

# DYNAMIC CARTOGRAPHY USING VORONOI/DELAUNAY METHODS

Chris Gold , Maciej Dakowicz

Department of Computing and Mathematics, University of Glamorgan, Pontypridd, Wales, CF37 1DL, UK  
cmgold@glam.ac.uk; mdakowic@glam.ac.uk

**KEY WORDS:** Voronoi; Delaunay; kinetic data structures; dynamic topology; interactive cartography

## ABSTRACT:

We describe two reversible line-drawing methods for cartographic applications based on the kinetic (moving-point) Voronoi diagram. Our objectives were to optimize the user's ability to draw and edit the map, rather than to produce the most efficient batch-oriented algorithm for large data sets, and all our algorithms are based on local operations (except for basic point location). Because the deletion of individual points or line segments is a necessary part of the manual editing process, incremental insertion and deletion is used. The original concept used here is that, as a curve (line) is the locus of a moving point, then segments are drawn by maintaining the topology of a single moving point (MP, or the "pen") as it moves through the topological network (visualized as either the Voronoi diagram (VD) or Delaunay triangulation (DT)). The trailing line accumulates the adjacency relationships of MP. There are thus three parts to our method: the maintenance of MP in the DT/VD; the use of MP to draw the constrained edges in the Delaunay triangulation; and the use of MP to draw the line segment Voronoi diagram. In all cases deletion is the inverse of the original drawing: move MP so as to "roll up" the desired segment. This approach also has the interesting property that a "log file" of all operations may be preserved, allowing reversion to previous map states, or "dates", as required.

## 1. INTRODUCTION

### 1.1 Objects and Fields

It is traditional in GIS to talk about "geometry" (coordinate information) and "topology" (connectivity information) separately when talking about "vector" data consisting of points, lines and polygons. More complex objects (e.g. building outlines) are composed of these elements, with attributes attached. The same approach may be used for space-covering polygon tessellations, where boundaries are shared by adjacent polygons, etc.

This requires the description of the relationships between several classes of entities, each of which may be a cartographic object (in that attributes may be associated with it) or else merely an entity linking these objects (perhaps a triangle edge in a TIN). A variety of tessellation data structures have been proposed over the years, and perhaps the most effective are edge data structures, where the edges provide the linking between points, as well as the relationships with the polygons on each side.

Where we are referring to discrete objects (e.g. houses along roads) these structures are inadequate, as they apply only to connected graphs – hence the traditional difficulties with islands within polygons. In order to take advantage of the connectivity properties of a graph our map needs to be always connected – usually, but not necessarily, as a tessellation. There are at least three ways to achieve this: with a triangulation, with a Voronoi diagram, and with a dual graph.

A simple TIN used for terrain modelling clearly is used to specify the spatial relationships between discrete data points – here the attribute information (the coordinates) is associated

primarily with the points, although an attribute such as colour may be associated with the triangular polygons. It is rare that attribute information is associated with the edges. Thus if we wish to separate our attributes (including coordinates) from our topology an edge data structure is an advantage.

If we are attempting to tessellate more complex discrete objects, such as houses, we run into difficulties as the building boundaries are clearly part of the cartographic object, as are interior triangles, while exterior triangle edges serve merely to give the relationships to adjacent buildings. This becomes particularly acute when building edges are sufficiently long that a standard Delaunay triangulation of the corner points fails to preserve the building edge, and a manually-inserted constrained triangulation edge is required. This is clearly different from the rest of the edges, and must have a label (attribute) attached. This breaks down our separation between objects (with attributes, including coordinates) and our pure topology. Our dynamic constrained DT (CDT) allows the insertion and deletion of constrained edges by tracking the topological connectivity of a moving point.

A second approach involves the construction of a Voronoi diagram of all the cartographic objects (which are themselves composed of "cartographic fragments" such as line segments and points). Each cartographic object is composed of several cartographic fragments. The internal relationships between these, and the external relationships between separate objects, are expressed by the dual triangulation. (As is well known, the dual of a VD is a DT.) Thus the DT contains the topology, completely separated from the cartographic objects and their descriptions.

The difficulties with this approach have been that the algorithms for the construction of the line-segment VD (LSVD)

are extremely complex and sensitive to the limitations of computer floating-point arithmetic. Held (2001) states that it took him ten years to achieve this for a static algorithm. Gold et al. (1995) constructed an LSVD of unconnected line segments. Imai (1996) constructed an incremental algorithm for line segments and closed polygons, where a maximum of two line segments meet at a vertex. Karavelas (2004) described a general incremental algorithm. We believe that we are the first to construct a dynamic algorithm, where points and line segments may be inserted, intersected and deleted – which is particularly useful for a dynamic GIS. The difficulties associated with this approach are primarily concerned with preserving the correct order of tangent points around the Delaunay circumcircles at locations where line segments meet. This is work is an extension of the original algorithm of Yang and Gold (1995), and is related to the CDT algorithm mentioned above.

## 1.2 Dynamic GIS

The number of dynamic algorithms for maintaining GIS data structures is limited, because they are usually based on a line-intersection spatial model, rather than a tessellation. The primary tessellation model is the Constrained DT, which is usually static, in that all vertices are added first to give a simple DT, and then constrained edges are added to give enforced boundaries. Deletion of these edges is not addressed, and even point deletion in the simple DT is only recently described (Devillers 1999, Mostafavi et al. 2003). As previously mentioned, there is no satisfactory separation of cartography and topology.

Our own approach to resolving these conflicting requirements was first of all to insist on a tessellation model, as this preserved the topological requirement of the ordering of edges around nodes. The VD/DT approach is fairly obvious as it gives a stable tessellation (and thus topology) from arbitrary vertex locations. Incremental insertion and deletion algorithms were used to give a dynamic solution.

Since line segments are also required to define cartographic features, and not just points, a kinetic model was developed which allowed the VD/DT to be maintained as the “pen” point moved through the tessellation: a line is the locus of a moving point. While the dynamic model could be developed with only the “CCW” and “InCircle” geometric predicates used in Computational Geometry (Guibas and Stolfi 1985), giving a guaranteed robustness, the kinetic model requires additional tests to handle near-degenerate cases, especially point collision. The Constrained DT was constructed in this fashion, with the additional benefit that any constrained edge could be deleted by reversing the movement of the pen. Thus building boundaries could be inserted and deleted as specific, labelled triangle edges.

In order to obtain the desired separation of cartography and topology these line segments needed to be distinguished from the triangle edges. This was achieved by creating line-segment Voronoi generators, in addition to the point generators, again by moving the “pen”. In this case, however, the result was not an arbitrary triangle edge but a line-segment object, and both point and line objects were connected by the triangulation topology. This was first reported in Yang and Gold (1993, 1995) but suffered from robustness difficulties due to the arithmetic operations needed for collision and circumcentre calculations. More recent work by Anton and Gold (1997), Anton et al. (1998) and ourselves appears to have resolved these problems, although no actual proof of robustness appears to be feasible.

This algorithmic approach has the following stages:

### 1: The Moving-Point (Kinetic) VD or DT

This requires the dynamic incremental insertion and deletion of data points. In addition, individual points may be moved from their previous location to some subsequent location – the “trajectory”. In order to maintain the VD/DT geometric properties there must be a predictive tool to specify at what location the neighbouring VD/DT edges must be modified – a “topological event” (TE).

### 2: The Kinetic Constrained DT

The Moving Point (MP) in the moving point VD is split from a previous “old” point (making two new adjacent triangles with “zero” area) and moved towards its “new” destination. The initial zero length triangle edge between the old and new points is flagged as constrained (CE), and any TE generated by the moving point is ignored if it involves switching the CE.

### 3: The Kinetic Line-Segment VD

Instead of flagging the CE in the initial position of the Constrained DT, a pair of “half-lines” is generated. These are two new generators in the VD – one for each side of the line, in addition to the two end points. Each of these is the potential generator of a Voronoi proximal region. As the MP moves, TEs are identified as before, and the topology updated, thus giving an expanding region associated with each half-line.

In this model the topological events are the same as before, but the circumcircle (CC) calculation must be expanded in order to work with distances from line segments as well as points. In earlier work (Gold, 1990, Yang and Gold, 1995) a direct calculation of Voronoi boundary intersections was used to find the circumcentre. This failed on occasion as arithmetic precision limitations could place the centre on the wrong side of a line segment, thus destroying the node-ordering necessary for topology maintenance. A new iterative algorithm was developed (Anton and Gold, 1997, Anton et al., 1998) that converged on the correct solution from an initial condition while preserving the necessary order of the generator locations around the circumcircle.

All the operations used have their inverses, as MP movement may expand or contract the trailing line (Míoc et al., 1999). Preserving the topological relationships during construction means that potential collisions may be detected in advance, and the appropriate join operations implemented. This is simplified as the lines and their proximal regions are embedded in the two-dimensional space, guaranteeing that, for example, one VD line segment may detect an imminent collision and form the appropriate junction that preserves the correct node and region ordering around the junction point.

## 2. THE KINETIC POINT VD AND ITS DUAL DT

“Kinetic” data structures maintain their topological structure while the entities move; “dynamic” ones merely permit local insertion and deletion of these entities (points, segments, etc.) Insertion algorithms are well known, but published point deletion algorithms are relatively recent.

## 2.1 The Dynamic VD and DT

The simple VD can be constructed in many ways, (Aurenhammer, 1991, Okabe et al., 1992) but the incremental algorithm has often been found to be both stable and simple (Guibas and Stolfi, 1985). In simple terms, each new point is inserted into the existing DT by first finding the enclosing triangle, using the CCW test of Guibas and Stolfi (1985), splitting it into three triangles using the new point, and then testing each edge recursively to see if it conforms to the Delaunay criterion: that neither of the adjacent triangles' CCs have an interior point. If they do, the common diagonal is switched and the new edges are added to the stack of edges to be tested. This CC test (INCIRCLE, Guibas and Stolfi 1985) can be shown to be equivalent to calculating the VD vertices and testing if the VD edges cross. Point deletion can be performed by approximately following the inverse process: switch DT edges if the result gives an exterior triangle whose CC is empty except for the point being deleted. When only three triangles remain the central point is deleted. There are two similar approaches: (Devillers, 1999 and Mostafavi et al., 2003). Thus the VD is updated at the same time as the DT. Both insertion and deletion may be considered as partitioning the DT into two parts: the valid DT exterior area and the valid DT interior area. Boundary edges are then switched until the two parts merge.

## 2.2 The Moving-point (Kinetic) VD and DT

When a point MP moves as part of a DT/VD it may either travel a short distance without requiring a topology update, or else triangle edges must be switched to maintain the Delaunay criterion. These topological events (Gold 1990, Roos 1990, Guibas et al., 1991) occur when MP moves into or out of a CC. "Real" CCs are those formed from triangles immediately exterior to the "star" or set of triangles connected to MP. "Imaginary" CCs are formed by triangles that would be created if MP was moved out of its CC, and are formed by triples of adjacent points around MP's star. Thus if MP moved into a constellation of points in a DT it would first enter the CC of a triangle, causing a triangle edge switch and adding the furthest point of the triangle to the star of MP. The original real CC is now preserved as an imaginary one. As MP continues to move, at some later time it would move out of this imaginary CC, the original triangle will be recreated and the CC will become real again.

## 3. THE KINETIC CONSTRAINED DT

Where MP collides exactly with a neighbouring point, the points are merged by removing MP along with the two triangles adjacent to the edge between these points. The reverse process may be used to create a new MP from a previous point (split). This creates a zero-length edge between them, which expands as MP moves away. In the normal course of events this edge will be switched once MP moves outside the imaginary CC formed by the previous point and its two adjacent points in the star. However, for many applications it would be desirable if the edge was preserved, and MP used to "draw" a triangle edge between two locations. In this case the trailing edge of MP is flagged "do not switch", and all tests to switch it are ignored. This generates the Constrained DT, where specific triangle edges are fixed (constrained, CE) and do not follow the DT/VD condition (Chew 1989, Shewchuk 1996).

The interesting thing about our approach, as opposed to those in the literature, is that it is incremental, allowing the addition of further points or constraints as required. A further property is that it is reversible – MP may be moved back along its (constrained) trajectory, "rolling it up" as it goes, until it reaches its starting point, with which it may be merged. Thus each operation has its inverse, giving a kinetic data structure which allows the construction and incremental or interactive modification of the desired map. In addition, the construction commands may be preserved as a "log file" for later reconstruction or modification. If timestamps are associated with each command the map may be rolled forwards or backwards to any desired time (Mioc et al., 1999).

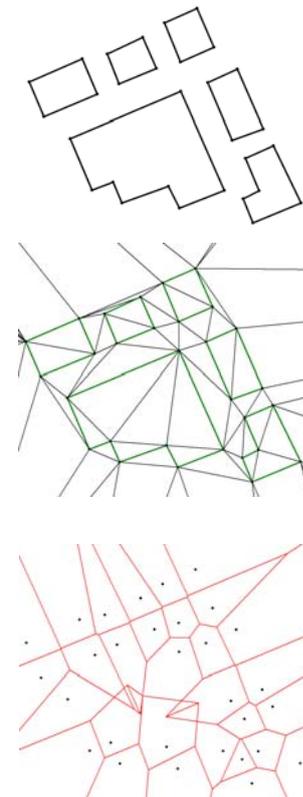


Figure 1. a) Building boundaries; b) Constrained Delaunay Triangulation; c) the Voronoi Diagram (note the two overlapping Voronoi boundaries where the constrained edge differs from the unconstrained edge)

Figs. 1a and 1b show the construction of the Constrained DT for a UK urban data set. This is of particular interest because of the work of (Jones et al., 1995, Jones and Ware 1998 and Ware and Jones 1998) who constructed the CDT of roads and building outlines and then used the adjacency information to modify and move the buildings as part of the process of map generalization. Fig. 1c shows that for this example only two constrained edges are absolutely necessary (their Voronoi edges overlap) – these prevent the switching to give a valid VD/DT – while the others would be unchanged. Fig. 2 shows the constrained DT for several buildings and roads.

#### 4. THE KINETIC LINE-SEGMENT VD

The primary problem with the Constrained DT is the confusion of entities. For a simple point DT/VD the primary objects are data points, which are the generators of the proximal Voronoi cells. The DT merely describes the dual relationships of the Voronoi edges: Delaunay edges are merely pointers expressing which pairs of data points are separated by Voronoi boundaries. For a simple TIN model it is convenient to imagine that these are geometrically defined as “straight”, as the triangle is a 2D simplex and hence forms a basis for linear interpolation within, but their real function is to support the set of equidistant boundaries that form the VD of a set of generators. (These generators may, if required, be any set of non-overlapping objects: the dual DT remains a triangulation.)

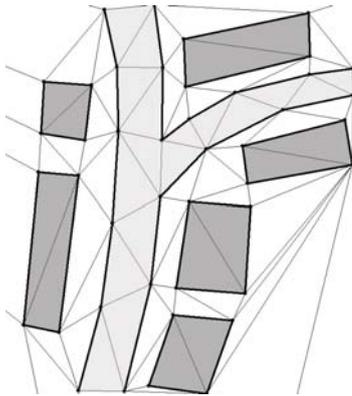


Figure 2. The Constrained DT for several buildings and roads

However, with the Constrained DT there is confusion between triangle edges that express duals of VD edges and those that have been manually added as objects – in the sense that a building outline is formed of point and line-segment objects. Thus the VD of a Constrained DT is broken at each constrained edge, and the VD edges that are correct on one side of a constrained edge are invalid when they penetrate to the other side. It is more correct to define the map objects separately (perhaps composed of points and line segments) and then to construct the DT/VD expressing the spatial relationships between them.

Unfortunately the construction of the VD of points and line segments has proved to be a difficult task, primarily due to the limited precision of computer arithmetic. This causes no great difficulty for well-separated individual line segments (e.g. Gold et al., 1995) but map objects constructed from connected points and line segments need to have tight guarantees that, for example, the circumcircle for a line segment, its end-point, and a line segment connected to that point, falls on the correct side of the polyline. (Geometrically it falls precisely on the common end-point, but topologically it must be associated with the correct side.) This has proved difficult to achieve, and workers have spent a great deal of time attempting to construct robust algorithms (e.g. Held 2001, Imai 1996, Shewchuk 1997 and Sugihara et al. 2000).

In addition, we have wanted to allow incremental, rather than batch, construction, so we followed the approach of Gold et al. (1995) which was based on the concept of the moving point VD described above. Instead of preserving a trailing triangle edge,

as described above for the Constrained DT, the “old point” OP and the “moving point” MP are connected with additional map objects: half-lines connecting OP and MP that stretch as MP moves away. Here a “half-line” is similar to the “half-edge” structure used in CAD consisting of one side of the desired line segment, in anticlockwise orientation viewed from the face. As both end points and both half-lines are map objects they are therefore generators of the VD, and thus they are vertices of the dual DT, as shown in Fig. 3. Half-lines HL1 and HL2 are linked with DT edge ne7, and they are linked by DT edges to endpoints OP and MP. In Karavelas (2004) an incremental algorithm was produced for exact arithmetic which allowed intersections by splitting line segments in advance, using exact arithmetic and implicit coordinates for intersections, as they have the original end points as support. We can not do this, as we allow arbitrary line segment deletion, but we allow MP to move from its origin to its destination and manage any collisions as it detects them. Thus calculation of the circumcircles of these triangles is more complex than for point data sets. In Yang and Gold (1995) this was calculated from the intersections of the curves forming the Voronoi boundaries, but this suffered from the arithmetic precision problems mentioned above.

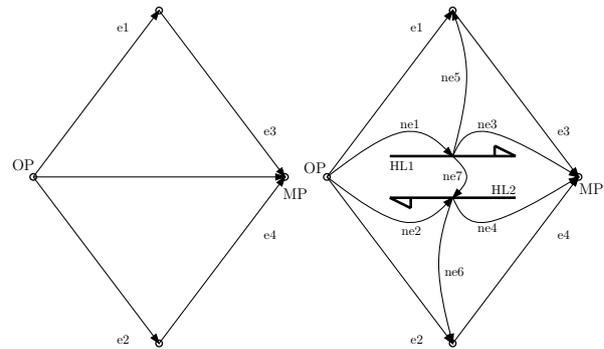


Figure 3. Half-lines between two data points

#### 4.1 Circumcircle

In our new work we use the approach of Anton et al. (1998) where the simple point circumcircle calculation was given an initial estimate based on the configuration of the points/line-segments used. Initially points and the mid-points of valid portions of the line segments were used for the INCIRCLE test. The centre was then projected onto each line segment, and a new CC calculated based on INCIRCLE. When the new circumcentre was projected onto each line, and the projection point was outside the line segment, then a point half way between the old projection point and the appropriate endpoint of the line was used. This was iterated to a suitable level of precision, and the method was guaranteed to preserve the order of the generating points around the CC, thus keeping the initial Voronoi edge order around the circumcentre (and thus the correct DT order as well), (Anton et al. 1998).

#### 4.2 Line Segment Construction

As with the Constrained DT, the MP is used to draw the line segment using the half-lines described above. As MP moves it acquires and loses Voronoi neighbours, as with the simple moving point, but when it loses them they are transferred to the trailing line segment: since this is the locus of MP, it retains all the neighbourhood relationships previously held by MP.

For a simple line segment with two end points there are four Voronoi regions: one for each end point and one for each half-line. This permits the querying of each side of a line, e.g. to find if a point is inside or outside a polygon. As shown in Gold (1990) the partitioning of the map space into proximal regions also makes buffer-zone generation an elementary operation on each region. Fig. 4 shows the line segment VD for the same urban dataset shown previously. When the DT is also displayed, note that each line segment is also a DT vertex. This clearly distinguishes the DT function of expressing the adjacency relationships, and not being part of the map object. Each of the map objects may be edited by the insertion or deletion of line segments (half-line pairs) and free vertices. The method is dynamic, in the sense of being locally updatable, and kinetic, in the sense that MP may move within the map space. However, line segments may only expand or shrink, and not sweep sideways, as collision detection and topology maintenance are based on MP alone. As with the Constrained DT, the Line-segment VD may have intersecting segments. The line segments are map objects having two separate sides (Fig. 3), allowing attributes (such as polygon colour) to be assigned to each side.

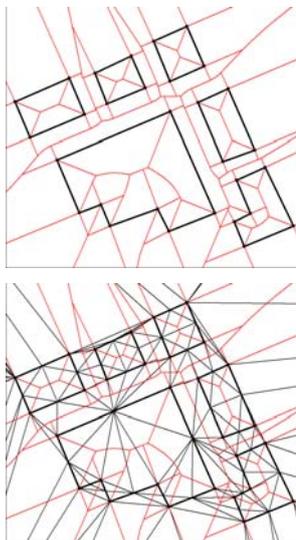


Figure 4. a) Line segment VD; b) VD plus DT for the simple buildings of Fig. 1.

Space does not permit the description of all the details of the suite of algorithms described in this paper. The key question in practice is the robustness of the method for all types of data input, given the problems of arithmetic precision. The underlying method described here consists of two parts: a geometric test and a topological update. Any arithmetic operation not resulting in a topological change causes no robustness problems. Only geometric tests used to trigger topological changes can cause robustness problems, and there are only two – calculation of CCs and a sidedness test (“walk” in Gold et al. 1977, “CCW” in Guibas and Stolfi 1985). These problems are described in more detail in Gold and Dakowicz (2006).

Our previous examples have been urban applications, showing building and street boundaries, for potential application in map generalization (Jones et al. 1995, Jones and Ware 1998). We will briefly show one other. Fig. 5 shows the Line-segment VD for a portion of a contour map. Both the points and line-

segments forming the contours are map objects, as would have been the intention of the compilers. In addition, the medial axis, or skeleton, between or within the contours is clearly seen (see Gold and Snoeyink (2001) for further discussion of the skeleton). This map is directly editable if required.

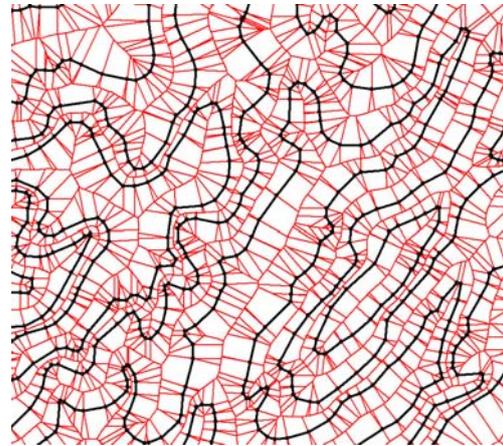


Figure 5. Line-segment VD for contours

## 5. CONCLUSIONS AND ACKNOWLEDGEMENTS

We have attempted to show that a tessellated spatial model has definite advantages for cartographic applications, and facilitates a kinetic structure for map updating and simulation. Firstly, the moving-point DT/VD model approximates human thinking, and manages collision detection, snapping and intersection at the data input stage by maintaining a topology based on a complete tessellation. Secondly, the Constrained DT allows the simulation of edges, and not just points, with only minor changes to the moving-point model, but at the cost of confusing map objects and topological entities. Thirdly, the Line-segment VD is a better-specified model of the spatial relationships for compound map objects built from points and line segments than is the Constrained DT. However, until now it has been more difficult to develop. We believe that this method is now viable for 2D cartography, and in many cases it should replace the Constrained DT. However, whichever method is used, the concept of using the moving point as a pen, permitting interactive navigation within the map under construction, together with the ability to delete and add line segments as desired in the construction and updating process, appears to be a very useful approach.

Resolution of these issues opens up a variety of elegant solutions to cartographic problems: the point in polygon problem; the buffer zone problem; the watershed and cumulative catchment area problems; feature adjacency problems; and incremental and interactive updating of the cartographic product with only local changes to the topology and hence to the properties just described. In addition, as described in Gold and Yang (1993) and Mioc et al. (1999) all updates may be time-stamped, and hence the map may be updated or rolled back to any desired date.

We would like to acknowledge the financial support of the EU Marie-Curie Chair in GIS at the University of Glamorgan.

## REFERENCES

- Anton, F. and Gold, C. M., 1997. "An iterative algorithm for the determination of Voronoi vertices in polygonal and non-polygonal domains", In Proceedings, Ninth Canadian Conference on Computational Geometry, Kingston, ON, Canada, pp.257-262.
- Anton, F., Snoeyink, J. and Gold, C. M., 1998. "An iterative algorithm for the determination of Voronoi vertices in polygonal and non-polygonal domains on the plane and the sphere", In Proceedings, 14th European Workshop on Computational Geometry (CG'98), Spain, pp.33-35.
- Aurenhammer, F., 1991. "Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure", ACM Computing Surveys, v. 23 (3), pp. 345-405.
- Chew, P., 1989. "Constrained Delaunay Triangulations", *Algorithmica*, v. 4 pp. 97- 108.
- Devillers, O., 1999. "On deletion in Delaunay triangulations", the 15th Annual ACM Symposium on Computational Geometry, pp. 181-188.
- Gold, C. M. and Dakowicz, M. 2006. Kinetic Voronoi - Delaunay drawing tools. Proceedings, 3<sup>rd</sup>. International Symposium on Voronoi Diagrams in Science and Engineering, Banff, Canada July 2006, pp. 76-84.
- Gold, C. M. and Snoeyink, J., 2001. "A one-step crust and skeleton extraction algorithm", *Algorithmica*, v. 30, pp.144-163.
- Gold, C. M. and Yang, W., 1993. The design of an urban GIS to manage frequent spatial updates. In D. Du, J. Chen, B. Forester, and X. Shi, editors, Proceedings AUSIA'93 Advances in Urban Spatial Information and Analysis, pages 1-7, Wuhan, China.
- Gold, C. M., Charters, T. D. and Ramsden, J. (1977), Automated contour mapping using triangular element data structures and an interpolant over each triangular domain, In Proceedings: Siggraph '77. Computer Graphics, v. 11, pp.170-175
- Gold, C. M., Remmele, P. R. and Roos, T., 1995. "Voronoi diagrams of line segments made easy", In Proceedings, 7th. Canadian Conference on Computational Geometry, (Eds.: Gold, C. M. and Robert, J. M.), Quebec, QC, Canada, pp.223-228.
- Gold, C.M., 1990. "Spatial Data Structures - The Extension from One to Two Dimensions", In: L.F. Pau (ad.), Mapping and Spatial Modelling for Navigation, NATO ASI Series F No. 65, Springer-Verlag, Berlin, pp. 11- 39.
- Guibas, L. and Stolfi, J., 1985. "Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams", *Transactions on Graphics*, v. 4, pp. 74-123.
- Guibas, L., Mitchell, J.S.B. and Roos, T., 1991. "Voronoi diagrams of moving points in the plane", Proceedings, 17th. International Workshop on Graph Theoretic Concepts in Computer Science, Fischbachau, Germany. Lecture Notes in Computer Science, vol. 70. Berlin: Springer-Verlag, pp. 113-125.
- Held, M., 2001. "VRONI: an engineering approach to the reliable and efficient computation of Voronoi diagrams of points and line segments", *Computational Geometry, Theory and Application*, v. 18 (2), pp. 95-123.
- Imai, T., 1996. "A topology oriented algorithm for the Voronoi diagram of polygons", In Proceedings, 8th Canadian Conference on Computational Geometry, Carleton University Press, Ottawa, Canada, pp. 107-112.
- Jones, C. B., Bundy, G. L. and Ware, J. M., 1995. "Map generalization with a triangulated data structure", *Cartography and Geographic Information Systems*, v. 22 (4), pp. 317-331.
- Jones, C.B. and Ware, J.M., 1998. "Proximity Search with a Triangulated Spatial Model", *Computer Journal*, v. 41 (2), pp. 71-83.
- Karavelas, M.I., 2004. "A robust and efficient implementation for the segment Voronoi diagram", *International Symposium on Voronoi Diagrams in Science and Engineering (VD2004)*, pp. 51-62.
- Mioc, D., Anton, F., Gold, C. M. and Moulin, B., 1999, "'Time travel" Visualization in a Dynamic Voronoi Data Structure", *Cartography and Geographic Information Science*, Vol. 26, pp. 99-108.
- Mostafavi, M., Gold, C.M. and Dakowicz, M., 2003, "Dynamic Voronoi/ Delaunay Methods and Applications", *Computers and Geosciences*, v. 29 (4), pp. 523-530.
- Okabe, A., Boots, B. and Sugihara, K., 1992. *Spatial Tessellations - Concepts and Applications of Voronoi Diagrams*, Chichester: John Wiley and Sons, 521p.
- Roos, T., 1990. "Voronoi Diagrams over Dynamic Scenes", Proceedings, Second Canadian Conference on Computational Geometry, Ottawa, pp. 209-213.
- Shewchuk, J.R., 1996. "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator", First Workshop on Applied Computational Geometry (Philadelphia, Pennsylvania), Association for Computing Machinery, pp. 124-133.
- Shewchuk, J.R., 1997. "Adaptive Precision Floating-Point Arithmetic and Fast Robust Geometric Predicates", *Discrete and Computational Geometry*, v. 18 (3), pp. 305 - 363.
- Sugihara, K., Iri, M., Inagaki, H. and Imai, T., 2000. "Topology-oriented implementation - an approach to robust geometric algorithms", *Algorithmica*, v. 27 (1), pp. 5-20.
- Ware, J.M. and Jones, C.B., 1998. Conflict Reduction in Map Generalization Using Iterative Improvement, *GeoInformatica*, v. 2 (4), pp. 383 - 407.
- Yang, W. and Gold, C. M., 1993. An outline of a dynamic solution to spatio-temporal windowing in an urban GIS. In D. Du, J. Chen, B. Forester, and X. Shi, editors, Proceedings AUSIA'93, Advances in Urban Spatial Information and Analysis, pp. 36-44, Wuhan, China.
- Yang, W. and Gold, C. M., 1995. Dynamic spatial object condensation based on the Voronoi diagram. In J. Chen, X. Shi,

and W. Gao, editors, Proceedings Fourth International Symposium of LIESMARS'95 - Towards three-dimensional, temporal and dynamic spatial data modelling and analysis, pp. 134-145, Wuhan, China.