Massachusetts Institute of Technology
Artificial Intelligence Laboratory

Working Paper 146 May 1977

Extracting topographic features from elevation data using contour lines

Anna R. Bruss

Abstract

This paper describes a method for finding such topographical features as ridges and valleys in a given terrain. Contour lines are used to obtain the desired result.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-75-C-0643.

Table of contents

Acknowledgements

I want to thank my adviser Professor Berthold Horn for suggesting this topic to me. Bob Woodham had a lot of patience with me while helping me to learn LISP and I owe him much gratitude for that. Part of the LISP code used was written by him. In discussions with him and Mark Lavin a lot of interesting ideas were developed.

1. Introduction

Topographical maps capture the surface features of the terrain they represent. A skilled person can deduce a lot of information from them such as where the stream-lines lie. But sometimes this is a long and tedious process. So we tried to write a computer program which would automate this task.

This paper deals with a particular question about the terrain structure. Where are the valleys and the ridges? One way a person would solve this problem is the following :

First one would draw a contour map. Some of the topographic features would be found by connecting "corners" of neighboring contour lines. It is quite clear which contour lines represent the same mountain. Between two different mountains there is a valley.

A valley is a depression between uplands and a ridge is an upper edge on the surface. Peaks of mountains will also be referred to as ridges.

These features describe the relative height between two parts of the surface. The absolute height of points of a ridge (valley) might be the same and therefore a contour line itself might sometimes represent a surface feature. To find other surface features a different technique must be applied which will be described later.

The following paragraph will explain briefly how a computer was programmed to automate the process described above.

For the purpose of this paper the elevations were given at grid points. The resolution of this grid determines how much detail of the terrain is captured. The first program finds all points representing any fixed elevation over the whole terrain using interpolation between grid points. Details of this algorithm are explained in the next section. The next step is to figure out which of these points represent contours of different mountains and valleys. Imposing a defined circulation on each contour line helps to seperate mountains. A corner finding algorithm detects those points which - properly connected - represent most of the valleys and ridges. Special configurations of contour lines give clues about the other surface features of interest.

2.Finding the contour map

> "A contour line connects the points on a land surface that have the same elevation."
>
> - Webster's seventh new collegiate dictionary. -

Elevation is typically available only at discrete grid points. If one desires to get a more detailed map one must superimpose a finer grid on the area. To find a particular contour a decision has to be made where to place each point since the elevation in question may not necessarily pass through any of the given grid points.

The "iterative subdivision" algorithm for finding points of a certain elevation works as follows: the input, which is a digital terrain model is subdivided into a number (depending on the size of the input) of smaller square areas Aj. Let a,b,c,d be the elavations at the corners of Aj. The task is to find points of height h. When h satisfies condition (1)

$$\min(a,b,c,d) \le h \le \max(a,b,c,d) \tag{1}$$

then Aj is subdivided into four equal squares Bi and condition (1) is tested again for the elevations at the corners of Bi. The algorithm keeps subdividing the squares until either condition (1) is not satisfied or the resolution of the input grid is reached. In the later case a point of height h is placed in the middle of the last square if it satisfies condition (1). Then the algorithm proceeds by working on the next subarea. By shifting the input grid by half its resolution in both x and y one finds that the points found by the algorithm are grid points. For points belonging to one contour the above described method guarantees that they are pairwise neighbors. (In a square tessellation each point has eight neighbors. In this paper the "neighborhood - property for points" refers to this definition of neighbor.)

The input has a resolution with respect to different heights. Trying to find two contours whose difference in elevation is smaller than the resolution of the input causes both contours to share points. This is not a disadvantage because it only makes sense to explore the data up to the resolution of the input - all further investigations are only guessing at the surface features.

So the relation between points on a given contour is precisely defined by the neighborhood-property and restricted to a small number of possibilities, a feature which will be exploited in the following steps.

This first program returns a list of unordered contour points with the above mentioned properties. The next task is to divide this list into new lists with each new list representing a contour line due to a surface feature.

The implementation is straightforward and is based on the notion of neighbor. As a point can have more than two neighbors one has to take care to arrange the points of a single contour line such that they preserve a certain circulation.

Lists consisting of single points are deleted because they are most likely due to noise. Figure 1 shows a contour map which was created by the computer. Four different heights are represented. It is an artifact of the display that contours do not close.

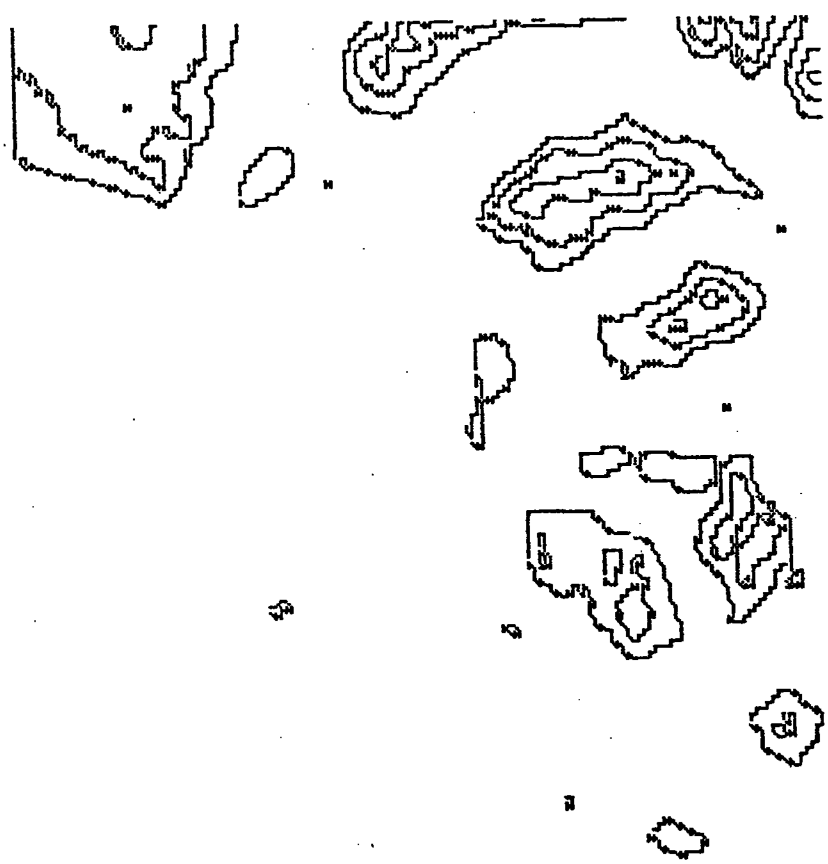

figure 1

3. Relation between contour lines

The program described in section 2 returns all contour lines of any desired height. But they do not necessarily all have the same circulation. In this paper the following orientation is defined. Looking in the direction of the contour line the next contour representing a lower (higher) elevation is to the left (right). This definition is ambiguous in the following case:

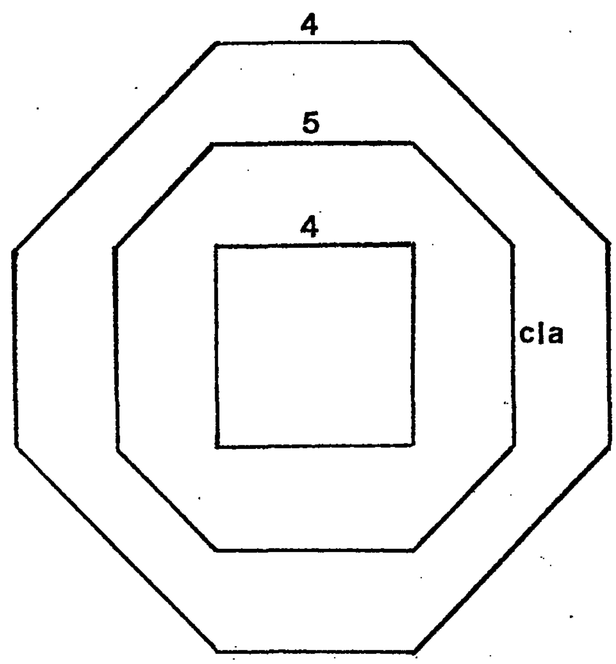

figure 2

The numbers next to the contours denote the elevation they represent. To either side of the contour cla are contours of a lower elevation. This problem is solved at the end of this section. Another application of uniform circulation is in deciding the concavity or convexity of a curve .

At the same time that a certain circulation is imposed on the contour lines one finds out how they relate to each other - that is which contour line to the left (right) denotes a lower (higher) elevation with respect to a given contour. For most contours a line and its neighbors will be adjacent, only ridges or valleys are exceptions.

One starts with the contour line of highest elevation cla in figure 3. For an arbitrary point a on this contour the nearest point b lying on a contour line clb of the next lower elevation is found. The vector connecting a and its successor point defines the current orientation of the line cla. Calculating the normal vector and its intersection with clb - let it be the point s - one can determine whether s lies to the left of cla. If not, the order of the points of cla is reversed. Then cla becomes the "right" neighbor of clb and clb becomes the "left" neighbor of cla. (This defines the "neighborhood - property for contours". Interchanging the notions left and right yields to an equivalent defintion.) Only the contours representing the lowest elevation can not be ordered that way. For these a point of the next higher elevation is used to determine the proper circulation. Figure 3 illustrates the method described above.

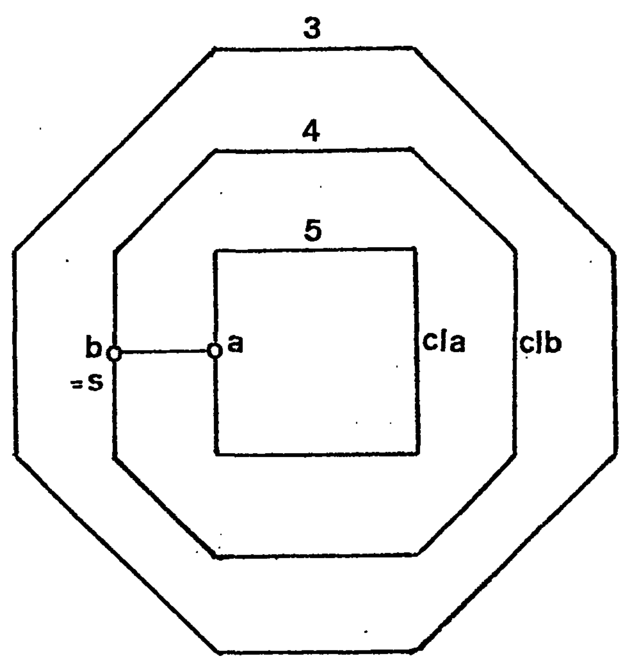

figure 3

Note that a contour and its neighbor must have the same circulation. Figure 4 shows a case when the wrong neighbor was chosen for a contour.

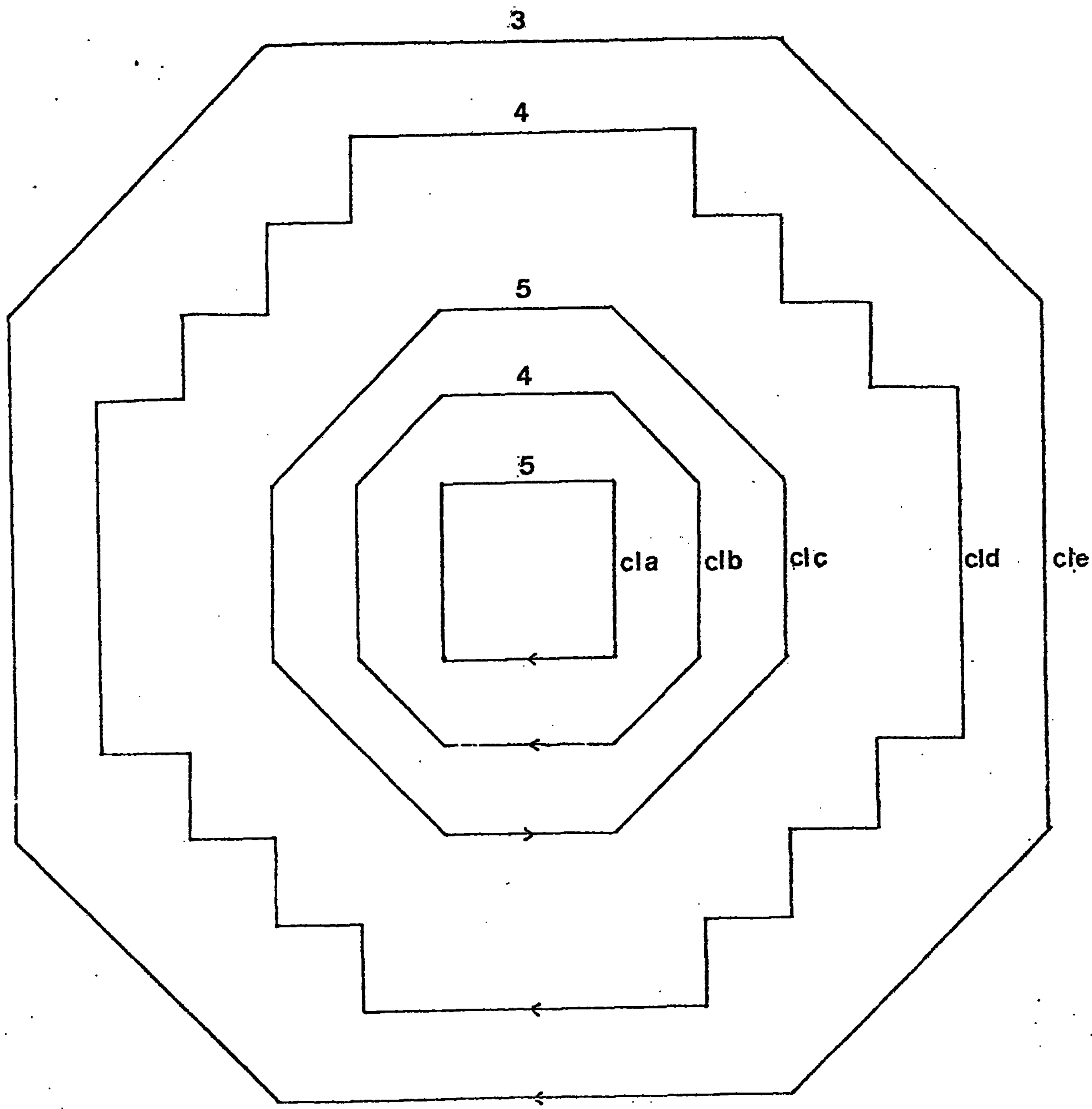

figure 4

Both cla and clc have clb as their left neighbor, but clb and clc have a different circulation. Therefore a new left neighbor must be found for clc. The same method as before is applied, taking into account that clb is not a proper candidate. At the same time the ambiguity with the orientaion of clc is solved. The lines representing the highest elevation do not have a right neighbor and the ones of the lowest elevation do not have any left neighbor. This feature enables us to detect very quickly these special contour lines.

It can happen that a contour line has more than one right neighbor. In section 5 this problem is discussed in detail. Exploiting the property of how many adjacent contours a line has gives us a lot of clues about the features of the terrain.

4. Detection of corners

The task of this project is to detect valleys and ridges in a certain area. Let us first find the corners of the contour lines. Assuming that contours can be parameterized as a function of one continuous variable then corners are the discontinuities in the first derivative of this variable. In the discrete case one has to set a threshold to decide whether there is a corner at a given point on a contour or whether there is only quantization noise. At the same time a corner is found, it is decided wether it is concave or convex.

For each contour line the following algorithm is applied. (This algorithm is similar to the one in the paper by Freeman [2]). A link is formed by three consecutive points of a contour line. For each link the angle between the x-axis and the link is calculated. If the difference of two consecutive angles is zero the points of the appropriate two links represent a "straight" line. If the difference of two consecutive angles is greater than the threshold, then the two links form a corner. The choice of how many points to use to form a link depends on the above mentioned threshold. Taking more points would smooth the line and therefore for a fixed threshold the algorithm would find only sharper corners. As already mentioned two consecutive points are neighbors and therefore the above defined links give only rise to a small number of different angles. Therefore it can not happen that the difference of two angles wobbles around zero. Let delta1, delta2 and delta3 be three consecutive angles and let delta1 = delta2 be different from delta3. Then the last point from link1 is a corner. Depending on the ratio between delta2 or delta3 this corner is concave or convex. Conversely when the difference of three consecutive angles form the sequence not zero, zero a corner is where the second link starts.

Figure 5 illustrates the corner finding algorithm. In a way the links drawn represent the smoothened curve and therefore the points marked with an arrow represent the corners.

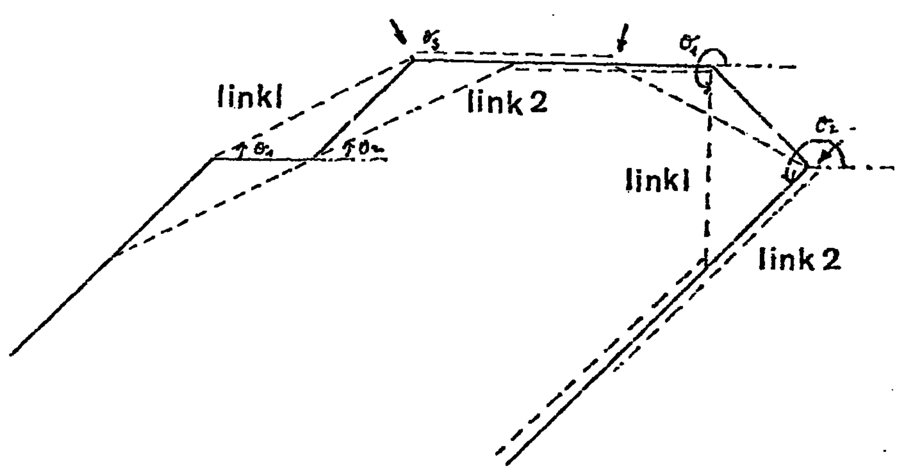

figure 5

Again a decision has been made about the definition of a corner. Figure 6 illustrates a sequence of differences of angles between succesive links: 0, -45, 270, -45, 0; The algorithm detects two corners, but one could argue that this is only one corner. Assuming that the data does not have too much noise the algorithm as implemented finds the details in the surface structure .

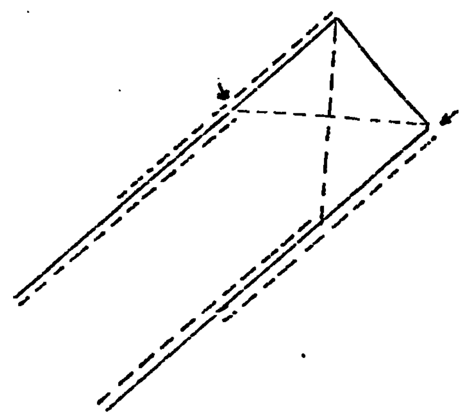

figure 6

5. Finding the stream-lines

Most contour lines have one right and one left neighbor. The line denoting a peak does not have a right neighbor. But it can happen that a depression is characterized by a contour which does not have a right neighbor as shown in figure 7.

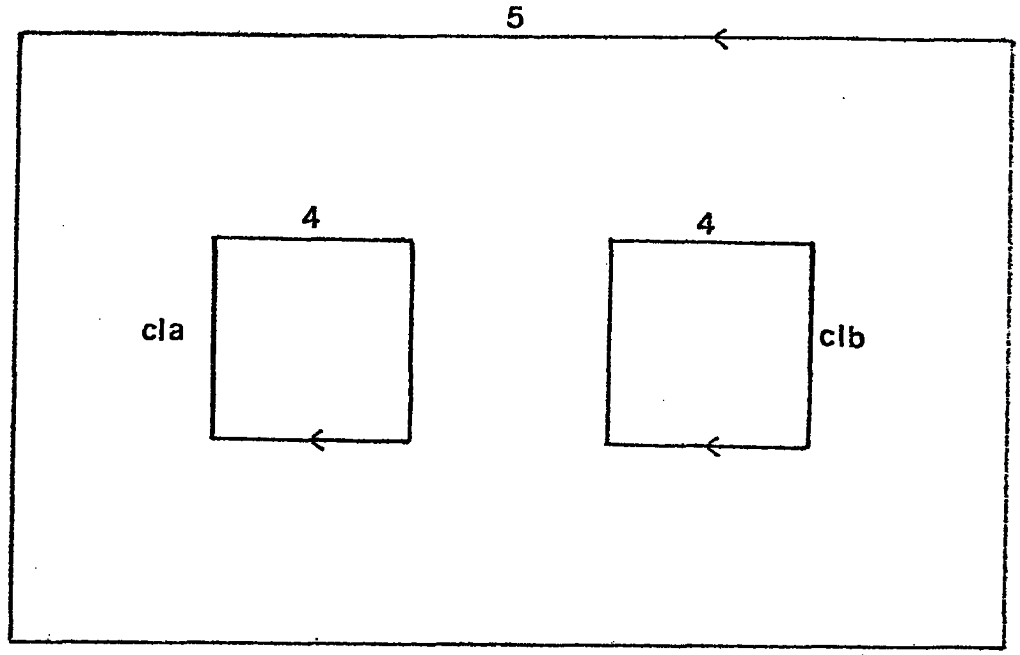

figure 7

Let cla be the contour which does not have a right neighbor. To find out if it is a peak or a depression one can apply this method. Find the nearest contour clb which has the same height as cla. Compute all the points a (on cla) and b (on clb) with the following property: the line connecting a and b does not intersect with either cla or clb. If the line ab does not have any points in common with either left or right neighbor of cla and clb then there is a ridge between the points a and b. Cla becomes also a left neighbor of the right neighbor of clb. Otherwise cla is a ridge. Note that an area can be a ridge and a valley without implying a contradiction. Slope is a measure which depends on the reference direction. Figure 8 illustrates such case.(marked with an arrow)

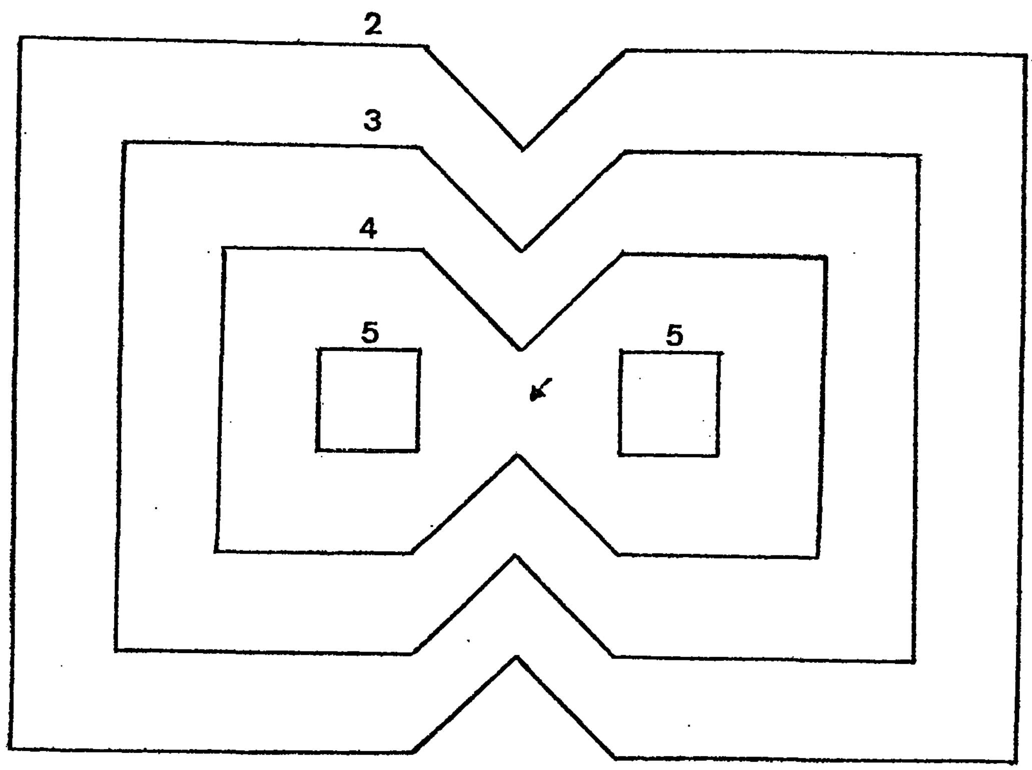

figure 8

Contours which have more than one right neighbor give some clues about ridges or valleys. Figure 4 (with the circulation of clc reversed) and figure 9 illustrate these cases.

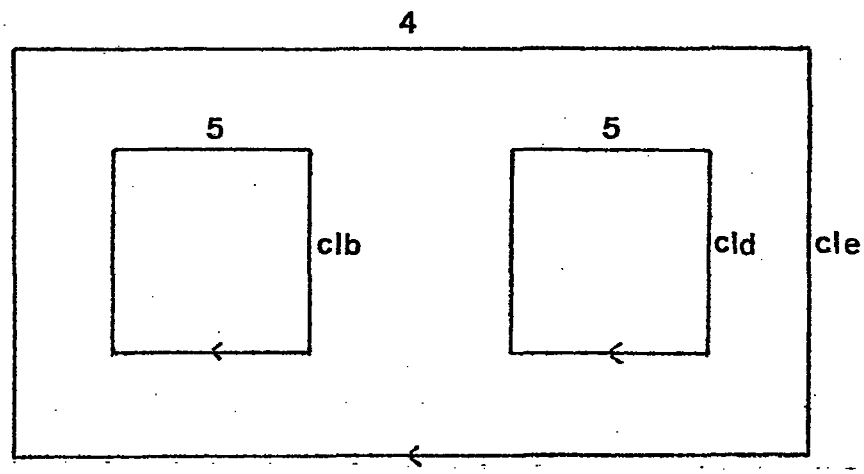

figure 9

Let clb and cld be the two right neighbors of the contour cle. (Contours with more than two neighbors are treated in an analogous fashion.) Again find all points b (on clb) and d (on cld) with the following property: the line connecting b and d does not intersect either clb or cld. If the line bd does not intersect with either a right neighbor of clb or a right neighbor of cld than there is a valley between b and d. Suppose it intersects with the right neighbor of cld, then clb is a valley. If it intersects with the right neighbor of clb, then cld is a valley.

The rest of the surface features are found by connecting properly the corner of the contour lines. One starts with the ridges (contours which do not have a right neighbor) and considers first the convex points on them. Let pa be a convex point on a ridge cla and let clb be the left neighbor of cla. Connect pa with a convex point on clb which has the smallest distance to pa. This process is continued until a valley is reached. For the concave points one proceeds analogously. If both adjacent contours of a ridge line are of the same height proper corners have to be connected to either side of the ridge, eg. compare figure 4 (with the circulation of clc reversed): clb and cld are adjacent contours of clc, cld is the left neighbor of clc and the convex (concave) points of both contours are connected. For finding the surface features on the slope between clb and clc one renames the convex (concave) corners on clc to concave (convex) corners and then connects the proper points between clc and clb. Note that concavity and convexity are directional measurements and therefore when changing the viewpoint these notions have to be interchanged.

6. Conclusion

Most of the method which was described in this paper was implemented and returned very satisfying results. The data used was sampled from some areas of Switzerland and Canada and had therefore approximately all features this world (the best of all possible worlds) has to offer.

Bibliography

(1) Herbert Freeman, "Computer Processing of Line-Drawing Images", Computing Surveys, Vol.6, No.1, March 1974

(2) Herbert Freeman, Larry S. Davis, "A Corner-Finding Algorithm for Chain-Coded Curves" , University of Maryland, Computer Science Center, TR-399