NAVIGATING IN THE MAP

Christopher M. Gold and François Antón Castro
Chaire industrielle en géomatique appliquée à la foresterie
Centre de recherche en géomatique
Université Laval, Pavillon Casault
Sainte-Foy, Québec, Canada G1K 7P4
e-mail: Christopher.Gold@scg.ulaval.ca

Abstract

Maps have traditionally been considered to be passive representations of geographic space, even after the introduction of digital techniques. The ability of the user to interact rapidly with the map on the screen has largely been limited to changes in classification and symbolism, as most spatial data structures do not permit local real-time modification. However, with locally-modifiable "topology" the user's cursor may be embedded in the map space, permitting the detection of neighbouring map objects and responding to potential collisions. This approach has been implemented using the Voronoi (Thiessen) spatial model. It has many advantages in interactive applications such as manual digitizing, fluid flow modelling and robotics. Map objects such as points or line segments may be added, deleted, interacted with or queried at any stage, and the method appears to have many potential applications.

1 Introduction

Maps are an expression of our mode of perception of our surroundings, at a very basic level. While there are many historical examples of this, perhaps the most obvious is the nearly-universal perception of "Planet Earth" that is a result of satellite views of the globe, and the now-frequent use of orthographic and similar projections.

It is not always easy to see the premises on which our perceptions and resulting uses of maps are based. Due to the difficulty of modification, paper maps inevitably imposed a static view of the terrain, implicitly locating the user above the map as a non-participating observer. This may be thought of as the "plane" approach, where the actions of the user are unrelated to the underlying terrain and its cartographic representation.

This approach has carried over to digitally-maintained maps, as the primary application of computer technology has been to generate the same type of static representation. The fundamental storage medium, however, no longer imposes this approach. Some attempts have been made to allow the user's (literal) point-of-view to be included in the definition of the display in various terrain "fly-over" applications. This is a move towards the currently-popular idea of "virtual reality".

However, while flight-simulator types of interaction are now possible, true interaction, where the user interacts with, and modifies, his environment has been left to the realm of computer games, which have been considered to be irrelevant to cartography. A major reason for this is not the inflexibility of the medium, but the inflexibility of the data structures used for major mapping applications. Another reason is that virtual reality has been considered to be an essentially three-dimensional activity.

This paper is an attempt to change these last two impressions, and to suggest that it is beneficial to consider the idea of virtual reality in two dimensions, as, due to the gravitational vector, most of our activities are limited to two degrees of spatial freedom. The "boat" approach to map-user relationships may replace the "plane" approach, with significant advantages. In the "boat" approach the user (as represented by his cursor, or equivalent) is embedded in the same space as the existing map, and interacts with existing objects, such as polygon boundaries, etc.

This important distinction is well illustrated by referring to the manual map digitizing process. Using traditional "plane" techniques, the user digitizes "spaghetti" lines, which may cross each other many times without detection during the data entry operation. A batch program is then run to detect the intersections and build the polygons - leaving many errors to be corrected later by the user. This approach is time-consuming, largely because the user performed the particular operations that caused the error long enough before that the details are forgotten. The first law of interactive programming - that the response to an action is given while the action is still in human short-term memory (a few seconds) - has been broken.

This should be contrasted with the "boat" approach. Here, when a user attempts to draw across a previously existing line, a message is displayed indicating the topological consequences (a node will be generated) and the user may respond. All map structuring takes place in real time, allowing much faster updating and error correction - and no subsequent batch processing is required. The cursor or "boat" bumps into an existing object and, as in real space, there are consequences! This navigation paradigm has many benefits, and there are many possible applications that may be developed in simulation as well as map making.

2 The representation of space

In order to develop such techniques, we need to re-think our understanding of space, especially as represented in the computer. Space as we perceive it may often be treated as two-dimensional, but it is not perceived as a Cartesian coordinate system with X and Y axes (although a vertical Z, because of gravity, is indeed perceived). While it is continuous, it is not primarily metric: relative relationships among objects embedded in that space are more critical. (Imagine a knife and a fork on a table. They are perceived as neighbours only until a plate is placed between them, even if they have not been moved.) These relationships change over time, and this temporal dimension is also important. Finally, two objects may not occupy the same location at the same time, and living creatures detect this in advance by some form of proximity detection.

Many of these properties are lost when space is being simulated in a computer. X and Y coordinates are used, instead of the relative positions of objects in that space. Change over time is not usually feasible, and continuous space must be partitioned into discrete pieces. Finally, adjacency relationships are not automatically available in vector systems, causing the digitizing problems already mentioned, as collisions between the user (cursor, or pen) and pre-existing map objects may not be detected.

The current research started with the observation that spatial tessellations (such as rasters) did not have the same adjacency problems: cells were adjacent if they had a common boundary. However, the cells were not directly associated with spatially embedded objects. An alternative was the Voronoi diagram, a tessellation where each tile consisted of the region closest to a particular object (usually a point). The limitations to this method were that it was not dynamic, in the sense that the adjacency relationships between tiles (or "bubbles") could not be updated as a point was moved by the user. In addition, representation of line segments and more complex objects was not supported. Based on the fact that a line is the locus of a moving point, research focused on the dynamic maintenance of the spatial relationships between the bubble of a moving point and the bubbles of its neighbouring objects.

3 The Voronoi implementation

The dynamic Voronoi approach, as outlined in various papers [9], [10], [11] was designed as a locally modifiable spatial structure, based on a well defined set of relationships with neighbouring objects. It is based on the better known static Voronoi data structure [1], [13].

Voronoi diagrams represent the partitioning of space into cells such that all locations within any one cell are closer to the generating object than to any other, and thus Voronoi edges are curves of equidistance between pairs of objects. Voronoi vertices are equidistant between triples of objects, and are formed at the intersection of three edges. In the general case, polygons meet only at triple junctions in the plane, and hence adjacency relationships may be preserved using the dual Delaunay triangulation. The static point Voronoi diagram is well known (Figure 1), and diagrams where the cells are generated by line segments as well as points have been developed by various researchers [7], [13] (Figure 2). Note that, in order to distinguish between individual connected line segments, a line segment interior is considered to be distinct from its end points.

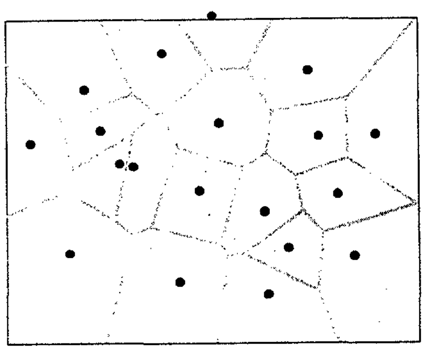

Figure 1: Simple point Voronoi diagram.

Roos [14] described the structure for maintaining the changes in the topological structure as a set of points moved over time. These "topological events" occur when a particular Voronoi edge appears or disappears due to the approach or separation of a particular pair of generating points, and can be considered as the exchange of the diagonal between two adjacent dual triangles. Gold [6] used the same approach for the movement of one point at a time in the generation of line segments.

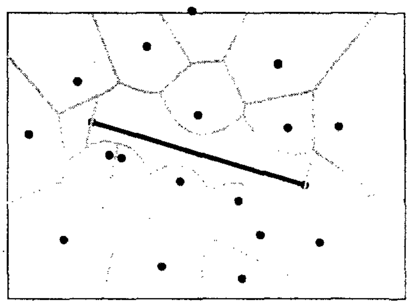

Figure 2: Addition of one line segment.

4 Traditional spatial data structures

The "polygon/arc/node" approach of traditional GIS vector topology is based initially on the detection of line intersections. Thus, the first step in the computer implementation is the detection of all line segments that intersect (or, possibly, almost intersect, given some specified tolerance). After this the graph structure of the map is built by following from one line to its neighbour, defined as another line that intersects it. This sequence is followed until the complete network (e.g. hydrography or thematic polygons) is constructed. This has been described in detail in several places [4].

Sufficient experience has been gained using this technique in the last fifteen years to identify its weaknesses. Basically, if an intersection is not found, then there is no knowledge of neighbours. Detection of intersections is both an expensive operation and prone to errors due to digitizing limitations - and thus some intersections may not be detected. In addition, there is no definition of adjacency other than that of intersecting lines, and features such as islands are difficult to place correctly. As a result, there is no readily-available incremental approach to modifying the spatial linkages, and hence no true dynamic system, permitting the addition or deletion of individual map objects. This implies that topological events, occurring at specified times, can not easily be handled, as these must be added or removed one at a time.

5 The Voronoi dynamic spatial data structure

The Voronoi spatial data structure has developed in response to these perceived limitations that are due to the fundamental dependence on line intersection tests to connect space together. The approach is based on processing the whole space, not just unrelated lines. In this way of looking at things, each elemental map object (currently a point or a line segment) is embedded in a *tile* with real spatial extent - this extent being the portion of the map that is closer to the generating point or line segment than to any other. This produces two immediate benefits: objects can be selected simply by pointing within its proximal tile (the Voronoi polygon); and each tile has a well-defined set of neighbours - those tiles with a piece of border in common. This makes the Voronoi approach somewhat like a raster system, but with irregular tiles - and also like a vector system, with individual nodes and line segments specified.

Two points may be made about the differences between traditional raster and vector methods. Firstly, a spatial tiling is not necessarily incompatible with preserving adjacency relationships between objects - if the tiling represents some zone around each map object (as for the Voronoi) and not some regular, but arbitrary, subdivision of space. Secondly, concerning manual vector methods: in a computer environment one must certainly work with numerical values and hence coordinates, but a restriction of these operations to those achievable with constructional (and hence coordinate and scale free) geometry removes many of the artifacts of scale and coordinate dependence that were previously mentioned. By emulating manual methods one naturally uses an incremental, rather than a batch (divide-and-conquer) approach, as this permits dynamic map modification, e.g. for editing or simulation.

Any technique using a one-for-one spatial tiling, constructional geometry, and incremental techniques would be potentially useful, but the Voronoi diagram is an obvious candidate. It subdivides the map space up into a set of tiles, one for each map object, so as to assign any map location to the tile of the closest object - hence an equivalent term, a proximal mapping. Basic references are [3],[12]. The definitions given above however, do *not* restrict us to map objects that are points. A Voronoi region can, in principle, be constructed around any map object - a house, a river, a road. Implementation of the two dimensional Voronoi diagram (in Euclidean space) for points and line segments allows subsequent construction of more complex objects.

6 Dynamic Voronoi diagram maintenance

The basic two-dimensional Euclidean Voronoi diagram of points and line segments combines the tiling and object adjacencies of raster and vector systems, at the cost, by comparison with raster, of making explicit rather than implicit the tiles' adjacency relationships. The Voronoi zone around each object is the region closer to the generating object than to any other - thus adjacency of objects is equated with Voronoi zones having a common boundary. The dynamic approach (involving the local updating of the spatial data structure, and permitting the insertion, deletion and movement of points and line segments) avoids the line intersection and batch problems described previously. Points may be moved about within the map region, and the adaptive nature of the data structure provides a built-in collision warning system. This point movement process deserves further discussion. It is the *basis* of dynamic maintenance.

If we examine any triangulation, and focus on any particular data point P that we wish to move, there is a set of N immediately adjacent triangles all having P as a vertex, as in Figure 3a. (The value of N averages six for a point data set, excluding boundary conditions.) There are therefore N neighbouring vertices to point P. Each adjacent triangle has one immediate exterior neighbouring triangle, with two vertices in common with the adjacent triangles, and one new vertex.

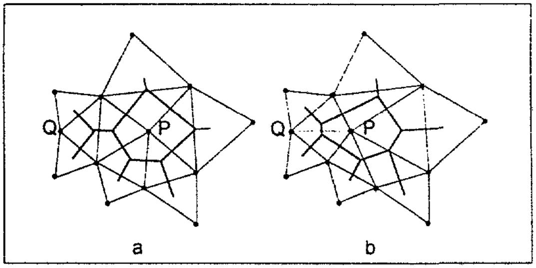

Figure 3: In (a) points P and Q are not neighbours; in (b) they are.

When P is moved a small distance towards Q, it modifies the shape of the boundaries between its Voronoi region and those of its neighbours. These boundaries are represented as the triangle edges connected to P in the dual triangulation. Although they have been modified, all the boundaries still exist, and hence the dual triangulation remains unchanged. For a somewhat larger perturbation of P, however, it's Voronoi bubble will either touch another that it did not touch before or else it will separate from another Voronoi bubble to which it was previously adjacent. In the first case, as seen in Figure 3b, two of the bubbles adjacent to P that *were* adjacent to each other no longer are - but P and a previously exterior bubble Q now touch. In terms of the triangulation, an adjacent/exterior triangle pair has been replaced by two adjacent triangles, simply by switching their common diagonal. In terms of the data points, Q is now a Voronoi neighbour to P.

The second case, where P separates from a bubble to which it was previously adjacent, is a direct reversal of the first case - requiring the switching of two adjacent triangles to form an adjacent/exterior pair. The *condition* under which the switch takes place is, in the first case, when P moves into the circumcircle of the exterior triangle and, in the second case, when P moves out of the circumcircle of the potential new exterior triangle (which would be made up of three data points adjacent to P). This switching process for a moving point has been described in [6],[7],[8]. The circumcircle criterion follows from one possible definition of the Delaunay triangulation - that the circumcircles of all its triangles are empty - i.e. they contain no data points in their interior.

Apart from the ability to move a point, a few other simple operations are required. Points are *created* by splitting an existing point, which is then moved to its final destination. Points are *deleted* in the reverse process. *Lines* are drawn by letting the trail of a moving point accumulate all the spatial adjacency relationships held by the point in its travels, and deleted by having the moving point re-trace its path. Figure 4 shows the Voronoi regions of the line segments forming a polygon set, constructed using the Voronoi moving-point approach. Note that free points, line segments or islands cause no problems.

As an example of a practical application, any free point clearly has a proximal tile or "bubble" with a well-defined set of neighbours. It therefore becomes trivial to detect if a point is interior to a polygon - simply check the neighbouring bubbles for a labelled polygon boundary. In addition, movement or navigation of the point through the existing map may also be achieved by checking the neighbouring bubbles. As the point moves, it acquires new neighbouring bubbles one at a time, and also loses neighbouring bubbles that no longer have a boundary in common.

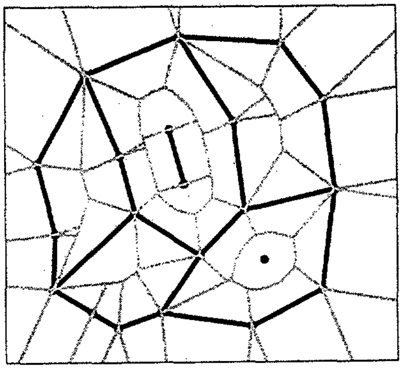

Figure 4: Voronoi diagram of a polygon set, plus a free point and line segment.

7 Topological events: a time-based view of spatial data structures

Each of these acquisitions/losses of a bubble (and its generating point or line segment) as a neighbour to the moving point is known as a "topological event" [14], [15], [2] because the topology of the bubble structure has changed. The Voronoi approach to map construction consists of splitting, merging and moving point map objects, and generating line segments by using the moving point as a "pen". Thus map construction may be thought of as a sequence of "pen commands", building the map a line at a time, while preserving the Voronoi form of topology.

An extension of this, to another level of detail, is to recognize that each command will result in several topological events, as bubbles change their relative locations. Thus each step of the drawn line may be observed. This, however, is less likely to be of interest with drawn map boundaries, as the whole boundary will have the same date-stamp, and is more likely to be interesting for dynamic simulation (e.g. of atmospheric motion) where each topological event will have its own distinctive time of occurrence. The maintenance of a valid topology through these topological events permits a dynamic tracking of the previously described changes over space and time.

8 Applications

The dynamic Voronoi technique allows a variety of applications, based on the "navigation" of a "boat" embedded in the map space, and the consequent dynamic updating of the local spatial relationships. Work to date has emphasized the digitizing process as a primordial form of interaction with the map. Suggestions have been made concerning its application to marine navigation problems [11], and development is under way. Of particular interest is that the same data structure is used for the navigation between obstacles and for deepest-channel finding by local interpolation. Fluid-flow modelling may be performed using the techniques of [15] for the simultaneous movement of multiple points. One approach is that of the free-Lagrange method [5]. Of more direct cartographic concern is a consequence of the incremental nature of the method, which simulates the operations of the cartographer's pen (or eraser). Date stamps may be put on each operation performed (or set of operations), as they are saved in a log file. The map may then be reconstructed to any desired date, by re-generating the map from this file: "the map as a movie". In addition, further updates to the map are saved as commands appended onto the previous log file, saving much trouble in version maintenance. This also permits the generation of an "audit trail" linking map objects between versions.

9 Conclusions

The Voronoi spatial model, and its implementation, appear to be sufficiently general to permit the development of several applications that would not be feasible with traditional structures, with a particular emphasis on preserving the spatial relationships or topology at all stages. In particular, it permits the implementation of a form of real-time interaction, with the user's cursor embedded in the map space and interacting with previously defined map objects, thus permitting topological structuring to be fully interactive. Experiments with this approach show significant improvements both in rapidity and in ease of use for manual map digitizing. Further applications are currently under development.

10 Acknowledgements

This research was made possible by the founding of an Industrial Research Chair in Geomatics at Laval University, jointly funded by the Natural Sciences and Engineering Research Council of Canada and the Association de l'industrie forestière du Québec.

References

[1] Aurenhammer, F., 1991. Voronoi diagrams - a survey of a fundamental geometric data structure. ACM Computing Surveys, v. 23, pp. 345-405.

[2] Bajaj, C.L. and W.J. Bouma, 1990. Dynamic Voronoi diagrams and Delaunay triangulations. Proceedings, Second Canadian Conference in Computational Geometry, Ottawa, August 1990, pp. 273-277.

[3] Bowyer, A., 1981. Computing Dirichlet tessellations. Computer Journal, v. 24, pp. 162-166.

[4] Chrisman, N.A., J.A. Dougenik and D. White (1992). Lessons for the design of polygon overlay processing from the ODYSSEY WHIRLPOOL algorithm. Proceedings, Fifth International Symposium on Spatial Data Handling, Charleston, August 1992, pp. 401-410.

[5] Fritts, M.J., W.P. Crowley and H.E. Trease, 1985. The Free-Lagrange Method. Lecture Notes in Physics 238. Springer-Verlag, New York.

[6] Gold, C.M., 1990. Spatial data structures - the extension from one to two dimensions. NATO ASI Series F No. 65, Mapping and Spatial Modelling for Navigation (L.F. Pau, ed.). Springer-Verlag, Berlin, pp. 11-39.

[7] Gold, C.M., 1991. Problems with handling spatial data - the Voronoi approach. CISM Journal, v. 45, pp. 65-80.

[8] Gold, C.M., 1992. An object-based dynamic spatial model, and its application in the development of a user-friendly digitizing system. Proceedings, Fifth International Symposium on Spatial Data Handling, Charleston, August 1992, pp. 495-504.

[9] Gold, C.M., 1993. An outline of an event-driven spatial data structure for managing time-varying maps. Proceedings, Canadian Conference on GIS, Ottawa, pp. 880-888.

[10] Gold, C.M., 1994. The interactive map. In: Advanced Geographic Data Modelling: Spatial Data Modelling and Query Languages for 2D and 3D Applications (M. Molenaar and S. de Hoop, eds.). Netherlands Geodetic Commission Publications in Geodesy, New Series, No. 40, pp. 121-128.

[11] Gold, C.M., and A.R. Condal, 1994. A spatial data structure integrating GIS and simulation in a marine environment. Marine Geodesy (in press).

[12] Green, P.J., and R. Sibson, 1978. Computing Dirichlet tessellations in the plane. Computer Journal v. 21, pp. 168-173.

[13] Okabe, A., B. Boots and K. Sugihara, 1992. Spatial Tessellations: Concepts and Applications of Voronoi Diagrams. John Wiley and Sons, Chichester.

[14] Roos, T., 1990. Voronoi diagrams over dynamic scenes. Proceedings, Second Canadian Conference on Computational Geometry, Ottawa, August 1990, pp. 209-213.

[15] Roos, T., 1993. Voronoi diagrams over dynamic scenes. Discrete Applied Mathematics, v. 43, pp. 243-259.