

Incremental Algorithms for Finding the Convex Hulls of Circles and the Lower Envelopes of Parabolas

Olivier Devillers * Mordecai J. Golin †

Revised version. First version in 6CCCG or INRIA Research report 2280

Abstract : The existing $O(n \log n)$ algorithms for finding the convex hulls of circles and the lower envelope of parabolas follow the divide-and-conquer paradigm. The difficulty with developing incremental algorithms for these problems is that the introduction of a new circle or parabola can cause $\Theta(n)$ structural changes, leading to $\Theta(n^2)$ total structural changes during the running of the algorithm. In this note we examine the geometry of these problems and show that, if the circles or parabolas are first sorted by appropriate parameters before constructing the convex hull or lower envelope incrementally, then each new addition may cause at most 3 changes in an amortized sense. These observations are then used to develop $O(n \log n)$ incremental algorithms for these problems.

Keywords: Algorithms, Computational geometry, Convex Hulls, Circles, Parabolas, Lower Envelopes.

*INRIA, B.P.93, F-06902 Sophia-Antipolis cedex, France. Partially supported by ESPRIT Basic Research Action r. 7141 (ALCOM II). email:Olivier.Devillers@sophia.inria.fr

†Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, email:golin@cs.ust.hk. Partially supported by HK RGC CRG grant HKUST 181/93E. Part of this research was performed while the author was visiting INRIA-Sophia.

1 Introduction

In this paper we describe a technique that yields $O(n \log n)$ time incremental algorithms for constructing the convex hulls of circles and the lower envelopes of axis-parallel parabolas. Since, there are already optimal algorithms for these problems, the main interest of this paper resides in the high simplicity of the proposed algorithms. Actually, once the input data have been sorted in a good order the algorithms are incremental and perform only $O(n)$ logarithmic operations on a balanced binary tree.

The convex hull of set of circles, $S = \{C_1, C_2, \dots, C_n\}$ is the smallest convex region containing all of the circles. It consists of a sequence of arcs and common tangent lines connecting consecutive arcs. (See Figure 1.) The convex hull can be constructed in $O(n \log n)$ time using a divide-and-conquer [3] technique or a transformation into a 3D convex hull of points [2].

Given a set of axis-parallel parabolas $S = \{p_1(x), \dots, p_n(x)\}$, $p_i(x) = a_i x^2 + b_i x + c_i$, its lower envelope is the function $F(x) = \min_{i \leq n} p_i(x)$. The lower envelope is composed of arcs of the parabolas. (See Figure 2.) It is of interest because it describes the evolution of the closest-pair in a set of points moving with constant speed [1]. Constructing the lower envelope involves identifying the arcs on the lower envelope. There exists an $O(n \log n)$ di-

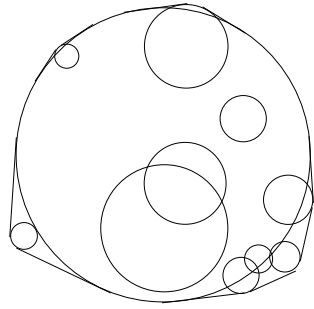


Figure 1: The convex hull of circles

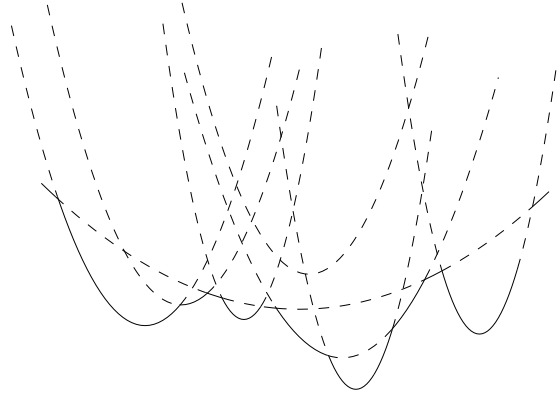


Figure 2: The lower envelope of parabolas

vide-and-conquer algorithm for this construction [1].

An incremental algorithm for constructing the convex hull of circles would incrementally construct the convex hull of the sets $S_i = \{C_1, C_2, \dots, C_i\}$, $i \leq n$. The problem with such an algorithm is that it would have to find *all* of the structural changes that occur while constructing all of the intermediate convex hulls and there may be $\Theta(n^2)$ of these.

As an example consider the following set of $2n$ circles (Figure 3). The first n circles C_1, \dots, C_n , are points (degenerate circles) equally spaced around the circumference of a unit circle centered at the origin e.g., $C_i = (\sin \frac{2\pi i}{n}, \cos \frac{2\pi i}{n})$. Circle C_{n+1} is centered at the origin with radius just enough smaller than 1 so that $CH(S_{n+1})$ is composed of n arcs on C_{n+1} with tangent lines connecting them to the n points. The remaining circles C_{n+1}, \dots, C_{2n} , are nested so that C_{n+i} is inside C_{n+i+1} and all of their radii are less than 1, e.g. C_{n+i} has radius $1 - \frac{1}{2^{n+i}}$. Each of the convex hulls $CH(S_{n+i})$ is composed of n arcs on C_{n+i} with tangent lines connecting them to the first n points. The number of structural changes encountered in going from $CH(S_{n+i})$ to $CH(S_{n+i+1})$ is n so the total number of

structural changes performed during an incremental construction of the final convex hull is $\Theta(n^2)$.

Similarly, an incremental algorithm for constructing the lower envelope of parabolas would calculate all of the lower envelopes $F_i(x) = \min_{j \leq i} p_j(x)$. Again it is possible to find examples where the total number of changes to the lower envelope is $\Theta(n^2)$.

In this note we describe $O(n \log n)$ time *incremental* algorithms for calculating the convex hull of circles and the lower envelope of parabolas. The basic idea in both algorithms is to show that, if the circles/parabolas are sorted by appropriate parameters before the incremental construction begins, then each new circle/parabola will create at most 3 structural changes in the amortized sense.

In section 2 we describe some properties of the convex hull of circles. In section 3 we use these properties to develop an $O(n \log n)$ incremental algorithm. In section 4 we describe the algorithm for constructing the lower envelope of parabolas. We conclude in section 5 by posing related open problems.

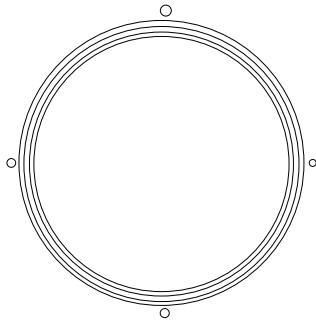


Figure 3: An example in which the number of structural changes in the incremental construction of the convex hull is $\Theta(n^2)$.

2 Convex Hulls of Circles

In this section we discuss some simple properties of the convex hulls of circles. Let $S = \{C_1, C_2, \dots, C_n\}$ be a collection of n circles in the plane. If C is a circle we use \overline{C} to denote the open disc it encloses. Similarly we use $CH(S)$ to denote the boundary of the convex hull of S and $\overline{CH(S)}$ to denote the open region it encloses. Finally, we denote the radius of circle C by $r(C)$.

The convex hull of S is a collection of arcs on the circles of S and tangent lines between the arcs, each circular arc being followed by a tangent line. Any one circle can contribute many circular arcs to $CH(S)$ (Figure 1) but it is the goal of this section to show that if C is a circle with smallest radius in S then C contributes at most one arc. To achieve this goal we describe a new characterization of the arcs that appear on $CH(S)$. This requires the introduction of two new definitions that are illustrated in Figure 4.

Definition 1 Let C, C' be two circles in the plane. The shadow cast on C by C' is

$$\text{Shadow}(C, C') = C \cap \overline{CH(\{C, C'\})},$$

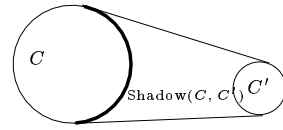


Figure 4: Definition of $\text{Shadow}(C, C')$.

i.e., the set of all $p \in C$ such that p can be seen from C' . For notational convenience we set $\text{Shadow}(C, C) = \emptyset$.

Definition 2 For $C \in S$ the Clear part of C is the section of C that does not appear in the shadow of some other circle in S :

$$\text{Clear}(C) = C \setminus \bigcup_{C' \in S} \text{Shadow}(C, C')$$

$\text{Shadow}(C, C')$ is an arc on C so $\text{Clear}(C)$ is a collection of disjoint arcs along C . What is remarkable is that $\text{Clear}(C)$ is exactly the set of arcs of C that appear on $CH(S)$.

Lemma 1

$$\text{Clear}(C) = CH(S) \cap C.$$

Proof. Let $p \in C$. First suppose that $p \notin \text{Clear}(C)$. Then there is some $C' \in S$ such that $p \in \text{Shadow}(C, C')$. Therefore $p \in \overline{CH(\{C, C'\})}$, so $p \notin CH(S)$ and $CH(S) \cap C \subseteq \text{Clear}(C)$.

Now suppose that $p \notin CH(S)$. Draw the line l through p tangent to C (Figure 5). Because p is not on the convex hull there must be some circle C' and $q \in C'$ such that q is on the other side of l from C . Draw the line segment from q to p . If the segment cuts C' at another point $q' \in C'$ replace q by q' . Then the line segment connecting p and q (or q') does not intersect either C or C' except at its endpoints. This implies that p is visible from q (or q') so

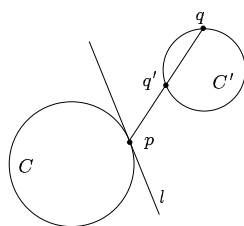


Figure 5: The case $p \notin CH(S)$.

$p \in \text{Shadow}(C, C')$ and $p \notin \text{Clear}(C)$. Thus $\text{Clear}(C) \subseteq CH(S) \cap C$. \square

In general, $\text{Clear}(C)$ can be the union of many arcs but, if C is the smallest circle in S , then $\text{Clear}(C)$ will be exactly one arc.

Lemma 2 *Let C be a circle such that $r(C) \leq r(C')$ for all $C' \in S$. Then $\text{Clear}(C)$ is either empty or consists of exactly one arc.*

Proof. Suppose $r(C) \leq r(C')$. Then $\text{Shadow}(C, C')$ covers at least half the circumference of C . (See Figure 4.) Therefore, each of the $n - 1$ arcs $\text{Shadow}(C, C')$, $C' \in S \setminus \{C\}$, covers at least half the circumference of C . This means that each pair of such arcs intersects and their union is exactly one arc. Thus, $\text{Clear}(C)$, their complement, is either empty or consists of one arc. \square

Corollary 3 *Let C be a circle such that $r(C) \leq r(C')$ for all $C' \in S$. Then $C \cap CH(S)$, the part of the convex hull lying on C , is either empty or consists of exactly one arc.*

Proof. Follows directly from the previous two lemmas. \square

Suppose now that the circles in S are sorted by decreasing radius with ties being broken arbitrarily: $r(C_1) \geq r(C_2) \geq \dots \geq r(C_n)$. Let $S_i = \{C_1, \dots, C_i\}$ be the set of the first i circles. What changes can occur while going from $CH(S_i)$ to $CH(S_{i+1})$?

If $C_{i+1} \subseteq \overline{CH(S_i)}$ then there are no changes. Otherwise, Corollary 3 tells us that C_{i+1} contributes exactly one arc to $CH(S_{i+1})$ along with the two tangent lines coming off the ends of that arc. These tangent lines must touch some arcs A_1, A_2 that were previously on $CH(S_i)$. There are two cases. The first is that $A_1 \neq A_2$. Any arcs appearing between A_1 and A_2 will be deleted from the convex hull as they will be in the shadow of C_{i+1} . This can only decrease the number of arcs on the convex hull. Also, the appropriate ends of A_1 and A_2 are in the shadow of C_{i+1} as well; this modifies their definitions but does not increase the number of arcs on the hull.

The second case is that $A_1 = A_2$. None of the other arcs on the convex hull will be affected by the insertion of C_{i+1} but the shadow of C_{i+1} covers the middle of A_1 splitting it into two arcs. This increases the number of arcs on the hull by 2.

Adding arc C_{i+1} can therefore at most increase the number of arcs on the hull by 2; one for the new circle and one for splitting an old arc. This yields yet another proof that the number of arcs on the convex hull is at most $2n - 1$ (The usual proof utilizes the techniques of Davenport-Schinzel sequences). It also proves that the total number of new arcs that exist during the incremental process – created either by insertion, modification of an endpoint, or splitting – is at most $3n - 2$.

3 Constructing the Convex Hull

In this section we use the results of the previous section to design an $O(n \log n)$ time incremental algorithm for constructing the convex hulls $S_i = \{C_1, C_2, \dots, C_i\}$, $i = 1, \dots, n$. Before running the algorithm the circles are sorted by decreasing radius in $O(n \log n)$ time with ties being broken arbitrarily: $r(C_1) \geq$

$r(C_2) \geq \dots r(C_n)$.

Let q be the center of C_1 , a point which will be in all of the convex hulls. The algorithm assumes that the $O(n)$ arcs in the convex hull of S_i are known and stored in the leaves of a balanced binary tree; the arcs are ordered in clockwise direction around the hull, i.e., they are sorted around q . The line segments on the convex hull are not explicitly stored but may be easily calculated as they are the common tangents of consecutive arcs.

We now discuss how to modify $CH(S_i)$ to become $CH(S_{i+1})$. We assume that we already know if $C_{i+1} \subseteq \overline{CH(S_i)}$ and, if not, then we know some point $p \in CH(S_i) \setminus CH(S_{i+1})$. Later, we shall see how to satisfy these assumptions in $O(\log n)$ time.

The facts that the collection of arcs on $CH(S_i)$ to be destroyed form a continuous chain and that C_{i+1} only contributes one arc to the new hull permit us to easily update the convex hull.

If $C_{i+1} \subseteq CH(S_i)$ then $CH(S_{i+1}) = CH(S_i)$ and there is nothing to do. We therefore assume that that this isn't so and that we know some $p \in CH(S_i) \setminus CH(S_{i+1})$.

Let $A \in CH(S_i)$ be the line segment or arc on $CH(S_i)$ that contains p ; $A \notin CH(S_{i+1})$. Walk clockwise from A along $CH(S_i)$ destroying all arcs traversed until arc A_1 , which shares a supporting tangent with C_{i+1} , is encountered. Then go back to A and walk counter-clockwise again destroying all arcs until arc A_2 , which also shares a supporting tangent with C_{i+1} , is encountered.

There are two cases. If $A_1 = A_2 = A$ then delete A from the tree and replace it by the two sub arcs on both ends of A that are not thrown into shadow by C_{i+1} . If $A_1 \neq A_2$ then modify A_1 and A_2 by trimming off their appropriate ends. In both cases, calculate the unique arc on C_{i+1} (the one bounded by the supporting tangents coming off of A_1 and A_2) in the new convex hull and insert it into the tree. The cost

of the modification will be $O(\log n + d_i \log n)$ where d_i is the number of edges deleted during the step.

The total cost of all of the modifications performed over the entire incremental construction is therefore

$$O\left(n \log n + \sum_i d_i \log n\right) = O(n \log n)$$

because $\sum_i d_i \leq 3n - 2$.

The procedure described above assumed knowledge of some point $p \in CH(S_i) \setminus CH(S_{i+1})$; We now describe an $O(\log n)$ procedure for deciding whether $C_{i+1} \subseteq \overline{CH(S_i)}$ and, if not, returning a point $p \in CH(S_i) \setminus CH(S_{i+1})$. The $O(n \log n)$ running time of the full algorithm will follow.

We first dispense with the case that C_{i+1} is totally outside of the hull. Let p' be any point on C_{i+1} . Consider the infinite ray drawn from q (the center of C_1) through p' and beyond. By searching the binary tree by angle we can find the unique point $p \in CH(S_i)$ on the ray. If p is located between q and p' on the ray then $p \notin CH(S_{i+1})$; we report it and stop the procedure. Otherwise we may assume that C_{i+1} is not outside $CH(S_i)$.

Therefore C_{i+1} either intersects the hull or is totally contained by it. Our procedure will maintain an arc chain of geometrically decreasing size such that if C_{i+1} intersects the hull it must intersect the chain. If we reduce the size of the chain down to one arc and still find no intersection then we will know that C_{i+1} is totally contained by the hull.

Let $\text{Chain}(A, A')$ be the clockwise chain of arcs on the hull connecting arcs A and A' not including arcs A or A' themselves.

Lemma 4 *Let A, A' a pair of arcs on the hull such that no point on them is contained inside C_{i+1} . Furthermore the pair possesses the further property that If C_{i+1} intersects $CH(S_i)$ then C_{i+1} intersects $\text{Chain}(A, A')$. Let A' be*

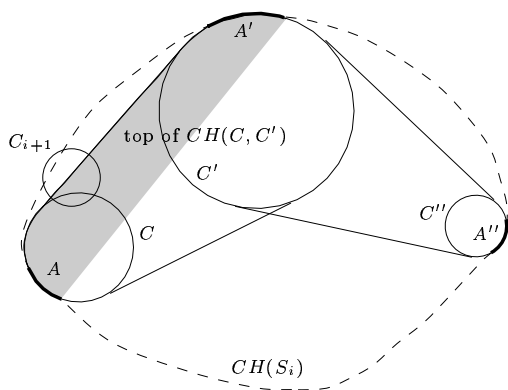


Figure 6: C_{i+1} intersecting $\text{Chain}(A, A')$ must intersect the “top” of $CH(C, C')$.

some arc in $\text{Chain}(A, A'')$. Then at least one of the following statements must be true and we can discover one which is true in $O(1)$ time.

1. Some point $p \in A'$ is inside C_{i+1} .
2. If C_{i+1} intersects $CH(S_i)$ then C_{i+1} intersects $\text{Chain}(A, A')$.
3. If C_{i+1} intersects $CH(S_i)$ then C_{i+1} intersects $\text{Chain}(A', A'')$.

Proof. Let C, C', C'' be the (not necessarily unique) respective circles to which A, A', A'' belong.

We can check in $O(1)$ time if there is some point in $p \in A'$ such that $p \in \overline{C_{i+1}}$. Therefore we may assume that there is no such point.

Examine the tube $CH(C, C')$. The *top* of this tube is the part connecting (and containing) A and A' in the clockwise direction. If $C \neq C'$ then the top has the form arc-line segment-arc while, if $C = C'$, then top is simply the clockwise arc of C' from the beginning of A to the end of A' . Similarly, the *bottom* of the hull is the part connecting A' and A in the clockwise direction (see Figure 6).

Recall that C_{i+1} does not intersect A or A'' . Therefore if C_{i+1} intersects $\text{Chain}(A, A')$ C_{i+1} must either be above or intersect the top of $CH(C, C')$ (see Figure 6).

Suppose then that C_{i+1} is either above or intersects the “top” of $CH(C, C')$, something we can check in $O(1)$ time. Because C_{i+1} is not larger than C, C', C'' we find that C_{i+1} can not also intersect the “bottom” of $CH(C, C')$. Along with the fact that C_{i+1} does not enclose any point in A'' this implies that C_{i+1} does not intersect $\text{Chain}(A', A'')$. Therefore if C_{i+1} intersects $\text{Chain}(A, A'')$ it must intersect $\text{Chain}(A, A')$.

Therefore in $O(1)$ time we can determine if (1) is true. If it isn't then we can determine in $O(1)$ time if C_{i+1} is either above or intersects the top supporting line of $CH(C, C')$. If yes then (2) is true, otherwise (3) must be true. \square

Using this lemma we can discover in $O(\log n)$ time whether C_{i+1} is inside the old convex hull. If it is not then we discover a point $p \in CH(S_i) \setminus CH(S_{i+1})$.

Recall that the arcs are kept in clockwise order. Each internal node v of the tree therefore corresponds to some $\text{Chain}(A, A'')$ of the convex hull. By looking at a node's two children we can also, in $O(1)$ time, find arc A' halfway between A and A'' on the chain.

Let v be the root of the tree. Let A be the first arc in the tree and A'' the last one (We will avoid discussing the purely technical details of maintaining a *circular* order). Check in $O(1)$ time if any point on A'', A , or the tangent line connecting them is contained in $\overline{C_{i+1}}$. If yes return such a point and stop. If not then if C_{i+1} intersects $CH(S_i)$ it must intersect $\text{Chain}(A, A'')$ so we can apply the lemma.

The general step of the algorithm assumes that it is located at some internal node v whose range is $\text{Chain}(A, A'')$ such that (i) no point in A , or A'' is contained in C_{i+1} and (ii) if

C_{i+1} intersects $CH(S_i)$ then it must intersect $\text{Chain}(A, A'')$. In $O(1)$ time find A' halfway between A and A'' on the chain. Now apply the lemma. If (1.) is true then we may return any point on A' inside C_{i+1} and stop because such a point is in $CH(S_i) \setminus CH(S_{i+1})$

Otherwise, if (2.) is true proceed down to the left child of v . If not, then (3.) is true and we proceed down to the right child of v . This process is repeated this until an internal node containing fewer than three leaves is reached. Since the tree is balanced this must happen after $O(\log n)$ steps. Such a node corresponds to three arcs and two line segments on the hull. Check by brute force in $O(1)$ time if any point on them is inside $\overline{C_{i+1}}$. If yes, report such a point. If no, then we have found that C_{i+1} does not intersect the old hull so we may report that it is totally contained inside the hull.

4 The Lower Envelope of Parabolas

Let $S = \{p_1(x), \dots, p_n(x)\}$ be a set of n parabolas, $p_i(x) = a_i x^2 + b_i x + c_i$. Their lower envelope is the function $F(x) = \min_{i \leq n} p_i(x)$. The lower envelope can also be thought of as a minimal sequence of pairs (x_j, i_j) , $i = 1 \dots m$ where $x_1 < x_2 \leq x_m$, $x_0 = -\infty$, $x_m = \infty$ such that $F(x) = p_{i_j}(x)$ for $x_{j-1} \leq x < x_j$.

Calculating the lower envelope involves finding these pairs, which describe where the lower envelope switches from being one parabola to another. The lower envelope is of interest because it describes the closest pair among a set of points moving with constant (but different) velocities [1].

The known algorithm for constructing the lower envelope is an $O(n \log n)$ divide-and-conquer one. The difficulty with developing an incremental algorithm is the same as it was in the case of circles; it is not difficult to construct a set of parabolas such that the

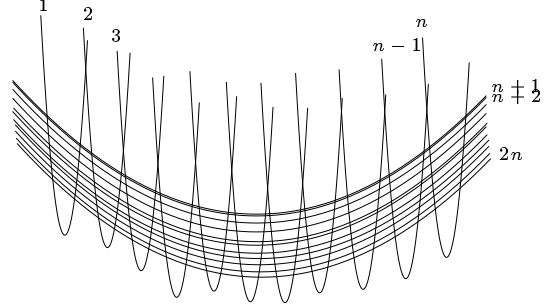


Figure 7: An example in which the number of structural changes in the incremental construction of the envelope is $\Theta(n^2)$.

total number of changes that occur during the incremental construction of the lower envelope is $\Theta(n^2)$ (see Figure 7).

In what follows we show that if the parabolas are appropriately sorted and then added incrementally, then each new parabola can cause at most 3 changes, in the amortized sense. This will yield an $O(n \log n)$ incremental algorithm for constructing the lower envelope. The algorithm follows the paradigm developed for constructing the convex hull of circles developed in the previous section.

We say that $p_i < p_j$ if $(a_i, b_i, c_i) < (a_j, b_j, c_j)$ in the lexicographic sense, i.e. $a_i < a_j$, or $a_i = a_j$ and $b_i < b_j$ or $a_i = a_j$, $b_i = b_j$ and $c_i < c_j$. Assume then that the parabolas have been sorted so that $p_1 < p_2 < \dots < p_n$. Let $F_i(x) = \min_{j \leq i} p_j(x)$ be the lower envelope of the first i parabolas.

We first prove an analogue to Corollary 3.

Lemma 5 *Parabola p_{i+1} can only intersect $F_i(x)$ at at most two points. Thus p_{i+1} can contribute at most one arc to $F_{i+1}(x) = \min(p_{i+1}(x), F_i(x))$.*

Proof.

First note that the lexicographic ordering of the parabolas implies that there is some X such that $p_j(x) < p_{i+1}(x)$ for all $j \leq i$, and all $x < X$, ie., $F_i(x) < p_{i+1}(x)$ for all $x < X$.

If, for all x , $F_i(x) < p_{i+1}(x)$, then the lemma is true. Otherwise let $x_1 = \min\{x : F_i(x) = p_{i+1}(x)\}$ be the leftmost intersection of the old lower envelope with the new parabola. At $x = x_1$ the parabola p_{i+1} switches from being above F_i to being below it. Let $x_2 = \min\{x > x_1 : F_i(x) = p_{i+1}(x)\}$ be the next intersection of the two curves. (If there is no such second intersection then the lemma is obviously true – this may only occur if $a_1 = a_2 = \dots = a_{i+1}$.)

Let p_j be the parabola such that $F_i(x_2) = p_j(x_2)$. Then at $x = x_2$ the parabola p_{i+1} switches from being below p_j to being above it. This implies that $p'_j(x_2) < p'_{i+1}(x_2)$ (where $p'(x)$ is the derivative of p at x). Since $p'_j(x) = a_j \leq a_{i+1} = p'_{i+1}(x)$ this implies that for all $x > x_2$, $p'_j(x) < p'_{i+1}(x)$ so $F_i(x) \leq p_j(x) < p_{i+1}(x)$. This means that x_1 and x_2 are the only intersection points of p_{i+1} with F_i so p_{i+1} can only contribute at most one arc to F_{i+1} . \square

This lemma states that each new parabola may add at most one arc to the lower envelope. There are two cases. The first is that this new arc intersects two different arcs on the old lower envelope. In this case it might also cover (be below) some old arcs, removing them from the lower envelope. The two ends of the new arc will cut off the previously existing ends of the arcs that they intersect on the lower envelope. The second case is that the new arc is totally below one previously existing arc; in this case the old arc is destroyed and replaced by two arcs, one from each of its ends.

Counting the changes implies that each new parabola can only increase the number of arcs on the lower envelope by 2, providing another proof that the maximum number of arcs on the lower envelope is $\leq 2n - 1$ (the standard

proof is a Davenport-Schintzel one). It also proves that the total number of arcs that will ever appear during the incremental construction (counting modified arcs as new arcs) is $\leq 3n - 2$.

The incremental algorithm is almost the same as it was in the circular case. First, sort the parabolas in increasing lexicographic order in $O(n \log n)$ time. Then, incrementally add the parabolas to the current lower envelope. At stage i we assume that the breakpoints x_j of the current envelope F_i are stored in the internal nodes of a balanced binary tree along with the indices i_j .

To proceed we need the following analogue to Lemma 4.

Lemma 6 *Suppose that $p_j < p_i$ in the lexicographic ordering and $p_j(x) < p_i(x)$ for some x . If p_i intersects p_j twice (which is the most it can do) then the intersection points must either both be to the left of x or both to the right of x .*

Proof. Directly from the geometry. \square

We now walk down the binary tree searching for a point x such that $p_{i+1}(x) \leq F_i(x)$. If we can not find such a point this will prove that no such point exists and $F_i = F_{i+1}$.

Start at the root. For each internal node v visited take the breakpoint x_j associated with it and test if $p_{i+1}(x_j) < F_i(x_j) = p_{i_j}(x_j)$. If it is we report x_j and stop. Otherwise we check which of the following three possibilities occur and take the appropriate action: (i) p_i never intersects p_{i_j} . Stop the procedure reporting that $F_i = F_{i+1}$. (ii) The intersection point(s) x where $p_i(x) = p_{i_j}(x)$ is/are to the left of x_j . Then go to the left child of v and continue. (iii) The intersection point(s) x where $p_i(x) = p_{i_j}(x)$ is/are to the right of x_j . Then go to the right child of v and continue.

If we ever reach a leaf of the tree we have reached a breakpoint x_j such that if p_{i+1} intersects F_i it must do so on the arc between x_j

and x_{j+1} . If they do intersect report an intersection point, otherwise report that $F_i = F_{i+1}$.

This procedure takes $O(\log n)$ time and either tells us that $F_i = F_{i+1}$ or reports a point x such that $p_{i+1}(x) \leq F_i(x)$. This point x is on the unique arc of p_{i+1} which is added to the new lower envelope. Given x we can therefore walk to the left and right in the tree deleting all arcs that have to be deleted (they form a chain) in time proportional to $d_i \log n$ where d_i is the number of deletions that need to be performed. After finishing we find the endpoints of the new arc that has to be added and, in $O(\log n)$ time, add it to the tree along with the two new (modified) arcs that it abuts.

Since $\sum_i d_i \leq 3n - 2$ the total running time of this algorithm is

$$O\left(n \log n + \sum d_i \log n\right) = O(n \log n).$$

5 Conclusion

In this note we developed incremental algorithms for constructing the convex hulls of circles and lower envelopes of axis-parallel parabolas in $O(n \log n)$ time. We did this by showing that, if the circles/parabolas are incrementally added in a special order, the geometry of the problem ensures that there won't be many changes at each step and each change can be easily found.

An interesting question left open by this note is just how powerful sorting is for these problems. Sorting lets us reduce the structural complexity of the construction from $\Theta(n^2)$ down to $\Theta(n)$. It does not let us improve upon the previously $\Theta(n \log n)$ time bounds for the problems. The question therefore is "if the circles/parabolas are cleverly presorted can the convex hull/lower envelope be constructed in $o(n \log n)$ time?"

A major open problem is the development of an online algorithm for solving these problems, e.g. given an arbitrary sequence of cir-

cles construct the sequence of convex hulls $CH(S_i)$, finding each convex hull in time 'proportional' to the number of structural changes needed to transform $CH(S_{i-1})$ into $CH(S_i)$, e.g. $O(d_i \log i)$.

We should also mention that the techniques that we describe yield $O(n \log n)$ incremental time algorithms for finding the lower envelope of circles and the convex hull of axis-parallel parabolas (where each parabola p_i is considered as representing the set $\{(x, y) : y \geq p_i(x)\}$.) We do not go into details because these problems do not seem to be of much interest.

Acknowledgement: The authors would like to thank Siu-Wing Cheng and Jacqueline Duquesne for comments and conversations that contributed to this work.

References

- [1] M. Atallah, "Some Dynamic Computational Geometry problems," *Comp. and Maths. with Applications*, **11**(12), 1985, 1171-1181.
- [2] J.-D. Boissonnat, A. Cérézo, O. Devillers, J. Duquesne, and M. Yvinec "An algorithm for constructing the convex hull of a set of spheres in dimension d " *Proc. 4th Canad. Conf. Comput. Geom.* 1992, 269-273.
- [3] D. Rappaport "A convex hull algorithm for discs, and applications," *Comput. Geom. Theory Appl*, **1** (3), 1992, 171-181.