THREE-DIMENSIONAL EXTENSION OF KIRKPATRICK'S PLANAR POINT LOCATION METHOD

ABDULLAH ZAWAWI TALIB
School of Computer Sciences
University of Science Malaysia
11800 USM Penang, Malaysia

AHMAD IZANI MD. ISMAIL
School of Mathematical Sciences
University of Science Malaysia
11800 USM Penang, Malaysia

ABSTRACT

The point location problem is an important problem in computational geometry. Most of the methods that have been developed have been for the planar point location problem. By contrast, the spatial point location problem has received less attention. In this paper we discuss the extension of an efficient implementation of an extensively used planar point location method to three dimensions.

KEY WORDS

Point location problems, computational geometry, geometric algorithms.

1. INTRODUCTION

The planar (or 2D) point location problem (de Berg et al [1]) can be stated as follows*: Given a subdivision S of the plane and a query point Z, determine the region R of S which contains Z.* Various methods for the point location problem have been developed. A method which is extensively used is Kirkpatrick's method [2]. This method creates and employs an efficient search structure which consists of a suitable small set of new triangles organized in a subdivision hierarchy called the $\mathcal{K}$-structure. An efficient implementation of Kirkpatrick's planar point location method has been developed by Talib et al [3]. The spatial (or 3D) point location problem has not received as much attention as the planar problem. The earlier methods for the spatial point location problem are basically direct extensions of their two-dimensional counterparts (Dobkin and Lipton [4], Cole [5]). However, some of the reported results present a different approach such as using *space-sweep/plane-sweep* techniques. Preparata and Tamassia ([6],[7]), and Tan et al [8] are two examples of the methods that employ this approach. Another spatial method is the Meshed Polyhedra Point Location method (MPPL) which uses the data structure for the Meshed Polyhedra Visibility Ordering (MPVO) algorithm (Williams [9]).

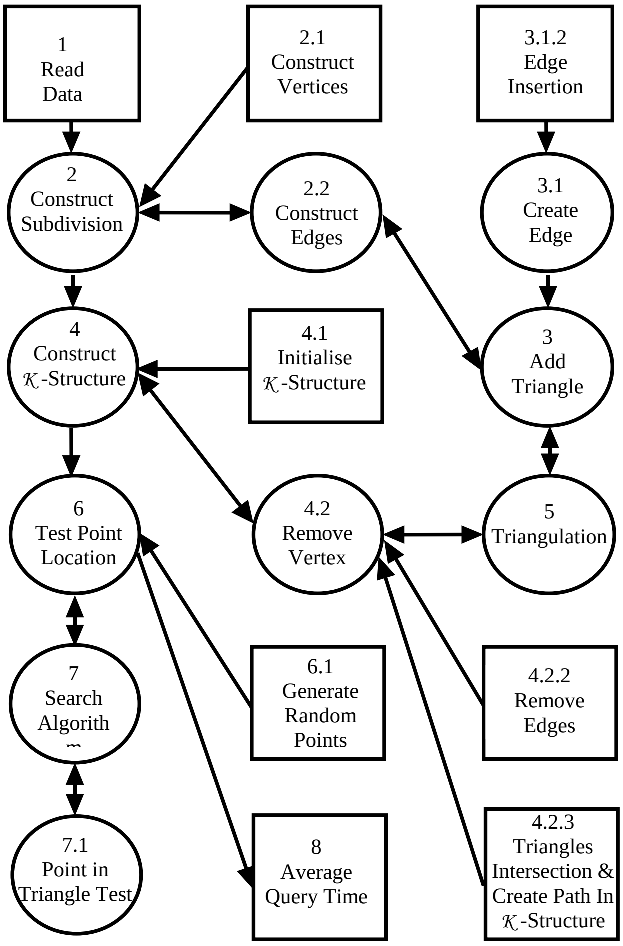
In this paper, the possibility of extending Kirkpatrick's planar point location method to locating a point in spatial subdivisions based on the corresponding planar method is investigated. It is anticipated that the same organisation for the $\mathcal{K}$-structure can be used for spatial point location search. However, the analogy of triangulation (*tetrahedralisation*) in 3D is not a straightforward process in that the number of tetrahedrons which is to appear is unknown a priori (Yvinec [10]). Furthermore, the number of vertices, edges, faces and regions in 3D are not necessarily linearly related (Yvinec [10], Croom [11]).

Figure 1 represents the overall structure of a program for an efficient implementation of Kirkpatrick's method developed by Talib et al [3]). Referring to Figure 1, the modifications required to fulfil a 3D implementation of the Kirkpatrick's method are as follows.

(i) Representation of the three-dimensional subdivision and the algorithm to construct such a structure.
(ii) Representation of the $\mathcal{K}$-structure for three-dimensional tetrahedral elements, $\mathcal{K}^3$-structure and the algorithms to construct such a structure including the 3D hierarchical tetrahedralisation process.
(iii) Algorithms for tetrahedralisation (instead of triangulation) and the intersection of tetrahedral elements (instead of triangles).
(iv) A new point location search algorithm for 3D.

2. ALGORITHMS AND DATA STRUCTURES FOR THE 3D SUBDIVISION

An efficient representation of 3D subdivisions $\mathcal{S}^3$ relies on the nature of the tetrahedralisation algorithm and the updates of $\mathcal{S}^3$ during the 3D hierarchical triangulation process. The 3D hierarchical triangulation process is identical to its 2D counterpart except that a tetrahedralisation algorithm is used to retriangulate the resulting polyhedron after the removal of a vertex and its associated edges and faces.



Circles - take in input and return output
Squares - either take in output or return input

FIGURE 1 THE OVERALL STRUCTURE OF THE PROGRAM FOR THE KIRKPATRICK'S PLANAR POINT LOCATION METHOD

Triangulation in 3D space or *tetrahedralisation* is much more complicated due to the fact that the number of tetrahedrons which is to appear is unknown a priori. In the 3D extension of Kirkpatrick's method, an algorithm to triangulate a nonconvex polyhedron is required. However, there exists an indirect approach to tetrahedralisation as described in Zienkiewicz and Taylor [12]. In this method, the polyhedron is sectioned into a number of quadrilaterals called *bricks* (eight-cornered elements). Each of these bricks can then be divided into five tetrahedra in two (and only two) distinct ways. Another alternative is to subdivide the bricks into six tetrahedra but this time several variations are available.

Without going into the details of the tetrahedralisation algorithm and the 3D hierarchical triangulation process, the following data structure for $\mathcal{S}^3$ is proposed for the purpose of extending the method to three dimensions. The most appropriate orientation of the data structure is that of the *edge-based* orientation as in the planar representation (Talib et al [3]). The proposed data structure is formulated based on the modified *edge-ordered representation* (Talib et al [3]). The representation is as for the planar representation (Talib et al [3]) with the following modifications:

(i) Each directed edge (v,w) has a list of faces that touch it.
(ii) Each face is associated with the names of the region lying immediately to the right and the left of it as opposed to the name of the region to the right of an edge in the planar representation.

The appropriate region i.e. left or right must be chosen based on the orientation of the face during the hierarchical triangulation process.

The declaration of the basic components of the structure are as shown in Table 1. The highlighted entries indicate the additional information proposed for the 3D implementation from the implementation of its 2D-counterpart (in Talib et al [3]). An example of the representation is shown in Figure 2. In this representation all duplicate instantiations of a face can be made to point to a single face structure.

TABLE 1: DECLARATION OF DATA STRUCTURES FOR THE 3D SUBDIVISION FOR THE IMPLEMENTATION OF THE 3D KIRKPATRICK'S METHOD

	Declaration	Description

Edge	**struct** edge {
	struct face *faces;
	struct edge *recedge;
	struct vnode *vertex ;
	struct edge *next;
	};
		Pointer to the list of faces.
		Pointer the reciprocal edge.
		Pointer back to the vertex.
		Pointer to the next edge.

Vertex	**struct** vnode {
	int nodenum;
	double x,y;
	int mark;
	int cornernode;
	struct vnode *predv;
	struct edge *edgelist;
	};
		Vertex identifier.
		Vertex co-ordinate.
		Marker for the vertex so that a set of independent nodes is removed at each stage.
		Marker for corner node but not the boundary node as nodes on the boundary edges are allowed - can be removed.
		Previous vertex in the list.
		Edge list.

Face	**struct** face {
	int rightregion;
	int leftregion;
	struct face *nextface;
	};
		The name of the region to the right.
		The name of the region to the left.
		Pointer to the next face in the list.

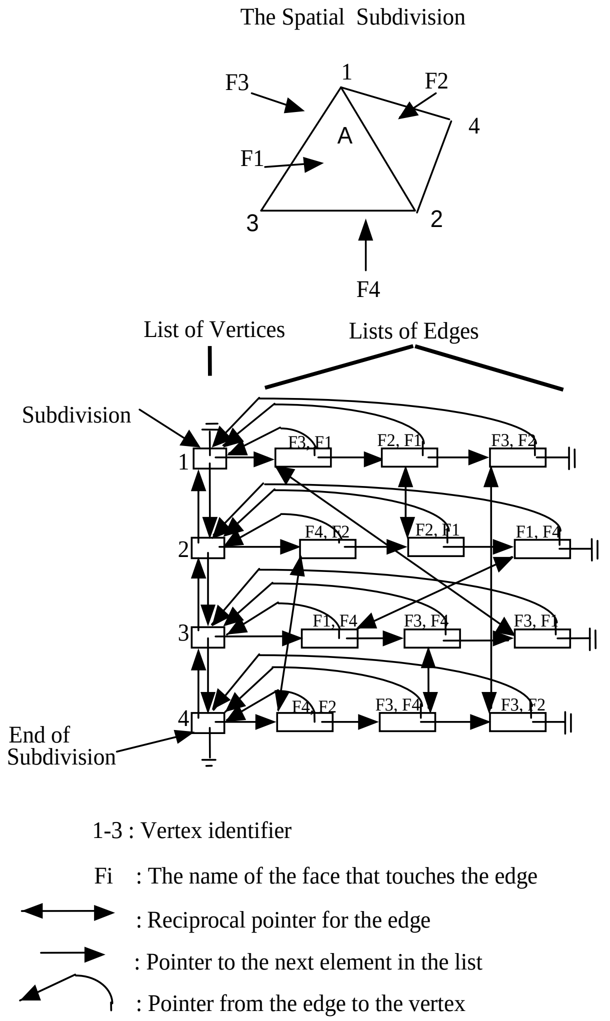


FIGURE 2: THE DATA STRUCTURE FOR THE 3D SUBDIVISION FOR THE 3D KIRKPATRICK'S METHOD

The data structure for $\mathcal{S}^3$ can be constructed in the same manner as in the planar implementation (Talib et al [3]). For the list of edges for each vertex, the correct insertion procedure into the respective edge list depends on the nature of the tetrahedralisation algorithm.

3. ALGORITHMS AND DATA STRUCTURES FOR THE 3D $\mathcal{K}$-STRUCTURE

As in the $\mathcal{K}$-structure for the planar method, the $\mathcal{K}$-structure for spatial point location search ($\mathcal{K}^3$-structure) is constructed based on the corresponding 3D hierarchical triangulation process which is performed on $\mathcal{S}^3$. Efficient data structures for both $\mathcal{S}^3$ and the $\mathcal{K}^3$-structure are needed to ensure an efficient implementation of the algorithm to construct the $\mathcal{K}^3$-structure and the point location search algorithm. Also, the construction of the $\mathcal{K}^3$-structure depends on the algorithms to determine the intersection of the tetrahedral elements and the data structure for the $\mathcal{K}^3$-structure. As mentioned earlier, the data structure for the 2D $\mathcal{K}$-structure can be modified easily to allow spatial point location searches. The only modification required is the inclusion of a fourth co-ordinate for the triangle node to convert it into a tetrahedral node.

The $\mathcal{K}^3$-structure can therefore be constructed in the same way as in the 2D implementation. In this case a path is created from a newly created tetrahedron to the tetrahedron which has been removed from the subdivision if the former intersects (except on the edges or vertices or faces) the latter tetrahedron. The calculation of *intersection of tetrahedral elements* is also modified easily from the calculation of triangle intersection by extending the latter to include the edges from the fourth vertices of the tetrahedron. The *point location search algorithm* for $\mathcal{S}^3$ is identical to its planar counterpart as described in Talib et al [3]. The only modification required is to replace the inclusion in a triangle test with the *inclusion in a tetrahedron* test.

4. CONCLUSIONS

It has been shown that the extension of the Kirkpatrick's method to spatial subdivisions is in some ways very straightforward. The only problem appears to be the need for an efficient tetrahedralisation algorithm for non-convex polyhedron. The rest of the method only needs minor modifications. This is especially true for the data structures for $\mathcal{K}^3$-structure and the point location search algorithm for this structure. A suitable data structure for $\mathcal{S}^3$ is also established. Nonetheless, the final structure is subject to the nature of the tetrahedralisation algorithm for the nonconvex polyhedron. Yet to be proven is its simplicity and practicality as no implementation is carried out. However, intuitively the notion is true as the proposed extension is based on its planar counterpart.

REFERENCES

[1] M. de Berg, O Schwarzkopf, M. van Icreveld, M. Overmass, *Computational geometry: Algorithms and applications (Springer-Verlag, 2000).*

[2] D. G. Kirkpatrick, Optimal Search in Planar Subdivisions, *SIAM Journal on Computing,* 12(1), 1983, 28-35.

[3] A. Z. Talib, A. I. Ismail, M. Chen, Efficient implementation of a planar point location method, *Proc. 2nd Int. Conf. on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*, ACM SIGGRAPH Publication, 2004, 155-161.

[4] D. P. Dobkin and R. J. Lipton, Multidimensional searching problems, *SIAM Journal on Computing*, 5(2), 1976, 181-186.

[5] R. Cole, Searching and storing similar lists", *Journal of Algorithms*, 7(2), 1986, 202-220.

[6] F. P. Preparata and R. Tamassia, Efficient spatial location, An extended abstract, *Lecture Notes In Computer Science*, Vol. 382, Springer Verlag 1989, 3-11.

[7] F. P. Preparata and R. Tamassia, Efficient point location in a convex spatial cell-complex", *SIAM Journal on Computing*, 21(2), 1992, 267-280.

[8] X.-H. Tan, T. Hirata and Y. Inagaki, Spatial Point Location and Its Applications, *Lecture notes in Computer Science*, Vol. 450, Springer-Verlag, 1990, 241-250.

[9] P. C. Williams, Visibility ordering meshed polyhedra, ACM Transaction on Computer Graphics, 11(2), 1992, 103-126.

[10] M. Yvinec, *Triangulation in 2D and 3D spaces, Lecture Notes in Computer Science*, Vol. 391, Springer Verlag, 1989, 274-291.

[11] F. H. Croom, *Basic Concepts of Algebraic Topology* (Springer-Verlag, 1978).

[12] O.C. Zienkiewizc and Taylor, *The Finite Element Methods* (McGraw-Hill, 2000).