

QUADTREES AND TRIANGULATION IN DIGITAL ELEVATION MODELS

Jan Terje Bjørke
The Norwegian Institute of Technology
Department of Geodesy and Photogrammetry
7034-Trondheim
Norway
Commission no.: IV

Abstract

This paper is a report on the design and the implementation of a program system for the construction of digital elevation models from arbitrarily distributed elevation points. The point-quadtree is used as a tool to answer the query: find all points inside a given area. Since the search properties of the point-quadtree are strongly dependent on the shape of the tree, an algorithm to balance the point-quadtree is implemented. The time complexity of this algorithm gives the program the running time $O(n \cdot \log(n))$, where n is the number of points in the point-quadtree.

The triangulation known as Delaunay triangulation, is used for the triangular decomposition of the surface of the terrain to be mapped.

Experiences arising from the application of the program system in some mapping projects are reported.

Introduction

There are several ways of generating a triangular mesh from random data. Elfick (1979) mentions four different methods. However, the time complexity of computer algorithms is important, and should be considered when the method of triangulation is selected. The method selected in our program system is based on the Delaunay triangulation. Delaunay triangulation has been applied for example, by McLain(1975), Elfick (1979), McCullagh and Ross (1980), and Lowden (1985).

The Delaunay triangulation requires an algorithm for fast access to geometric data in order to answer queries such as how to find all points inside a given area. There are several algorithms for this purpose. Some of these algorithms are based on grouping the data points into buckets, as in the Excell method (Tamminen 1981) and Grid file method (Hinrichs 1985) for example. Our algorithm, however, applies the point-quadtree (Samet 1984). It should be stressed that search algorithms based on the point quadtree assume that the search can be done in core.

The triangulation

The third point of a triangle is defined by the circumscribing circle passing through the base line and the third point. The

rightmost/leftmost neighbour of a base line is found when the right/left part of the circumscribing circle contains within it no other data points, see Fig. 1.

In order to simplify the calculations, the coordinates of the candidates are transformed to a local coordinate system with the X-axis along the base line and the origin at one of the end points of the base line.

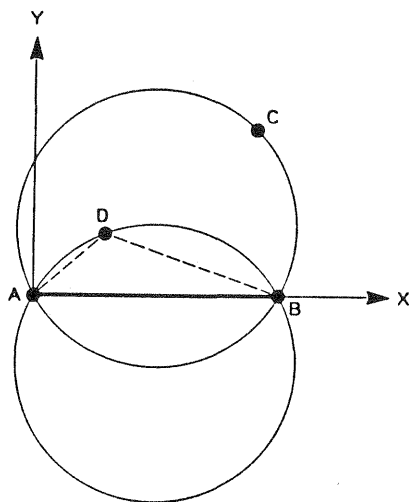


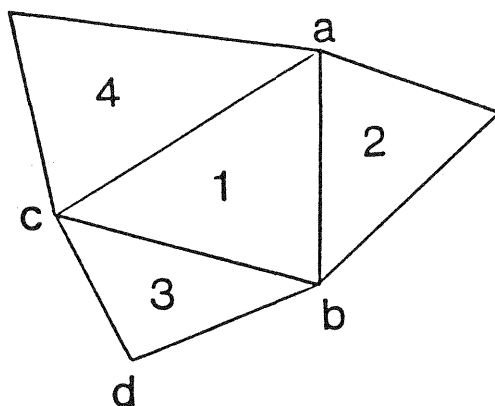
Fig. 1. Search for the third point of a triangle.

The search for the third point is a critical step for the efficiency of the algorithm. If, for example, all the n points in the overall set are checked for every triangle to be formed, the program will require a running time $O(n^2)$. A better strategy, therefore, is to select the most probable subset of data points. This subset can be defined by a circle located on the relevant side of the base line. The initial diameter of the circle is set equal the length of the base line. If the circle contains no data points within it, its size can be increased and a new search performed, as in McCullagh and Ross (1980). This stepwise growth of the circle or the window, goes on until an appropriate candidate is found or the window reaches a maximum value.

Since the coordinates of the points are expressed in the given local coordinate system, the appropriate third point of a triangle belongs to the circumscribing circle whose centre point has the lowest Y-value. The X-values of the centre points are, of course, equal for all the circumscribing circles belonging to the same base line.

The data structure selected for the triangulation is shown in Fig.2.

Two tables, the E-table and the T-table, define the edges and their associated triangles. Eft and Est hold pointers to the first triangle and the second triangle of the edges, and Ife, Tse, Tte hold pointers to the three edges of the triangles. If an edge is a part of one triangle only, the edge is located at the perimeter of the network, and its pointer to the T-table is set to nil.



E-table					T-table			
Entry	Eft	Est	Efrom	Eto	Entry	Tfe	Tse	Tte
1	1	2	a	b	1	1	2	3
2	1	4	a	c	2	1	4	5
3	1	3	c	d	3	3	6	7
4					4	2	8	9

Fig.2. Data structure for the triangulation.

The E-table is initialized by entering the first triangle side of the network and marking its two associated triangles unvisited, that is $Eft(1)=unvisited$ and $Est(1)=unvisited$. Then a search for the third point of triangle $Eft(1)$ is started. When this point is found, the two new triangle sides are stored in the E-table and their associated triangle is stored in the T-table. Then a new search is performed for the second triangle $Est(1)$, and the tables are updated in the same way as for the first triangle. Thereafter the procedure continues with edge 2, edge 3 and so on until all triangles are constructed. As soon as a data point is used in the formation of a triangle, it is flagged, because the triangulation algorithm has to know if a point has been used before or not.

Some special cases may occur in the triangulation. One of them is shown in Fig.3. This case can be termed the regular grid case, since the data points are distributed in the same way as points in a regular rectangular grid. The circumscribing circle therefore passes through four points, as shown in Fig. 3. An additional test has to be introduced, but if this test is not carefully designed, overlapping triangles may be formed.

There are at least two possible formulations for this test. One method is to add small values to the coordinates. Another method is to define nearest neighbours on the basis of the angular distance from the base line. The rotational direction, however, is defined clockwise

or anticlockwise dependent on the direction of the base line.

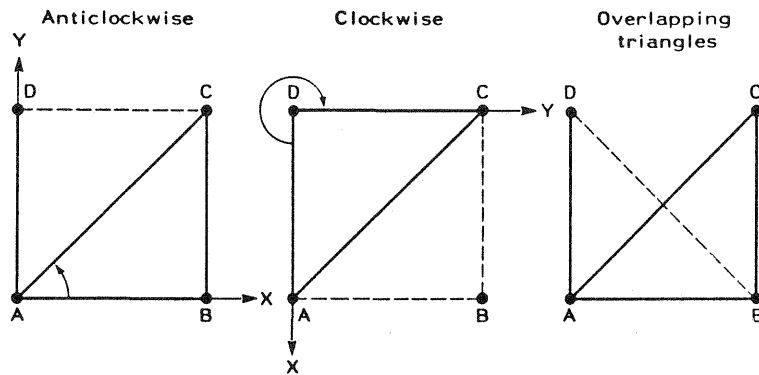


Fig. 3. The regular grid case.

Balancing the point quadtree

A point quadtree (p-q-tree) and its partition of the XY-plane is shown in Fig. 4.

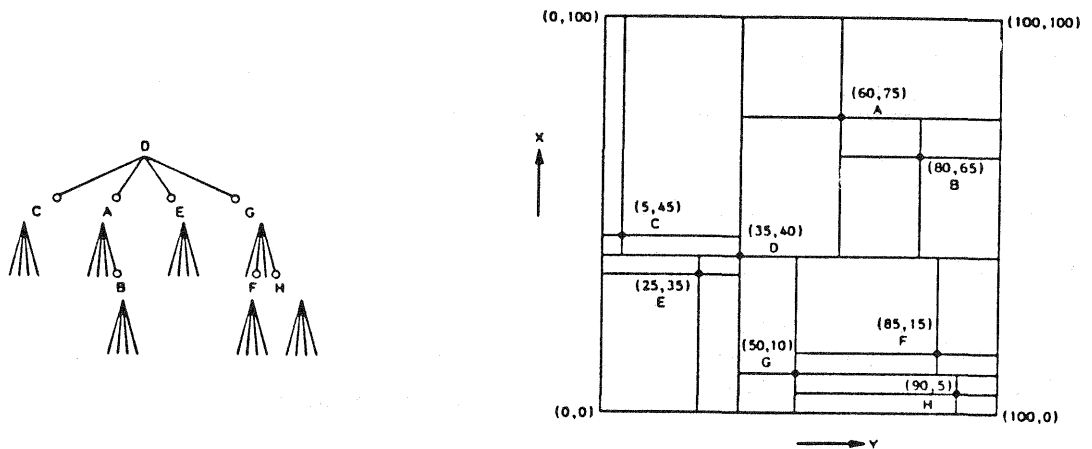


Fig. 4. A point quadtree and its partition of the XY-plane.

Assume a well-balanced p-q tree and an exact match query. The answer can be given in $O(h)$ time; where h = the height of the tree. In the Delaunay triangulation region queries are given. If the window, however, is relatively small, the answer can be found in approximately the same time as in the exact match case.

The shape of the p-q tree is strongly dependent on the succession of the points to be inserted. Therefore, it is appropriate to implement a procedure to balance the tree.

The basic idea of this algorithm is to find the most suitable split points. There may be several ways to select these points. One method is to calculate the mean point of the current interval as

$$XC = \frac{EX}{n} ; YC = \frac{EY}{n}$$

and select as split point the data point nearest to (XC, YC) . This

method is applied by Midtbø (1987).

This point then splits the interval into four subsets. Thereafter the procedure can be repeated recursively for each of those subsets. This procedure cannot, of course, guarantee a well-balanced p-q tree for all possible distributions of the data points. If, for example, the data points are distributed along a straight line, the split point will partition the data set into two subsets only.

The balancing procedure is illustrated in Fig. 5. An array P holds the coordinates of the data points, one record for each pair of coordinates. A and B are pointers to the start and end of an interval of P.

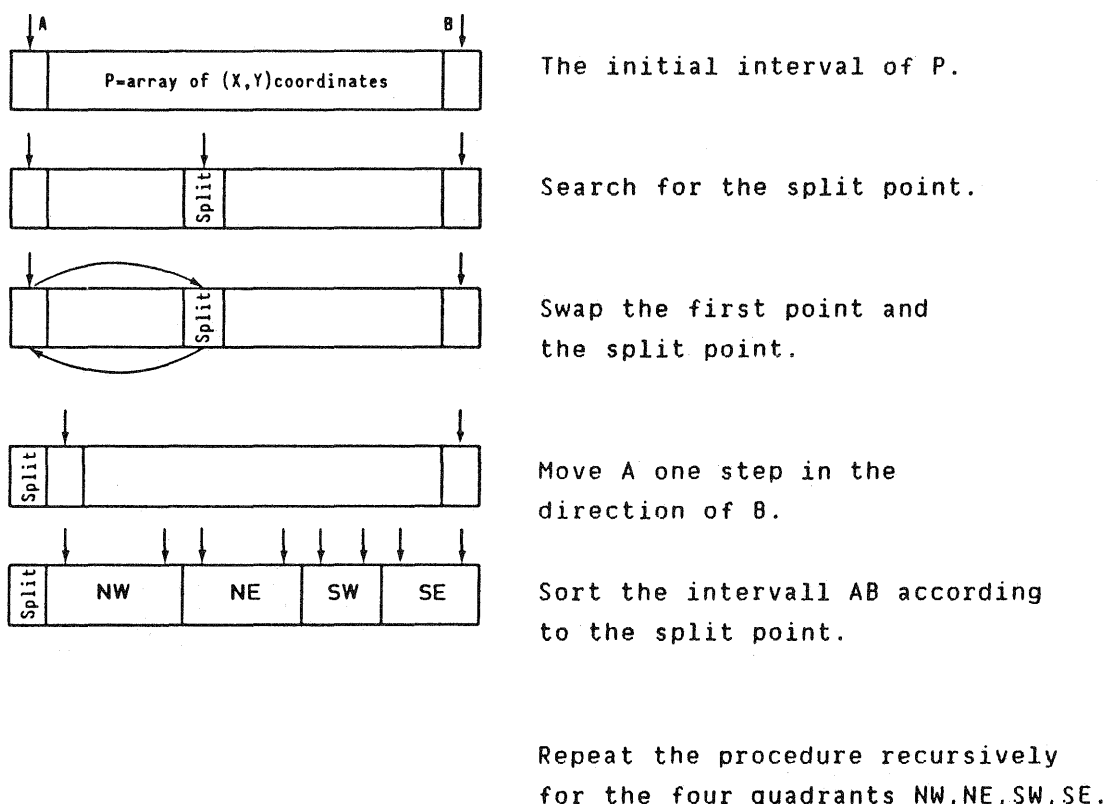


Fig. 5. The balancing procedure.

If the data points can fit into a well-balanced p-q tree, the running time of the given balancing routine is $O(n \cdot \log(n))$.

Experiences

The program system at hand has been used in several mapping projects. Fig. 6 shows the network of triangles and isolines generated from 1000 data points. The points, in this case are measured by use of digital land surveying equipment. Since the program system allows the land surveyor to measure the data points in an arbitrary sequence, he can do his field work in the most convenient way.

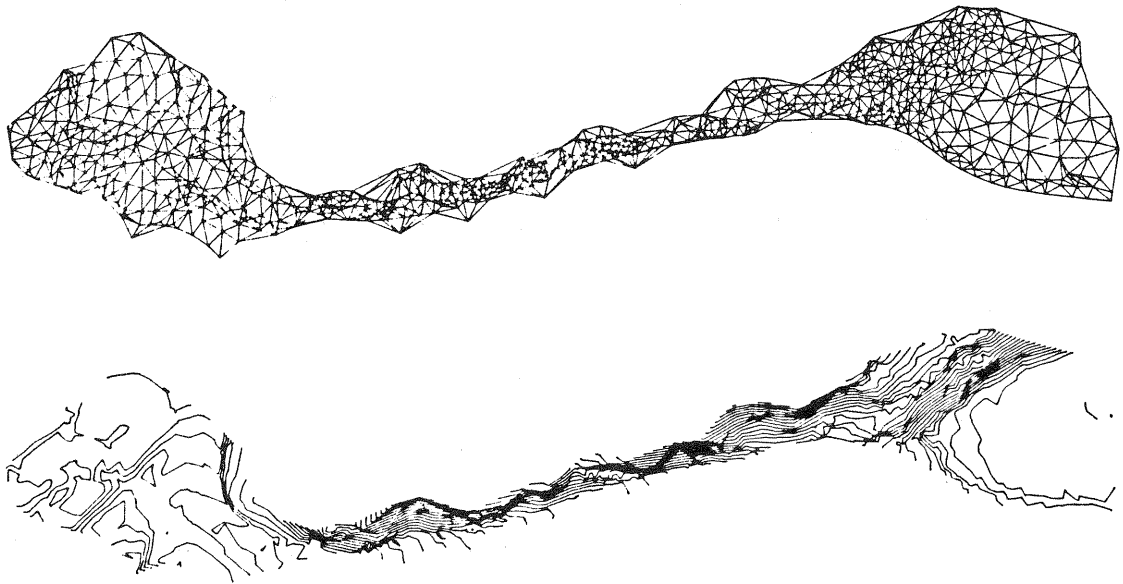


Fig. 6. Triangles and isolines.

In another project the part above the sea level of some icebergs were mapped and their volume above this level computed. Linear interpolation and triangular meshes were very useful in this project, because the surface of the icebergs has sharp breaklines, like a surface composed of inclined planes. The volume in this case is easily computed by summing the volume of the prisms as $V = \sum (F/3)(H_1 + H_2 + H_3)$.

Conclusion

When the time complexity of the algorithms and the size of the required storage space are considered, the Delaunay triangulation and the balanced point quadtree are good bases for the implementation of digital elevation models.

From a land surveyor's point of view, the triangular mesh is a natural way of thinking. Therefore he can recognize intuitively the consequences of adding new data points to his set of collected data points.

References

- Elfick M. H. 1979. Contouring by Use of a Triangular Mesh. The Cartographic Journal vol. 16, no. 1, pp. 24-29.
- Hinrichs K. H. 1985. The grid file system: Implementation and case studies of application. Dissertation, Institut für Informatik, ETH, Zürich, 112 p.

- Lowden B. 1985. A Storage-Efficient Method for Producing Smooth Contour Maps. Infor vol. 23, no. 4, pp. 447-468.
- McLain D.H. 1975. Two-dimensional interpolation from random data. The Computer Journal vol. 19, no. 2, pp. 178-181.
- McCullagh M. J., Ross C. G. 1980. Delaunay Triangulation of a Random Data Set for Isarithmic Mapping. The Cartographic Journal vol. 17, no. 2, pp. 93-99.
- Midtbø T. 1987. Digitale terrengmodellar. Hovedoppgåve, The Norwegian Institute of Technology, Department of Geodesy and Photogrammetry, Trondheim 124 p.
- Samet H. 1984. The Quadtree and Related Hierarchical Data Structures. Computing Surveys vol. 16, no. 2, pp. 187-260.
- Tamminen M. 1981. Acta Polytechnica Scandinavica, Mathematics and Computer Science Series no.34, Helsinki University of Technology, 57 p.