Int. J. Open Problems Compt. Math., Vol. 7, No. 3, September, 2014
ISSN 1998-6262; Copyright @ICSRS Publication, 2014
www.i-csrs.org

An Algorithm for Computing the Convex Hull of a Set of Imprecise Line Segment Intersection

Keivan Borna[1], Morteza Asadi

Faculty of Mathematics and Computer Science, Kharazmi University, Tehran, Iran
e-mail: borna@khu.ac.ir

Faculty of Electrical and Computer Engineering, Islamic Azad University, Qazvin, Iran
e-mail: info@asadiweb.ir

Abstract

Data imprecision constitutes an important gap between theory and practice in computational geometry. A lot of research about imprecision in computational geometry is directed at computing the convex hull of imprecise points rather than imprecise line segment intersection. In this paper we introduce an algorithm to construct the convex hull for a set of n imprecise line segment intersection in $O(nlogn)$ time.

Keywords: *computational geometry, convex hull, imprecise line, imprecision, intersection.*

2000 Mathematics Subject Classification: *68U05, 68T37.*

1 Introduction

The two most basic elements in the Euclidean geometry are points and lines. Data imprecision constitutes an important gap between theory and practice in computational geometry. Many authors including [1, 4] studied imprecise data. Concentrating on convex hull, in computational geometry, much emphasis is

---

[1] Corresponding author

based on the computation of the convex hull for imprecise points than convex hull for a set of imprecise line segment intersection. According to the survey conducted by many authors, the problem of finding convex hull for imprecise lines has not been studied until now and this represents the freshness of this topic. This paper is dedicated to this study and an optimal algorithm for constructing the convex hull for a set of imprecise line segment intersection is presented. The organization of this paper is as follows. In the next section related works and history of convex hull for imprecise points are studied. In Section 3, our proposed approach is presented. In Section 4 the time complexity of our algorithm is discussed. Finally Section 5 is devoted to proposing the open problem.

2 Related works

Nagai and Tokura [2] computed the union and intersection of all possible convex hulls to obtain bounds on any possible solution. As imprecision regions, they used circles and convex polygons to obtain an O(nlogn) time algorithm.

Boissonnat and Lazard [3] studied the problem of finding the shortest convex hull of bounded curvature that contains a set of points, and showed that this is equivalent to finding the shortest convex hull of a set of imprecise points modeled as circles that have the specified curvature. They also presented a polynomial time approximation algorithm.

One of the latest papers about convex hull of imprecise points is due to Löffler [4]. In this paper, a relatively detailed work, the largest and smallest convex hull has been proposed for imprecise points. In this paper, various types of problems have been studied and many algorithms have been proposed to solve them.

3 Introducing our proposed method

Before introducing our proposed method, we need to become familiar with some concepts of imprecise lines and how to model them.

3.1 Imprecise Line

An imprecise line is a line for which we don't know exactly where it is, but we do have some information about it. To model it, the definition proposed by Löffler in [5] is used. According to this model, each imprecise line is modeled by a set of candidate lines, that is, a set L of lines in the plane $\mathbb{R}^2$. These lines must admit the following three features:

1. Limit angle: There should exist two lines l, $m \in$ L, which have the smallest and largest slope. Let the angle between l and m be α. We call α, the limit angle of L.
2. Connectedness: If two lines can be transformed into each other by a continuous movement inside L, they are connected.

3. Convexity: We will call a set of lines L convex if there exists a direction d such that no line *l* in L lies in direction d, and for any pair of lines *l* and *m* in L the followings hold:
   a. If *l* and *m* are parallel, then all other lines parallel to them and between them are also in L.
   b. If *l* and *m* intersect in a point p, then all other lines through p with a direction between l and m that can be rotated until they do not encounter d, are also in L.

Figure 1 shows an imprecise line.

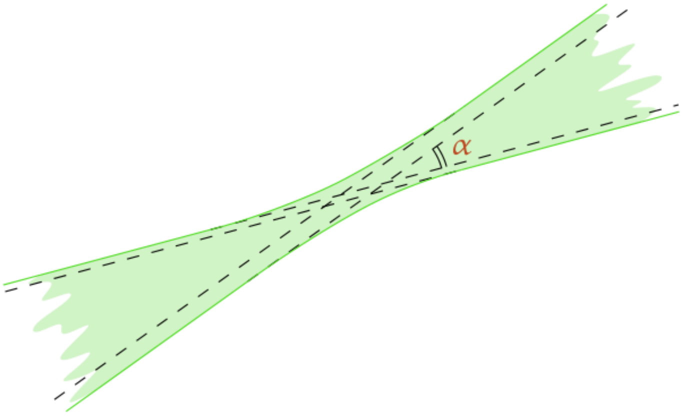

Figure 1. An imprecise line with limit angle α.

For the sake of simplicity, the set of candidate lines for imprecise lines (the set L of lines in $\mathbb{R}^2$) is called *bundle*. If all of candidate lines are oriented, then bundles have orientation (Fig. 2-a). The union of all half planes to the right of lines in L is called u_L (Fig. 2-b) and the intersection of all half planes to the right of lines in L is called I_L (Fig. 2-c).

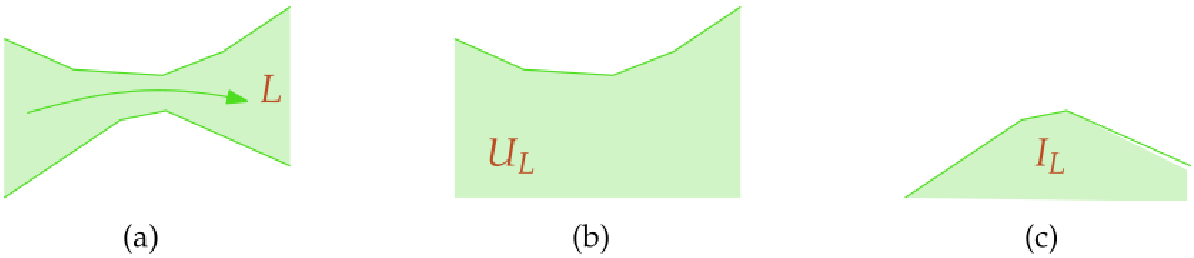

Figure 2. Characteristic of oriented bundle L

4 Main results

4.1 Finding the convex hull of ordinary line segment intersection

Convex hull can be defined for intersecting line segments and is the convex hull of intersection points of them. Computing the convex hull of ordinary (non-imprecise) line segments is simple and was done already by *Mikhail Atallah*, where he proposed an algorithm of $O(nlogn)$ time [6]. The algorithm is as follows:

1. Sort the lines in descending order in terms of slope.
2. Find the set of intersection points of consecutive lines in the ordering (including the last and first lines).
3. Compute the convex hull of these points using an $O(nlogn)$ algorithm such as Graham Scan.

Fig. 3 shows an example computing the convex hull of some line segment intersection:

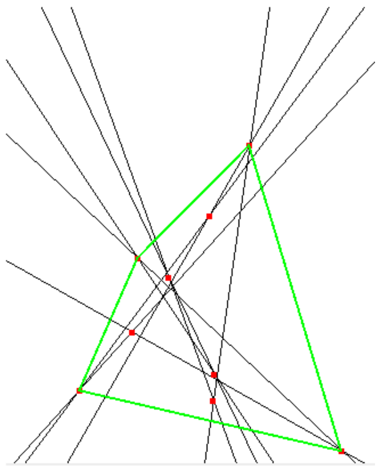

Figure 3. Compute the convex hull of ordinary line segment intersection

4.2 Our algorithm for convex hull of imprecise line segment intersection

To find the convex hull of intersection of imprecise line segments, first we must compute the border of bundle. Therefore, we have to calculate the upper and lower boundaries for each bundle and then use the algorithm of finding convex hull for ordinary line segment intersection. Thus the steps of our algorithm are as follows:

1. Find the boundary segments u_L and l_L for every bundle.
2. Sort the boundary segments u_L and l_L in descending slope order.
3. Find the intersection point of boundary segments u_L and l_L .

4. Compute the convex hull of these points using an $O(nlogn)$ algorithm such as Graham Scan.

Fig. 4 shows an example of our algorithm for finding the convex hull of imprecise line segment intersection.

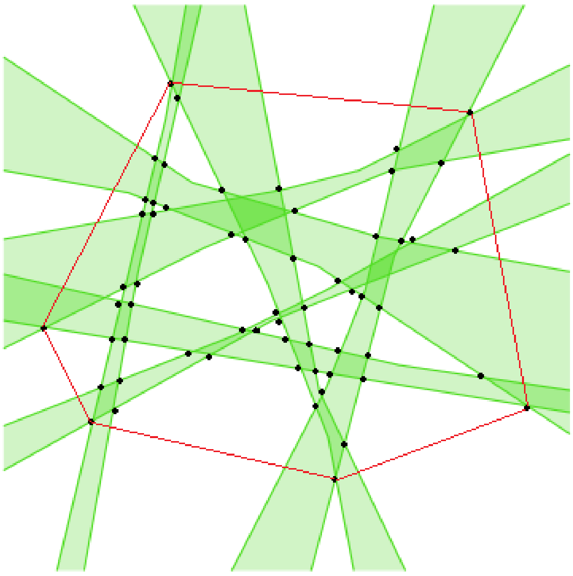

Figure 4. Computing the convex hull of seven imprecise lines

4.3 Time complexity of our proposed algorithm

Our algorithm has four steps. The first step has linear complexity, because the algorithm in [5] to calculate the boundary segments u_L and l_L runs in linear time. The second step, sorting, takes $O(nlogn)$ time. Calculating the intersection of n line segments has $O(nlogn)$ Order, [6]. Finally, calculating the convex hull of n points with Graham Scan algorithm is of order $O(nlogn)$. Thus the total time order is $O(nlogn)$, which is optimal and acceptable.

4.4 Implementation

For the implementation we used the C# programming language. To examine the intersection of two line segments, *doIntersect* method is used, where the method *orientation* is called to determine whether two lines have the same direction and if so the third method *onSegment* checks whether the two lines coincide with each other. If two lines are crossed, the coordinates of the intersection point is calculated. Finally the convex hull of all intersection points is obtained using Graham Scan algorithm. Fig. 5 shows an execution of the algorithm.

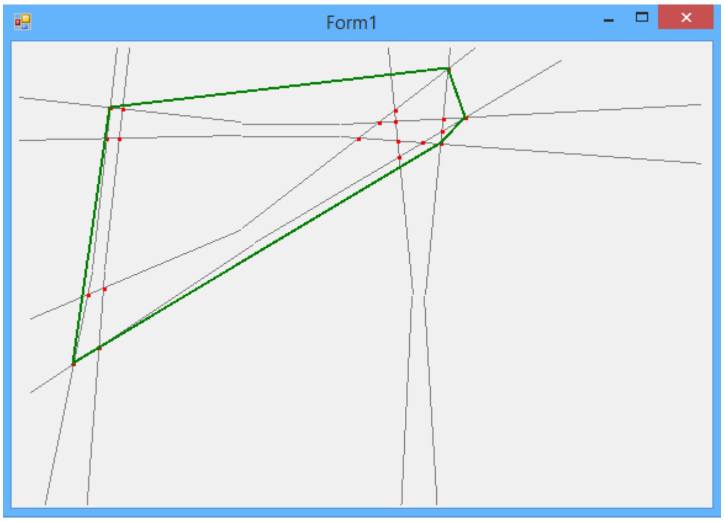

Figure 5. An execution of the proposed algorithm in C#

5 Open Problems

In this paper we described an algorithm for calculating the convex hull of intersecting imprecise line segments. To model imprecise lines, we used a set of candidate lines called bundles. Then we calculated the convex hull of intersecting imprecise lines by finding the union and intersection of all right-side half planes of each bundle, sorting line segments on their slopes, and finally using the Graham Scan algorithm. As we mentioned the total time complexity of our approach is $O(nlogn)$, where n is the number of input lines. This time order is optimal and reasonable. We also note that in the computation of convex hull of intersecting imprecise line segments, we used Graham Scan algorithm, now it makes perfect sense to apply some changes in the implementation and use other fast algorithms instead of famous Graham Scan. Several open problems are left to be considered in the future works.

First how one can compute the convex hull of moving imprecise line segments? Is it possible to maintain this problem in $O(nlogn)$ time as well?

Second how about the convex hull of other imprecise objects instead of line segments like rectangles, triangles or circles? For example for a rectangle, it could be modeled by four line segments and so if the number of rectangles is n, then the number of such line segments is 4n and one can continue the ideas introduced in this paper to obtain the convex hull of imprecise rectangles.

Finally, we note that the same problem can be considered for objects in three dimensions where techniques of 3D-convex hull construction shall be applied.

ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to Dr. Farnaz Sheikhi for reading the first draft of this paper and giving very fruitful comments and useful feedbacks.

References

[1]. Davoodi, M., Khanteimouri, P., Shikhi, F., Mohades, A., Data Imprecision under λ -Geometry: Finding the Largest Axis-Aligned Bounding Box, EuroCG'11 Morschach, Switzerland, March 28-30 (2011), 135-138.

[2]. Nagai, T., Tokura, N., Tight error bounds of geometric problems on convex objects with imprecise coordinates. In: Proceedings of the 4th Japanese Conference on Discrete and Computational Geometry. LNCS, vol. 2098, pp. 252–263. Springer, Berlin (2000).

[3]. Guibas, L., Salesin, D., Stolfi, J., Constructing strongly convex approximate hulls with inaccurate primitives. Algorithmica 9 (1993), 534–560.

[4]. Löffler, M., Van Kreveld, M., Largest and Smallest Convex Hulls for Imprecise Points. Algorithmica 56, 2 (2010), 235-269.

[5]. Löffler, M., Van Kreveld, M., Geometry with Imprecise Lines. EuroCG'08 Nancy – March 18-20, (2008), 133-136.

[6]. Atallah, M., Computing the Convex Hull of Line Intersections. Computer Science Technical Reports, Purdue university, Report number: 84-499 (1984).