Persistent spatial relations - a systems design objective.

Christopher M. Gold,
Chaire Industrielle en Géomatique,
Centre de Recherche en Géomatique,
Pavillon Casault, Université Laval,
Québec, Qc Canada G1K 7P4.
Telephone: (418) 656-3308,
Internet: CGOLD@VM1.ULAVAL.CA

Abstract

This paper approaches the selection and development of appropriate computational geometry algorithms from a system design viewpoint: for one particular - but important - application. This is the domain of GIS (Geographic Information Systems) or Geomatics.

We start with an outline of the subject, followed by a brief description of current commercial practice, then set specifications for a "next generation" system that attempts to overcome current limitations. Appropriate methods are then selected on the basis of this analysis, and the status of the current implementation of such a system is described. We conclude with a discussion of remaining questions - whether the specifications are appropriate, the efficiency and practicality of the algorithms, and large-scale implementation issues.

1. Introduction.

The first Geographic Information System was developed in Canada in the late 1960s as the Canada Geographic Information System [23]. The first operational vector GIS as we use it today was developed at the Harvard Laboratory for Computer Graphics and Spatial Analysis in the late 1970s [24], using the work of [3, 17]. This was based on the development of a functional polygon overlay system. These systems were designed for the processing of polygonal thematic maps, in order to respond to a basic set of queries. It was found that queries of the type "Show me all the areas with Type 1 or Type 2 soil, not zoned as industrial, within 50m of a lake" could be addressed by three basic operations. The first operation, "Reclassify and Merge" or "Dissolve" reclassifies a polygon map on the basis of each polygon's attributes, and combines adjacent polygons of the same resulting class. The second operation, "Corridor" or "Buffer Zone" constructs a polygon set with boundaries at a specified distance around the target object set (lakes in our example). The third operation, "Polygon Overlay" combines two polygon coverages into a single coverage such that each polygon has properties from each original coverage (e.g. Type 1 and industrial in our example). All three of these are global operations on polygon sets, and depend for their operation on a complete rebuild of the "topology" (i.e. the polygon/arc/node relations) after each operation.

Due to the complexity of the topology construction from the original coordinate input, which is full of special cases, GIS software rapidly became a commercial product with the development of the first more-or-less reliable polygon overlay/building routines. Due to the effort required to create the basic tools; due to the rapidly-expanding market (now in the billions of dollars) involving government and industry at many levels; and due to the increasingly large data sets, the development of software has been considered to be beyond the ability of academic researchers. This has put GIS as a discipline in a difficult position, either stuck with conceptual issues or regressing to technical training in the use of commercial products.

2. The "Next Generation"

However, in the last few years, dissatisfaction has started to grow concerning the limitations of the global, batch, polygon based spatial model being used. Even minor modifications to the map require a global rebuild, although a variety of techniques are used to modify a small patch and then to sew it back into the main map. The digitizing process becomes extremely error-prone, as arcs are added incrementally, but the topology required to define the connectivity is only added as a batch operation afterwards. As connectivity is defined by the intersections of arcs, this is only functional for polygonal map tilings (and, to a much lesser extent, networks). Thus no spatial structure can readily be defined for disconnected objects - point sets, islands, etc.

As the emphasis on the user interface continues to grow, it is clear that the current non-interactive spatial model will be superseded by something closer to the current human interaction with a paper map using a pen - or even our interaction with real-world geographic space. This requires that our actions take effect immediately, both in spatial queries and spatial modification of our map. If we can detect the neighbouring objects to our moving pen at any time during map construction - so must the GIS. If we can detect polygons or other structures that fail to close exactly - so must the GIS. If we can locate the polygon containing our pen, an isolated data point, or an island without special processing - so must our GIS.

All of the above problems arise because our spatial relationships are based entirely on the "vector" model of intersecting line segments being our only definition of adjacency. Yet even with the polygon-arc-node model there is a more stable definition: polygons are adjacent if they have a common boundary. In the raster spatial model (also frequently used in spatial analysis) this is trivially true. We thus need a spatial model possessing the tiling and adjacency properties of the raster but the direct relationship to real-world objects of the vector model.

3. Selection of appropriate methods.

While this has puzzled the GIS community for some years, at least one potential answer is readily available from computational geometry. The Voronoi diagram was discussed in some of the earliest work (e.g. [21] for point sets), and many algorithmic improvements and extensions have been made since then. Aurenhammer [1] quoted that it was "one of the most fundamental constructs defined by a discrete set of points". In GIS in particular, the diagram is of interest primarily (but not exclusively) in two dimensional Euclidean space. The order-1 Voronoi diagram is most often used, although [16] shows many applications of order-k diagrams for various types of spatial queries. Of particular value for our two-dimensional mapping applications are diagrams for points and line segments (e.g. [14, 5]), although more complex objects are also of considerable interest [15].

The Voronoi diagram for points and line segments as described above resolves the basic difficulties with the line-intersection spatial model, as well as providing basic spatial adjacency properties between map objects - when two Voronoi cells touch, the generating objects are neighbours, and these relationships may be stored as the Delaunay triangulation. However, these algorithms are usually batch (as in the sweep-line method), and are not able to respond to queries about neighbouring objects during the "topology" building process - a necessity for reasonable user efficiency during digitizing. Some fully dynamic algorithm is required.

4. Dynamic algorithms.

A semi-dynamic algorithm for the construction of the construction of the Voronoi diagram of a set of points and line segments has been developed in [4, 12]. Schwarzkopf [20] maintains a history of the additions and deletions to the Voronoi diagram for points and line segments, which must be rebuilt from time to time. Dynamic Voronoi diagrams for point movement have recently been developed in [2, 18, 19, 13]. These have been dynamic in the sense that the Voronoi diagram or the Delaunay triangulation are preserved during point movement - often for the case of all points moving simultaneously. Gold [6, 8] addressed the case of one point moving at a time, but in addition being able to create and delete points by splitting them from previously existing points, or else merging them. Line segments are handled in a similar fashion, as the locus of the moving point. This gives a fully dynamic algorithm for the creation and deletion of points and line segments, and for point movement. Figure 1 shows a point Voronoi diagram, before and after a single line segment was extended from point 17. Figure 2 shows the effect on the dual triangulation of splitting one polygon on the map into two (the process used for generating new points in the Voronoi diagram, and the reverse for deleting them). Figure 3 shows the "topological event", or triangle switch, as point P moves towards point Q. Figure 4 shows that buffer zone generation is a simple operation on individual Voronoi cells, and that the point-in-polygon problem may be easily resolved by finding the nearest object (point or half-line) to the query point as in [7].

5. Status of system implementation.

Because it is not yet considered feasible to persuade large organizational users to risk their data on an untried system, current work is targeted to spatial decision support systems (SDSS), where the user intends to download the available data and "play" with it, both to familiarize himself with the its behaviour, and also to select strategies that may not be optimal, but will not be catastrophic. This has been called the "flight simulator" approach [9] - or alternatively a "spatial spreadsheet". This suggests a PC environment, with a Microsoft Windows operating and development system. The basic libraries are now in place and a variety of query types are available. Figure 5 shows two typical Voronoi diagrams for simple maps.

The dynamic structure as defined above possesses various additional advantages that we are just beginning to exploit. Because the topological structure is a Delaunay triangulation, both global and local searching techniques are directly achievable - see [10, 12, 22]. Because there are no direct pointers from objects to the triangulation, only one copy of an object need be preserved even if the object is referenced in several map coverages (a significant data base issue). Because the update process is incremental, addition or deletion of objects is immediate, and queries may be asked at once - of great value in identifying objects to be joined, for example, as it is only necessary to point within the Voronoi cell of an object in order to select it. Also, an incremental map construction opens various ways of managing the history of a map (usually snapshots are preserved at selected times; here a "tape recording" of all map construction commands permits the reconstruction of a map to any desired date - or its playback as a movie). Finally, spatial operations not normally associated with the polygonal topology may be implemented using a single spatial data structure. These include robot navigation, interpolation, fluid flow simulation, dynamic network analysis, etc. See [11] for some examples.

6. Future work.

While the approach appears to be of great potential value, no formal analysis has been made of its appropriateness. Assuming that the future needs are reasonable, what other spatial structures are possible candidates? How do we discuss efficiency of operations in an interactive environment? How do we handle geometric precision issues for circumcircle and circle/line intersection tests? What is an appropriate hierarchical search mechanism, given that a local adjacency search is available, and that the triangulation is being continually modified, even by some types of queries? While current techniques are effective for in-memory applications, how do we partition a large triangulation for disc storage, when perhaps it spans the whole globe? If the basic operations may be considered to be those of a general-purpose "Voronoi engine", then several engines may be operational at the same time - either in separate map layers, or in several portions of the same layer. (Alternatively, several users may be updating the same map at the same time from different workstations.) How should such an approach be parallelized, to handle potential conflicts and deadlocks? Full-scale system development requires adequate responses to many of these questions.

7. Conclusions.

We have identified a set of weaknesses in current GIS practice that can be attributed to a poor model of space, and we have suggested an alternative. This has been implemented to a level that validates the proposed system design. However, even a successful analysis of future GIS needs leads to significant implementation issues, and the field of computational geometry could contribute greatly to a systematic resolution of an important field of spatial analysis. Nevertheless, the concept of a seamless, persistent, locally-modifiable spatial data structure that is always complete except for very brief modifications provides an important new approach to spatial data handling.

Acknowledgements

The funding for this research was made possible by the foundation 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. Voronoi diagrams - a survey of a fundamental geometric data structure. ACM Computing Surveys, v. 23, (1991), pp. 345-405.

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

[3] Bureau of the Census. The DIME Geocoding System. Report No. 4, Census Use Study. Washington: U.S. Department of Commerce, Bureau of the Census (1970).

[4] Chazelle, B., H. Edelsbrunner, L.J. Guibas, M. Sharir, and J. Snoeyink. Computing a face in an arrangement of line segments. Proceedings, ACM Symposium on Discrete Algorithms, (1991), pp. 441-448.

[5] Fortune, S. A sweepline algorithm for Voronoi diagrams. Algorithmica, v. 2, (1987), pp. 153-174.

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

[7] Gold, C.M.. Space revisited - back to the basics. Proceedings, Fourth International Symposium on Spatial Data Handling, Zurich, (1990), pp. 175-189.

[8] Gold, C.M.. 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, (1992), pp. 495-504.

[9] Gold, C.M. Forestry spatial decision support system classification and the "flight simulator" approach. Proceedings, GIS'93, Vancouver, (1993), pp. 797-802.

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

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

[12] Guibas, L.J., D.E. Knuth, and M. Sharir. Randomized incremental construction of Delaunay and Voronoi diagrams. Algorithmica, v. 7, (1992), pp. 381-413.

[13] Guibas, L.J., J.S.B. Mitchell, and T. Roos. Voronoi diagrams of moving points in the plane. In: Lecture Notes in Computer Science, No. 570, (1993).

[14] Kirkpatrick, D.G. Efficient computation of continuous skeletons. Proceedings, Twentieth Annual IEEE Symposium on Foundations of Computer Science, (1979), pp. 18-27.

[15] McAllister, M., and J. Snoeyink. The spoke diagram: a compact approximate Voronoi diagram for spatial computation. Proceedings, GIS'94, Vancouver, (1994).

[16] Okabe, A., B. Boots and K. Sugihara. Spatial Tessellations - Concepts and Applications of Voronoi Diagrams. Chichester: John Wiley and Sons, (1992), 521 p.

[17] Peucker, T. and N. Chrisman. Cartographic data structures. The American Cartographer, v. 2, (1975), pp. 55-69.

[18] Roos, T. Voronoi diagrams over dynamic scenes. Second Canadian Conference on Computational geometry, Ottawa, (1990), pp. 209-213.

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

[20] Schwarzkopf, O. Dynamic maintenance of geometric structures made easy. Proceedings, IEEE Symposium on Foundations of Computer Science, (1991), pp. 197-206.

[21] Shamos, M.I. and D. Hoey. Closest-point problems. IEEE Symposium on Foundations of Computer Science, (1975), pp. 151-162.

[22] Teillaud, M.. Towards Dynamic Randomized Algorithms in Computational Geometry. Lecture Notes in Computer Science, No. 758, (1993), Springer-Verlag, Berlin.

[23] Tomlinson, R.F., H.W. Calkins and D.F. Marble. Computer Handling of Geographical Data (1976). Paris: The UNESCO Press.

[24] White, D. A design for polygon overlay. In: Harvard Papers on Geographic Information Systems, v. 6, (1978).

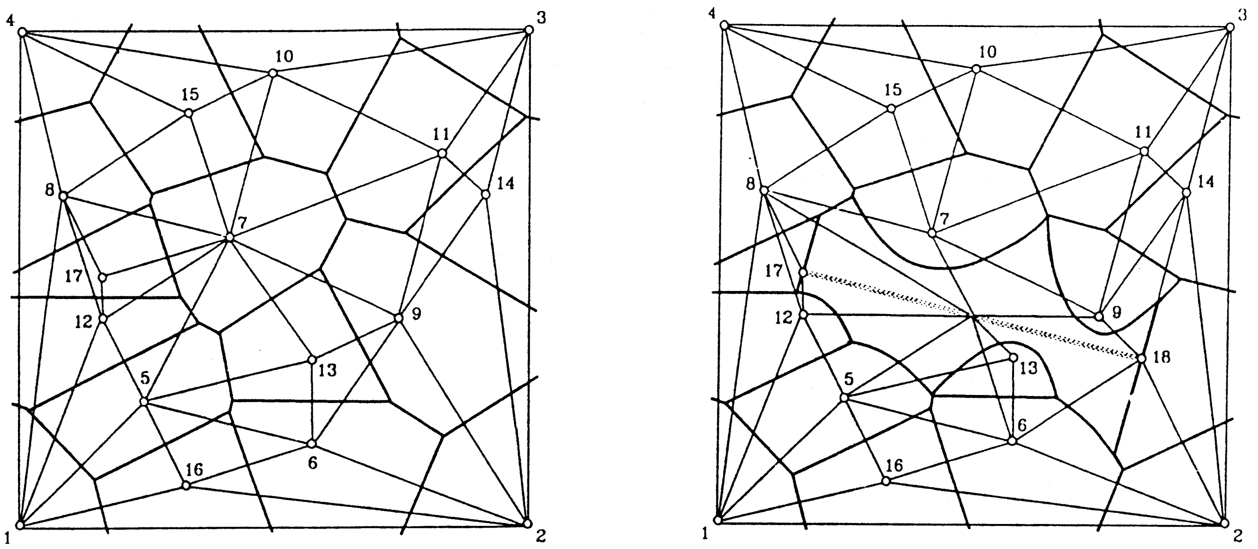

Figure 1. A simple point Voronoi diagram, and the addition of a single line segment.

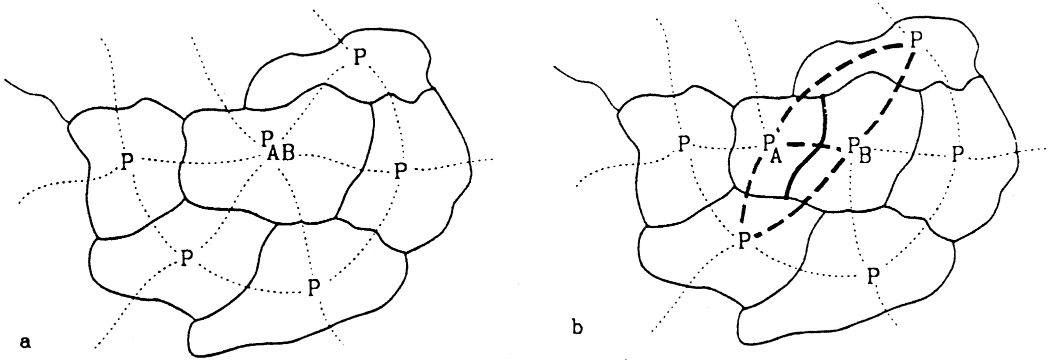

Figure 2. Modifying the dual triangulation to accommodate a polygon split.

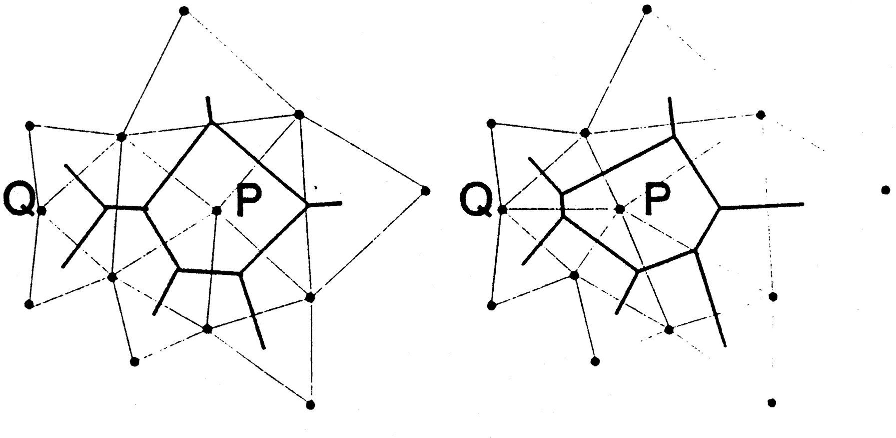

Figure 3. A topological event caused by point P approaching point Q.

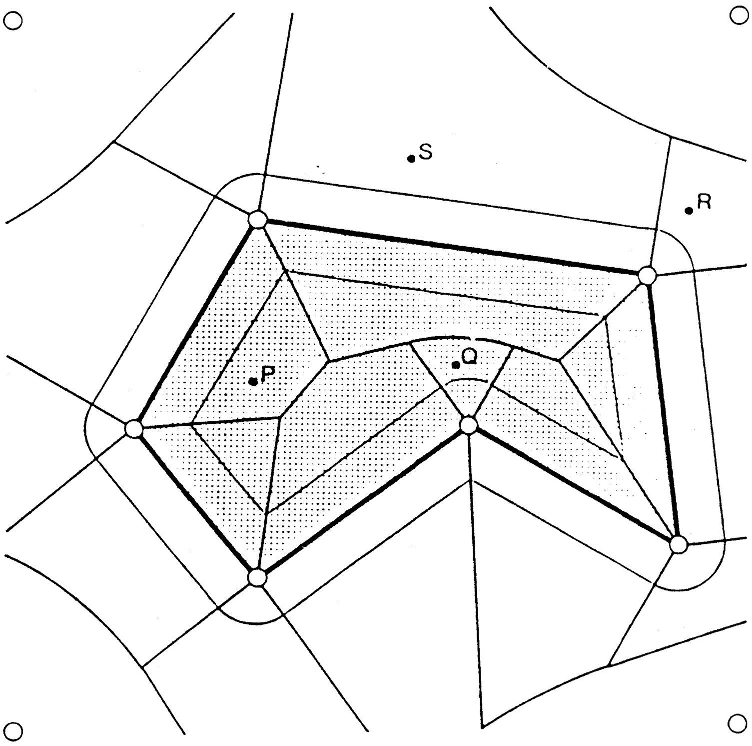

Figure 4. Buffer zone generation and the point-in-polygon problem using the Voronoi diagram.

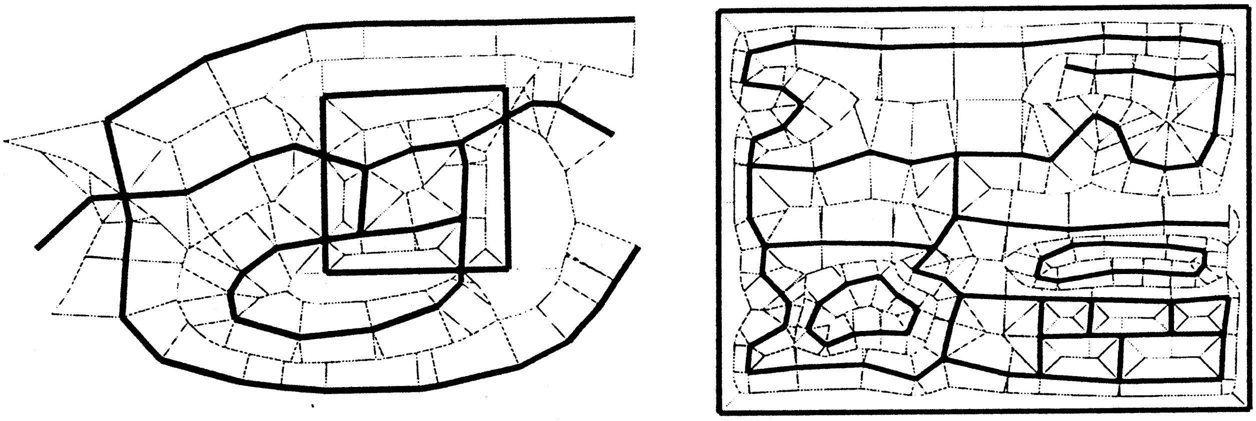

Figure 5. The Voronoi diagram of two simple maps.