Exploiting Coherence of Shadow Rays

Andrzej Łukaszewski*

Institute of Computer Science, University of Wrocław
currently at:
Max Planck Institut für Informatik in Saarbrücken.

Abstract

We present independent method to reduce the number of shadow ray tests. It can be used with standard acceleration ray tracing algorithms. Our method is conservative and produces the same results. We test just one shadow ray in modified scene insted of group of rays in original scene. If our ray is not obstructed in modified scene we know that all the rays in this group are not obstructed. The results give rise to many applications when there are many shadow rays e. g., in bidierectional path tracing or stochastically sampling area light sources. The formal proof of the method uses formalism of Minkowski operators and they can also be used in implementation details.

CR Categories: I.3.7 [Computer Graphics]: Three–Dimensional Graphics and Realism—Raytracing;

Keywords: rendering, ray tracing, offseting, soft shadows, ray coherence

1 Introduction

Shadow rays are widely used in visibility checking and it is the most time consuming part of many algorithms for global illumination. Monte Carlo methods are based on the ray tracing principles and use shadow rays extensively. The general concept of shadow rays includes not only the rays from shaded point to the points on light sources but also the rays between two paths in bidirectional path tracing [10]. Shadow rays are also applied in some of the radiosity methods to calculate form factors where visibility has to be determined. What makes them different from other rays is that we have just the ray segment with fixed end points and we only require the boolean information whether the ray is obstructed or not. We are not interested in the exact intersection calculation.

In ray tracing literature there has been many solutions how to exploit spatial coherence of objects in the scene to reduce the costly calculations of ray intersections and these methods include bounding boxes, grids, octrees or BSP-trees. There has been also some work devoted specially to shadow rays e. g., light buffers [4]. For the review of the methods see [3].

We propose a novel technique which exploits the spatial coherence of shadow rays in a new way. Therefore it is possible to use it together with standard ray tracing acceleration methods mentioned above and to obtain better results when time is critical. There are also some recent methods using studies of visual perception to guide the calculations into important regions. In this way we avoid calculating effects which are not perceived by humans. Our technique is also not contradictory with them. Instead of tracing groups of rays we test just one ray in a modified scene with expanded objects. If this one ray is not obstructed we guarantee that all the rays from this group are not obstructed. What is advantegous is that we can use any standard acceleration techniques in both original scene and the modified one.

Our method is based on offsetting operation and its generalization using Minkowski operators. Part of the work presented here has also been published in [7, 8] where some more details of experiments for penumbra generation can be found.

*Institute of Computer Science, University of Wrocław, ul. Przesmyckiego 20, 51-151 Wrocław, Poland, e-mail: anl@ii.uni.wroc.pl, WWW: http://www.ii.uni.wroc.pl/~anl

2 Notation

We will use the notation $R(p, q)$ for the ray segment with endpoints p and q and $B(c, r)$ for the ball centered at the point c and with the radius r.

We define here the way to expand objects and construct the additional scene for tracing shadow rays. Minkowski operators (e. g., as in [6]) and solid offsets provide a convenient way to express set operations. These operators give us formalism and allow as to prove that the visibility informations we obtain will be exactly the same. They also give a reference to similar methods which are used in robot motion planning.

Definition 1 (Minkowski sum and difference) *For two subsets A and B of Euclidean vector space,* Minkowski sum *and* difference *are defined as:*

$$A \oplus B = \{a + b \mid a \in A, b \in B\} \ , \quad (1)$$
$$A \ominus B = \{a - b \mid a \in A, b \in B\} \ . \quad (2)$$

Using Minkowski operators to expand objects in the case of spherical light sources will simplify to the operation of solid offsetting as defined in [1, 9].

Definition 2 (Solid offset) *For an object Q and a distance d, a* solid d-offset $\mathcal{O}_d(Q)$ *is defined as the set of points that are not farther than d from Q, i. e.,*

$$\mathcal{O}_d(Q) = \{p \mid \exists\, q \in Q : d(p, q) \le d\} \ . \quad (3)$$

it can also be expressed using Minkowski operators as:

$$\mathcal{O}_d(Q) = Q \ominus B(0, d) = Q \oplus B(0, d) \ . \quad (4)$$

3 Rays with common origin

First let us start with classical shadow rays. For non-point light sources we sample them stochastically to approximate the visibility angle. We have the collection of shadow rays with common origin for a given point we are shading. The other ends of rays belong to the light source and they are spatially coherent.

All the rays belong to the visibility cone which is included in the offset $\mathcal{O}_r$ of the ray segment $R(c, p)$ what is depicted on figure 1. If we shrink the light source L to a point c and, at the same time, increase the occluding object Q by r the radius of L we can assure that if the ray segment $R(c, p)$ does not intersect the increased object (which is not the case on the figure) then all the rays are not occluded. More formally we can formulate the following lemma.

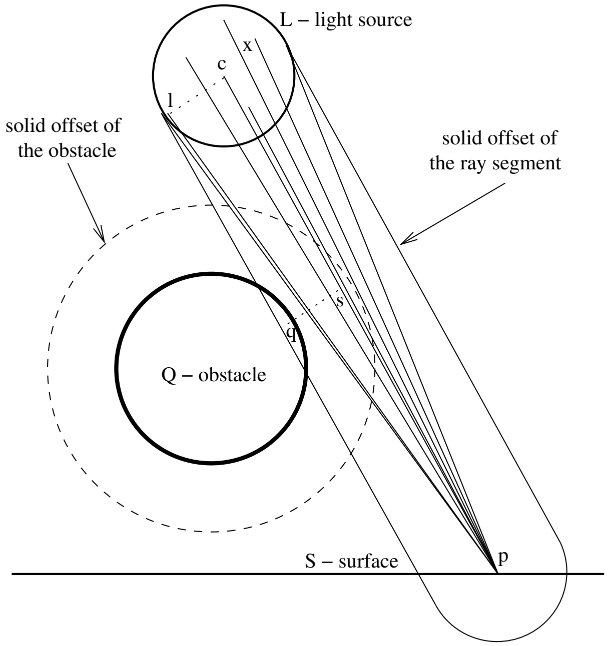

Figure 1: Checking visibility of bundle of rays

Lemma 1 *Let L be a convex or star-convex set respective to a point $c \in L$. If the ray $R(c, p)$ does not intersect $Q \ominus (L \ominus \{c\})$ then $\forall l \in L$ the ray $R(l, p)$ is not obstructed (shadowed) by Q.*

Proof (by contradiction): Let $C = \bigcup_{x \in L} R(x, p)$ denote the visibility cone of the set L as seen from point p. If there would be a ray $R(l, p)$ obstructed by Q then $C \cap Q \neq \emptyset$ but since the set L is star-convex respective to the point c we have $C \subset R(c, p) \oplus (L \ominus \{c\})$. Thus, we have $(R(c, p) \oplus (L \ominus \{c\})) \cap Q \neq \emptyset$ which means that there are points $s \in R(c, p)$, $l \in L$ and $q \in Q$ such that $s + (l - c) = q$. But this is equivalent to $s = q - (l - c)$ which means that $R(c, p) \cap (Q \ominus (L \ominus \{c\})) \neq \emptyset$. Hence, the ray does intersect the expanded object.

3.1 Discussion

This lemma allows us to trace the collection of coherent rays at the cost of one intersection since the scene with increased objects can be constructed in preprocessing time. The exact offsetting operation for spheres is just increasing its radius but for more complex objects it can be expensive. However we can always take a simpler object which includes our increased one or to do increasing of just the bounding objects. Implementations can use couple of methods. In the second modified scene with expanded objects we can create :

- Only bounding boxes of expanded objects and acceleration structures for ray tracing. If the bounding box is hit we do rest of calculations in original scene. This is quite universal and easy to implement.

- The exact expanded objects or larger ones with their bounding boxes and acceleration structures. The intersection test for them should not be significantly more expensive than the intersection test for original object. This way our test will succeed in more cases but there is a trade of between complexity of expanded objects (their intersection test) and how close they are to expansion from lemma. If test fails and there is an intersection we have to do ray tests in orginal scene to guarantee the same results or to use other approximate methods.

The usage of general Minkowski operators is more effective for arbitrary shaped light than the usage of solid offsets. If we enclose a linear or planar shape in a bounding sphere, we can handle it with solid offsetting as described above. However, the approximate shadow volume is unnecessary large. The advantage of Minkowski operators compared to solid offsets becomes clear when we look at non-spherical light sources. For instance, let us consider a flat light source lying parallel to the plane xy. With the help of Minkowski operators the extended objects are bounded by the original bounding boxes extended only in dimensions x and y according to the size of the light source and a chosen central point c. Figure 2 depicts the scenario for a linear light source. If we used just solid offsets, we would have to put the light source into a ball and expand all the bounding boxes of the objects equally in all directions.

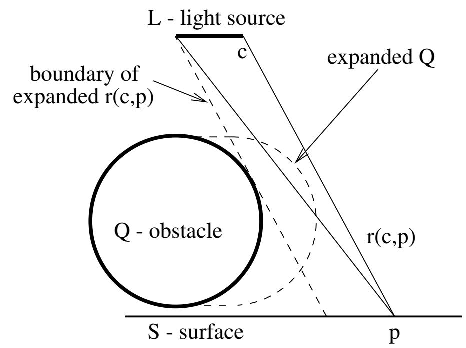

Figure 2: Linear light source and x–axis extension of obstacle

3.2 Multiple light sources and optimization

Our method can handle multiple light sources. We can create for each spatial light source the additional scene in which we will perform the test for given shadow rays. We can also create just one expanded scene for all shadow ray tests taking into account the biggest expansion. If the light sources are of the similar size taking only one scene is as efficient as multiple ones and is less memory consuming.

We also considered further optimization of the extension operation using the fact that the rays belong to the cone and using distance relations between point, light source and obstacle. If we calculate the minimum distance from the scene to the light source and diameter of the scene then we can shrink expansion by a factor and basically we obtain smaller extensions. Namely we can substitute:

$$Q \ominus t_{max} \cdot (L \ominus \{c\})$$

in lemma 1 for the expanded object using t_{max} defined by distance relations:

$$t_{max} = \frac{Scene\_Diameter}{Scene\_Diameter + Minimal\_Scene\_to\_Light}$$

4 Rays with coherent origins

We can use the same framework for exploiting coherence of other shadow rays. Let us collect rays (e. g., in bidirectional path tracing) such that we have the set of rays with origins in the balls A and B of radius r. Then we can also guarantee that none of the rays is occluded by Q if the ray $R(p, q)$ does not intersect Q increased by $\mathcal{O}_r$ offsetting. Following is the formal lemma which is illustrated in Figure 3.

Lemma 2 *Let p, q be the given points and $r > 0$ a given distance. If the ray $R(p, q)$ does not intersect $Q \ominus B(0, r)$ then none of the rays $R(s, t)$ for $s \in B(p, r)$ and $t \in B(q, r)$ intersects Q.*

The proof of this lemma can be obtained using the same method which was used for proving lemma 1.

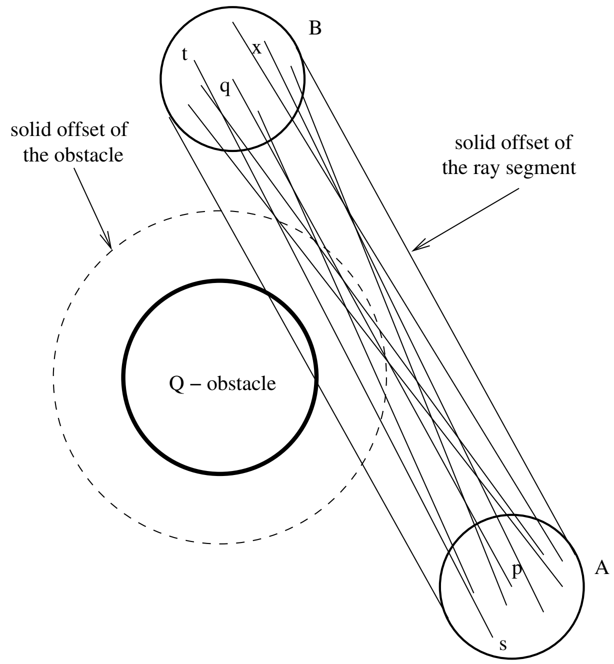

Figure 3: Checking visibility of coherent bundle of rays

Using lemma 2 we can trace at once the whole bundle of rays which have both the starting and ending points coherent and included in respective balls of given radius. To check that none of the rays is obstructed by any object Q from the scene we test just one ray in extended scene, only if the test fails then we have to do the normal tests for each ray or to try the test once again for a smaller bunch of rays. If the test succeeds we are often n times faster, where n is the number of rays in a bundle, since the cost is in most cases is the same.

5 Fast penumbra method implementation

We have incorporated the method into a quite efficient ray tracer [2] to detect penumbra regions and limit the use of expensive stochastic sampling for soft shadow calculation. It can be incorporated into any ray tracing kernel.

We compare the run times of the new method to the run times of traditional ray tracing without penumbra and classical stochastically sampled area light sources. More details are added as remarks in Table 1. We use several examples of geometric data sets: a simple scene with five *cylinders* and spheres, a complex *molecule* transformed from the Brookhaven Protein Data Bank, a *bust* modeled as a mesh of triangles, and the fractal *balls* and the *rings* from the SPD-benchmark [5]. All measurements are done on a Sun SparcEnterprise 4000, 168 MHz, 1.125 GByte RAM and reflect the real time of the ray tracing loop without preprocessing.

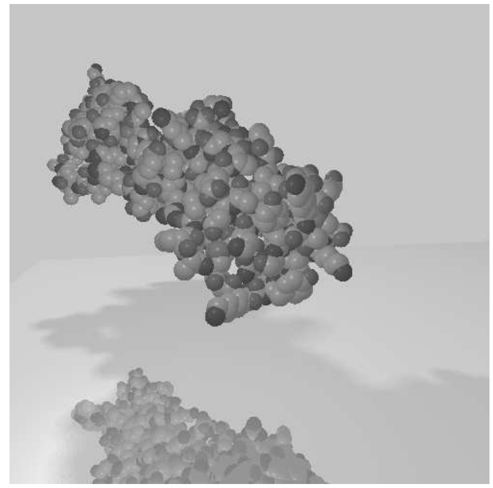

Figure 4: Penumbra for a compact molecule.

As long as the scene description is small (simple scene) the additional memory required for the fast penumbra calculation is negligible. For the larger scenes at most twice as much memory is required.

The speedup which is obtained with the new method depends on the geometry, especially on the size of the light sources and on the size of the visible penumbras. We sampled stochastically area light sources using 32 shadow rays for each point and area light source. It is minimal small amount to obtain any good results. When using larger number of rays our metod is better. The speedup for the fast method ranges from 1.76 to 7.83 depending on the size of the light sources: the smaller the light sources, the better the improvement in run time. The method adapts well to the size of the penumbra. If it is small there are less calculations and the speed–up factor is larger. If penumbra region is quite large the calculations are necessary anyway. Using only the stochastic method the cost in both cases is the same i. e., unnecessary large.

6 Conclusion

We presented a new framework to speed–up the intersection tests for shadow rays using the notion of Minkowski operators and solid offsets. We proved formally that the method works correctly. It can be used in any ray tracing method when we can group the shadow rays with coherent origins.

We made tests in the case of stochastically sampled area light sources and the method essentially renders the same images in shorter time. The effectiveness of the algorithm depends on the particular geometric data set. On average, the presented sample scenes could be rendered two times faster compared to the run time of classical stochastic method. However, if the penumbra regions are small respective to the visible regions in the scene, much higher speedups can be obtained. The additional memory requirements never exceeded in our experiments a factor of two since we constructed only one additional scenery for all light sources.

References

[1] R. Farouki. Exact Offset Procedures for Simple Solids. *Computer Aided Geometric Design*, 2(4):257–279, 1985.

		Cylinders	Balls4	Rings	Molecule	Bust
a)	simple ray tracing	0.91	5.42	1.83	1.34	3.15
b)	stochastic ray tracing	12.05	64.21	32.35	8.25	37.47
	fast penumbra					
c)	standard	6.10	43.48	15.11	5.66	32.46
d)	optimized detection	5.72	26.81	10.54	5.02	24.02

Table 1: Run times in seconds for different images and algorithms.

Remarks: *a)* Traditional ray tracing which produces sharp shadows. *b)* Penumbra with classical stochastic ray tracing. *c)* Our method with shadow data sets. *d)* Optimized method with reduced extended objects.

		Cylinders	Balls4	Rings	Molecule	Bust
	# objects	11	7383	62	1685	98506
	# spherical light sources	2	3	3	2	1
	# reflected rays	20167	10976	2962	7303	0
a)	# shadow rays	31592	51719	47415	4260	32136
	# I_{geom}	0.85	1.74	1.40	0.99	0.94
b)	# shadow rays	1027757	1397529	1459345	105153	496704
	# I_{geom}	1.30	2.36	1.22	1.80	2.13
c)	# shadow rays	324911	430300	367355	34615	196667
	# I_{geom}	1.87	2.43	3.31	3.94	2.87
	# I_{shadow}	1.13	2.32	1.72	2.27	4.62
d)	# shadow rays	289054	215151	215155	26800	135324
	# I_{geom}	2.01	4.03	4.39	4.53	4.00
	# I_{shadow}	1.08	0.85	1.39	1.79	2.05

Table 2: Characteristics of the example scenes.

Remarks: In all scenes the number of primary rays was equal to 16384. The numbers I_{geom} and I_{shadow} denote the number of intersection tests per ray in the geometry data set and in the shadow data set, respectively. For the description of the different methods see remarks in Table 1.

[2] Arno Formella and Christian Gill. Ray Tracing: A Quantitative Analysis and a New Practical Algorithm. *The Visual Computer*, 11(9):465–476, December 1995.

[3] Andrew Glassner (editor). *An Introduction to Ray Tracing*. Academic Press, 1989.

[4] E. A. Haines and D. P. Greenberg. The Light Buffer: a Shadow Testing Accelerator. *IEEE Computer Graphics and Applications*, 6(9):6–16, 1986.

[5] Eric A. Haines. A Proposal for Standard Graphics Environments. *IEEE Computer Graphics and Applications*, 7(11):3–5, November 1987.

[6] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[7] A. Łukaszewski. Finding Ray–offset Intersection for Rational Bézier Surfaces. Technical Report 97/04, Institute of Computer Science, University of Wrocław, Poland, May 1997.

[8] A. Łukaszewski and A. Formella. Fast Penumbra Calculation in Ray Tracing. *Sixth International Conference in Central Europe on Computer Graphics and Visualization (Winter School on Computer Graphics)*, February 1998. Held in University of West Bohemia, Plzen, Czech Republic, 09-13 February 1998.

[9] J.R. Rossignac and A.A.G. Requicha. Offseting Operations in Solid Modelling. *Computer Aided Geometric Design*, 3:129–148, 1986.

[10] Eric Veach and Leonidas Guibas. Bidirectional Estimators for Light Transport. In *Fifth Eurographics Workshop on Rendering*, pages 147–162, Darmstadt, Germany, June 1994.