The use of the Dynamic Voronoi Data Structure in Autonomous Marine Navigation.

Christopher M. Gold
Centre de Recherche en Geomatique
Universite Lava1
Ste-Foy, Quebec, Canada G 1 K 7P4
Tel: (418) 656-3308; E-mail: Christopher.Gold@scg.ulaval.ca

Abstract- **Marine navigation involves problems not found on land. As well as fixed obstacles, there are objects whose positions change frequently. Apart from surface obstacle avoidance, a boat must contend with bathymetric depth, which is affected by sea floor and tidal changes. Maintenance of a dynamic data structure for automatic pilotage is a complex task.**

The dynamic Voronoi diagram of points and line segments provides a natural structure for dealing with adjacency-based collision-avoidance problems. Embedding the moving boat within the diagram permits the detection of immediate neighbours within any specified tolerance. Depths may be estimated from neighbouring depth soundings using "area-stealing" techniques, and accommodation made for tidal values at that time. The dynamic structure allows real-time local updating of obstacle and marker positions. Historical log files permit the replaying of past system states. While global navigation goals must be developed by other means, the dynamic Voronoi structure allows for safe local navigation.

1. INTRODUCTION: MARINE DATA STRUCTURES.

Traditional GIS data structures have always been something of a problem in a marine context. There is a great deal of point data, and relatively little polygonal data by comparison with terrestrial applications. It would thus be very desirable to be able to work within spatial data structures that could both handle objects that were not necessarily connected, and provide the necessary basis of spatial adjacency or proximity. People have no difficulty navigating around isolated features, so why should a computer?

Additionally, in a marine environment even more than in a land-based one, objects are likely to change location, and it would be nice if this could be handled in a dynamic system. Finally, while in both terrestrial and marine systems we may be representing both discrete objects and "fields" that vary continuously over the map, in a marine context these fields are less often classified into a discrete polygon tiling.

Our recent work has been concerned with the development of spatial data structures with some of the desired properties mentioned above, as a potential replacement for the usual terrestrial techniques. In addition, we are concerned with time-varying situations - because the objects have changed, or because the operator wishes to interact with the system in a timely (and maybe urgent!) fashion.

The techniques developed at Lava1 University are based on the idea of Voronoi tessellations, used as a dynamic spatial data structure. These are described in various articles (e.g. [I] to [5]), and the purpose of this paper is to evaluate them for use in marine navigation. We consider that a reasonable "wish list" for a marine GIS would include the ability to handle non-connected and connected objects, as well as field-type data, that would be able to vary their relative positions and values over time. At any appropriate moment the spatial structure should respond to queries concerning values or spatial relationships, e.g. for navigation purposes. While not feasible with static polygon structures, implementation is possible using dynamic Voronoi tessellations.

2. PRINCIPLES OF VORONOI METHODS

Static point Voronoi tessellations are well known in the literature, and algorithms have been used for many years (see [6] for a summary). Less well known are dynamic algorithms, that allow point creation, deletion and movement, and also Voronoi tessellations of more complex objects - typically line segments as well as points.

Algorithms for generating the simple point Voronoi tessellation have improved significantly in theoretical efficiency in recent years. Where the whole structure may be constructed at once, randomized incremental algorithms such as [7], [8] and [9] can create these diagrams in expected time O(n log n), which is optimal. However, as a major motivation for this work concerned the maintaining of a map when one or more objects are moving, an alternative technique was developed that maintained the Voronoi spatial relationships while map objects were being inserted, deleted, or displaced. This is achieved by determining when the Voronoi cell of a moving point gains or loses a neighbouring cell, moving the point to that location, and locally updating the topological structure accordingly. For the case of all points moving simultaneously, [10] give a rather complex theoretical efficiency based on Davenport-Schinzel sequences, but in the case of one point being inserted at a time by splitting it from the nearest pre-existing point and then moving it to its destination (see below) the expected time efficiency should again approximate O(n log n).

3. DYNAMIC SPATIAL DATA STRUCTURES

The development of a dynamic algorithm from the static point Voronoi tessellation, permitting the points to be moved around without destroying the "bubble" structure of the cells, allowed greater freedom in the development of spatial data structures for time-varying maps. While still generating the Voronoi tessellation, the algorithms are significantly different, with the emphasis on detecting when the Voronoi cell of a moving point will change its set of neighbours. This has been described for point sets by [11], [12] and [13], and for points and line segments by [I]. In two dimensions this involves switching the diagonals of adjacent Delaunay triangle pairs as a point moves, in order to preserve the empty circumcircle criterion for each triangle. Given methods to split an old generating point and its cell into two, generating a new one capable of being moved to its desired destination, and the inverse operation of merging two points and cells, a dynamic data structure was obtained that could be used as a replacement for the static point tessellation. Apart from the ability both to add and to delete points, however, it permitted points to be moved about the map in sequence, while preserving their spatial relationships - i.e. their Voronoi neighbours - at all times. This has obvious applications for navigation, where one or more of the points represent ships, etc. A further development includes the area-stealing model within this navigation process, allowing interpolation of the sea-floor, etc., from the neighbouring soundings, and thus permitting direct navigation from the data themselves rather than from a pre-compiled contour map or equivalent.

From the moving-point approach two more general methods may be developed. The first method is based on the fact that if a point may be moved while preserving its spatial relationships (as expressed by the adjacency of the Voronoi cells), then it may be used to draw a line. This line would be formed between where the point was split from a previous point, and the final destination. The first description of this moving-point approach was given in a NATO workshop on navigation GIS [1].

The second method involves the observation that the topological construction and point movement just described is a sequential process, to which a specific time or date may be assigned. Thus the sequence of commands used to build the map (or to simulate point movement) may be preserved in a log file that represents the history of the generation of the map, or the movement of the ship. Any editing of the map due to later changes in the real world may be performed by adding new commands to the log file. This has several advantages - it is not necessary to preserve snapshots of the map at different historical dates, with all the resulting difficulties in comparing them later. In addition, the map may be played back at any time, like a movie, stopping at any desired time, and requiring only one file containing all steps of map creation and modification.

4. APPLICATIONS OF THE DYNAMIC VORONOI METHOD.

We have briefly reviewed the basic concepts of dynamic Voronoi data structures, as described in the literature. To support the approach for use in a marine GIS, it is perhaps best to illustrate some of the advantages by using several applications as examples. The key point for us is that while each of the applications may be developed separately, to our knowledge no other underlying methodology is capable of being applied to all of them - that is, using the same basic data structure and set of spatial operations. This strongly suggests an underlying generality that may be used for rapid development of further applications as needed.

4. I. Interpolation of Precise, Arbitrarily-distributed Point Data.

The “area-stealing” [14] or “natural neighbour” [15] [16] method of interpolation is based on one idea: that if the “query point” (at the location where it is desired to make an interpolation estimate) is inserted into the Voronoi tessellation formed by the real data points, then it will reduce (“steal”) some area from the adjacent Voronoi cells. These areas are ideally suited as weights to be used for calculating the weighted average of the heights of the neighbouring data points. The areas have zero value when a data point ceases to be a neighbour to the query point (i.e. their Voronoi cells no longer touch), and when the query point coincides with a data point all of the stolen area will be taken from that particular data point’s Voronoi cell. This produces one of the few spatial models that can guarantee that an interpolated surface both passes through all data points and has no discontinuities or other artifacts. It is not dependent on the data distribution - indeed, it has been used successfully with extremely anisotropic data. [15] showed that the interpolated surface resulting from this is continuous within the convex hull of the data, and has continuous slope, except at the data points themselves.

4.2. Topology of Complex Objects (Points Plus Line-segments).

As described in [1][5], it is possible to maintain the Voronoi topological structure while a point is moving across the map. A point is created by splitting it from an existing point and moving it to the desired location, and a point is deleted by reversing this operation. Line segments, which are the loci of moving points, may be generated in the same way, by growing the segment behind the moving point. Line segments may be deleted in the reverse process. The result is a dynamic Voronoi topological structure for arbitrary points and line segments that can be extended to completed polygon sets with islands, etc. without any special treatment.

4.3. Buffer Zones And Minkowski Sums

It is a standard operation in a terrestrial GIS to generate buffer zones around lakes, rivers, etc. to represent environmentally protected areas, for example. The same function is used in the marine context for national marine limits, and for restricted access areas. The standard GIS treats this as a batch “polygon overlay” calculation. What is less understood is that a buffer zone is a “distance to the closest portion of the target set” type of operation, and as such may be directly resolved with the Voronoi tessellation, which by definition is the partition of the map space into proximal regions associated with each map object (e.g. the individual points or line segments used to construct more complex features). Thus the generation of a buffer zone consists merely of the evaluation of each Voronoi cell, and the drawing of a circular arc/straight line within the cell associated with a point/line segment. (See Fig. 1, which also illustrates that the point-in-polygon problem may be resolved by finding the nearest point or half-line.)This is a vector equivalent of the Minkowski sum of the map object and an appropriately-sized circle. For navigation problems involving robots whose size and shape may be represented by a circle, it is obvious that the Voronoi diagram suffices as a simple mechanism for path planning along the medial axis transform of the navigable space. The flexibility of the dynamic structure allows the scene to change over time - a common requirement in the marine environment.

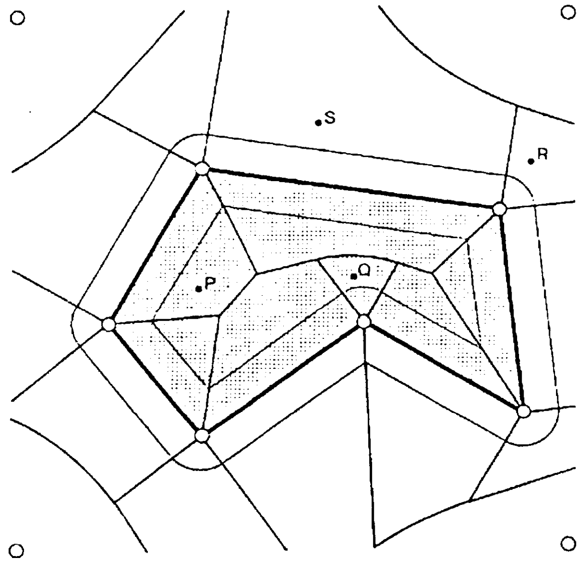

Fig. 1 Buffer zone generation.

4.4. Dynamic Modification of Topologically Structured Maps.

Because the Voronoi topology may be maintained during the incremental map building process (and indeed the persistence of a complete topology is necessary to the navigation process required for map building) a map may be built in an incremental fashion. It is not necessary, however, that the points and line segments form a complete space-filling polygon set, as with traditional systems - the Voronoi cells themselves form the complete tiling necessary for a continuous model of space. Thus the map may be edited at some later date without any necessity to re-build the whole structure. Points and line segments may be added and deleted at will, and methods exist to detect collisions between the moving "pen" and the existing map - in which case an intersection point may be generated. All of these are local operations, which do not require the modification of the topological structure outside the immediate area of interest -just the addition or removal of a Voronoi cell, and the consequent modification of its immediate neighbours.

4.5. Robot Navigation Between Objects.

The map-building process involves the control of a moving point within the Voronoi tessellation representing the current map objects and their spatial relationships. This moving point corresponds to the cursor or pen. When the pen attempts to cross a pre-existing line, the Voronoi structure is capable of detecting this in advance [5] and generate an intersection. The same mechanism may be employed for simulated navigation among obstacles in the map as, for example, a boat may be directed through a marked channel. Because Voronoi cell adjacencies define local neighbours, the small number of neighbouring objects to the moving boat may be tested for possible collision - for example boat "A" in Fig. 2. This is also closely related to two other techniques: buffer zone generation, as described previously; and the use of the Voronoi tessellation as the medial axis transform of a polygon or some other set of objects. The medial axis transform, or "skeleton", defines the path furthest from the adjacent object [17]. Work by [18] and others, have used this in various object recognition problems, where they refer to the "endoskeleton" and "exoskeleton". Thus the same data structure used for map or chart generation may be used to select the best route or to detect potential collisions on the course specified (using the local neighbours).

4.6. Robot Navigation Within a Bathymetric Channel.

The collision-detection method just described, using the moving-point model, may be modified based on the area-stealing interpolation described earlier. Since the moving point has a Voronoi cell, it has neighbours, and it has "stolen" area from each of them. It thus can be used, as before, to estimate the elevation (or depth) at that location - for example boat "B" in Fig. 2. The slope may also be determined, and thus the simulated boat may be navigated to the deepest channel or, if preferable, controlled by the landmarks (map objects) of the previous section but with some warning tolerance of available depth. (These could be combined with tidal variations if so desired.) Using this structure, contoured bathymetric models might not be necessary, as individual depths would be calculated as required.

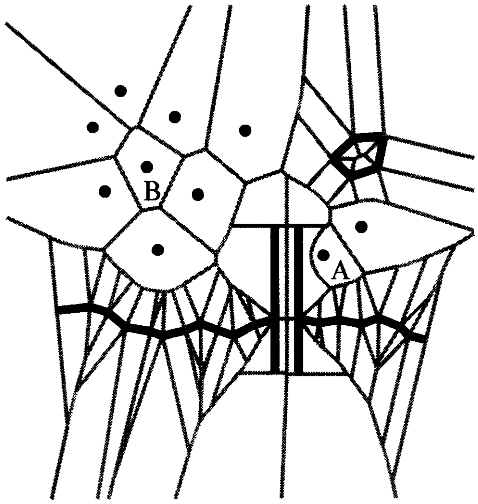

Fig. 2 Schematic navigation System.

5. CONCLUSIONS

These examples suggest that it is by no means infeasible to integrate many or all of the above functions within one system with one common data structure, allowing complex time and space simulation/navigation/queries. The system is implemented as a “Voronoi engine”, with a small set of commands and queries available to the specific application. Given this common spatial engine, individual applications may be developed that build on its basic properties. Alternatively, the engine may be used as the basis for a fully integrated marine GIS, where a common, consistent set of operations are used for a wide variety of spatial processes.

In more general terms, it is clear that “GIS” as we have known it is of limited application, and what is needed is a new set of tools and structures. Marine navigation is one example of where a more flexible and general-purpose spatial data structure is required.

ACKNOWLEDGEMENTS

The funding for this research was made possible by the foundation of an Industrial Research Chair in Geomatics at Lava1 University, jointly funded by the Natural Sciences and Engineering Research Council of Canada and the Association de I’lndustrie Forestière du Quebec.

REFERENCES

[I] C.M. Gold, “Spatial data structures - the extension from one to two dimensions,” in L.F. Pau (ed.), *Mapping and Spatial Modellingfor Navigation,* NATO ASI Series F No. 65, Berlin’ Springer-Verlag, 1990, pp. 1 l-39.

[2] C.M. Gold, “Space revisited -- back to the basics,” in *Proceedings, Fourth International Symposium on Spatial Data Handling,* Zurich, 1990, pp. 175- 189.

[3] C.M. Gold, “Problems with handling spatial data - the Voronoi approach,” *CISM Journal,* vol. 45, 1990, pp. 65-80.

[4] C.M. Gold, “Dynamic spatial data structures - the Voronoi approach” in *Proceedings, The Canadian Conference on GIS,* Ottawa, 1992, pp.245-255.

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

[6] F. Aurenhammer, “Voronoi diagrams - a survey of a fundamental geometric data structure,” *ACM Computing Surveys, vol. 23,* 1991, pp.345-405.

[7] J.D. Boissonat and M. Teillaud, “A hierarchical representation of objects: The Delaunay Tree” in *Proceedings, Second ACMSymposium on Computational Geometry,* ACM, 1986, pp. 260-268.

[8] L. Guibas, D.E. Knuth and M. Sharir, “Randomized incremental construction of Delaunay and Voronoi diagrams,” *Algorithmica,* vol. 7, 1992, pp. 38 l-4 13.

[9] M. Teillaud, *Towards Dynamic Randomized Algorithms in Computational Geometry,* LectureNotes in Computer Science, vol. 758, Berlin: Springer-Verlag, 1993: p. 156.

[10] L. Guibas, J.S.B. Mitchell and T. Roos, “Voronoi diagrams of moving points in the plane,” in *Proceedings, 1 7th. International Workshop on Graph Theoretic Concepts in Computer Science,* Fischbachau, Germany. Lecture Notes in Computer Science, vol. 570, Berlin: Springer-Verlag, 199 1, pp. 113-125.

[1 1] C.L. Bajaj, and W.J. Bouma, “Dynamic Voronoi diagrams and Delaunay triangulations,” in *Proceedings, Second Canadian Conference in Computational Geometry,* Ottawa, 1990, pp. 273-277.

[12] T. Roos, “Voronoi diagrams over dynamic scenes,” in *Proceedings, Second Canadian Conference in Computational Geometry,* Ottawa, 1990, pp. 209-2 13.

[13] T. Roos, “Voronoi diagrams over dynamic scenes,” *Discrete Applied Mathematics,* vol. 43, 1993, pp. 243-259.

[14] C.M. Gold, “Chapter 3 - Surface interpolation, spatial adjacency and G.I.S,” in J. Raper (ed.), *Three Dimensional Applications in Geographic Information Systems,* London: Taylor and Francis, 1989, pp. 2 l-35.

[15] R. Sibson, “A brief description of natural neighbour interpolation,” in V. Bamett (ed.), *Interpreting Multivariate Data,* New York: John Wiley, 198 1, pp. 2 l-36.

[16] D.F. Watson, and G.M. Philip, “Neighbourhood-based interpolation,” *Geobyte, vol. 2, 1987,* pp. 12-16.

[17] H . Blum,“A transformation for extracting new descriptors of shape,” in W. Whaten-Dunn (ed.), *Models for the Perception of Speech and Visual Form,* Cambridge, Mass.: MIT Press, 1967, pp. 153-I 71.

[18] R. Ogniewicz, and M. Ilg, “Skeletons with Euclidean metric and correct topology and their application in object recognition and document analysis,” in *Proceedings, First European Conference on Geographical Information Systems,* Amsterdam, 1990, pp. 15-24.