Within the last fifteen years or so in the field of automated cartography (AC), and more recently geographic information systems (GIS), a body of experience has been built up concerning the development and use of a variety of spatial algorithms. Operations on a single coverage of a map, for example, have been developed on an ad-hoc basis both in vector and raster modes. Within the AC/GIS environment there is now some agreement on the basic required procedures, but not on the algorithms to be used. It is perhaps time to review progress and consolidate methodologies.

Single coverage operations form the basis of AC/GIS and are a frequent source of confusion as they require both coordinate and graph operations, and transformation between the two. The Voronoi diagram has many of the desirable properties of both metric and graph methods, but until recently computer methods have only been available for point-type map objects. It can now be computed for both point and line objects, and generated in an incremental update fashion (e.g. inserting, deleting or moving single map objects) compatible with the most commonly desired map operations. The basic operations on this Voronoi spatial model may be directly equated with the operations traditionally performed on a one dimensional linked list, extending these well known operations to higher dimensions.

Given the availability of a dynamic updatable Voronoi diagram of both point and line objects, many of the commonly required single coverage AC/GIS operations may be performed in a consistent and straightforward manner. Examples of these include point in polygon problems, corridor or buffer zone calculations, polygon build and close operations, polygon skeleton generation, direct search for the closest map object, and polygon overlay by the updating of one of the coverages. In addition, well behaved interpolation may be performed on a variety-of-map data-types, such as point and line objects, as well as providing a consistent method of interpolation for continuous surface modelling, slope estimation, discrete surfaces such as in choropleth maps, and interpolation (or the "missing census district" problem. The ability to form a consistent definition of the spatial adjacency of map objects is of value to implement many of the above mentioned operations, and also as a tool for interrogation of the resulting map.
Introduction — Problems and Overview

By introducing computers, with their discrete logic, to the problems of digital maps we inevitably introduce various strengths and weaknesses into our basic operating environment. One particular weakness of discrete computers is well-known: the inability to provide exact results for line intersection and similar problems. Consequently geometric (co-ordinate) operations can not guarantee consistent graphical structures (topology) except with considerable care. Discrete mathematics, however, especially graph theory, becomes an increasingly valuable tool. Recent work in computational geometry, although of an increasingly theoretical nature, provides many of the basic insights into the feasibility of various algorithms for geometric operations. Of particular interest are studies of the generation of various classes of Voronoi diagrams.

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 graph-structured form amenable thereafter to graph-traversal and other fundamental discrete operations. Voronoi diagrams of points and line segments may be generated by inserting or deleting one object at a time using two dimensional equivalents to one dimensional linked list operations. Once the Voronoi generator has constructed this data structure, conventional graph traversal algorithms (e.g. depth-first and breadth-first searches, shortest path and minimum-spanning tree construction) may be used to answer many basic geographic queries.

General polygons — the basic spatial model

Unambiguous polygon adjacency

In the field of spatial data processing it has always been a problem to specify spatial relationships in a consistent and agreed manner. This is true even for the more restricted case of "spatial adjacency". Perhaps one of the few definitions that would be generally accepted is that polygons with a common boundary are spatially adjacent. This definition is used in the work described here. Thus for any contiguous polygon set, polygon pairs with a common boundary are adjacent.

Dual triangulation "relationships"

It polygon pairs are adjacent there is a "relationship" between them. The set of all adjacency relationships can be expressed as the dual triangulation of the polygon set, where each triangle edge represents one relationship. A triangulation can be used since most boundaries meet at triple junctions, and special cases can be rearranged to
conform. Each relationship so recorded need not merely be the 
presence of a common boundary; it could also express some 
property of that boundary (length, convolution, fuzziness) or 
the presence of some interaction (such as flow between cells).
This is illustrated schematically in Figure 1.

Polygon split and merge

In the case of any set of general polygons we can create a set 
of basic construction 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 2a. 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 2b 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. This action converts Figure 2b back into Figure 2a.

Voronoi polygons

In the case of point data rather than polygon data, the same 
concepts of polygon generation and adjacency apply, except that 
the polygon boundaries are generated mathematically, each data 
point becoming the "nucleus" of a polygon whose enclosed area 
is that part of the map closer to that nucleus than to any 
other (the Voronoi criterion). Program to construct the point 
Voronoi diagram in two dimensional Euclidean space (along with 
the dual Delaunay triangulation) are readily available, using 
either an incremental addition of each new point, or else a 
computational geometry-divide and conquer" process (see Gold, 

As a result of this process, and using the previous definition 
of spatial adjacency, the set of adjacency relationships 
between data points is readily available. Thus "spatial 
adjacency" is a concept that can be applied to disconnected 
points in space as well as to conventional polygonal systems. 
This spatial adjacency data structure for
points is entirely a function of their original location, the Voronoi polygon definition, and the polygon definition of spatial adjacency, and is an intrinsic property of the data. The data structure is the dual triangulation of the generated Voronoi polygons.

Point movement is possible with triangle switching

The generated Delaunay triangulation expresses the common boundaries in existence between the generated Voronoi polygons. If a data point is moved a small distance from its original position it is probable that the a-sit of boundaries will remain unchanged, as will the dual triangulation, even though the individual boundaries will change position. For a larger movement, however, individual boundaries will appear or disappear, requiring the updating of the triangulation recording the existence of adjacency relationships. This updating can be performed, for example, whenever the point moves into the circumcircle of an existing triangle, or whenever the point leaves the circumcircle of a "potential" triangle formed by triples of its immediate neighbours. See Gold (1990a) for details.

In each of these cases a pair of triangles are involved, forming a quadrilateral. The diagonal of the quadrilateral specifies the existence of a Voronoi boundary between an opposite pair of points--one of the two possible choices must be correct (see Figure 3). If one of these vertices moves, e.g. into the circumcircle of the triangle formed by the remaining three points, then the diagonal of the quadrilateral must be "switched" to record the replacement of one Voronoi boundary with the other. Thus movement of a point through a swarm of other points may be performed while preserving the Voronoi criterion throughout by selectively switching adjacent triangle pairs as the point moves. The switch process was previously described by Gold (1978) and Lawson (1977) for building triangulations.

Polygon creation and deletion

Combining this movement process with the previously defined polygon splitting and merging permits the dynamic maintenance of a point Voronoi data structure. To add a new point, a nearby point already in the data structure has its polygon "split" into two: like binary fission of a cell, we now have two cells, with two nuclei very close to each other. Two new dual triangles have been created. The newly generated point is now moved to its final destination as described above. If a point is to be deleted it is first moved next to a nearby point and a "merge" process performed to join the two polygons into one, as with the case of general polygons described above. Figure 4a shows a simple Voronoi diagram produced by this method.
This dynamic maintenance of the Voronoi diagram and dual Delaunay triangulation gives great flexibility in a wide variety of spatial problems. One example not discussed further in this paper involves the Free Lagrange method of modelling fluid dynamics, where discrete "parcels" of fluid are forced into movement by global forces or by moving neighbours (see various examples in Fritts et al., 1985).

Linked lists

One dimensional insertion and deletion

The Voronoi diagram in two dimensions may profitably be compared with the operation of a conventional one dimensional doubly linked list, such as one used for maintaining a sequence of numbers in ascending order.

Fundamental operations on one dimensional doubly linked lists include several basic operations. An initialize process usually involves setting up two end nodes with values selected to be outside the range of the data to be inserted. An insert operation permits 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. Another 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.

For one dimensional doubly linked lists, it is conventional to consider a node containing a data value together with two pointers, one pointing to the next "lower" node and one to the next "higher". In this case it is difficult to see the existence of the "dual" data structure. If, however, we separate the pointer data structure from the data, we can consider the pointer set to be the "dual" of the ordered set.
of data, and this dual would have pointers both to the data objects as well as to adjacent dual cells (see Figure 5). The potential value of this visualization lies in its extension to higher dimensions.

This reduction of the two dimensional concept to the one dimensional case can readily be visualized by considering a cross section through a Voronoi tessellation along the X axis. Various polygons are crossed, each with their generating point. What has happened to our original points, polygons and triangulation of relationships? The points are equivalent to the data values (in order in the X direction) in a one dimensional linked list. The polygons have become the "zones of influence" of each data value, expressed in the "X" direction only, as shown in Figure 5. The relationship triangulation has become the left and right pointer linkage of our doubly linked list. The insertion of a new value into the list requires the creation of a new one dimensional Voronoi region for the new point, obtained by stealing part of the adjacent regions. (This is one way of considering the linear interpolation problem in both one dimension and higher—see Gold 1989.) Insertion of the new point requires searching and region splitting techniques in any number of dimensions—the significant difference in one dimension being the one-for-one correspondence between the data objects (data values) and the dual relationship pointer set, permitting both value and pointers to be stored in the same node.

Extension to higher dimensions

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 always tiled with 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.

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,
Search procedures

Simple walk

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) or Gold and Cormack (1986, 1987). 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. This spatial ordering of a triangulation is based on a reference "viewpoint", and the triangulation may be traversed as a binary tree based on the triangle edges being "towards" or "away from" the viewpoint. The same is also true in the one dimensional case—but only two viewpoints are available, giving either an increasing or decreasing ordering in the number space.

Superimposed tree structures

Instead of a staple walk, it is clearly easy to superimpose any desirable tree structure on our data structure. In one dimension this would be a binary tree; in two dimensions it could be a quadtree. In any case, the construction and operation are obvious: to perform a rapid search of the data structure to locate the closest value to the one desired, whether this is based on a single coordinate, or two.

Polygon trees

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 \(B\) as in Figure 2. 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.

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 that the next data point to
be inserted into the data structure will be close to the previous one. Where a higher order of efficiency is desired the binary tree structure previously mentioned may be implemented.

Line segments

History of movement = sum of relationships

The ability to move points within the Voronoi diagram readily gives rise to another useful property: the generation of a line segment as the locus of a moving point. Thus a line segment is the record of all the adjacency relationships associated with the moving point during its migration. A noticeable difference is that the implicit boundary between a point "nucleus" and a line "nucleus" is parabolic, while between "pairs of points and pairs of lines the boundary is linear. As before, the relationships are preserved as the dual triangulation.

Because a line segment is the history of the movement of a point, a distinction must be made between the interior of the line segment (which has a particular distance-function relationship with neighbouring objects) and the two end points which it connects. Thus the connection of two existing points by a line segment adds additional information to the nap, and this is reflected in the insertion of an additional object (the line segment interior) into the Voronoi diagram and into the dual triangulation. Figure 4b shows the Voronoi diagram for a single line segment among a set of points which could equally well be considered the history of the movement of the end point. For details see Gold (1990a).

Rasters and the Voronoi Adjacency Graph

The dual triangulation expressing the Voronoi adjacency relationships between point and line objects is referred to as the Voronoi Adjacency Graph (VAG, Gold 1990a), as the term "Delaunay triangulation" was specifically coined for point-point relationships, not for those between various types of nap objects (the set of which includes points and line segments, but is not restricted to them). The dual of the VAG is therefore a set of contiguous zones about the (point or line segment) nuclei which are closer to that nucleus than to any other. Unlike raster images there is a one-to-one correspondence between map objects (or nuclei) and their Voronoi polygons. Like rasters, however, adjacency is defined by common boundaries between polygons—although explicitly represented by the VAG rather than implicitly by row and column location. Rasters are thus point Voronoi tessellations with a regular and implicit VAG.
Applications

Area-stealing Interpolation

In interpolation problems, such as contouring or perspective view modelling, it is difficult to generate an interpolated surface that will always honour all data points, whatever their distribution. Figure 6a shows a staple Voronoi tessellation of a small point data set. Figure 6b shows the result of inserting a new data point, marked X. This new point is not a "real" data point, but simply a sampling location where an elevation value is desired. Figure 6b shows the new polygonal region carved out from the Voronoi polygons of the real data points themselves. Figure 6c shows the areas of each of these polygons "stolen" by the Voronoi polygon of this new dummy point. These stolen regions are of interest as they permit straightforward interpolation between arbitrarily distributed data points. The relative areas stolen from adjacent data points are used as weighting functions, generating a weighted average of these adjacent points to form the estimated elevation at the point marked X. A 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, 1988, 1989, 1990b for more details and examples).

Corridors, skeletons and point-in-polygon problems

Figure 7 shows the result of generating a simple polygon from points and line segments using the methods described previously. The basic Voronoi operations performed were: "search" to locate the nearest existing point to the location desired for the new object? "split" to generate a new point object from the desired parent object (together with a connecting line segment it desired); "move" to nova the new point to its final location, preserving all Voronoi relationships on the way; and "merge" to close the polygon loop at the end by deleting the unwanted moving point once it reaches the location of the first vertex of the polygon.

The result of these basic operations is the Voronoi diagram of the points and line segments forming the polygon. By definition, the polygon skeleton is formed by the boundaries of the interior Voronoi regions—somewhat similar to the "bisector skeletons" of Brassel and Jones (1984). In addition the corridor or butter zone around a map object may be derived directly from the Voronoi diagram, as also illustrated in Figure 7; Voronoi zones of line segments have linear corridor edges, and Voronoi zones of points have circular arcs. Graph traversal techniques ensure that the corridor boundaries never cross each other, even for wide corridors.
Figure 7 also illustrates a direct implementation of Nordbeck and Rystedt's (1972) "Enlarged Orientation Theorem for Polygons", to solve the point-in-polygon problem. This states:

1. A point P lies in the interior of a polygon if it lies closer to its nearest edge \((v(i), v(i+1))\) than to its nearest vertex and if the orientation of the triangle \([v(i), v(i+1), P]\) is the same as that of the polygon.
2. A point P lies in the interior of a polygon if it lies closer to its nearest vertex \(v(i)\) than to its nearest edge, and the vertex \(v(i)\) is concave.

Given the Voronoi diagram of the polygon the problem reduces to determining the Voronoi region within which the point falls (i.e. the closest map object). In Figure 7, P is interior, closest to a line segment; Q is interior closest to a vertex; R is exterior closest to a vertex, and S is exterior closest to a line segment. The method generalizes efficiently to determining which polygon of many a point falls within.

Note that the point-in-polygon problem may also be treated as an area-stealing interpolation problem if desired, illustrating that point-in-polygon is really a form of interpolation over a step shaped surface formed by the polygons themselves. The "missing-census-district" problem may also be handled by identical area-stealing interpolation using the Voronoi diagram of the polygon set, and error estimation may be performed for all the above cases (Gold, 1990b).

Building polygons and islands

Figure 8 shows the Voronoi diagram of another polygon--this time with an interior island. Note that, based on our adjacency definition, segments of the interior island polygon are adjacent to segments of the exterior polygon. Thus islands need not be treated as in any way differing from other polygons, lines or points. As in Figures 4 and 6, the relationships are stored as the dual triangulation or VAG, not as the boundaries, but this has been left out of Figures 7 and 8 for visual simplicity.

In addition, in Figure 8 the interior polygon has been left incomplete--its right-hand edge has not quite been extended to the lower right vertex, as would be common during the map digitizing process. Nevertheless, the Voronoi boundary relationships and VAG, because they are complete for any set of unconnected map objects, are also complete here. Thus a simple examination of the adjacency relationships around the moving point would be sufficient to "snap together" the vertices of the almost-closed polygon. Thus real-time monitoring of the polygon digitizing process is readily achieved, as is subsequent editing.
Conclusions

A Voronoi based spatial model has been proposed and implemented which forms a conceptual link between vector and raster systems, and with one dimensional or higher linked lists. Discrete maintenance of dynamic nap Modification is possible, given a definition of polygon adjacency, the Voronoi criterion, and polygon split, merge and move (switch) operations. Point Movement extends directly to line segment Voronoi generation. A wide class of applications has been shown to be directly resolvable using the point-and-line Voronoi diagram and VAG. It is anticipated that further GIS processes will show themselves amenable to this approach in the future.

Acknowledgements

The funding for this research was provided in part by a Natural Sciences and Engineering Research Council of Canada operating grant, and in part under the Energy, Mines and Resources Canada research agreement program.

References


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

Figure 2.  a) General polygon set with triangulation.  
b) Result of splitting P(ab) into P(a) and P(b).

Figure 3.  a) Triangle pair contradicting Voronoi criterion.  
b) Pair switching establishes Voronoi criterion.
Figure 4. a) A Point Voronoi Diagram.
   b) Extension of a line region: point 18 was split from point 17 and moved to its final location.

Figure 5. Voronoi approach to a one dimensional linked list; Data objects, Dual graph and Voronoi regions

Figure 6. a) Point Voronoi diagram and dual triangulation.
   b) Introduction of point X.
   c) Areas stolen from neighbouring regions.
Figure 7. Polygon with Voronoi regions; Corridor and point-in-polygon problems.

Figure 8. Polygon with interior island.