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

Olivier Devillers * Mordecai J. Golin †

Abstract : The existing $O(n \log n)$ algorithms for finding the convex hulls of circles and the lower envelope of parabolas work using 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: Convex Hulls, Circles, Parabolas, Lower Envelopes.

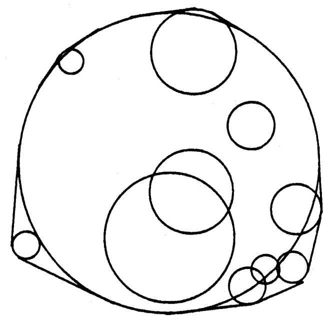

Figure 1: The convex hull of circles

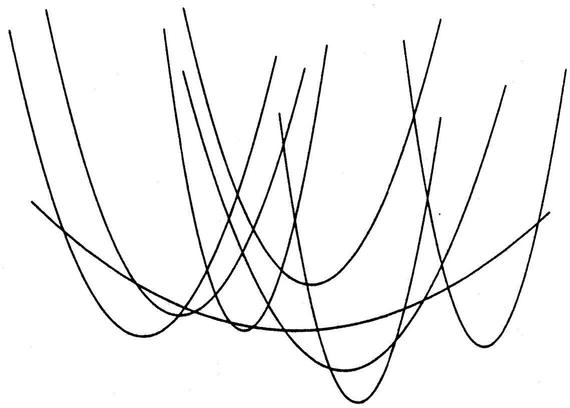

Figure 2: The lower envelope of parabolas

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.

Given a set of circles, $S = \{C_1, C_2, \ldots, C_n\}$, its convex hull is the smallest convex region containing all of the circles. The convex hull consists of a sequence of arcs followed by tangent lines to the arcs (connecting them to the next arc on the hull). See Figure 1. The convex hull can be constructed in $O(n \log n)$ time using divide-and-conquer [3] or a transformation into a 3D convex hull of points [2].

Given a set of axis-parallel parabolas $S = \{p_1(x), \ldots, p_n(x)\}$, $p_i(x) = a_i x^2 + b_i x + c_i$, 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 way the closest pair of a set of points moving with constant speed changes [1]. Constructing the lower envelope involves identifying the arcs on the lower envelope. There exists an $O(n \log n)$ divide-and-conquer algorithm for performing this construction.

An incremental algorithm for constructing the convex hull of circles would incrementally construct the convex hull of the sets $S_i = \{C_1, C_2, \ldots, C_i\}$, $i \leq n$.. 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 convex hulls and there can be $\Theta(n^2)$ of these.

As an example consider the following set of $2n$ circles

*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.

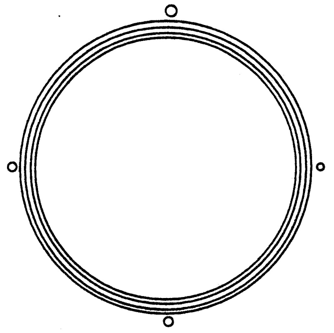

Figure 3: An example in which the number of structural changes in the incremental construction of the convex hull is $\Theta(n^2)$.

(Figure 3). The first n circles $C_1, \ldots, C_n$, are points (small circles) equally spaced around the circumference of a unit circle centered at the origin. 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 $C_{n+1}, \ldots, C_{2n}$, are nested so that C_{n+i} is inside C_{n+i+1} and all of their radii are less than 1. 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 needed to go 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 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 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 a related open problem.

2 Convex Hulls of Circles

In this section we discuss some simple properties of the convex hulls of circles. Let $S = \{C_1, C_2, \ldots, C_n\}$ 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 $r(C)$.

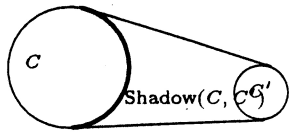

Figure 4: Definition of Shadow(C, C').

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*

$$\begin{aligned} \text{Shadow}(C, C') &= \{p \in C : \textit{p can be seen from } C'\} \\ &= C \cap \overline{CH(\{C, C'\})}. \end{aligned}$$

For notational convenience we set 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:*

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

¿From the definitions it is obvious that Shadow(C, C') is an arc on C so Clear(C) is a collection of disjoint arcs along C. What is remarkable is that Clear(C) is exactly the set of arcs that appear on $CH(S)$.

Lemma 1

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

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

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' with some point $q \in C'$ 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

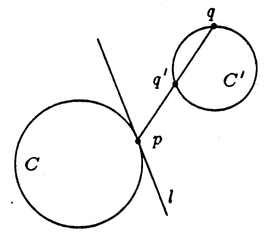

Figure 5: The case $p \notin CH(S)$.

endpoints. This implies that p is visible from q (or q') so $p \in \text{Shadow}(C, C')$ and $p \notin \text{Clear}(C)$. □

In general, Clear(C) can be the union of many arcs but if C is the smallest circle in S then 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* Clear(C) *is either empty or consists of exactly one arc.*

Proof. Suppose $r(C) \leq r(C')$. Then Shadow(C, C') covers at least half the circumference of C. Therefore, each of the $n-1$ arcs 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, Clear(C), their complement, is either empty or consists of one arc. □

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

Proof. Follows directly from the previous two lemmas. □

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

If $C_{i+1} \in \overline{CH(S_i)}$ then there are no changes. Otherwise, Lemma 2 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 that of 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 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 $S_i = \{C_1, C_2, \ldots, C_i\}$, $i = 1, \ldots, n$. Before running the algorithm the circles are sorted by decreasing radius in $O(n \log n)$ time with ties being broken arbitrarily: $r(C_1) \geq r(C_2) \geq \cdots r(C_n)$.

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 a balanced binary tree sorted by the angle made by the lines connecting q with the endpoints of the arcs. The algorithm constructs $CH(S_{i+1})$ from $CH(S_i)$ in a two step phase. In the first step (which we will describe below) it discovers in $O(\log n)$ time whether $C_{i+1} \subseteq \overline{CH(S_i)}$. If it is, it stops the phase. Otherwise, it returns a point $p \in C$ outside of $CH(S_i)$ and proceeds to the second step.

Next it does an $O(\log n)$ search in the tree to find the unique point $q \in CH(S_i)$ on the line connecting q to p. The line segment or arc that q belongs to will no longer be on the convex hull so we can walk clockwise from q along $CH(S_i)$ destroying all arcs traversed until arc A_1, which shares a supporting tangent with C_{i+1}, is encountered. We can then go back to q and walk counterclockwise, again destroying all arcs until arc A_2, which also shares a supporting tangent with C_{i+1}, is encountered. If $A_1 = A_2$ then delete it from the tree and replace it by the two arcs on A_1 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 edges. 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) and insert it into the tree. The cost of the second step 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 the entire algorithm is therefore

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

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

It remains to describe an $O(\log n)$ procedure for performing the first step of the phase – deciding whether

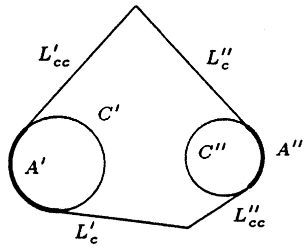

Figure 6: Hammock(A', A'').

$C_{i+1} \subseteq CH(S_i)$, and if not, returning a point on C_{i+1} outside of the convex hull. The $O(n \log n)$ running time of the full algorithm will follow.

To continue we will need the following definitions and lemma:

Let A', A' be two arcs on $CH(S_i)$ belonging, respectively, to circles C', C''. Let L'_c, L'_{cc}, L''_c, L''_{cc} be the respective clockwise and counterclockwise tangent lines leaving the arcs. Define Hammock(A', A'') to be the convex region bounded by A', A'', and the four tangent lines (Figure 6).

Suppose that circle $C \subseteq$ Hammock(A, A'). We say that C is on the *right* side of the hammock if $C \not\subseteq \overline{CH(C', C'')}$ but C is to the 'right' of the leftmost supporting line of $\overline{CH(C', C'')}$. Similarly, we say that C is on the *left* side of the hammock if $C \not\subseteq \overline{CH(C', C'')}$ but C is to the 'left' of the rightmost supporting line of $\overline{CH(C', C'')}$.

In general a circle may be both on the left and right sides of the hammock but, if $r(C) \leq r(C), r(C'')$ then C may only be on one side or the other. We gather the important facts together in the next lemma:

Lemma 4 *Given A', A'' as defined above, exactly one of the following situations occurs. The situation which occurs can be identified in $O(1)$ time.*

1. $C_{i+1} \subseteq \overline{CH(\{C', C''\})}$. *In this case* $C_{i+1} \subseteq \overline{CH(S_i)}$.
2. $C_{i+1} \not\subseteq$ Hammock(A', A''). *In this case* $C_{i+1} \not\subseteq CH(S_i)$ *and a point* $q \in C_{i+1} \backslash \overline{CH(S_i)}$ *can be found in* $O(1)$ *time.*
3. C_{i+1} *in the right side of* Hammock(A', A'').
4. C_{i+1} *in the left side of* Hammock(A', A'').

Proof. Parts (1) and (2) follow directly from the definition of the convex hull which implies that

$$\overline{CH(C', C'')} \subseteq \overline{CH(S_i)} \subseteq \text{Hammock}(A', A'').$$

If (2) occurs then we can also find a point on C_{i+1} outside of the hammock. Parts (3) and (4) follow from the definition of the hammock and the fact that $radius(C_{i+1})$ is not larger than the radii of C' and C''. □

Using this lemma we can discover whether C_{i+1} is inside the old convex hull. Essentially, we will perform a binary search on the arcs of the hull searching for two arcs $A,' A''$ such that if $C_{i+1} \not\subseteq \overline{CH(S_i)}$ then $C_{i+1} \not\subseteq$ Hammock(A', A''). We start with two arcs whose clockwise and counterclockwise distances from each other on the convex hull are approximately equal. At each step of the algorithm we will split the appropriate chain between the two arcs in half and recurse on the appropriate half.

Start with A' being the leftmost arc in the tree and A'' being the arc corresponding to the root. The number of arcs on each of the chains connecting A and A' is approximately $m/2$ where $m \leq 2i - 1$ is the number of arcs on the hull. Apply the lemma. If case (1) applies then $C_{i+1} \subseteq \overline{CH(S_i)}$ so we can stop. If case (2) applies we have found a point on C_{i+1} outside of the convex hull so we can proceed to step 2 of the phase. Otherwise, we know that C_{i+1} must either be on the left side of the hammock or the right side. Assume the right (the left side is symmetric). Let v be the left child of the root of the tree.

In the general step of the algorithm we have two arcs A' and A'' and have just descended to node v in the tree which holds the arc halfway between A' and A''.

Let A''' be the arc corresponding to v. Using the lemma compare C_{i+1} to Hammock(A', A'''). If case (1) or (2) occurs we stop the step and either go on to the next step or the next phase. If case (3) applies then we know that if $C_{i+1} \not\subseteq \overline{CH(S_i)}$ then C_{i+1} must contain a point outside of the part of the convex hull going from A' to A'''; set $A'' := A'''$ and $v := leftchild(v)$. Otherwise case (4) occurs and we set $A' := A'''$ and $v := rightchild(v)$.

Continue this procedure until either a case (1) or case (2) situation is encountered (and we can stop) or v is a leaf. This takes $O(\log n)$ time. If v is a leaf then we have reached a situation where our chain of arcs only contains three arcs. Let C', C'', C''' be the circles on which these arcs are situated. Any point on C_{i+1} outside of the convex hull of these three circles is also outside $\overline{CH(S_i)}$ so we can, in constant time, decide if $C_{i+1} \subseteq \overline{CH(S_i)}$, and, if not, find a point on C_{i+1} outside of the convex hull.

4 The Lower Envelope of Parabolas

Let $S = \{p_1(x), \ldots, p_n(x)\}$ 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), $i = 1 \ldots m$ 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 total number of changes that occur during the incremental construction is $\Theta(n^2)$.

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 constucting the lower envelope. The algorithm is very much like the one 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 $p_1 < p_2 < \cdots < p_n$. Let $F_i(x) = \min_{j \le i} p_j(x)$ 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.

If x is negative enough then $p_j(x) < p_i(x)$ for all $i < j$. This follows from the lexicographic ordering of the parabolas. Thus there is some X such that $F_i(x) < p_{i+1}(x)$ 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 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 $a_1 = a_2 = \cdots = a_{i+1}$.)

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 $p'_j(x_2) < p'_{i+1}(x_2)$. Since $p''_j(x) = a_j \le a_{i+1} = p''_{i+1}(x)$ this implies that for all $x > x_2$, $p'_j(x) < p'_{i+1}(x)$ so $F_i(x) \le p_j(x) < p_{i+1}(x)$. 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}. □

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 numbrer of arcs on the lower envelope is $\le 2n - 1$ (the standard proof is a Davenport-Schintzle 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 $\le 3n - 2$.

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 intersections points must either both be to the left of x or both to the right of x.*

Proof. Obvious from the geometry. □

We now walk down the binary tree searching for a point x such that $p_{i+1}(x) \le F_i(x)$. 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 $p_{i+1}(x) \le F_i(x)$. 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 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 paper 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.

One major open problem in the area is the development of an online algorithm for solving these problems, e.g, given a set of non-sorted circles, 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 also 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.