

# A Pedagogic JavaScript Program for Point Location Strategies \*

Pedro Machado Manhães de Castro  
KAUST  
Thuwal, Kingdom of Saudi Arabia.  
Pedro.deCastro@kaust.edu.sa

Olivier Devillers  
INRIA Sophia Antipolis Méditerranée  
Sophia-Antipolis, France  
Olivier.Devillers@inria.fr

## ABSTRACT

Point location in triangulations is a classical problem in computational geometry. And walking in a triangulation is often used as the starting point for several nice point location strategies. We present a pedagogic JavaScript program demonstrating some of these strategies, which is available at: [www-sop.inria.fr/geometrica/demo/point\\_location\\_strategies/](http://www-sop.inria.fr/geometrica/demo/point_location_strategies/).

## Categories and Subject Descriptors

I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—*Geometric algorithms, languages, and systems*

## 1. INTRODUCTION

Given a query point  $q$  and a partition of the  $d$ -dimensional space in regions, the problem is to retrieve the simplex (or triangle, in two dimensions) containing  $q$ . This problem is called *point location in triangulations*, and is a classical problem in computational geometry; not surprisingly, the literature is vast [9, 3, 5, 8, 4, 1].

Listing all the classes of point location strategies produced by the scientific community up to now is way beyond the scope of this work. Instead, we concentrate our efforts on some point location strategies based on *walking* [6]. Walking in a triangulation is the basis for several efficient and practical point location strategies [7, 5, 2]. Figure 1 shows a screenshot of the Delaunay hierarchy view.

This short paper is organized as follow: Section 2 describes in few words the walking procedure; Section 3 quickly describes some successful strategies based on walking, which are implemented in the program; and finally, Section 4 serves as a reference on the program's functionalities.

\*This work is supported by ANR Project *Triangles* ANR-07-BLAN-0319 and Région PACA; also part of this work was done when P. M. M. de Castro was at INRIA Sophia Antipolis Méditerranée.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SCG'11, June 13–15, 2011, Paris, France.

Copyright 2011 ACM 978-1-4503-0682-9/11/06 ...\$10.00.

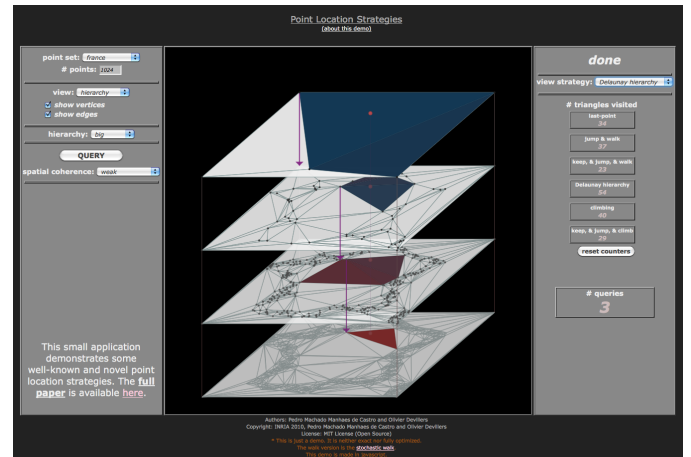


Figure 1: Screenshot.

## 2. WALKING IN A TRIANGULATION

In two dimensions, the *visibility graph*  $\mathcal{VG}(\mathcal{T}, q)$  of a triangulation  $\mathcal{T}$  and a query point  $q$  is a directed graph  $(V, E)$ , where  $V$  is the set of triangles of  $\mathcal{T}$ , and a pair of triangles  $(\tau_i, \tau_j)$  belongs to the set of arcs  $E$  if  $\tau_i$  and  $\tau_j$  are adjacent in  $\mathcal{T}$  and the supporting line of their common edge *separates* the interior of  $\tau_i$  from  $q$ ; see Figure 2. When two triangles  $\tau_i$  and  $\tau_j$  are such that  $(\tau_i, \tau_j) \in E$ , we say that  $\tau_j$  is a *successor* of  $\tau_i$ . Now, a *visibility walk* consists in repeatedly walking from a triangle  $\tau_i$  to one of its successor in  $\mathcal{VG}$  until the triangle containing  $q$  is found; a *walking strategy* describes how this successor is chosen.

Two strategies are standard: the straight walk, and the stochastic walk. The *straight walk* is a visibility walk where each visited triangle intersects some segment having  $q$  as end-point; and (ii) the *stochastic walk* is a visibility walk where the next visited triangle, from a triangle  $\tau$ , is randomly chosen amongst the successors of  $\tau$  in  $\mathcal{VG}$ . The straight walk is easier to analyze, whereas the stochastic walk is easier to implement, and faster in practice [6]. We choose the stochastic walk for the program presented in this work.

## 3. STRATEGIES BASED ON WALKING

- **Last-point** [6]. The last-point strategy starts from the triangle containing the last query point and walk

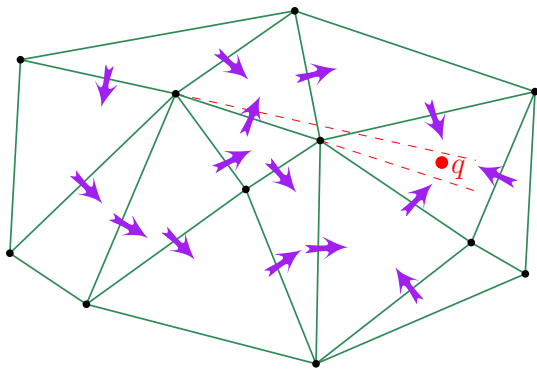


Figure 2: Graph  $\mathcal{VG}$ . Arrows represent arcs of  $\mathcal{VG}$ .

until the triangle containing the current query point is reached.

- **Jump and Walk [7].** A certain number of landmarks are selected randomly from the vertices of the triangulation. For each query, we have a first jump step, which consists in selecting the closest landmark to the query point as starting point of a walk. And finally we have a walk step from the chosen landmark to the query point.
- **Keep, Jump, and Walk [2].** A slight modification in Jump and Walk leads to the Keep, Jump, and Walk strategy. Instead of taking as landmarks random vertices of the triangulation, the last query points are taken.
- **Delaunay Hierarchy [5].** The Delaunay Hierarchy maintain a hierarchy of samples of the vertices. The ratio of the sizes of samples at two consecutive levels is some constant. To find a query point, we walk in each level's triangulation, beginning the procedure at the coarser one; the walk starts at the location found at the level right above.
- **Climbing [2].** The Delaunay hierarchy can be used in another way. We start the walk at the last query point in the lowest level, if the walk encounters a vertex present at an higher level, then the walk is pursued at this level. When a triangle containing the query is found at some level, the hierarchy is descended as describe above to finally locate the query at the lowest level.
- **Keep, Jump, and Climb [2].** To improve further the climbing procedure, a certain number of landmarks in the lowest level of the hierarchy are chosen. These landmarks have the same use as in keep, jump, and walk, i.e., the closest landmark to the query point is chosen in order to start the climbing procedure.

#### 4. SOFTWARE FUNCTIONALITIES

- **Input.** Building the Delaunay triangulation of a set of points; the input set can be chosen amongst several random, realistic, and synthetic point sets.

- **Queries.** Generating a custom or random query point. Random query points are generated by pressing the QUERY button, and custom query points are generated by simply mouse clicking inside the triangulation domain.
- **Spatial Coherence.** Choosing the spatial coherence of random query points; The options are weak, normal, and strong; selecting one of them can be done by choosing the appropriate option in the SPATIAL COHERENCE select box.
- **Strategies.** Visualizing six different strategies working in order to retrieve the current query point. Choosing and switching between strategies can be done by selecting the appropriate option in the VIEW STRATEGY select box. All the strategies considered in Section 3 can be selected for visualization.
- **Size of Hierarchy.** Choosing the size of the hierarchy. The options are normal, big, and huge size; selecting one of them can be done by choosing the appropriate option in the HIERARCHY select box.
- **Counters.** Counting the number of query, and the number of triangles visited per strategy.
- **View.** Choosing between the standard 2D view (*normal*), or the hierarchical view (*hierarchy*); the later is more appropriate for the strategies that are based on hierarchy, while the former is more appropriate for the strategies that are not based on hierarchy.
- **Vertices and Edges.** Viewing/Hiding vertices and edges can be done by choosing the appropriate options in the SHOW VERTICES and SHOW EDGES check boxes.

#### 5. REFERENCES

- [1] S. Arya, T. Malamatos, and D. M. Mount. A simple entropy-based algorithm for planar point location. *ACM Trans. on Algo.*, 3(2):17, 2007.
- [2] P. M. M. de Castro and O. Devillers. Simple and Efficient Distribution-Sensitive Point Location, in Triangulations. In *ALLENEX*, 127–138, 2011.
- [3] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.
- [4] E. D. Demaine, J. Iacono, and S. Langerman. Proximate point searching. *Comput. Geom. Theory and Appl.*, 28(1):29–40, 2004.
- [5] O. Devillers. The Delaunay hierarchy. *Intern. J. Found. of Comput. Sci.*, 13:163–180, 2002.
- [6] O. Devillers, S. Pion, and M. Teillaud. Walking in a triangulation. *Intern. J. of Found. of Comput. Sci.*, 13:181–199, 2002.
- [7] L. Devroye, E. P. Mücke, and B. Zhu. A note on point location in Delaunay triangulations of random points. *Algorithmica*, 22:477–482, 1998.
- [8] J. E. Goodman and J. O'Rourke, editors. *Handbook of Discrete and Computational Geometry*. CRC Press LLC, Boca Raton, FL, 2004. 2nd edition.
- [9] D. G. Kirkpatrick. Optimal search in planar subdivisions. *SIAM J. on Computing*, 12(1):28–35, 1983.