Output file: Contour format

This program enables a map to be digitised using a Tektronix 4014 digitising tablet. The method that has been adopted for digitising contours is for the cursor to be moved along a contour and for a key to be depressed on the Tektronix control unit each time it is required to digitise a position on the contour. This mode has been adopted, rather than one where points are continually digitised without the intervention of the user, because as a result of a hardware problem the system has proved unreliable if the computer controls the frequency (either in time or as a function of the distance the cursor is moved) with which points are digitised. Nevertheless, the method adopted has the advantage that for those inexperienced in digitising contours it is likely to yield more accurate results although it does have the drawback that it can be considerably more time consuming. Prior to commencing to digitise contours, the user is required to enter a line of text describing the contents of the file and to digitise up to four fiducial points, such as the four corners of the map. Before digitisation of a contour is started the user is required to enter the contour type code and the value to be associated with the contour. During digitisation, care must be exercised to ensure that for closed contours in particular no part of the contour is digitised twice. After digitisation of a contour is completed the program performs two checks before writing the data to the output file so as to ensure that the data is in the required format:

(1) It ensures that no two adjacent points on a contour have the same co-ordinates.

(2) For closed contours it ensures that the first and last points have the same x and y co-ordinates.

CONT.EDIT

Input file: Contour format Output file: Contour format

This program enables individual contours to be deleted from a file. This program can for example be used to produce a file containing only the outermost contour of a set of concentric closed contours.

CONT.SUM

Input file: Contour format Output file: Program specific

This program provides a summary of the contents of a contour file. Two levels of summary are available: low or high. The low level summary provides the text associated with the file and provides some simple statistics on the three types of contour and of all the contours. The statistics given are the number of contours, the average, minimum and maximum number of points in a contour, and the total number of points in all the contours. The high level summary provides the same information as that given by the low level summary, but in addition provides a listing of the headers of each of the contours. This program is particularly useful for determining the contents of unknown files and for ascertaining the number of points associated with a map, a parameter which reflects the amount of computation involved in processing the map data.

CONT.EX

Input file: Contour format Output file: Contour format

This program extracts all the contour data contained within a specified rectangle, the sides of which must be parallel to the axes of the co-ordinate system in which the contour data is defined. Closed contours are redefined as open if they do not lie wholly within the specified rectangle. The primary application of this program arises when it is required to perform subsequent processing on only part of a map.

CONT.MERGE

Input files: Contour format Output file: Contour format

This program merges two or more contour files into one, thus permitting a map to be digitised in two or more sessions. (Individual contours must not be split between two or more files because if this is done the parts are treated as separate contours and are not recombined.) The text of the first file can be retained for that of the output file or new text given, likewise the fiducial points of the first file can be retained or new points specified. The text and fiducial points of all subsequent files are not incorporated into the output file, and all contours in the output file are numbered consecutively from 1.

CONT.THIN

Input file: Contour format Output file: Contour format

This program reduces the number of points defining a map. This is achieved by discarding points which lie closer than a given distance (specified by the user) from the preceding point. This program is used when the accuracy with which contours have been digitised is not required for subsequent processing and a lower accuracy is acceptable, thus reducing the computation required in the subsequent processing of the data.

CONT.ROTATE

Input file: Contour format Output file: Contour format

This program performs a Cartesian co-ordinate rotation and/or origin displacement. A rotation will be required if, for example, the sides of the grid that it is required to produce with program CONT.GRID are not parallel to the x and y axes of the contour data.

CONT.ORIGIN

Input file: Contour format Output file: Contour format

This program provides the x and y limits of the data contained in a contour file and it can if required perform an origin displacement. The program can also be used to change the text and redefine the fiducial points of a file. The primary applications of this program are to determine the area covered by the contour data, so that the origin of the co-ordinate system can be redefined, *eg* to a position defined by the minimum x and y values of the data, and to enable the fiducial points to be used to define the four corners of a rectangle which just wholly contains the contour data.

CONT.SCALE

Input file: Contour format Output file: Contour format

This program changes the x and y scaling factors of contour data so that if required the positional co-ordinates can be defined in more appropriate units.

CONT.PLOT

Input file: contour format

This program plots the contents of a contour file on a graph plotter. Three plotting modes can be selected.

(1) Contours only are plotted.

(2) Spot heights only are plotted.

(3) Both contours and spot heights are plotted.

In addition, three modes are available for plotting contours.

(1) Points only are plotted.

(2) Points are joined by a straight line. Points are not plotted.

(3) Points are joined by a curve derived using average tangents interpolation. Points are not plotted.

CONT.IM

Input file: Contour format Output file: Image format

This program converts contour data into contours in an image format so that a contour map may be displayed on a VDU or superimposed upon images (using program IM.MERGE - see Appendix D). In addition to defining the area of the map that is to be converted into an image format the user has to specify the pixel size (which are assumed to be square) and the width of the contours in pixels. Average tangents interpolation is used to fit a curve through the data points.

CONT.GRID

Input file: Contour format Output file(s): Grid format

This program converts a contour map into an equally spaced rectangular grid of points. Two contour fitting algorithms are available: linear for use when the contour points are very close together, and average tangents for use when the points are too far apart for linear interpolation to be appropriate. The program can also generate gradients, this data being written to two separate output files, one containing x direction gradients and the other y direction gradients. These gradients represent the gradients along the cuts that are taken across the contour map when the height matrix is generated.

Two versions of the program are available: one which allows an output grid of 257 × 257 and the other which allows an output grid of 513 × 513. These limits are purely arbitrary and are not fundamental to the processing performed.

CONT.GRIDX

Input file: Contour format Output file: Grid format

This program converts a series of unconnected data points into an equally spaced grid of points using the interpolation algorithm described in Appendix A. This type of transformation is a necessary step towards generating a contour map representation of such data. The maximum size permitted for the output grid is arbitrarily set to 513 × 513.

Appendix C

GRID PROCESSING PROGRAMS

GRID.CHECK

Input files: Grid and contour format — Output file: Program specific

This program calculates values at specified points within a grid. Two output options are available. Option HIGH gives the positional co-ordinates and data values at the reference points, the associated values derived from the grid, the differences between the two sets of values, and the statistics of these differences. Option LOW only provides the statistics of the differences.

GRID.SLOPE

Input file: Grid format — Output file(s): Grid format

This program evaluates the x and y direction gradients, the slope and aspect (these are defined in section 4.3) and smoothed values of data in a grid file. These quantities are evaluated by fitting a two-dimensional second order polynomial to a 3×3, 5×5 or 7×7 matrix of points. The user is given the option of generating only some of the five parameters, each of which is written to a separate output file.

GRID.EXPAND

Input file: Grid format — Output file: Grid format

This program expands or contracts the number of points in a grid. The scaling factors for the x and y directions can be chosen independently. Average tangents interpolation is employed.

One particularly important use of this program is associated with the display on a VDU of data in a grid format. Typically the number of pixels required to fill the screen is 512×512, a number that may be considerably larger than is contained in the grid. This program can thus be used to increase by interpolation the number of points in the grid so that it is closer to the number required to fill the screen. The facility for different expansion factors for the x and y directions means that the aspect ratio (width/height) of the grid can be altered if necessary so that the final image appears correctly proportioned if the aspect ratio of the display unit is not unity.

GRID.AVE

Input file: Grid format — Output file: Grid format

This program reduces the number of points in a grid file by evaluating the mean value in each $n \times n$ matrix of points, where n is an integer specified by the user.

GRID.BASE

Input file: Grid format — Output file: Grid format

This program sets all values in a grid file less than a user specified threshold to another user specified value.

GRID.RATIO

Input files: Grid format Output file: Grid format

This program divides one grid by another if both grids are of the same dimensions and have the same spacing between points. The division is not performed for points where either the numerator or the denominator are less than user specified values. Points in the output grid for which a division is not performed are assigned a user specified value.

GRID.SECTION

Input file: Grid format Output file: Program specific

This program enables any section across a grid to be taken. The user has to specify the co-ordinates of the two end points of the cut, which can lie inside or outside the grid, and the number of points required along the section. Average tangents interpolation is used.

GRID.SECT.PLOT

Input file: Output file of GRID.SECTION

This program plots the data contained in the output file of program GRID.SECTION. Average tangents interpolation is used to draw a curve through the data points.

GRID.GRAD

Input files: Grid format Output files: Grid format

This program calculates slope and aspect from the x and y direction gradients.

GRID.GAUSS

Output file: Grid format

This program generates a grid to be used in a convolution when it is required to degrade the spatial resolution of data in a grid format. The program is only applicable when it can be assumed that Gaussian point spread functions can be associated with the data in question both before and after the spatial resolution is degraded.

GRID.CONV

Input files: Grid format Output file: Grid format

This program convolves one grid with another. The maximum size for the convolution grid (*ie* the one that is to be moved over the other) is 100×100.

GRID.SOL

Input file: Grid format Output file: Grid format

This program evaluates the variation in the direct solar illumination from a terrain height matrix (see section 4.3) for user specified sun elevation and azimuth angles. The slope and aspect of the terrain are evaluated by fitting a two-dimensional second order polynomial to a 3×3, 5×5 or 7×7 matrix of points. The variation in the illumination is expressed as a percentage of the illumination on flat terrain.

GRID.IM

Input file: Grid format Output file: Image format

This program converts a grid file into an image file. The program determines the minimum and maximum values of the input data so as to assist the user in specifying two values to be associated with the input variable and two corresponding intensities for the output image file. It is assumed that a linear relationship is to exist between the input data and the output image intensity. The intensity range of the output image is 0-255 and calculated intensities that lie outside this range are set to the appropriate limit.

Appendix D

IMAGE PROCESSING PROGRAMS

IM.GRID

Input file: Image format — Output file: Grid format

This program converts an image file into a grid file, thus allowing the grid processing programs listed in Appendix C to be employed. The maximum size that is permitted for the input image is 5000 × 5000.

IM.MERGE

Input files: Image format — Output file: Image format

This program combines two images by replacing a pixel in one image by the corresponding pixel in the second image. This operation is performed if the pixel value in the second image is equal to or greater than a user specified value, or alternatively if it is less than or equal to the user specified value. The purpose of this program is to:

(1) superimpose contour data on an image;

(2) to mask out data that is not required in an image by using an image containing a mask (see section 3.4);

(3) to merge two images each containing masks.

Note: Both of these programs are restricted to image files which only handle pixel values in the range 0-255.

Appendix E

DERIVATION OF THE CONVOLUTION FUNCTION WHEN THE POINT SPREAD FUNCTIONS OF THE MEASUREMENT AND REQUIRED BEAMS ARE GAUSSIAN

It is required to determine $T(\bar{x})$, which is given by

$$T(\bar{x}) = \begin{matrix}\text{INV}\\ \text{FT}\end{matrix} \left[\frac{\tilde{B}'(\bar{u})}{\tilde{B}(\bar{u})}\right]$$

where $\tilde{B}(\bar{u})$ and $\tilde{B}'(\bar{u})$ denote respectively the Fourier transforms of the measurement beam $B(\bar{x})$ and the required beam $B'(\bar{x})$. (Here $\bar{u}$ is used to denote the frequency domain and $\bar{x}$ the spatial domain.) The shapes of the two beams are given by

$$B(x,y) = Ae^{-\alpha^2 x^2} e^{-\beta^2 y^2}$$

$$B'(x,y) = Be^{-a^2 x'^2} e^{-b^2 y'^2}$$

where $\alpha^2 = \dfrac{4 \log_e 2}{\theta_\alpha^2}$, $\beta^2 = \dfrac{4 \log_e 2}{\theta_\beta^2}$, $a^2 = \dfrac{4 \log_e 2}{\theta_a^2}$, $b^2 = \dfrac{4 \log_e 2}{\theta_b^2}$.

θ_α, θ_β, θ_a and θ_b denote half-power widths as illustrated in Fig 9, and A and B are the peak amplitudes. (As will become evident in the final expression for $T(\bar{x})$ the measurement beam must be wholly contained within the required beam.) In Fig 9 θ is the orientation of the $\bar{x}'$ co-ordinate system with respect to the $\bar{x}$ system, *ie*

$$x' = y \sin\theta + x \cos\theta$$

$$y' = y \cos\theta - x \sin\theta$$

or

$$\bar{x}' = R\bar{x}$$

where $R = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$.

Now

$$B(u,v) = \int_{-\infty}^{+\infty} B(x,y)e^{2\pi iux}e^{2\pi ivy}dxdy$$

and since

$$e^{-\alpha^2 x^2} \xrightarrow{FT} \frac{\sqrt{\pi}}{\alpha} e^{(-\pi u/\alpha)^2}$$

$$B(u,v) = \frac{A\pi}{\alpha\beta} e^{-(\pi u/\alpha)^2} e^{-(\pi v/\beta)^2} .$$

Similarly

$$B'(u',v') = \frac{B\pi}{ab} e^{-(\pi u'/a)^2} e^{-(\pi v'/b)^2} \quad .$$

If there is a simple rotation in the space of $\bar{x}$, the Fourier transform is invariant if the same rotation is made in the space of $\bar{u}$ (see Ref 9), hence

$$B'(u,v) = \frac{B\pi}{ab} e^{-(\pi/a)^2(u\cos\theta+v\sin\theta)^2} e^{-(\pi/b)^2(-u\sin\theta+v\cos\theta)^2} \quad .$$

Hence

$$T(u,v) = \frac{B\alpha\beta}{\Lambda ab} e^{-(\pi u/p)^2} e^{-(\pi v/q)^2} e^{-(\pi/r)^2 uv}$$

where
$$\frac{1}{p^2} = \frac{1}{b^2a^2}(b^2\cos^2\theta + a^2\sin^2\theta) - \frac{1}{\alpha^2}$$

$$\frac{1}{q^2} = \frac{1}{b^2a^2}(b^2\sin^2\theta + a^2\cos^2\theta) - \frac{1}{\beta^2}$$

$$\frac{1}{r^2} = \frac{1}{b^2a^2}\sin 2\theta(b^2 - a^2) \quad .$$

Now

$$T(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} T(u,v)e^{-2\pi iux}e^{-2\pi ivy}dudv \quad .$$

Integrating with respect to v first requires the following integral

$$I = \int_{-\infty}^{+\infty} e^{-(\pi v/q)^2} e^{-(\pi/r)^2 uv} e^{-2\pi ivy} dv \quad .$$

This integral can be written in the form

$$e^{g^2}\int_{-\infty}^{+\infty} e^{-(fv+g)^2}dv = e^{g^2} \cdot \frac{\sqrt{\pi}}{f}$$

where $f = \pi/q$

and $2fg = 2\pi iy + (\pi^2/r^2)u$

hence

$$g^2 = \frac{q^2}{4}\left(2iy + \frac{\pi}{r^2}u\right)^2 \quad .$$

Thus

$$T(x,y) = \frac{B\alpha\beta}{Aab}\frac{q}{\sqrt{\pi}} e^{-q^2y^2} \int_{-\infty}^{+\infty} e^{-(\pi u/p)^2} e^{(\pi^2q^2u^2/4r^4)} e^{i u\pi q^2 u/r^2} e^{-2\pi i u x} du \quad .$$

Again the integral can be written in the form

$$e^{t^2} \int_{-\infty}^{+\infty} e^{-(su+t)^2} du = e^{t^2} \frac{\sqrt{\pi}}{s}$$

where $s = \frac{\pi}{2pr^2}(4r^4 - q^2p^2)^{\frac{1}{2}}$

and $t^2 = p^2 \frac{(2r^2x - yq^2)^2}{(q^2p - 4r^4)}$.

Thus

$$T(x,y) = \frac{B\alpha\beta}{Aab} \frac{2pqr^2}{\pi\sqrt{4r^4 - q^2p^2}} e^{\left[\frac{p^2(2r^2x-yq^2)^2}{(q^2p^2-4r^4)}\right]} e^{-q^2y^2} \quad .$$

Under certain conditions it is possible that $1/p^2$, $1/q^2$ and $1/r^2$ can be zero, thus the above equation is rewritten in the following final form

$$T(x,y) = \frac{B\theta_a\theta_b}{A\theta_\alpha\theta_\beta} \frac{1}{\pi\sqrt{\frac{1}{p^2q^2} - \frac{1}{4r^4}}} \exp -\left[\frac{\frac{x^2}{q^2} + \frac{y^2}{p^2} - \frac{xy}{r^2}}{\left(\frac{1}{p^2q^2} - \frac{1}{4r^4}\right)}\right]$$

where $\frac{1}{p^2} = \frac{1}{4\log_e 2}\left(\theta_a^2 \cos^2\theta + \theta_b^2 \sin^2\theta - \theta_\alpha^2\right)$,

$$\frac{1}{q^2} = \frac{1}{4\log_e 2}\left(\theta_a^2 \sin^2\theta + \theta_b^2 \cos^2\theta - \theta_\beta^2\right)$$

and $\frac{1}{r^2} = \frac{\sin 2\theta}{4\log_e 2}\left(\theta_a^2 - \theta_b^2\right)$.

Appendix F

LEAST SQUARES FITTING TO A RECTANGULAR GRID OF POINTS

For deriving the slope in the x and y directions at a point in a grid it is required to fit a two-dimensional second order polynomial to an $n \times n$ matrix of points centred on the point of interest (for this n is required to be an odd integer), *ie* it is required to fit to the n^2 points the surface

$$z = ax^2 + by^2 + cx + dy + exy + f \quad . \tag{1}$$

For convenience it is assumed that the origin of the Cartesian co-ordinate system lies at the point of interest and that the axes lie parallel to the sides of the grid. It is also assumed in the derivation given here that the grid spacing is unity. If this is not the true grid spacing it is only necessary to divide the derived gradients by the actual spacing. From equation (1) the value of the function and gradients at the point of interest are given by

$$z = f$$

$$\frac{dz}{dx} = c$$

$$\frac{dz}{dy} = d \quad .$$

In order to perform the least squares fit it is required to minimise the quantity

$$\varepsilon = \sum_{i=1}^{n^2} (z_i - F_i)^2 \quad . \tag{2}$$

where $F_i = ax_i^2 + by_i^2 + cx_i + dy_i + ex_iy_i + f$

and x_i, y_i and z_i are the co-ordinates of the data points. (Henceforth, the limits on summations will be omitted as they are always those of equation (2).)

The coefficients of the polynomial surface are obtained by differentiating equation (2) with respect to each of the coefficients in turn, *ie*

$$\frac{d\varepsilon}{da} = 0 = -2\sum x_i^2(z_i - F_i) \tag{3}$$

$$\frac{d\varepsilon}{db} = 0 = -2\sum y_i^2(z_i - F_i) \tag{4}$$

$$\frac{d\varepsilon}{dc} = 0 = -2\sum x_i(z_i - F_i) \tag{5}$$

$$\frac{d\varepsilon}{dd} = 0 = -2\sum y_i(z_i - F_i) \tag{6}$$

$$\frac{d\varepsilon}{de} = 0 = -2\sum x_i y_i(z_i - F_i) \tag{7}$$

$$\frac{d\varepsilon}{df} = 0 = -2\sum (z_i - F_i) \ . \tag{8}$$

Because of symmetry, the only non-zero terms in the above six equations of the form $\sum x_i^m y_i^n$ (where m and n are integers greater than or equal to zero) are those involving even powers of x_i and y_i. Thus the only non-zero terms that exist are

$$\sum x_i^4 \ , \quad \sum y_i^4 \ , \quad \sum x_i^2 \ , \quad \sum y_i^2 \ , \quad \sum x_i^2 y_i \ , \quad \sum 1$$

$$\sum x_i^2 z_i \ , \quad \sum y_i^2 z_i \ , \quad \sum x_i z_i \ , \quad \sum y_i z_i \ , \quad \sum x_i y_i z_i \ , \quad \sum z_i \ .$$

Equations (3) to (8) thus become

$$0 = \sum x_i^2 z_i - a\sum x_i^4 - b\sum x_i^2 y_i^2 - f\sum x_i^2 \tag{9}$$

$$0 = \sum y_i^2 z_i - a\sum x_i^2 y_i^2 - b\sum y_i^4 - f\sum y_i^2 \tag{10}$$

$$0 = \sum x_i z_i - c\sum x_i^2 \tag{11}$$

$$0 = \sum y_i z_i - d\sum y_i^2 \tag{12}$$

$$0 = \sum x_i y_i z_i - e\sum x_i^2 y_i^2 \tag{13}$$

$$0 = \sum z_i - a\sum x_i^2 - b\sum y_i^2 - f\sum 1 \ . \tag{14}$$

The x and y direction gradients, given by c and d respectively, are easily obtained from equations (11) and (12) viz

$$\frac{dz}{dx} = c = \left(\sum x_i z_i\right) \Big/ \left(\sum x_i^2\right) \tag{15}$$

$$\frac{dz}{dy} = d = \left(\sum y_i z_i\right) \Big/ \left(\sum y_i^2\right) \quad . \tag{16}$$

The value of z at the point of interest is given by f and is obtained by solving equations (9), (10) and (14), which gives

$$f = \frac{\left(\sum x_i^2\right)\left(\sum x_i^2 z_i + \sum y_i^2 z_i\right) - \left(\sum z_i\right)\left(\sum x_i^4 + \sum x_i^2 y_i^2\right)}{2\left(\sum x_i^2\right)^2 - n^2\left(\sum x_i^4 + \sum x_i^2 y_i^2\right)} \quad . \tag{17}$$

The equations to determine c , d and f are given in the following table for grids of 3 × 3, 5 × 5 and 7 × 7 points.

Quantity / Grid size	f	c	d
3 × 3	$\dfrac{5 \sum z_i - 3 \sum \left(x_i^2 + y_i^2\right) z_i}{9}$	$\left(\sum x_i z_i\right) \big/ 6$	$\left(\sum y_i z_i\right) \big/ 6$
5 × 5	$\dfrac{27 \sum z_i - 5 \sum \left(x_i^2 + y_i^2\right) z_i}{175}$	$\left(\sum x_i z_i\right) \big/ 50$	$\left(\sum y_i z_i\right) \big/ 50$
7 × 7	$\dfrac{11 \sum z_i - \sum \left(x_i^2 + y_i^2\right) z_i}{147}$	$\left(\sum x_i z_i\right) \big/ 196$	$\left(\sum y_i z_i\right) \big/ 196$

Table 1

HEIGHT MATRIX COMPARED WITH SPOT HEIGHTS FOR FIG 5

Spot height (ft)	Grid height (ft)	Difference (ft)
436.0	433.7	2.3
343.0	346.4	-3.4
365.0	359.3	5.7
313.0	320.6	-7.6
353.0	358.2	-5.2
281.0	294.2	-13.2
335.0	330.7	4.3
318.0	321.8	-3.8
602.0	600.9	1.1
394.0	400.4	-6.4
379.0	378.1	0.9
367.0	366.4	0.6
442.0	428.6	13.4
489.0	481.7	7.3
414.0	419.6	-5.6
453.0	436.4	16.6
381.0	381.8	-0.8
316.0	320.9	-4.9
377.0	376.5	0.5
571.0	566.9	4.1
329.0	335.7	-6.7

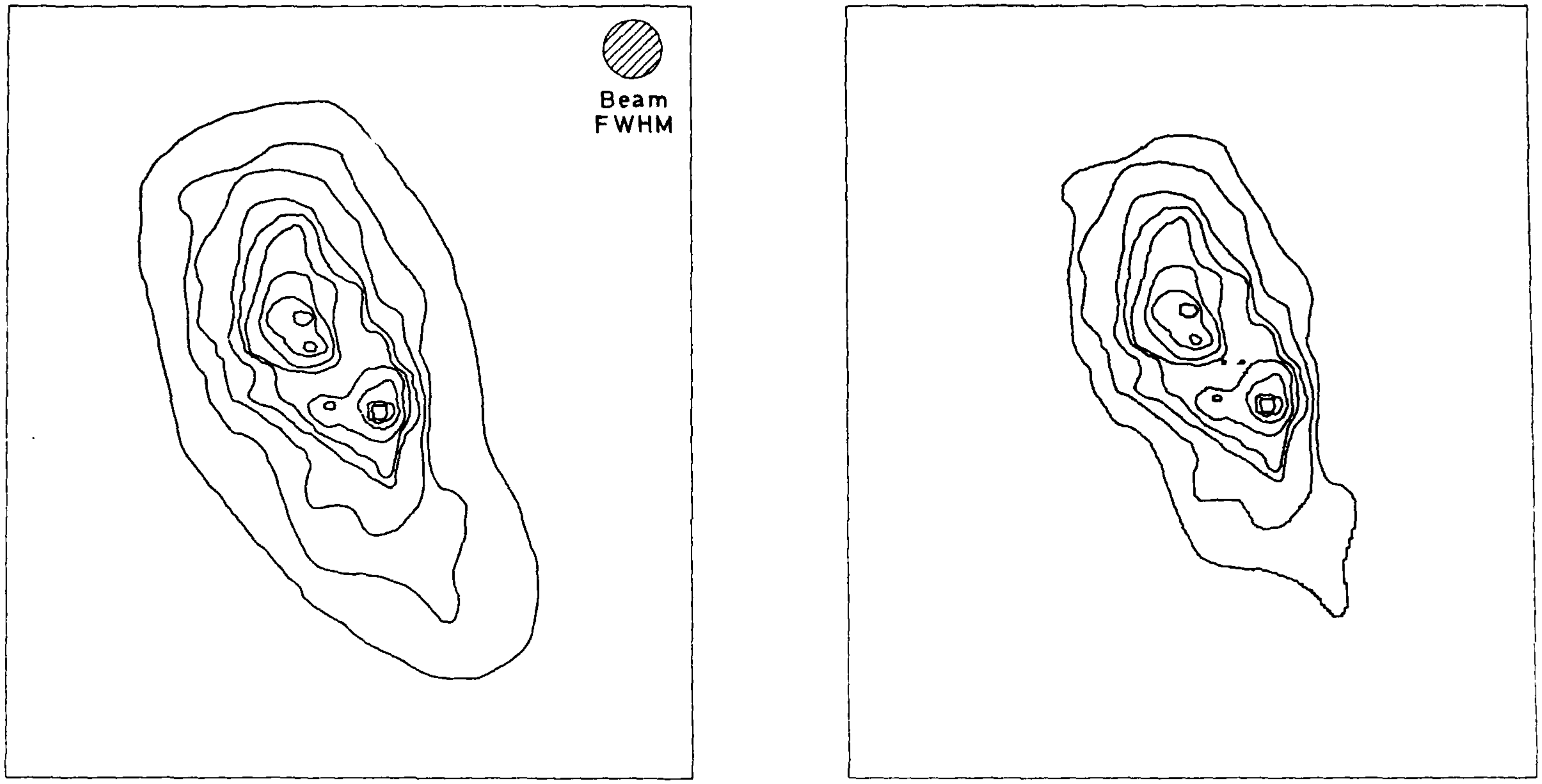
Mean difference: -0.04 ft

Standard deviation of the difference: 7.1 ft

REFERENCES

No.	Author	Title, etc
1	P.A. Roberts	Computer based technique for converting a contour map into an equispaced grid of points. RAE Technical Report 80110 (1980)
2	J.G. Hayes	Data-fitting algorithms available, in preparation, and in prospect, for the NAG library. NPL Report DNACS 5/78, September 1978
3	D. Shepard	A two-dimensional interpolation function for irregularly spaced data. Proc. of Association of Computing Machinery Meeting 1968
4	G.J. Davison	Image processing software (Issue 1). RAE Technical Memorandum Space 280 (1980)
5	A.H. Benny	A technique for line extraction from Landsat MSS satellite data with some applications of the technique. RAE Technical Report 81010 (1981)
6	D.M. Balston	IDP 3000 - An image analysis system for earth resource surveys. Systems technology, No.28, April 1978
7	-	Image data processor - Plessey IDP 3000 User Guide TP 3332, September 1979
8	J.B. Harley	Ordnance Survey Maps, a descriptive manual. Southampton Ordnance Survey, pp 159-168 (1975)
9	J. Arsac	Fourier transforms and the theory of distributions. Prentice Hall Inc, pp 106-110 (1966)

a b

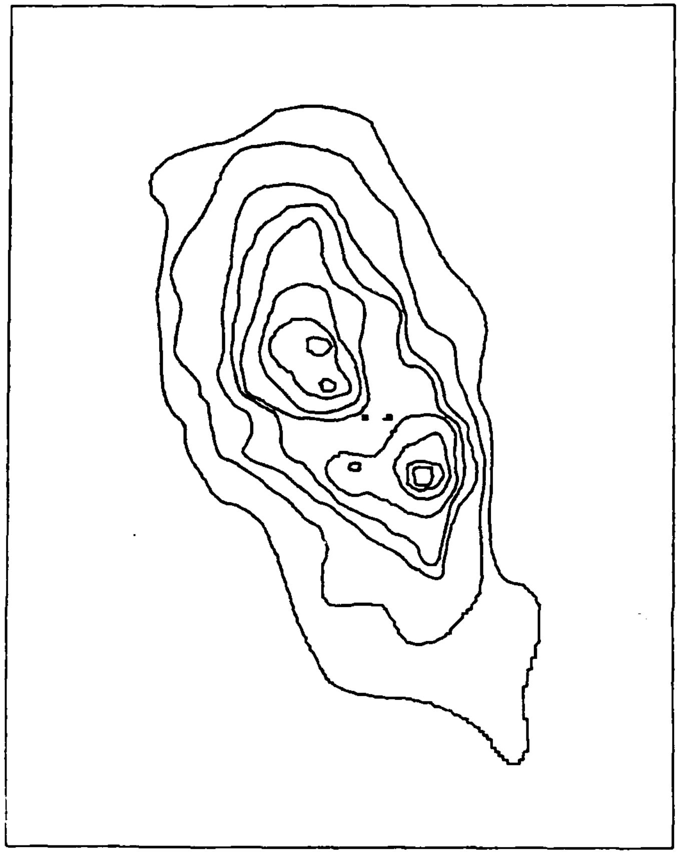


Before transformation   After transformation

Fig 1a&b Comparison of a contour map before and after the data has been transformed into an equispaced grid of points (contour levels starting with the outermost contour are: Fig 1a. 0, 5, 10, 15, 18, 21, 24, 27 and 30. Fig 1b: 5, 10, 15, 18, 21, 24, 27 and 30)

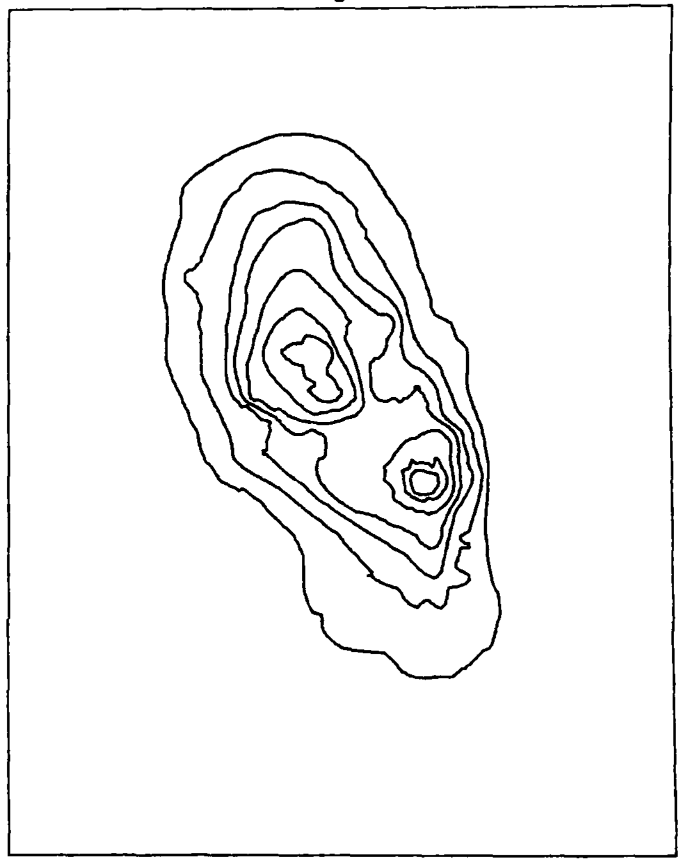
Fig 2 Example of the change in a contour map when the spatial resolution is degraded (contour levels starting with the outermost contour are 5, 10, 15, 18, 21, 24, 27 and 30. R is the factor by which the resolution is degraded)

a



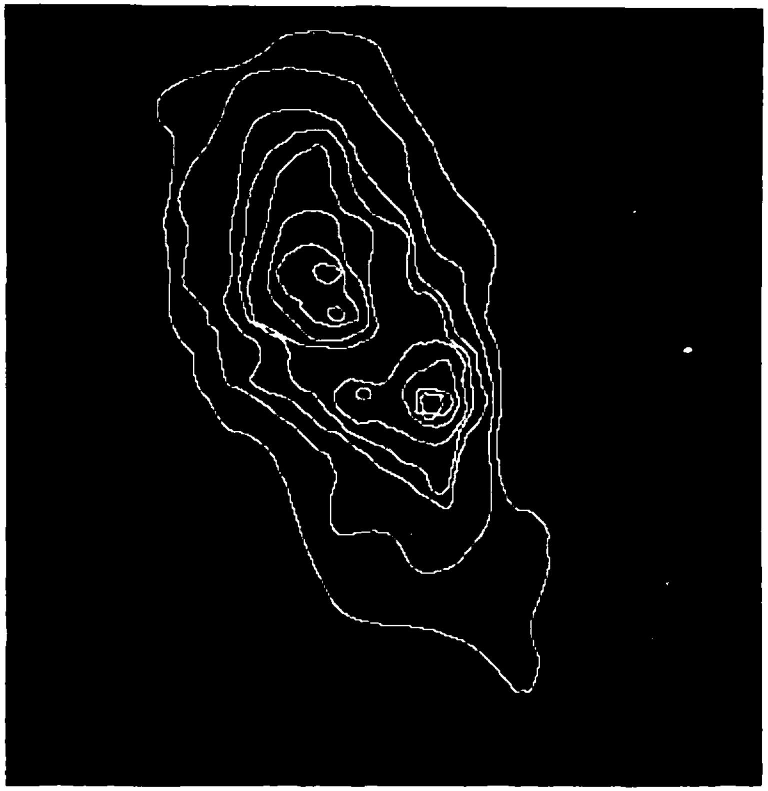
Original contours
(contour levels starting with the outermost contour
are 5,10,15,18,21,24,27 and 30)

b

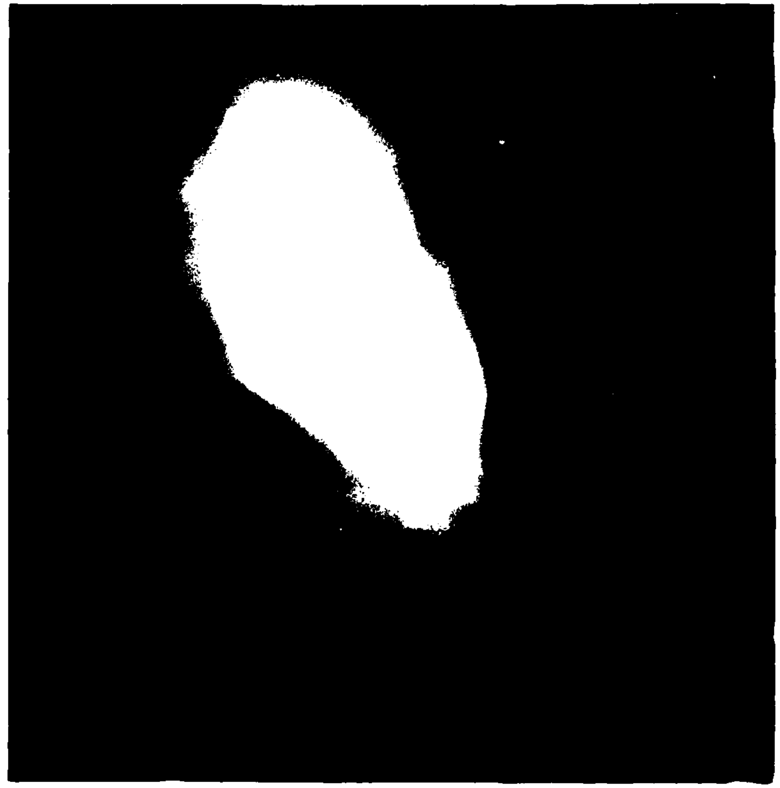


Intermediate contours
(contour levels starting with the outermost contour
are 7.5,12.5,16.5,19.5,22.5,25.5 and 28.5)

Fig 3a&b Example of contours generated at intermediate values



Map displayed as a series of contours (contour levels starting with the outer most contour are 5, 10, 15, 18, 21, 24, 27 and 30)



Map data transformed into a grid of points (darkest pixels = 0, lightest pixels 31.5)

Fig 4 A contour map displayed in two image formats

Fig 5

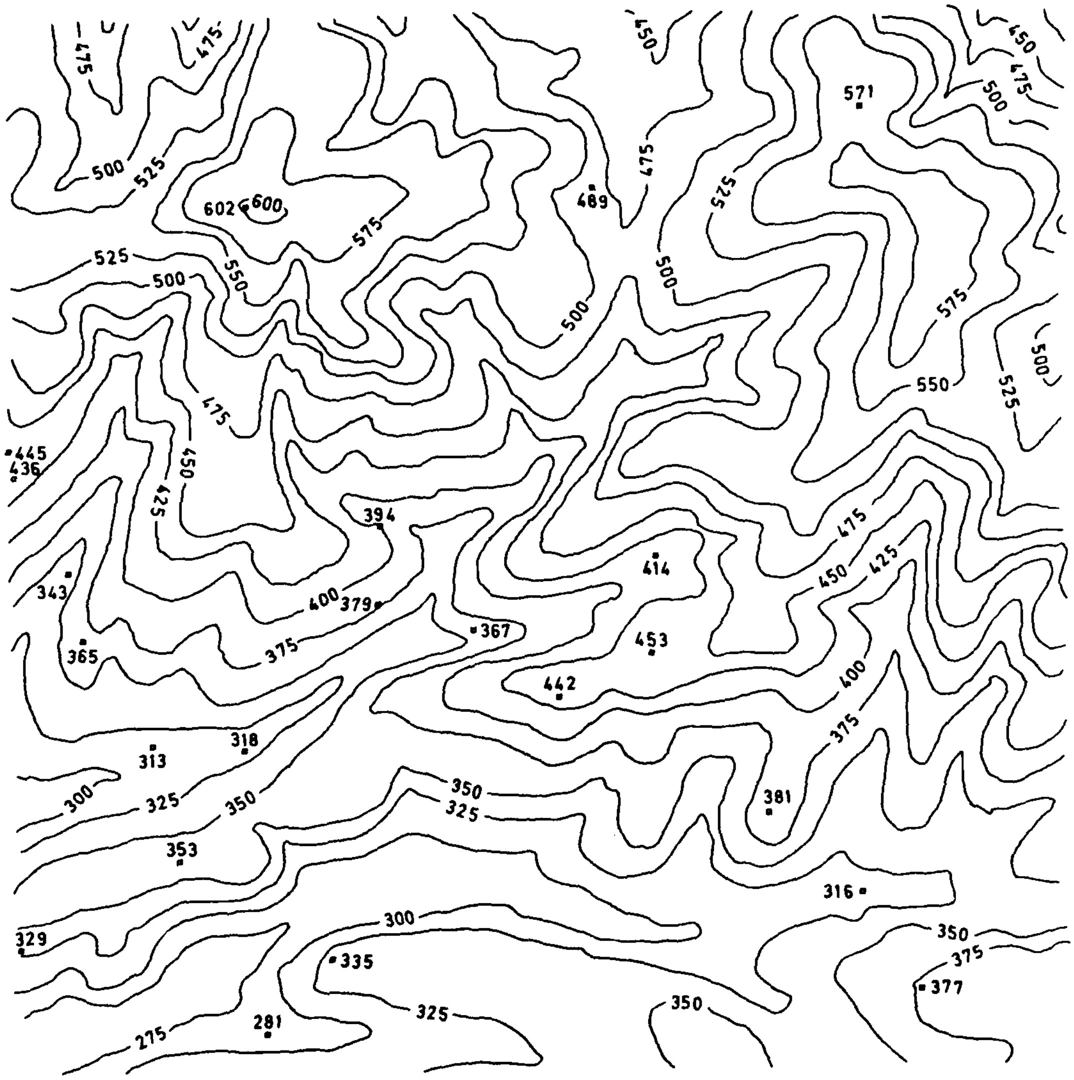


Fig 5 A 5 km square of ordnance survey map SU54 after digitisation of contours and spot heights (Note: all heights are in feet)

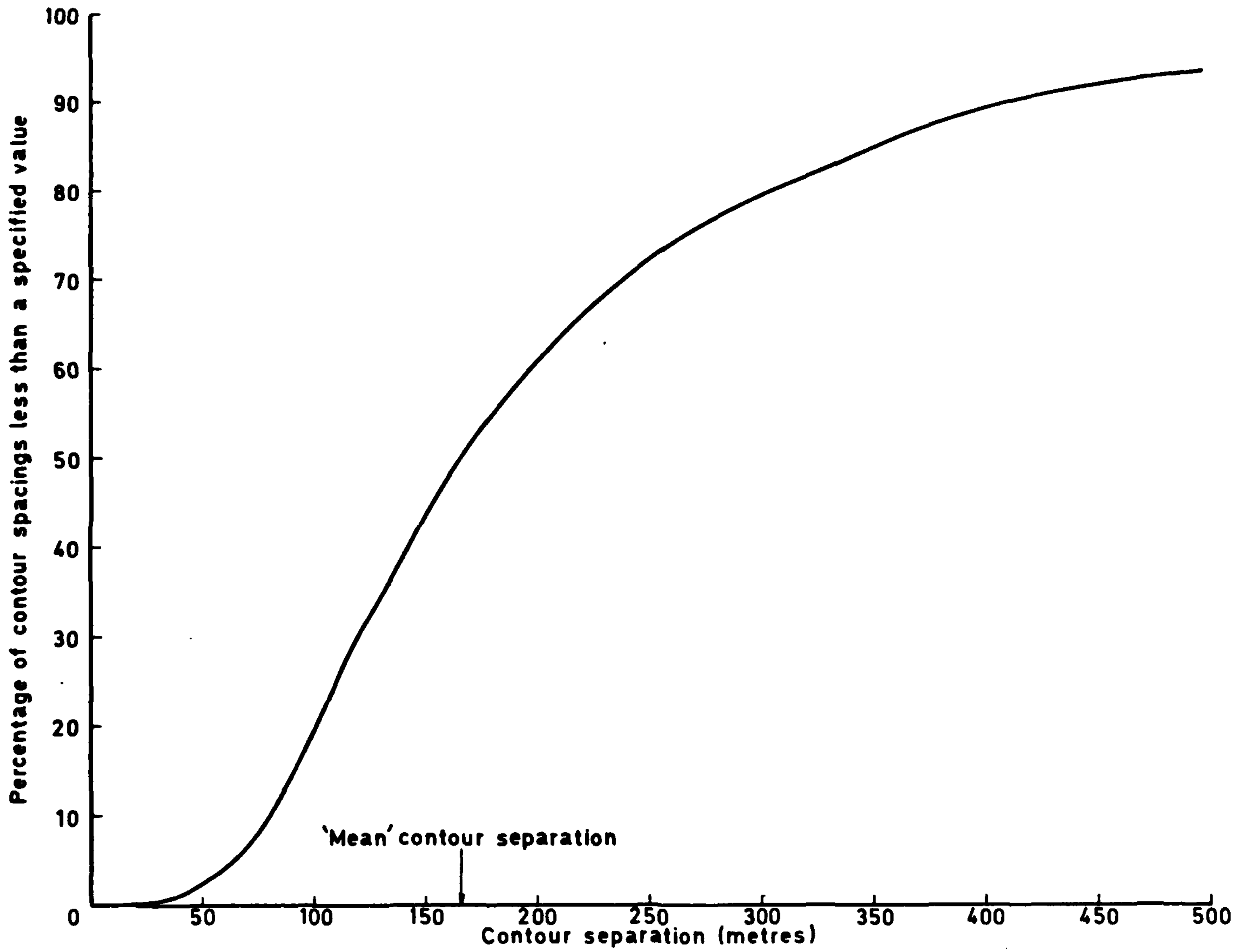


Fig 6 Plot of the percentage of contour spacings less than a specified value against contour separation for the map shown in Fig 5

Fig 7

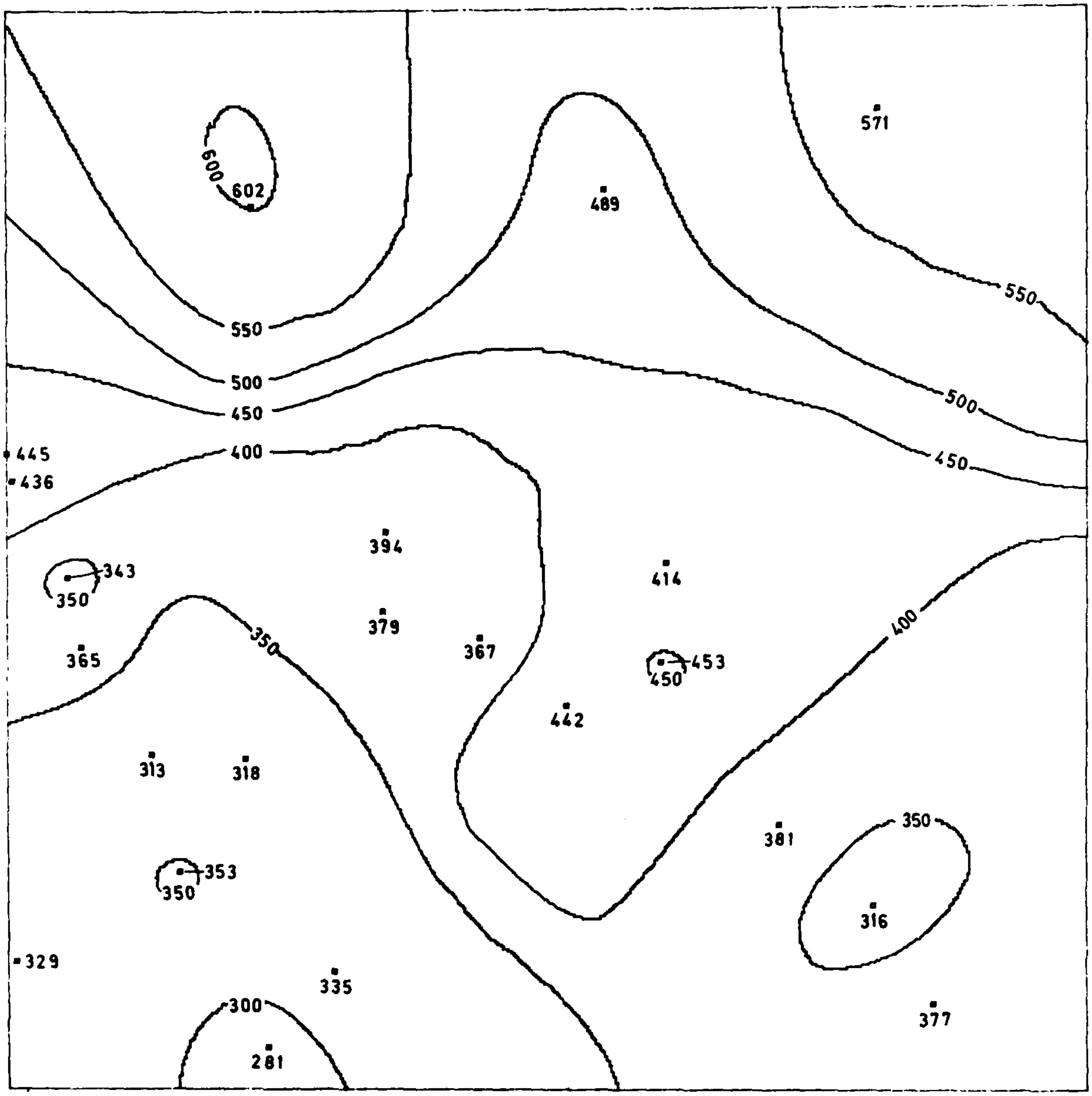
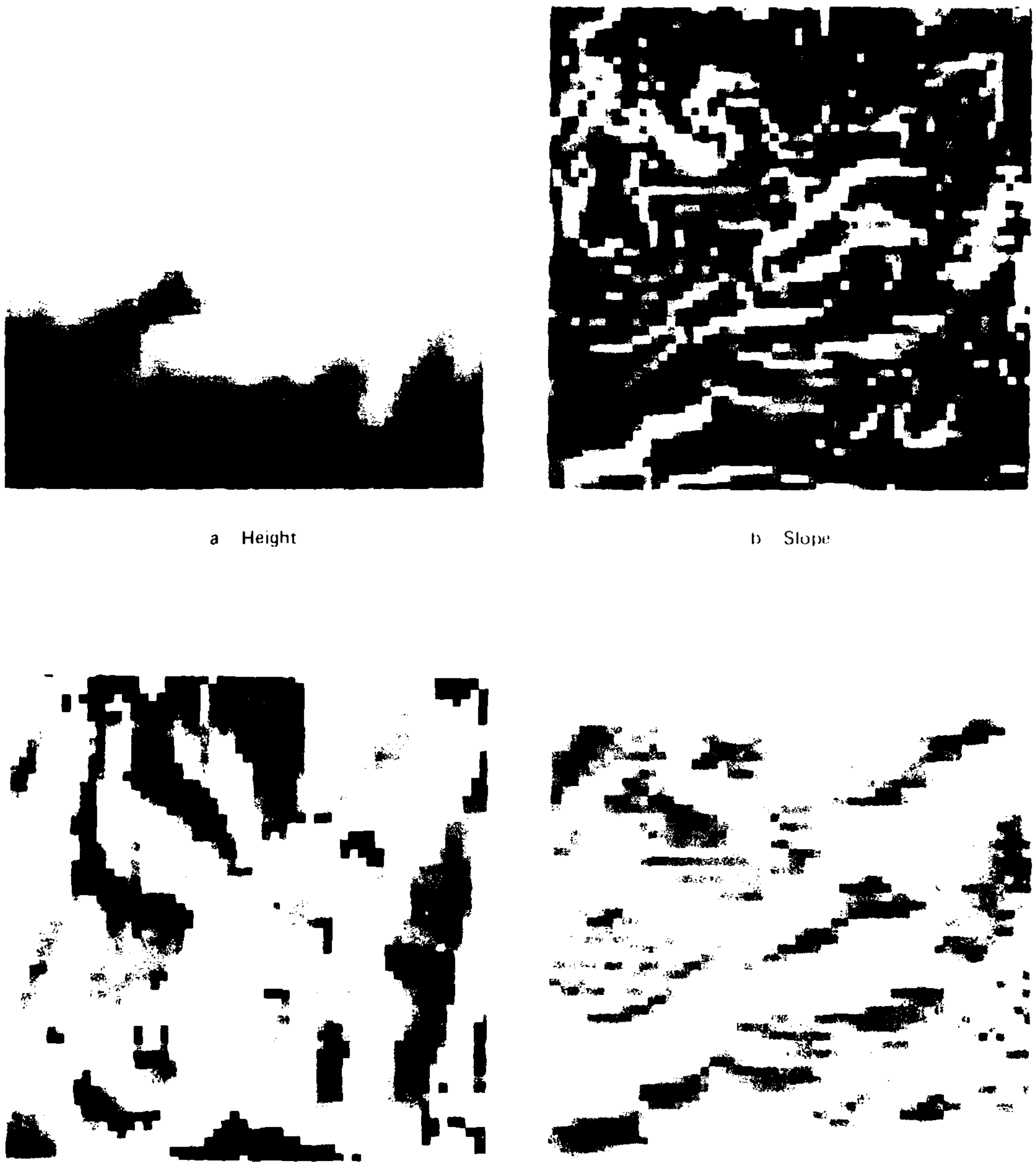


Fig 7 Contour map generated from spot heights (Note: all heights are in feet)

TR 81152



a Height

b Slope

c Aspect

d Variation in direct solar illumination

Fig 8a-d Grey tone image representations of the four terrain parameters for the central 4 km square of the test area. The pixel size is 70 m

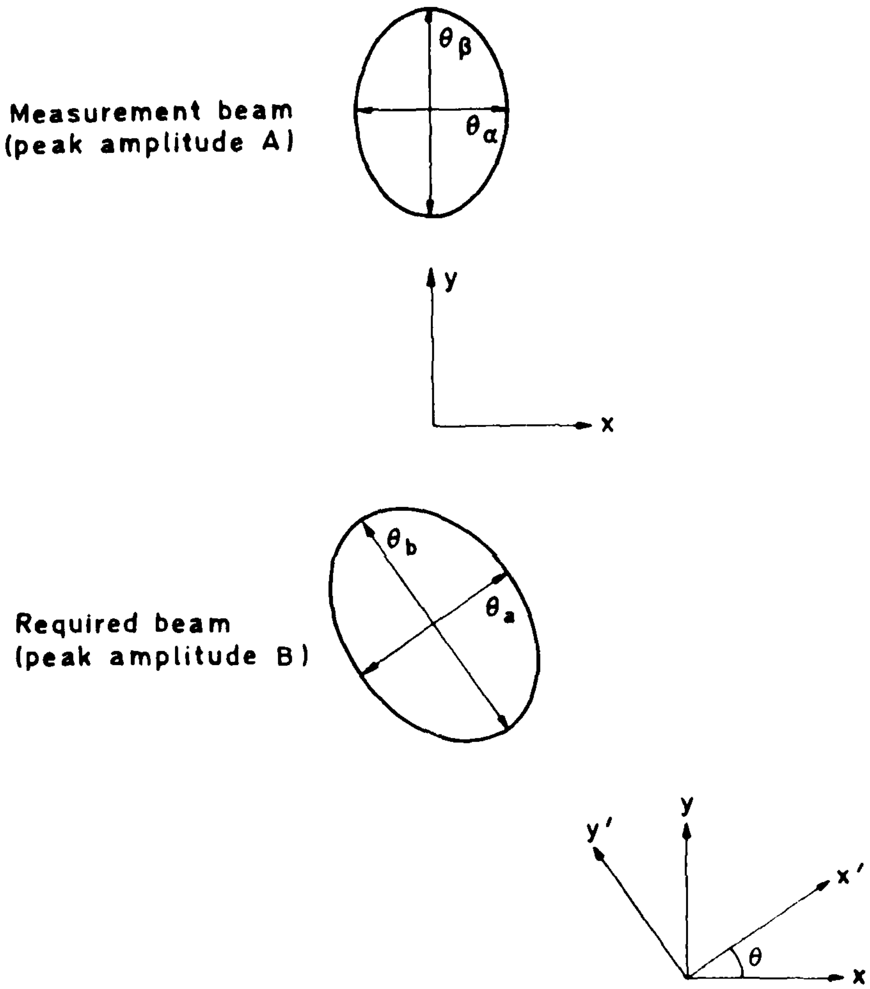


Fig 9 Geometry of the measurement and required beams referred to in Appendix E

REPORT DOCUMENTATION PAGE

Overall security classification of this page

UNCLASSIFIED

As far as possible this page should contain only unclassified information. If it is necessary to enter classified information, the box above must be marked to indicate the classification, e.g. Restricted, Confidential or Secret.

1. DRIC Reference (to be added by DRIC)	2. Originator's Reference	3. Agency Reference	4. Report Security Classification/Marking
	RAE TR 81152	N/A	UNLIMITED

5. DRIC Code for Originator	6. Originator (Corporate Author) Name and Location
7673000W	Royal Aircraft Establishment, Farnborough, Hants, UK

5a. Sponsoring Agency's Code	6a. Sponsoring Agency (Contract Authority) Name and Location
N/A	N/A

7. Title Transformation of data into a raster-scan format and some image processing applications

7a. (For Translations) Title in Foreign Language

7b. (For Conference Papers) Title, Place and Date of Conference

8. Author 1. Surname, Initials	9a. Author 2	9b. Authors 3, 4	10. Date	Pages	Refs.
Roberts, P.A.			December 1981	48	9

11. Contract Number	12. Period	13. Project	14. Other Reference Nos.
N/A	N/A		Space 610

15. Distribution statement
(a) Controlled by – Head of Space Dept, RAE (RAL)
(b) Special limitations (if any) –

16. Descriptors (Keywords) (Descriptors marked * are selected from TEST)

Contour maps. Digitals maps. Image processing. Interpolation.

17. Abstract

An account is given of techniques that can be used to process and manipulate data that is in the form of either a series of contours or a set of unconnected points, so that it can be transformed into a digital raster-scan format. Two applications of this work are described; these are the deliberate degradation in the spatial resolution of contour maps and the use of terrain height information in conjunction with Landsat imagery. In the discussion of both these topics reference is made to programs available at the Remote Sensing Unit of Space Department, RAE.

F5910/1

RAE Form A143

DA

FIL

08