Moving Objects Management in a 3D Dynamic Environment

L. Hashemi Beni[1], M. A. Mostafavi[2], M. L. Gavrilova[3]

[1]Department of Geomatics, Laval University, Quebec, QC, G1K 7P4, Canada
Telephone: 1(418) 656-2131 ex. 12677
Fax: 1 (418) 656-7411
Email: leila.hashemi.1@ulaval.ca

[2] Department of Geomatics, Laval University, Quebec, QC, G1K 7P4, Canada
Telephone: 1 (418) 656-2131 ex. 2750
Fax: 1 (418) 656-7411
Email: mir-abolfazl.mostafavi@scg.ulaval.ca

[3]Department of Computer Science, University of Calgary, 2500 University Dr. N.W., Calgary, AB, T2N1N4, Canada
Telephone: 1 (403) 220-5105
Fax: 1 (403) 284-4707
Email: marina@cpsc.ucalgary.ca

1. Introduction

For many applications, we need to represent and simulate dynamic objects or phenomena to better understand, analyze, predict and manage their behaviours. Some examples of this application are: simulation of fluid flows, meteorological and environmental applications, aircraft control, ship navigation, and inter-satellite communication, etc. Here, objects are embedded in 3D environment and are free to move and interact with each other over time and space. This implies that we often need to be able to track and make quarries on the state of objects and their neighbours in the past, present and future time. Then, it is essential to be able not only to define 3D topological relations between the moving objects in 3D environment but also to update these spatial relationships as the objects change their position in time. However, the current GIS data structures are unable to accomplish these tasks as most of them are static and limited to 2D space.

In order to overcome theses problems, we propose and implement a 3D kinetic topological data structure based on Delaunay tetrahedralization which has ability to deal with static and dynamic objects at the same time. In addition, the structure is capable of maintaining and updating the spatial relations between object even when they are moving. In term of objects, it should be noted in some applications there is an exact definition of objects such as aircrafts for aircraft control. While, in other applications such as simulation of a fluid flow, with objects we refer to discrete elements (particles) that represent continuous phenomena. The proposed data structure has the potential to manage the moving object in both cases. The topological relations between the moving objects are created and maintained by Delaunay tetrahedralization which is described in more details in the following section.

2. Delaunay Tetrahedralization

Delaunay Tetrahedral (DT) for a set of points in 3D space is defined by the partitioning of the space into tetrahedrons based on the empty circumsphere test (fig.1). It means the circumsphere of each tetrahedron does not contain any other point of the data set. Thus, among all the possible tetrahedralization of the set of points, it introduces a unique 3D mesh, except when there are degenerate cases in the set (ex. if five or more points are co-spherical in 3D space) (Gold et al. 2005).

There are several methods to construct Delaunay tetrahedralization. In static Delaunay data structure, the whole data set (points or objects) is known prior to the partitioning of the space according to the empty circumsphere test such as Sweep line method or Divide and conquer method. Here, the structure can not be updated locally and should be reconstructed globally if we need to inserting or deleting an objects from the data structure. While in a dynamic data structure, it is not necessary to know the whole data set prior to Delaunay tetrahedralization, and new objects (point) can be inserted or removed from the existing data structure. On the other words, local modification in the data structure is allowed after any change.

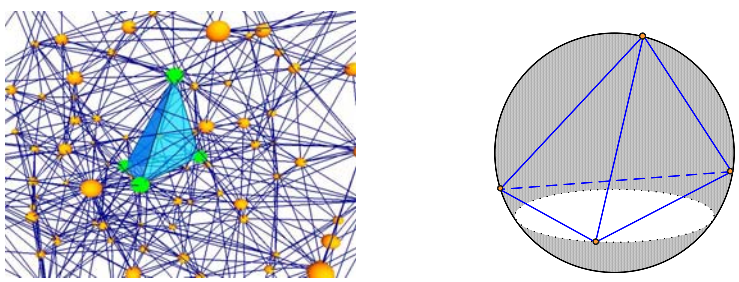

Figure 1. Delaunay Tetrahedralization; circumsphere of each tetrahedron does not contain any other point of the data set.

Kinetic Delaunay data structure allows several objects (points) to move simultaneously in the mesh with the capacity for local topology updating. To construct a kinetic Delaunay data structure, it is essential to use a dynamic Delaunay tetrahedralization method. Among the various methods that are studied in the field of computational geometry, the incremental method is the only dynamic method. Therefore, we use this method to develop a 3D kinetic data structure for the moving objects management in a 3D dynamic environment, as seen in the next section.

3. Kinetic Delaunay Tetrahedralization

In order to extend a dynamic DT to a kinetic DT, we first describe the moving of one data point (object) from its initial position toward a given destination within the 3D mesh and then the method is extended to several moving points in a 3D space.

3.1. Moving a point in a Delaunay Tetrahedralization

Point movement changes the configuration of the tetrahedrons containing the moving point (its stars) and its neighbours. Continuous movement of a point in the 3D mesh can be implemented by inserting, deleting and re-inserting the point in its new position within mesh that is a computationally expensive operation. However, if the location of the point is changed without any topological changes, the spatial relationship dose not need to be updated. It means: "*no topological event happens*" Therefore, to present the continuous movement of a point, it is possible to detect all topological events on the trajectory of the point and move the point to its new positions, one by one on its trajectory, until the point arrives at its destination. In a 3D DT, a topological event occurs when a point moves in or out of the circumsphere of tetrahedron. This must be detected to preserve the empty circumsphere criterion and then the topological modification must be done locally (fig.2).

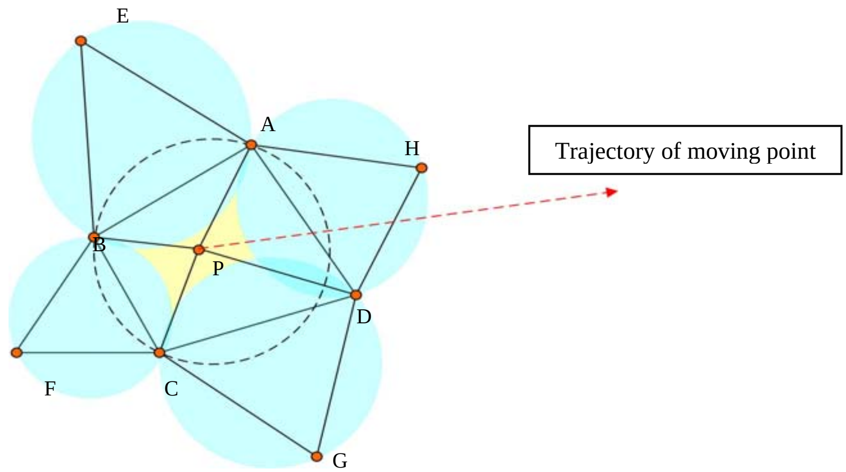

Figure 2. Topological event; spatial relationship dose not need to be updated as long as the vertex p is moved within the yellow polyhedron in 3D (from top).

The proposed algorithm for the moving points is a generalization to 3D of Mostafavi and Gold's method in 2D (2004). In this algorithm, to detect the topological events when one point moves in the mesh, only the spatial information of its neighbouring tetrahedrons are used and the rest of tetrahedra of mesh do not need to be tested (Albers et al. 1998, Gavrilova and Rokne 2003a, Gavrilova and Rokne 2003b, Guibas and Russel 2004, Roos 1997). In order to consider all possible topological events on the trajectory of the moving point, we need to define adjacent imaginary and real tetrahedra in the Delaunay tetrahedralization with respect to moving point. In figure 2, ΔAEB, ΔBFC, ΔCGD and ΔDHA are examples of real triangles in 2D and ΔABC is an example of an imaginary triangle that would exist if p was moved out of its imaginary circumcircle.

To move a point from its actual position towards a given destination, it is essential to move the point step by step to the closest topological event and make appropriate local updates. The closest topological event ($t_{\min}$) is the location where intersection between the circumsphere of the real or imaginary tetrahedra with the trajectory is closest. For this purpose, a simple test that computes the intersection between a line and a sphere is used. The geometric object moves by replacing each of its coordinates with a function of time:

$$p_{new}(t) = p_{old} + t \cdot D \qquad (1)$$

Where D is the distance of moving point p to its destination and $t \cdot D$ is the closet topological event distance. Substituting the equation (1) into the equation of a sphere gives a quadratic equation. Regarding this equation, the three possible line-sphere intersections are:

- No intersection,
- The sphere is tangent to the line (Point intersection),
- The line intersects the sphere (Two intersection points).

In addition, since we use the relative distance, the intersection will be t=0, 0<t<1 and t>1, if moving point is at the origin, between the origin and the destination or after its destination, respectively. We ignore t<0 that means intersection occurs before the origin.

In practice, we move the point to the new position by deleting it from its current position, re-insert it in the new position (closest topological event) and make appropriate local updates. To do this, we need some dynamic operations in 3D space. The most important operations are:

- *Point location (Walk):* The task of the point location operator is finding the tetrahedron σ within an existing 3D mesh that contains the query point p.

- *Insert:* this operation inserts a new point in the existing structure. The *Insert* operation calls *point location operator* to find the tetrahedron containing p (σ) and splits σ into four tetrahedrons, each having p as a vertex.

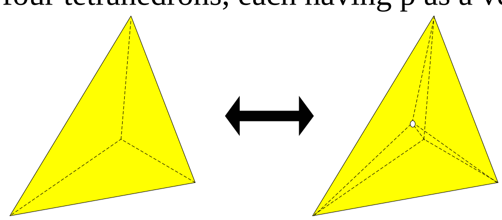

Figure 3. Flip 14, the tetrahedron that contains the new added point is divided to four new tetrahedrons by connecting the new point to the vertices of the tetrahedron.

Optimization (Flip): an optimization operator is a local topological operation that modifies the configuration of adjacent tetrahedrons to satisfy the Delaunay criterion (i.e. empty circumsphere test). This condition is verified via a determinant computation (equation 2), this determinant test if point p is inside, outside or lies on circumsphere of a tetrahedron (a, b, c and d are tetrahedron vertices). When the determinant result is negative, it means that the tetrahedron is not Delaunay and a flip (optimization) operation must be performed.

$$\begin{vmatrix} x_p & y_p & z_p & x_p^2+y_p^2+z_p^2 & 1 \\ x_a & y_a & z_a & x_a^2+y_a^2+z_a^2 & 1 \\ x_b & y_b & z_b & x_b^2+y_b^2+z_b^2 & 1 \\ x_c & y_c & z_c & x_c^2+y_c^2+z_c^2 & 1 \\ x_d & y_d & z_d & x_d^2+y_d^2+z_d^2 & 1 \end{vmatrix} \qquad (2)$$

For example in some cases to optimize the new tetrahedron, two neighbour tetrahedrons convert to three tetrahedrons (flip 23) with respect to the Delaunay criterion or vice versa (flip 32)(fig.4).

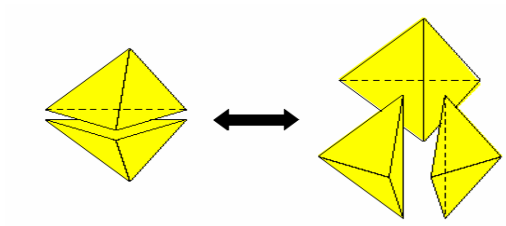

Figure 4. Optimizing the new triangles with respect to the Delaunay criterion by flip 23 and flip 32

- *Delete:* This operation removes a point from an existing mesh. Deletion consists of locating the point p to be deleted and restructuring the configuration of the tetrahedral incident to p with a sequence of flips; for example, when a point is removed the adjacent tetrahedrons should be merged together.

In the proposed algorithm, the geometrical and topological information are stored based on 3D Tetrahedron data structure, which included every tetrahedron by its four pointers for its points (objects) and four pointers to its four adjacent tetrahedral.

3.2. Moving several points in a Delaunay Tetrahedralization

In order to manage the motion of several points in 3D space, we need to compute the topological events of all moving points at the same time and then process them in order. Hence we compute the time taken to reach their closest topological event (t_t) for each point that is defined: $t_t = \frac{d}{v}$, where d is the distance to the closest topological event and v is velocity of point. Furthermore, for each point, a global time (t_g) needs to be kept that is the closest topological time of each point since the start of the simulation ($t_g = 0$):

$$t_g^i = t_t^i + t_c^i \qquad (3)$$

Where t_c^i is the time that it takes for each point to move for its origin to its current position.

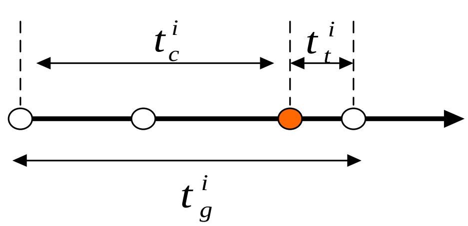

Figure 5. The three types of time needed for moving several points at the same time (t_g, t_c, t_t).

We put the global times in a priority queue. The point with smallest global time is processed first i.e. point is moved to its new location and necessary updates in the topological relationships are made. Following any motion, the priority queue is also updated. This process is reiterated until all points are moved to their respective destinations.

There are some degenerate cases that can arise during the movement of one or several points at time such as sphere tangential to the trajectory of moving point, co-spherical points, etc. in addition, collision cases must be managed. In our research work, we deal with these complex cases.

4. Conclusion

In this paper the development and the implementation of a 3D kinetic topological data structure based on Delaunay tetrahedralization was discussed. The proposed data structure allows the establishment and analysis of the topological relations between the objects and their updates. The structure provides the necessary operations to easily track the moving objects in 3D environment and make necessary queries on their location and configuration at any time. Hence, the proposed data structure has many significant potential for wide variety applications that need to represent and manage moving objects.

5. Acknowledgement

The authors would like to acknowledge the support of the GEOIDE Network under GeoTopo3D project. Also, they would like to thank Dr Christopher Gold from University of Glamorgen, and Dr Hugo Ledoux from Delft University, for their valuable contributions to this work.

6. References

Albers G, Guibas LJ, Mitchell JSB, and Roos T, 1998, Voronoi diagrams of moving points. *International Journal of Computational Geometry and Applications*, 8:365–380.

Edelsbrunner H, 2001, Geometry and Topology for Mesh Generation. Cambridge University Press, Cambridge, 190 pp.

Devillers O (2002b). On deletion in Delaunay triangulations. *International Journal of Computational Geometry and Applications*, 12(3):193–205.

Ledoux H, 2006, Modelling Three-dimensional Fields in Geosciences with the Voronoi Diagram and its Dual. *PHD thesis*, University of Glamorgan.

Gavrilova M and Rokne J, 2003a, Collision detection optimization in a multi-particle system. *International Journal of Computational Geometry and Applications*, 13(4): 279-302.

Gavrilova ML and Rokne J, 2003b. Updating the topology of the dynamic Voronoi diagram for spheres in Euclidean d-dimensional space. *Computer Aided Geometric Design*, 20:231–242. Gold CM, Charters TD, and Ramsden J, 1977, Automated contour mapping using triangular element data structures and an interpolant over each triangular domain. In George J, editor, *Proceedings Siggraph '77*, volume 11(2) of Computer Graphics, pages 170–175.

Gold CM, Ledoux H and Dzieszko M, 2005, A data structure for the construction and navigation of 3D Voronoi and Delaunay cell complexes. In Proceedings *13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pages 21-22, Plzen, Czech Republic.

Guibas LJ and Russel D, 2004, An empirical comparison of techniques for updating Delaunay triangulations. *Symposium on Computational Geometry*, 170-179.

Mostafavi MA, Gold CM, and Dakowicz M, 2003, Delete and insert operations in Voronoi/Delaunay methods and applications. *Journal of Computers & Geosciences*, 29(4):523–530.

Mostafavi MA and Gold CM, 2004, A global kinetic spatial data structure for a marine simulation. *International Journal of Geographical Information Science*, 18(3):211–228.

Roos T, 1997, New upper bounds on Voronoi diagrams of moving points, *Nordic Journal of Computing*, 4:167-171.