AN EVENT-DRIVEN APPROACH TO SPATIO-TEMPORAL MAPPING

Christopher M. Gold, Centre de recherche en géomatique
Pavillon Casault, Université Laval, Québec, Québec

There are many problems involved in trying to model space dynamically in a computer. Since it is impractical to preserve the states of all possible locations in space and time, the preservation of changes of state is a feasible alternative. A (very preliminary) taxonomy of possible types of maps is given, along with the state-change conditions that should be preserved. Traditional static maps are limited in this regard, and the dynamic Voronoi diagram is discussed as a mechanism for managing changes of state, both over time while holding spatial location constant, and while moving through space while holding time constant.

On rencontre de nombreuses difficultés en essayant de modéliser dynamiquement l'espace dans un ordinateur. Puisqu'il est peu réaliste de préserver les états de tous les sites possibles dans l'espace et dans le temps, la préservation des changements d'état constitue une meilleure solution. On fournit une taxonomie (très préliminaire) des types de cartes possibles, de même que les conditions de changement d'état qui devraient être préservées. Les cartes statiques traditionnelles sont limitées à cet égard et on présente le diagramme Voronoi dynamique comme mécanisme de gestion des changements d'état, dans les deux cas suivants : en se déplaçant dans le temps tout en maintenant la position spatiale constante et en se déplacant dans l'espace tout en maintenant le temps constant.

Introduction

The computer programs called "GIS" (Geographic Information Systems) were originally developed from computer-assisted mapping tools, with an emphasis on choropleth mapping - for example, see the early work at Harvard University, described in *Chrisman et al.* [1992]. This led to vector mapping tools, again with an emphasis on choropleth maps, together with a variety of contouring and interpolation programs. GIS as we know them were formed by the integration of these vector mapping tools with a database management system. Still later terrain and network tools were added.

Confusingly, the academic discipline also called GIS arose out of the development of algorithms for automated cartography, as a way to display spatial information - often of census or similar data. With the development of available commercial systems the emphasis on algorithms declined, and was replaced by an emphasis on spatial analysis using the available tools. However, the linking of a discipline with a specific technology is always an uncomfortable fit, and today's interest in spatio-temporal issues makes it unlikely that the discipline (perhaps better named Geomatics) and the technology will be quite so closely linked in the future.

This paper addresses issues of managing space and time. From the preceding paragraph, it would be unfortunate to identify the research described solely as "GIS". The basic questions are: what are the issues and properties of space and time that we need to preserve in the computer in order to perform some type of "analysis"; and what data structures and methods are necessary to implement them?

Despite much recent work in spatial information theory (e.g. *Frank and Kuhn [1995]; Edwards* [1996]), there is as yet no common set of elements and operators that is agreed upon when discussing space and time. To take but an obvious example: space/time is continuous (at least at the level of generalization to which we are accustomed), and regions may be subdivided to an unlimited extent. This fits uncomfortably with a database model based on discrete objects and properties. In addition, what is "analysis"? In practice this often means "answering the types of question GIS were built for." These questions tend to be global in nature - e.g. the overlay of various polygon sets to find the regions suitable for some particular activity. However, it is arguable that the most widespread form of spatial analysis - used by

GEOMATICA Vol. 50. No. 4,1996. pp. 415 to 424

almost all sentient beings - is local: the ability to navigate within the locally perceived spatial obstacle course. With this model the observer is embedded within Euclidean space. (It may be noted here that two dimensions are often enough, due to the vertical effects of gravity.) In addition, there is evidently some form of "topology" involved, as well as metric information, in the perception of the relationships between the obstacles, and between the obstacles and the observer. A purely "database" approach is not necessarily appropriate, as this precludes direct topological relationships between objects. The "navigation" model just described, while clearly a valid approach to some spatial queries, does not fit comfortably within traditional structures of database management or GIS, and must be examined further.

Navigation is fundamentally temporal in nature: it takes time to move and to observe change.

Navigation is fundamentally temporal in nature: it takes time to move and to observe change. However, it is a matter of interpretation as to whether the observed change is a function of spatial position or of elapsed time. When traveling along a road we interpret observed change as due to change in spatial position, but when revisiting a childhood home or watching a storm we interpret the change as due to elapsed time. Thus in practice we may interpret change (or perhaps variation is a better word in this context) in the attributes of the objects around us as being over space or over time. This variation is in the sequential views received by the embedded observer, with interpretation depending on the context. The only presupposition is the ongoing existence of the observer, who traces a path in space/time (see Edwards [19961).

If we apply the navigational model to the study of maps, some interesting issues arise. Maps express variation across space. Cartography has attempted in various ways to express the existence of physical objects (e.g. houses); imaginary objects (e.g. administrative districts); and continuous fields (e.g. temperature, elevation). A moving observer would thus detect continuous variation in a field such as temperature, or alternatively an abrupt variation upon encountering a discrete object such as a house. If however we allow the observer to stay still and await events, he may detect continuous events such as temperature variation, and/or discrete variation such as falling rocks or moving cars. This time dimension is not readily handled by traditional GIS.

There is therefore a direct parallel between events occurring over time. and the equivalent over space, as the distinction is largely a matter of interpretation. For both cartography and our observer, the key objective is to record variation when observed. In a computer context, recording variation is the most feasible way to reduce the storage of the space/time continuum to manageable proportions. The next sections therefore attempt to describe a variety of maps that may already be managed, or that could be managed in a futuristic setting, in terms of their variation. These include current static maps, as well as maps that vary over time. Following *Egenhofer and Al-Taha [* 1992], variation may apply to a polygonal object's boundary or its interior, and may be abrupt (discrete) or continuous (smooth). Most of the examples are of forest maps, as that industry has a particular interest in variation over time. Subsequent to that discussion, a particular implementation of the navigation model will be examined, to see to what extent it is applicable to the problem.

Modeling Variation in Space and Time

If a map changes over time, the representation in the computer simulates this if it tracks the variation in the real world. At the minimum one redraws only those portions of the graphic display of the map that have changed. In the case of a simple vector map a locally updatable topological structure is a big advantage, in that it eliminates the need for snapshots of the whole map at each time interval. For example, if the data states that a particular forest polygon boundary changes between one time step and another, the old arc must be moved (or deleted and re-drawn) at the appropriate simulation time. This gives the "map as a movie'* approach. In this case the update history of the map may be replayed (or played backwards) by examining the log file of the update commands [Gold 1994].

We have proposed [Gold 1993] that simulation (or variation in space with time), may be modeled within the computer not by snapshots of the whole map at regular time intervals, but by local update of the location of spatial variation, at the time that this occurred. A detected variation may be termed an *event*. In addition, in order to preserve our spatial model, the topological structure must be maintained at all times. This differs from other workers [*Langran* 1992, 1993; *Peuquet and Duan* 1995] where the emphasis has been on the management and query of spatio-temporal information within the database, without discussing in particular the local spatial relationships. *Snodgrass* [1992] is a well known reference on temporal databases in general, and *Armenakis* [1992], and *Worboys [1992a,* b], have discussed various aspects of the organization and modeling, of spatio-temporal data, as have several other workers,

To clarify our terms. we can define several types of variation within the map.

The first type is variation in attribute over space (Z/S). A simple continuous example is the gradient, at any location, of a terrain model. A discrete example is the jump in the attribute value at a polygon boundary in a choropleth map. The second type is variation in spatial location over time (S/T). This would be a discrete jump when a map boundary is updated by redigitizing, or a continuous change over time if a point movement simulation was being performed. Examples are robot navigation or free-Lagrange fluid flow modeling *[Gold and Condal* 1994]. The third type is variation in an interior of a forest stand over time (Z/T). The terminology **used** (Z/S etc.) is intended to reflect calculus symbolism *(dZ/dS* etc.), but without making any statement about the level of continuity of the variables.

There are two qualifiers that may be appended to these three types of variation. The variation may apply either to the boundary or the interior of a (polygonal) object. Thus the first qualifier may be a *"b"* or an *"i"* respectively although *"b"* does not apply to a raster representation of a field, and *"i"* does not apply to point or line objects. In addition, the variation may be discrete (non-differentiable, e.g. at the boundary between two polygons with differing attributes), or continuous (differentiable, for example at any location on a **surface** defined by a mathematical function). Thus the second qualifier may be a *"d"* **or a "c"** to express those conditions. Not all combinations make sense with any particular type of map, but, since the alternative is to store all values of the attribute for all combinations of spatial location and time, it is desirable to store only non-zero variations at particular locations in space/time, and these will be subsets of the classes just described. Some examples will hopefully clarify the concept. Note that in some cases, especially after the necessary discretization of a continuous spatial function, there may be more than one way to represent the change.

Examples of Variation in Maps

- A static polygon map, classifying a region into various categories: save *Z/S(b,d).*

A traditional choropleth map illustrates discrete variation over space at the polygon boundaries - "walking" across the map, an observer would notice differences only at boundaries. Thus saving each boundary location and the values on each side would suffice to describe the map. Of course, other topologically-equivalent alternatives, such as pointers to the polygon interior associated with each attribute, are possible.

- A forest map, with simulation of forest growth within each polygon: save *Z/S(b,d)* and *Z/T(i,c).*

Here a complete model of variation within the map would require the boundary, as marking the location of spatial variation, as well as the polygon interior, associated with variation of the attribute (forest growth). The topological link between the boundary and the polygon interior would be the most desirable way of identifying variation across the boundary at any particular moment in time.

- The forest map as above. but with simulation of boundary spread: save *S/T(b,c)* as well.

The rate of movement of the boundary of one forest type encroaching on another would need to be preserved for each boundary. Note that at some moment in the simulation individual boundaries will appear or disappear, giving "topological events" as well *[Roos* 1990]. In the case of the approximation of continuous boundary movement by the re-digitizing of forest stand boundaries every few years during forest inventories, the continuous variation *S/T(b,c)* will become discrete *S/T(b.d).*

- A contour map, or other representation of a smooth surface: save *Z/S(i,c)* throughout.

Here the gradient of the surface should be preserved at all possible locations. This not being possible, except implicitly for mathematically defined functions, approximations are made. In one case a grid is used to partition space into polygons, permitting the surface to be represented as a choropleth map, but using implicit boundaries. The other common alternative is the contour map, where the contour lines themselves, implying zero variation along their length, indicate a maximum variation (of the attribute over space) perpendicular to themselves.

- An undulating surface, varying over time: save *Z/S(i,c)* and *Z/T(i,c)* throughout.

Here, as before, in the ideal model of space there would be no discretization but variation of the mapped attribute over space, and over time,

> To clarify our terms, we can define several types of variation within the map-

would be continuous within the map interior. In the grid approximation Z/T(i,c) would be saved at each grid cell, whereas in the contour representation S/T(h,c) would preserve the displacement of contour lines over time.

- A finite-difference flow model, with water movement between fixed cells: save Z/T(i,c) and Z/S(b,d).

  This is the same situation as for the forest growth model, with variation across boundaries as well as within each cell. Note that flow across the boundary is a function of the gradient between cells. In finite difference methods this is presumed not to be infinite at an abrupt boundary, but to be defined by the attribute difference between cells divided by the spatial separation of the cells' "centroids".

- A free-Lagrange type of flow simulation *[Fritts, Crowley and Trease* 1985], where packets of air or water interact with each other and move: as above, but save *S/T(b,c)* also.

  This is the same case as the forest model with moving boundaries. The boundaries are often represented by Voronoi cells generated automatically around the "centroids" representing the centers of mass. Thus *S/T(b,c)* for each boundary may be generated from the relative motions induced in each centroid. In addition, topological events also occur as cell boundaries appear and disappear.

- Robot navigation through a field of obstacles: save Z/S(b,d) and *S/T(b,c)* (if using the Voronoi navigation model [Gold *and Condal 1994]).*

  Here again we have the case of the forest model with moving boundaries, but without attribute variation 'over time within each cell. Attributes associated with each object and its surrounding cell are usually object labels. Boundaries are generated automatically, and the primary information required is the set of spatial relationships of pairs of objects with a common boundary (or of edges in the equivalent dual triangulation). This gives the variation Z/S(b,d). *S/T(b,c), as* in the free-Lagrange method, which may be represented by the relative motions of the generating objects. For a single moving robot only one object will be moving at a time. Thus the triangulation plus the current movement of the robot suffices.

The process of partitioning space and time involves an inevitable loss of information.

Discretization as a Transformation

The process of partitioning space and time involves an inevitable loss of information. This may occur at the data collection stage, as in census data aggregation by district, or within the computer, as with gridding. This discretization may be treated as a series of transformations. A preliminary list is given below. Note that each moves from a continuous model towards a discrete representation, and that there may be more than one path between two states.

- *Z/T(i,c)* may be discretized over *T*, giving *Z/T(i,d).* Thus the value of Z within a region will jump between time steps.
- *Z/T(i,c)* may be discretized over Z, giving *Z/T(i,d)* again. Here the partition of Z into categories determines the time step used to display the variation.
- *Z/S(i,c)* may be partitioned by S, giving Z/S(b,d). Thus a smooth surface in continuous space becomes a step function with variations at boundaries.
- *Z/S(i,c)* may be partitioned by Z, giving Z/S(b,c) as a first step. Here contour lines, having continuous normal gradients, partition the surface.
- Z/S(b,c) may be further partitioned by Z, giving Z/S(b,d). In this context, this second step consists of the partitioning of Z into categories, therefore giving a step function.
- Z/S(b,c) may be partitioned by S, giving Z/S(b,d). A typical example is the gridding of a contour map.
- S/T(b,c) may be partitioned by *T*, giving *S/T(b,d).* An example is robot movement or free-Lagrange flow modeling with an iteration for each time step.
- *S/T(b.c)* may be partitioned by S, giving *S/T(b,d)* again. Here the boundary is moved when it is to be drawn within the next screen pixel, for example.

The Inverse Process

We have been talking about the transformation from continuous to discrete representation of space and time as we move from the real world to the computer, based on both data collection and digital data representation issues. The inverse process may also occur: a discrete set of observations within the computer may be used to generate a continuous representation of the phenomenon. This is normally called interpolation. It may assume that the "fields" generated by each observation are non-overlapping (i.e. only one may occupy any spatial location at any one time), or overlapping. In the first case we have a surface precisely matching each generating object [Gold and *Roos* 1994] whereas in the second case fields may penetrate each other and the generating objects, giving surface approximations that will not usually correspond to the data point observations. The Voronoi "area-stealing" approach, as it embeds objects and proximity fields within the same spatial structure, is of the first type, whereas traditional "gravity model" interpolation is of the second type. It should be noted that this is not a precise inverse transformation from discrete to continuous representations. The real world has potential observations at all possible locations, whereas the interpolation process merely generates a representation at any desired location.

Simulation of Process within Geographic Space and Time

Once we have created a model of variation over space and time we may represent the results of our simulation in a fashion that mimics, to a known degree, the behavior of space in the real world. Simulation of a physical process then becomes the development of numerical models to describe the variations Z/S, *Z/T* and *S/T* as required, taking into account the transformations induced by the discretization process. The mathematical functions describing the geographical processes are application specific, and largely dependent on the spatial partitioning process chosen. Examples have been given above for forest modeling, finite-difference groundwater flow, free-Lagrange flow methods and robot navigation, but this is only a small subset of the possibilities. In many of these cases the forces acting on entities within our spatial model are local, based on the set of neighbors, rather than global. The structures suggested should facilitate the modeling of local interactions - a form of "virtual reality" in two dimensions. This is a developing field, with much to be done.

Summary - An Event-Driven Approach

We have identified three forms of variation - Z/S *Z/T* and *S/T,* together with possible topological events. These may refer to continuous or discrete variation, and may refer to a region interior (a field) or an object (often a boundary). Not all combinations appear useful at present. *Z/T* applies to interiors, not boundaries (as well as unbounded fields), while Z/S and *S/T* apply to boundaries alone within the digital model of space.

Z/T(i.c), Z/S(i,c) and *S/T(b,c)* are taken to represent the continuity of space and time in the real world. In the previous descriptions of map types all "c" type representations will need to be converted to "d' for actual computer implementations, as digital computers are inherently discrete. Spatial boundaries used for the discretization of space may be explicit (as in digitized census districts) or implicit (as in grid cells or Voronoi cells). Attribute variation across boundaries (Z/S) is best represented by referring to the attribute values of the left and right region interiors, using some appropriate topological structure.

The three types of variation in space and time need to be modeled in the computer if we are to handle the types of "maps" described above. Indeed, at this point there appears to be little difference between dynamic maps and simulation, just as in flow-modeling there are equations both for steady-state and for transient conditions. The detection of a non-zero variation in Z/S, *SIT* or *Z/T* may be recorded as an event, equivalent to the "topological event" of *Roos* [1990]. Thus a general-purpose system to manage spatio/temporal mapping and simulation should be an event-processing system, triggered by topological or variation-detection events. As with object-oriented programming, the intention is to take the appropriate action upon receipt of messages describing detected variation in the four categories below:

- Z/S : attribute over space - drawing map boundaries, contours, etc.
- *S/T* : location over time - managing the dynamic movement of map objects.
- *Z/T* : attribute over time - recording varying attribute values.
- Topological events - recording changing neighborhood relationships,as in *Roos* [1993].

We have identified three forms of variation - **Z/S, Z/T** and S/T, together with possible topological events.

One approach that is capable of handling this kind of information, because it was designed to handle the navigation approach described earlier, is the dynamic Voronoi diagram. This approach will be summarized, and then its properties will be compared with the event-processing model just described.

The Voronoi Implementation

The dynamic Voronoi approach, as outlined in various papers [Gold 1993], [Gold 1994], [Gold *and Condal* 1994] was designed as a locally modifiable spatial structure, based on a well defined set of relationships with neighboring objects. It is based on the better known static Voronoi data structure *[Aurenhammer* 199 1], [Okabe et *al.* 1992]. It has been described previously for static applications *[Gold* 1991], but its use in managing variation over time requires elaboration.

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 l), and diagrams where the cells are generated by line segments as well as points have been developed by various researchers *[Gold* 1991], *[Okube et al.* 1992] (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.

Roos [1990] described the structure for maintaining the modification of the topological structure as a set of points moved. 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 [* 1990] used the same approach for the movement of one point at a time in the generation of line segments.

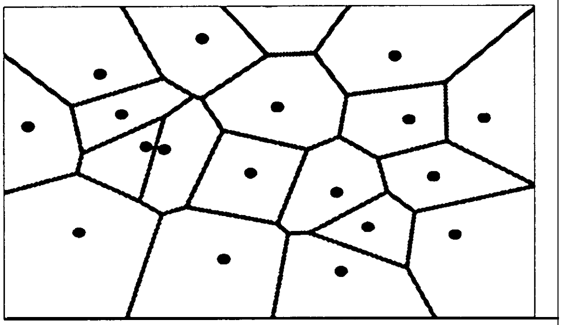

Figure 1: Simple point Voronoi diagram.

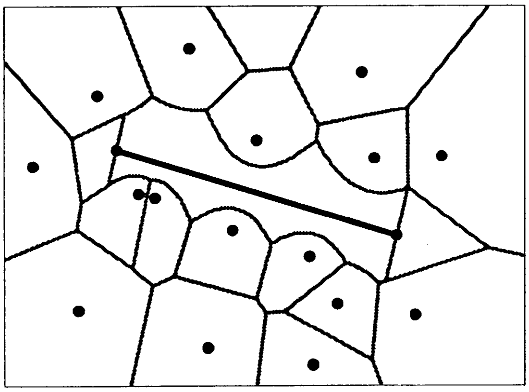

Figure 2: Addition of one line segment.

Being dynamic, this form of Voronoi diagram can accommodate object displacement over time. This makes it relatively attractive as a structure for simulating geographic processes, as object displacement over space (S/T) is readily available. The area-stealing method of interpolation, using the Voronoi method *[Gold and Roos* 1994], allows the local discretization of space to be used to reconstruct variation in attribute over space (Z/S). The variation in attribute value over time (Z/T) has traditionally caused few problems. Thus the Voronoi model is one possible spatial structure for representing a wide range of spatial processes within a digital environment, while traditional polygon/arc/node structures have a more limited use.

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 neighbor, 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 *Chrisman et al.* [1992], for example.

Sufficient experience has been gained using this technique in the last 15 years to identify its weaknesses. Basically, if an intersection is not found, then there is no knowledge of neighbors. Detection of intersections is 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 map objects. This implies that topological events, occurring at specified times, can not easily be handled, as they must be added or removed one at a time.

The Voronoi Dynamic Spatial Data Structure

The Voronoi spatial data structure has developed in response to these perceived limitations - 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 neighbors - 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 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 cause implementation difficulties. By emulating manual methods we naturally use 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 ***Gold [*** 1993], Green ***and Sibson [*** 1978]. 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 gives us the opportunity to construct more complex objects later on.

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 any dynamic maintenance.

Two points may be made about the differences between traditional raster and vector methods.

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 neighboring vertices to point ***P.*** Each adjacent triangle has one immediate exterior neighboring triangle, with two vertices in common with the adjacent triangles, and the exterior one.

When ***P*** is moved a small distance towards Q, it modifies the shape of the boundaries between its Voronoi region and those of its neighbors. These boundaries are represented as the triangle edges connected to ***P*** in the dual triangulation. While modified, all the boundaries still exist, and hence the dual triangulation remains unchanged. For a somewhat larger perturbation of ***P,*** however, its 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 neighbor 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 *Gold [* 1990], *Gold* [1991] and Gold [1992a]. The circumcircle criterion follows from one possible definition of the Delaunay triangulation - that all its circumcircles 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.

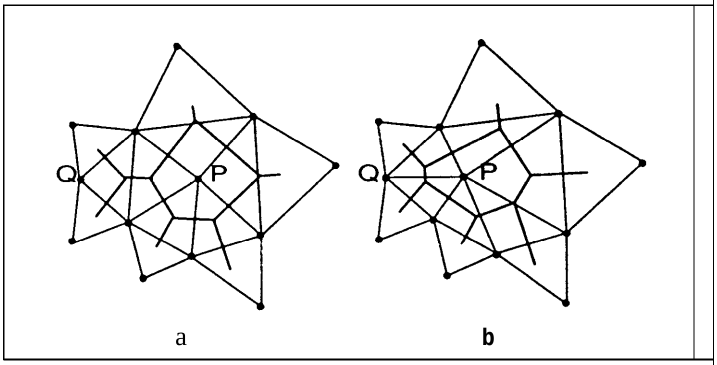

Figure 3: In (a) points P and Q are not neighbors; in (b) they are.

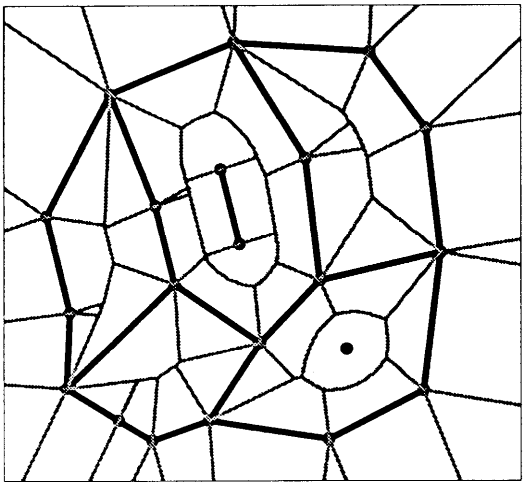

Figure 4: Voronoi diagram of a polygon set, plus a free point and line segment.

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

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 neighbor to the moving point is known as a "topological event" *[Roos* 1990; *Roos* 1990; *Bajaj and Bouma* 1990] 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 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 variation over space and time. The navigation approach enabled by the Voronoi spatial model allows the detection of internally-generated events, as when the operator of a manual digitizing system attempts to draw one line across another. The system is able to detect this, and to query the operator as to his intentions: is this to be a line intersection or not? Collisions may also be detected in simulation experiments, where point objects meet boundaries.

The Log File View of a Map

The Voronoi system reacts to a limited number of messages in order to create or modify the **map** [Gold 1992a; Gold 1994]. The point move operation (with triangle switches) is common to all of them. There are three initial conditions: just move the point; split off a new point and move that; and split off the new point and move it, while trailing a line segment behind. The three possible final conditions are the reverse: just stop moving the point; merge the point with some destination object (eliminating the point); and merge the point with the destination object while eliminating any line segment that was connecting the two. This gives nine possible combinations of initial and final conditions. These nine commands may be used to build and modify the map, and they may also be preserved in a log file, with a date stamp appended. At a later time this log file - described in *Gold and Condal [1994]* - may be used to rebuild the map up to some specified date. (At present this is only implemented for database time, not for real time.) Indeed, the log file may be played backwards as well as forwards. allowing a basic roll-back function. In addition, internally generated events, such as line intersections, will be recorded. There were four types of event management needed to handle the previously described spatio/temporal maps:

- Z/S : attribute over space - drawing map boundaries, contours, etc.
- SIT : location over time - managing the dynamic movement of map objects.
- *Z/T* : attribute over time - recording varying attribute values.
- Topological events - recording changing neighborhood relationships. \end{ itemize}

Of these, Z/S events, *SIT* events and topological events are already managed by the Voronoi command processor. *Z/T* events may be readily added, recording when the attribute of some particular object varied. The boundary “b” and the interior “i” of a Voronoi object are distinguished by assigning attributes to the generator (point or line segment) or to the Voronoi tile. Thus attributes may be assigned to a set of line segments representing a road, or to the tiles on one side of a set of line segments representing a polygon.

However, as described, these operations are associated with discrete *“d”* as opposed to continuous “c” variations. Work is in progress to implement all the *“d”* event handlers, but more research is required to define “c” event handlers. It is hoped that progress will be possible in these cases as well, as point movement may be managed in a pseudo-continuous fashion by specifying small steps, and interpolation may be managed using the “area-stealing” approach described in *Gold and Condal [1994].*

The Voronoi system reacts to a limited number of messages in order to create or modify the map.. . .

Conclusions

The different types of variation - Z/S, *Z/T*, and *S/T* - have been examined, and several basic transformations have been described that result from the discretization enforced by data collection and processing within the digital computer. 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. The generality of the approach is based on the digital simulation of space and time themselves, with an adequate understanding of the transformations imposed by the necessary discretization processes. By storing events (variation) in the spatio/temporal continuum, storage costs become at least feasible. The use of the log file as a record of events opens the possibility of true event-driven models for mapping space/time. The event-driven approach described here corresponds closely to the architecture of ter *Haar* [1993], where he describes “Dynamic GIS” as being based on the concepts of interactive on-line topology, dynamic function-based attributes, attribute based symbology, and dynamic triggers.

Acknowledgments

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

Armenakis, C. 1992. Estimation and organization of spatio-temporal data. *Proceedings, Canadian Conference* on GIS, Ottawa, March 1992, pp. 900-911.

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

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.

Bowyer, A. 1981. Computing Dirichlet tessellations. *Computer Journal, 24,* pp. 162-166.

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.

Edwards, G. 1996. Geocognostics - a new paradigm for spatial information? *Proceedings, AAAI Spring Symposium,* March 1996, Stanford University.

Egenhofer, M.J. and K.K. Al-Taha. 1992. Reasoning about gradual changes of topological relationships. Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, *Lecture Notes in Computer Science* 639, Springer, Berlin, pp. 196-219.

Frank, A.W. and W. Kuhn. (Eds.), 1995. Spatial Information Theory - A Theoretical Basis for GIS. *Proceedings, COSIT'9.5.* Semmering, Austria, September *1995, Lecture Notes in Computer Science 988.* Springer, Berlin, 568 p.

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

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.

Gold. C.M. 1991. Problems with handling spatial data - the Voronoi approach. *CISM Journal ACSCC, 45(*1), pp. *65-80.*

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.

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.

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.40,pp. 121-128.

Gold. C.M., and A.R. Condal. 1994. A spatial data structure integrating GIS and simulation in a marine environment. *Marine Geodesy* 18, pp. 213-228.

Gold, C.M. and T. Roos. 1994. Surface modelling with guaranteed consistency - an object-based approach. IGIS'94: Geographic Information Systems. *Lecture Notes in Computer Science 884,* Springer-Verlag, Berlin, pp. 70-87.

Green, P.J., and R. Sibson. 1978. Computing Dirichlet tessellations in the plane. *Computer Journal 2* 1, pp. 168-173.

Langran, *G. 1992. Time in Geographic Information Systems.* Taylor and Francis, New York.

Langran, G. 1993. Manipulation and analysis of temporal geographic information. *Proceedings, Canadian Conference on* GIS, Ottawa, pp. 869-879.

Okabe, A., B. Boots and K. Sugihara. 1992. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams.* John Wiley and Sons, Chichester, 532 p.

Peuquet, D.J. and N. Duan. 1995. An event-based spatio-temporal data model (ESTDM) for temporal analysis of geographical data. *International Journal of Geographical Information Systems, 9,* pp. *7-24.*

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

Roos, T. 1993. Voronoi diagrams over dynamic scenes. *Discrete Applied Mathematics, 43,* pp. *243-259.*

Snodgrass, R. 1992. Temporal databases: Theories and methods of spatio-temporal reasoning in geographic space. *Lecture Notes in Computer Science 639,* Springer, Berlin, pp. 22-64.

ter Haar, P.W. 1993. Dynamic analysis for dynamic problems. *Proceedings, EGIS'93:* Fourth European Conference on Geographical Information Systems, Genoa, March 1993, pp. 746-753.

Worboys, M.F. 1992a. A model of spatio-temporal information. *Proceedings, Fifth International Symposium on Spatial Data Handling,* Charleston, August 1992, pp. 602-611.

Worboys, M.F. 1992b. Object-oriented models of spatio-temporal information. *Proceedings, GIS/LIS.* November 1992, San Jose, 2. pp. 825-834.

Author

Christopher Gold

Christopher Gold holds an NSERC Industrial Research Chair in geomatics applied to forestry, co-funded by the Quebec Forest Industry Association (AIFQ), and is a member of the Centre for Geomatics Research (CRG) at Laval University, Quebec City. He is an Academician of the International Eurasian Academy of Sciences. He holds a BSc and MSc from McGill University in geology and agriculture respectively, and a PhD from The University of Alberta in geology and computer methods. He was an early worker with triangulation methods for terrain modeling, and his primary research interests are with dynamic topological spatial data structures, spatio-temporal information management and interactive spatial decision support systems. He is a strong promoter of greater collaboration between computer science and GIS.