International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:7, No:5, 2013

On Constructing Approximate Convex Hull

M. Zahid Hossain, M. Ashraful Amin

Abstract—The algorithms of convex hull have been extensively studied in literature, principally because of their wide range of applications in different areas. This article presents an efficient algorithm to construct approximate convex hull from a set of n points in the plane in $O(n + k)$ time, where k is the approximation error control parameter. The proposed algorithm is suitable for applications preferred to reduce the computation time in exchange of accuracy level such as animation and interaction in computer graphics where rapid and real-time graphics rendering is indispensable.

Keywords—Convex hull, Approximation algorithm, Computational geometry, Linear time.

I. Introduction

THE construction of planar convex hull is one of the most fundamental problems in computational geometry. The applications of convex hull spread over large number of fields including pattern recognition, regression, collision detection, area estimation, spectrometry, topology, etc. For instance, computer animation, the most crucial section of computer gaming, requires fast approximation for real-time response. Consequently, it is evidential from literature that numerous studies focus on fast approximation of different geometric structures in computer graphics [1], [2]. Moreover, the construction of exact and approximate convex hull is used as a preprocessing or intermediate step to solve many problems in computer graphics [3], [4].

Convex hull for a given finite set $P \subset \mathbb{R}^d$ of n points where $\mathbb{R}^d$ denotes the d-dimensional Euclidean space, is defined as the smallest convex set that contains all the n points. A set $S \subset \mathbb{R}^d$ is convex if for two arbitrary points $a, b \in S$, the line segment $\overline{ab}$ is entirely contained in the set S. Alternatively, the convex hull can be defined as the intersection of all half-spaces (or half-planes in $\mathbb{R}^2$) containing P. The main focus of this article is limited on the convex hull in Euclidean plane $\mathbb{R}^2$.

II. Previous Work

Because of the importance of convex hull, it is natural to study for improvement of running time and storage requirements of the convex hull algorithms in different Euclidean spaces. Graham [5] published one of the fundamental algorithms of convex hull, widely known as Graham's scan as early as 1972. This is one of the earliest convex hull algorithms with $O(n \log n)$ worst-case running time. Graham's algorithm is asymptotically optimal since $\Omega(n \log n)$ is the lower bound of planar convex hull problem. It can be shown [6] that $\Omega(n \log n)$ is a lower bound of a similar but weaker problem of determining the points belonging to the convex hull, not necessarily producing them in cyclic order.

Department of Computer Science and Electrical Engineering, North South University, Bangladesh. Email: mzhossain@gmx.com
School of Engineering and Computer Science, Independent University, Bangladesh. Email: aminmdashraful@ieee.org

However, all of these lower bound arguments assume that the number of hull vertices h is at least a fraction of n. Another algorithm due to Jarvis [7] surpasses the Graham's scan algorithm if the number of hull vertices h is substantially smaller than n. This algorithm with $O(nh)$ running time is known as Jarvis's march. There is a strong relation between sorting algorithm and convex hull algorithm in the plane. Several divide-and-conquer algorithms including MergeHull and QuickHull algorithms of convex hull modeled after the sorting algorithms [8] and the first algorithm Graham's [5] scan uses explicit sorting of points.

In 1986, Kirkpatrick and Seidel [9] proposed an algorithm that computes the convex hull of a set of n points in the plane in $O(n \log h)$ time. Their algorithm is both output sensitive and worst case optimal. Later, a simplification of this algorithm [9] was obtained by Chan [10]. In the following year Melkman [11] presented a simple and elegant algorithm to construct the convex hull for simple polyline. This is one of the on-line algorithms which construct the convex hull in linear time.

Approximation algorithms for convex hull are useful for applications including area estimation of complex shapes that require rapid solutions, even at the expense of accuracy of constructed convex hull. Based on approximation output, these algorithms of convex hull could be divided into three groups – near, inner, and outer approximation algorithms. Near, inner, and outer approximation algorithms compute near, inner, and outer approximation of the exact convex hull for some point set respectively.

In 1982, Bentley et al. [12] published an approximation algorithm for convex hull construction with $O(n + k)$ running time. Another algorithm due to Soisalon-Soininen [13] which uses a modified approximation scheme of [12] and has the same running time and error bound. Both of the algorithms are the inner approximation of convex hull algorithm. The proposed algorithm in this article is a near approximation algorithm of $O(n + k)$ running time.

III. Approximation Algorithm

Let $P \subset \mathbb{R}^2$ be the finite set of $n \geq 3$ points in general position and the (accurate) convex hull of P be $CH(P)$. Kavan, Kolingerova, and Zara [14] proposed an algorithm with $O(n + k^2)$ running time which partitions the plane $\mathbb{R}^2$ into k sectors centered in the origin. Their algorithm requires the origin to be inside the convex hull. (It is possible to choose a point $p \in P$ and translate all the other points of P accordingly using additional steps in their algorithm). Conversely, we partition the plane $\mathbb{R}^2$ into k vertical sector pair with equal central angle α in the origin and for our algorithm the origin O could be located outside of the convex hull. The sets represent the vertically opposite sectors that form the vertical sector pairs

defined as

$$S_i^{\oplus} = \{p \in \mathbb{R}^2 : \text{atan2}(p) \in [\alpha i, \alpha(i+1))\}$$
$$S_i^{\ominus} = \{p \in \mathbb{R}^2 : \text{atan2}(p) \in [\pi + \alpha i, \pi + \alpha(i+1))\}$$

where, $i = 0, 1, \ldots, k-1$ and the central angle $\alpha = \pi/k$. Then, the sets $s_i^{\oplus}$ and $s_i^{\ominus}$ denote the points belonging to the set P in sectors $S_i^{\oplus}$ and $S_i^{\ominus}$ respectively. Formally,

$$s_i^{\oplus} = S_i^{\oplus} \cap P$$
$$s_i^{\ominus} = S_i^{\ominus} \cap P$$

A pair of unit vectors $u_i^{\oplus}$ and $u_i^{\ominus}$ obtain in ith vertical sector pair as

$$u_i^{\oplus} = \left(\cos\left(\alpha i + \frac{\alpha}{2}\right), \sin\left(\alpha i + \frac{\alpha}{2}\right)\right)$$
$$u_i^{\ominus} = \left(-\cos\left(\alpha i + \frac{\alpha}{2}\right), -\sin\left(\alpha i + \frac{\alpha}{2}\right)\right)$$

The maximum projection magnitudes in the directions of $u_i^{\oplus}$ and $u_i^{\ominus}$ are

$$m_i^{\oplus} = \max_{p \in s_i^{\oplus}} \langle u_i^{\oplus}, p \rangle$$
$$m_i^{\ominus} = \max_{p \in s_i^{\ominus}} \langle u_i^{\ominus}, p \rangle$$

The definition of *max* function is extend to return $-\infty$ for

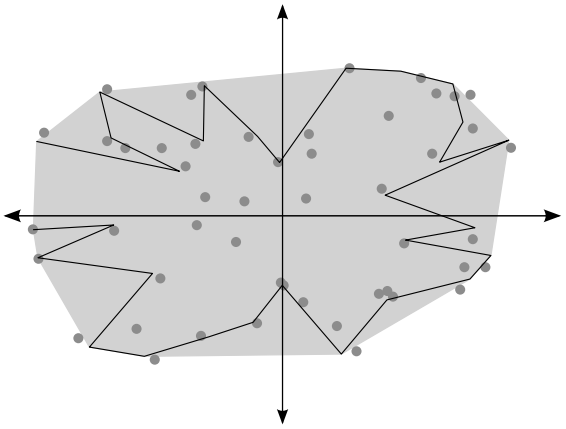

Fig. 1. An example of convex hull CH_{16} constructed using proposed algorithm based on a set of 50 points.

no parameter. The sets of points which provide the maximum projection magnitude in the sectors of ith vertical sector pair are

$$M_i^{\oplus} = \{p \in s_i^{\oplus} : \langle u_i^{\oplus}, p \rangle = m_i^{\oplus}\}$$
$$M_i^{\ominus} = \{p \in s_i^{\ominus} : \langle u_i^{\ominus}, p \rangle = m_i^{\ominus}\}$$

The vectors that produce the maximum magnitude in the directions of $u_i^{\oplus}$ and $u_i^{\ominus}$ for some points in the ith vertical sector pair are

$$v_i^{\oplus} = \max\left(\max_{p \in (s_i^{\ominus} - M_i^{\ominus})} \langle u_i^{\oplus}, p \rangle, m_i^{\oplus}\right) u_i^{\oplus}$$
$$v_i^{\ominus} = \max\left(\max_{p \in (s_i^{\oplus} - M_i^{\oplus})} \langle u_i^{\ominus}, p \rangle, m_i^{\ominus}\right) u_i^{\ominus}$$

The magnitude of the vectors $v_i^{\oplus}$ and $v_i^{\ominus}$ could be $\pm\infty$ for the ith vertical sector pair containing less than two points. The sets $V^{\oplus}$ and $V^{\ominus}$ containing all the finite vectors in the ranges $[0, \pi)$ and $[\pi, 2\pi)$, are

$$V^{\oplus} = \left\{v_i^{\oplus} : \|v_i^{\oplus}\| \neq \infty\right\}_{i=0}^{k-1}$$
$$V^{\ominus} = \left\{v_i^{\ominus} : \|v_i^{\ominus}\| \neq \infty\right\}_{i=0}^{k-1}$$

Let, $V = V^{\oplus} \cup V^{\ominus}$ and V contains at least three terminal points of the vectors in general position to construct the convex hull. The convex hull approximation of k vertical sector pairs according to the proposed algorithm in this article is

$$CH_k(P) = CH\left\{(w_x, w_y) \in \mathbb{R}^2 : w \in V\right\}$$

IV. Implementation

The input of the algorithm $P \subset \mathbb{R}^2$ is a set of $n \geq 3$ points in general position. For simplicity, we assume that the origin $O \notin P$ and $k \geq 2$. (This assumption can be achieved by taking a point arbitrarily close to the origin instead of the origin itself, within the upper bound of error calculated in Section V).

APPROXIMATE-CONVEX-HULL(P, k)
01. $\alpha \leftarrow \pi/k$
02. **for** $i \leftarrow 0$ **to** $k-1$
03. $U_i \leftarrow (\cos(\alpha i + \alpha/2), \sin(\alpha i + \alpha/2))$
04. $U_{i+k} \leftarrow -U_i$
05. $M_{i+k} \leftarrow M_i \leftarrow -\infty$
06. **for each** $p \in P$ **do**
07. $i \leftarrow \lfloor \text{atan2}(p)/\alpha \rfloor$
08. $t \leftarrow \langle U_i, p \rangle$
09. **if** $M_i \leq t$ **then** $M_i \leftarrow t$
10. **else** $M_{i+k} \leftarrow \max(M_{i+k}, -t)$
11. $V \leftarrow \langle\rangle$
12. $f \leftarrow \text{anglex}(M)$
13. **for** $i \leftarrow f$ **to** $f + 2k - 1$
14. **if** $M_{i+k} \in (-\infty, 0)$ **then** $V \leftarrow V \cup \langle M_{i+k} U_{i+k} \rangle$
15. **if** $M_i \in (0, \infty)$ **then** $V \leftarrow V \cup \langle M_i U_i \rangle$
16. **return** MELKMAN-CONVEX-HULL(V)

Fig. 2. The proposed algorithm to compute an approximate convex hull in $O(n+k)$ time from inputs P and k where $P \subset \mathbb{R}^2$ is a set of n points in the plane and k is the number of vertical sector pair partitioning the plane.

We also assume that at least two vertical sector pairs together contains minimum three points (where none of these two are empty). The assumption can be reduced to one of the requirements of minimum three points input (i.e., $|P| \geq 3$) of convex hull. To illustrate that, let us consider p and q to be two points in P such that $\angle pOq \leq \pi - \alpha$ where O is the origin. Such two points do exist if no three points are collinear in P (i.e., the points of P are in general position). If Ot is the bisector of $\angle pOq$, then adding the angle of Ot from positive x – axis as an offset to every vertical sector pair ensures that all the input points cannot be in the same vertical sector pair. Thus, the assumption is satisfied. Alternatively, if less than three absolute values in M are finite, then for each $M_i \in M$, assign $M_i \cos\alpha$ to M_{i-1} and M_{i+1} where these are infinite.

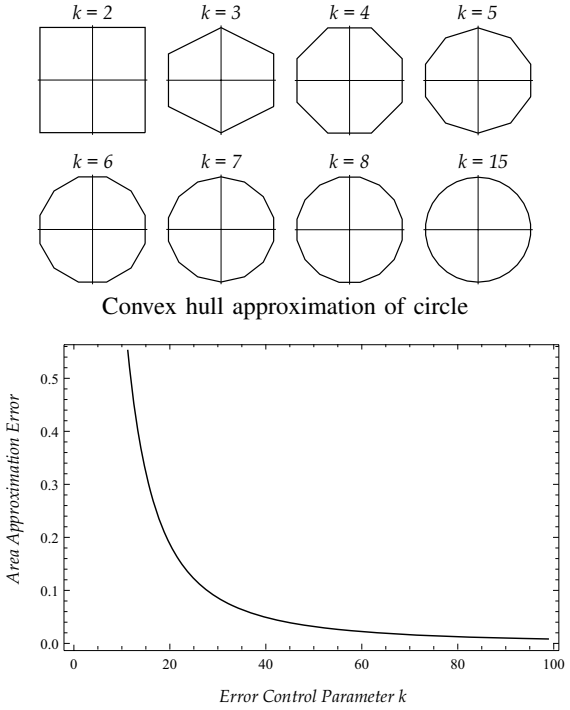

Fig. 3. The graph showing approximation error of a circle area with respect to error control parameter k where the number of input points is $n = 16 \times 10^3$ lying on a circle of radius 4 units.

(The next paragraph contains details about M.) Therefore, the number of points in V must be at least three.

A circular array U is used to contain the k pairs of unit vectors of all the k vertical sector pairs and another circular array M is used to hold the number of k pairs of maximum projection magnitude in all the k vertical sector pairs. Both circular arrays have the same size of $2k$ and use zero based indexing scheme. The function *atan2* is a variation of function *arctan* with point as a parameter. The function returns the angle in radians between the point and the positive x – axis of the plane in the range of $[0, 2\pi)$. The function *anglex* searches sequentially for the index of maximum angular distance between two consecutive positive finite vectors (computed using projection magnitude with index referring angle). If the index is i such that maximum angle occurs in between i and j, the *anglex* function returns j. The final convex hull is constructed using Melkman's [11] algorithm from set of V points which are the terminal points of finite vectors computed in steps 14 and 15. If the first three points of V are collinear, displacing one of these points within the error bound solves the problem.

Since the vertices of the convex hull produced by the proposed algorithm are not necessarily in the input point set P, the algorithm cannot be applied straight away to solve some other problems. Let us consider another circular array Q of $2k$ size which used to contain the points generating the inner products of M. Adding the point Q_j instead of M_jU_j to the sequence V in steps 14 and 15 ensures that the vertices of the convex hull will be the points from P. These modifications of the algorithm allow us to solve some problems including approximate farthest-pair problem but increase the upper bound of error (described in Section V) to $r \sin(\pi/k)$.

V. Error Analysis

There are different schemes for measuring the error of an approximation of the convex hull. We measured the error as distance from point set of accurate convex hull $CH(P)$. The distance of an arbitrary point x from a set S is defined as

$$\text{dist}(x, S) = \inf\{\|x - y\| : y \in S\}$$

Formally the approximation error E can be defined as

$$E = \sup\{\text{dist}(p, CH(P)) : p \in CH_k(P)\}$$

It is sufficient to determine the upper bound of error E of

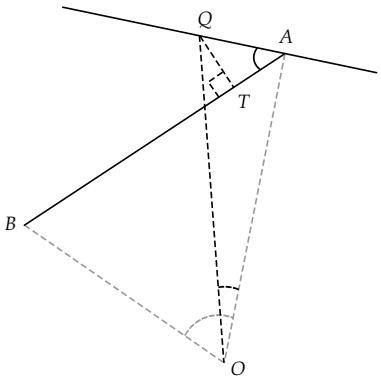

Fig. 4. The approximation error of the proposed algorithm measured as a distance $\overline{TQ}$ of the point Q lying outside of the approximate convex hull with an edge $\overline{AB}$.

the approximate convex hull $CH_k(P)$. Let, Q be be a point lying outside of the convex hull $CH_k(P)$ and O be the origin. Suppose that, $\overline{AB}$ is an edge of the approximate convex hull (as shown in Figure 4). Therefore, the distance of the point Q from the $CH_k(P)$ is

$$TQ = AQ \ \sin\angle TAQ$$

The distance of the point Q from vertex A is $AQ = OQ \ \sin\angle AOQ$. Thus,

$$TQ = OQ \ \sin\angle AOQ \ \sin\angle TAQ$$

Let, $d = TQ$ and $r = \max_{p \in P} \|p - O\|$. Thus we obtain

$$\begin{aligned} d &= OQ \ \sin\angle TAQ \ \sin\angle AOQ \\ &\leq OQ \ \sin\frac{\pi}{2} \ \sin\angle AOQ \\ &\leq r \sin\frac{\pi}{2k} \end{aligned}$$

It follows that the minimum distance d directly depends on k which is denoted as function $d(k)$. Thus, the upper bound of approximation error E is $r \sin(\pi/2k)$. If k approaches to infinity, the $CH_k(P)$ converges to $CH(P)$.

$$\lim_{k \to \infty} d(k) \leq \lim_{k \to \infty} r \ \sin\frac{\pi}{2k} = 0$$

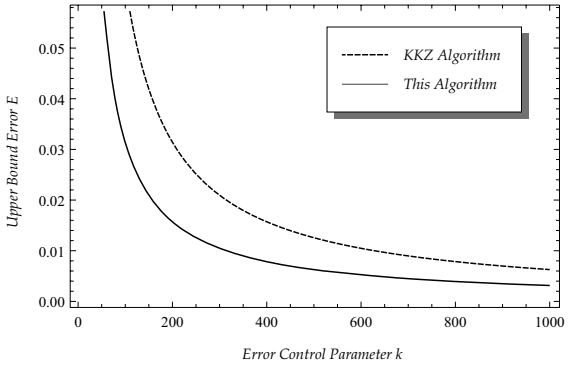

Fig. 5. The graph representing the relation between the error control parameter k and upper bound of error E. The upper error bounds $r\sin(\pi/k)$ and $\max(r\tan(\pi/k), 2r\sin(\pi/k))$ are calculated in this article and in [14] (i.e., KKZ Algorithm) respectively where r is unit in the graph.

VI. Correctness

Theorem 1: The approximation algorithm produces the convex hull from a set of points in $\mathbb{R}^2$ correctly within the prescribed error bound.

Proof: Since, Melkman's algorithm can construct the convex hull correctly for points on a simple polygonal chain, it suffices to prove that the sequence of points V denotes a simple polygonal chain. (Melkman [11] published the on-line algorithm of convex hull with formal proof of correctness in 1987). Suppose that, the plane $\mathbb{R}^2$ is partitioned into

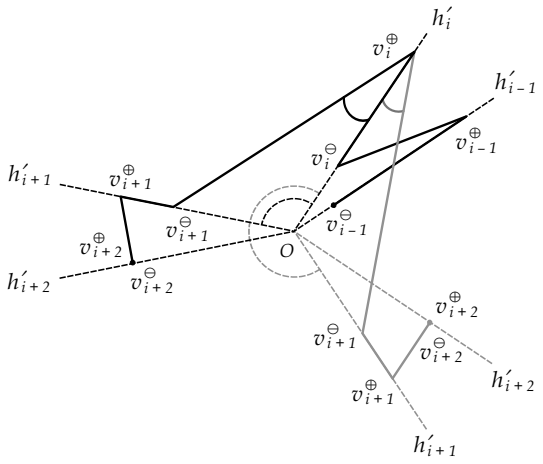

Fig. 6. The proof of correctness of the algorithm that consider both the simple and non-simple variation of polygonal chain $v_{i-1}^{\ominus}v_{i-1}^{\oplus}v_i^{\ominus}v_i^{\oplus}v_{i+1}^{\ominus}v_{i+1}^{\oplus}v_{i+2}^{\ominus}v_{i+2}^{\oplus}$ with $\angle h'_i h'_{i+1} < \pi$ and $\angle h'_i h'_{i+1} \geq \pi$.

k vertical sector pairs which correspond to the sequence $S = \langle s_0, s_1, \ldots, s_{2k-1}\rangle$ of $2k$ simple sectors. The sequence S of sectors is ordered according to the angle measured anticlockwise. If h_i is a half-line (denoting the set of points on the half-line) from the origin in the direction of the unit vector of the sector s_i, then the sequence $H = \langle h_0, h_1, \ldots, h_{2k-1}\rangle$ represents all the half-lines correlated with the sequence S. According to the algorithm all the points of V must be distinct (as referred in steps 9 – 10) and lying on some of the half-lines of H. The sequence of half-lines $H' \subseteq H$ where each contained at least one point from V, is

$$H' = \langle h'_i \in H : h'_i \cap V \neq \emptyset\rangle$$

Each half-line $h'_i \in H'$ can contain at most two points of V. Let, $v_i^{\ominus}, v_i^{\oplus} \in (h'_i \cap V)$ are points on each half-line h'_i such that $\|v_i^{\ominus} - O\| \leq \|v_i^{\oplus} - O\|$. If a half-line h'_i contains only one point of V, the length of virtual $\overline{v_i^{\ominus}v_i^{\oplus}}$ is zero with $v_i^{\ominus}$ and $v_i^{\oplus}$ refer to the same point of V (e.g., h'_{i+2} contains only one point in the Figure 6). Let $\angle h_i h_j$ denotes the angle from h_i to h_j where h_i and h_j are half-lines from the origin. Since the angle between two consecutive half-lines $\angle h'_i h'_{i+1} \geq \pi/k$ and $O \notin V$ (because $t > 0$ for our assumptions $O \notin P$ and $k \geq 2$ in the algorithm), no two line segments $\overline{v_i^{\ominus}v_i^{\oplus}}$ and $\overline{v_j^{\ominus}v_j^{\oplus}}$ intersect each other, for all $i \neq j$. However, the line segment $\overline{v_i^{\oplus}v_{i+1}^{\ominus}}$ could cross the polygonal chain $v_0^{\ominus}v_0^{\oplus}v_1^{\ominus}v_1^{\oplus}\ldots v_i^{\ominus}v_i^{\oplus}$ if the angle $\angle h_i h_{i+1} \geq \pi$. The equation of $\angle O v_i^{\oplus} v_{i+1}^{\ominus}$ (derived using the law of sines and basic properties of triangle) also illustrates this fact mathematically for $\triangle O v_i^{\oplus} v_{i+1}^{\ominus}$ (as shown in the Figure 6)

$$\angle O v_i^{\oplus} v_{i+1}^{\ominus} = \operatorname{arccot}\left(\frac{Ov_i^{\oplus}}{Ov_{i+1}^{\ominus}\ \sin\angle v_i^{\oplus} O v_{i+1}^{\ominus}} - \cot\angle v_i^{\oplus} O v_{i+1}^{\ominus}\right)$$

The solution with minimum magnitude of the above equation is negative for $\pi < \angle v_i^{\oplus} O v_{i+1}^{\ominus} \leq 2\pi$, even if $Ov_i^{\oplus} > Ov_{i+1}^{\ominus}$. Thus the line segment $\overline{v_i^{\oplus}v_{i+1}^{\ominus}}$ could intersect with the edges of the polygonal chain only if $\angle v_i^{\oplus} O v_{i+1}^{\ominus} \geq \pi$. If the maximum angle between two consecutive half-lines is $\angle h'_i h'_{i+1}$ for some i, then *anglex* function returns the index $i+1$ that ensures the construction a simple polygonal chain $v_{i+1}^{\ominus}v_{i+1}^{\oplus}v_{i+2}^{\ominus}v_{i+2}^{\oplus}\ldots v_{i+m}^{\ominus}v_{i+m}^{\oplus}$ where $m = |H'|$ and all the indices are modulo m. Thus the sequence of points V represents a simple polygonal chain. (It is possible to prove the algorithm obtained by interchanging the steps 14 and 15, using a similar method). ■

Theorem 2: If n is the number of input points and k is the number of vertical sector pairs in $\mathbb{R}^2$, then the running time of the proposed algorithm is $O(n+k)$.

Proof: Let us estimate the running time for each part of the algorithm to prove that the algorithm compute the approximate convex hull in $O(n+k)$ time. It is clear that, the initialization steps 2 – 5 take $O(k)$ time. Since, the next loop of steps 6 – 10 iterates for each point $p \in P$, thus it takes $O(n)$ time considering constant time for floor function. According to the description of *anglex* function in Section IV, the function can be implemented in $O(k)$ time because it requires $2k$ iterations to compute the index. The loop of steps 13 – 15 takes $O(k)$ time and Melkman's [11] algorithm runs in linear time. Steps 1 and 11 require constant time. Thus the running time of the algorithm is $O(n+k)$. ■

VII. Conclusion

Geometric algorithms are frequently formulated under the non-degeneracy assumption or general position assumption [15] and the proposed algorithm in this article is also not an exception. To make the implementation of the algorithm

robust an integrated treatment for the special cases can be applied. There are other general techniques called perturbation schemes [16], [17] to transform the input into general position and allow the algorithm to solve the problem on perturbed input. Both symbolic perturbation and numerical (approximation) perturbation (where perturbation error is consistent with the error bound of the algorithm) can be used on the points of P to eliminate degenerate cases.

APPENDIX

The article describes a near approximation algorithm for convex hull however it is possible to extend the concept for inner as well as outer approximation algorithms for convex hull. An illustration of inner approximate convex hull algorithm is shown in Figure 7.

INNER-APPROXIMATE-CONVEX-HULL(P, k)

01. $\alpha \leftarrow \pi/k$
02. **for** $i \leftarrow 0$ **to** $k-1$
03. $\quad U_i \leftarrow (\cos(\alpha i + \alpha/2), \sin(\alpha i + \alpha/2))$
04. $\quad U_{i+k} \leftarrow -U_i$
05. $\quad Q_{i+k} \leftarrow Q_i \leftarrow (0,0)$
06. $\quad M_{i+k} \leftarrow M_i \leftarrow -\infty$
07. **for each** $p \in P$ **do**
08. $\quad i \leftarrow \lfloor \text{atan2}(p)/\alpha \rfloor$
09. $\quad t \leftarrow \langle U_i, p \rangle$
10. $\quad$ **if** $M_i < t$ **then** $(Q_i, M_i) \leftarrow (p, t)$
11. $\quad$ **elseif** $M_{i+k} < -t$ **then** $(Q_{i+k}, M_{i+k}) \leftarrow (p, -t)$
12. $V \leftarrow \langle\rangle$
13. $f \leftarrow \text{anglex}(M)$
14. **for** $i \leftarrow f$ **to** $f + 2k - 1$
15. $\quad$ **if** $M_i \in (0, \infty)$ **then** $T \leftarrow \langle Q_i \rangle$ **else** $T \leftarrow \langle\rangle$
16. $\quad$ **if** $M_{i+k} \in (-\infty, 0)$ **then** $T \leftarrow T \cup \langle Q_{i+k} \rangle$
17. $\quad V \leftarrow V \cup \text{sort}(T)$
18. **return** MELKMAN-CONVEX-HULL(V)

Fig. 7. The proposed algorithm to compute an inner approximate convex hull in $O(n+k)$ time from inputs P and k where $P \subset \mathbb{R}^2$ is a set of n points in the plane and k is the number of vertical sector pair partitioning the plane.

REFERENCES

[1] R. Bellman, B. Kashef, and R. Vasudevan, "Mean square spline approximation," *Journal of Mathematical Analysis and Applications*, vol. 45, no. 1, pp. 47–53, 1974.
[2] J. Hu and H. Yan, "Polygonal approximation of digital curves based on the principles of perceptual organization," *Pattern Recognition*, vol. 30, no. 5, pp. 701–718, 2006.
[3] J. T. Klosowski, M. Held, J. S. B. Mitchell, H. Sowizral, and K. Zikan, "Efficient collision detection using bounding volume hierarchies of k-dops," *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, pp. 21–36, January 1998.
[4] X. Zhang, Z. Tang, J. Yu, and M. Guo, "A fast convex hull algorithm for binary image," *Informatica (Slovenia)*, vol. 34, no. 3, pp. 369–376, 2010.
[5] R. L. Graham, "An efficient algorithm for determining the convex hull of a finite planar set," *Information Processing Letters*, vol. 1, no. 4, pp. 132–133, 1972.
[6] A. C.-C. Yao, "A lower bound to finding convex hulls," *J. ACM*, vol. 28, pp. 780–787, October 1981.
[7] R. A. Jarvis, "On the identification of the convex hull of a finite set of points in the plane," *Information Processing Letters*, vol. 2, no. 1, pp. 18–21, 1973.
[8] F. P. Preparata and M. I. Shamos, *Computational geometry: an introduction*. New York, NY, USA: Springer-Verlag New York, Inc., 1985.
[9] D. G. Kirkpatrick and R. Seidel, "The ultimate planar convex hull algorithm," *SIAM Journal on Computing*, vol. 15, pp. 287–299, February 1986.
[10] T. M. Chan, "Optimal output-sensitive convex hull algorithms in two and three dimensions," *Discrete & Computational Geometry*, vol. 16, no. 4, pp. 361–368, 1996.
[11] A. A. Melkman, "On-line construction of the convex hull of a simple polyline," *Information Processing Letters*, vol. 25, pp. 11–12, April 1987.
[12] J. L. Bentley, F. P. Preparata, and M. G. Faust, "Approximation algorithms for convex hulls," *Communications of the ACM*, vol. 25, pp. 64–68, January 1982.
[13] E. Soisalon-Soininen, "On computing approximate convex hulls," *Information Processing Letters*, vol. 16, no. 3, pp. 121–126, 1983.
[14] L. Kavan, I. Kolingerova, and J. Zara, "Fast approximation of convex hull," in *Proceedings of the 2nd IASTED international conference on Advances in computer science and technology*. Anaheim, CA, USA: ACTA Press, 2006, pp. 101–104.
[15] J. O'Rourke, *Computational geometry in C (2nd ed.)*. New York, NY, USA: Cambridge University Press, 1998.
[16] H. Edelsbrunner and E. P. Mücke, "Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms," *ACM Transactions on Graphics*, vol. 9, pp. 66–104, January 1990.
[17] I. Z. Emiris, J. F. Canny, and R. Seidel, "Efficient perturbations for handling geometric degeneracies." *Algorithmica*, vol. 19, no. 1/2, pp. 219–242, 1997.