4 H − 1

Finding the Envelope of Segments in Parallel

Wei Chen Koichi Wada
Department of Electrical and Computer Engineering
Nagoya Institute of Technology, Showa, Nagoya 466
E-mail: (chen, wada)@elcom.nitech.ac.jp

Abstract

We present parallel methods for finding the upper envelope of n possibly-intersecting planar line segments in the EREW PRAM. We show that the envelope can be computed in $O(\log^{1+\epsilon} n)$ time using $O(n/\log^{\epsilon} n)$ processors for any positive constant ϵ, or in $O(\log n \log\log n)$ time using $O(n)$ processors. The first result achieves cost optimal and the second runs faster. We use a technique called multi-divide-prune-conquer in our algorithms.

1 Introduction

Let S be a set of n segments. Considering the segments as opaque barriers, the upper envelope of S, denoted as $UE(S)$, is made of the portions of the segments visible from $y = +\infty$. The upper envelope of segments is an important concept in visibility and motion planning and has many applications [4, 2, 11]. The complexity of the upper envelope is the number of distinct pieces of segments that appear on it. If the segments are nonintersecting, the complexity of $UE(S)$ is linear in n. In this case, $UE(S)$ can be computed optimally in $O(n \log n)$ time and it acturally can be found in $O(n)$ time if the segments endpoints are sorted in the left-to-right order [2]. Generally, the complexity of $UE(S)$ is $\Theta(n\alpha(n))$, where $\alpha(n)$ is the extremely slowly growing functional inverse of Ackermann's function [3, 8]. Atallah [3], Hart and Sharir [8] presented an algorithm that computes $UE(S)$ in $O(n\alpha(n)\log n)$ time. Hershberger improved their result and gave an optimal $O(n \log n)$ time algorithm [9]. His method can be easily parallelized to a cost optimal algorithm which runs in $O(\log^2 n)$ time and $n/\log n$ processors in the EREW PRAM (the PRAM is a synchronous parallel computational model employing a number of processors which share a common memory, and the EREW PRAM is the weakest version of the PRAM in which neither concurrent reading nor concurrent writing is allowed). Although related researches are continued [7, 10], but no much progress is made in the PRAM (It was claimed that the envelope can be found in $O(\log n)$ using n pocessors in CREW PRAM, the stronger version of the PRAM in which concurrent reading is allowed but concurrent writing is not, but the paper was withdrawn later [5]). On the other hand, if $O(n^{1+k})$ processors are available for any positive constant k, $UE(S)$ can be computed in $O(\log n)$ time by ordinary n^c-way divide-and-conquer (c is any constant with $c \leq k$) easily. Therefore, it is interesting to design algorithms in the EEW PRAM which computes $UE(S)$ in $o(\log^2 n)$ time using $O(n)$ or $o(n)$ processors.

We present two efficient algorithms for computing the envelope of segments in the EREW PRAM. It is easily to see that if we have at most n processors, ordinary d-way divide-and-conquer which usually gives nice parallel algorithms is difficult to be used directly for sloving the envelope problem of segments in parallel. The segments can not be partitioned easily: if we divide the segments into subsets without cutting segments, the segments in different subsets may intersect each other, and if we cut the segments with vertical lines such that the subsets are separated each other, the number of the segments (subsegments) may become very large. Our algorithms are based on a complex divide-and-conquer which we call *multi-level divide-and-conquer* and a prune technique which are used to to partition the segments efficiently and prevent the total number of the segments become larger and larger in recursive steps.

2 Cost optimal algorithm

Let $L : y = x$ be a vertical line and S' be a set of segments. L is *a left (or right) base* of S' if the left (or right) endpoint of each segment of S' is on line L. If S' has a left (right) base, the envelope of S' contains at most $2n - 1$ segments[12].

Lemma 1 *[9] (1) The envelope of any n segments can be computed in $O(\log^2 n)$ time using $O(n/\log n)$ processors. (2) If the envelope of n segments with a left (right) base can be computed in $O(\log n \log\log n)$ time using $n/\log\log n$ processors, then the envelope of any n segments can be found in the same complexity.* ∎

We use *multi-level divide-and-conquer* which was first introduced in [6] to compute the envelope of segments with a left (right) base as follows.

First recursive step: Divide the set S of n segments into δ subsets such that each subset contains n/δ segments, and then recursively construct the upper envelope of each subset in parallel. Note that the resulting upper envelopes may intersect with each other.

Second recursive step: Using $n/\delta - 1$ vertical lines to partition the δ envelopes obtained in the first recursive step into n/δ separated parts, and then recursively find the upper envelope of each part in parallel.

Merge step: Concatenate the n/δ separated upper envelopes obtained in the second recursive step from left to right. ∎

Theorem 1 *The envelope of n segments can be computed in $O(\log^{1+\epsilon} n)$ time using $O(n/\log^{\epsilon} n)$ processors for any positive constant ϵ on the EREW PRAM.* ∎

3 Faster algorithms

Let S be a set of n segments. A set $L(S)$ is said *induced from* S if $L(S) = \{l \mid l$ is a straight line containing at least one segment of $S\}$. The envelope of $L(S)$ can be found easily by using the following dual transformation f: segment $y = ax + b \overset{f}{\leftrightarrow}$ point (a, b). Let $D(L(S))$ be the dual set of $L(S)$. $UE(L(S))$ can be found in $O(\log n)$ time using $O(n)$ processors in the EREW PRAM since the convex hull of $D(L(S))$ can be computed in the same complexity [1]. Let $S(L_1, L_2)$ denote the portion of S between vertical lines $L_1 : y = x_1$ and $L_2 : y = x_2$. For each segment s of $S(L_1, L_2)$, let $o(s)$ denote the original segment of S contains s. The following lemma reveals the relation between the upper envelopes of lines and segments.

Lemma 2 *Let S be a set of n segments. If S has a left base $L_1 : y = x_1$ and a right base L_2: $y = x_2$ ($x_1 < x_2$), then $UE(S) = UE(L(S))(L_1, L_2)$.* ∎

Given lines L_1 and L_2, set $S' = S\ (L_1, L_2)$ can be divide into two subsets: increasing-segment set $I(S')$ and original-segment set $O(S')$, where for each segment $s \in I(S')$, then original segment $o(s)$ intersects with both L_1 and L_2, and for each segment $t \in O(S')$, the original segment $o(t)$ intersects at most one of L_1 and L_2. Since $UE(S') = UE(UE(I(S')) \cup UE(O(S')))$, the following lemma 3 holds from Lemma 2.

Lemma 3 *Let S be a set of segments, L_1 and L_2 be two vertical lines and $S' = S(L_1, L_2)$. $UE(S') = UE(UE(L(I(S')))\ (L_1, L_2) \cup UE(O(S'))$ holds.* ∎

According to Lemma 3, when we recursively compute $UE(S')$, we first compute $UE(L(I(S')))\ (L_1, L_2)$ directly by the envelope algorithm of lines, and then recursively compute $UE(O(S'))$. This lets us prune $I(S')$ from S' in recursive computing. Using the prune technique to the previous algorithm, we can get the following result.

Theorem 2 *The upper envelop of n segments can be found in $O(\log n \log\log n)$ time using n processors in the EREW PRAM.* ∎

References

[1] A. Aggarwal, B. Chazelle, L. Guibas, C. O'Dunlaing and C. Yap, "Parallel computational geometry," *Algorithmica*, No.3, 293-327 (1988)

[2] Ta. Asano, Te. Asano, L. Guibas, H, Hershberger and H. Imai, "Visibility of disjoint polygons," *Algorithmica*, No.1, 49-63 (1986)

[3] M. Atallah, "Some dynamic computational geometry problems,," *Comput. Math. Appl.* Vol.11, No.12, 1171-1181 (1985)

[4] L. Boxer and R. Miller, "Common intersections of polygons," *Information Processing Letters*, Vol.33, No.5, 249-254 (1988).

[5] L. Boxer and R. Miller, "Common intersections of polygons," *Information Processing Letters*, Vol.35, No.1, pp.53 (1990).

[6] W. Chen, K. Wada and K. Kawaguchi, "A parallel method for finding the convex hull of discs," *IEEE First International Conference on Algorithm and Architectures for Parallel Processing*, 1995, pp. 274-281.

[7] F. Dehne, C. Kenyon and A. Fabri, "Scalable and architecture independent parallel geometric algorithms with high probability optimal algorithm," *Proc. Sixth IEEE Symposium on Parallel and Distributed Processing*, 586-593 (1994).

[8] D. Harel and M. Sharir, "Nonlinearity of Davenport-Schinzel sequences and of generalized path compression schemes," *Combinatorica*, No.6, 151-177, (1986), 181-185.

[9] J. Hershberger, "Finding the upper envelope of n line segments in $O(n \log n)$ time," *Information Processing Letters*, Vol.33, No.4, 169-174 (1989).

[10] J. Hershberger, "Upper envelope onion peeling," *Comput. Geom. Theory Appl.*, Vol.2, No.2, 93-110 (1989).

[11] K. Kedem. and M. Sharir, "An efficient motion-planning algorithm for a convex polygonal object in two-dimensional polygonal space" *Discrete Comput. Geometry*, vol.5, no.1, 43-755, (1990).

[12] D. Rappaport, "A convex hull algorithm for discs, and applications," *Computational Geometry: Theory and Applications*. 2 (1992) 171-187.

[13] M. Sharir and P. K. Agarwal, *Davenport-Schinzel Sequences and Their Geometric Applications* (Cambridge University Press, 1995).