Pseudo Triangulation of Point Set Using Polygon Construction

Reza Naghizadeh Majid
Faculty of Information Science and Technology, UKM, Malaysia
r.naghizade@gmail.com

Abstract— **Minimum weight pseudo triangulation is one of the computational geometry problems that has not found distinct algorithm and this is an open problem in computational geometry. Recently only one approximation algorithm was presented that its results are close to optimal results. In this paper we propose another algorithm based on construction of simple polygons. After running both algorithms, we consider results of each algorithm in various samples, and comparison of both results show efficiency of our algorithm.**

Keywords-computational geometry; minimum weight pseudo triangulation; polygon

I. Introduction

Computational geometry is a field of computer science that uses algorithmic approaches to solve the geometric problems. The main purpose of computational geometry to consider as a scientific is advanced in computer graph.

Pseudo triangulations are planar partitions that recently received considerable attention mainly due to their applications in visibility, ray shooting, kinetic collision detection, rigidity, and guarding [1,2]. A pseudo triangle is a planar polygon that has exactly three convex vertices, called corners. Convex vertex is the vertex which internal angle is less than π degree. If all angles of simple polygon except angles of unbounded face were less than π, we called this polygon as convex polygon. Triangle is a convex polygon and it is a type of pseudo triangulation [3].

A pseudo-triangulation of a set S of n points in the plane is a partition of the convex hull of S into pseudo triangles whose vertex set is exactly S. The convex hull of finite set of points in the plane is the smallest convex polygon P that encloses S, smallest in the sense that there is no other polygon P' such that P ⊃ P' ⊃ S [4].

One of the problems in pseudo triangulation which recently was suggested by Rote et al. is found an algorithm for pseudo triangulation of point set that finally total length of edges which called as weight of pseudo triangulation be minimum [5].

This paper is organized as follows. In sections 2 and 3 we explain an algorithm that produces pseudo triangulation of simple polygon and algorithm of Levcopoulos and Gudmundsson that give for pseudo triangulation of point set [6]. In section 4 we present our algorithm and finally compare results of algorithms.

II. Minimum Weight Pseudo Triangulation of Simple Polygon

Let $p_1 \ldots p_n$ be the vertices of P in counter-clockwise order. Let $\delta(p_i, p_j)$ be the shortest (directed) geodesic path from p_i to p_j within P, and $L[i, j]$ is a total edge length of an optimal pseudo triangulation between i and j Fig. 1b.

Define the order of a pair of points p_i , p_j to be the value $((n+j-i) \bmod n)$, i.e., the number of edges on the path from p_i to p_j along the perimeter of P in counter-clockwise order. Sort the pairs with respect to their order, ties are broken arbitrarily. Note that every pair of points p_i and p_j will occur twice, once as (p_i, p_j) and once as (p_j, p_i). Process each pair in sorted order as follows.

Assume we are about to process (p_i, p_j) and that the path $\delta(p_i, p_j)$ goes through the vertices p_{a0} , p_{a1} ,…, p_{ak-1}, p_{ak}, where $p_i = p_{a0}$ and $p_j = p_{ak}$. As above we define P' to be the (maybe degenerate) subpolygon of P bounded by the counter-clockwise path p_i, p_{i+1},…, p_j along the boundary of P and the path $\delta(p_i, p_j)$. The path $\delta(p_i, p_j)$ is said to be convex if it turns right at every intermediate vertex along $\delta(p_i, p_j)$ from p_i to p_j , as shown in Fig. 1a. If the path does not contain any right hand turns along $\delta(p_i, p_j)$ from p_i to p_j then $\delta(p_i, p_j)$ is said to be concave, see Fig. 1.

- If (p_i, p_j) contains only one edge then (p_i , p_j) is one of the edge optimal pseudo triangulation and the length of (p_i , p_j) add to $L[i, j]$.
- If $\delta(pi, p_j)$ is concave path containing more than one edge. We know that any pseudo triangle must have three concave chains and each chain create one convex vertex with connecting to other chain, so in chain $\delta(p_i, p_j)$ that is concave, p_i, p_j *is* two convex vertex and we choose p_m as a third vertex, between p_i, p_j, then we obtain optimal pseudo triangle between $\delta(p_i, p_m)$ and $\delta(p_m, p_j)$ and add the total edge length of pseudo triangle to $L[i, j]$.
- If $\delta(p_i, p_j)$ is convex path containing more than one edge then we will have several polygons for each edges of p_{al} , p_{al+1} ,… , p_{j-1}, $i < a_l < j$ that have already been computed.

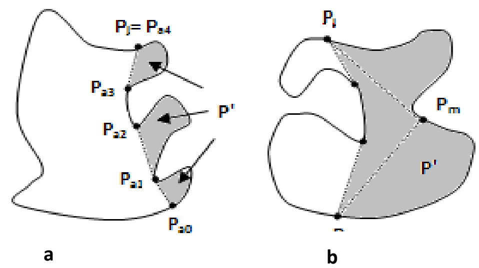

Figure 1. (a) $\delta(p_i, p_j)$ is a convex path and P' is a degenerate polygon (b) $\delta(p_i, p_j)$ is a concave vertex

Algorithm Pseudo triangulation (P).

Input: polygons

Output: minimum pseudo triangulation of polygon

For all (p_i, p_j) **do**.
  Find the number of points between (p_i, p_j).
  Sort pair of points respect on their order.
  Find path of $\delta(p_i, p_j)$
    If number of path (p_i, p_j) = 1 **then**
      Calculate $L[i, j]$.
  else
    If path of $\delta(p_i, p_j)$ is convex **then**
      /* $L[i, j]$ has computed before */
  else
    If path of $\delta(p_i, p_j)$ is concave **then**
      For $i \leq m \leq j$ **do**
        Compute (p_i, p_m) , (p_m, p_j).

Figure 2. Pseudo code of optimal pseudo triangulation for simle polygon

If we do all above steps for each edge we computed the minimum weight pseudo triangulation for simple polygon in $O(n^3)$, so we can draw the minimum weight pseudo triangulation for polygon P (see Fig. 2).

III. AN APPROXIMATION ALGORITHM

In this section we will give an approximation algorithm for the minimum weight pseudo triangulation problem [5]. As we mentioned before, there is not exact algorithm for minimum weight pseudo triangulation, but Levcopoulos and Gudmundsson presented an approximation algorithm that results is close to the optimal results.

As input we are given a set S of n points in the plane, first we construct the convex hull and minimum spanning tree of S, this partition convex hull of S into simple polygons. Finally, according to the previous section we can pseudo triangulate these simple polygons, and we can compute pseudo triangulation of point set in time $O(n^3)$ Fig. 3,4, also we can see the pseudo code of algorithm in Fig. 5.

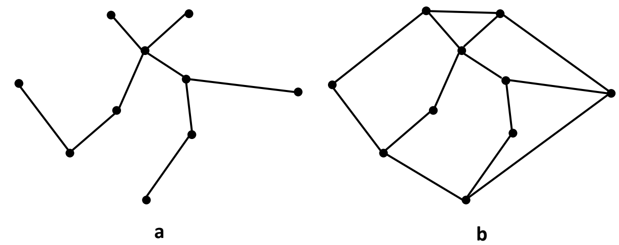

Figure 3. (a) minimum spanning tree (b) convex hull

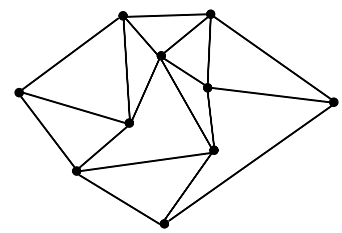

Figure 4. Pseudo triangulation of polygons

Algorithm Approximation Algorithm (S).

Input: Set of points S.

Output: Pseudo triangulation of S.

Construct minimum spanning tree(S).
Construct *convex hull* ()
For all polygons P' **do**.
  Pseudo triangulation (P').

Figure 5. Pseudo code of approximation algoritm

IV. THE PROPOSED ALGORITHM

This algorithm is based on construction of simple polygon that pseudo triangulate the point set in $O(n^3)$ time. Such that, assume a point set of S in plane and they are allocated by (x, y) coordinates for each point, then follow these steps:

1) Find minimum (x, y) of given points.

2) Compute the angle of other points to this point, and sort the points due to their angles.

3) Connect each of these points respectively with an edge, and then connect last point to the first point, Fig. 6.

4) Compute the convex hull of this polygon. Hence, we create several simple polygons and we can easily pseudo triangulate these simple polygons according to the section 2, Fig. 7.

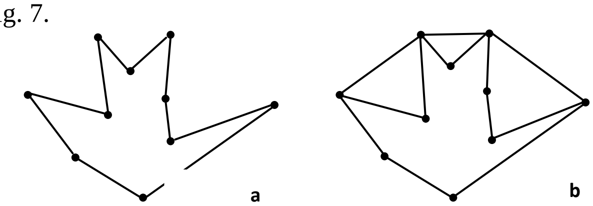

Figure 6. (a) construct a polygon (b) convex hull

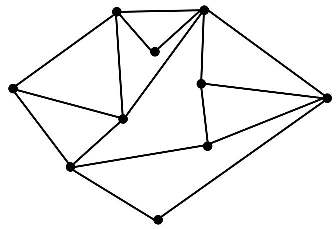

Figure 7. Pseudo triangulation of polygons

Algorithm Pseudo triangulation of points set based on polygon construct ()

Input: Set of points *S*
Output: Pseudo triangulation of points

Find the lowest point (smallest *y* and *x*)
Compute counter clockwise angle θ between lowest point and the other points.
Sort all points respect on their angle and put it in S_1.
Draw (lowest point, $S_1[0]$) as an edge.
For all index of S_1 draw an edge **do**
    Draw ($S_1[n]$, lowest point) as edge.
Construct *convex hull* ()
For all polygons *P′* **do**
    Pseudo triangulation (*P′*)

Figure 8. Pseudo code of proposed algorithm

V. RESULTS

In this paper we presented an approximation algorithm and our proposed algorithm. Now in this section we compare results of each algorithm for various samples.

After implementing two algorithms, we put *n* points in various positions, and apply these two algorithms for them. For each position, we calculate the length of edges obtained from pseudo triangulation of point set, then average them. The results show that the average values obtained by proposed algorithm for each *n* points is less than approximation algorithm, so it can be close to optimal result.

The feature of our algorithm that causes to obtain better results than other is that, numbers of simple polygons created in our algorithm are less than approximation algorithm, so it can help us to use less number of edges in pseudo triangulating of point set and that is very effective in decreasing weight of pseudo triangulation.

Now we show the above comparisons for some samples. Table I shows the average weights obtained by two algorithms for each point set. We compare results until 30 points and put the average weight obtained in centimeter for any number of points in different positions in this table.

In Fig. 9 we draw the graph obtained from 30 points, and that is clear that the results of proposed algorithm are better than approximation algorithm.

TABLE I. THE AVERAGE WEIGHTS OBTAINED BY APPROXIMATION ALGORITHM AND PROPOSED ALGORITHM

Number Of Points	Approximation Algorithm (cm)	Proposed Algorithm (cm)
5	674.3	640.7
10	1451.2	1429
15	2053.3	2008.5
20	2542.1	2528.3
25	2896.5	2882.1
30	3685.4	3681.2

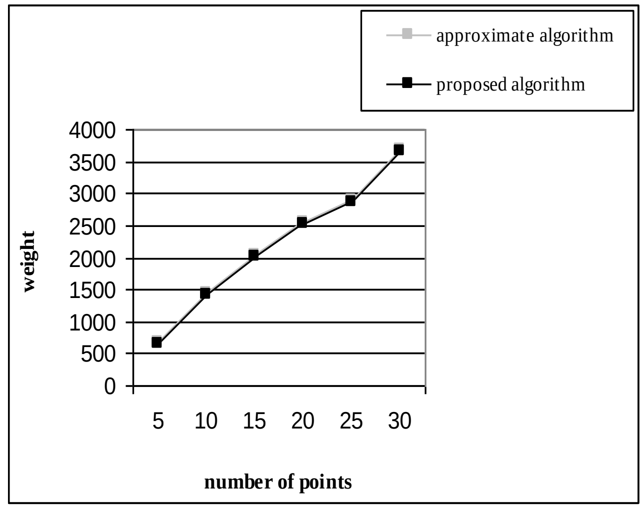

Figure 9. Comparison between approximate algorithmand proposed algorithm for 30 points

REFERENCES

[1] Pocchiola and Vegter. Topologically sweeping visibility complexes via pseudo triangulations. Discrete & Computational Geometry, 16(4):419–453, 1996.

[2] G. Rote, F. Santos, and I. Streinu. "pseudo triangulation a survay", Discrete Applied Mathematics, Volume 154, Issue17, Pages 2470_2483, 15 November 2006.

[3] O. Aichholzer, G. Rote, B. Speckmann, and I. Streinu. “The Zigzag Path of a Pseudo-Triangulation”, Proc 8th International Workshop on Algorithms and Data Structures, pp. 377-388, Lecture Notes in Computer Science 2748, Springer Verlag, 2003.

[4] Joseph O’Rourke, Computational Geometry In C, Published by the Press Syndicate of the university of Cambridge CB2 IRP 40 West 20th Street,New York , NY 10011-4211,USA, 1993.

[5] Rote, Wang, Xu,” On constrained minimum pseudotriangulations”, in: Proc. 9th Symposium on Computing and Combinatorics, in: LNCS, vol. 2697, pp. 445–454, 2003.

[6] Joachim Gudmundsson, Christos Levcopoulos, “Minimum weight pseudo triangulations” Computational Geometry, Volume 38, Issue 3, Pages 139-153, October 2007.