New Algorithms for Computing Maximum Perimeter and Maximum Area of the Convex Hull of Imprecise Inputs Based On the Parallel Line Segment Model

Wenqi Ju* Jun Luo†

Abstract

In this paper, we present new algorithms for computing maximum perimeter and maximum area of the convex hull of imprecise inputs based on the parallel line segment model. The running time of our algorithms for maximum perimeter problem is $O(n^4)$ which improve over the previous results of $O(n^5)$ in [12]. For maximum are problem with different size of parallel line segments [12] gives $O(n^3)$ algorithms. We give $O(n \log n)$ algorithms with the assumption that all parallel line segments have the same size.

1 Introduction

The problem of computing the convex hull has been studied for several decades because it is the fundamental problem in computational geometry. Computing convex hull covers many other application domains such as pattern recognition [1], data mining [3], stock cutting and allocation [5], image processing [4] and so on.

All the classic algorithms for computing convex hull are based on the assumption that the locations of the input points are known exactly. But, in practice, that is not the case. We can, more often than not, only obtain the data varying within some range. For example, the locations of a moving object have the uncertain property [8]. Moreover the location data obtained from physical devices are inherently imprecise due to measurement error, sampling error and network latency [9], [7]. Location privacy protection is another issue may lead to imprecise data [2], [10], [11].

We could attack the problems with imprecise data by using the model of parallel line segment, circle and square etc. In this paper, the model of parallel line segment is studied. In [12] the problems of the largest area and perimeter of convex hull based on the parallel segment model are studied and their running time are$O(n^3)$ and $O(n^5)$ respectively.

In this paper, we present a new algorithm to compute the largest area based on the same size parallel line segment model and the running time is $O(n \log n)$. Then we also present an algorithm to compute the largest perimeter of convex hull and its running time is $O(n^4)$.

*Institute of Computing Technology, Chinese Academy of Sciences, China, wq.ju@sub.siat.ac.cn

†Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China, jun.luo@sub.siat.ac.cn

2 Problem Definition

The problems we discuss in this paper are as follows:

Problem A Given a set of parallel line segments, choose a point on each line segment such that the area of the convex hull of the resulting point set is as large as possible.

In practice, the imprecise inputs often have the same error range. Thus we can also define the alternative version of Problem A as follows:

Problem B Given a set of parallel line segments of the same size, choose a point on each line segment such that the area of the convex hull of the resulting point set is as large as possible.

Problem C Given a set of parallel line segments, choose a point on each line segment such that the perimeter of the convex hull of the resulting point set is as large as possible.

3 Introduction to the Algorithms by Löffler and van Kreveld [12]

In this section we give brief introduction of the algorithms by Löffler and van Kreveld [12] for Problem A and Problem C since several interesting properties of [12] are used in this paper. Let L=$l_1, l_2, ..., l_n$ be a set of n vertical line segments, where l_i lies to the left of l_j if $i < j$. Let l_i^+ denote the upper endpoint of l_i, and l_i^- denote the lower endpoint of l_i. In [12], the authors prove the following lemma:

Lemma 1 *There is an optimal solution to Problem A and Problem C such that all points are chosen at endpoints of the line segments.*

3.1 The Algorithm for Maximum Area Convex Hull

In order to solve Problem A, Löffler and van Kreveld consider the polygon P_{ij} for $i \neq j$ which starts at l_i^+ and ends at l_j^-, and optimally solves the subproblem to the left of these points, that is, contains only vertices l_k^+

with $k < i$ or l_k^- with $k < j$, but not both for the same k, such that the area of the polygon is maximal.

The authors [12] solve Problem A by using dynamic programming and the solution to it is either of the form P_{kn} or P_{nk} for some $0 < k < n$, and can thus be computed in linear time once all P_{ij} are known. When $1 < i < j$, then

$$P_{ij} = \max_{k<j;k\neq i}(P_{ik} + \triangle l_i^+ l_j^- l_k^-) \quad (1)$$

Analogously, when $1 < j < i$, then

$$P_{ij} = \max_{k<i;k\neq j}(P_{kj} + \triangle l_i^+ l_j^- l_k^+) \quad (2)$$

The algorithm runs in $O(n^3)$ time.

3.2 The Algorithm for Maximum Perimeter of The Convex Hull

In order to solve Problem C, the authors of [12] define $P_{ij\sim km}$ to be the chain that starts at l_i^+, then goes to the left to l_j^+, then via a number of other upper endpoints to the leftmost point, then back to the right via l_k^-, and finally to l_m^-, such that it is convex and of maximal length. They also use $P_{ij\sim km}$ to denote its length.

if $i < m$, then

$$P_{ij\sim km} = \max_{h<k}(P_{ij\sim hk} + \overline{l_k^- l_m^-} | \angle l_h^- l_k^- l_m^- \textit{ is convex}) \quad (3)$$

if $m < i$, then

$$P_{ij\sim km} = \max_{h<j}(P_{hj\sim km} + \overline{l_j^+ l_i^+} | \angle l_h^+ l_j^+ l_i^+ \textit{ is convex}) \quad (4)$$

Because every $P_{ij\sim km}$ has to be solved by using the solution of $P_{ij\sim hk}$ in equation (3) and $P_{ij\sim hk}$ in equation (4), the complexity of the algorithm is $O(n^5)$.

4 Our Algorithm for the Largest Area with the Same Size Parallel Line Segments

According to [12], the vertices on the optimal convex hull must be endpoints of the segments; vertices on the upper chain must be upper endpoints, and vertices on the lower chain must be lower endpoints. Also, the leftmost and rightmost vertices on the optimal hull must be endpoints of the leftmost and rightmost segments.

We illustrate our algorithm as follows:

1. If the mid-points of all the segments lie on the same line, then the maximum area convex hull only depends on the endpoints of l_1, l_2, l_{n-1}, l_n and has nothing to do with the endpoints of other line segments. Therefore we could get the maximum area convex hull in $O(1)$ time.
2. If the mid-points of all the segments don't lie on the same line. We construct the convex hull of l_1^+ and l_n^+ with all endpoints of l_2 to l_{n-1} and denote it as CH_1. Similarly, CH_2 is the convex hull of l_1^- and l_n^+ with all endpoints of l_2 to l_{n-1}^+ and CH_3 is the convex hull of l_1^+ and l_n^- with all endpoints of l_2 to l_{n-1} and CH_4 is the convex hull of l_1^- and l_n^- with all endpoints of l_2 to l_{n-1}.

We denote the upper and lower convex convex chain of CH_j $(1 \leq j \leq 4)$ as UCC_j and LCC_j respectively and denote the leftmost vertex and the rightmost vertex of CH_j as P_{left}^j and P_{right}^j respectively. Note that the vertices of UCC_j (LCC_j) can only be the vertices of upper (lower) endpoints of l_2 to l_{n-1} with P_{left}^j and P_{right}^j.

For the remaining part of this paper, we assume that the mid-points of all the segments don't lie on the same line. We also assume that $n \geq 4$.

Lemma 2 *The area of convex hull CH_j $(1 \leq j \leq 4)$ is no less than the area of all the convex hulls formed by choosing a point on each line segment of l_2 to l_{n-1} plus P_{left}^j and P_{right}^j.*

Proof. We denote the convex hull, of which the leftmost vertex is P_{left}^j and the rightmost vertex is P_{right}^j but not $CH_j (1 \leq j \leq 4)$, as CH_j^* and their upper chain as UCC_j^* and their lower chain as LCC_j^*. For a vertical line l, let the y coordinates of the four intersection points of l with $UCC_j, UCC_j^*, LCC_j, LCC_j^*$ be y_U, y_U^*, y_L, y_L^* respectively. Now we prove $y_U \geq y_U^*$ and the proof of $y_L \leq y_L^*$ is analogous. Suppose $y_U < y_U^*$, let the line segment of UCC_j^* intersecting with l be $l_i^+ l_k^+$, where $1 \leq i < k \leq n$. Therefore we know at least one of the endpoints of $l_i^+ l_k^+$ is above UCC_j which contradicts with the construction of UCC_j. Thus according to the theory of area integral, the area of CH_j must be maximum among all the convex hulls formed by choosing a point on each line segment of l_2 to l_{n-1} plus P_{left}^j and P_{right}^j. □

From lemma 2, it seems that we only need to compute the areas of CH_j $(1 \leq j \leq 4)$ and the maximum one is the answer to Problem B. However note that some of CH_j is not valid solution to our problem since it may use two endpoints of the same line segment as vertices of CH_j. Nevertheless, we can prove there are at most two such kind of line segments.

Lemma 3 *If there is a line segment l_k $(2 \leq k \leq n-1)$ whose upper endpoint lies on UCC_j and whose lower endpoint lies on LCC_j $(1 \leq j \leq 4)$, then it must be l_2 or l_{n-1}.*

Proof. According to the construction of CH_j $(1 \leq j \leq 4)$, we know that the parallelogram $l_2^+ l_{n-1}^+ l_{n-1}^- l_2^-$

is included in CH_j. Suppose there is a line segment $l_k(3 \le k \le n-2)$ whose upper endpoint lies on UCC_j and whose lower endpoint lies on LCC_j $(1 \le j \le 4)$. Since l_k is between l_2 and l_{n-1}, it must go through the parallelogram $l_2^+ l_{n-1}^+ l_{n-1}^- l_2^-$ which means the length of l_k is greater than the length of l_2 and l_{n-1}. It contradicts the fact that all the line segments have the same size. □

Lemma 4 *There is at least one CH_j $(1 \le j \le 4)$ such that all vertices of CH_j are the end points of different line segments.*

Proof. According to lemma 3, if there is a line segment whose upper and lower endpoint are the vertices of CH_j, it must be l_2 or l_{n-1}. Without loss of generality, we assume it is l_2 and CH_j is CH_2 which means the leftmost vertex is l_1^- and the rightmost vertex is l_n^+. We substitute the vertices l_1^- with l_1^+ (actually the newly formed convex hull is CH_1) and construct the new upper convex chain UCC_1 and new lower convex chain LCC_1 which start at l_1^+ and end at l_n^+. If l_2^+ is not on UCC_1, it must be below UCC_1 which means it is inside CH_1. Moreover, because l_1^+ is higher than l_1^-, the vertices of LCC_1 remain the same as the vertices of LCC_2 except the leftmost vertex and no new vertex except the most left vertex will be added into the upper convex chain. Therefore l_2 is no longer the one whose upper endpoint and lower endpoint are the vertices of the new convex hull and no other line segment becomes such kind of segment.

If l_2^+ is the vertex of UCC_1, all the upper endpoints of the line segments l_k $(3 \le k \le n)$ must be on or below the line $\overline{l_1^+ l_2^+}$ and the lower endpoint of the line segments l_k $(3 \le k \le n)$ must be on or above $\overline{l_1^- l_2^-}$ according to the property of convex hull. Since all the line segments are the same size, $\overline{l_1^+ l_2^+}$ and $\overline{l_1^- l_2^-}$ are two parallel lines and all upper endpoints of l_1 to l_n are on $\overline{l_1^+ l_2^+}$ and all lower endpoints of l_1 to l_n are on $\overline{l_1^- l_2^-}$. Thus the midpoints of l_1 to l_n are in the same line which contradicts our assumption that the mid-points of all the segments of l_2 to l_{n-1} don't lie on the same line. Therefore l_2^+ could not be the vertex of UCC_1. □

Lemma 5 *If there is a line segment l_2 or/and l_{n-1} whose upper endpoint lie on UCC_j and whose lower endpoint lie on $LCC_j(1 \le j \le 4)$, then P_{left}^j or/and P_{right}^j must not be the vertices of the optimal convex hull.*

Proof. Without loss of generality, we assume the two endpoints of l_2 are the vertices of CH_2. According to lemma 4, there is no line segment whose upper endpoint and lower endpoint are the vertices of CH_1. Now we prove $area(CH_1) \ge area(CH_2)$.

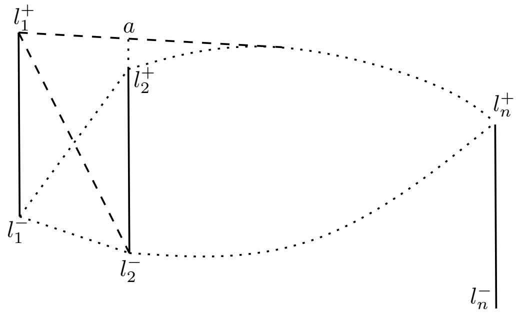

Figure 1: Illustration of lemma 5.

We extend the line segment l_2 and let it intersect with UCC_1 at point a (see Figure 1). The line segments al_2^- separate CH_1 into left and right parts CH_1^L and CH_1^R and separate CH_2 into left and right parts CH_2^L and CH_2^R. We can see CH_2^R is inside CH_1^R which means $area(CH_2^R) \le area(CH_1^R)$. Actually CH_1^L is the triangle $\triangle l_1^+ l_2^- a$ and CH_2^L is the triangle $\triangle l_1^- l_2^- l_2^+$. $area(\triangle l_1^+ l_2^- a) \ge area(\triangle l_1^- l_2^- l_2^+)$ since they have the same height (the horizontal distance between l_1 and l_2) but $|l_2^- a| \ge |l_2^- l_2^+|$. □

According to the lemma 4, there is at least one valid convex hull among CH_j $(1 \le j \le 4)$. The invalid convex hull is not the optimal solution according to lemma 5. Therefore we only need to compare the area of four convex hulls CH_j in order to find out the optimal convex hull. The running time is $O(n \log n)$ since n line segments have to be sorted. If n line segments are presorted, then the running time is $O(n)$ since UCC_j and LCC_j $(1 \le j \le 4)$ can be computed in linear time.

Theorem 6 *There is an $O(n)$ algorithm for Problem B provided that the input is sorted. Otherwise the running time is $O(n \log n)$.*

5 $O(n^4)$ Algorithm for Maximum Perimeter Problem

In this section, we will illustrate our $O(n^4)$ algorithm for Problem C in detail. The improvement of linear factor for the running time comes from that we can solve the recursive equations 3 and 4 in constant time instead of linear time. Now we show how to achieve this. We only focus on equation 4 since the other one is symmetric.

First of all, for each line segment l_i, we calculate angles α^+ between the line segment $l_f^+ l_i^+$ and l_i where $1 \le f < i \le n$ (see Figure 2). Denote the set of the above angles as $\mathcal{A}_i^+$. Then we sort the $\mathcal{A}_i^+$ in ascent order which needs $O(n \log n)$ for every i. We also need to calculate every angle β^+ between the line segment $l_r^+ l_i^+$ and l_i where $1 \le i < r \le n$. Denote the set of the above angles as $\mathcal{B}_i^+$. Then we sort them in descent order.

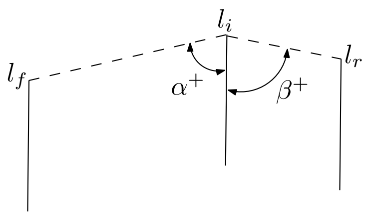

Figure 2: the angle α^+ and the angle β^+.

Let $\mathcal{A}_i^+[g]$ and $\mathcal{B}_i^+[g]$ denote the g-th element in $\mathcal{A}_i^+$ and $\mathcal{B}_i^+$ respectively. For each $\mathcal{B}_i^+[g]$, we can find $\mathcal{A}_i^+[g']$ such that $\mathcal{A}_i^+[g'] \leq \mathcal{B}_i^+[g] \leq \mathcal{A}_i^+[g'+1]$ in $O(\log n)$ time. If $\mathcal{B}_i^+[g]$ corresponds the angle $\angle l_i^- l_i^+ l_r^+$ and $\mathcal{A}_i^+[g']$ corresponds the angle $\angle l_i^- l_i^+ l_f^+$, let $\mathcal{L}^+[ri] = f$. Obviously, the total running time for this preprocessing step is $O(n^2 \log n)$.

Suppose j, k, m are fixed and h could be any number less than j and not equal to k, m. We compute $P_{jh\sim km}$ in the order as the corresponding α_j^+ appears in $\mathcal{A}_j^+$. After compute each $P_{jh\sim km}$, we not only store the value of $P_{jh\sim km}$ but also keep the largest value of $P_{j1\sim km}, P_{j2\sim km}, ..., P_{jh\sim km}$ in a four dimension array C with index j, h, k, m. Now the recursive equation 4 becomes

$$P_{ij\sim km} = C[j, \mathcal{L}^+[ij], k, m] + \overline{l_j^+ l_i^+} \ \texttt{if} \ m < i \qquad (5)$$

We can compute this recursive equation in constant time. Thus in total the running time is $O(n^4)$.

Theorem 7 *Given a set of n arbitrary sized, parallel line segments, the problem of choosing a point on each segment such that the perimeter of the convex hull of the resulting point set is as large as possible can be solved in $O(n^4)$ time.*

6 Conclusions

In this paper, we present an algorithm for computing the largest area convex hull with the model of the same size parallel line segments. The running time is $O(n \log n)$. Then we also present an algorithm for computing the maximum perimeter convex hull of different size parallel line segments and the running time of our algorithm is $O(n^4)$. In the future, we want to extend our results for maximum area convex hull problem to more general situation such as the parallel line segments have $m < n$ different sizes and to unit square model.

References

[1] S. G. Akl and G. T.Toussaint. Efficient convex hull algorithms for pattern recognition applications. *Int. Joint Conf. on Pattern Recognition*, Kyoto, Japan, 483-487, 1978.

[2] A. R. Beresford and F. Stajano. Location Privacy in pervasive Computing. *IEEE Pervasive Computing*, 2(1), 2003.

[3] C. Bőhm and H. Kriegel. Determing the convex hull in large multidimensional databases. *Proceedings of the Third International Conference on Data Warehousing and Knowledge Discovery*, Lecture Notes in Computer Science 2114, 294-306, 2001.

[4] D. Taubman, High Performance Scalable Image Compression. *IEEE Transactions on Image Processing*, 9:1158-1170, 2000.

[5] H. Freeman and R. Shapira. Determining the minimum-area encasing rectangle for an arbitrary closed curve, *Comm. ACM*, 18(7):409-413, 1975.

[6] O. Wolfson, P. A. Sistla, S. Chamberlain, and Y. Yesha. Updating and querying databases that track mobile units. *Distributed and Parallel Databases*, 7(3):257-287,1999.

[7] D. Pfoser and C. S. Jensen. Capturing the uncertainty of moving-objects representations, *SSD '99: Proceedings of the 6th International Symposium on Advances in Spatial Databases*, 111-132, 1999.

[8] R. Cheng, D. V. Kalashnikov and S. Prabhakar. Querying Imprecise Data in Moving Object Environments. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1112-1127, 2004.

[9] P. A. Sistla, O. Wolfson, S. Chamberlain and S.Dao. Querying the uncertain position of moving objects, *In Temporal Databases: Research and Practice*. 310-337, 1998.

[10] R. Cheng, Y. Zhang, E. Bertino and S. Prabhakar, Preserving user location privacy in mobile data management infrastrutures. *Proc. of the 6th Workshop on Privacy Enhancing Technologies*, 2006.

[11] B. Gedik and L. Liu. A customizable k-anonymity model for protecting location privacy. *In ICDCS*, 620-629, 2004.

[12] M. Löffler, M. van Kreveld. Largest and Smallest Convex Hulls for Imprecise Points. *Algorithmica*, 2008.