

# Dynamic location in an arrangement of line segments in the plane

Olivier Devillers, Monique Teillaud, Mariette Yvinec

► **To cite this version:**

Olivier Devillers, Monique Teillaud, Mariette Yvinec. Dynamic location in an arrangement of line segments in the plane. [Research Report] RR-1558, INRIA. 1991. inria-00075003

**HAL Id: inria-00075003**

**<https://hal.inria.fr/inria-00075003>**

Submitted on 24 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Dynamic location  
in an arrangement of line segments in the plane\*  
Localisation dynamique  
dans un arrangement de segments du plan

Olivier  
Devillers<sup>†</sup>

Monique  
Teillaud<sup>†</sup>

Mariette  
Yvinec<sup>‡</sup>

*Programme 4* : Robotique, Image et Vision.

**Key words** : arrangements, trapezoidal maps, dynamic algorithms, randomized algorithms.

**Rapport de recherche INRIA 1558, Novembre 1991.**  
paru dans *Algorithms Review* **2(3):89-103**.

---

<sup>†</sup>INRIA, 2004 Route des Lucioles, B.P.109, 06561 Valbonne cedex (France), Phone : +33 93 65 77 62, E-mail : odevil or teillaud@alcor.inria.fr. Part of this work was done while these authors were visiting Max Planck Institut

<sup>‡</sup>LIENS, CNRS URA 1327, 45 rue d'Ulm, 75230 Paris cedex 05

\*This work has been supported in part by the ESPRIT Basic Research Action Nr. 3075 (ALCOM).

## Abstract

We describe in this paper an algorithm which dynamically maintains the trapezoidal map of an arrangement of line segments. This algorithm is a generalization of the Influence Graph data structure used in [BDS\*] to deal with the semi-dynamic case.

Our complexity results are randomized, i.e. the insertion sequences are supposed to be evenly distributed among the  $n!$  possible sequences of  $n$  segments, and when an object is deleted, we suppose that it can be any of the already inserted objects with the same probability. Under these assumptions the algorithm needs  $O(n + a)$  expected space,  $O(\log n + \frac{a}{n})$  expected insertion time,  $O((1 + \frac{a}{n}) \log \log n)$  expected deletion time and  $O(\log n)$  time to locate any query point in the arrangement, where  $a$  is the current size of the arrangement and  $n$  the current number of segments.

## Résumé

Nous décrivons dans cet article un algorithme permettant de maintenir dynamiquement la carte des trapèzes d'un arrangement de segments. Cet algorithme est une généralisation du graphe d'influence développé pour traiter le cas semi-dynamique dans [BDS\*].

Les résultats de complexité sont randomisés, c'est à dire que l'ordre d'insertion des segments est une permutation quelconque, parmi les  $n!$  permutations possibles d'un ensemble de  $n$  segments, de manière équiprobable. Quand un segment est supprimé, on suppose que c'est n'importe lequel des segments, également de manière équiprobable. Sous ces hypothèses, l'algorithme utilise un espace mémoire moyen  $O(n + a)$ , un temps d'insertion moyen  $O(\log n + \frac{a}{n})$ , un temps de suppression moyen  $O((1 + \frac{a}{n}) \log \log n)$  et un temps de localisation  $O(\log n)$  pour un point quelconque du plan, où  $a$  est la taille courante de l'arrangement et  $n$  le nombre courant de segments.

# 1 Introduction

One of the fundamental topics in computational geometry deals with planar subdivisions. The most common question is, for a planar subdivision of size  $n$ , to answer queries of the following type : *which is the region containing a given point ?* This problem received a lot of attention in the literature, and has been solved in different ways (see [Pre90] for a survey), in optimal  $O(\log n)$  time and  $O(n)$  space after an  $O(n \log n)$  preprocessing.

If the aim is not only the computation of a location structure to answers queries, but also to be able to modify dynamically the planar subdivision, the deterministic solutions known at this time involve complicated algorithms, and none of them is optimal (the query time [CJ90] or the update time [CT91] is  $O(\log^2 n)$ ). A very new result for deterministic solutions to this problem is due to H. Baumgarten *et al.* [BJM92] and reach query and insertion time  $O(\log n \log \log n)$  and deletion time  $O(\log^2 n)$  with linear space.

After the work of Clarkson and Shor [CS89] studying this problem in a randomized framework, a lot of algorithms based on this principle have been developed. The principal interest of randomized algorithms is their simplicity and their good complexity and the counterpart is that these complexities are only randomized. Namely, for a dynamic algorithm, the update sequence must verify some hypotheses detailed in the sequel.

Concerning the problem of planar location, randomization was first applied to solve the static problem by Clarkson and Shor [CS89] using the Conflict Graph technique with optimal expected complexity :  $O(\log n)$  query time,  $O(a + n \log n)$  preprocessing time and  $O(a + n)$  space to treat an arrangement of  $n$  segments with  $a$  intersecting points. [BDS\*] developed a new structure, *the Influence Graph*, allowing semi-dynamic algorithms with the same complexity (the whole set of segments has not to be known in advance, but it still must verify the randomized hypothesis). In the special case where the line segments form a known connected planar graph (so  $a = n$ ) the preprocessing can be improved to  $O(n \log^* n)$  [Sei91].

Recently, some results appeared concerning fully dynamic randomized algorithms. The first paper on this topic [DMT90] generalized the Influence Graph structure to be able to remove a site from a Voronoï diagram.

Some results followed : [CMS92,Sch91] use basically the same idea as [DMT90], while [MS91,Mul91] use a completely different approach which is not based on the order of insertion of the points. [Sch91] reaches a complexity of  $O(\log^2 n)$  expected update and query time for a set of non intersecting segments. The algorithm in [Mul91] has  $O(\log n)$  query time and  $O(\log n + \frac{a}{n})$  insertion time and  $O(1 + \frac{a}{n})$  deletion time with high probability, where  $a$  is the number of intersections between segments.

In this paper we extend the result of [BDS\*] to generalize the Influence Graph for the fully dynamic construction of an arrangement of line segments, in the same way as for the Delaunay triangulation in [DMT90]. A line segment is added in  $O(\log n + \frac{a}{n})$  expected time and deleted in  $O((1 + \frac{a}{n}) \log \log n)$  expected time ; the algorithm uses  $O(n + a)$  expected space and a query point is located in  $O(\log n)$  expected time. The