On Incremental Rendering of Silhouette Maps of a Polyhedral Scene

Li Zhang Alon Efrat* Leonidas J. Guibas†
Olaf A. Hall-Holt

Computer Science Department
Stanford University
Stanford, California 94305
{alon,guibas,olaf,lizhang}@cs.stanford.edu

This paper was presented in SODA 2000

Abstract

A fundamental problem in Computer Graphics is to render a 3-D scene, consisting of a collection of polyhedral objects, while the viewpoint is moving. This problem is known in the graphics community as *incremental rendering*. Since in practice the number of polygons that need to be rendered is in many cases huge — hundreds of thousands or millions, it is sometimes preferable to render only the silhouettes of the objects, rather than the objects themselves. This can have a dramatic effect on the rendering complexity, especially when the objects are finely tessellated. Such an approach is regularly used in the domain of *non-photorealistic rendering*. The hard part in efficiently implementing a kinetic approach to this problem is to realize when the picture undergoes a *combinatorial change* (defined below).

In the first part of this paper, we obtain bounds on a number of combinatorial problems involving the complexity of these events for a collection of k objects, with a total of n edges. We assume that our objects are convex polytopes, and that the viewpoint is moving along a straight line, or along an algebraic curve. The resulting bounds will then depend on both the number of objects, and the number of polygons, in such a way that we can describe the advantages of focusing on the silhouettes of large objects.

We also study the special case that the scene is a *polyhedral terrain*, and present bounds in this case. Based on the upper bound on the combinatorial changes, we obtain algorithms that compute all the changes occuring during a linear motion, (both for general scenes and for terrains) in time (respectively), in an order of $O(k^2 n \log n)$ and $O(kn\beta(k)\log^2 n)$ time. Here $\beta_s(k) = \lambda_s(k)/k$, where $\lambda_s(k)$ is the maximum length of a Davenport-Schinzel sequence of order s with k letters. When s is constant, $\alpha_s(k)$ is almost constant.

1 Introduction

There is an increasing demand in computer graphics applications for rendering large and complex environments involving scenes with many millions of polygons. The computational demands of such a task have to be addressed by both improved hardware and better algorithms. The very high complexity of these environments in terms of simple geometric primitives, such as triangles, is in part an artifact of the traditional rendering pipeline of current graphics systems, which are based on triangle scan-conversion as the basic primitive. In general the number of different objects present in a scene is much less than millions – and the high triangle count is due to the tessellation of more complex curved objects into polyhedral approximations that can be rendered by the hardware.

*This author is supported by a Rothschild Fellowship and by DARPA grant DAAE07–98–C–L027.
†This author is supported by DARPA grant DAAE07–98–C–L027 and by ARO MURI grant DAAH04–96–1–007

Triangle edges have to be handled properly in order to obtain high quality renderings of a scene. There is vast literature in computer graphics on how to deal with edge problems such as jaggies, antialiasing, etc. Yet it is important to realize that not all edges are created equal. Edges in the rendered image separating two different objects are much more likely to be problematic than edges separating two polygons belonging to the same object. Across the former we will have depth discontinuities, a different reflectance function on the two sides, different colors, etc. Across the latter simple edges, interpolatory smoothing techniques work well to simulate the appearance of a smooth surface. The former edges are *silhouette* edges, and they are the topic studied in this paper.

We consider a small number of objects that have been tessellated into a much larger number of triangles. Given a point of view, each object has a *silhouette*, a collection of edges forming closed cycles that separate triangles visible from triangles invisible to the viewpoint. We focus on the geometric structure of these silhouette and their arrangements. If we can compute these silhouette for the viewpoint, and also maintain them as the viewpoint moves around (incremental rendering), not only do we know the most important edges in the image we want to render, but we can also facilitate many other rendering operations, such as ray tracing, shadow calculations, etc.

The hard part in efficiently implementing any algorithm for rendering moving objects is to realize when the picture goes through a *combinatorial change*, defined as a change where either the topology of the rendered picture is changed (e.g. when a hole in their union appears), or when an pair of silhouette edges start or stop intersecting.

The input to the first type of problems we investigate is a set $\mathcal{S} = \{P_1 \dots P_k\}$ of polytopes in 3-D that we have to render, and a perspective point p. As mentioned, we make the realistic assumption that the the number of polytopes k is much smaller than the total number n of vertices of these polytopes. Let S be a small cube centered at the perspective point p. The *shadow* of an object is the perspective projection of the object on S from q[1]. The *silhouette* of a polytope is the boundary of its shadow, which is a convex polygon. The *silhouette arrangement* is the arrangement of the silhouettes. The *silhouette map* is the arrangement on S with the hidden part removed. Formally, we assign a unique color to each object. For any point q on the background sphere S, assign q the color of the first object hit by the ray starting from the perspective point and shooting to q. The boundary of the unicolor regions is exactly the silhouette map. The *union-of-silhouettes*, (abbreviated *uo-silhouette*) is the union of all the shadows on S.

We describe combinatorial bounds on these geometric structures, in each of the following three cases

- *Static view-point* — the viewpoint p is static.
- *Linear motion* — p moves along a straight line, and the goal is to bound the number of combinatorial changes.
- *Algebraic motion* — p moves along an algebraic curve, and the goal is to bound the number of changes each of the structures goes through.

In this paper we present the following bounds for an arbitrary collection of convex polytopes.

We also investigate a special case of terrains. We consider a terrain with k mountains with total complexity

structure	Static viewpoint	linear motion	Algebraic motion
silhouette arrang.	$\Theta(kn)$	$\Theta(k^2 n)$	$\Theta(kn^2)$
silhouette map	$\Theta(kn)$	$\Theta(k^2 n)$	$\Theta(kn^2)$
union of silhouette		$O(k^2 n)$ $\Omega(kn + k^3)$	$O(kn^2)$ $\Omega(n^2 + k^2 n)$

n. Roughly speaking, a mountain is a up-convex body with the base on the xy-plane. For such a terrain

[1] We ignore the effect of the edges of S, and treat shadows projected on S as being projected on a plane

and a vertically moving perspective point, we are able to obtain a roughly $\Theta(kn)$ bound on the number of combinatorial changes. It is better than the $\Theta(n^2)$ bound on the number of changes of the aspect graph, where the lower bound can be achieved by two mountains.

One key lemma we use to achieve the upper bounds is to bound the number of so called EEE events, i.e. the number of times when the shadow of three edges on the silhouettes come together. This is in turn by bounding the number of lines that passes through the perspective point and touches three convex polytopes. We show that for a linear moving point, this number is linear to the total complexity of these three convex polytopes instead of quadratic and thus obtain better upper bounds.

Based on the upper bound of the combinatorial changes, we can obtain algorithms that compute the all the changes occuring during a linear motion, (both for general scenes and for terrains) in time (respectively), in an order of $O(k^2 n \log n)$ and $O(kn\alpha(n)\log^2 n)$ time.

Related results: Similar problems were investigated both analytically (usually in the Computational Geometry community) and empirically (in the Computer Graphics community). Among the analytic results, de Berg, Halperin, Overmars and van Kreveld, [dBHOvK97], described a list of results regarding the complexity of the aspect graph for different scenarios, and its relations to complexity of arrangements. Barequet et al. [BDG$^+$99] showed how to use the BAR-tree to obtain fast rendering of silhouette of a (not necessarily convex) polytope.

Other works [LE97, Cro77] in the graphics community also use shadows and silhouettes as a means to simplify the description of a complicated environment. Silhouettes are also useful in collision detection [BV95], and other applications.

2 Lower bounds

2.1 Silhouette structures from a static point From a fixed point, the shadow of each polytope is a convex polygon on the background plane. Aronov and Sharir [AS97] showed that the arrangement of k convex polygons with n vertices has complexity $\Theta(kn)$ and the complexity of the boundary of the union of k convex polygons with n vertices is $\Theta(k^2 + n\alpha(k))$ in the worst case. These bounds yield the tight bounds for silhouette arrangements and union-of-silhouettes. For the silhouette map, we will construct an example to show the complexity of $\Omega(kn)$ and thus obtain a tight bound of $\Theta(kn)$ for silhouette maps.

Figure 1 (i) shows an example of our construction of k fat convex polygons, and figure 1(ii) shows an enlargement of a neighborhood containing k polygon corners. The goal of the construction is to create n/k corner neighborhoods with silhouette map complexity k^2 each, yielding a total complexity of $(n/k) \times \Omega(k^2) = \Omega(kn)$. The corner of a polygon consists of the corner vertex, and two incident edges, which we will call the left and right edges. Let us focus on a particular corner neighborhood N (such as the one depicted in figure 1(ii)). To obtain the desired complexity, we would like to ensure that the left edge of each polygon corner contributes a vertex to the silhouette map at the right edge of all preceding polygons in the depth order. Intuitively, we seek a set of nearly parallel left edges with increasing slope, such that each left edge lies to the right of the endpoints of the previous left edges (in the depth order).

To ensure that this is possible, consider a sequence of tangents to the unit hyperbola in the first quadrant of the plane ($y = \sqrt{x^2 - 1}$). Let the tangent points be a_i, where they are given in order of increasing x-coordinate. For a tangent to the hyperbola at a_i, let b_i be the intersection of the tangent with the asymptote $y = x$. Let the b_i be our corner vertices, and let the left edge from b_i extend down toward a_i. Thus a left edge must pass below the corner vertices of all preceeding polygon corners. We still need to construct the spacing among the a_i. First let all the right edges proceed to the right, with slope $0 < \alpha < 1$. Now let a_{i+1} be the intersection of the previous right edge with the hyperbola. That is, we make each right edge go up toward the hyperbola, and choose all tangent points except the first one to be the intersections of the right edges with the hyperbola. Thus, we also ensure that any left edge will be above the tangent points corresponding to all previous left edges, and hence visibly intersecting all previous right edges.

To construct the k polygons from such a neighborhood, we place n/k rotated copies of the neighborhood near the vertices of a sufficiently large n/k-gon, and connect corresponding left and right edges. In order for this to work, we need to make sure that the corner angles are sufficiently large. In the neighborhood construction the position of the first tangent point and the parameter α were left unspecified. These together determine a lower bound on all the corner angles, which can be anything less than π, so we can always make them large enough.

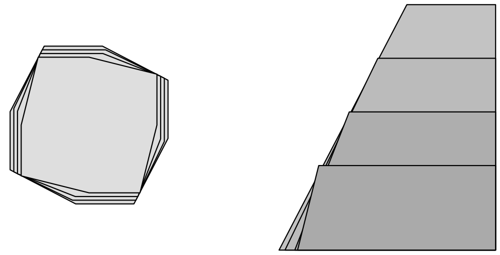

Figure 1: The lower bound construction of the silhouette map. Left: The global arrangement. Right: Close up of a corner.

In summary, we have that:

THEOREM 2.1. *In the worst case, the silhouette arrangement, silhouette map, and uo-silhouette have complexity* $\Theta(kn)$, $\Theta(kn)$, *and* $\Theta(k^2 + n\alpha(k))$, *respectively.*

2.2 Lower bounds for a moving perspective point. Now that we have tight bounds for a fixed viewpoint, we now consider the number of changes for a moving viewpoint. First, we will give a generic construction for lower bounds. Suppose that A is the maximum complexity from a static perspective point p. There exists a small ball B around that point such that for any point q inside B, the complexity of the structure from q has complexity A. For linear motion, we can put k line segments inside B so that when we move the viewpoint p linearly, the shadow of those k line segments sweep over the silhouette structure. This way, we create kA changes. For algebraic motion, we take the classical example of a quadratic curve intersecting a convex n-gon n times. Then, this way, we can create nA changes for algebraic motion. Combining this approach with the maximum possible complexity for a static point, we now have the following:

THEOREM 2.2. *For linear motion, the silhouette arrangement, silhouette map, uo-silhouette can change* $\Omega(k^2n)$, $\Omega(k^2n)$, $\Omega(k^3 + kn\alpha(k))$ *times, respectively. For algebraic motion, the lower bounds are* $\Omega(kn^2)$, $\Omega(kn^2)$, *and* $\Omega(k^2n + n^2\alpha(k))$, *respectively.*

3 Upper bounds

In the paper of de Berg et al. [dBHOvK97], they bound the number of different views to a scene of k convex objects. They derived an upper bound of $O(kn^2)$ on the number of surface patches forming a partition of the viewpoint space into cells with same view. Since a constant degree algebraic curve can intersect such a surface patch only a constant number of times, we can obtain the following upper bound by borrowing Theorem 5.2 of [dBHOvK97].

THEOREM 3.1. *For a scene that consists of k convex polyhedra with total complexity n, there can be* $O(kn^2)$ *combinatorial changes to the silhouette-arrangement when the perspective point moves along an algebraic curve.*

However, when the point moves linearly, we may obtain better bounds on the number of changes of the silhouette arrangement.

THEOREM 3.2. *For a scene that consists of k convex polyhedra in general position with total complexity n, there can be* $O(k^2n)$ *combinatorial changes to the silhouette-arrangement when the perspective point moves linearly. This bound holds also for the number of changes in the silhouette-map.*

For a convex polyhedron P, a line ℓ is said to be *tangent* to P if ℓ intersects P only on the boundary of P. When ℓ intersects P at exactly one point, it is called *strictly tangent* to P. The following simple fact is useful.

Fact 3.1. *A line ℓ is strictly tangent to P if and only if ℓ is strictly tangent to $P \cap \gamma$ for any plane γ that contains ℓ. In addition, this is the case if and only if there* ***exists*** *a plane γ' that contains ℓ so that ℓ is strictly tangent to $P \cap \gamma'$.*

The key step to prove Theorem 3.2 is to bound the number of lines that touch a given line and three convex polytopes. We will show that the number of lines is linear in the total complexity of those three convex polytopes.

Lemma 3.1. *For any given line ℓ and three convex polyhedra P_1, P_2, and P_3 in general position, the number of lines that touch ℓ and are tangent to P_1, P_2, and P_3 is $O(|P_1| + |P_2| + |P_3|)$.*

In what follows, without loss of generality, we assume that the line ℓ is the z-axis. Consider the family of planes that pass through the z-axis. We parametrize them according to the angles they make with the x-axis: $\Gamma = \{\gamma(\theta) : 0 \leq \theta < \pi\}$. For a convex polyhedron P, denote by $P(\theta)$ the intersection between P and $\gamma(\theta)$. Clearly, $P(\theta)$ is a convex polygon lying on $\gamma(\theta)$. For two convex polyhedra P, Q and any θ, $0 \leq \theta < \pi$, let us define the slope function $\phi_{P,Q}(\theta)$ (or $\phi(\theta)$ if P, Q are clear from the context) to be the slope of the lower outer bi-tangent between $P(\theta)$ and $Q(\theta)$. Let $\phi_{P,Q}$ denote the graph of the function $\phi_{P,Q}(\theta)$ as θ varies.

First, we observe that:

Lemma 3.2. *The graph $\phi_{P,Q}$ consists of $O(|P| + |Q|)$ arcs, each of which is a low degree rational function of* $\tan(\theta)$.

Proof. For any particular θ, the cross intersection $P(\theta)$ is a convex polygon. A vertex of this polygon is either a vertex v of P or an edge $e \cap \gamma(\theta)$ for an edge e of P. For all the vertices created by the same edge, we think of them as a single vertex moving on a low degree rational curve as the plane rotates. If a bi-tangent is defined by the same pair of vertices, then the slope is just a rational function in terms of $\tan\theta$. When can a breakpoint be created? There are two possibilities: first, when either a previous vertex is deleted or a new one is created; second, when three vertices are collinear and the line that passes through them is a bi-tangent line. Clearly the first type of events can happen at most $O(|P| + |Q|)$ times.

For the second type of events, suppose that for some θ, $p_1, p_2 \in P(\theta)$ and $q \in Q(\theta)$ are collinear and the line ℓ' determined by p_1p_2q is a bi-tangent line to $P(\theta), Q(\theta)$. First of all, p_1, p_2 must be adjacent vertices on $P(\theta)$ by convexity. Further, by the general position assumption, ℓ' is strictly tangent to $Q(\theta)$ at q, which implies that ℓ' is strictly tangent to Q according to Fact 3.1.

Consider the edges $e_1, e_2 \in P$ that correspond to p_1, p_2, i.e. $p_1 = e_1 \cap \gamma(\theta)$ and $p_2 = e_2 \cap \gamma(\theta)$. We know that e_1, e_2 must be on the same face, say f (Figure 2). Consider the plane β that contains f. Observe that $\beta \cap Q$ is again a convex polygon. Suppose that β intersects ℓ at point r. We claim that rq is strictly tangent to $\beta \cap Q$. This simply follows from the fact that ℓ' is indeed strictly tangent to Q and by Fact 3.1. From any point, we can draw at most two tangent lines to another convex polygon. This is to say that for any face f of P, there can be at most two such points q on Q so that for any $p_1, p_2 \in f \cap \gamma(\theta)$, q is collinear with p_1, p_2 as a bi-tangent line. Therefore, the second type of events can happen at most $O(|P| + |Q|)$ times. This concludes the proof of the lemma.

According to Lemma 3.2, we now can prove Lemma 3.1.

Proof of Lemma 3.1. For P_1, P_2, P_3, we plot two functions $\phi_1 = \phi_{P_1,P_2}$ and $\phi_2 = \phi_{P_1,P_3}$. By Lemma 3.2, ϕ_1 consists of $O(|P_1| + |P_2|)$ low-degree rational curves and ϕ_2 consists of $O(|P_1| + |P_3|)$ such curves. The graphs ϕ_1 and ϕ_2 can intersect at $O(|P_1| + |P_2| + |P_3|)$ points. For a line ℓ' that touches ℓ, consider the plane γ that is determined by ℓ and ℓ'. For ℓ' to be tangent to P_1, P_2, P_3, ℓ' must be a common tangent to $P_1 \cap \gamma, P_2 \cap \gamma$, and $P_3 \cap \gamma$. This is related to an intersection point between ϕ_1 and ϕ_2. Therefore, in total this is bounded by $O(|P_1| + |P_2| + |P_3|)$. Of course, there are different types of tangents. But this is not a problem as there are constantly many combinations (16 combinations to be precise). We therefore proved Lemma 3.1. **QED**

Next, we proceed to prove Theorem 3.2.

Proof of Theorem 3.2. For the silhouette to change, there are three cases.

1. The **first type** occurs when the viewpoint crosses a plane supporting a facet.

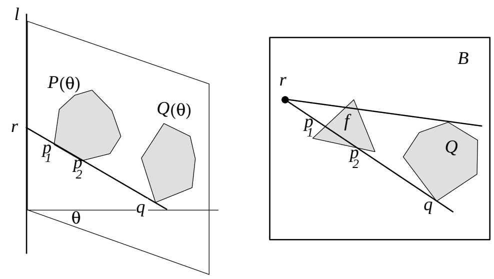

Figure 2: Combinatorial change to a bi-tangent

2. The **second type** occurs when the viewpoint crosses a plane determined by a vertex v and an edge from different objects P_i.

3. The **third type** occurs when there is a ray from the viewpoint that touches three edges on three different objects.

The first type of events are clearly bounded by $O(n)$. When this type of event happens, what happens to the silhouette is that an edge is replaced by two other edges, or two edges become collinear and are replaced by a single edge. Each such event causes at most $O(k)$ changes to the silhouette. Therefore, this kind of event causes $O(kn)$ changes.

For the second type of event, consider the double cone C forms by the union of all lines passing through v and P_i. Second type events can happen only when the viewpoint on ℓ crosses the boundary of C, which can clearly happen at most twice. This implies that the second type of event can happen at most $O(kn)$ times. Once such event happens, it can cause $O(1)$ changes to the silhouette as it makes a vertex cross an edge in the silhouette.

The third type is the hard case, when there is a line from the viewpoint that goes through the boundary of three polytopes. By Lemma 3.1, we know that this is bounded by

$$\sum_{i,j,k} O(|P_i| + |P_j| + |P_k|) = O(k^2 n)\,.$$

This concludes the proof of the theorem. **QED**

Notice that we actually bound the number of changes of the silhouette arrangement, and therefore the silhouette map and uo-silhouette. For the silhouette arrangement and silhouette map, these upper bounds match the lower bounds of Theorem 2.2. Unfortunately, there still is a gap remaining between the lower and upper bounds for the uo-silhouette.

4 Terrain with k mountains

As another application of Lemma 3.2, we may bound the number of changes of the silhouette for special terrains, namely, terrains that consist of mountains. A convex object M is called a *mountain* if for any vertical line ℓ, $\ell \cap M = [0, a)$ for some $a \geq 0$. Or intuitively, a mountain is an upper-convex object whose base is on the xy-plane.

Cole and Sharir showed in [CS89] that for a viewpoint moving vertically in a terrain with n vertices, the visibility changes $\Theta(n^2)$ times, beating the naive bound of n^3. Having a small number of mountains does not help to reduce the lower bound there as the $\Omega(n^2)$ lower bound can be constructed by using *two* mountains. However, for a viewpoint moving vertically, we may obtain a roughly $\Theta(kn)$ bound on the number of changes of *uo-silhouette* of terrains with k mountains.

THEOREM 4.1. *The silhouette changes $\Omega(kn)$ and $O(kn\beta_s(k))$ times*[2] *for a point moving vertically in a terrain with k mountains and n vertices where s is a constant depending only on the degree of the motion.*

[2] See the introduction for definitions

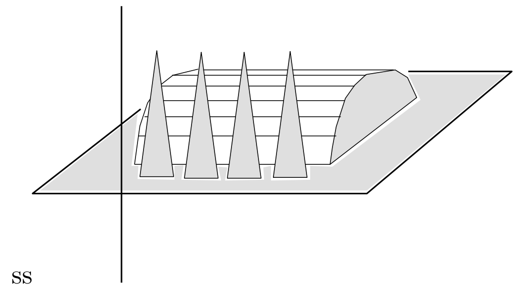

Figure 3: Lower bound construction for terrain.

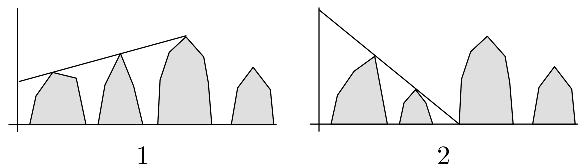

Figure 4: Lines that touch z-axis and three mountains.

Proof. For the lower bound, consider the picture in which we have cylindrical mountain P with n sides. In front of the mountain, we have another k peaks(skinny tetrahedra). Then when the viewpoint moves vertically, each time it crosses a plane supporting a facet of P, it causes $\Theta(k)$ changes to the silhouette. In total, the changes are $\Omega(kn)$.(Figure 3)

For the upper bound, let us again consider the three types of event used previously. The first two cases are bounded by $O(kn)$ as we have seen from the argument for general convex polytopes.

To bound the third event type, we now need to count the number of lines that touch the z-axis and three mountains and avoid all the other mountains. Again consider Γ, the family of the planes that pass through the z-axis. There are two cases where such a line can appear. One case is when there is a line that is tangent to three mountains from the same side and avoid all the other ones; and the other case is when there is a line that touches the base of one mountain and tangent to two other mountains. (Figure 4)

For the first case, we have a combinatorial change happens in the convex hull of the $P_i(\theta)$'s as θ increases from 0 to π. For any θ, $P_i(\theta)$ is said to be to the left(right) of $P_j(\theta)$ if they both are not empty and the horizontal ray starting from the origin and shooting to the left hits $P_i(\theta)(P_j(\theta))$ first. Since the P_i's are mountains, they have a consistent ordering, i.e. P_i cannot be both to the left and to the right of P_j.

Now, let us focus on one polytope, say P_1. Define a function $\phi_j(\theta)$ as the slope of the outer bi-tangent of $P_1(\theta)$ and $P_j(\theta)$ if $P_j(\theta)$ is to the left of $P_1(\theta)$ and undefined otherwise. Also define $\xi_j(\theta)$ in the same manner for the objects to the right of P_1. As we have shown in Lemma 3.2, the functions ϕ_j, ξ_j consist of $O(|P_1| + |P_j|)$ algebraic arcs. Just as in the case of combinatorial changes for moving points, a change on the convex hull can be charged to the overlay of the lower envelope formed by the ϕ_j's and the upper envelope formed by the ξ_j's. By the standard argument, the complexity is bounded by $\beta_s(k) \sum_j (|P_1| + |P_j|)$. Summing this up for all the polytopes, we have:

$$\sum_i (\beta_s(k) \sum_j (|P_i| + |P_j|)) = kn\beta_s(k)\,.$$

The second case is simpler as we can charge it to the changes of the lower envelopes of the slope functions of the inner common tangents. The details are omitted in this abstract.

To summarize, the number of changes of the silhouette for a vertically moving point is $O(kn\beta_s(k))$ in a terrain with k mountains and n vertices.

5 Algorithms

We can apply the above combinatorial bounds for linearly moving perspective point, both in general scene and terrains, to devise algorithms to compute different silhouette structures for a point that moves along a given line.

For the general scene that consists of k convex objects, imagine a plane γ that rotates around z-axis from 0 to π. (again, we assume that the given line is the z-axis.) Consider the cross intersection of γ and the convex polytopes, which are a set of convex polygons. When γ rotates, those polygons deform and move. We wish to detect when there forms a common tangent line to some three convex polygons during the motion. This can be solved by tracking the bi-tangents of each pair of convex polygons. Further more, for each particular object, we maintain a list of tangents that touch it, sorted by their slopes. This way, we can detect all the lines that touch three convex polyhedra and z-axis. And it can be seen that the events that happen in our algorithm can be counted exactly as those counted by Theorem 3.2 and Lemma 3.2. The processing time for each event is $O(\log n)$. We can apply similar algorithms to terrains. The only difference is that instead of maintaining the sorted list of all the tangents to an object, we maintain the lowest(or highest) one according to which side it is as described in the proof of Theorem 4.1. This can be done by the kinetic tournament data structure presented in [BGH97].

Thus, we have that:

THEOREM 5.1. *For k convex polyhedra with n vertices in total and a given line ℓ, in $O(k^2 n \log n)$ time, we can compute the partitioning of ℓ into intervals so that from the perspective points in the same interval, the silhouette structures remain the same. When it is a terrain with k mountains and ℓ is a vertical line, such subdivision can be computed in $O(kn\beta_s(k)\log^2 n)$ time.*

6 Open Questions

The main open questions are:

1. The bound in Theorem 3.2 is not tight. The suspected right bound is $\Theta(k^3 + kn)$.
2. The corresponding bounds for algebraic motions are $\Omega(k^2 n + n^2)$ and $O(kn^2)$. Again, there is a gap of $O(k)$.
3. We described algorithm to compute all the changes for a perspective point moving on a given line in $O(k^2 n)$ time. Can we do it in an on-line manner, for example, under kinetic framework?

Acknowledgment

We thank Danny Halperin who suggested the proof of Theorem 3.1. We also thank Tamara Munzner and an anonymous referee for pointing us to some relevant literature.

References

[AS97] Boris Aronov and Micha Sharir. The common exterior of convex polygons in the plane. *Comput. Geom. Theory Appl.*, 8:139–149, 1997.

[BDG+99] G. Barequet, C.A. Duncan, M.T. Goodrich, S. Kumar, and M. Pop. Efficient prespective-accurate silhouette computation. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 417–418, 1999.

[BGH97] J. Basch, Leonidas J. Guibas, and J. Hershberger. Data structures for mobile data. In *Proc. 8th ACM-SIAM Sympos. Discrete Algorithms*, pages 747–756, 1997.

[BV95] W. Bouma and G. Vaněček Jr. Velocity-based collision detection. In A. Paeth, editor, *Graphics Gems V*. Academic Press, 1995.

[Cro77] F. C. Crow. Shadow algorithms for computer graphics. *Comput. Graph.*, 11(2):242–248, 1977.

[CS89] R. Cole and Micha Sharir. Visibility problems for polyhedral terrains. *J. Symbolic Comput.*, 7:11–30, 1989.

[dBHOvK97] M. de Berg, D. Halperin, M. Overmars, and M. van Kreveld. Sparse arrangements and the number of views of polyhedral scenes. *Internat. J. Comput. Geom. Appl.*, 7:175–195, 1997.

[LE97] David Luebke and Carl Erikson. View-dependent simplification of arbitrary polygonal environments. In *Proc. SIGGRAPH '97*, Computer Graphics Proceedings, Annual Conference Series, pages 199–208. ACM SIGGRAPH, ACM Press, August 1997.