This review is intended to supplement the Spatial Data Handling Symposium Workshop by giving the main lines of argument leading to the development of the Voronoi spatial model. Three points will be emphasized. Firstly, the objects used to create a seamless data structure (i.e. one without holes or joins) are themselves discrete, even if the world they came from is not. Secondly, a flexible data structure should be "dynamic", permitting the immediate insertion and deletion of individual map objects without destroying the underlying spatial linkages (which would then require a complete "rebuild" operation for even small changes). Thirdly, and perhaps most importantly, there are other ways of thinking about spatial data structures than the raster model, or the point/line/polygon model of the last fifteen years.

Without delving too far into the history of science it may be true to say that for the last few centuries whenever spatial problems have arisen the immediate response has been to visualize a square grid superimposed over the area of interest. While appropriate for many tasks, it is not the universal panacea, and this mind set may preclude the visualization of more appropriate approaches. For the purposes of this presentation cartesian coordinates are required solely to specify the location of a map object, and all further work is done using the more local concept of distance. In other words, instead of remembering elementary-school graph paper whenever a spatial problem occurs, elementary-school geometry (with compass and ruler only) should be recalled.

Given these assumptions (closer to the real-world situation) what can we say about spatial data handling? For the purposes of this workshop the main ideas will be given as a set of propositions, grouped to cover basic issues concerning spatial data structures. It is hoped that these propositions, taken in sequence, will provide a basic tutorial on an alternative way of visualizing space. They are, however, intended as supplementary material to the workshop presentation, rather than complete in themselves.
A. Polygons and duals: two dimensional space can be represented by triangles. (Figure 1)

1. Polygons on a map are formed from interconnected vertices and edges. In order for a polygon to be topologically complete pointers must exist between vertices (if defined) and edges. The resulting regions on the graph may then be labelled.

2. Each vertex (node) in a polygonal map can be forced to have a valence of three by creating an imaginary zero-length edge and splitting the original node. In the natural world more than three regions do not usually meet at a point, so this splitting would be needed only infrequently.

3. The dual of the modified polygon set is a triangulation, where all polygons are represented in the triangulation by nodes and all nodes (boundary junctions) in the polygon set are represented as triangles. The original arbitrary boundaries between adjacent polygons are replaced by triangle edges representing an adjacency relationship between two polygons. (See Gold, 1989b.)

4. Triangulations in this context express relationships between triples of objects - in this case polygons. A triangle edge means that two polygons are adjacent (have a common boundary). This boundary may be real, or it may express some other form of adjacency relationship: a gradational (rather than abrupt) change; interregional flow; etc.

5. Triangulations may readily be stored as fixed-length records storing the three vertices, the three adjacent triangles and, if required, the three bounding edge record numbers for each triangle. (See Gold et al. 1977 or Gold 1978.)

6. An alternative to a triangulation as a basic record type is a line segment. This is also of fixed length, storing pointers to the two end vertices and the two (anticlockwise) adjacent line segments. Both line segments and triangulations are valid data structures whose relative advantages are minor and depend on the application.

7. As described in Gold and Maydell (1978) and Gold and Cormack (1987), any triangulation may be processed as an ordered binary tree with respect to some viewpoint, permitting front-to-back or radially-outwards ordering of objects on a map. This provides a partial ordering of points in coordinate space.

B. Concepts of Spatial Adjacency. (Figures 2 and 3)

8. Adjacency means that two polygons have a common boundary of zero or greater length.

9. Voronoi regions are polygons that may be visualized as "bubbles" growing at equal rates around each map object (point) until all the map space is occupied. Thus any point on the map falls within the Voronoi region of the map object that is closest.
10. If the dual triangulation expresses the adjacency relationships between any set of polygons, then it is an appropriate expression of the adjacency relationships between Voronoi regions. It is thus an expression of the adjacency relations between the original generating data points. This therefore forms a consistent definition of spatial adjacency for any distribution of data points.

11. The objects associated with the triangle vertices need not be points - they may be any objects: in particular, in this application, points plus line segments. (See Gold 1989b.)

12. As the act of connecting two end-points with a line segment adds information to the map, the interior of a line segment is a separate object from its end points, and itself forms a vertex of several of the dual triangles. Conventional map polygons may be constructed from points plus line segments.

13. The Voronoi region for any object is defined in the same manner as for points (item 9), and may readily be calculated. The Voronoi boundary between any two map objects is equidistant from each. Between a point and a line the boundary forms a parabola; between point pairs and line pairs the boundaries are lines.

14. The boundaries between Voronoi regions are implicit in the relationship between any two adjacent vertices (objects) in the triangulation, and need not be preserved, as they may be recalculated whenever required. The centre of the triangle (i.e. the junction between three Voronoi boundaries, or the point that is equidistant from all three triangle vertices) is more critical in determining which boundaries are to be preserved to form the triangulation. In this case a triangulation structure appears preferable to a line-segment data structure.

15. Any set of map objects (here points plus line segments) may thus have Voronoi regions constructed around them, and these objects are adjacent when their regions have a common boundary. The most convenient way to preserve this adjacency information is in the dual triangulation. For a point data set this is referred to as the Delaunay triangulation.

How is this spatial data structure built? To understand this we need to examine the basic problems of representing geometric information in the digital computer.

C. Strengths and Weaknesses of Digital Computers

16. Digital computers operate in a discrete (rather than a continuous) fashion. Thus some of the most successful computer applications involve the representation of discrete objects (e.g. people) with discrete pieces of memory (records) at discrete locations (addresses). Where there are relationships between records, pointers are defined.

17. In spatial analysis, graph-theoretic models and applications are increasingly valuable tools. Spatial data structures, with pointers between map objects, are themselves graphs, and hence are readily handled in a discrete computing system.
18. Coordinate information is initially used to specify a map object's location, but one particular weakness of discrete computers is well-known: the inability to provide exact results for line intersection and similar problems. (For example, after calculating the coordinates of the intersection of two lines, the resulting point will not usually fall precisely on either line.) Consequently geometric (co-ordinate) operations can not guarantee consistent graphical structures (topology) except with considerable care.

Given these properties of the digital computers with which we work, how should we proceed to define our model of space? With care it is possible to use the Voronoi concept to define a discrete space-covering tiling, and then to use graph-theoretic and linked list operations upon these tilings.

Spatial Data Structures and Linked Lists. (Figures 4 and 5)

19. In order to handle geometric problems in a discrete manner it is necessary, for large data-sets, to maintain node-and-pointer data structures internally, where the nodes represent map objects (points or line segments) and the pointers represent adjacency relationships between objects. In a one dimensional example, of an ordered set of numbers, this would be implemented as a traditional linked list. In two dimensions these relationships would be implemented as a triangulation, in three dimensions as tetrahedra, etc. There is a direct equivalence between linked lists and spatial data structures based on the Voronoi spatial model.

20. One set of basic operations for linked lists is -- initialize, insert, delete, move and search. Direct equivalents exist for triangulations.

21. The initialize operation consists of creating a bounding triangle to enclose the subsequent data set. Much the same is used in one-dimensional linked lists when initial starting and ending nodes are given very large negative and positive values.

22. Insert and delete operations are equivalent to splitting and merging adjacent (non-Voronoi) polygons. If polygon AB is to be split into polygon A and polygon B, a new boundary line must separate them, and this must be reflected by the creation of two new dual triangles. Conversely, if A and B are merged to form AB then one boundary and two dual triangles must be deleted.

23. For Voronoi regions the insert operation consists of finding a data point close to the desired final location and splitting the Voronoi region into two (as described in item 22) thus producing two region and generating a new data point directly adjacent to the original. This new point is then moved to its final location.

24. The Voronoi delete process is the reverse of insertion. After moving the desired point to a neighbouring point's location, the two are merged.

25. Moving a data point through an existing adjacency network consists of a series of switch operations. This is directly equivalent to the exchanging of nodes during the insertion of a single new value into a linked list using a bubble-sort.
26. The switch operation is performed whenever the common edge between two adjacent triangles does not generate the correct Voronoi region boundary, due to point movement. Switching consists of taking a pair of adjacent triangles forming a quadrilateral and exchanging the incorrect diagonal for the alternative one.

27. A search operation consists of finding the map object that is closest to the query location. This is equivalent to finding the Voronoi region within which the point falls, or to performing a sequential search in a one-dimensional linked list.

28. A Voronoi search, unlike a Voronoi move, does not modify the data structure. It consists of examining the map objects adjacent to some starting object, finding one with a smaller minimum distance between it and the query location, moving to that object, and repeating the process until no closer object is found.

29. While sequential moves and searches are inefficient in one dimension, as the dimensionality increases the efficiency improves, as it is not necessary to examine all map objects (data) while traversing the map from one side to the other. Where sequential searches need to be even more efficient, one of several additional data structures, e.g. quad-trees, may be employed.

30. In the same way that higher-dimensional data-structure operations may be considered as extensions to the one-dimensional linked list, this linked list may be considered to be a one-dimensional Voronoi structure.

31. A line segment is merely the locus of a moving point. Thus creation of a line segment consists of splitting an existing point (one end of a line) and moving the new end point to the desired final location - with the additional provision that a line segment interior remains in existence between the two, holding all the spatial adjacencies possessed by the moving point during its travels.

E. Applications of the Voronoi Spatial Data Structure. (Figures 6 and 7)

32. Interpolation may be performed by the judicious insertion and deletion of dummy sampling points in order to determine the relative areas of the adjacent Voronoi regions stolen by the new dummy point. The estimated value at the dummy sampling point is obtained by taking the weighted average of the neighbouring data points, where the neighbouring data point values are weighted by the stolen areas. Gold (1989a, 1990a) discusses the Voronoi approach to contour mapping and interpolation.

33. As described in item 31, line segments are constructed from their two end points and a connecting link. If these end points and the line segments are inserted into the Voronoi network they will each generate their own Voronoi region. For line segments connected to form a polygon, the interior boundaries of these regions form the skeleton or medial axis transform of the polygon. This is of value for label placement and other applications.
34. The Voronoi regions of each edge of a polygon express the areas that would be stolen by adjacent polygons if the central polygon were removed. Thus the area-stealing interpolation methods previously described for point data sets are equally applicable to information associated with polygon sets.

35. Polygon map construction is directly obtained from the Voronoi regions or the dual triangulation. Starting from some labelled seed point inside a polygon, a simple graph traversal will identify all edges forming the polygon boundary. Direct edge colouring methods permit the detection of many mistakes in data entry. Polygons only exist as collections of similarly labelled Voronoi regions of points and half-lines.

36. Point-in-polygon problems are directly resolved by performing the search procedure described in item 28. This determines the closest object to the desired location, and its Voronoi region. This information is sufficient to determine the closest polygon edge or vertex, and hence the appropriate enclosing polygon may be obtained. Details may be found in Gold (1990b).

37. Polygon editing involves the insertion and/or deletion of a small number of line segments or points at a time. As the Voronoi construction procedure is dynamic, the updating of the spatial data structure may be performed as part of the digitizing process, and any errors signalled immediately.

38. Polygon overlay processing consists firstly of creating two separate Voronoi maps, one for each overlay, and then editing and graph-colouring the edges. The resulting polygon boundary segments thus have one polygon, and hence one colour, on each side. The overlay process consists of tracing one map onto the other without inserting it, thus determining the new polygon or colour to be associated with a line segment, as well as any intersections. After repeating this from the second map onto the first, the two sets of fully coloured and cut line segments are assembled to form a new polygon coverage. Island polygons require no special handling.

39. Corridor generation is directly obtainable from the Voronoi regions of any polygon or line set by performing a graph traversal outwards from the central object or polygon for the desired distance, ignoring those Voronoi zones that terminate too close to be used. The resulting corridor boundary is composed of line segments and circular arcs, and has no crossings.

Conclusions

The concept of Voronoi diagram construction for point and line objects provides a good general framework for a wide variety of spatial data processing applications.

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.
Because the Voronoi spatial model does not depend on line intersections to capture spatial adjacency, spatial operations may be performed on partially complete data sets, as well as those with no intersection information available at all -- e.g. point data. This, combined with the dynamic nature of spatial data structure modification, permits relatively simple data compilation and local spatial query. In the context of this workshop, its particular strengths lie in the generation of consistent spatial data structures that may readily be transformed into the data structures required by other systems, in particular those described in the other parts of this workshop presentation.

References


