アルゴリズム 53-10
(1996. 9. 13)

点集合の強凸-包含包を求めるアルゴリズム

陳慰, トウショウブン, 和田幸一, 川口喜三男

名古屋工業大学電気情報工学科

Sをサイズ n の平面上の点集合とするとき, Sのϵ-強凸-包含包は Sの凸包を拡張したもので, 次の条件を満たす凸多角形 Pである: (1)Pは n 個以下の頂点を持つ,(2)Pは Sの全ての点を含む, (3)Pの各頂点は Sの凸包との距離は d 以下である, (4)Pの各頂点が距離ϵ以内を移動しても Pが凸のままである. 本研究では,Sのϵ-強凸-包含包を求める初めてのアルゴリズムを提案する. このアルゴリズムは、任意の$\epsilon \geq 0$ に対して, $O(n \log n)$ 時間で Sの 4ϵ-強凸-包含包を求める.

Constructing A Strongly Convex Superhull of points

Wei Chen Xiao Wen Deng Koichi Wada *and* **Kimio Kawaguchi**

Department of Electrical and Computer Engineering
Nagoya Institute of Technology, Showa, Nagoya 466, Japan

E-mail: (chen, deng, wada, kawaguchi)@elcom.nitech.ac.jp

Let S be a set of n points in the plane and $CH(S)$ be the convex hull of S. We introduce a new concept of strongly convex approximate hulls called *ϵ-strongly convex δ-superhull*, which is a convex polygon P satisfying the following conditions: (1) P has at most n vertices, (2) P contains all the points of S, (3) no vertex of P lies farther than δ outside the convex hull of S, and (4) P remains convex even if its vertices are perturbed by as much as ϵ. In this paper, we present the first method for solving such a generalized convex hull problem. We show that an ϵ-strongly convex 4ϵ-superhull can be constructed in $O(n \log n)$ time for any $\epsilon \geq 0$.

1 Introduction

Given a set S of n points in the plane, the convex hull of S is the smallest convex polygon containing all the points of S. The convex hull problem is one of the simplest and the oldest problem in computational geometry and many algorithms about it have been developed [1-5,8,9]. Since the properties from the convexity usually make efficient algorithms, computing the convex hull problem is often used as a subroutine in many applications. It is natural that we may desire that the solution has strong convexity so that many properties from the convexity can be preserved in some fashion even if they are tested with imprecise computations such as ordinary floating point arithmetic.

The concept of strongly convex approximate hulls first appeared in the problem of constructing an ϵ-strongly convex δ-hull of a set S of points in the plane [11], which is a convex polygon P satisfying the following conditions:

(i) the vertices of P are taken from S,

(ii) no point of S lies farther than δ ($\delta \geq 0$) outside P and

(iii) P remains convex even if the vertices of P are perturbed by as much as ϵ ($\epsilon \geq 0$).

According to the definition, the ordinary convex hull is the 0-strongly convex 0-hull. It is easily understood that given an ϵ, the smaller the value of δ is, the higher the accuracy of the resulting ϵ-strongly convex δ-hull is. Li and Milenkovic present the first algorithm for the problem which computes an ϵ-strongly convex $(12\epsilon + 228\sqrt{2}\mu)$-hull in $O(n \log n)$ time [11]. Guibas, Salesin and Stolfi give an algorithm which computes an $(6\epsilon + 7\alpha)$-hull in $O(n^3 \log n)$ time [10]. And recently, Chen, Wada and Kawaguchi show a parallel algorithm which computes an $(6\epsilon + 16\beta)$-hull in $O(\log^3 n)$ time using n processors [6]. Note that μ, α and β are the error units of the primitive operations defined in their algorithms, respectively. The error units are equal to 0 when using exact arithmetic (the arithmetic causes no numeric errors).

One drawback of the above strongly convex approximate hulls is that the points of S may locate outside of the resulting hull. However, in many applications, the hulls which contain all the points of S are requested. It is easily understood that if ϵ is positive there may not exist any ϵ-strongly convex δ-hull which contains all the points of S, therefore, a new concept of strongly convex approximate hulls is necessary. In this paper, we consider the hulls which contain all the points of S. An *ϵ-strongly convex δ-superhull* of S is a convex polygon P satisfying the following conditions:

(i) P has at most n vertices,

(ii) P contains all the points of S,

(iii) no vertex of P lies farther than δ outside the convex hull of S, and

(iv) P remains convex even if its vertices are perturbed by as much as ϵ.

Note that in the above definition, the vertices of P are not requested to take from S and by this property P can contain all the points of S. It is seen that the ordinary convex hull of S is the 0-strongly convex 0-superhull. Let A be an ϵ-strongly convex δ-hull of S and let B be an ϵ-strongly convex δ-superhull of S. When using exact arithmetic, B can be obtained from A by expanding the boundary of A with size δ. Therefore, when using exact arithmetic, Li and Milenkovic's algorithm, Guibas, Salesin and Stolfi's algorithm and Chen, Wada and Kawaguchi's algorithm can construct an ϵ-strongly convex δ-superhull, where δ is 12ϵ, 6ϵ and 6ϵ, and running time is $O(n \log n)$ $O(n^3 \log n)$ and $O(n \log^3 n)$, respectively. But when using imprecise arithmetic, B can not be obtained from A easily since the numerical errors caused in the expansion may make the result even not convex. In this paper, we compute an ϵ-strongly convex 4ϵ-superhull of S in $O(n \log n)$ time with exact arithmetic. We expect that our method can be generalized to one for imprecise computation (we are going to prepare it in the forth coming paper [7]). Comparing with the above three algorithms, it is easily seen that (1) when using exact arithmetic, our algorithm constructs a more accurate ϵ-strongly convex approximate superhull and runs fastest, and (2) our method can be generalized to imprecise computation, on the other hand, it seems to be difficult for the other known algorithms.

2 Definitions and Lemmas

In this section, we give some definitions.

Definition 1 (δ-superhull) *A simple polygon P is a δ-superhull of a set S of points ($\delta \geq 0$), if P contains all the points of S, and no vertex of P lies farther than δ outside the convex hull of S.* ∎

Definition 2 (ϵ-strongly convex) *A simple polygon P is ϵ-strongly convex ($\epsilon \geq 0$), if P is convex and remains convex even after each vertex of P is perturbed as far as ϵ.* ∎

Definition 3 (the ϵ-strongly convex δ-superhull) *A simple polygon P is a ϵ-strongly convex δ-hull of a set S of points ($\epsilon \geq 0$), if P is a δ-superhull of S, P has at most n vertices and P is ϵ-strongly convex.* ∎

A spacial case is that S consists of the vertices of a polygon P. In this case, we use the items like: a *δ-superhull of polygon P* and a *ϵ-strongly convex δ-superhull of polygon P*.

Definition 4 (distance) *Let A, B and C be three points in the plane. Define $d(B, AC)$ to be the signed distance from point B to segment AC, where $d(B, AC) > 0$ if A, B and C are in counter-clockwise, $d(B, AC) < 0$ if they are in clockwise, or $d(B, AC) = 0$ if they are collinear.* ∎

Definition 5 (ϵ-convex vertex) *Let P be a convex polygon and A, B and C be three contiguous vertices of P. B is said to be ϵ-convex if distance $d(B, AC) \geq 2\epsilon$.*

Lemma 1 *([11]) Let P be a convex polygon. If each vertex of P is ϵ-convex, then P is ϵ-strongly convex.*

Finally, given two points p and q, let $l(p, q)$ be the straight line passing through p and q.

3 Sweep-and-Expansion Technique

Let S be a set of n points in the plane and $CH(S)$ be the convex hull of S. We construct an ϵ-strongly convex 4ϵ-superhull of S in two steps: (I) we construct $CH(S)$, the convex hull of S, (II) we construct an ϵ-strongly convex 4ϵ-superhull of $CH(S)$. Since (I) can be done by any known $O(n \log n)$ time convex hull algorithm, in the rest of the paper, we show how to construct an ϵ-strongly convex 4ϵ-superhull of a convex polygon with n vertices. Our algorithm runs in $O(n)$ time.

Let P be a convex polygon with n vertices. From Lemma 1, P is ϵ-strongly convex, if all the vertices of P are ϵ-convex. Therefore, to construct an ϵ-strongly convex 4ϵ-superhull of P, we scan the vertices of P one by one in counter-clockwise order and revise them if necessary so that each vertex of the resulting P is ϵ-convex and lies not farther than δ outside the original P. In the process, we call the vertex being visited *active* and denote the convex polygon obtained by then as P'. Suppose vertex B is active, and vertice A and C are the vertices directly before B and after B in P', respectively. Obviously, A is ϵ-convex if B is not the first visited vertex of P. We deal with B as follows. If distance $d(B, AC) \geq 2\epsilon$, vertex B is ϵ-convex. We leave it and handle the next vertex. If $D(B, AC) < 2\epsilon$, we revise P' by deleting vertex B and adding a new vertex B' which is ϵ-convex. Since the new P' must contain the old P' and each vertex of the new P' must lie within 2ϵ from P, in general, B' is not easily be found. In order to find B', we scan the vertices of P' one by one from B until we arrive vertex D such that the distance from the intersection of lines $l(A, B)$ and $l(D, E)$ to segment AD is at least 2ϵ (see Fig. 1), where E is the vertex directly after D. Then, we find point B' on line $l(A, B)$ such that the distance of B' to segment AD is equal 2ϵ. Finally, we delete the vertices between A and D in P', and add B' to P'. It is easily seen that the resulting new P' satisfies the required properties. We call this process *sweep-and-expansion.*

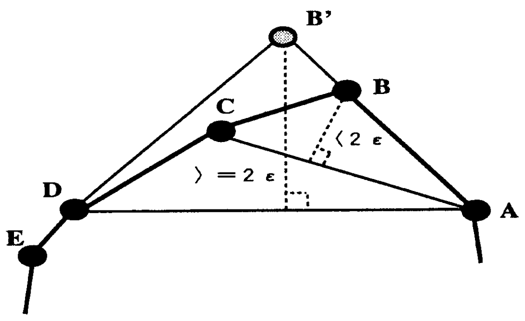

Figure 1: Expansion happened when scaning vertex B

4 Outline of The Algorithm

Algorithm
Input: Convex polygon P with n vertices represented by a sequence of vertices listing in counter-clockwise order.
Output: An ϵ-strongly convex 4ϵ-superhull of P.
method: Use the sweep-and-expansion technique. Sweep the active vertex and make the revision when necessary in counter-clockwise order. Let p_i be the active vertex and $P' = (p_1, p_2, \ldots, p_m)$ be the polygon obtained until arriving p_i (at the beginning, we let $P' = P$ and let the active vertex be the first one of P). We process p_i as follows.
(Phase I) If $p_i = p_n$, i.e., p_i is the last vertex of P, execute Phase II. otherwise, determine whether $d(p_i, p_{i-1}p_{i+1}) \geq 2\epsilon$. If $d(p_i, p_{i-1}p_{i+1}) \geq 2\epsilon$, sweep the next vertex p_{i+1}, else do the following.

Compute q, the intersection of lines $l(p_{i-1}, p_i)$ and $l(p_{i+1}, p_{i+2})$ (if $l(p_{i-1}, p_i)$ and $l(p_{i+1}, p_{i+2})$ are parallel, q is a infinite point), and compute $d(q, p_ip_{i+1})$. Determine whether $d(q, p_ip_{i+1}) \leq 0$.

If $d(q, p_ip_{i+1}) \leq 0$ (Fig. 2), find a point q' such that q' is on line $l(p_{i-1}, p_i)$ and $d(q', p_{i-1}p_{i+1}) = 2\epsilon$. Obviously, q' exists. Delete p_i from P' and add q' to P' between

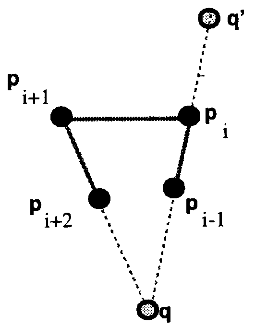

Figure 2: The intersection of lines $l(p_{i-1}, p_i)$ and $l(p_{i+1}, p_{i+2})$

p_{i-1} and p_{i+1} (this can be executed in constant time if let P' be a list linked by pointers). Then, sweep p_{i+1}.

Else (i.e., $d(q, p_i p_{i+1}) > 0$) scan the vertices of P' from one by one from p_i until we meet a vertex p_k such that the distance from the intersection of lines $l(p_{i-1}, p_i)$ and $l(p_k, p_{k+1})$ to segment $p_{i-1}p_k$ is at least 2ϵ. Let q' be the intersection of lines $l(p_{i-1}, p_i)$ and $l(p_{k-1}, p_k)$. According to whether $d(q', p_{i-1}p_k) \geq 2\epsilon$ (note that the sweep until now only guarantees $d(q', p_{k-1}p_{i-1}) < 2\epsilon$), we consider the following two cases.

(Case 1) $d(q', p_{i-1}p_k) < 2\epsilon$ (Fig. 3). Find a point w between q and q' on line $l(p_{i-1}, p_i)$ such that $d(w, p_k p_{i-1}) = 2\epsilon$. Delete the vertices p_i, p_{i+1}, $\ldots$, p_{k-1} from P', and add w between p_{i-1} and p_k to P'. Then Sweep p_k.

In new P', w is ϵ-convex and lies within 2ϵ from input P since the convex chain consisting of p_i, p_{i+1}, $\ldots$, p_{k-1} are contiguous vertices of P and w lies within 2ϵ from this chain by the method of finding w.

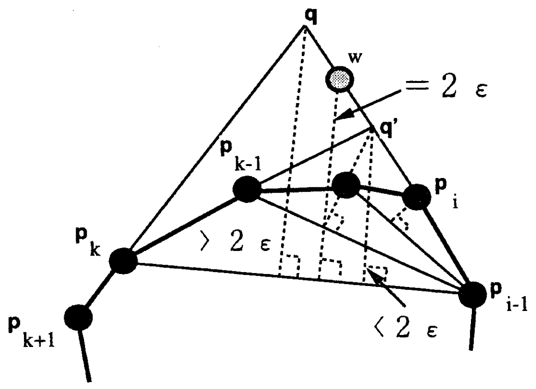

Figure 3: Expansion: case 1

(Case 2) $d(q', p_{i-1}p_k) \geq 2\epsilon$ (Fig. 4). Find point w' between p_{k-1} and p_k on line $l(p_{k-1}, p_k)$ such that the distance from the intersection w of lines $l(p_{i-1}, p_i)$ and $l(p_{k-1}, p_k)$ to $l(p_{i-1}, w')$ is eaual to 2ϵ (obviously, w' exsits). Delete the vertices p_i, p_{i+1}, $\ldots$, p_{k-1} from P', and add w and w' to P' such that p_{i-1}, w, w', p_k are listed contiguously in P'. Then Sweep w'.

It is also easily seen that in new P', w is ϵ-convex and w lies within 2ϵ from P.

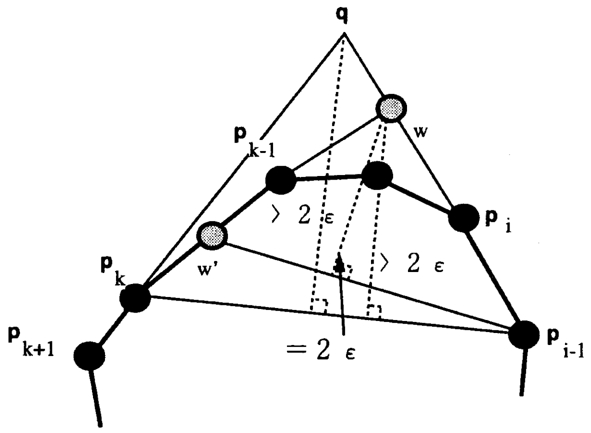

Figure 4: Expansion: case 2

(Phase II) p_i is the last vertex of P. In this case, all the vertices of P were scanned except p_i. If p_i is ϵ-convex in P', terminate the algorithm. Else do the following. Let H be a line passing through p_i and parallel to line $l(p_{i+1}, p_{i-1})$ (Fig. 5). Compute the intersection u of lines H and $l(p_{i+1}, p_{i+2})$ and the intersection v of lines H and $l(p_{i-2}, p_{i-1})$, delete p_{i-1}, p_i, and p_{i+1} from P' and add u and v to P' such that p_{i-2}, v, u, p_{i+2} are listed contiguously in P'.

It is easily seen that vertices p_{i-2}, v, u and p_{i+2} are all ϵ-convex in new P' and new vertices u and v lie within 2ϵ from P' (we omit the proof here). Note that the expansion at p_i does not affect any other veritices of P'. ∎

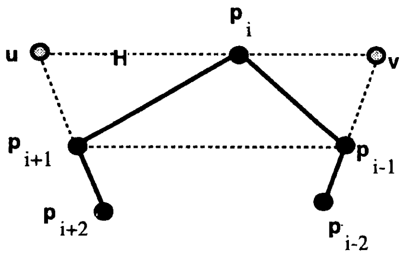

Figure 5: Expansion at the last vertex

Theorem 1 *Let P be a convex polygon with n vertices. An ϵ-strongly convex 4ϵ-superhull of P can be computed in $O(n)$ time.*

(Proof) In the algorithm, each vertex of P is scanned exactly once and finally each vertex of the resulting convex polygon is ϵ-convex. Each of Phase I and Phase II expand the polygon within 2ϵ. Therefore, the resulting convex hull is 4ϵ-superhull. It is easily seen that the number of the vertices in the resulting convex polygon is not larger than n. We omit the details of the proof here. ∎

5 Implementation And Experiment

We show that our algorithm has very good average performance. Let the size of the input (a point set) be 612. The inputs are taken from three distributions. The points of the first kind of inputs are uniformly distributed inside a square, the points of the second kind are uniformly distributed inside a disk, and the points of the third kind of the inputs are uniformly distributed on the boundary of a disk. The results of the experiments are listed in the following table, where the data of (I), (II) and (III) corresponding to the first, the second and the third distributions, respectively.

ϵ				δ			
requested	actual			theoretical	actual		
	(I)	(II)	(III)		(I)	(II)	(III)
0.0200	0.0200	0.0365	0.0213	0.0800	0.0074	0.0000	0.0070
0.0800	0.0800	0.0800	0.0891	0.3200	0.0374	0.0731	0.0300
0.3240	0.3263	0.3294	0.4565	1.2960	0.0966	0.0506	0.1350
0.5600	0.5863	0.7190	0.7249	2.2400	0.2448	0.2186	0.2379
0.9700	1.0343	1.1714	1.4130	3.8800	0.5113	0.3396	0.2748
1.6300	1.6344	2.0273	2.1700	6.5200	0.5383	0.4839	0.4040
2.0460	2.0530	2.2143	3.0188	8.1840	1.1901	0.4400	1.0460

In Section 4, we proved that our algorithm can construct an ϵ-strongly convex 4ϵ-hull. From the above table, we see that our algorithm computes an ϵ'-strongly convex δ'-superhull of S, where ϵ' is at least ϵ and δ' is much smaller that 4ϵ. Thus, our algorithm is shown to be fairly good performance.

6 Conclusion

In this paper, we introduced a new concept of strongly convex approximate hulls called *ϵ-strongly convex δ-superhull* and gave the first method for solving such a generalized convex hull problem. We showed that an ϵ-strongly convex 4ϵ-superhull of a set S of n points in the plane can be constructed in $O(n \log n)$ time for any $\epsilon \geq 0$ with exact arithmetic.

When using imprecise computation, the work is expected more difficult. First, we must find a convex approximate superhull of S and then make it to a strongly convex approximate superhull with the method of this paper. We are going to prepare it in the forth coming paper.

References

(1) A. Aggarwal, B. Chazelle, L. Guibas, C. O'Dunlaing, and C. Yap: Parallel computational geometry. *Algorithmica*, 3, 293-327, 1988.

(2) M. J. Atallah and M. T. Goodrich: Efficient parallel solutions to some geometric problems. *Parallel and Distributed Computing*, 3, 492-507, 1986.

(3) M. J. Atallah and M. T. Goodrich: Parallel algorithms for some functions of two convex polygons. *Algorithmica*, 3, 535-548, 1988.

(4) W. Chen,K. Nakano, T. Masuzawa, and N. Tokura: Optimal parallel algorithms for finding the convex hull of a sorted point set (in Japanese). *IEICE Trans. J74-D-I*, 12, 814-825, 1991.

(5) W. Chen, K. Nakano, T. Masuzawa and N.Tokura: A parallel method for the prefix convex hulls problem. *IEICE Trans. Fundamentals*, E77-A, 10, 1675-1683, 1994.

(6) W. Chen, K. Wada and K. Kawaguchi: Parallel Robust Algorithms for Constructing Strongly Convex Hulls. *Proc. of the 12th Annual ACM Symposium on Computational Geometry*, 1996.

(7) W. Chen, K. Wada and K. Kawaguchi: Robust Algorithms for Constructing Strongly Convex Superhulls (In preparation).

(8) R. Cole and M. T. Goodrich: Optimal parallel algorithms for point-set polygon problems. *Algorithmica*, vol. 7, pp.3-23, 1992.

(9) P-O. Fjällström,J. Katajainen, C. Levcopoulos and O. Petersson: A sublogarithmic convex hull algorithm. *Bit*, 30, 378-384, 1990.

(10) L. Guibas, D Salesin and J. Stolfi: Constructing strongly convex approximate hulls with inaccurate primitives. *Algorithmica*, 9, 534-560, 1993.

(11) Z. Li and V. J. Milenkovic: Constructing strongly convex hulls using exact or rounded arithmetic. *Algorithmica*, vol. 8, 345-364, 1992.