Approximating the Location of Integrand Discontinuities for Penumbral Illumination with Linear Light Sources

Marc J. Ouellette Eugene Fiume

Department of Computer Science
University of Toronto
Toronto, ON M5S 3G4 Canada
e-mail: {vv1|elf}@dgp.toronto.edu

Abstract

One of the benefits of shading with linear light sources is also one of its major challenges: generating soft shadows. The primary difficulty in this task is determining the discontinuities in the linear light source integrals that are caused by occluding objects. We demonstrate in this paper that the computed location of each discontinuity only needs to be moderately accurate, provided that the expected value of this location is a continuous function of the actual value of the location. We introduce *Random Seed Bisection* (RSB), an algorithm that has this property. We use this algorithm to efficiently find the approximate location of a discontinuity, in order to partition the domain of integration into subintervals (*panels*) over which the integrand is naturally smooth, and approximate the integral efficiently over each panel using low-degree numerical quadratures. We demonstrate the effectiveness of this solution for shadowing problems with at most 1 discontinuity in the domain of integration. We also provide efficient heuristics that take advantage of the coherence in a scene to handle shadowing problems with at most 2 discontinuities in the domain of integration. This work is a first step toward a comprehensive approach to efficiently solving numerical integration problems for extended light sources.

Résumé

L'un des bénéfices de l'utilisation de sources lumineuses linéaires représente aussi son plus grand défi, soit le calcul des ombres progressives. Pour résoudre ce problème, on doit déterminer les discontinuités dans la fonction à intégrer qui sont dues à une visibilité partielle de l'environnement. Nous démontrons qu'une approximation modérément précise de ces discontinuités est suffisante, à condition que la valeur calculée varie de façon continue en fonction de la valeur actuelle. Nous présentons un algorithme avec cette caractéristique, soit le *Random Seed Bisection* (RSB). Nous nous servons de cet algorithme pour trouver d'une façon efficace l'endroit approximatif d'une discontinuité, et pour ensuite subdiviser le domaine de l'intégrale en sous-domaines où l'on peut évaluer l'intégrale rapidement en se servant d'une méthode d'intégration à base d'interpolation polynomiale de faible degré. Nous démontrons l'efficacité de cette solution pour des problèmes d'ombres où l'intégrale possède une seule discontinuité. Nous proposons aussi des heuristiques pour les problèmes à 2 discontinuités qui sont basées sur la cohérence inhérente d'une scène. Ce travail constitue une première étape d'une nouvelle façon de résoudre les problèmes d'ombres progressives dues aux sources lumineuses étendues basée sur une approche numérique efficace.

Key words: Numerical quadratures, integration, linear light sources, soft shadows, random seed bisection.

1 Introduction

The process of determining the illumination provided by an extended light source can be separated into two phases. First, given a point to be shaded, we must determine the visibility of the source, that is, the domain of the source that is fully visible from the given point. Second, given the visible portion of the source, we must compute the reflected light due to this portion. In environments where sources emit light uniformly and surfaces reflect light diffusely, the latter integration problem can either be solved analytically — for example if the resulting area to integrate is polygonal [9] — or quickly approximated using low degree quadratures. In such cases, the limiting factor in obtaining accurate penumbral shadows lies in the ability to solve the visibility problem.

In the general setting of extended light sources, three main techniques have been used to handle the visibility of a source. The earliest techniques determined visibility of the source by either approximating it by point light sources [1], or by point sampling the source itself [4, 15]. This is prone to aliasing if an insufficient number of samples is used. Images of a higher quality can be achieved using algorithms that use shadow volumes and/or discontinuity meshing to determine the exact visibility of a source [11, 10, 2, 3, 7, 5, 14, 6]. Such techniques are very expensive and have only been designed to compute exact visibility for polygonal environments. Finally, shadow maps have also been used to approximate soft shadows, most recently by convolving source and occluder images to produce a soft shadow texture[12, 13]. Such algorithms can produce convincing shadows for environments with arbitrary types of objects, but incorrect shadows are sometimes produced, for example, if large occluding objects touch the receiver. Most of these extended light source shadow algorithms have been developed for area light sources, and not specifically for linear light sources, with some notable exceptions [11, 10].

Linear light sources, despite being useful illumination primitives, have often been neglected in computer graphics, or simply approximated with point light sources to avoid added rendering complexity. In this paper, we introduce a novel approach for solving the visibility problem in the context of linear light sources. Consider the problem of computing the reflectance at a given point on a surface due to light emanating from a linear light source. This is a one-dimensional integration problem over the angle defined by the point on the surface and the source. If we assume that the source emits light uniformly and that the surface reflects diffusely, the integration problem is trivial to solve, unless there is one or more occluding objects. These occluding objects create discontinuities in the integrand itself, since the integrand becomes zero over the portion of the source that is being occluded. Notice that these discontinuities are in the integrand, and are distinct from the discontinuities in the radiance function of the illuminated surface, as defined in discontinuity meshing algorithms.

In this paper, we explore the penumbral shadow problem by casting it in terms of an integration problem with a smooth integrand everywhere but at a small number of discontinuities. First, we introduce the *Random Seed Bisection* (RSB) algorithm, an algorithm for finding the approximate location of a discontinuity in an integrand. We demonstrate that this algorithm has an important property: the expected value of the discontinuity found by RSB is a continuous function of the actual value of the discontinuity. The RSB algorithm is also general in that it can find discontinuities for arbitrary objects, as long as an intersection routine can be written for that object.

We then use RSB to subdivide the domain of integration into panels bounded by a discontinuity and efficiently approximate the integral over each panel using a numerical quadrature of low degree. We demonstrate the effectiveness of this technique for integrands with at most 1 discontinuity. We then provide heuristics that take advantage of scene coherence to enable the technique to handle integrands with 2 discontinuities. Finally, we demonstrate some results for moderately complex scenes where the number of discontinuities in each integrand is small, but not necessarily less than three.

2 Discontinuities and Their Enumeration

The variation in the illumination function over a scene from fully lit regions in space, to penumbra, to umbra can be viewed as various kinds of discontinuities in that function. If such discontinuities can be accurately found, then domains of integration can be defined, over which the function is smooth. Both Drettakis[5] and Stewart[14] take this approach. However, the enumeration of all discontinuities is very expensive and perceptually unnecessary. Further, an efficient discontinuity preprocessing algorithm for non-polyhedral environments appears to be out of reach.

In this paper, we consider a different kind of discontinuities, namely, discontinuities in the linear light source integrand itself. This approach is motivated by the observation that in many scenes, most of the integrands resulting from the illumination of a point on a surface by a linear light source will have at most a small number of discontinuities. Even if, from a given point to be shaded, the total number of objects occluding the source is high, the effective number of discontinuities introduced in the source integrand can still be quite low. If several objects cause a single portion of a linear light source to be blocked, they will contribute at most 2 discontinuities to the resulting integrand. Consider the point to be shaded on the sphere in Figure 1. Even though the source is being occluded by a patch discretized into triangles, a cube, and a cone, only 2 discontinuities occur in the integrand due to these occlusions.

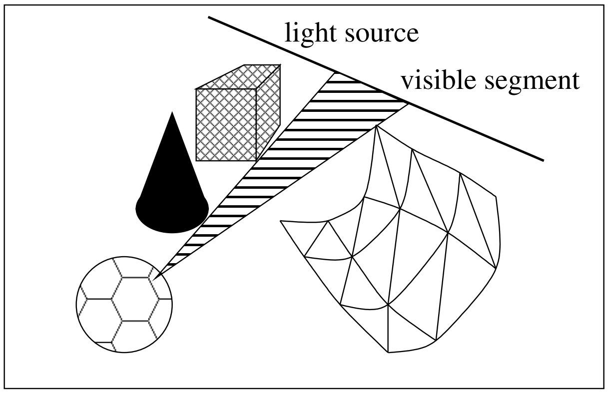
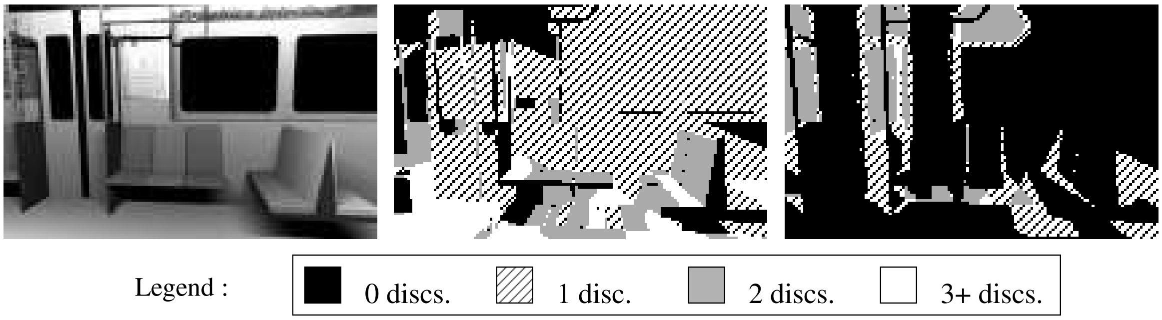


Figure 1: Multiple objects causing 2 discontinuities.

Consider the scene consisting of the interior of a subway car, as illustrated in Figure 2(a). The illumination in this scene comes from two linear light sources, run-



(a) Rendered Scene (b) Far Light Discontinuities (c) Near Light Discontinuities

Figure 2: Number of discontinuities arising from linear light sources. The number of discontinuities in the integrand at a pixel corresponds to its shade, as indicated in the above legend.

ning along each side of the car. The number of discontinuities in the two resulting integrands can be approximated for each pixel by calculating the visibility at a thousand points on each source and computing the number of visibility changes as the source is traversed from one end to the other. In Figure 2(b), we show, for each pixel, the number of discontinuities present in the resulting integrand due to the source along the far wall (i.e., the visible wall) of the car. Notice that more than 75% of the pixels have fewer than 2 discontinuities in the integrand, and only about 11% of the pixels have more than 2 discontinuities in the integrand. In Figure 2(c), we show the number of discontinuities in the resulting integrand due to the source along the near wall (this wall is behind the viewer and thus not seen). In this case, the results are even better, as 88% of the pixels have at most 1 discontinuity in their integrand, and less than 1% of the pixels have more than 2 discontinuities in their integrand.

In many scenes we consider to be "typical", we have observed that the total number of discontinuities arising from illumination with a linear light source is quite small. This suggests that specialized algorithms should be considered when rendering penumbral regions from linear light sources (and possibly other classes of light sources). In the following sections, we present a general algorithm for finding a discontinuity in a linear light source, and efficient algorithms to handle illumination integrals involving 0, 1, or 2 discontinuities. The algorithms only need to find the approximate location of these discontinuities in order to render visually pleasing penumbrae.

3 Iterative Discontinuity Finding Algorithms

In this section, we review three basic iterative algorithms for finding the approximate location of a discontinuity in an integrand, and define the new Random Seed Bisection (RSB) algorithm. Given an interval which is known to contain a single discontinuity λ, an iterative discontinuity finding algorithm finds $\tilde{\lambda}$, the approximate location of the discontinuity in this interval, within a tolerance ϵ. We will demonstrate in the next this section that even approximate knowledge of the location of a discontinuity can considerably improve the calculation of the penumbral shadow due to a linear light source.

Let $V(x)$ be the function that defines the visibility of a source from a point to be shaded. Assume that each linear light source (and its visibility function V) is parameterised on the unit interval $U = [0, 1]$. We further assume that each point $x \in U$ is either fully visible ($V(x) = 1$) or fully occluded ($V(x) = 0$). V is therefore a sequence of step functions. A discontinuity in the integrand occurs at a point λ iff $V(x)$ is discontinuous at λ.

Suppose that we know that a given V has at most 1 discontinuity. If $V(0) = V(1)$, then V is constant over U, and has 0 discontinuities. If $V(0) \neq V(1)$, then V has 1 discontinuity in U. Without loss of generality, we will assume for convenience that $V(0) = 1$. The problem of finding the discontinuity λ within a given tolerance ϵ can then be formulated as follows. Given a step function $V(x)$ defined as

$$V(x) = \begin{cases} 1 & \text{if } x \in [0, \lambda] \\ 0 & \text{if } x \in (\lambda, 1] \end{cases}, \quad (1)$$

find $\tilde{\lambda}$, the approximate location of the discontinuity, such that $||\tilde{\lambda} - \lambda|| \leq \epsilon$. We now examine four different iterative algorithms for solving this problem.

3.1 Pure Bisection

The *Pure Bisection* (PB) algorithm is defined by the methodical bisection (i.e., choosing the midpoint) of the interval containing the discontinuity. It is based on the observation that if an interval containing a single discontinuity is bisected, precisely one of the two resulting subintervals will contain the discontinuity.

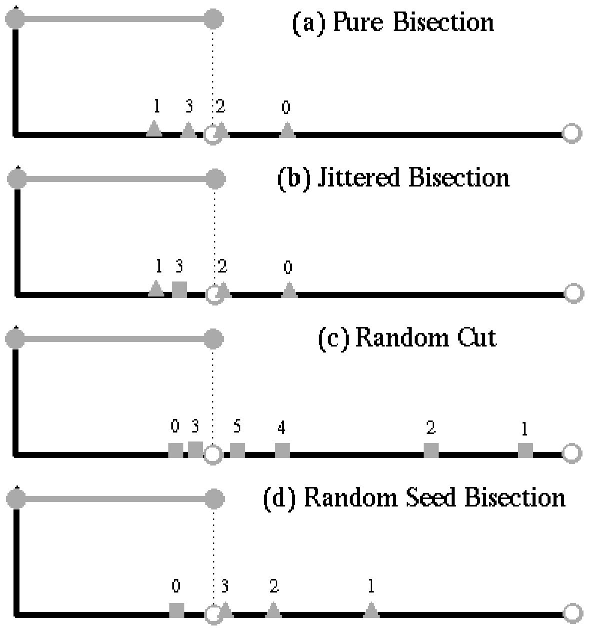


Figure 3: Illustration of iterative discontinuity finding algorithms with $V(x)$ a step function. Samples chosen as a midpoint are shown as triangles; samples chosen randomly are shown as squares.

PB provides the optimal convergence bound. The error is bounded by $(\frac{1}{2})^n$ after $n-1$ iterations. The expected value of $\tilde{\lambda}$ is a discontinuous function of λ:

$$E(\tilde{\lambda}) = \frac{2\lfloor 2^{n-1}\lambda \rfloor + 1}{2^n}. \qquad (2)$$

$E(\tilde{\lambda})$ has $2^{n-1}-1$ discontinuities in U.

3.2 Jittered Bisection

If PB is modified such that the point in the final interval is chosen randomly from a uniform distribution, it is called a *Jittered Bisection* (JB) algorithm. Since only the last point is chosen differently than in PB, and since the expected value of this last point is precisely the midpoint of the last subinterval (i.e., the last point chosen for PB), $E(\tilde{\lambda})$ is as in Equation (2), and again has $2^{n-1}-1$ discontinuities.

3.3 Random Cut

In a *Random Cut* (RC) algorithm, the sub-interval known to contain the discontinuity is subdivided by randomly choosing a point from a uniform distribution. RC has an expected value of $\tilde{\lambda}$ that is a continuous function of λ:

$$E(\tilde{\lambda}) = \frac{1/2 + \lambda(2^n - 1)}{2^{n+1}}. \qquad (3)$$

n	λ range	$E(\tilde{\lambda})$
1	$[0,1]$	$1/4 + \lambda/2$
2	$[0,1/2]$	$1/8 + 1/4\lambda + \lambda^2$
	$[1/2,1]$	$-3/8 + 9/4\lambda - \lambda^2$
3	$[0,1/4]$	$1/16 + 1/8\lambda + 49/18\lambda^2$
	$[1/4,1/2]$	$13/144 + 41/72\lambda + 1/2\lambda^2$
	$[1/2,3/4]$	$-23/144 + 113/72\lambda - 1/2\lambda^2$
	$[3/4,1]$	$-275/144 + 401/72\lambda + 49/18\lambda^2$

Table 1: Expected value of $\tilde{\lambda}$ for RSB.

The convergence bound for RC is worse than for PB. After $n-1$ steps, the expected value of the length[1] of the subinterval containing λ is $(\frac{2}{3})^n$.

3.4 Random Seed Bisection

The *Random Seed Bisection* (RSB) algorithm is a simple yet surprisingly effective modification of PB. Instead of splitting the original interval by testing the midpoint, we test a seed s that is chosen randomly from a uniform distribution over U. RSB then proceeds as in PB. It can be shown that the convergence bound for RSB is almost as good as the one for PB. Specifically, after $n-1$ steps, the expected value for the length of the subinterval containing λ is $(\frac{2}{3})(\frac{1}{2})^{n-1}$. This is easy to understand if we consider that, on average, after the initial random cut, we are applying PB on an interval of average length $2/3$.

The bound from RSB is not only very close to the optimal bound: the expected value of $\tilde{\lambda}$ is a continuous function of λ. $E(\tilde{\lambda})$ is tedious to calculate analytically. For example, for $n=2$, $E(\tilde{\lambda})$ is defined over $[0,1/2]$ by:

$$\int_{s=0}^{\lambda} \frac{3s+1}{4}\,ds + \int_{s=\lambda}^{2\lambda} \frac{3}{4}s\,ds + \int_{s=2\lambda}^{1} \frac{1}{4}s\,ds. \qquad (4)$$

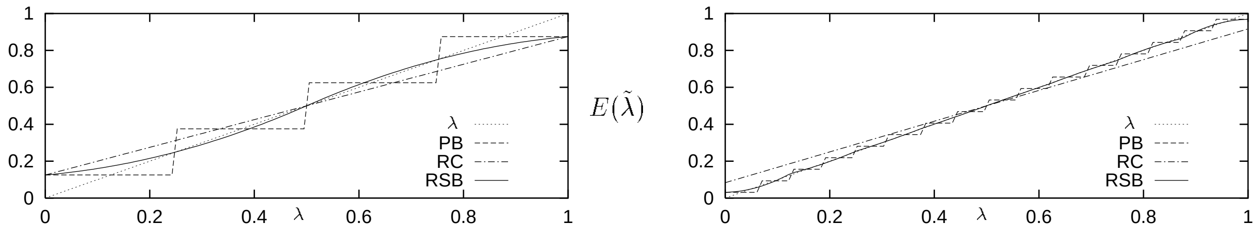
One can show that for any value n, the resulting function is a piecewise quadratic function of λ that is everywhere continuous. In Table 1, we have shown the quadratic functions for the first few values of n.

The four algorithms are illustrated in Figure 3 with $V(x)$ being a step function with a single discontinuity. The subdivision points chosen are enumerated at each step. Subdivision of the interval containing the discontinuity occurs either at the midpoint (shown as a triangle) or at a randomly selected point (shown as a square) in the interval.

3.5 Expected Value of Algorithms

In Figure 4, we have shown the expected value of $\tilde{\lambda}$ as a function of λ, for all the discontinuity finding algorithms.

[1] The expected length of the subinterval of $[0,1]$ containing λ is $\int_{\lambda=0}^{1}[\int_{x=0}^{\lambda}(1-x)\,dx + \int_{x=\lambda}^{1} x\,dx]\,d\lambda = \int[\frac{1}{2} + \lambda(1-\lambda)] = \frac{2}{3}$.



(a) n=2

(b) n=4

Figure 4: Expected value of algorithms. Abscissa is the location of the discontinuity, ordinate is the expected value.

The expected value plots are shown for $n = 2$ and $n = 4$ subdivisions. Since PB and JB have the same expected value, only the plot for PB is shown.

The expected value for RSB is continuous, and it also converges more quickly and in a more uniform fashion than PB. After only four subdivisions, the expected value of $\tilde{\lambda}$ for RSB has almost converged to $E(\tilde{\lambda}) = \lambda$. RC is continuous, as expected, but converges much more slowly than the other algorithms. Finally, notice that the discontinuities in $E(\tilde{\lambda})$ result in the staircase plots for PB.

4 Single Discontinuity Integration Algorithm

If we know that part of a scene, or perhaps an entire scene, is illuminated by linear light source integrals such that the resulting integrands have at most 1 discontinuity, we can use a discontinuity finding algorithm to help us evaluate the integral efficiently and accurately. Given a specific integration problem, we first test the visibility of each end point of the source, that is, $V(0)$ and $V(1)$. If both are 1, we have a fully visible source and evaluate the integral over it with no further visibility tests. If both are 0, we have a fully occluded source and the integral is zero. Finally, if $V(0) \neq V(1)$, we use an iterative discontinuity finding algorithm to approximate the location of the discontinuity, and integrate the source over the resulting visible portion with no further visibility tests. If the reflectance function of the surface is not highly specular, the visible portion of the integral can be efficiently approximated using a low degree quadrature such as a 2 point Gauss quadrature.

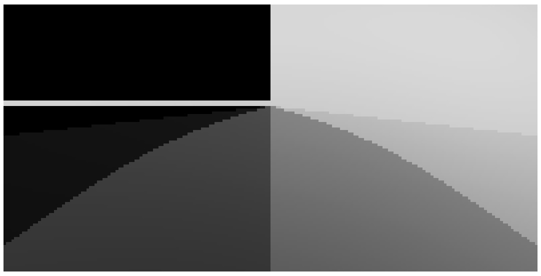
Figure 5 depicts a scene consisting of a linear light source illuminating a matte floor. The source is located at the back of the scene, and is partially obstructed on the left side by a wall. All visible points in this scene generate linear light source integrands (for direct illumination) that have at most 1 discontinuity. The source, which extends slightly beyond the visible image, is integrated by first finding the approximate location of the discontinuity and then using a 2 point Gauss quadrature on the resulting visible segment.

The first four images in the left column of Figure 5 were generated using the four different discontinuity finding algorithms to compute the location of the discontinuity within a tolerance of $\epsilon = 0.25$. The first four images in the right column were generated using the same algorithms with a tolerance of $\epsilon = 0.05$. The banding present in Figure 5(a) can be attributed to PB's discontinuous expected value. Notice that JB does not alleviate the banding in Figure 5(c), which is not surprising because JB's expected value is also discontinuous. Even for $\epsilon = 0.05$, the images in Figure 5(b) and 5(d) still show banding artifacts. In contrast, the images generated by both the random cut and the random seed bisections methods show no banding, a small amount of noise with $\epsilon = 0.25$, and almost no noise with $\epsilon = 0.05$.

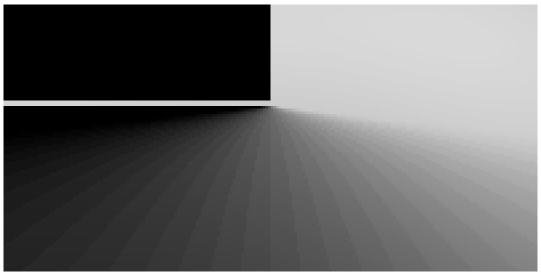
ϵ	PB	JB	RC	RSB
0.25	3.78	3.78	4.95	4.21
0.05	5.47	5.47	6.78	5.47

Table 2: Average number of visibility tests for tolerances of $\epsilon = 0.25$ and $\epsilon = 0.05$.

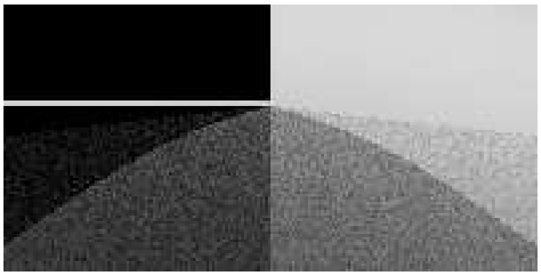
In terms of rendering times, the resulting cost can be broken down into two principal components: the cost of finding the location of the discontinuity, and the cost of calculating the integrand on the visible interval of integration. Because visibility has been approximately determined, the 2 point quadrature performs no visibility testing. Since a 2 point Gauss quadrature is used in every case, and since almost every pixel requires exactly one application of the quadrature, the difference in cost between the various algorithms can be directly attributed to the cost of finding the discontinuity. In Table 2, we have tabulated the average number of visibility tests for each method, for both $\epsilon = 0.25$ and $\epsilon = 0.05$. Notice that since PB and JB are identical in complexity, they share the same cost. The results are as predicted in theory, and the only surprise is that for a tolerance of 0.05, RSB performed as well as PB and JB.



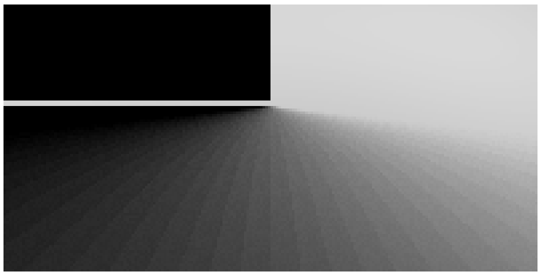
(a) Pure Bisection - $\epsilon = 0.25$



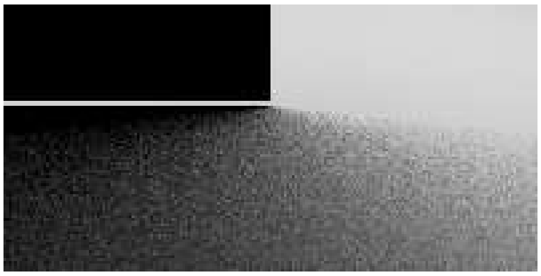
(b) Pure Bisection - $\epsilon = 0.05$



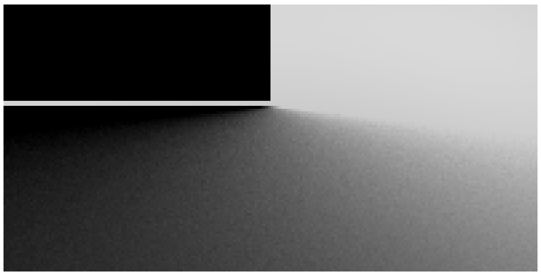
(c) Jittered Bisection - $\epsilon = 0.25$



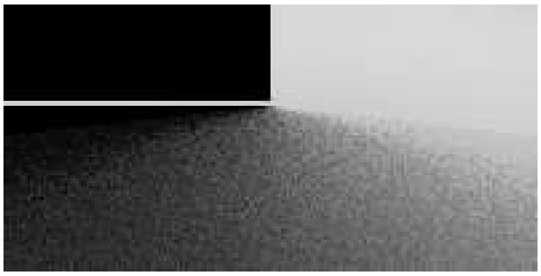
(d) Jittered Bisection - $\epsilon = 0.05$



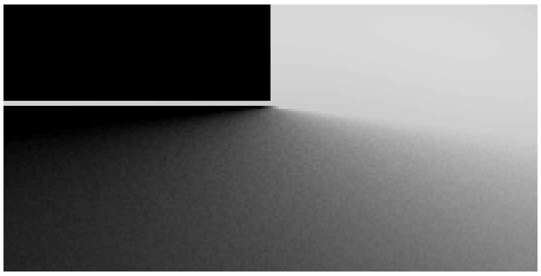
(e) Random Cut - $\epsilon = 0.25$



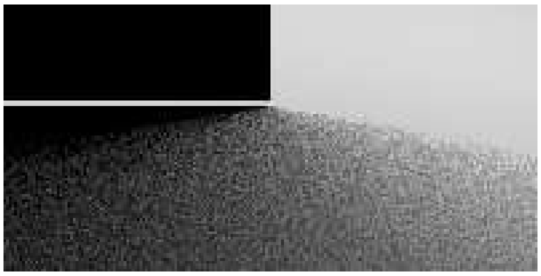
(f) Random Cut - $\epsilon = 0.05$



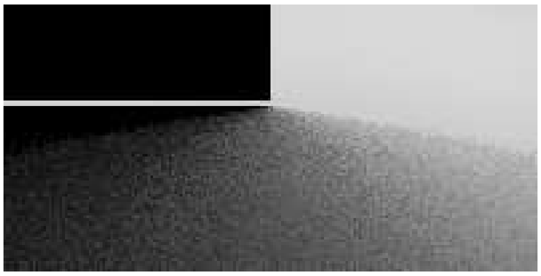
(g) Random Seed Bisection - $\epsilon = 0.25$



(h) Random Seed Bisection - $\epsilon = 0.05$



(i) HPD with 4 samples



(j) HPD with 6 samples

Figure 5: Penumbral region comparison for a scene with at most 1 discontinuity.

In Figure 5(i) and 5(j), we show the results of applying an integration method of cost similar to RSB. Since the integrand has an unknown discontinuity location, the most suitable[2] type of numerical technique to integrate in the presence of this discontinuity is one with a *blue noise* signature such as the *Hierarchical Poisson Disk* (HPD) [8] sampling strategy. We use such a strategy to calculate the integral over the entire source, testing for occlusion at each shading sample. Since we allowed roughly 4 occlusion tests for $\epsilon = 0.25$ and 6 occlusion tests for $\epsilon = 0.05$, we have shown the results for both 4 and 6 shading samples using HPD. Notice that this comparison is actually biased in favour of HPD since it does not take into consideration the cost of calculating the illumination from the additional shading samples (e.g., RSB with $\epsilon = 0.05$ uses on average 5.47 visibility tests and 2 shading samples, whereas HPD uses 6 visibility tests and 6 shading samples). Comparing Figure 5(g) to Figure 5(i) and Figure 5(h) to Figure 5(j), we conclude that RSB outperforms HPD.

The results of this comparison illustrate the importance of the method used to find the approximate location of a discontinuity. Using the results of the discontinuity finding methods, we can efficiently find good visual approximate solutions to the penumbral integration problem.

5 Two Discontinuity Integration Algorithm

In this section, we present a *Two Discontinuity Finding* (TDF) algorithm, which is an extension of the algorithm of the previous section. TDF enables us to efficiently handle integrands with up to 2 discontinuities. Let us consider the end points of a linear light source whose visibility function $V(x)$ is known to have at most 2 discontinuities over U. If $V(0) \neq V(1)$, then we only have 1 discontinuity and we can use RSB to find it. Otherwise, $V(0) = V(1)$ and the end points are either both visible, or both occluded. We must decide if we have 0 or 2 discontinuities.

5.1 Scene Coherence and Visibility Changes

The heuristics we have designed are motivated by the observation that the integrand will have 2 discontinuities iff there exists a point $m \in (0, 1)$ such that $V(m) \neq V(0)$. Let P be the number of discontinuities detected in the integrand for the previous (i.e., adjacent) pixel, and let these discontinuities be p_1 and p_2, if they exist. We have designed a heuristic for each value of P that either finds a point m such that $V(m) \neq V(0)$, or returns a failed status. If m is found, we use RSB to find a discontinuity in $[0, m]$ and in $[m, 1]$, otherwise we conclude that there is no discontinuity. We now present the heuristics of TDF.

[2]Any strictly deterministic integration method with 6 or less points will suffer from banding.

State $P = 0$

Let ν be a user-specified tolerance, and assume that the visibility function for the source is defined over U, and let $x_{-2} = 0$ and $x_{-1} = 1$. Choose a random value $x_0 \in U$ and determine $V(x_0)$. If $V(x_0) \neq V(0)$, then let $m = x_0$ and return. Otherwise, at each subsequent step i, we choose x_i as the midpoint of the largest subinterval $[x_a, x_b]$ such that $a, b \in [-2, i-1]$ and there is no $c \in [-2, i-1]$ such that $x_c \in (x_a, x_b)$, with ties being broken randomly. In other words, this algorithm chooses the next point by examining the Voronoi diagram corresponding to the points already chosen and selecting the midpoint of the largest Voronoi region. For this reason, we will refer to this algorithm as *Voronoi Search* (VS).

The iteration stops as soon as some $V(x_i) \neq V(0)$, in which case we let $m = x_i$ and return. If we reach the tolerance ν, that is, if the largest subinterval becomes smaller than ν, we stop and conclude that, probabilistically[3], we have 0 discontinuities, and return a failed status.

State $P = 1$

Since the integrand for the previous pixel had 1 discontinuity, then one end point of the source was visible, and one was occluded. Since the endpoints are now both occluded or both visible, one of the end points has changed visibility, say v_i. Based on scene coherence, it is likely that if a change of visibility still occurs in the integrand (i.e., if we now have 2 discontinuities), it does so near the previous discontinuity p_1. Since p_1 was only known within a tolerance of ϵ, we let $m = \frac{p_1+v_i}{2}$ and compute $V(m)$. If $V(m) \neq V(0)$, we return successfully. Otherwise, we use VS to look for a change of visibility between p_1 and v_i.

State $P = 2$

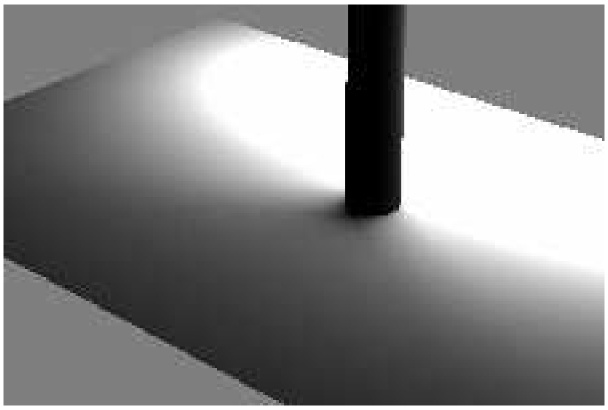
The integrand for the previous pixel had 2 discontinuities. Based on scene coherence, it is likely that if we still have 2 discontinuities in the integrand, the midpoint $m = \frac{p_1+p_2}{2}$ between p_1 and p_2 will be such that $V(m) \neq V(0)$. If this is the case, return successfully. Otherwise, we use VS to look for a change of visibility in U.

5.2 Results

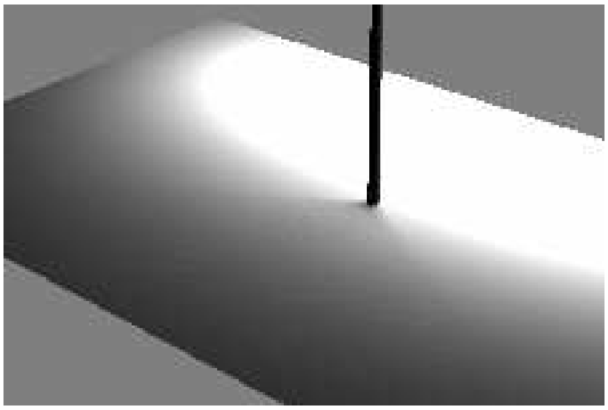
Figure 6 depicts a scene consisting of a linear light source illuminating a matte floor. The source is located above and behind the top right edge of the floor, and is partially obstructed by a cylinder. All visible points in this scene generate linear light source integrands (for direct illumination) that have at most 2 discontinuities.

Cylinders of two different radii were chosen to test different degrees of difficulty in finding discontinuities. In both cases, the light is of length 50 and is 21 units away

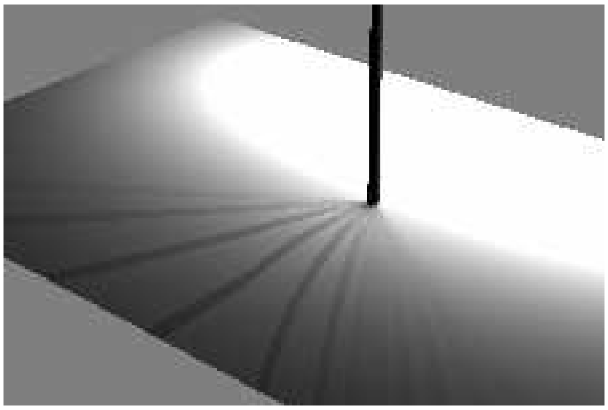
[3]Specifically, there is no gap $G \subseteq U$ of size $||G|| > \nu$ such that $V(X) \neq V(0), \forall x \in G$.



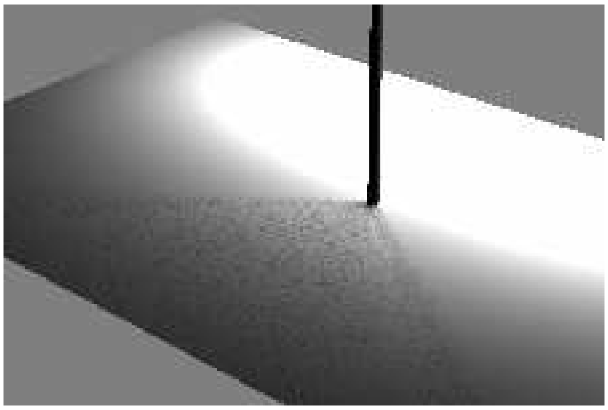
(a) TDF ($\epsilon = 0.05, \nu = 0.25$); radius 2



(b) TDF ($\epsilon = 0.05, \nu = 0.25$); radius 0.4



(c) 7 point Gauss; radius 0.4



(d) 7 point HPD; radius 0.4

Figure 6: Penumbral region comparison for a scene with at most 2 discontinuities.

(at a 45° angle) from the base of the cylinder. The cylinder in Figure 6(a) has a radius of 2, whereas the cylinder in Figures 6(b), 6(c), and 6(d) has a radius of 0.4.

The first two images were generated using TDF and a tolerance of $\epsilon = 5\%$ in RSB, and of $\nu = 25\%$ in VS. TDF required an average of 5.9 visibility tests in Figure 6(a), and 5.6 visibility tests in Figure 6(b), to determine the location of the discontinuities. The resulting segments were then integrated using a 2 point Gauss quadrature. The shadows are very smooth in both cases. The pixel adjacency heuristics used in states $P = 1$ and $P = 2$ were very successful at predicting the location of a discontinuity without having to resort to VS: 100% in Figure 6(a) and 98% in Figure 6(b).

Figure 6(c) was generated using a 7 point Gauss quadrature. Notice the aliasing appearing as shadow bands in the image. Figure 6(d) was generated using an approximation to HPD using 7 samples per integral, and a cylinder of radius 0.4. HPD was the best representative among the methods which do not take into consideration the location of discontinuities. The resulting image does not suffer from aliasing but is very noisy.

The number of samples (7 for both Gauss and HPD) was chosen to roughly match the average number of visibility tests (5.6) required to compute Figure 6(c) plus the average number of shading calculations (2.5), such that the algorithms had the same rendering time (less than 3 seconds). From Figure 6(b), 6(c) and 6(d), we can conclude that the integrand discontinuity finding method outperforms both the Gauss quadrature and the HPD method.

In Figure 2 we showed an example of a moderately complex scene modelled with polygons, cylinders and spheres. It is illuminated by two linear light sources, with only 5% of the resulting integrands having more than 2 discontinuities. In Figure 7 we show the result of applying TDF to this scene. On average, 5 visibility tests were needed to locate the discontinuities on each light source, for a total rendering time of 3.4 minutes. The resulting shadows are very convincing in all the regions where fewer than 3 discontinuities were present. Notice the smoothness of the shadow in the difficult area near the bottom of the wall, immediately to the right of the set of

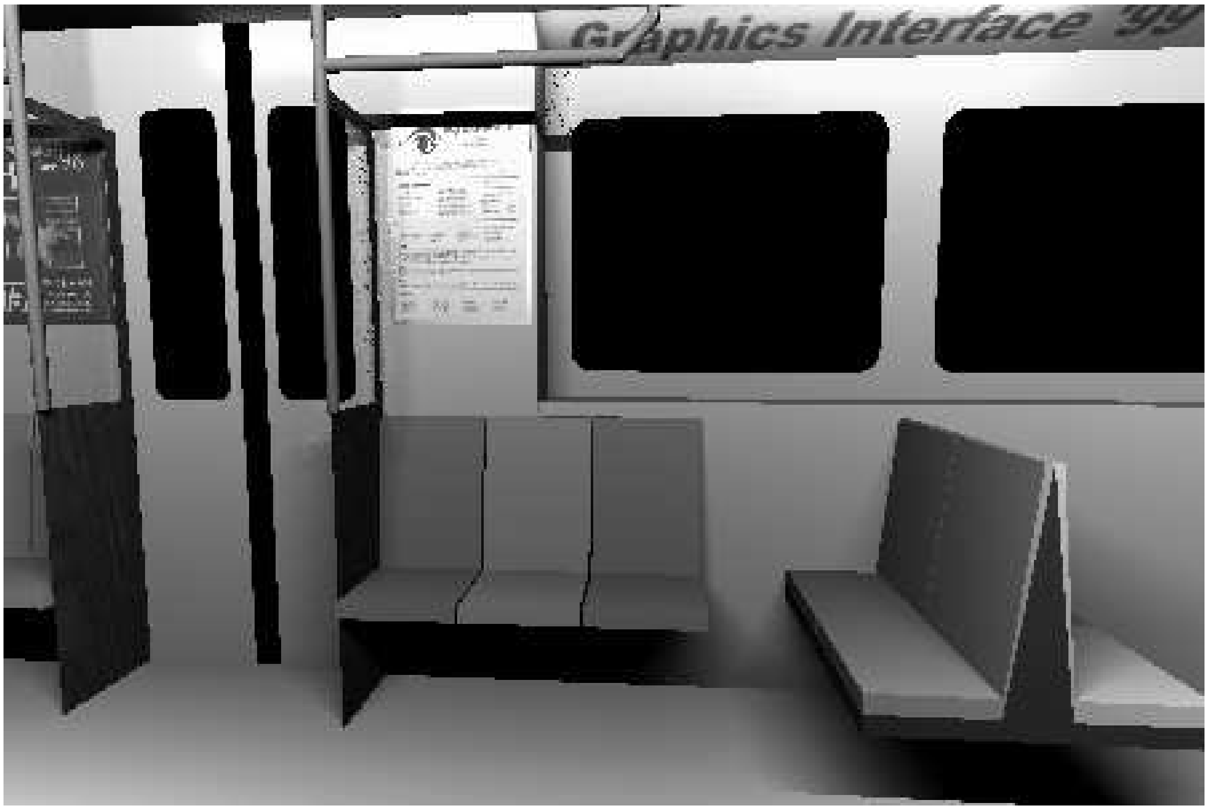


Figure 7: Subway scene computed with TDF ($\epsilon = 0.05$, $\nu = 0.25$).

three seats. In this area, a small portion of each source is visible, even though the end points themselves are occluded. In some parts of the image, the limitation of the algorithm is evident in the form of speckling.

6 Discussion

One of the important characteristics of the Two Discontinuity Finding (TDF) algorithm is the use of tolerances to control the quality of the discontinuity approximations. On one hand, this gives users a powerful means of balancing the quality of the shadows with its cost. On the other hand, given an integrand with at most two discontinuities, the RSB tolerance ϵ and the VS tolerance ν are reliable predictors of the quality of the shadow. Given a pair of discontinuities due to a small occluder, the probability of TDF missing the occluder is inversely proportional to the fraction of the source that is being occluded. If a small occluder is found, RSB guarantees that its extent will be determined correctly, within the tolerance of ϵ. Therefore the difference between a small occluder being missed or being found is bounded above by the actual contribution of the occluder, within RSB's tolerance ϵ. This is an important characteristic for low sampling densities, since "accidentally finding a small occluder" does not lead to an overly dark shadow. This is in contrast to more traditional low density visibility testing where one sample "accidentally finding a small occluder" leads to overestimating the importance of the shadow, as was evident in Figure 6(c).

To isolate the effectiveness of the algorithms, anti-aliasing was not performed on any of the images included in this paper. An interesting direction of research will be to combine anti-aliasing techniques with our algorithms. One possible application would be to compute the integrand discontinuity only once per pixel, thus amortizing the cost over all the anti-aliasing samples used for each pixel. This would allow the algorithms to either use smaller tolerances, or to speedup their performance.

A question that remains to be answered is how to use the algorithms we have developed to handle integrands with more than 2 discontinuities. As we saw in Figure 7, the results are promising but problems do exist. For example, the speckling near the middle-top is a result of a source integrand with 3 discontinuities — one caused by the recessed wall, the other two by the overhanging handrail — incorrectly being handled. The algorithm correctly detects that one end point is occluded and one visible, and uses RSB to find the discontinuity. The resulting pixel is noticeably too dark if the discontinuity found is on the handrail instead of on the edge of the wall, since then too much of the source is classified as occluded. To alleviate this type of problem, one could track variations in irradiance in a small neighbourhood about each pixel and render more carefully (e.g., by subdividing the source) those pixels whose variance exceeds a certain threshold.

7 Conclusion

In this paper, we have presented a new approach to solving the penumbral problem for linear light sources. This approach is based on the observation that most integrands are smooth in their domain of integration, except at a small number of discontinuities. We have demonstrated that in many cases, knowing the approximate location of a discontinuity, in conjunction with using a low degree quadrature method is sufficient to provide quality penumbral shadows.

We introduced Random Seed Bisection (RSB), a new algorithm for computing the approximate location of the discontinuity in a linear light source. This algorithm converges nearly as quickly as the traditional bisection method, but provides continuous expected values for the approximate location of the discontinuity. This results in smooth penumbral shadows when integrands with 1 discontinuity are present. RSB is also a general algorithm, since it can be used with any type of object, provided that an intersection routine can be written for that object.

Finally, we combined RSB with effective heuristics based on scene coherence to develop a Two Discontinuity Finding (TDF) algorithm. TDF efficiently handles integrands with at most 2 discontinuities. We presented convincing results for test cases of varying difficulty, and discussed the important characteristics of TDF. We also presented some promising results for a complex scene containing some integrands with more than 2 discontinuities.

In the future, we hope to further develop methods to efficiently handle multiple discontinuities in linear light source integrands. Two particularly interesting areas of research are handling more than 2 discontinuities, and combining these algorithms with anti-aliasing. Finally, we will continue the investigation of the usage of approximate knowledge of integrand discontinuities for computing penumbral shadows due to area light sources, and to investigate the importance of similar algorithms for other aspects of the rendering process.

8 Acknowledgements

The comments by the reviewers helped to improve the paper, and were much appreciated. We gratefully acknowledge the funding of NSERC and CITO in our research.

References

[1] Brotman, L.S., Badler, N.I., “Generating Soft Shadows with a Depth Buffer Algorithm”, *IEEE Computer Graphics and Applications*, 4(10), Oct. 1984.

[2] Campbell, A.T., “Modeling Global Diffuse Illumination for Image Synthesis”, Ph.D. Thesis, University of Texas at Austin, Dec. 1991.

[3] Chin, N., Feiner, S., “Fast Object-Precision Shadow Generation for Area Light Sources Using BSP Trees”, *ACM Computer Graphics (SIGGRAPH Symp. on Inter. 3D Graphics 1992).*

[4] Cook, R.L., Porter, T. and Carpenter, L., “Distributed Ray Tracing”, *Computer Graphics*, 18(3), July 1984.

[5] Drettakis, G., Fiume, E., “A Fast Shadow Algorithm for Area Light Sources Using Backprojection”, *ACM SIGGRAPH Annual Conference Series*, July 1994.

[6] Hedley, D., Worrall, A., Paddon, D., “Selective Culling of Discontinuity Lines”, *8th Eurographics Workshop on Rendering*, June 1997.

[7] Lischinski, D., Tampieri, F., Greenberg, D., “Discontinuity Meshing for Accurate Radiosity”, *IEEE C.G. & Appl.*, 12(6), Nov. 1992.

[8] McCool, M., Fiume, E., “Hierarchical Poisson Disk Sampling Distributions”, *Graphics Interface 92*, 1992.

[9] Moon, P., *The Scientific Basis of Illuminating Engineering*, McGraw-Hill, New York, 1936.

[10] Poulin, P., Amanatides, J., “Shading and Shadowing with Linear Light Sources”, *Eurographics*, 1990.

[11] Nishita, T., Okamura, I., Nakamae, E., “Shading Models for Point and Linear Light Sources”, *ACM Transactions on Graphics*, 4(2), Apr. 1985.

[12] Soler, C., Sillion, F.X., “Automatic Calculation of Soft Shadow Textures for Fast, High Quality Radiosity”, *9th Eurographics Workshop on Rendering*, June 1998.

[13] Soler, C., Sillion, F.X., “Fast Calculation of Soft Shadow Textures Using Convolution”, *ACM SIGGRAPH Annual Conference Series*, July 1998.

[14] Stewart, J.A., Ghali, S., “Fast Computation of Shadow Boundaries Using Spatial Coherence and Backprojections”, *ACM SIGGRAPH Annual Conference Series*, July 1994.

[15] Wallace, J.R., Elmquist, K.A., and Haines, E.A., “A Ray Tracing Algorithm for Progressive Radiosity”, *Computer Graphics*, 23(3), July 1989.