SPATIAL ADJACENCY - A GENERAL APPROACH

Christopher M. Gold
Department of Geography
Memorial University of Newfoundland
St. John's, Nfld., Canada A1B 3X9.
BITNET Address: CGOLD@MUN.

ABSTRACT

It is fact universally acknowledged that discrete computing systems are ill-equipped to process vector-based spatial information: inexact line intersection calculations and similar geometric (co-ordinate) operations can not readily guarantee consistent graphical structures (topology). It is proposed here that use of Voronoi diagrams, especially euclidean-distance nearest-object Voronoi diagrams of points and line segments in the plane, permits a general-purpose conversion of geometric information to a graphically-structured form amenable thereafter to graph traversal and other fundamental discrete operations appropriate to the computing environment employed. While the divide-and-conquer approach is efficient, object-at-a-time insertion and deletion techniques build on the current adjacency structure; preserve it and are consistent with database updating methodology; and direct comparisons can be used between one and two-dimensional linked-list operations. This approach permits the handling of spatial information in a manner consistent with computer strengths - by using linked-list, graph-traversal and tree-search algorithms well known to computing science to answer a wide variety of basic geographic queries, including interpolation, spatial ordering and medial-axis transforms.

INTRODUCTION

This paper is intended to review the applications of Voronoi diagrams to a wide variety of spatial adjacency problems, with particular emphasis on applications in automated cartography and geographic information systems. Topics covered include: what is a Voronoi diagram? What is the current research in this field? What can they do for us? How may they be implemented - firstly involving the implementation of general polygon structures and their duals, and secondly referring explicitly to Voronoi polygons rather than general polygons? Reference is made to the significance of boundaries in these general polygon structures, and then to the construction techniques for Voronoi diagrams. A comparison is made between the construction of general two dimensional triangular networks and the more conventional one dimensional linked-lists and trees familiar to computing science. Having provided a general background, discussion then covers a variety of applications, including interpolation, skeleton encoding, and spatial ordering.

WHAT ARE VORONOI DIAGRAMS?

Consider a set of objects (points) in the plane. Each of these objects may be considered to have a sphere of influence, defined as the region which is closer to that object than to any other object. The result of this zoning activity is to partition the plane into a set of polygonal regions, each region associated with a particular object. For points in the plane these polygonal regions can be shown to be convex polygons. The result of this process is referred to as a Voronoi tessellation.

While the mathematical definition is straightforward it must be emphasized that Voronoi diagrams are not at all abstract entities. They may be created by the use of blotting-paper and wicks, the magnetic fields of adjacent magnets, etc. (see Morgan, 1967). Thus Voronoi diagrams are closely related to real physical processes, which simplifies both the visualization of the technique and the potential for the modelling of these physical processes.

Considerable research activity has been dedicated to studying Voronoi diagrams in the last few years. While theoretical algorithms are the particular specialty of the field of computational geometry, the applications aspects have not yet been fully explored. The efficient construction of point Voronoi diagrams in the euclidean plane has been well known for some years, but other particular Voronoi diagrams - using other metrics, furthest-point Voronoi techniques, cases with boundaries, and other special applications - are still subjects of ongoing research. The major sources of information on the topic are the textbook by Preparata and Shamos (1985), and the ACM/SIGRAPH annual proceedings on computational geometry. The approach has various characteristics, which include the use of "divide and conquer" methods to obtain the most efficient construction techniques. These result in methods that are not necessarily the easiest to implement on a computer, and in many cases have not been implemented. Finally, the use of divide and conquer techniques implies the construction of the diagram for the whole data set at one time, rather than permitting the updating of the data set in the process of the application.

VORONOI DIAGRAMS AND CO-ORDINATE GEOMETRY PROBLEMS.

Problems in co-ordinate geometry arise frequently in computer implementations of a variety of science and engineering applications. These are associated with the fact that the specification of geometric x,y co-ordinates for some object being described does not automatically provide information about the relationships between line segments or objects themselves. Thus in both automated cartography and computer aided design the specification of object co-ordinates is not sufficient to link these defined objects together to form a coherent whole. As a general statement, co-ordinates do not of themselves produce relationships, that is: graph theoretical structures relating objects in space. This is due partly to the finite resolution of computer word lengths representing co-ordinates of intersection points etc., but primarily because the two branches of mathematics involved have very little overlap in the problems described here. Graph theoretic techniques require that relationships (adjacency relationships in particular) be previously defined, while the straightforward definition of co-ordinates in conventional geometry provide no information of itself about the linkage between points and objects in space.

It is suggested in this paper that the use of a Voronoi generating process may simplify the transition from co-ordinate based information to graph theoretic (adjacency) based structures. Once graph theoretic structures are available many otherwise difficult processes may readily be implemented on the discrete machines available for computing problems. The rest of this paper will therefore be concerned with the storage of polygons in a computing environment, the specific issues of creation and storage of Voronoi polygons in the computer, and applications that ensue from the availability of the resulting structures.

THE STORAGE OF GENERAL POLYGON INFORMATION

Given any map composed of polygons, several things should be noted. Firstly, the two dimensional plane is entirely covered by adjacent polygons: there are no gaps. Thus every polygon has an adjacent polygon, with special care being taken at the boundaries of the map. Secondly, in the two dimensional plane there are only three basic classes of objects: points or nodes (zero-), arcs (one-) and polygons (two-) dimensional objects. Thus a polygon may be defined by its several boundaries, by its several nodes, which occur at the junctions between boundaries, and also by the several adjacent polygons that bound it. Line segments or arcs may be defined in terms of the two end points (nodes), and also the "left polygon" and "right polygon". Information about nodes could include all of the arcs or boundary segments that meet at it and in addition all of the polygons that themselves meet at that node. A useful summary of the options for storing the relationships between polygons, arcs and nodes may be found in Gold (1988a).

A polygon set is in fact a graph. A graph is formed of regions, edges, and nodes, which are directly related to the polygons, arcs and nodes previously discussed. Graph nodes have a valence associated with them - that is, the number of edges that meet at that node. In a two dimensional planar graph, such as a map, most nodes will have a valence of 3. All nodes with a valence of 4 or more may be reduced to nodes of valence 3 by inserting dummy line segments of zero or near-zero length into the data structure. Thus if we can restrict ourselves to nodes of valence 3, all polygons may be represented by the dual triangulation. The dual of a graph is formed by replacing all regions (polygons) with nodes; replacing all nodes with regions; and replacing all edges (boundaries between adjacent polygons) with new edges that connect the "centres" of each original region. Thus polygons convert to nodes, nodes convert to triangles (since they are all of valence 3) and edges convert to new edges.

Figure 1 shows a polygon set and the associated dual triangulation. Polygons A through F are represented in the dual by nodes A through F. Each triangle edge represents or shows the existence of an original polygon edge, and any property associated with that original polygon-polygon boundary may now be associated with the new triangle edge. Thus in a computer structure the triangle edge may point to the x,y co-ordinates forming the irregular polygon-polygon boundary and may also inform the user of the kind of boundary involved. It should be noted that the boundaries need not be simply hard lines as is conventionally represented on a map, but may involve other properties such as fuzziness, faintness, convolutedness, or even flow between adjacent polygons. Thus a boundary - and here a triangle edge - represents a relationship between two adjacent polygons. This relationship may be of any type required by the application. Thus if a soil type map is known to have gradational boundaries between soil types, as is usually the case, and if the soil scientist can describe this gradational relationship, the data structure is capable of preserving this information for future use.

Thus a triangulation structure permits the storage of information concerning polygons, arcs and nodes. A triangulation is one appropriate data structure, since in the two dimensional plane nodes are usually of valence 3. Thus a <u>triangulation network</u> is a <u>relationship storage device</u>. One of the advantages of preserving triangulations rather than polygon sets in the original form is that triangulations have a known number of vertices and edges, simplifying internal storage concerns in a computing system. One possible way of storing a triangulation data structure is to preserve the three adjacent triangles and the three vertices for each triangle record (see Gold et al., 1977). In that particular case triangle edges are not themselves preserved. Another alternative is to preserve the triangulation as a series of edges rather than as a series of triangles: each edge record consists of a "from" node , a "to" node and the next edge record clockwise (or anti-clockwise) from each end node. This particular data structure is also of fixed length, and hence of simple implementation, but in addition detailed information about the boundary itself between any two polygons may readily be added. See Gold (1988a) for more details on the selection of data structures. While both of these data structures, as well as variants, are appropriate formats for the storage of the dual triangulation of a polygon set, the line record format appears to be better where arbitrary boundaries are involved, whereas the triangle record format, while not preserving any specific boundary information, seems to be particularly appropriate to the storage of Voronoi diagrams, where polygon boundaries are not arbitrary but are implicit in the relationship between the two adjacent map objects. As will be seen, in the preservation of Voronoi polygons in a computer data structure, the storage of the junction between the three boundary segments is the most useful property to preserve, and one of these "circumcentres" exists for each triangle.

THE IMPLEMENTATION AND STORAGE OF VORONOI POLYGONS

We have discussed the storage of general polygons and some of the possible data structures to use. As previously mentioned, the triangulation data structure appears appropriate for the preservation of Voronoi polygons in particular. Figure 2a shows a simple set of points in the plane, their associated Voronoi polygons (solid line), and the resulting dual triangulation (dashed line). In this particular case, rather than using divide and conquer techniques to generate the whole Voronoi diagram at once, individual points are inserted one at a time into the data structure. In Figure 2a, a new data point, marked X, is to be inserted into the data set. Figure 2b shows the results of inserting the new point and in consequence creating a new polygon at the expense of the previously existing polygons. Figure 2c shows the portions of the previous polygons "stolen" by the new Voronoi polygon. This simple insertion technique is theoretically less efficient than divide and conquer methods, but it is simple to implement and cost-effective for all but the largest data sets. In addition, the ability to insert and delete individual points is crucial in many applications. Figure 3 shows the result of generating the Voronoi polygons, and dual triangulation, for a test data set from Davis (1973).

The objects inserted into the two dimensional plane need not be restricted to points. Figure 4 shows the case where individual points and line segments are inserted. Some increased complexity therefore exists - in particular, while the boundary equidistant between two adjacent points is a straight line, and the boundary between two adjacent line segments is also a straight line, the boundary between a point and a line segment forms a parabola.

In order to insert a line segment into a Voronoi diagram, first of all the two end points must be inserted as described previously, and then the connecting line segment itself added. This is consistent with the fact that connecting the two end points adds additional information to the map. The Voronoi region for a line segment therefore has boundaries that consist of straight line segments and parabolic segments, and it need not necessarily be convex. Figure 5 illustrates the insertion of a line segment into the Voronoi diagram.

Figures 4 and 5 also show the triangulation of the Voronoi regions. Point objects are represented as solid dots and line objects are represented as dashed lines. Line segments are considered as distinct objects from their end points. Since the Voronoi regions are in fact polygons the result is a triangulation as described previously for general polygons. Since the Voronoi regions are entirely defined by the relationships between points and points, points and lines, or lines and lines, there is no need to save explicit *boundary information and* thus no need to implement the line segment data structure previously described. On the other hand, the junctions between line segments and parabolic segments are of considerable importance, and as one of these junctions is associated with each triangle, a triangulation based data structure appears appropriate for this problem. In a simple point-Voronoi diagram the junction of these three boundaries is at the circumcentre of the particular triangle. When the Voronoi diagram is extended to include line segments a similar definition holds: the centre is at an equal distance from the three objects at the vertices of the triangle. Thus an appropriate circle would pass through any vertex that consisted of a data point, and would be tangent to any vertex consisting of a line segment. Nevertheless, all triangles in this structure have a circle centre and radius representing the maximum distance one can get away from each of the three vertex objects. For further details see Gold (1988c).

In conclusion boundaries are implicit between objects of known type, therefore Voronoi boundaries need not be explicitly preserved. The intersections of Voronoi boundaries define the available valid relationships. Thus the dual triangulation data structure with "circumcentres" should be preserved to define adjacency relationships based on euclidean distance.

RELATIONSHIPS BETWEEN GENERAL POLYGONS, DUAL TRIANGULATIONS AND ONE DIMENSIONAL LINKED LISTS.

Fundamental operations on one dimensional conventional linked lists include the following basic operations. Firstly: an initialize process, usually involving setting up two end nodes with values selected to be outside the range of the data to be inserted. Secondly: an insert operation, permitting the insertion of a new node between two previous nodes. These nodes, in a linked list application such as simple sorting, would each consist of a left pointer, a right pointer, and a value field - probably containing one of the numeric values to be sorted. Assuming that the linked list is to be maintained in ascending numerical order, a search technique must be available to determine whether a particular numerical value has either already been inserted, or alternatively to determine the values immediately below and immediately above the new value to be inserted. This search algorithm could involve either a simple "start at one end and keep looking until you get there" process, or a more elaborate binary search. A third necessary linked-list operation would be a delete procedure, permitting the deletion of a particular value no longer desired, and the elimination of the associated node in the linked list. Finally, in some cases (e.g. a bubble sort) a "switch" operation may be of value. This operation switches the values of two adjacent nodes. All of these operations, with the exception of the search, are of O(n) efficiency. The efficiency of the search technique itself may vary from O(n**2) for a simple minded "read the whole list", to O(n log n) for either a binary search technique or else a tree search - if it has been considered desirable to include a hierarchical tree structure above the one dimensional linked list.

In the case of a set of general polygons (not specifically Voronoi) we can create an equivalent set of operations. An initialization operation consists of defining a large exterior polygon, such as a map boundary, enclosing all subsequent data. This region will be divided into a space-covering polygon set as data is inserted or deleted. A partially-completed polygon set is shown in Figure 6a. The dual triangulation is also illustrated. Note that each node in the dual triangulation represents one of the original polygons, and each triangle in the dual triangulation has one associated node (with valence 3) in the original polygon diagram.

In Figure 6b the central polygon has been divided into two by a new boundary. The result of this operation is to create one new boundary segment and two new 3-valence nodes. Thus in the dual triangulation representation two new triangles have been created. This "split" process may be replaced by a reverse "merge" process. In this case a boundary between two adjacent polygons is deleted, and hence two polygons become one In the dual triangulation representation two adjacent triangles are deleted, and the two nodes on their common boundary are merged into one.

We may therefore consider the equivalent of a simple insert process in a one dimensional linked list to be a split process in the two dimensional polygon context. Thus rather than "inserting" a new node we are splitting one node into two. This is appropriate since in the polygon problem it is assumed that the whole plane is tiled in polygons. The equivalent of a one dimensional delete process is the merge process described above for the polygon problem. Thus for any general polygon set we have the equivalent of insertion and deletion in a one dimensional linked list. In addition, this is readily implemented in the dual triangulation of the space-covering polygon set.

An additional property of this insert/split approach is that it allows us to subdivide space in a hierarchical tree fashion without imposing any specific restrictions on the shape of any particular set of polygons. Thus the insert (or split) process involves the taking of the initial polygon, let us call it AB, and splitting it into two sub-polygons A and B. In terms of conventional tree structures this produces a binary tree with all polygons at the leaves. The delete/merge process takes two leaf polygons A and B, deletes them both and replaces them with their common parent polygon AB, which itself becomes a leaf.

The tree structure previously suggested is directly relevant to problems concerning the order of efficiency of the search process. The simplest one dimensional search technique is merely to "walk" through the linked list starting at one end until the appropriate value in the ordered list is found. For multiple searches it is reasonable to continue the new search from the point of termination of the previous one. This local walk technique can be applied to a triangulation in two dimensions. For details see Gold et al. (1977). This walk through a triangulation in two dimensions is approximately of O(n**1.5), as opposed to O(n**2) for the one dimensional case. The walk in two dimensions is based on geometric criteria - thus it is readily used in the case of Voronoi polygons and dual triangulations, where the geometric relation between the triangulation and the dual polygons is straightforward, but the approach is less obvious where the dual triangulation is of a general polygon set, the boundaries are arbitrary and it is unclear where the appropriate "centres" of the original polygons should be.

Nevertheless for the Voronoi polygons a simple geometric walk is readily implemented and reasonably efficient under most circumstances, since data on input is usually naturally ordered by the process of acquiring the data in the first place: thus there is a tendency for the next data point to be inserted into the data structure to be close to the previous one. Where a higher order of efficiency is desired the binary tree structure previously mentioned may be implemented. Note that no rules have been given as to precisely when two polygons should be split or merged. This would be a function of the particular mapping information desired. It is therefore flexible, but does require implementation of splitting and merging rules based on knowledge of the data. It is nevertheless the same technique - whether applied to simple hierarchical subdivision by map sheet, subdivision by census district, county and higher order region, or any other desired natural hierarchical order to the polygon data.

The last of the processes to be described is the "switch" operation. Any two adjacent triangles will have a common boundary. The quadrilateral formed by these two triangles may be divided into two triangles either in the original fashion or by connecting the two opposing points - thus changing the diagonal of the quadrilateral. This was previously described in Gold et al. (1977). The switch operation is equivalent to the switching of two nodes in a one dimensional linked list. However, in order to decide whether a pair of triangles should be switched in any particular case, an appropriate criterion should be used. The most appropriate criterion is generally accepted to be the Voronoi. On this basis triangles perturbed by nearby network modifications may be tested to see if an adjustment (switch) is required to preserve the Voronoi property. Thus the testing and switching of all edges of the triangulation that have been modified by insertion or deletion, or by the switching of nearby edges, permits the ready preservation of the Voronoi property for any object insertion or deletion. This operation can be guaranteed to be a local process - in fact on the average the insertion of a new data point can be expected to cause 6 switch operations to be performed. Thus no insertion or deletion in one corner of a map sheet can have any influence on remote portions of the triangulation.

We have thus shown for the case of the general triangulation the relationships that exist between the basic operations of initialize, insert, delete, switch and search in the one dimensional linked list case well known to computer science, and the two dimensional triangulation case which may be applied to any space-covering polygonal set. In the special case of the Voronoi polygons the switch operation can maintain the Voronoi criterion subsequent to any perturbation of the network by insertion or deletion processes.

APPLICATIONS

The primary function of the implementation of the Voronoi tessellation for a set of points or line segments is to allow coordinate geometry problems to be approached from the graph theoretic viewpoint. Some specific applications are given.

Figure 7 is taken from Gold and Cormack (1987). The ordering techniques were first described in Gold et al. (1977). If a triangulation has been formed by the previously mentioned techniques (not necessarily Voronoi) it is possible to perform operations upon triangular elements of this network in a spatially consistent order. In the example shown a viewpoint labelled X is located near the centre of the triangulated data set. After the first triangle has been processed there remain three adjacent triangles. Each of these may be processed in turn. These subsequent triangles have either one or two adjacent triangles that are further away from the viewpoint than they are themselves. By appropriate geometric tests, described in Gold and Cormack (1987), it is possible to process each triangular element in a nearest-to-furthest order with respect to the specified viewpoint. Thus, since the triangulation may be ordered, so also may the objects from which the Voronoi polygons, and the dual triangulation, were produced. This permits the general solution of a variety of adjacency problems. Hidden line problems may be processed in a front to back or back to front ordering with respect to the eye position by following this procedure. For pen plotter applications pen movement may be minimized by processing the map in an order based on the triangular patches formed by the triangulation process. Radial searches outwards from the viewpoint are readily performed using the technique, permitting easy retrieval of all data objects close to the desired starting location. This graph theoretic approach is particularly desirable where a selection of neighbours is required, as in interpolation.

In interpolation problems, such as traditional contouring or perspective view modelling, it is difficult to generate an interpolated surface that will always honour every data point, whatever their distribution. Figure 2a shows a simple Voronoi tessellation of a small point data set. Figure 2b shows the result of inserting a new data point, marked X. This new point however is not a "real" data point, but simply a sampling location where an elevation value is desired. Figure 2b shows the new polygonal region carved out from the Voronoi polygons of the real data points themselves. Figure 2c shows the areas of each of these polygons "stolen" by the Voronoi polygon of this new dummy point. These stolen regions are of considerable interest, as they permit straightforward interpolation between arbitrarily distributed data points. The relative areas stolen from adjacent data points are used as weighting functions to generate a weighted average of these adjacent points, to form the estimated elevation at the point marked X. A particular strength of this approach is that *only neighbouring* data points which have a finite positive area stolen from them are defined as neighbours to the interpolation point X. Thus no discrepancy exists between the selection of the neighbouring points and the weighting function used upon them (see Gold, 1988b, 1989).

As an additional application, Figure 8 shows a map of a small village region. A variety of roads, houses, streams etc. are displayed. In any geographic information system it is frequently desirable to be able to determine which map entities are adjacent to which other map entities. An example would be to determine which houses are adjacent to a particular road. It is of course possible to generate Voronoi zones about each object defined on the map. First of all it is necessary to break up certain features such as roads into individual segments - this is a cartographic problem not addressed in this paper. The result of constructing the Voronoi diagram of the major objects on this map is also shown. On the basis of this Voronoi diagram it is possible to make reasonable statements about whether a particular house, shed, etc., is adjacent to a particular road, or to another building. The answer to this question would be "yes" if the Voronoi regions of the two objects under query are adjacent to each other and have a common boundary. Indeed the extent of the common boundary between them could be a measure of the adjacency itself. Note that in a few cases, e.g. where a stream goes under a road, nodes with an order of 4 as opposed to an order of 3 may be found on the Voronoi diagram. As before, this Voronoi diagram can be expressed as a dual triangulation. For details see Gold (1987).

A final application concerns the skeleton encoding of polygons. Figure 9a shows a polygon with one concave vertex. A "wave-front" analogy has been used to show the growth inwards of parallel bands along the boundary itself. Figure 9b shows the result when these wave-fronts have met and completely engulfed the original polygon. Each line segment on the original boundary now has associated with it an interior region bounded by edges formed where the various wave-fronts met. These regions are the interior components of the Voronoi region for each of the line segments of the boundary (and as such have a valid dual triangulation). In the case of the single concave vertex shown, an interior region is defined for the vertex itself, and not merely for the line segments involved. (Figures 4 and 5 illustrate point and line Voronoi diagram generation.) This interior boundary between a convex vertex and the opposing line segment generates a parabolic interior segment to the polygon skeleton. (This example is taken from Brassel and Jones, (1984), where "bisector skeletons" perform a similar operation.) This polygonal skeleton is of value as a graph theoretic description of the general shape of the polygon, and as such (in raster mode) is frequently used in character recognition applications. In the field of cartography the technique is of value as a label or name placement aid.

CONCLUSIONS

It is hoped that this paper has shown the basic relevance of the Voronoi tessellation as an aid in converting co-ordinate geometric problems to graph theoretic approaches. On this basis a large variety of applications may be attacked using a common set of tools. The basics of the approach have been described along with appropriate data structures, and several applications have been outlined. Other applications are expected to be developed in the near future.

ACKNOWLEDGEMENTS

The funding for this research was provided in part by an operating grant from the Natural Sciences and Engineering Research Council of Canada, and in part from the Energy, Mines and Resources Canada Research Agreement Programme.

REFERENCES

Brassel, K.E. and P.L. Jones, 1984, The construction of bisector skeletons for polygonal networks. IN: Proceedings, International Symposium on Spatial Data Handling, Zurich 1984, v.1 pp. 117-126.

Davis, J.C., 1973, Statistics and data analysis in geology, (New York: John Wiley and Sons), 313p.

Gold, C.M., 1989 (in press), Surface interpolation, spatial adjacency and G.I.S. IN: 3-D G.I.S., (J. Raper, ed.), (London: Taylor and Francis Ltd.)

Gold, C.M., 1988a, PAN Graphs: an aid to G.I.S. analysis. International Journal of Geographical Information Systems, v. 2 no. 1, pp. 29-42.

Gold, C.M., 1988b, Point and area interpolation and the digital terrain model. IN: Trends and concerns of Spatial Sciences, Second Annual International Seminar - Proceedings (June 1988) pp.133-147 + discussion (Y.C. Lee, ed.).

Gold, C.M., 1988c, Further research on Voronoi diagrams - a common basis for many applications. IN: Trends and concerns of Spatial Sciences, Third Annual International Seminar - Proceedings (Y. Bedard, ed.).

Gold, C.M., 1987, Mapping is a diffusion problem. IN: Proceedings of Carto-Quebec/Canadian Cartographic Association Meeting, Quebec, May 1987, pp. 332-362.

Gold, C.M. and S. Cormack, 1987, Spatially ordered networks and topographic reconstruction. International Journal of Geographical Information Systems, v. 1 no. 2, pp. 137-148.

Gold, C.M., T.D. Charters and J. Ramsden, 1977, Automated contour mapping using triangular element data structures and an interpolant over each triangular domain. Computer Graphics, v. 11, June 1977, pp. 170-175.

Morgan, M.A., 1967, Hardware models in geography. IN: Models in Geography, edited by R.J. Chorley and P. Haggett, (London: Methuen), pp. 727-774.

Preparata, F.P. and Shamos, M.I., 1985, Computational Geometry, (New York: Springer-Verlag), 390p.

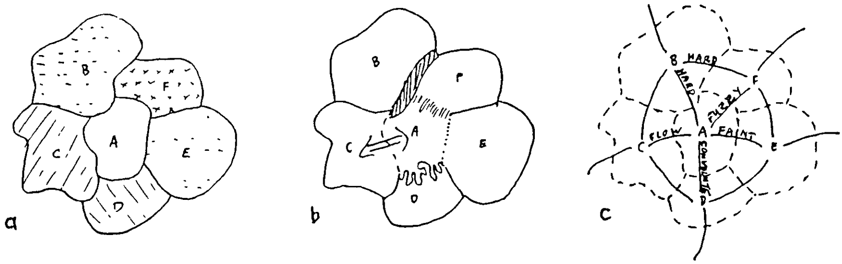

Figure 1. a) Polygon set.
b) Possible boundary properties.
c) Relationship triangulation (dual graph).

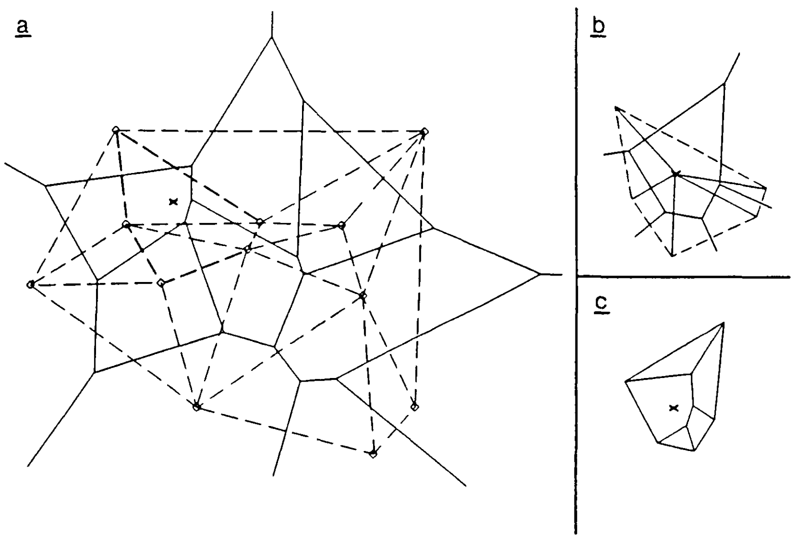

Figure 2. a) Point-Voronoi diagram and dual triangulation.
b) Introduction of point X.
c) Areas stolen from neighbouring regions.

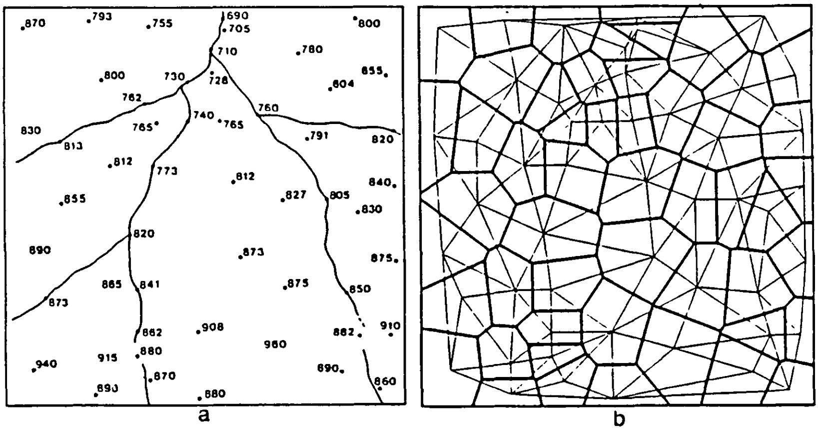

Figure 3. a) Elevation data from Davis (1973).
b) Resulting Voronoi regions and triangulation.

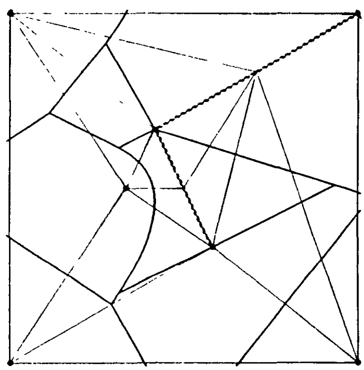

Figure 4. Voronoi regions for points and line segments.

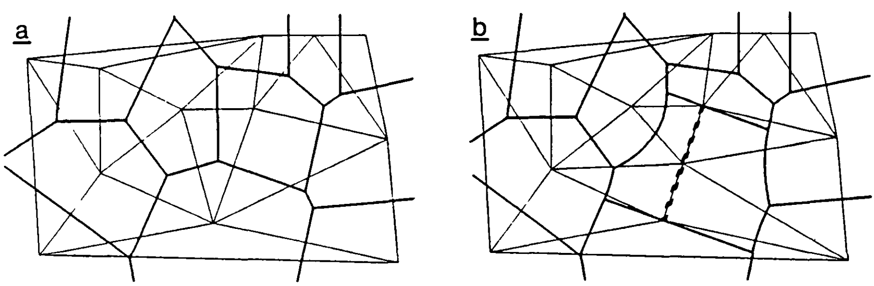

Figure 5. a) Point Voronoi regions.
b) Insertion of a line segment.

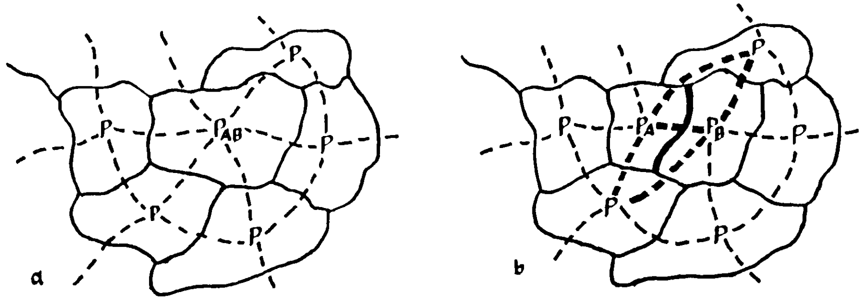

Figure 6. a) General polygon set with triangulation.
b) Result of splitting P(ab) into P(a) and P(b).

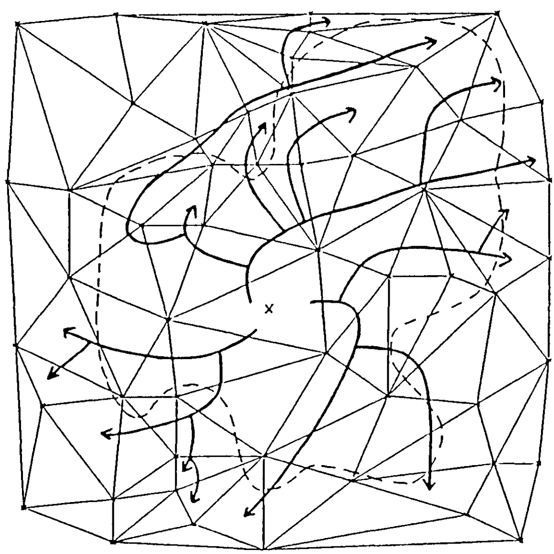

Figure 7. Triangle ordering from viewpoint X.

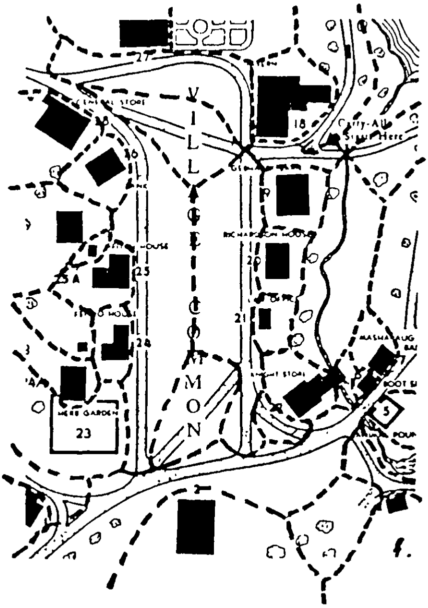

Figure 8. Map, showing map-objects and Voronoi regions.

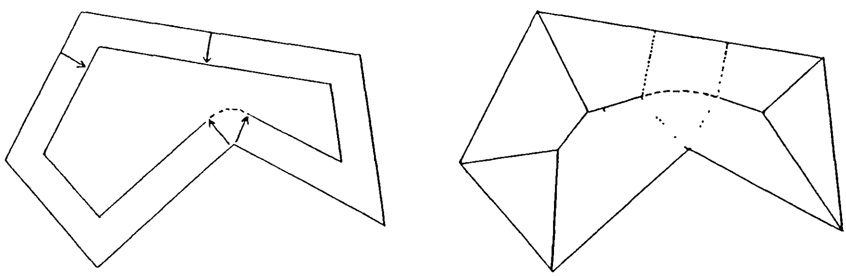

Figure 9. a) Polygon, showing wave-front propagation.
b) Internal Voronoi regions of polygon boundary.