New Results on Polygonal Chain Simplification with Small Angle Constraints

Ovidiu Daescu* Anastasia Kurdia†

1 Introduction

We consider a problem of polygonal chain simplification: Given a n-vertex polygonal chain in $\mathbb{R}^3$, find another chain with a minimum number of links that stays within ε-tolerance zone ($\varepsilon > 0$) of the original chain and such that the turn angle between any two consecutive links of the new chain is at most δ (*min-#* problem with maximum turn angle constraint). It was introduced in [1] and solved there for $\delta \in [\pi/2, \pi)$ in $O(n^2 \log n)$ time, $O(n^2)$ space (these bounds match the bounds for the unconstrained *min-#* problem). In [2] we gave a solution for $\delta \in [0, \pi/2)$ that takes $O(n^2 \log^2 n)$ time and $O(n^2)$ space and requires solving n 2D-instances of the so called OLBIS problem:

OLBIS (Off-Line Ball Inclusion Search) Problem: Given a sequence $\mathcal{E} = (e_1, e_2, \ldots, e_n)$ such that each e_i is either a ball B_i or a point p_i, for every point p_k find the smallest-index ball $B_j \in \{e_1, e_2, \ldots e_{k-1}\}$ such that $p_k \in B_j$, or report no such ball exists.

Our soulution [2] for the general 2D-OLBIS problem has $O(n \log n)$ preprocessing and $O(\log^2 n)$ query time. The instances of the 2D-OLBIS that arise in the chain simplification problem have a particular structure: the input sequence $\mathcal{E} = (D, P)$ consists of a set of disks $D = \{D_1, D_2, \ldots, D_n\}$ of equal radius r, followed by a set of query points P. In this work solve such a special instance of the OLBIS problem with $O(n \log n)$ preprocessing and $O(\log n)$ query time. As a result, the *min-#* problem with maximum turn angle constraint can be solved in $O(n^2 \log n)$ time for $\delta \in [0, \pi/2)$, matching the bounds for $\delta \in [\pi/2, \pi)$ and for the unconstrained min-# problem.

2 Special instance of OLBIS

An orthogonal grid with step r is constructed on the plane. Any input point p can only be in one, two or four cells of the grid, so only the disks that intersect the cell(s) containing p are the candidates to be the OLBIS answer for p. For each of the cells intersected by a disk, we construct a constant number of subdivisions of the cell into indexed regions.

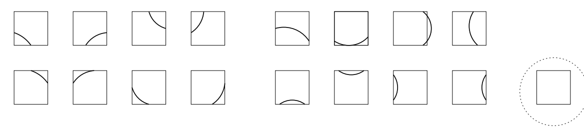

Figure 1: Possible intersections of a square cell and a disk.

The OLBIS answer for each input query point p is then obtained in two steps: first, a cell containing p is found, then p is located in each subdivision associated with that cell and the minimum of indices of the regions containing p is returned as an OLBIS answer.

In the remainder of this paper we assume the query points and input disks are all distinct and each input point belongs to one cell of the grid (otherwise just return the minimum of the OLBIS answers for each cell containing the point).

Each input disk D_j intersects at most 9 cells of the grid. We only keep those cells $C = \bigcup C_i, 1 \leq i \leq m$, that are intersected by at least one disk and ignore the remaining empty cells ($m = |C| = O(n)$). The cells of C are stored in a balanced binary search tree; each cell C_i is also associated with a set of disks $D(C_i) = \{D_{i_1}, \ldots, D_{i_k}\}$, $i_1 < \ldots < i_{k_i}, \{i_1, \ldots i_{k_i}\} \subset \{1, \ldots, n\}$ intersecting it.

Let C_i be the cell that contains p. We have to select a minimum-index disk from $D(C_i)$ that contains p. The intersection of a disk and a cell is simple. With respect to corners and edges of C_i, the disks in $D(C_i)$ form a number of disjoint groups: disks containing exactly one, two, three (one group for each corner, pair of corners, triple of corners respectively) or four corners of the cell, and disks that do not contain any corner of the cell but contain a portion of one of the edges of the cell (one group for each edge), see Figure 1 (the cases when a cell has only one point in common with a disk intersecting it can be ignored).

The number of groups is constant for C_i. We obtain OLBIS answers for each group independently (*partial OLBIS answers*) and then select the mini-

*Department of Computer Science, University of Texas at Dallas, daescu@utdallas.edu. Daescu's research was partially supported by NSF award CCF-0635013.

†Department of Computer Science, University of Texas at Dallas, akurdia@utdallas.edu

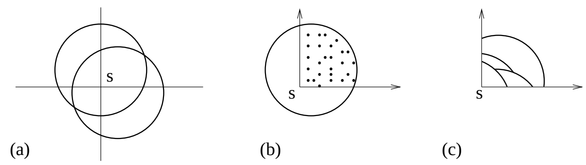

Figure 2: (a) Illustration for Observation 1. (b) Disk piece with corner point s (dotted). (c) An example subdivision R.

mum of these answers. Here we only consider a group $\bar{D} = \{D_1, \dots, D_k\}$ of disks of $D(C_i)$ that contain exactly one corner of C_i (say, left bottom corner, denoted by s). Preprocessing and query answering for the other groups is very similar to those for $\bar{D}$.

Observation 1. *For two equal-radius disks with at least one point s in common, the number of intersections between the boundaries of these disks in any quadrant defined by the vertical and horizontal lines intersecting at s is at most 1 (Figure 2(a)).*

Consider a disk D_i containing a point s, and two orthogonal axis-parallel rays emanating from s and directed upwards and to the right. By $Q_i(s)$ we denote the part of D_i bounded by these rays and call the resulting figure a *disk piece*, point s a *corner point* of a disk piece, and the part of boundary of D_i bounded by the rays *an arc* of the disk piece (Figure 2(b)).

Lemma 2. *Let $Q(s) = Q_1(s), \dots, Q_k(s)$ be a set of k disk pieces in the plane sharing the same corner point s. Let $U_i = \bigcup Q_i(s), i \in \{1, \dots, k\}$, $R_1 = Q_1(s), R_i = U_i \setminus U_{i-1}, i \in \{2, \dots, k\}$ and $R = \{R_1, \dots, R_k\}$ (Figure 2(c)). Then: (i) For any $i \in \{1 \dots k\}$ each disk piece $Q_i(s)$ contributes at most one arc A_i to the boundary of U_i. (ii) The complexity of R is $O(k)$.*

Point location structure. Let $R = \{R_\emptyset, R_1, \dots, R_k\}$ be the subdivision of the plane induced by $\bar{D}$ such that: $R_1 = Q_1(s)$, $U_1 = R_1$, $U_j = U_{j-1} \bigcup Q_j(s), R_j = U_j \setminus U_{j-1}, \forall j \in \{2, \dots, k\}$, $R_\emptyset = \mathbb{R}^2 \setminus U_k$. The index of a face of R containing a query point p corresponds to the partial OLBIS answer for p.

Construction of R. The portions of disks in $\bar{D}$ inside C_i are disk pieces with a corner s. We add disk pieces in order of the indices of their corresponding disks in $\bar{D}$. We store the arcs on the current boundary of R ordered by their left endpoint in a red-black tree T. Let A and B be left and right endpoint of the current boundary, respectively.

Consider step i, when the disk piece induced by D_i is added to R. If $A \in D_i, B \in D_i$, we remove all arcs from T and insert the arc formed by the intersection of the boundary of D_i (∂D_i) with the horizontal and vertical rays originating at s.

If $A \in D_i, B \notin D_i$, we start at B and check all arcs on the boundary moving towards A until we hit an arc A_i intersecting D_i. All arcs encountered before A_i are removed from T, A_i is updated and a new arc contributed by D_i is inserted in T. (Similarly for $A \notin D_i, B \in D_i$).

If $A \notin D_i, B \notin D_i$, we search for intersection points of D_i and the boundary of R in T as follows: At the current node v we check whether ∂D_i intersects the arc A_v associated with this node. If so, under our conditions (disks are distinct, contain s, have equal radius) there can only be one intersection point of ∂D_i and A_v. If a right endpoint of A_v is in D_i, we go to the right subtree to locate the second intersection point, else we go to the left subtree. If $A_v \cap D_i = \emptyset$, we check the two points of intersection of ∂D_i and $\partial D_v, v < i$ (D_v supports the arc A_v). What endpoint of A_v is closer to these points along ∂D_v, that subtree should be searched.

After the points of intersection of ∂D_i with R are found, a new face is created in R.

Point location. Let $P(C_i)$ be a set of the query points contained in C_i. The partial OLBIS answers for all the points of $P(C_i)$ are found simultaneously by performing a plane sweep with a vertical line L. The set of event points contains endpoints of arcs of R, all left- and rightmost points of arcs of R that are not vertices of R, and all points of $P(C_i)$ sorted by x-coordinate. The sweep-status contains all arcs currently intersected by L ordered vertically. Each time an event point p from $P(C_i)$ is encountered, the first arc above or below p is found and the index of the corresponding region is reported. Each time the event point not belonging to $P(C_i)$ is encountered, we update the sweep-line status.

It takes $O(k \log k)$ time to construct R and $O(k + |P(C_i)| \log k)$ to perform point location queries in R.

Theorem 3. *The* min-# *problem with angle constraints in $\mathbb{R}^3$ can be solved in $O(n^2 \log n)$ time and $O(n^2)$ space.*

[1] Danny Z. Chen, Ovidiu Daescu, John Hershberger, Peter M. Kogge, Ningfang Mi, and Jack Snoeyink. Polygonal path simplification with angle constraints. *Computational Geometry*, 32(3):173–187, 2005.

[2] Ovidiu Daescu and Anastasia Kurdia. Polygonal chain simplification with small angle constraints. In *Proceedings of the 20th Canadian Conference on Computational Geometry*, pages 191–194, 2008.