Visibility Area of a Point in Existence of Circular Obstacles

[1] **S. Aliakbarian** [2] **Seyedsadegh Nikoosepehr**

[1] [2] Department of Computer Science, Sharif University of Technology, saeed@aliakbarian.com
Iran University of Science and Technology, s.sadegh@nikoosepehr.ir

Abstract

In this paper, we consider the special class of visibility problems that is when there are a number of circular obstacles in the plane limiting the visibility area of an observer. Here we propose an algorithm to compute the visibility area, in $O(nlogn)$ time. For the query version of the problem, data structure of size $O(n^5)$ can be built in time $O(n^5 logn)$ so that the visibility sequence of every point q can be reported in $O(logn + |V(q)|)$ time where $|V(q)|$ is the size of the output visibility sequence. For the streaming version of the problem, where new obstacles appear one after another, the new visibility sequence can be computed in $O(log|V(q)|+k)$ time using the previous visibility sequence, where k is the number of the current visibility sequence elements bounded between two tangent lines of the new circle. When only one new circle is going to appear, a faster algorithm can be proposed which uses $O(|V(q)|^2)$ space, takes $O(|V(q)|^2)$ preprocessing time, and computes the new visibility sequence of the point q, in $O(log|V(q)|+k')$, where k' is the number of times the new circle appears in the new visibility sequence.

1 Introduction

Visibility is one of the most famous concepts in computational geometry. In most cases, the visible area is a polygon and is called visibility polygon. In some cases, the visibility area is expressed as a sequence of elements in the space, representing the required visibility area.

Consider the problem of reporting V(q) in a polygon P with h holes. Asano et al. [1] first showed that V(q) can be reported in $O(n)$ query time after preprocessing steps taking $O(n^2)$ time and $O(n^2)$ space. Vegter [2] showed that the query time can be a function of the output size k using an algorithm which reports V(q) in $O(klog(n/k))$ query time after preprocessing steps taking $O(n^2 logn)$ time and $O(n^2)$ space.
In this paper, we consider a special class of visibility problems when there are a number of circular obstacles in the plane limiting the visibility area of an observer.

2 Visibility of a Query Point with Circular Obstacles

Given a number of disjoint circular obstacles in the plane, consider the problem of finding the visibility area of a query point q outside all the obstacles. We define the visibility area for this problem, as the sequence of tangent lines' and circles' labels stated in the counterclockwise order. In the case of the visibility of infinite space, we use the infinite space symbol, C_{inf} , instead of the circle label. The lines determine the start and the end of the visible parts of the circles from q. The part of space bounded between two consecutive lines is called a sector. The exact boundary for the area can be computed with $O(1)$ processing per reporting element, while reporting this sequence.

As the boundary lines can be computed having only the circles, we define the visibility sequence as the list of the seen circles and infinity.

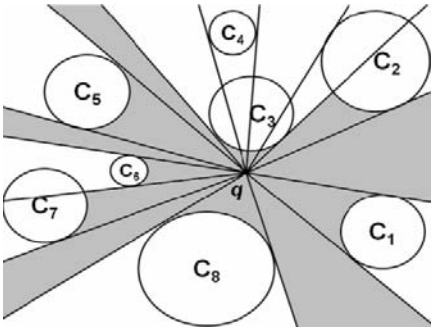

Figure 1: Visibility area of point q

For example, the visibility sequence of the point q, illustrated in figure 1, is (C_1,C_{inf},C_2,C_3,C_{inf},C_5,C_{inf},C_6,C_7,C_{inf}, C_8,C_{inf}). To extract the original visibility sequence we can traverse the given sequence, performing one of the following actions in each step:

1. If we encounter a circle after another one, there will be two cases. If the new one is nearer (according to the center point) to the query point, then the line between them is the tangent from q to the right of the new circle, otherwise the line is the tangent line to the left of the previous obstacle.
2. For the case where either the first or the second circle we are investigating is *Cinf* , simply take the *Cinf* as the farther one.

Following the above approach, the visibility area of q in figure 1, $VA(q)$, will be (RT,C_1,LT,C_{inf},RT,C_2,RT,C_3,LT,C_{inf},RT,C_5,LT, C_{inf},RT,C_6,LT,C_7,LT,C_{inf},RT,C_8,LT,C_{inf}), where LT is the left tangent line of the previous circle and RT is the right tangent line of the next circle. The total time for reporting the visibility sequence will be $O(nlogn)$ which is optimal in the worst case.

3 Query Version

In many cases, with the same set of obstacles, we need to calculate the visibility area for a number of query points. Given a number of disjoint circular obstacles in the plane, in this section we want to preprocess the input so that we can find the visibility area of a point in the plane more efficiently.

We define the visibility decomposition of the plane for this problem as the plane decomposed to regions by drawing the 4 tangent lines between each two circles in the plane. Therefore,

we will have a decomposition where all the points in the same cell of such a visibility-decomposition have the same visibility sequence. The result will be the following theorem:

Theorem 2 *Given a set of n disjoint circles in the plane, a data structure of size* $O(n^5)$ *can be built in time* $O(n^5 \log n)$ *so that the visibility sequence of every point q can be answered in* $O(\log n+k)$ *time where k is the size of the output visibility sequence.*

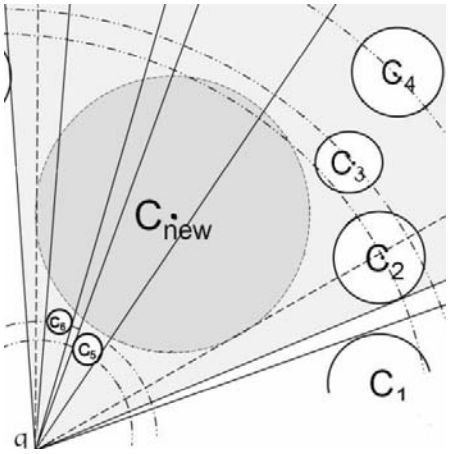

Figure 2: Adding anew circle

4 Newly Appearing Circles

Having calculated the visibility sequence of a set of circles, a new circle may appear and change the visibility sequence. In this section, we are going to calculate the new visibility sequence after the addition of a new circle, disjoint with all the existing ones. As the addition of a new circle will not result in seeing an unseen circle, we use $|V(q)|$ as the number of circles instead of n.

4-1 Streaming version

The case where new circles may appear in arbitrary order and rate, is called streaming version of the problem.

Theorem 3 *Given the visibility sequence of a number of disjoint circles, V(q), the new visibility sequence after adding every new circle disjoint with all existing circles, can be computed in* $O(\log|V(q)+k)$ *time, where k is the number of elements in the current visibility sequence between two tangent lines from q to the new circle.*

Proof. Sectors' angle information can be stored along with the visibility sequence elements so that the sector containing any given angle is computed in $O(\log|V(q)|)$. Using the above technique, we compute the sectors containing the two tangent lines from q to the new circle. Then we start traversing through the sectors, starting from the first one going toward the last one counterclockwise. For each sector, if the new circle is nearer than the visible circle in that sector, the new circle is reported in the visibility sequence instead of the current one; otherwise, the element is unchanged. As for the first and the last sector, most probably (except that the new circle covers that sector completely), the sector is divided into two sectors, in one of which the current circle and in the other one the new circle is reported. This will take $O(\log|V(q)|+k)$ time to check all the sectors.

4-2 A Faster Algorithm When Only One Circle is Added

As you can see in figure 2, when the new circle covers a number of continues elements in the visibility sequence, traversing all of them is not required as only the new circle is visible in the new sector covering all the old sectors. Similarly, when a number of continues circles cover the new circle, traversing all of them is not required as the visibility sequence is not changed in that part.

Theorem 4 *Given the visibility sequence of a number of disjoint circles, V(q), the new visibility sequence after adding a new circle disjoint with all existing circles, can be computed in* $O(\log|V(q)|+k')$ *time, where k' is the number of times the new circle appears in the new visibility sequence. This algorithm will take* $O(|V(q)^2|)$ *preprocess time and space.*

Proof. In the new algorithm, we divide the plane in two directions; in the angular direction, it is divided into sectors by the sector borders in the visibility sequence, and in outward direction, it is divided by tracks centered at q and with their radius equal to the distance of every visible circle center from q. In fact, as you can see in figure 2, the tracks pass over the current visible circles' centers. For each sector we store a sorted list, here called the outward list, containing one element per track, so that given a distance from q, the track immediately farther or with the same distance can be found in $O(\log|V(q)|)$ time. We also store three pointers for each element of these outward lists named *Next*, *NextNearer* and *NextFarther*. Computing such pointers will take $O(|V(q)^2|)$ time, which is done during the preprocess.

Having the above structure and pointers, our algorithm, stated and proved in the full paper, will fulfill the time and space requirements of theorem 4.

Combining this algorithm with streaming version one, one can achieve a method to keep an updated visibility sequence over a time, while being able to respond to a new appearing obstacle more quickly. This can be done, by first using the faster algorithm to report the visibility sequence whenever a new obstacle is appeared, and then updating the streaming version data structure, and at last updating the more complex above stated structure afterwards in the system idle time.

5 References

[1] T. Asano, L. J. Guibas, J. Hershberger, and H. Imai. *The Visibility of disjoint polygons.* Algorithmica,1986.

[2] G. Vegter. *The visibility diagram: a data structure for visibility problems and motion planning.* In Proceedings of the 2nd Scandinavian Workshop on Algorithm Theory, Lecture Notes in Computer Science, Springer-Verlag, Berlin, Germany, 1990.