A USER-FRIENDLY COMPUTER-AIDED CONSTRUCTION OF BOUNDARIES

Dipl.-Ing. K. Grundey and Dr. U. Schweitzer and Dr. R. Neuser

Summary

To calculate a Digital Terrain Model (DTM) of any set M of measuring points

$$M = \{(x(i),y(i),z(i)) \mid i=1,\dots,n\}$$

it is often neccessary to combine some subsets of M. Such subsets may be special point sets e.g. boundaries, slopes, break points, etc. Such point sets are important for calculating the altitudes of DTM grid points, because they influence the selection of the correct neighbourhood of the grid point in question. We wish to present a fast and user-friendly algorithm to calculate such point sets.

Calculation of Boundaries

There are several methods of computing boundaries of a given terrain, but we wish to calculate a polygon around a given set of measuring points closer than the convex hull. So the final boundary can be formed by only a few interactive manipulations.

The first method uses a recursive algorithm which creates the one to one convex polygon around the terrain

$$M = \{(x(i),y(i)) \mid i=1,\dots,n\} \; ,$$

where the edges of the polygon are measuring points.
The first three points of M form a triangle which is just the convex hull of these three points. To deduce from k to k+1 $(3 \le k < n)$ let

$$B(k) = \{(x(k(j)),y(k(j))) \mid j=1,\dots,m(k)\}$$

be the set of boundary points of

$$M(k) = \{(x(i),y(i)) \mid i=1,\dots,k\} \; .$$

If the next point (x(k+1),y(k+1)) is inside the polygon, i.e. if (x(k+1),y(k+1)) is on the left side of the line segment between each

$$(x(k(j)),y(k(j))) \text{ and } (x(k(j+1)),y(k(j+1)))$$

(j=1,...,m(k), where k(m(k)+1):=k(1)), then set

$$B(k+1) = B(k) \; .$$

Otherwise it has to be determined by moving counter-clockwise along the polygon B(k) if (x(k+1),y(k+1)) is on the left hand or right hand side of the considered line segment. Figure 1 shows how to deform the polygon by inserting the new point (x(k+1),y(k+1)) .

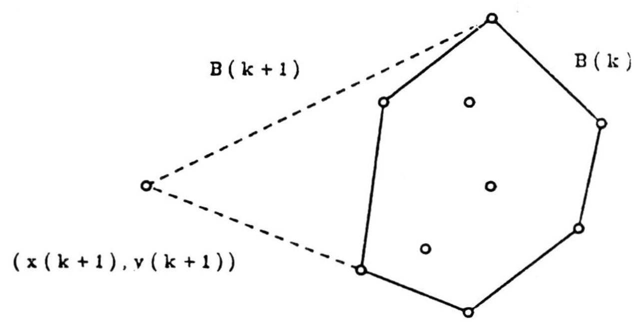


Figure 1

By continuing this algorithm to each measuring point, a convex polygon around M is finally obtained.
This method can be applied to construct a boundary which fits the terrain best in some way. To achieve this, the rectangle

$$R = [xmin,ymin] * [xmax,ymax]$$

is subdivided into N*N rectangles of length

$$L = (xmax-xmin)/N$$

and width

$$W = (ymax-ymin)/N \; ,$$

where the 'subdivision number' N is assumed to

be fixed. Its value is calculated later.
For $1 \le i,k \le N$ let

$$R(i,k) = \{(x,y) \mid xmin+(i-1)*L \le x \le xmin+i*L \text{ and } ymin+(k-1)*W \le y \le ymin+k*W \}$$

and

D(i,k)=1, if R(i,k) contains a measuring point,

otherwise D(i,k)=0.

A closed polygon, which is not yet the final boundary, is constructed in the following fashion (see Figure 2). The points of the polygon are not measuring points but are centered on grid rectangles.

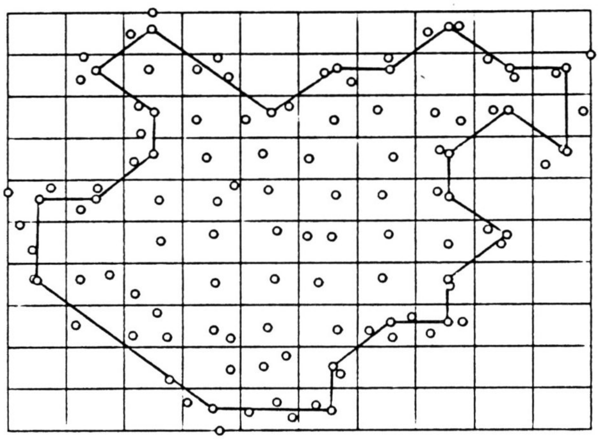


Figure 2

Starting in the first column (from bottom) and moving from left to right there is a first rectangle R(j,1) which contains at least one measuring point, i.e.

$$j = \min \{i=1,\ldots,N \mid D(i,1)=1 \} .$$

The polygon is constructed inductively moving counter-clockwise as described in Figure 3.

There are eight possible directions of movement, numbered from 0 to 7. The direction of the first movement is 6. If the direction of the last movement was m and if

$$n = \mathrm{mod}\ (2*\mathrm{int}(m/2)+7\ ,\ 8)$$

then one must consider the neighbouring rectangle in the direction n. If this rectangle does not contain a measuring point then one must consider the direction mod (n+1 , 8) relative to the previous rectangle, etc., until a rectangle R(i,k) with D(i,k)=1 is found. This process must be continued until the first rectangle, i.e. the starting point, is reached again, and thus the polygon is closed.

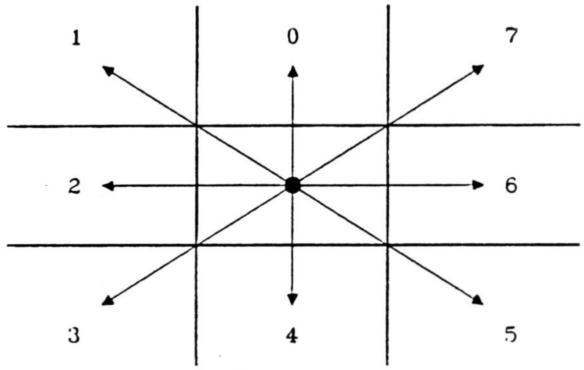


Figure 3

It may happen that one or more rectangles R(i,k) with D(i,k)=1 are situated outside the closed polygon. In that case the subdivision number N was too large. To obtain a reasonable N, the following postulates should be fulfilled:

- If a closed polygon is constructed by the process described above, there must be no rectangle R(i,k) with D(i,k)=1 outside of the polygon.

- The polygon must not contain two or more identical points (see Figure 5).

Now the final boundary B of the given terrain is constructed:

Starting with a fixed number N (which depends on the density of the measuring points, but usually N=20 is acceptable) it must be determined if the two postulates hold or not. If at least one of the postulates is violated, then one must start with a lower number N=N-1, etc. until both postulates are valid or until N=1. The case N=1 is a special case, because then the final polygon is just the convex boundary described above.
With the fixed number N the process can be started by searching the first rectangle R(m,1) with D(m,1)=1 and by constructing an open convex polygon within this rectangle, such that all measuring points belonging to that rectangle are positioned on the left side of the polygon (see Figure 4). The initial point and the end point of the polygon depend on the direction of the

previous and the following movement respectively. These points can be maximum or minimum x- or y-values or they can be the nearest to one edge

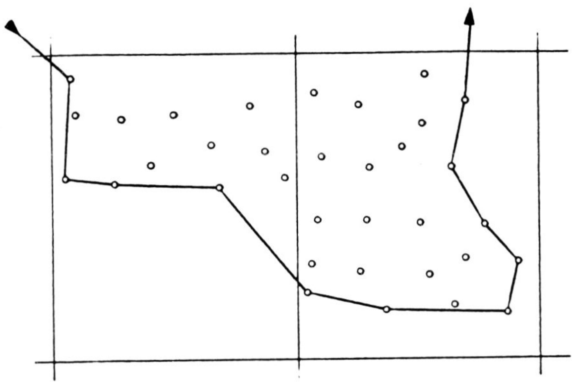


Figure 4

Subsequently the next rectangle must be sought by the process belonging to Figure 3, and within this rectangle the next open convex polygon must be constructed, etc. up to the first rectangle R(m,1). By connecting all such obtained polygons, a closed polygon is finally obtained, which is the boundary of the terrain (see Figure 5).

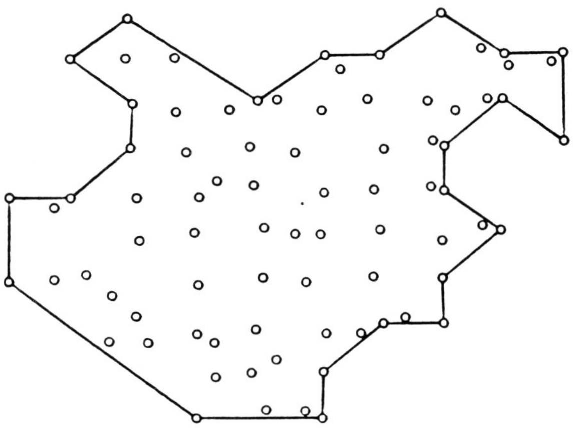


Figure 5

Calculation of Inner Boundaries

Many Terrain Models need an inner boundary, because they may contain regions where points could not be measured, e.g. regions such as lakes or buildings. To compute an inner boundary it is neccessary to know the position of the inner boundary approximately. The position must be fixed by the user by the input of an open or closed polygon which lies within the inner boundary (see Figure 6).

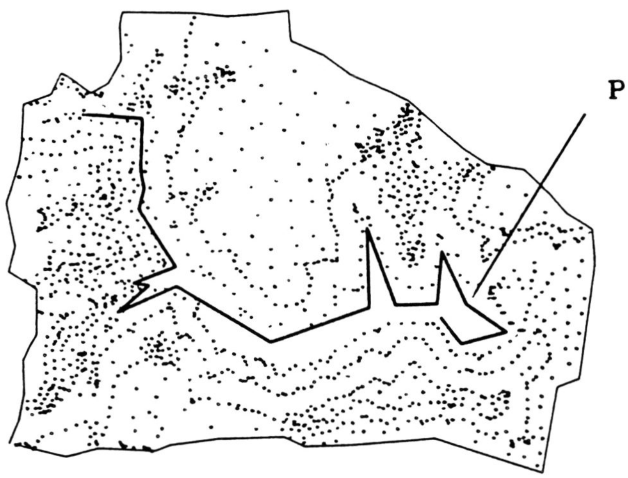


Figure 6

The problem can be reduced as follows:
First a closed polygon of measuring points must be calculated around the open or closed polygon previously fixed. This algorithm is like inflating the original polygon. Without loss of generality we assume the original polygon to be closed. Otherwise we close it by moving along from the starting point to the end point and then back to the starting point again.

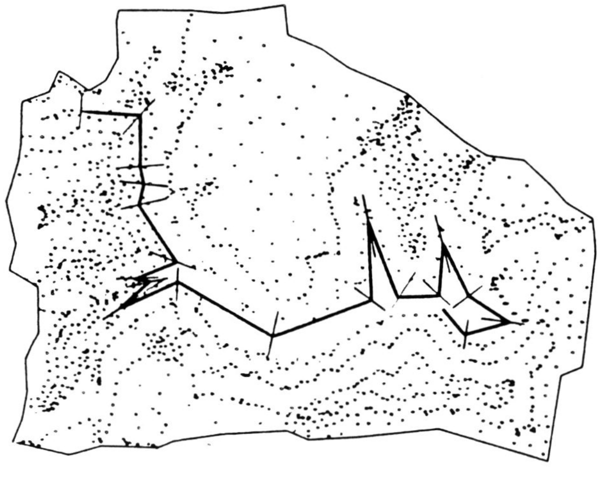


Figure 7

Let the polygon P be given by the points

$$\{(x(i),y(i)) \mid i=1,\ldots,n\} ,$$

where

$$x(n) = x(1) \quad \text{and} \quad y(n) = y(1) .$$

The terrain outside the closed polygon is divided into n regions which are limited by the line segments of the polygon and the bisectors of the angles between two neighboured polygon segments (see Figure 7).
The inner boundary will now be constructed subsequently by calculating an open polygon within each segment. Let us pick out one segment for example (see figure 8). This segment can be divided into N subsegments as follows:

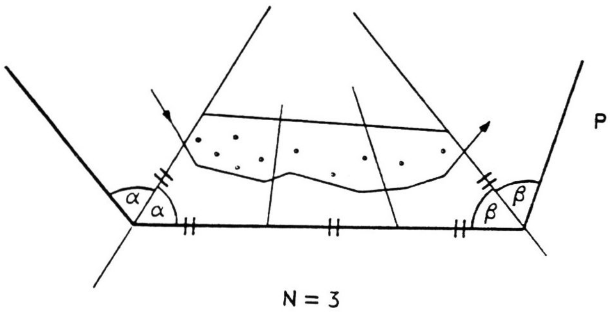


Figure 8

If any subsegment contains no measuring point, the division of the segment was too fine and a lower dividing number N=N-1 must be taken until N=1 or until each subsegment contains some measuring point.

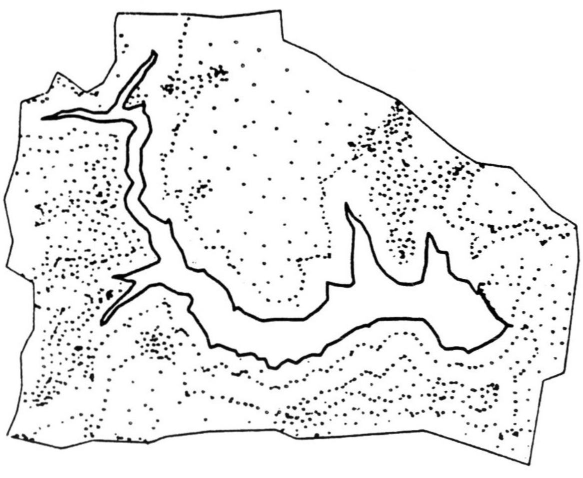
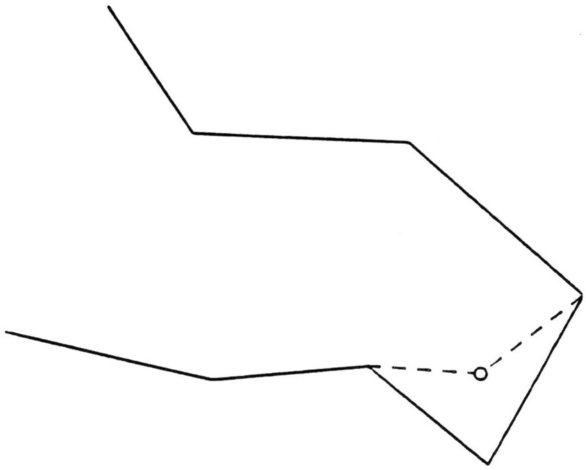


Figure 9

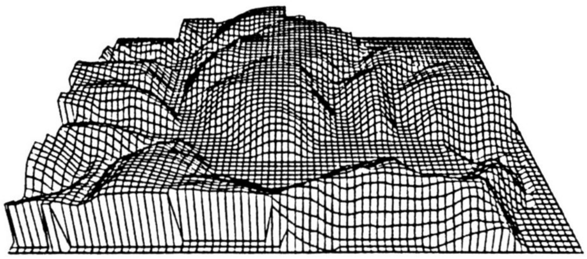
Within each subsegment a convex polygon is now constructed as shown in Figure 9. By continuing this algorithm for each line segment of the given polygon P we get a closed polygon around P.
It may happen that a few measuring points exist within the calculated inner boundary (see Figure 10). For such a case these points must be inserted into the polygon by searching for the nearest point and its associated neighbouring point of the polygon and inserting the point between the polygon points.



Inserting inner points

Figure 10

It should be mentioned that this algorithm can also be applied to calculate the boundary described at the beginning. The convex boundary is then to be considered as the initial polygon P.



REFERENCES

1. Goritskii, Yu.A. and Zharinov S.E. : Organisation of a Graphical Interactive System
Avtometriya 1 (Jan.- Febr. 1983), p. 18-23

2. Newman, W.M. and Sproull, R.F. :
Principles of Interactive Computer Graphics
New York, 1979

3. Little, Steve :
The Organisational Implications of CAAD
CAD'84, Butterworth, p. 156 - 164

4. Schweitzer, U. and Grundey, K. :
Standortwahl durch Bewegungsmassenoptimierung mit Hilfe des Digitalen Geländemodells
III. Symposium on Operations Research, Mannheim, 1978, p. 489 - 493

5. Schweitzer, U. and Grundey, K. :
Prüfung von Meßdaten für Digitale Geländemodelle
CAMP 83, VDE-Verlag, p. 1230 - 1239

6. Schweitzer, U. and Neuser, R. and Grundey, K. :
Real Interactive Graphical Technics
CAMP 84, AMK Berlin, p. 464 - 466

7. Schweitzer, U. and Neuser, R. and Grundey, K. :
A Multi-User Concept for Digital Models
MICAD '85, Paris

8. Schweitzer, U. and Neuser, R. and Grundey, K. :
A Microcomputer Based Concept with Real Interactive Graphical Technics
International Seminar on Microcomputer Technology, Jakarta 1985

by Dipl.-Ing. Klaus Grundey
Kamperfeld 41
D-4300 Essen 1 , F.R.G.

and

Dr. Udo Schweitzer
Memelstr. 25
D-4300 Essen 15 , F.R.G.

and

Dr. Rolf Neuser
Nahestr. 16
D-4300 Essen 18 , F.R.G.