Kinetic Visible Set Maintenance in the Plane

Olaf Hall-Holt *

Graphics Lab., Computer Science Dept., Stanford University
Stanford, CA 94305, U.S.A.
`olaf@cs.stanford.edu`

Abstract

This paper examines the problem of maintaining the visible set of a moving observer among n non-overlapping convex objects in the plane, where the motion is specified on-line using constant degree pseudo-algebraic trajectories. The visible zone *is a linear size visibility structure analogous to a topological line through the visibility complex. Initializing a visible zone requires $O(n \log n)$ time, and querying a visible zone for the visible set requires time proportional to the size of the visible set. For static scenes where the angle between two bitangents connected to a single object can be bounded from below by a constant, the cost of maintaining a visible zone is proportional to the number of changes in the visible set. The visible zone can also be maintained for moving objects, without increasing the cost associated with changes in the observer motion.*

1 Introduction

A classic problem in computational geometry is computing the objects visible to a point observer in a three dimensional scene [5]. In many applications of this problem in computer graphics, both the observer and scene objects are assumed to move. This motion can change frequently, since it depends on on-line user input. Thus, the focus of this on-line problem shifts from computing to maintaining the visible set.

A simplified, planar version of the above problem can be stated as follows: Let S be a set of n non-overlapping convex objects in the plane. Given an on-line description of the motion of a point observer and scene objects, compute the visible set of the observer as time progresses. Ideally, a solution to this problem will have time and space update complexity that is a function of the number of changes in the output, rather than the size of the entire scene, even given unpredictable changes in the motion.

Visibility problems within a simple polygon in the plane are well studied (see, e.g. [8]). Let p be a point on the boundary of a polygon which is occluded from the point of view of the observer. If the observer wishes to see the point p, then the observer must round a known set of "corners." To keep track of what the observer sees, it suffices to link the occluded parts of the polygon with the associated corners, and update this information as the observer moves. This strategy yields a linear size data structure with optimal or near-optimal maintenance costs [1].

When multiple objects are present, the paths linking an observer to a given object boundary point are not necessarily homotopic. However, it is still true that knowing the set of objects seen around a corner provides enough information to maintain the visible set. Storing what can be seen around every corner results in an algorithm with maintenance costs logarithmic in the size of the scene [11] (see also [7]). However, the size of the data structure may be quadratic, and the maintenance costs for moving objects may also be undesirably high.

For static scenes, many approaches have already been explored. If the observer moves along a linear trajectory, then the scene can be preprocessed to help identify the points along the trajectory that correspond to changes in the visible set (in 3D, see e.g. [3]). For observer motion that changes on-line, various spatial decompositions can be computed in advance, such as BSP trees (e.g. [9]) or octrees (used with "large occluders" in [4]). These approaches can

*Partially supported by an FPM Fellowship and by DARPA grant DAAE07–98–C–L027.

work well in practice, but appear not to generalize easily to the moving object case.

One of the challenges of handling scenes that are not static involves distant dis-occlusions. Suppose an observer sees only a few nearby objects of the scene, yet can see a great distance between them. Many object motions far from the observer can be hidden by the nearby objects, but arbitrarily distant objects may nevertheless suddenly enter the field of view. How is the significance of one particular object motion to be flagged, without testing all possible moving objects? This challenge motivates a search for a structure analogous to the Delaunay triangulation, where maintaining a local property is sufficient to certify a global characteristic.

Two recent fundamental developments permit a new approach to this problem, namely the introduction of the *visibility complex* [10] and Kinetic Data Structures [2]. The visibility complex expresses the combinatorial structure of visibility relationships between objects in the plane. Kinetic Data Structures provide a foundation for reasoning about and maintaining geometric properties in an on-line setting.

Together, substructures of the visibility complex and kinetic maintenance permit the construction of an on-line algorithm for visible set maintenance with the following properties:

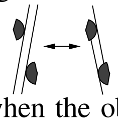
- The cost of initialization is $O(n \log n)$ time and linear space.
- The structure is optimal for handling frequent changes in motion, in the sense that when the observer changes course, the cost of this update is proportional to the size of the visible set.
- For static scenes where the difference in angle between two bitangents connected to a single object is bounded from below (i.e. where objects are not arranged in a row), the cost of maintaining the visible set as the observer moves along a constant degree pseudo-algebraic trajectory is proportional to the number of changes in the visible set.
- Any number of moving objects is allowed.

We have implemented a version of this algorithm, and tested it on scenes with up to one million objects. The details of representing the chosen substructure of the visibility complex and updating such a representation will be described in a future paper. The focus of this paper is the correctness and asymptotic complexity of the algorithm.

Section 2 illustrates the similarity of these new visibility structures to radial decompositions of the plane around the observer. Section 3 interprets this development in the context of the visibility complex, and defines the visible zone. Section 4 summarizes some combinatorial properties of *zone transformations*, and the effect of motion. Section 5 describes the kinetic maintenance algorithm, including a fundamental difference between static scenes from the point of view of a moving observer, and scenes with moving objects. Section 6 describes and generalizes the above mentioned complexity bounds, and section 7 concludes.

2 Flexible Radial Decompositions

A first approach to maintaining the visible set of a moving observer in a moving scene is to maintain the radial decomposition around the observer (Fig. 2a on page 8 gives an example of a radial decomposition). This method takes advantage of temporal coherence, in the sense that, for small motions, the combinatorial structure of the radial decomposition may not change. Also, handling moving objects is straightforward, and the sum of update maintenance costs for constant degree pseudo-algebraic motions is $O(n^2)$.

Let us call the segments of the radial decomposition **silhouette segments**, since they indicate the silhouettes of the objects as seen from the observer. In general, a silhouette segment is a segment of the plane, tangent to one object, connected to two other objects, and not passing though the interior of any object. In maintaining the radial decomposition, combinatorial changes take the following specific form: two silhouette segments turn until they share the same supporting line, and then they separate again, but with different objects at their endpoints.  Note that these events occur exactly when the observer crosses the supporting line of some bitangent.

A drawback to maintaining the radial decomposition, particularly when the motion of the observer changes, is that all segments of the decomposition are moving all the time, no matter how far they lie from the observer. If the scene is composed of point-like objects, this behavior is inevitable, but for many scenes, in particular densely occluded scenes, it seems as though this cost should not be so high.

A second approach to maintaining the visible set is to decouple the motion of silhouette segments not directly connected to the observer from the motion of the observer, but still require that the silhouette segments form a spatial decomposition: start with the radial decomposition, and as the observer or an object moves, only update silhouette segments in the neighborhood of the observer or moving object. The necessary updates in this method would be of two types: moving one silhouette segment when another is about to intersect it at an endpoint, and handling combinatorial changes. The latter update would be identical to the update used for maintaining the whole radial decomposition.

The complexity of this latter approach is difficult to analyze, since the updates depend not only on the combinatorial structure of silhouette segments, but also on their geometric position. In addition, the changes in the spatial decomposition necessary to maintain its planarity enforce a type of rigidity in the structure, so that motion of the observer could affect the silhouette segments far from the observer.

A third approach to maintaining the visible set is to reduce this rigidity further by allowing silhouette segments to intersect, but not too much. That is, we enforce the following rule: two silhouette segments are allowed to intersect, but only if they do so on the same side of their respective tangent points, and only if they both end on the same object (on that side). This allows silhouette segments to cross one another at their endpoints, while still restricting their relative position somewhat. In particular, the combinatorial characteristics of the silhouette segments are sufficient to describe and maintain this quasi-spatial decomposition.

It turns out that this latter local restriction is sufficient to ensure the desired global property, where the set of silhouette segments always preserve enough adjacency information to be used in concert. This third approach is examined in the remaining sections of the paper. Proving the correctness of this method relies on properties of silhouette segments that may be difficult to visualize in the plane. A natural embedding for the set of silhouette segments is to consider them in the context of the visibility complex.

3 The Visible Zone

The visibility complex has been defined in several similar ways; the following definition is most suitable for our analysis. Let S be a set of non-overlapping convex objects in the plane, including a special point object for the observer. A **maximal segment** is a segment that does not intersect the interior of any object, and cannot be lengthened along its supporting line without doing so. Intuitively, a maximal segment joins objects that can see each other.

The **visibility complex** is the set of maximal segments, grouped into connected **faces**, **edges**, and **vertices**, according to combinatorial type. That is, a face of the visibility complex is a two dimensional set of maximal segments which connect two objects, and which can obtained by moving a single maximal segment around, always connecting the same two objects. An edge of the visibility complex corresponds to the set of maximal segments obtained by rotating a silhouette segment as far as possible in both directions (along the boundary of its tangent object), without changing its combinatorial type. A vertex of the visibility complex is a maximal segment tangent to two objects at once. All vertices and edges in this paper consist of a maximal segment, or a set of maximal segments.

As indicated in the previous paragraph, silhouette segments find a natural expression in the visibility complex. The particular silhouette segments of the radial decomposition are related to each other as follows. Let $\mathcal{R}$ be the set of all maximal segments whose supporting lines pass through the observer. Then $\mathcal{R}$ includes the silhouette segments of the radial decomposition. The remaining segments of R can be grouped into one-dimensional sets of maximal segments, each set connecting one silhouette edge to another. Thus, the maximal segments of $\mathcal{R}$ together constitute a *curve* through the space of maximal segments, whose intersections with the edges of the visibility complex are the silhouette segments.

As the observer moves, this curve $\mathcal{R}$ sweeps through some part of the visibility complex. In particular, in an event when two silhouette segments come together, the curve passes through a vertex of the visibility complex: before the event, the silhouette segments lie on two edges of the visibility complex. After the event, the silhouette segments have moved to two other edges of the visibility complex. The intersection of these four edges is a vertex of the visibility complex. Thus the motion of $\mathcal{R}$ is analogous to the motion of a sweep line moving across a planar arrangement of lines.

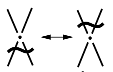
The maximal segments of a face of the visibility complex can be ordered according to a locally defined relative angle: one maximal segment is considered **above** another if its supporting line has greater angle than that of the other maximal segment. In this local order, the boundary of a face includes one maximal and one minimal vertex, which divide the boundary into two **side chains**. In speaking of this local ordering on maximal segments, we always identify the relative angle with height, so that the maximal vertex of a face f is said to be locally **above** the minimal vertex of f.

The previous section considered silhouette segments whose exact position along the boundary of an object did not matter, provided that the combinatorial type was fixed. In such a context, the curve $\mathcal{R}$ is part of an equivalence class of curves through the visibility complex, which all intersect the same set of edges of the visibility complex. The following definition follows the terminology first used in the topological sweep paper [6], this time applied to the visibility complex, instead of an arrangement of lines.

Definition 1: A **cut** is a set of edges (called **cut edges**) with the property that, for any face f with a cut edge on its boundary, there is exactly one cut edge on each of the side chains of f.

A cut is a slight generalization of the *anti-chain* topological sweep structure defined in [10], using a slightly different definition for the visibility complex. The **zone** of a cut is the set of faces (called **zone faces**) with cut edges on their boundaries. A particularly interesting cut, the **perspective cut**, is precisely the set of edges containing the silhouette segments of the radial decomposition. The perspective cut thus captures the notion of the freedom of these silhouette segments to move around, possibly intersecting each other, without losing the underlying combinatorial information.

In a topological sweep of an arrangement of lines, or of the visibility complex, a cut can be modified incrementally, as follows:

Definition 2: An **elementary step** is an operation by which a cut is changed, by removing two cut edges that share a vertex, and replacing them with the other two edges incident to the vertex. 

We mentioned before that maximal segments in a local neighborhood can be ordered by relative angle of their supporting lines, and that this order can be associated with height. An elementary step is considered to go **up** if it involves replacing edges below a vertex with those above, or **down** if it goes the other way. A cut never contains edges both above and below a vertex, since they would both lie on a single side chain. It can be shown that applying an elementary step to a cut always results in another cut.

To put these definitions another way, a cut captures the combinatorial relationships among a set of silhouette segments, and an elementary step corresponds to the event where two silhouette segments come together and swap endpoint objects. As long as a set of silhouette segments includes the silhouette segments emanating from the observer, the visible set is immediately available. By applying elementary steps that modify the cut in regions not visible to the observer, the perspective cut can be transformed into other cuts, without losing information about the visible set. This observation motivates the following definition:

Definition 3: A **visible zone** is the zone of any cut obtainable by applying elementary steps to the perspective cut, without disturbing edges that connect to the observer.

A visible zone is thus a linear substructure of the visibility complex. A visible zone captures the combinatorial information of the visible set, as well as a set of relationships extending to all objects of the scene. It can be visualized as a quasi-spatial decomposition of silhouette segments, or as a sort of topological line through the visibility complex, or, as a *pseudo-triangulation* (defined in [10]). The visible zone is the basis for all the visibility algorithms considered in this paper. Let us consider the conditions under which a zone can be maintained as objects move.

4 Zone Transformations

In order to describe properties of visible zones, in particular the effect of object motion, we examine the structure of sequences of operations that can be applied to modify a cut. Intuitively, this is very similar to considering how a topological sweep line can be moved around in a planar arrangement of lines. The treatment of this interesting topic is necessarily brief, in order to allow a full consideration of the maintenance algorithm for a visible zone. The first subsection below states some important definitions and properties for these sequences. The second subsection examines the implications of motion.

4.1 Transformation Sequences

To transform one zone into another, one or more elementary steps may be applied, each of which moves the zone locally up or down. A more general transformation operation can also be defined, called a **composite step**, which moves the cut to the opposite side of a set of vertices all at once. In general, a **zone transformation** is an ordered sequence of these elementary and composite steps, or **steps** for short. Each step can be represented by a set of vertices, since the direction of the step is implied by the location of the cut. A transformation is represented by a sequence of sets of vertices V_1, V_2, ..., V_n. Two zones are **connected** if there is a transformation taking one to the other.

Two consecutive steps in a zone transformation with the same set of vertices (whether elementary steps or composite steps) are called a **redundant pair**, since their effects on the cut cancel each other out. When a transformation contains a redundant pair, removing this redundant pair from the sequence is in some sense a trivial adjustment of the sequence. Also, if two consecutive steps of a transformation operate on completely different parts of the visibility complex, transposing their order in the sequence (an **allowable** transposition) is also in some sense a trivial change.

Definition 4: Two sequences of steps T_1 and T_2 that both transform a zone z_1 into another zone z_2 are equivalent ($T_1 \sim T_2$) if by allowable transpositions in the order of steps, and the removal of redundant pairs, T_1 and T_2 can be reduced to the very same sequence.

One example of when two steps in a transformation sequence can be transposed is the case of a pair of non-redundant consecutive steps with opposite directions. We examine this case in detail; the analysis for many other properties is similar in form. Suppose v_i and v_{i+1} are the vertices of two consecutive elementary steps, which are upward and downward steps, respectively. The vertices v_i and v_{i+1} are distinct, by non-redundancy, and so the only way for these two elementary steps to have anything in common is to share an edge. However, just after the step across v_i, the edge above the vertex v_i is a cut edge (by the operation of the elementary step), and just prior to the step across v_{i+1}, the edge below v_{i+1} is not a cut edge, so such an edge cannot be shared. Again, just after the step across v_i, the edge below the vertex v_i is not a cut edge, whereas the edge above the vertex v_{i+1} is a cut edge, so such an edge cannot be shared either. Therefore an upward elementary step can always be transposed with a downward elementary step next to it in the sequence, provided that these two do not form a redundant pair.

The two operations on transformations defined above, namely deleting redundant pairs and transposing steps, can be used in several ways. A **unidirectional** transformation has all upward steps, or all downward steps. It can be shown that any transformation can be reduced to a concatenation of two unidirectional transformations, with no vertices shared in common between their respective steps. This reduced form is called a **non-redundant transformation**.

In the definition of the visible zone, we required certain cut edges to remain in the zone, regardless of which transformation was applied. A general mechanism for specifying restrictions on transformations is to specify some set of vertices (the **restricted vertices**) which are not allowed to appear in the steps of any transformation. These transformations are then called **restricted transformations**.

It can be shown that all restricted transformations connecting a given pair of zones are equivalent. A restricted transformation cannot have an arbitrarily large number of steps, because it would eventually have to cross a restricted vertex. For any zone, all upward (resp. downward) restricted transformations of maximal length are equivalent. In a space of connected zones connected by restricted transformations, there is a unique **maximum** zone, obtained by applying the longest possible upward restricted transformation.

Theorem 4.1 (Unique Minimum). *Let z be a zone, and let $\mathcal{T}$ be a maximal set (by inclusion) of non-redundant, restricted transformations of z that each have exactly one upward step involving a vertex v. Then there exists a minimum length transformation T of $\mathcal{T}$ that occurs as a prefix to all the other transformations of $\mathcal{T}$, up to allowable permutations in the order of steps.*

A minimum length transformation T has several interesting properties, including the following: if T includes a step over a vertex w, then there is no other step of T over w. Thus in order to move the zone to a desired location of the visibility complex, if this location can be described as being on "the other side" of a particular vertex, then there exists a single most efficient transformation for doing so, and this transformation will never cross any vertex twice.

4.2 Effect of Motion

When the objects of a scene begin to move, no change in the combinatorial structure of the visibility complex occurs until a single maximal segment becomes tangent to three objects at once. The instantaneous formation and disappearance of such a triple tangent is called a **triple event**.

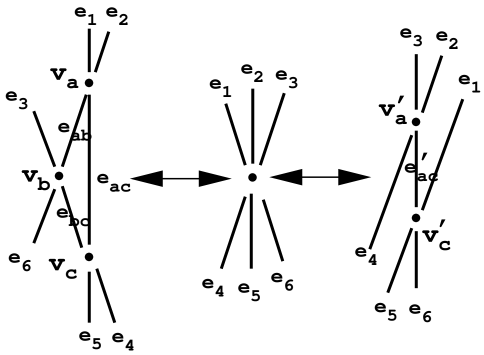


Fig. 1: combinatorics of a triple event

As can be verified by lining up three glasses on a table, a triple event involves either the creation or destruction of a face of the visibility complex. When a face f is about to disappear, it has three vertices along its boundary, and three edges. On the other side of the triple event, there are two vertices, connected by a single edge of the visibility complex. The combinatorial changes in the visibility complex due to a triple event are summarized in Fig. 1.

The connections between zones can be restricted by barring steps across a given set of vertices, and they can also be restricted in another way: by only allowing transformations that are composed of elementary steps. These **elementary transformations** will be seen to be simpler from an algorithmic point of view than the general case. The results and tools of the previous subsection apply to elementary transformations as well as general transformations.

The essential problem in maintaining a zone is determining whether its connectivity to other zones can be negated by the motion of objects. For general transformations, this connectivity is assured.

Theorem 4.2 (General Connectivity). *If z_1 and z_2 are two connected zones in a visibility complex C, and a triple event changing C into another visibility complex C' does not remove any of their cut edges, then z_1 and z_2 are also connected in C'.*

For elementary transformations, a later section will give an example where this result does not apply. However, the connectivity property is invariant for an elementary transformation if the following condition can be met:

Acyclicity Condition: A nonredundant elementary transformation T satisfies the acyclicity condition if, for any pair of elementary steps V_i and V_j of T whose vertices are found at the opposite ends of a disappearing edge e (disappearing due to a triple event), the order of elementary steps of T can be permuted so that V_i and V_j are adjacent in the new ordering.

Once we are assured of the connectivity of zones under object motion, the only events that need to be checked to ensure the correctness of maintaining a zone are those that affect the zone directly, namely changes in the cut edges, or changes in the position of the zone relative to a restricted vertex. The next section describes how these changes can be handled in detail.

5 Visible Zone Maintenance

The second section described a first approach to maintaining the visible set by maintaining the radial decomposition, and a second approach based on maintaining a related spatial subdivision. In this section we describe the general algorithm corresponding to the third approach, based on visible zones. We will describe this algorithm with silhouette segments, but the underlying combinatorial changes are the same when other representations of the visible zone are used.

The first subsection below considers the case of a static scene, where only the observer moves. The second subsection describes the general case of moving objects.

5.1 Observer Motion

Imagine starting with a radial decomposition, and allowing only the edges connected to the observer to move as the observer moves. An update is required exactly when one of the edges connected to the observer becomes tangent to another object. At this moment, the visible set changes.

Let us call the silhouette segments connected to the observer **visible** silhouette segments, and the edges of the visibility complex connected to the observer **visible** edges. The visible silhouette segments are exactly the silhouette segments belonging to visible edges. All updates in a static scene are triggered in the above manner, when a visible silhouette segment becomes tangent to another object, or equivalently when the set of visible edges changes.

The simplest update occurs when there is another silhouette segment right next to the moving visible silhouette segment, so that all that is needed is a single elementary step involving the two silhouette segments. When this elementary step has been applied, the set of visible silhouette segments has increased or decreased by one member, and the visible set is correctly maintained.

It may happen, however, that the other silhouette segment necessary for an elementary step is not in the correct position to allow this elementary step. In this case, a more complex update is necessary. Although it is not possible to apply the desired elementary step immediately, it is nevertheless clear which other silhouette segment would be involved in this elementary step. The update in this case consists of turning the second silhouette segment toward the visible silhouette segment, until the elementary step can be made. Turning a silhouette segment around the boundary of its tangent object, however, may require additional elementary steps, any of which may require turning yet other silhouette segments.

The full update algorithm proceeds recursively, as follows.

Update Algorithm: Given a silhouette segment m and an elementary step v that we would like to apply to m, first locate the other silhouette segment n that would participate in v. If n is not able to participate in v immediately (n may not be right nearby, combinatorially), then repeatedly locate the first elementary step w that needs to be applied to n to bring it closer to v and call the update algorithm with n, until n is able to participate in the elementary step v. Finally apply the elementary step v to m and n.

Note that this algorithm, if it terminates, turns silhouette segments in only one direction, toward the place they would lie in the radial decomposition. Thus the complexity of maintenance in this case is always bounded from above by the complexity of maintaining the radial decomposition.

To prove that the update algorithm terminates correctly, we apply the tools developed in the previous section. In particular, we would like to know whether a zone remains a visible zone as the observer moves, in order to ensure that the space of connected zones never collapses. As before, let C be a visibility complex just before a triple event t, and let C' be the new visibility complex just after t. To use the above terminology on transformations, a visible zone can be defined as a zone connected to the zone of the perspective cut via a restricted transformation, where the restriction refers to the upper and lower vertices of the visible edges.

Lemma 5.1. *Let z be a visible zone in C, none of whose cut edges are removed by a triple event t (resulting from observer motion). Then the corresponding zone z' in C' is also a visible zone.*

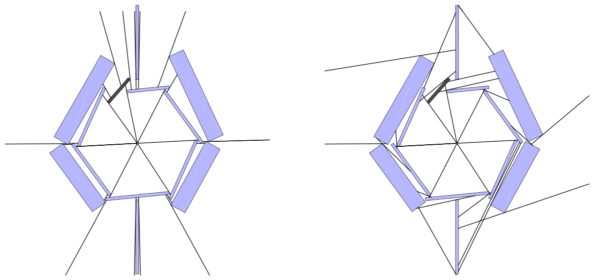
The proof of this lemma involves showing that the vertices of a connecting transformation from z to the zone of the perspective cut are never destroyed by t. This lemma ensures that when a transformation is required to move to the other side of restricted vertex, such a transformation will always exist. The update algorithm identifies the shortest such transformation, and thus will terminate correctly.

5.2 Moving Objects

The above algorithm for maintaining the visible zone of a moving observer must be augmented in several ways in order to handle the case of moving objects, but the basic idea is the same. Unfortunately, it is no longer true that elementary steps will always suffice in performing updates of the zone.

Once again, we would like to prove that a zone remains a visible zone as motion occurs in the scene. The section on transformations asserted that elementary steps would be sufficient, provided that a certain acyclicity condition was satisfied. It can be shown that the acyclicity condition is always satisfied for the kind of cuts called anti-chains (see [10]), but it is unfortunately not always satisfied for visible zones.

Let S be the scene shown in Fig. 2. The radial decomposition is shown in Fig. 2a. Among all the zones connected by elementary steps to the zone of the perspective cut, the zone with the least 'freedom of movement' is the maximal zone obtained by applying a maximal upward elementary transformation, as shown in Fig. 2b.



a) radial decomposition b) maximum zone

Fig. 2: example of a cycle

Let A be the smallest object of S, namely the one shaded dark grey. If A is moved down and to the left, a triple event occurs that does not affect any of the cut edges, but destroys the elementary step connection to the perspective cut. If A continues this trajectory until the visible set changes, the resulting update cannot be performed using elementary steps.

Thus for the general case, it is necessary to handle composite steps. A composite step is analogous to a set of elementary steps performed simultaneously. In order to break a composite step into pieces, one approach is to allow the temporary creation of "negative" silhouette segments, and force an initial elementary step where it could normally not be performed by creating a positive/negative pair of silhouette segments where they are needed. It can be shown that composite steps can be broken up into a linear sequence of normal and forced elementary steps, and that at the end of this sequence, all negative silhouette segments once again have corresponding positive silhouette segments. More sophisticated methods based on the pseudo-triangulation representation may also be applicable.

In the moving object case, the basic update algorithm is similar to that given above for the static case, when it is observer motion that triggers the update. The only difference is that, during the recursion, if a given silhouette segment is encountered twice, a composite step must be applied. The proof of correctness then relies on the general connectivity theorem.

A second type of update also becomes necessary, due to the collapse of cut edges as objects of the scene move. In this case, there are two possibilities: either to move the zone up above the collapsing cut edges, or to move it down. For triple events where a face of the visibility complex disappears, this event can always be handled with a single elementary step, since the two disappearing cut edges share a single vertex. For triple events where only one cut edge disappears, the corresponding silhouette segment is moved toward the position it would occupy in the radial decomposition. In practice, we compute this direction via the simple heuristic of the nearest possible choice. The above update algorithm can then be applied to this silhouette segment also, since the underlying machinery involved (moving the zone across a particular vertex) is the same.

A violation of the acyclicity condition can be shown to involve a cycle of cut edges that links objects in a circle around the observer. In order for such a cycle to arise, the motion of these objects must be choreographed in a rather restrictive manner. In all our test scenes with random motions, we were unable to produce any examples of this behavior. Thus it appears that in many situations the algorithm to handle composite steps would only be used rarely, if at all.

6 Complexity Analysis

The construction of the visible zone consists of the construction of the radial decomposition, possibly followed by additional local computations, if a representation other than silhouette segments is used. Thus, at least for the case of silhouette segments, the construction time complexity is $O(n \log n)$, and in any case the space requirement is linear.

Perhaps the most important maintenance cost consideration in many practical applications is the cost of changing the motion of the observer. In a real-time fly-through, for example, the motion of the observer may change often. In the kinetic setting, a change in the flight plan of the observer requires a recomputation of all the *certificates* (see [2]) that involve the observer. In the above algorithm, with or without moving objects, the certificates involving the observer are all associated with visible silhouette segments, and so the number of these certificates is proportional to the size of the visible set. Thus when the observer changes its motion, only these certificates need to be recomputed, in constant time each, for a total cost for each change in motion proportional to the size of the visible set.

The cost of updating the structure for a change in the motion of a moving object depends on the number of certificates involving that object, which in turn depends on the representation of the zone. In general, this number may be greater than a constant, due to "look-ahead" techniques for identifying potential cut edge collapses. However, the overall zone representation can still be of linear size, and the average number of connections to any one object across all scene objects is a constant.

For the remainder of this section, we examine the complexity of maintenance updates when the scene is static, and the observer moves along a single trajectory describable by a constant degree pseudo-algebraic expression. The time complexity will always be expressed in terms of the number of elementary steps. A similar analysis can be done for moving objects, but the bounds will not be as good. Better bounds can be proved for moving objects if certificates are allowed between the moving objects and the observer, but we have deliberately avoided any additional certificates involving the observer.

In the section describing the update algorithm, the complexity of these updates was already shown to be no more than that of maintaining the radial decomposition. In certain cases, namely extremely sparse scenes, this cost is unavoidable, since it reflects the number of changes in the visible set. For dense scenes, it is often possible to do better.

In the second section, three approaches to maintaining the visible set were compared according to their *rigidity*, intuitively referring to the fact that the motion of one silhouette segment may require distant silhouette segments to move in order to maintain the global property of interest. A precise bound on rigidity can be expressed as follows.

Let α be an angle, and let V_α be a vertical decomposition for the scene, with α taken as the "up" direction. Let C_α be the cut consisting of those edges of the visibility complex containing the segments of V_α. Let e be a cut edge of C_α. The **upward lift cost** of e is the number of elementary steps of the minimal transformation of the zone of C_α that includes an upward step across the lower vertex of e. The **downward lift cost** of e is defined similarly, with respect to the upper vertex of e.

Definition 5: The **maximum rigidity** of a scene is the maximum of the upward and downward lift costs of all cut edges, across all cuts C_α for α between 0 and 2π.

The maximum rigidity expresses how difficult it is to turn a silhouette segment through full turn. When the observer follows a constant degree pseudo-algebraic trajectory, the total angle swept by any silhouette segment is bounded by a constant. This is the essence of the following overall bound:

Lemma 6.1. *In a scene with maximum rigidity r, the cost of any update is bounded by r^l, for some constant l depending on the degree of the motion.*

In order to relate the maximum rigidity to other more familiar quantities, consider a **dense** scene S, namely one with at most D bitangents connecting to any given object. Unfortunately, this bound is not enough to bound the maximum rigidity by a constant. We also need to bound the number of objects that lie nearly in a row, as follows.

Let $b_1, b_2, ..., b_k$ be a sequence of bitangents, such that any consecutive pair of bitangents has at least one tangent object in common. Let us define the angle of a bitangent sequence to be the minimal angle swept by an upward path in the space of maximal segments that passes through all these bitangents. An **unaligned** scene is a scene where there exist two constants B and ϕ, such that any bitangent sequence of at least B bitangents has angle greater than ϕ.

Lemma 6.2. *In a dense, unaligned scene, the maximum rigidity r can be bounded by the following constant: $D^{2\pi B/\phi}$.*

The source of the exponential in this constant is the fact that D new objects may be reachable each time we connect to another object. However, this is still a constant bound on the maximum rigidity. As a special case, the maximum rigidity must be bounded above by a constant for scenes with a constant lower bound on the angle between two bitangents connecting to the same object.

Theorem 6.3. *In a dense, unaligned scene, or in any scene where the maximum rigidity can be bounded above by a constant, the cost of each maintenance update for an observer moving along a constant degree pseudo-algebraic trajectory is O(1).*

In practice, the constant factors involved may be quite low. For example, in a simulation of a linear movement of the observer through a randomly generated scene of 100,000 objects, where the average number of bitangents connected to each object was approximately 74, the average number of elementary steps involved in an update was approximately 39.

7 Conclusion and Future Work

This paper describes an application of Kinetic Data Structures to a substructure of the visibility complex, in order to maintain the visible set of a moving observer. The underlying visibility structure is called a *visible zone*, which is defined using a slight generalization of the *anti-chain* topological sweep structure introduced in [10]. The combinatorial foundations for maintaining a visible zone of the visibility complex have been explored; this analysis exposes a fundamental difference between maintaining the view of a moving observer in a static scene, and allowing scene objects to move. A general maintenance algorithm has been described, for which the cost of updates when the observer changes its motion is optimal. For a static scene, the cost of maintenance is no more than the cost of maintaining the radial decomposition. If in addition the angular difference between two bitangents connected to a single object can be bounded from below by a constant, the cost of maintenance for an observer moving along a constant degree pseudo-algebraic trajectory is proportional to the number of changes in the visible set.

Maintaining the visible set of a moving observer can be applied to a number of related problems. For example, in order to precompute the "potentially visible set" from a region, an observer can be moved along the boundary of the region, noting whenever new objects are seen. During a real-time simulation, computing the visibility from a region surrounding the current observer position may also be helpful, in order to estimate the spatial extent of the observer's view over the next few frames, allowing pre-fetching of data.

Generalizing this method to other types of zones (for example, keeping track of various orthogonal views) or to non-convex objects (by maintaining relative convex hulls) should be possible. An open question remains as to whether for all scenes there exists a method whose update costs are always proportional to the number of changes in the visible set. We are also investigating generalizations of these approaches in the three-dimensional setting.

8 Acknowledgments

Thanks to my advisor, Leonidas J. Guibas, for encouraging me to study the visibility complex in a kinetic setting. Thanks to my wife Christy, who helped make this work possible. Thanks also to whoever designed visibility itself.

References

[1] B. Aronov, L. J. Guibas, M. Teichmann, and L. Zhang. Visibility queries in simple polygons and applications. In *ISAAC '98*, LNCS, pages 357–366. Springer, 1998.

[2] J. Basch, L. J. Guibas, and J. Hershberger. Data structures for mobile data. *J. Algorithms*, 1:1–28, 1999.

[3] M. Bern, D. Dobkin, D. Eppstein, and R. Grossman. Visibility with a moving point of view. *Algorithmica*, 11:360–378, 1994.

[4] S. Coorg and S. Teller. Real-time occlusion culling for models with large occluders. In *Proc. 1997 Sympos. Interactive 3D Graphics*, 1997.

[5] S. E. Dorward. A survey of object-space hidden surface removal. *Internat. J. Comput. Geom. Appl.*, 4:325–362, 1994.

[6] H. Edelsbrunner and L. J. Guibas. Topologically sweeping an arrangement. In *Proc. 18th STOC*, pages 389–403, 1986.

[7] S. Ghali and A. J. Stewart. Incremental update of the visibility map as seen by a moving viewpoint in two dimensions. In *Proc. 7th Eurographics*, pages 1–11, Aug. 1996.

[8] J. Hershberger and S. Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. *J. Algorithms*, 18:403–431, 1995.

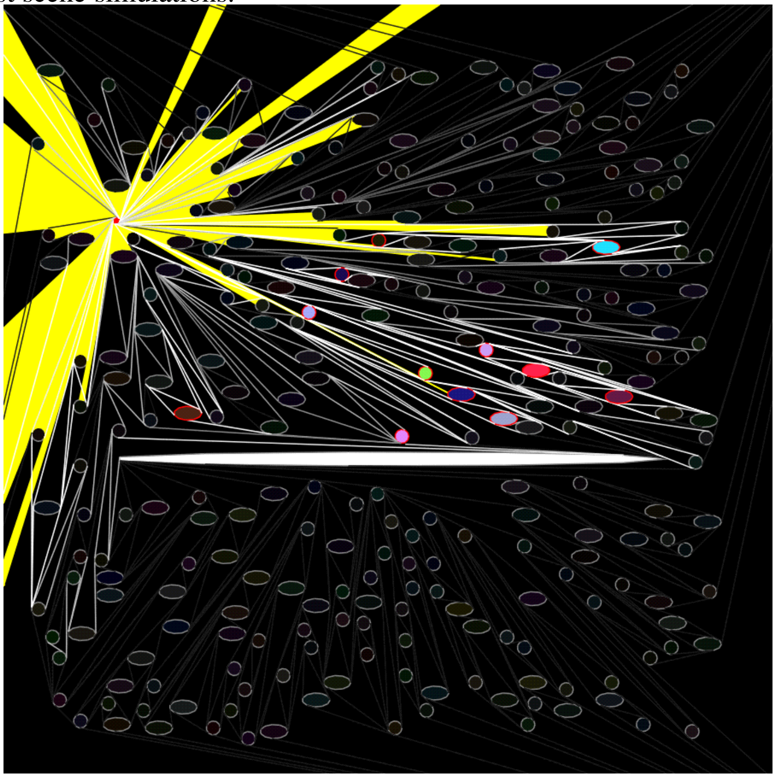
[9] M. S. Paterson and F. F. Yao. Efficient binary space partitions for hidden-surface removal and solid modeling. *Discrete Comput. Geom.*, 5:485–503, 1990.

[10] M. Pocchiola and G. Vegter. Topologically sweeping visibility complexes via pseudo-triangul ations. *Discrete Comput. Geom.*, 16:419–453, Dec. 1996.

[11] S. Rivière. Dynamic visibility in polygonal scenes with the visibility comp lex. In *Proc. 13th ACM Sympos. Comput. Geom.*, pages 421–423, 1997.

Appendix

The following snapshot is taken from one of our test scene simulations.



The visible zone is represented as a **pseudo-triangulation**, as defined in [10].