

Frequency Analysis and Sheared Filtering for Shadow Light Fields of Complex Occluders

KEVIN EGAN

Columbia University

FLORIAN HECHT

University of California, Berkeley

FRÉDO DURAND

MIT CSAIL

and

RAVI RAMAMOORTHI

University of California, Berkeley

Monte Carlo ray tracing of soft shadows produced by area lighting and intricate geometries, such as the shadows through plant leaves or arrays of blockers, is a critical challenge. The final image often has relatively smooth shadow patterns, since it integrates over the light source. However, Monte Carlo rendering exhibits considerable noise even at high sample counts because of the large variance of the integrand due to the intricate shadow function. This article develops an efficient diffuse soft shadow technique for mid to far occluders that relies on a new 4D cache and sheared reconstruction filter. For this, we first derive a frequency analysis of shadows for planar area lights and complex occluders. Our analysis subsumes convolution soft shadows for parallel planes as a special case. It allows us to derive 4D sheared filters that enable lower sampling rates for soft shadows. While previous sheared-reconstruction techniques were able primarily to index samples according to screen position, we need to perform reconstruction at surface receiver points that integrate over vastly different shapes in the

This work was supported in part by NSF grants 0701775 and 0924968 and the Intel STC for Visual Computing. The authors also acknowledge ONR for their support (PECASE grant N00014-09-1-0741), as well as generous support and Renderman licenses from Pixar, an NVIDIA professor partnership award and graphics cards, and equipment and funding from Intel and Adobe. Authors' addresses: K. Egan, Computer Science Department, Columbia University, 116th Street and Broadway, New York, NY 10027; email: ktegan@cs.columbia.edu; F. Hecht, Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA; F. Durand, MIT CSAIL, MIT; R. Ramamoorthi, Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2011 ACM 0730-0301/2011/04-ART9 \$10.00

DOI 10.1145/1944846.1944849

<http://doi.acm.org/10.1145/1944846.1944849>

reconstruction domain. This is why we develop a new light-field-like 4D data structure to store shadowing values and depth information. Any ray tracing system that shoots shadow rays can easily incorporate our method to greatly reduce sampling rates for diffuse soft shadows.

Categories and Subject Descriptors: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—*Color, shading, shadowing, and texture*

General Terms: Algorithms

Additional Key Words and Phrases: Soft shadows, area lights, sampling, frequency analysis, light fields, sheared reconstruction

ACM Reference Format:

Egan, K., Hecht, F., Durand, F., and Ramamoorthi, R. 2011. Frequency analysis and sheared filtering for shadow light fields of complex occluders. *ACM Trans. Graph.* 30, 2, Article 9 (April 2011), 13 pages.

DOI = 10.1145/1944846.1944849

<http://doi.acm.org/10.1145/1944846.1944849>

1. INTRODUCTION

Many algorithms have been used to generate soft shadows cast by area lights, but Monte Carlo sampling is the method of choice for production rendering due to its simplicity and widespread use for offline rendering. Unfortunately, when computing shadows from intricate geometry (see Figure 1), the (binary) visibility function on the light source is complex and high frequency. While the *integral* of this function can still be relatively smooth, the Monte Carlo point samples (shadow rays) have high variance and considerable noise persists even for large sample counts (Figure 1), requiring the use of a prohibitive number of shadow rays. This is frustrating because the resulting shadows can be smooth and simple, despite the complex and costly calculation that went into them.

We propose to efficiently sample and filter the 4D shadow light field from a complex occluder, thanks to a new analysis of shadow sampling and reconstruction. We introduce a new 4D shadow light field cache that allows for integration and reuse across pixels. The sampling of our method is driven by a frequency analysis at the visible receivers, and a new sheared filter allows neighboring receiver points to share data and reduce sample count. Our specific contributions include the following.

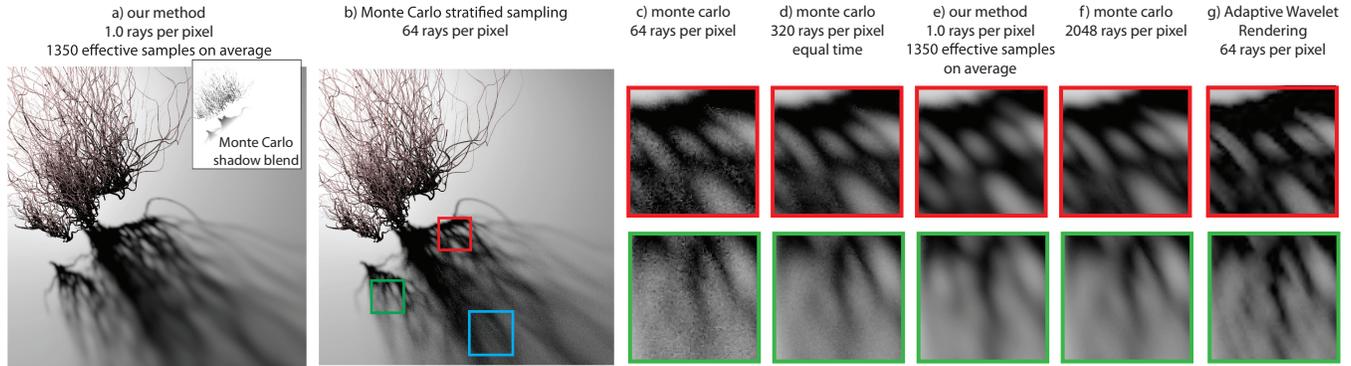


Fig. 1. (a) Our method casting 1 shadow ray per pixel. Our wide filter gives an average effective sampling rate of 1350 samples for every pixel that is partially occluded. We use brute-force Monte Carlo ray tracing for self-shadowing and near-field occlusion, and then blend into our results for mid- and far-field occlusion, as shown in the inset. (b) Monte Carlo stratified sampling with 64 samples has large amounts of noise due to the complex geometry, also shown in the insets in (c). (d) Even with 320 samples the shadow still has visible noise. (e) Our method using 1 ray per pixel. By sharing samples between neighboring receiver points, we obtain an effective sampling rate of 1350 samples per pixel. (f) Ground truth, generated using 2048 shadow rays per pixel. (g) Comparison to Adaptive Wavelet Rendering with 64 samples per pixel. Visible artifacts can be seen due to the high variance of the shadow samples. (blue box) Our method exhibits some overblurring in the area highlighted with the blue box. See Section 7 and Figure 15 for more details.

Frequency Analysis of Shadow Signal. We first show that only a narrow wedge of the Fourier spectrum usually has significant amplitude if the depth range of the blockers is limited. Complex occluders with a bounded depth range are common in cases like dense foliage or irregular arrays of blockers. Our analysis subsumes and extends convolution soft shadows in parallel planes [Soler and Sillion 1998].

Sheared Filters for Shadows. We introduce a new reconstruction filter that is sheared in the receiver-light domain, and enables very sparse sampling since visibility samples can be shared among adjacent pixels. We generalize previous work on sheared filters in other contexts [Chai et al. 2000; Egan et al. 2009] to irregular reconstruction problems; the depths of the receiver points may vary, which in turn causes the bundle of rays that we integrate over to have different shapes. We first design the sheared filter in the native coordinate system of the receiver point, and then transform to a parameterization that is agnostic to the receiver point.

Practical Algorithm. An overview of our method can be seen in Figure 2. We first sparsely sample the occlusion light field by shooting a small number of shadow rays. We then store all ray samples in a ray database. Finally, at each receiver pixel, we use our frequency analysis to calculate the best filter shape for the receiver, and filter over the samples in our ray database. Our analysis shows that we can often use a wide filter across the shadow light field, effectively reusing rays cast from nearby receiver points.

2. PREVIOUS WORK

As a full review of shadow algorithms is beyond the scope of this article, we focus on approaches that produce accurate soft shadows. Readers are encouraged to read a survey of approximate real-time soft shadow techniques [Hasenfratz et al. 2003], as well as a comparison of more recent methods [Johnson et al. 2009].

Frequency Analysis and Reconstruction Methods. Often high-dimensional signals have long narrow spectra in the Fourier domain. In these cases adaptive sampling of the spectra can be used [Soler et al. 2009], as well as sheared filters that compactly

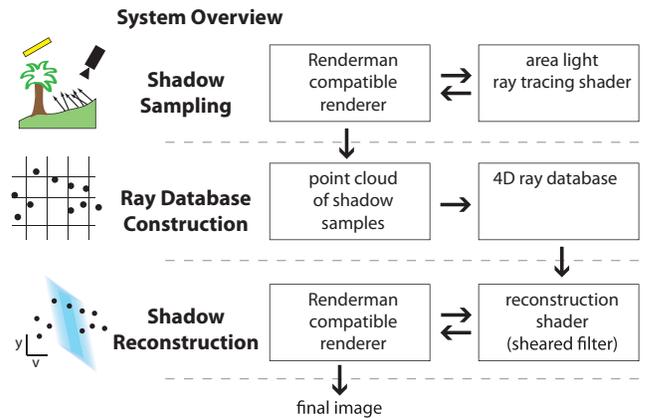


Fig. 2. A flow chart showing the architecture and data flow in our system. (Shadow Sampling) Sparsely sample the light field using shadow rays and write out each ray result to disk. (Database Creation) Read in the ray samples and create a ray database. (Shadow Reconstruction) Query the ray database and use sheared filters to reconstruct the shadow.

capture the spectra in the Fourier domain and allow the use of sparser sampling rates [Shinya 1993; Chai et al. 2000; Zwicker et al. 2007; Egan et al. 2009]. The shape of shadow spectra has been studied in the Fourier domain [Durand et al. 2005; Ramamoorthi et al. 2005; Lanman et al. 2008]. We extend these analyses by showing that the frequency spectrum for practical scenes is most often a wedge based on the minimum and maximum depth of the occluder. We also draw attention to extreme cases where this assumption does not hold (Section 7). The use of first-order gradients to aid in reconstruction has been studied [Ramamoorthi et al. 2007], and several new techniques for reconstructing general signals have also been developed [Hachisuka et al. 2008; Overbeck et al. 2009]. We use a sheared filter and extend previous work to solve the more general problem where the pixel integrands are not aligned to a regular grid. In Section 6 we compare to Adaptive Wavelet Rendering [Overbeck et al. 2009], the state-of-the-art in contrast-based adaptive

reconstruction, and show that for low sample counts our sheared filter produces more accurate results.

Sheared Filters. Our work is perhaps closest to the sheared filters developed for other problems like light fields [Chai et al. 2000] and motion blur [Egan et al. 2009]. Our theoretical analysis relates to these approaches, but we focus on shadows and show how our analysis reduces to convolution soft shadows in the special case of parallel planes [Soler and Sillion 1998]. Moreover, previous methods assume a regular grid of cameras or that all pixels integrate over the same shutter interval. In contrast, we are integrating over a fixed plane (the light), but our sampling and filtering happen at points that are at many different depths. Thus, we must solve a more general irregular reconstruction problem. We therefore introduce an additional step, going from the actual receiver to a shadow light field that is independent of receiver depth.

Ray Traced Shadows. Brute-force ray tracing computes correct answers but is expensive [Cook et al. 1984]. Photon mapping shoots shadow photons as an optimization to classify areas that are unoccluded, occluded, or partially occluded from direct lighting [Jensen and Christensen 1995]. Our method focuses on areas with partial occlusion, whereas most photon mapping implementations fall back to Monte Carlo sampling in these areas rather than directly visualizing the shadow photon map. Multidimensional lightcuts uses a hierarchical tree graph for receiver points and point light sources, makes cuts through the receiver and light graphs at each pixel, and shoots shadow rays for all pairs of nodes along the graph cuts [Walter et al. 2006]. Our work is complementary to both photon mapping and multidimensional light cuts, since our sheared filter can be incorporated to select a large set of appropriate shadow rays to share for a given receiver point, further reducing shadow ray casts. Coherence across occluders and receivers has been used [Bala et al. 1999; Hart et al. 1999; Agrawala et al. 2000; Ben-Artzi et al. 2006], as well as separating near- and far-field occlusion [Arikan et al. 2005]. Blurring sharp ray traced results in image space can also be used to approximate soft shadows and blurry reflections [Robison and Shirley 2009]. Other methods have prefiltered partial occlusion at kd-tree cells, but darkening can occur when locally prefiltered nodes are composited together [Lacewell et al. 2008]. Our system enables sparser sampling than previous methods because we share samples and exploit coherence in the full 4D shadow light field.

Light Fields and Precomputed Radiance Transfer. Many previous methods have used light fields for rendering [Gortler et al. 1996; Levoy and Hanrahan 1996; Isaksen et al. 2000; Chen et al. 2002; van der Linden 2003; Stewart et al. 2003]. The shape of occlusion light fields has been studied [Durand 1999], as well as how to capture occlusion light fields [Lanman et al. 2008]. These methods usually use image-based rendering where data is captured by photographs taken in a regular grid, whereas we sparsely sample only the areas of the light field that are used by the receivers of the image. Precomputed radiance transfer methods can also be used for relighting problems involving complex shadows [Ng et al. 2003; Zhou et al. 2005; Sun and Ramamoorthi 2009], but most methods require dense sampling of an object or scene, and its light transport.

Shadow Maps. There are a variety of area light source methods that use shadow maps [Yang et al. 2009], or a statistical description of occlusion [Annen et al. 2008]. The main drawback to using

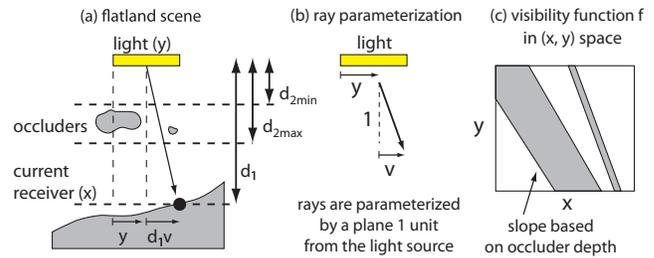


Fig. 3. A simple illustration in flatland. (a) Note that we handle many occluders in a range of depths [d_{2min} , d_{2max}], and that d_1 is the distance to the current receiver, but we do not assume all receivers are coplanar. The vertical line at the left side of the light serves as the origin of spatial coordinates for all planes. (b) We parameterize rays based on the ray origin and directional offset at a plane 1 unit away. (c) Occlusion in (x, y) space has coherent diagonal bands where occluders block the light source.

shadow maps is that most area light source techniques either have a fixed resolution for the shadow map that can miss geometric detail, or they process occluders independently and use approximate methods to composite the result [Johnson et al. 2009]. One exception to this rule is the Sample Based Visibility method that uses alias-free shadow maps and conservative triangle rasterization [Sintorn et al. 2008]. The generation of soft shadow textures by Soler and Sillion [1998] shows that for parallel plane occluder-receiver pairs the resulting shadow is a convolution between the light source and the planar occluder, leading to a multiplication of light and occluder spectra in the frequency domain. We show that our analysis of nonplanar occluders and receivers generalizes their approach (Section 3.2). Furthermore, our implementation samples across a 4D ray database and can handle receiver surfaces that smoothly vary from close to far away from the light source. In comparison, their method captures 2D information from a single point on the light source and can have discontinuities in areas that transition from one soft shadow texture to another.

Object-Based Methods for Shadows. Storing silhouette edges allows for efficient sampling of the light source [Laine et al. 2005]. Penumbra wedges [Assarsson and Akenine-Möller 2003] and beam tracing can also be used [Overbeck et al. 2007]. These methods process triangle edges, which becomes a bottleneck for highly tessellated scenes or scenes with spiky geometry.

3. SHADOW SIGNAL AND LIGHT FIELD

We start our analysis of the shadow signal and light field with a simple scene in flatland, where the distance from the planar light to the current receiver point is d_1 , and the occluding geometry is contained within a depth range of [d_{2min} , d_{2max}] measured from the light (see Figure 3(a)). In our implementation, this analysis is applied to the local extent of a single pixel, allowing our algorithm to use a different value of d_1 per pixel and handle nonplanar receiver surfaces. Parts of our analysis will examine frequencies of the receiver, and for these problems it is most natural to use a two-plane (x, y) parameterization, where y is the absolute distance along the planar light source, and x is the absolute distance along the plane parallel to the light source with distance d_1 (the receiver plane). However, because we want to share rays across many receivers, we store ray samples in a receiver-independent (v, y) parameterization, where y is still a distance along the light source, and v is measured as an offset from y at a plane one unit from the light (similar to Durand et al. [2005]; see Figure 3(b)).

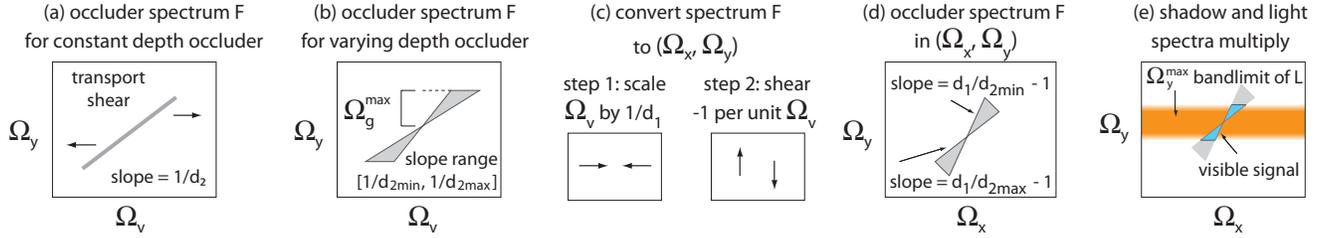


Fig. 4. We design our filter in the Fourier domain, later reinterpreting these steps in the primal domain to obtain the filter used in our implementation. Based on Equation 3, the frequency content for the occluder in (v, y) space will be (a) a line for occluders with a constant depth, and (b) a wedge for occluders with a range of depths. (c) We scale and shear this picture to (d) the frequency space of the receiver, (Ω_x, Ω_y) . Based on Equation 5, our filter must cover the overlap between the occluder spectrum and the light spectrum (e).

Our analysis in flatland is easy to extend to 3D where the light field has four dimensions (v_1, v_2, y_1, y_2) . If we use orthogonal basis vectors to parameterize the area light source, the (v_1, y_1) subspace is linearly independent of the (v_2, y_2) subspace. Because of this, most of the computations can be broken down into two separate 2D problems.

We consider a single planar occluder parallel to the light source at distance d_2 away from the light source. The occluder is defined by its transparency function $g()$ in this plane, where $g()$ takes a 1D spatial parameter in flatland. Because a ray (v, y) intersects the occluder at spatial coordinate $(d_2v + y)$, the visibility function $f(v, y)$ is defined by

$$f(v, y) = g(d_2v + y), \quad (1)$$

where a value of one is fully visible and a value of zero is fully occluded. We will extend this to occluders with a range of depths later. As seen in Figure 3(c), each occluder creates a diagonally shaped band in the x - y pixel-light space, and all bands are multiplied together to get the final visibility function. Our method efficiently exploits the coherence of these diagonal bands across the light field.

For shadow calculations we use the shadow light field $f(v, y)$ in conjunction with a single receiver point. The receiver is parameterized by a plane at a distance d_1 from the light, and an offset x along the plane. Note that we do not assume that all points are coplanar; we allow d_1 to vary with x . The incoming irradiance, with shadow $h(x)$, is

$$h(x) = r(x) \int f\left(\frac{x-y}{d_1}, y\right) l(y) dy, \quad (2)$$

where $l(y)$ is the intensity of the light source, and $r(x)$ captures the geometric form factor from the receiver point to the area light (separating the form factor from the visibility is a common approximation [Soler and Sillion 1998]). Since $r(x)$ is independent of shadows, we will omit it from later derivations. In our current implementation we consider diffuse BRDFs, and the reflected color will simply be the surface color multiplied by $h(x)$ (see Section 7 for a discussion of more general BRDFs). Note that in many applications, BRDFs are split into a diffuse component and a glossy component, with shadowing applied to the diffuse component and a different reflection technique employed for the glossy component.

3.1 Fourier Analysis

A Fourier analysis enables the design of a filter that is customized to the frequency content of shadows. Capital letters like F , G , and H denote Fourier transforms. Figure 4 shows the process of mapping a given occluder spectrum into the receiver's local parameterization.

We first use the Fourier transform to compute the frequency spectrum of visibility $\mathcal{F}(f(v, y))$. Appendix A provides a detailed algebraic derivation—that also follows directly from Eq. (1) and the Fourier linear transformation theorem [Bracewell et al. 1993],

$$F(\Omega_v, \Omega_y) = G(\Omega_y)\delta(\Omega_v - d_2\Omega_y), \quad (3)$$

where $\delta(\cdot)$ is a delta function. For a constant depth, Eq. (3) shows that the occluder spectrum lies along a line with slope $1/d_2$, as seen in Figure 4(a). For practical scenes with a range of depths, the occluder spectrum has a wedge shape, shown in Figure 4(b), with slopes bounded by $[1/d_{2min}, 1/d_{2max}]$ [Chai et al. 2000]. The approximated bandlimit for the occluder function $g()$ is Ω_g^{\max} , and it bounds the $F(\Omega_v, \Omega_y)$ spectrum along Ω_y (Figure 4(b)).

Our next step is to consider the frequency spectrum of the shadow light field on the receiver, rather than in its canonical parameterization. This transformation is shown in Figure 4(c), and involves both a scale and a shear. Formally, we compute $\mathcal{F}[f(\frac{x-y}{d_1}, y)]$ from Eq. (2), using the Fourier linear transformation theorem or the detailed derivation in Appendix A,

$$\mathcal{F}\left[f\left(\frac{x-y}{d_1}, y\right)\right] = d_1 F(d_1\Omega_x, \Omega_y + \Omega_x), \quad (4)$$

where we now use x and Ω_x along the receiver rather than v and Ω_v . Note the scaling $d_1\Omega_x$ in the first argument. The further the receiver point is from the light (large d_1), the more compressed the frequency spectrum (shadows are smoother). On the other hand, for a receiver point close to the light (small d_1), the frequency spectrum is less compressed, with high-frequency effects near contact shadows.

We now take the Fourier transform of $h(x)$ to find shadow frequencies on the receiver. We use the fact that the integral in Eq. (2) can be seen as convolving the product of f and l with a constant function of 1 to derive Eq. (5). It follows that in the Fourier domain we only need the constant (zero frequency) of $(F \otimes L)$, where \otimes represents a convolution.

$$H(\Omega_x) = d_1 \int F(d_1\Omega_x, \Omega_y + \Omega_x)L(-\Omega_y) d\Omega_y \quad (5)$$

The calculation of $H(\Omega_x)$ is done by integrating over the product of the light frequencies L and the occlusion function F . In other words, to compute the shadow frequencies for H , we need to find all places where the nonzero amplitudes of L and F overlap, as shown in Figure 4(e).

3.2 Relation to Parallel Plane Convolution

The preceding results generalize the seminal parallel plane convolution result of Soler and Sillion [1998]. In particular, if d_1 and

d_2 are fixed, we simply substitute Eq. (3) into Eq. (4), so that the frequencies of the shadow light field are reparameterized for Ω_x at the receiver.

$$\begin{aligned} \mathcal{F} \left[f \left(\frac{x-y}{d_1}, y \right) \right] &= d_1 G(\Omega_y + \Omega_x) \delta(d_1 \Omega_x - d_2(\Omega_y + \Omega_x)) \\ &= \left(\frac{d_1}{d_2} \right) G \left(\frac{d_1}{d_2} \Omega_x \right) \delta \left(\left(\frac{d_1}{d_2} - 1 \right) \Omega_x - \Omega_y \right) \end{aligned} \quad (6)$$

$$(7)$$

In the last line, we bring the d_2 factor outside of the delta function and then use the delta function to set $\Omega_y = (d_1/d_2 - 1)\Omega_x$. Note that this implies that the occluder spectrum in the receiver coordinate space will have Fourier slope $d_1/d_2 - 1$. In our case, the spectrum is not simply a line, but a wedge with slopes ranging from $d_1/d_{2\min} - 1$ to $d_1/d_{2\max} - 1$, as shown in Figure 4(d).

If we now substitute Eq. (7) in Eq. (5), the integral involves a delta function, and will therefore simply result in the integrand, in particular $L(-\Omega_y)$, being evaluated at $\Omega_y = (d_1/d_2 - 1)\Omega_x$,

$$H(\Omega_x) = \left(\frac{d_1}{d_2} \right) G \left(\frac{d_1}{d_2} \Omega_x \right) L \left(\left[1 - \frac{d_1}{d_2} \right] \Omega_x \right), \quad (8)$$

which is a simple multiplication in the frequency domain, and hence a (suitably reparameterized) convolution in the spatial domain.¹

Our method generalizes this approach by keeping all needed frequencies of F for a range of depths (thus considering a frequency wedge for F rather than a simple line), and therefore allowing the receiver and blockers to be general (they need not be restricted to parallel planes). Note also that Figure 4(e) therefore involves an integration against the full wedge; when this wedge reduces to a line, the integration becomes a simple multiplication in frequency space, or a primal-space convolution as in Soler and Sillion [1998].

4. SHEARED FILTER

In this section, we present a new sheared filter that operates over shadow light fields. In the Fourier domain, we design our sheared filter to be as compact as possible to enable the tight packing of replicas in the Fourier domain and sparse sampling in the primal domain. Our filter must cover the overlap of the light L and occluder F spectra to reconstruct the shadow signal H accurately. In Section 5, we will use the shape of the sheared filter in the primal domain to enable sparse sampling across our 4D ray database. We begin by calculating the width and shear of the sheared filter in the Fourier domain. We then examine how to apply transformations to convert a simple axis-aligned filter into a sheared filter in the Fourier and primal domains. Because shadow receivers integrate over irregular domains in the light field, our final step is to transform the primal filter from the (x, y) parameterization to the receiver-independent (v, y) parameterization used for our ray database.

Simple and Sheared Filter Shapes. We first look at a simple filter in Fourier space that covers all displayable frequencies, as shown in Figure 5(a). This simple filter is axis aligned in (Ω_x, Ω_y) space and captures all frequencies within $\Omega_x \in [-\Omega_{\text{pix}}^{\max}, \Omega_{\text{pix}}^{\max}]$ and $\Omega_y \in [-\Omega_y^{\max}, \Omega_y^{\max}]$, where Ω_y^{\max} is the bandlimit of the light

¹Our notation differs slightly from Soler and Sillion [1998], with their d_1 corresponding to our $d_1 - d_2$, and their α corresponding to our $(d_1/d_2) - 1$. We also use x for receiver and y for light source, instead of vice versa. Finally, they integrate over a 2D light source, causing the outside factor in their convolution equation to be squared.

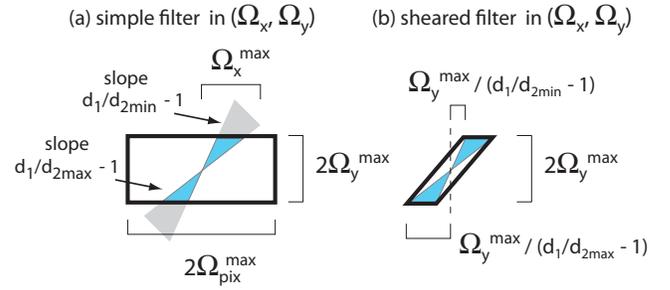


Fig. 5. (a) A simple filter that captures all displayable frequencies in (Ω_x, Ω_y) . The bandlimits for display are the pixel bandlimit $\Omega_{\text{pix}}^{\max}$ and the light bandlimit Ω_y^{\max} . (b) Our new sheared filter compactly covers the same nonzero frequencies that the simple filter does, but its compact shape enables much sparser sampling rates.

intensity function $I(y)$, and $\Omega_{\text{pix}}^{\max}$ is the maximum frequency in x that can be displayed in the output image. In the pixel domain the $\Omega_{\text{pix}}^{\max}$ bandlimit is easy to define as 0.5 wavelengths per pixel. By measuring the projected x distance that a given pixel subtends, we can simply set $\Omega_{\text{pix}}^{\max}$ to 0.5 wavelengths per subtended x pixel distance.

Our sheared filter, shown in Figure 5(b), has the same spectral extent along Ω_y as the simple filter, but our filter is scaled and sheared to compactly bound the nonzero frequencies. Based on the distances from the $\Omega_x = 0$ axis, as shown in Figure 5(b), we can see that the width of our filter in the Fourier domain is simply the difference of these two offsets $\Omega_y^{\max}((d_1/d_{2\max} - 1)^{-1} - (d_1/d_{2\min} - 1)^{-1})$. Similarly, the shear is the ratio between the height of the filter and the average of the offsets $\frac{1}{2}((d_1/d_{2\max} - 1)^{-1} + (d_1/d_{2\min} - 1)^{-1})$.

Transformation to Sheared Filter in Primal Domain. As with many previous analyses, the key insights come from Fourier theory, but our practical implementation operates directly on primal domain samples, and does not need explicit Fourier transforms. Now that we know the exact dimensions and slope of the sheared filter in the Fourier domain, we can derive the transformations necessary to convert a simple filter into a sheared filter in the Fourier domain. Knowing the Fourier domain transformations then makes it easy to compute the corresponding primal domain transformations.

In the Fourier domain, the first step is to scale along Ω_y by the sheared filter width divided by the simple filter width. Fourier theory dictates that for the primal domain we need to scale along x by the inverse amount.

$$\text{primalScale} = \frac{2\Omega_{\text{pix}}^{\max}}{\Omega_y^{\max}} \left[\left(\frac{d_1}{d_{2\max}} - 1 \right)^{-1} - \left(\frac{d_1}{d_{2\min}} - 1 \right)^{-1} \right]^{-1} \quad (9)$$

The next step in the Fourier domain is to shear in Ω_y per unit Ω_x . In this case, Fourier theory tells us that we need to shear by the negated amount in y per unit x .

$$\text{primalShear} = -\frac{1}{2} \left[\left(\frac{d_1}{d_{2\max}} - 1 \right)^{-1} + \left(\frac{d_1}{d_{2\min}} - 1 \right)^{-1} \right] \quad (10)$$

The original shape of the simple filter in the primal domain is axis aligned, integrating over the projected x pixel distance and the light source y extent, as shown in Figure 6(a). Using the transformations of Eqs. (9) and (10), this simple filter is transformed into a sheared

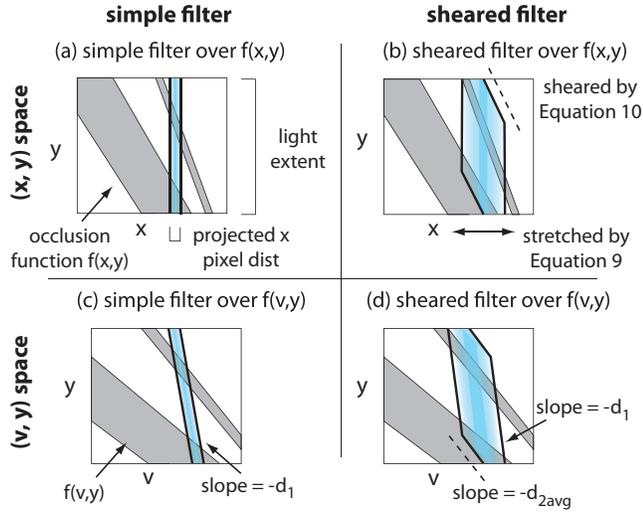


Fig. 6. The occlusion signal f and the simple and sheared filters in the primal (x, y) and (v, y) domains. (a) The simple filter is axis aligned in (x, y) . (b) We create the sheared filter shape by taking the simple filter in (a) and applying the transformations in Eqs. (9) and (10). (c) The simple filter transformed to (v, y) using Eqs. (11) and (12). (d) The sheared filter transformed to (v, y) using Eqs. (11) and (12). The d_{2avg} slope of the filter is between d_{2min} and d_{2max} . Our ray database stores samples in (v, y) so this is the final shape of our filter in flatland. For our practical implementation in 3D there are two additional dimensions (v_2, y_2) that form an orthogonal subspace and independently undergo the same transformations.

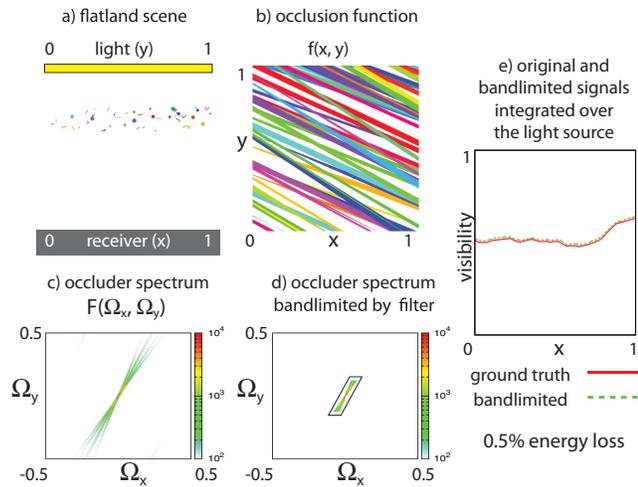


Fig. 7. Numerical verification of our Fourier theory. (a) A set of occluders with random orientations and positions. (b) Graph of the occlusion function $f(x, y)$ for this scene. (c) We take the Fourier transform of $f(x, y)$ to get $F(\Omega_x, \Omega_y)$. (d) Our method captures frequencies inside the footprint of the sheared filter. All frequencies outside the filter are set to zero. (e) A graph of the original signal $f(x, y)$ and the corresponding bandlimited signal from (d). Each point x integrates over the light y to obtain the final visibility.

filter, as shown in Figure 6(b). Note that the shearing seeks to align the filter with the diagonal bands from occluders.

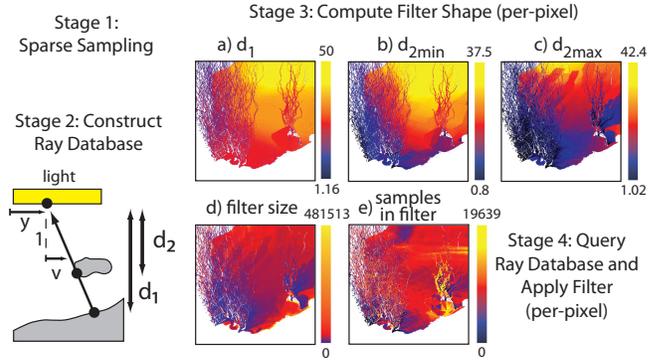


Fig. 8. Individual steps of the algorithm illustrated using the scene from Figure 13. Stage 1 traces rays and writes samples to disk. Stage 2 converts the 3D information to the (v_1, v_2, y_1, y_2) 4D parameterization, along with distances d_1 and d_2 . Stage 3 computes the shape of the filter using d_1 , d_{2min} , d_{2max} . Stage 4 uses the filter to average together multiple samples and compute the shadow signal.

Sheared Filter in (v, y) . In our implementation we store all samples in (v, y) space, so the last step is to transform the primal filter from (x, y) to (v, y) , as shown in Figures 6(c) and 6(d). We know that $(v, y) = ((x - y)/d_1, y)$, and from this we can derive that we first need to shear -1 units in x per unit y , then scale by $1/d_1$ in x .

$$vyShear = -1 \tag{11}$$

$$vyScale = \frac{1}{d_1} \tag{12}$$

Numerical Verification. We verify our frequency analysis by plotting a complex scene in flatland, and examining the Fourier transform, as shown in Figure 7. Our flatland scene is composed of thin and round elements randomly placed in a depth range near the light source, with x and y both ranging between 0 and 1 (Figure 7(a)). In Figure 7(b) we graph the occlusion function in (x, y) , using the object colors from Figure 7(a) as a means to visualize. We then take the Fourier transform of the occlusion function, as shown in Figure 7(c). Note that the Fourier spectrum has our predicted wedge shape, and that amplitudes dissipate rapidly in areas farther from the constant zero frequency. We then zero out all frequencies that are outside of a sheared filter that covers a light bandlimit of Ω_y^{max} equal to $\frac{1}{8}$, as shown in Figure 7(d) (in this example 0.5% of the energy lies outside of the filter). Using both the original Fourier spectrum as well as the bandlimited spectrum, we convert back to the primal domain and integrate with the light source to compute the final receiver values (Figure 7(e)). This final graph of visibility shows that sheared filters can reconstruct shadow signals with minimal loss of fidelity, despite the high-frequency nature of the original signal.

5. ALGORITHM

Our rendering system provides a practical way to sample occlusion in the scene, produce a ray database of all samples, and reconstruct the shadows based on the previous analysis. We use a very sparse sampling, often 1 shadow ray per pixel, and a suitably wide sheared reconstruction filter at each pixel. An overview of the algorithm is depicted in Figure 2 and detailed steps are shown in Figure 8.

Shadow Sampling. The first stage is to sparsely sample occlusion in the 4D light field (Stage 1 in Figure 8). To generate samples we trace a small number of shadow rays from each pixel that contains a receiver surface. Our sampling is driven by the receivers that are visible in the actual image, unlike most other light field techniques that sample uniformly and densely in the 4D space of rays. Our implementation uses a simple programmable shader that traces rays from the receiver to the light source, and for each sample writes the receiver point, ray direction, and the occluder distance d_2 to a file (if the ray is unoccluded we set d_2 to -1).

In principle, we could compute sampling rates at each pixel, directly from the Fourier analysis, as described in Appendix B. In practice however, we have found that areas that receive soft shadows need very sparse sample counts, on the order of 1 to 8 samples per pixel. We have developed a program to do adaptive sampling, but the quality of the final image is usually easier to control by simply setting a uniformly low sample density.

When we shade a receiver point that lies inside the shadow light field’s $[d_{2\min}, d_{2\max}]$ depth bounds, our theory can no longer provide tight bounds on the spectrum, and we cannot safely apply a shear or scale to the primal filter (this can happen with self-shadowing and other cases of near-field occlusion). In this case we simply revert back to using stratified Monte Carlo sampling.

Ray Database Construction. The second stage reads samples from disk, computes the 4D parameterization of each ray, and stores the sample into a 4D ray database (Stage 2 in Figure 8). Our current implementation uses a simple 2D grid as an acceleration structure, indexing across direction parameters v_1 and v_2 . We have experimented with a 4D grid and other bounding volumes, but have found so far that they delivered little speedup when queried with the highly anisotropic shapes and varied orientations of sheared filters generated by a practical scene. The depth range of the light field, $[d_{2\min}, d_{2\max}]$, is also calculated at this stage.

The memory requirements for our method are small, consisting only of loading the ray database into memory. Each sample in the ray database consists of (v_1, v_2, y_1, y_2) 32-bit floating point coordinates, with an additional distance d_2 , that stores the distance to the closest occluder or indicates an unoccluded ray. For the scene in Figure 1 the final ray database was 17MB.

Shadow Reconstruction. To reconstruct shadows, another rendering pass uses a programmable shader that accesses the ray database. For each receiver point, the shader computes the shape of the appropriate sheared filter, queries the ray database with the filter shape, and weights all samples inside the filter’s 4D footprint (Stage 3 in Figure 8).

The first step is to compute d_1 , $d_{2\min}$, and $d_{2\max}$ for the current receiver as shown in Figure 8(a)–(c) (see Optimizations that follows for more details). The next step is to compute the shape of the sheared filter. Although the shape of a sheared filter in 4D may be hard to visualize, it is simple to compute: We require that the planar area light is parameterized with orthogonal basis vectors, guaranteeing that (v_1, y_1) and (v_2, y_2) span orthogonal 2D subspaces of the 4D light field. Consequently, we treat the sheared filter as the product of two 2D sheared filters in (v_1, y_1) and (v_2, y_2) . For each 2D subspace, we first determine the basis vectors that define the light and pixel filter extent of a simple filter in (x, y) (Figure 6(a)). We then transform the basis vectors using Eqs. (9)–(12) such that the basis vectors now represent the centerline and “shear axis” of the sheared filter in (v, y) (Figure 6(d)). The range of primalScale values (see Eq. (9)) is shown in Figure 8(d), as well as the total number of samples inside the filter in Figure 8(e).

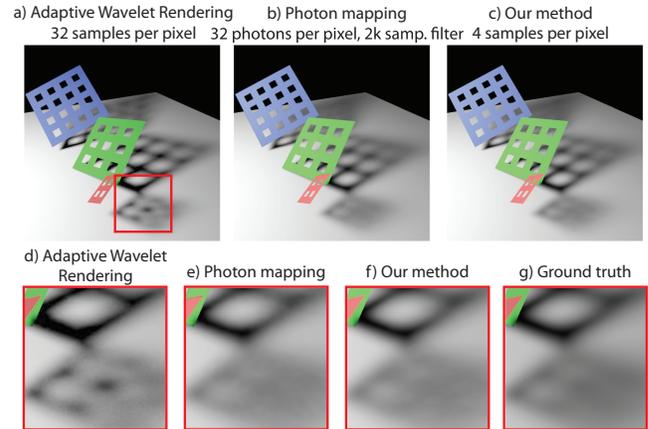


Fig. 9. (a) Adaptive Wavelet Rendering with 32 samples per pixel. (b) Direct visualization of photon map to compute soft shadows with 32 photons per pixel (photon map requires 1.8GB of memory). The soft shadows near the top of the image still have a fair amount of noise. (c) Our method using 4 samples per pixel. (d) Adaptive Wavelet Rendering computes a soft image, but the shape of the soft shadow is slightly off in this case. (e) Direct visualization of the photon map has converged in this area, but other areas have noise. (f) Our method accurately reconstructs the soft shadow signal. (g) Ground truth using 2048 rays with Monte Carlo.

The next step is to determine where the filter is centered. Focusing on the (v_1, y_1) dimensions, and given the positioning of the receiver point and the light in 3D space, we can compute a v_1 value (ray direction) for any given y_1 value (light position). It is convenient to compute v_1 for $y_1 = 0$ since this is always defined to be one edge of the light in our implementation. Similarly we compute v_2 for $y_2 = 0$, completely anchoring the centerline of our 4D sheared filter.

We have now defined the placement and shape of our filter. We now process every sample in every grid cell that lies inside the filter’s v_1 and v_2 extents. We then calculate the sample’s coordinates relative to the transformed light extent and pixel extent basis vectors. These coordinates can then be interpreted in the original (x, y) space, where one coordinate determines the pixel filter response and the other determines the light intensity (Stage 4 in Figure 8).

Optimizations. Computation of the depth bounds $d_{2\min}$ and $d_{2\max}$ can often be done by simply evaluating the global range of d_2 occluder values contained in the ray database (this was done for Figure 1). For more complicated scenes with many interacting occluders and receivers, it becomes necessary to compute $d_{2\min}$ and $d_{2\max}$ per receiver (this was done for Figures 12 and 13). To compute $d_{2\min}$ and $d_{2\max}$ per receiver, we precompute a 3D hierarchical sphere tree with all occluder positions to supplement the ray database. For each receiver we cull out points that are outside of the receiver-light frustum, and then compute the $[d_{2\min}, d_{2\max}]$ bounds on the remaining points. This computation is inexpensive relative to filtering, but it can at times lead to discontinuities in $d_{2\min}$ or $d_{2\max}$ across receiver points.

6. RESULTS

We demonstrate our results with five scenes, which showcase a variety of challenging situations in Figures 1, 9, 11, 12, and 13. We also show comparisons and timings with respect to stratified Monte Carlo sampling, the current method of choice, as well as optimizations like photon mapping and the recent development of Adaptive Wavelet Rendering [Overbeck et al. 2009].

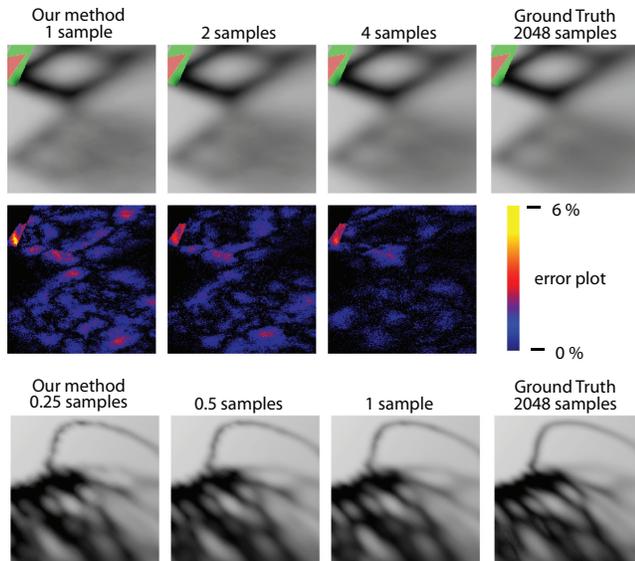


Fig. 10. Analysis of different quality levels for different sample counts. The first and second rows show a closeup from the scene in Figure 9. As the quality increases to four samples we can see in the error plot that we are converging. The third row shows a new closeup from the scene in Figure 1, near the upper part of the shadow. We see here the spotting artifacts that can occur in areas of undersampling.

All examples were run using Pixar’s Renderman Pro Server 15.0 on a dual quad-core Xeon 2.33 GHz processor with 4GB of memory. Due to our modular plug-in architecture, our code trivially runs in parallel for any number of threads. All scenes in this paper use a planar area light with a circular Gaussian falloff that captures two standard deviations within the light radius.

6.1 Canonical “Grid” Scene

We start with the canonical scene in Figure 9, which shows grid occluders with shadows that smoothly go from sharp to wide. The very regular and smooth growth of the penumbra makes small artifacts easier to spot, but our method produces high-quality results.

Figure 9 also compares to alternative rendering approaches, like photon mapping [Jensen and Christensen 1995] (Figures 9(b) and 9(e) use similar parameters to those in our method). We see that direct visualization of the photon map has converged in some areas, while other areas still have a fair amount of noise, even when using 32 photons per pixel. In addition, storing the 33M photons takes 1.8GB of memory in Renderman’s implementation. For these reasons, the photon map is usually used for caustic or global illumination effects, and rarely visualized directly for soft shadows.

Therefore, in the remainder of the article, we focus on comparing to stratified Monte Carlo sampling, and to the state-of-the-art adaptive reconstruction method, Adaptive Wavelet Rendering (AWR) [Overbeck et al. 2009]. The AWR comparisons in Figure 9 (and 1) directly use the original AWR software, with the same scene setup and light source location and falloff as our method. We see in Figure 9 that our method better captures the widening blur of some areas of the shadow signal. Adaptive Wavelet Rendering with 32 samples per pixel has not converged due to the high variance of the shadow signal.

We analyze the effect of increasing sample counts in our method in the first two rows of Figure 10 (we will analyze the bottom row in

Section 6.2). We first note that even when using only one sample or shadow ray per pixel, our method is quite accurate, with the maximum pixel error less than 6%. We are able to use such low sample counts because our sheared reconstruction filter effectively shares samples between many neighboring pixels. However, some difficult regions can be noisy, and these rapidly become more accurate with a moderate increase in sample count. Indeed, 4 samples per pixel reduces error almost to 0 everywhere in the image.

6.2 Detailed Occluding Geometry

In Figure 1, we show a detailed model (1.3M triangles) with many complex silhouettes and thin features casting a shadow on a flat receiver. Our method works well in this case because it can handle complex occluders, and the scene has predominantly mid- and far-field occlusion, which lets our method use vastly fewer samples than other algorithms. For this scene we traced 1 ray per pixel during the initial sampling phase and created a ray database with 637,000 samples. In this scene we used Monte Carlo ray tracing with 4 rays for self-shadowing within the occluder and 64 rays for near-field occlusion on the receiver (a higher number of rays were necessary because of the extremely thin features of the occluder). We used the Monte Carlo solution for receiver positions with $d_1 \leq d_{2\max}$, and did a smooth blend between the Monte Carlo solution and our solution up to a user-specified distance of $(1.1)d_{2\max}$ (as shown in the Figure 1(a) inset).

Figure 11 shows another difficult example with two complex tree occluders, this time shadowing a curved receiver. The tree trunk and branches are modeled with subdivision surfaces. In Figures 11(a)–(d), a medium-sized light source is used, and ray tracing shadows is fairly coherent. In Figures 11(e)–(h) a larger light source is used, causing incoherence among rays and a much more expensive cost per ray. Our method is most beneficial when the cost per ray is high, which can be seen in more detail in the timings section that follows.

Stratified Monte Carlo sampling is still usually the method of choice for high-end rendering. However, for complex occluders in scenes like Figures 1 and 11, stratification has minimal benefit, since every shadow ray has very high variance; there is almost no coherence across the occlusion signal. Therefore, stratified Monte Carlo sampling requires a very large number of samples before the variance of the shadow is not visually noticeable (approximately 2048 samples in our case). While Adaptive Wavelet Rendering in Figure 1(g) is beginning to converge with 64 samples, the high variance of the signal leads to some low amplitude aliasing in the wavelet basis. In contrast, the third row of Figure 10 shows that our method can get decent results even when using 0.25 samples per pixel during the sparse sampling stage. In this case most of the spotting artifacts are removed by going up to just 1 ray per pixel.

Timings. We report wall clock running times for Figures 1 and 11. For both images we measure the cost of rendering the right half of the image, since the shadows are mostly concentrated there (the costs on the left side of the image are dominated by scan conversion of the occluding geometry, which is not relevant to our or other algorithms). In our tests, we noticed that timing results can be fairly nonlinear with the number of rays traced per pixel, which we believe is primarily dependent on how ray trace queries interact with Renderman’s geometry caching algorithm. We therefore report numbers for canonical numbers of samples, which allow for equal time and quality comparisons with stratified Monte Carlo.

In Figure 1(a), our method (with 1 ray per pixel) took 1 min 17 sec for the sparse sampling phase and 4 min 1 sec to reconstruct the shadows using the ray database, for a total time of 5 min 18 sec. In

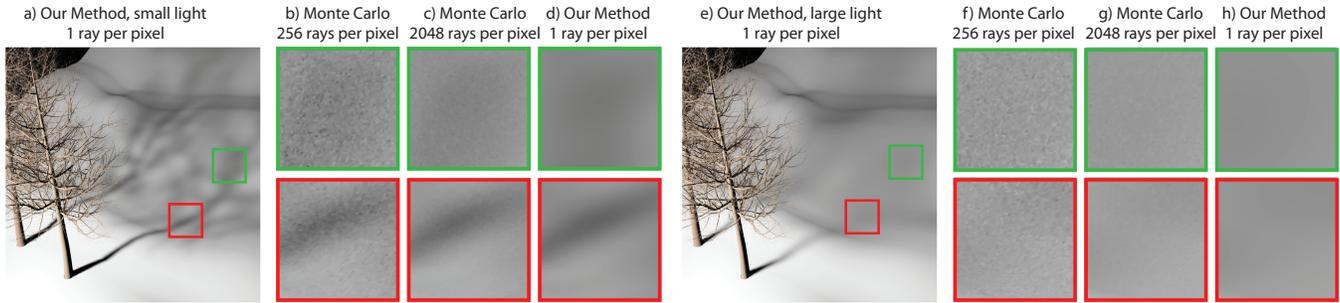


Fig. 11. Comparisons between our method and stratified Monte Carlo sampling. We show two scenes: a smaller area light in (a) through (d), and a larger area light in (e) through (h). The timings show that as rays become more incoherent (in this case due to a larger light source), brute-force ray tracing becomes very expensive. This is primarily due to the cost of updating the cached geometry for ray tracing. All timings are for rendering the right half of the image where shadow computation dominates the overall time of execution (the one exception is our sparse sampling pass, which processes the entire image).

Figure 1(b), Monte Carlo sampling with 256 samples per pixel took 5 min 6 sec, and in Figure 1(e), Monte Carlo sampling with 2048 samples took 23 min 3 sec.

In Figure 11(a), our method (with 1 ray per pixel) took 5 min 25 sec for sparse sampling, 1 min 25 sec for reconstruction, and a total of 6 min 50 sec. Monte Carlo with 256 samples in Figure 11(b) took 6 min 9 sec, and Monte Carlo with 2048 samples in Figure 11(c) took 16 min 19 sec. The wider area light source in Figure 11(e) produces more significant speedups because of the incoherent shadow rays. Our method took 6 min 32 sec for sparse sampling, 6 min 30 sec for reconstruction, and 13 min 2 sec total. Monte Carlo with 256 samples in Figure 11(f) took 1 hr 15 min, and Monte Carlo with 2048 samples in Figure 11(g) took 3 hr 34 min, for a net speedup for our method of more than an order of magnitude.

The AWR implementation uses an optimized packet ray tracer and a more stripped down shading system for speed, making comparisons to our Renderman plug-in difficult. They report taking 34 sec to reconstruct their wavelet basis using 32 samples at image resolutions of 1024×1024 . In our test scenes it appears that any noise in the wavelet basis is not visually noticeable after 256 samples. Because our method uses drastically fewer samples, our method will be preferable whenever ray tracing is expensive, such as for highly tessellated models that may not fit into main memory. Note that in our results we have shown significant performance gains relative to the highly optimized Renderman ray tracer.

In general we note that our system drastically reduces ray tracing computation, making a trade for increased filter computation. We have already shown that this is beneficial even for moderately complex scenes. A commonly noted trend in production rendering is that when computational power increases, artists will immediately increase the complexity of their scenes rather than enjoy faster render times. As long as this trend continues, geometric complexity for rendered scenes will increase, and our substitution of ray casts for filter computation will become more and more valuable.

6.3 Robustness: Complex Occluders and Receivers

Figure 12 shows a scene with many interacting occluders and receivers. In Figure 12(d), we show that the range of occluder depths $[d_{2\min}, d_{2\max}]$ computed at each pixel can vary by large amounts. In Figure 12(e) we show the number of pixels contained in each pixel's custom reconstruction filter. Note that our method produces smooth results because of the high number of samples processed by

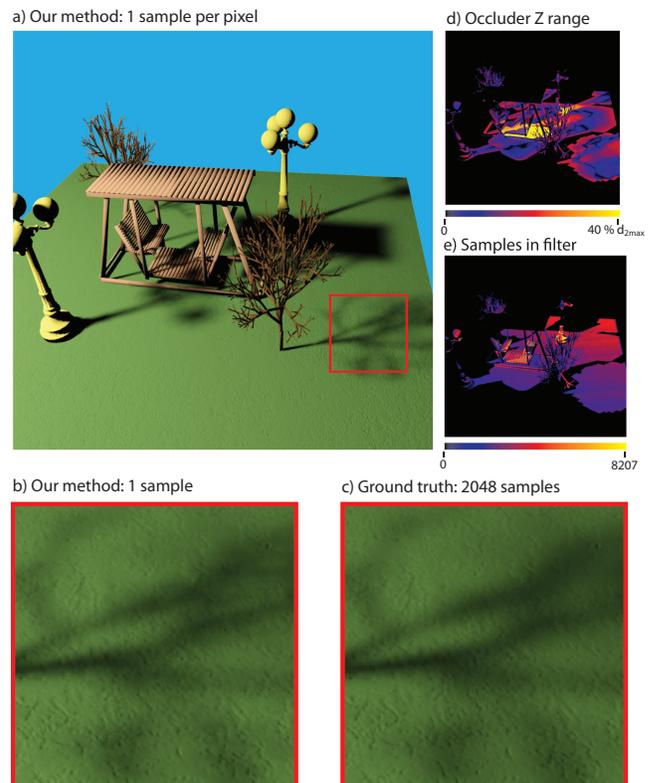


Fig. 12. Scene with a wide array of occluders and receivers, as well as a curved ground surface with high frequency displacement.

each pixel. Finally, we show a scene with the foliage model from Figure 1, as well as a complex displacement-mapped receiver surface, and a number of other objects. This scene showcases a variety of intricate shadowing effects, such as complex objects casting and receiving shadows. In both scenes we used 1 sample per pixel for the sparse sampling stage. In Figure 13 we used 4 Monte Carlo samples per pixel to compute near-field occlusion, while in Figure 12 we did not use any Monte Carlo sampling. Figures 12 and 13 show the robustness of our method for dealing with a range of complex occluder and scene configurations.

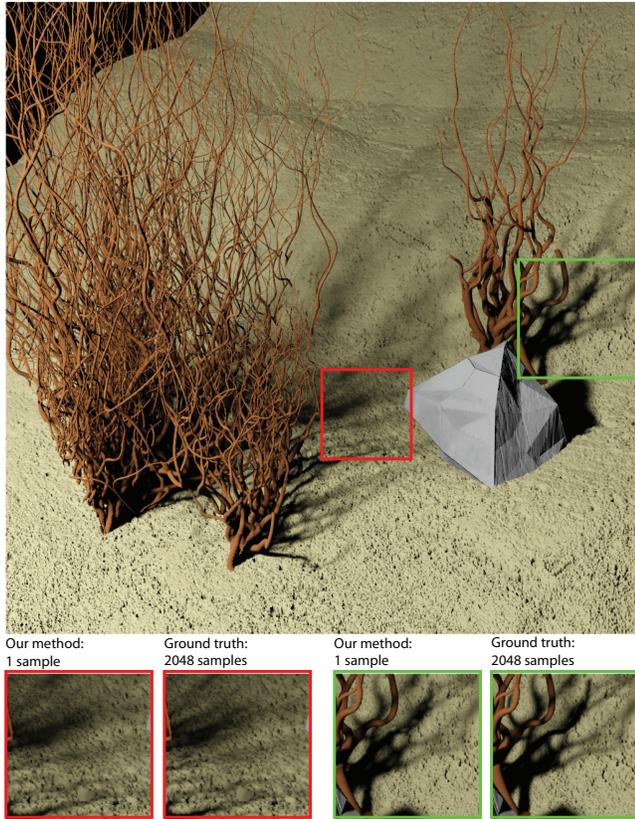


Fig. 13. Complex scene with multiple occluders and receivers.

6.4 Animation

We have focused on still images, but it is also interesting to examine whether our method can produce stable animations. We show that our method can indeed produce high-quality animations, but may require a higher sampling rate to eliminate temporal aliasing.

In our supplementary video we animate the grids and trees scenes (stills from the video are shown in Figure 14). In the grids animation, the grids descend towards the ground plane, and we show that with 0.3 rays per pixel there are noticeable artifacts, but these artifacts go away using 3.0 rays per pixel. We rotate both trees in the tree scene (Figure 11(a)) to provide a stress test of many thin occluders moving relative to each other. In this case with 3.0 rays per pixel, the still images are often visually acceptable, but flickering can be seen as the tree rotates during animation. Small amounts of undersampling may cause medium to low frequency error relative to ground truth, but these errors are often visually imperceptible for still images. However, during animation these small errors can flicker, which is much more noticeable. When we increase the sampling rate to 10.0 rays per pixel the animating shadows become more stable and the flickering artifacts disappear. With 10.0 rays per pixel the sparse sampling pass took 5 min 46 sec, 16 min 16 sec for reconstruction (right half of 1k image), and a total of 22 min 2 sec.

Our supplementary video also compares our results to Monte Carlo integration during animation. Even with 2048 samples per pixel a small amount of noise is still visible in the animation using Monte Carlo. Our method with 10.0 rays per pixel delivers a smoother result with no visual flickering.

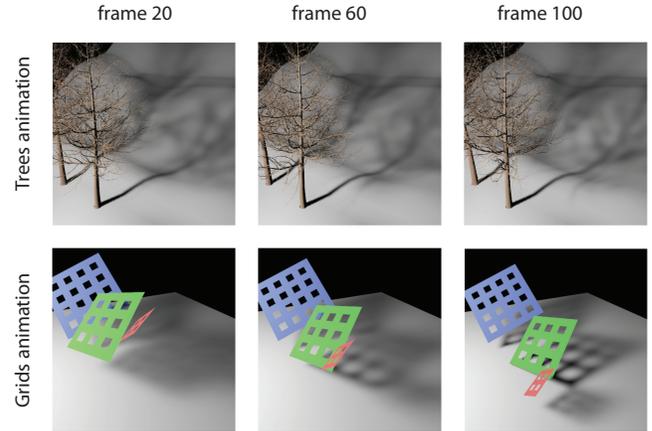


Fig. 14. Still frames from our supplementary video. The grids animate relative to the ground plane (3.0 rays per pixel). The trees each rotate relative to the ground plane (10.0 rays per pixel).

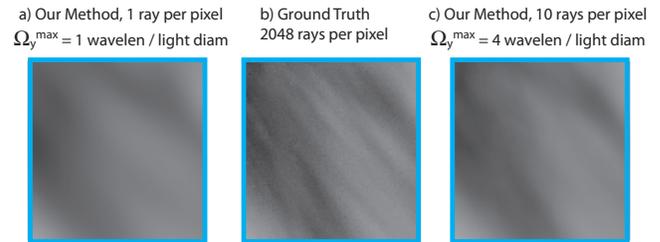


Fig. 15. An inset from Figure 1 with contrast increased by 4x. With 1 ray per pixel and a low value for the Ω_y^{\max} light bandlimit our results are overblurred. By increasing the number of samples to 10 rays per pixel and increasing the Ω_y^{\max} light bandlimit our method can capture more shadow frequencies. However, even with these higher-quality settings our method is missing some detail.

7. ARTIFACTS AND CONVERGENCE

We discuss the limitations and possible artifacts that come from our method. We first look at the artifacts that occur from undersampling occlusion and how the light bandlimit Ω_y^{\max} affects rendering. We also discuss how these two factors affect the convergence of our method. We then look at how undersampling can also manifest itself in the use of the occluder depth bounds d_2^{\min} and d_2^{\max} . Finally we discuss two extreme cases where precise occluder configurations break our assumptions.

Undersampling and Light Bandlimit. If the sparse sampling pass does not adequately sample the occlusion signal, our results will have mid to low frequency artifacts, as seen in Figure 10. Our method can also overblur if we set the Ω_y^{\max} light bandlimit too low, as seen in Figure 15. From Eq. (9) we can see that the higher the Ω_y^{\max} light bandlimit is set, the smaller the scale of the reconstruction filter. Using a smaller filter subsequently requires a higher sampling rate to avoid spotty artifacts. If Ω_y^{\max} is set too low then the reconstruction filter sizes will be large and overblurring may occur.

Undersampling can be more visually noticeable during animation as it can lead to flickering artifacts. This is shown in the supplementary video and discussed in Section 6.4.

In Figure 16 we can see how the sampling rate and Ω_y^{\max} light bandlimit interact with each other. As we increase Ω_y^{\max} we see more

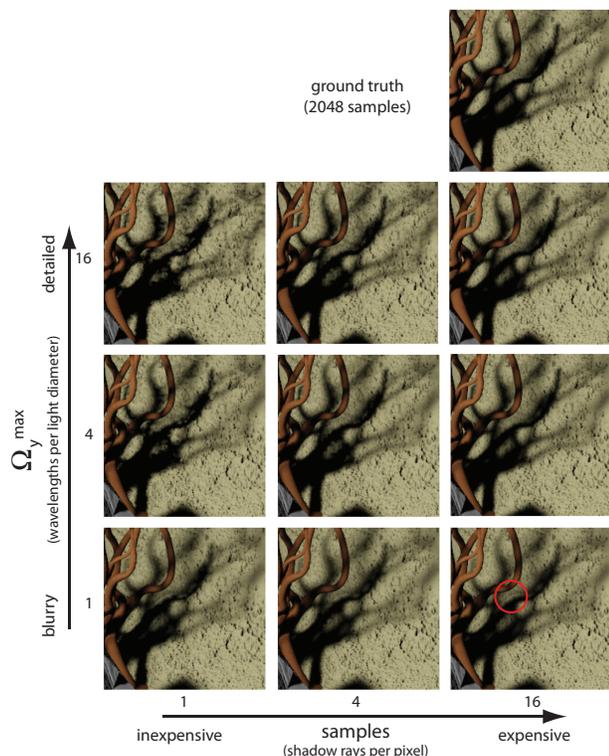


Fig. 16. An inset from Figure 13 rendered with different sampling rates and Ω_y^{\max} light bandlimit. In general the best quality per cost is shown along the diagonal from lower left to upper right (upper right being the highest quality and the most expensive). The upper left image is inexpensive and noisy, and the lower right image is expensive and overblurred. The red circle highlights a ringing artifact that can occur when depth bounds change suddenly.

and more details in the shadow. If we use a large value for Ω_y^{\max} but keep a low sampling rate, we can start to see noise (upper left image in Figure 16). If Ω_y^{\max} is too low then the shadows will stay blurry even as we increase samples (bottom row of Figure 16). For low sampling rates it is best to keep Ω_y^{\max} lower (lower left image in Figure 16), and for high-quality renders that use a high sampling rate it is best to use a higher value of Ω_y^{\max} (upper right image in Figure 16).

Depth Bounds d_2^{\min} and d_2^{\max} . When computing the d_2^{\min} and d_2^{\max} depth bounds per receiver, sudden changes in these bounds can sometimes get ringing artifacts (this can be seen in the bottom row of Figure 16 and in Figure 17(ii)). This is due to one pixel using a filter that is much wider than the neighboring pixel’s (blurring the d_2^{\min} and d_2^{\max} bounds as is done in Egan et al. [2009] would help to alleviate this problem). The depth bounds can also be inaccurate when nearby occluder hit points are culled by the receiver-light frustum, which in turn leads to improper filtering, as seen in Figure 17(i). We believe that this is due to our current implementation using a receiver-light frustum that converges to a point instead of properly covering the entire extent of the receiver pixel.

General BRDFs. Our current implementation only handles diffuse BRDFs. While other reflection techniques are often more appropriate for glossy BRDFs, for future work we would like to extend our method to handle any general BRDF. This can be achieved by replacing the lighting response $l(y)$ with the product of lighting and the BRDF response of the current receiver point $\rho(y)$. If we replace

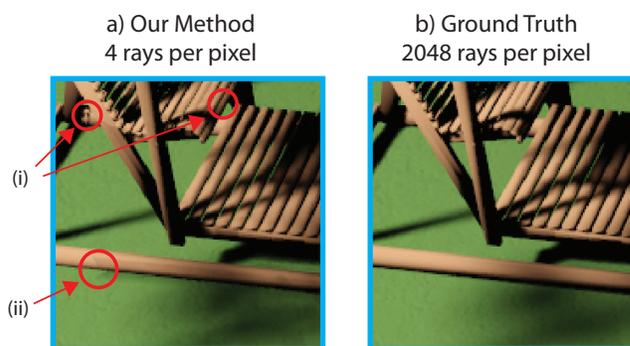


Fig. 17. Our method with 4 samples produces results which are very close to the ground truth (scene is an inset from Figure 12). But there are issues which can produce subtle differences: (i) Shadow regions can be inaccurate if the d_2^{\min}/d_2^{\max} calculation misses occluder samples. (ii) In this case the d_2^{\min}/d_2^{\max} calculation is not smooth, which leads to jumps in the filter size.

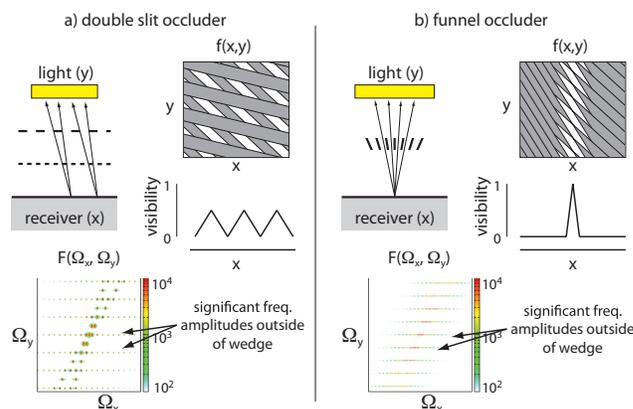


Fig. 18. Failure cases for our method. (a) A double slit configuration that causes the visibility of the receiver to change roughly as a triangle wave. (b) A funnel configuration that shadows all areas except for one small area that transitions to full visibility. The slanted line segments of the funnel have a wedge-shaped occlusion signal in (x, y) . In both failure cases the precise shapes of the occluders create regular patterns aligned along the y -axis, creating sharp changes in visibility across the receiver x -axis. In the Fourier domain there is significant energy in $F(\Omega_x, \Omega_y)$ that exists outside of the modeled wedge.

L with $(P \otimes L)$ in our analysis, we see that for specular BRDFs containing high frequency content we will have less savings, as it becomes more and more difficult to share rays between receivers. Any second-order terms from surface curvature should be minimal, since our analysis is local to the receiver surface subtended by a single pixel.

Theoretical Limitations. For all practical scenes that we have tested, the shape of the occluder spectrum has been a good fit with the wedge shape shown in Figure 4(b). However, there are extreme cases that break the wedged-shaped spectrum assumption used by our method and previous work [Chai et al. 2000; Egan et al. 2009]. We depict the first case in Figure 18(a). Using an array of planar occluders with length and separation proportional to the distance to the receiving plane, the final shadow is roughly a triangle wave (the signal will be an exact triangle wave for infinitely wide area lights). Using this setup, we can create arbitrarily high shadow frequencies

with no change in amplitude, for any two depths, by scaling the length and gaps between occluders closer and closer to zero.

It is also possible to create a funnel-shaped occluder that provides complete visibility to an arbitrarily small area, which then quickly fades to no visibility outside of the area, as shown in Figure 18(b). By squeezing the funnel edges closer and closer together, we can achieve a sharper and sharper “spike” in visibility. The visibility in (x, y) space for both of these cases is also shown in Figure 18. Both cases apply directly to previous work in light field rendering [Chai et al. 2000], and both cases can be applied to motion blur by replacing the receiver plane with a camera that moves across different x positions over time [Egan et al. 2009]. In most practical scenes some high frequencies may exist due to correlation of occluders, but the amplitude of these high frequencies will usually be very low compared to the overall signal.

8. CONCLUSION AND FUTURE WORK

We have presented a new frequency analysis of complex occluders, and a rendering algorithm that leverages sparsity in the Fourier domain of the 4D light field. We have shown large speedups for a range of complex occluders and scene configurations. Furthermore, our results show that our method excels when dealing with very soft shadows, which is precisely where other methods have the most difficulty.

For future work, we would like to look at hierarchical integration methods to speed up filtering. Prefiltering samples before filtering is challenging because the sheared filters used to query the ray database are thin (relative to the overall size of the database), and sheared at many different angles.

Our method delivers the biggest performance gains for soft shadows cast by mid to far occluders. However, we could extend our current system to gracefully handle more general cases of self-shadowing by subdividing occlusion data into multiple light fields. This could provide a large improvement to our method’s performance when occluders are visible at many different depths.

Looking forward, we expect that our generalization of sheared filtering to irregular integrands, as well as the use of more sophisticated filtering techniques, will spur further advances for rendering and other areas.

APPENDIXES

Appendix A: Fourier Derivations

To derive Eq. (3) we have

$$\mathcal{F}[f(v, y)] = \int \int g(d_2v + y) \exp(-i2\pi(v\Omega_v + y\Omega_y)) dv dy \quad (13)$$

$$\begin{aligned} u &= d_2v + y & y &= u - d_2v & dy &= du \\ &= \int \int g(u) \exp(-i2\pi(v\Omega_v + (u - d_2v)\Omega_y)) dv du \\ &= \int \int g(u) \exp(-i2\pi(v(\Omega_v - d_2\Omega_y) + u\Omega_y)) dv du \\ &= \int \left[\int g(u) \exp(-i2\pi u\Omega_y) du \right] \\ &\quad \times \exp(-i2\pi v(\Omega_v - d_2\Omega_y)) dv \\ &= G(\Omega_y) \int \exp(-i2\pi v(\Omega_v - d_2\Omega_y)) dv \\ \mathcal{F}[f(v, y)] &= G(\Omega_y) \delta(\Omega_v - d_2\Omega_y). \end{aligned} \quad (14)$$

To derive Eq. (4) we have

$$\begin{aligned} \mathcal{F}\left[f\left(\frac{x-y}{d_1}, y\right)\right] &= \int \int f\left(\frac{x-y}{d_1}, y\right) \\ &\quad \times \exp(-i2\pi(x\Omega_x + y\Omega_y)) dx dy \quad (15) \\ u &= \frac{x-y}{d_1} & x &= ud_1 + y & dx &= (d_1) du \\ &= \int \int f(u, y) \exp(-i2\pi((ud_1 + y)\Omega_x + y\Omega_y)) (d_1) du dy \\ &= d_1 \int \int f(u, y) \exp(-i2\pi(ud_1\Omega_x + y(\Omega_x + \Omega_y))) du dy \\ \mathcal{F}\left[f\left(\frac{x-y}{d_1}, y\right)\right] &= d_1 F(d_1\Omega_x, \Omega_y + \Omega_x). \end{aligned} \quad (16)$$

Appendix B: Sampling Rates

Sampling in the primal domain creates replicas in the Fourier domain, and the sparser the sampling rate the closer together the replicas are packed. We want to compute the lowest possible sampling rate such that we prevent the replicas from overlapping the footprint of our filter. The compact shape of our sheared filter allows for much tighter packing of replicas, which allows for much lower sampling rates, which in turn leads to faster render times. We can use a derivation similar to Egan et al. [2009] to compute the minimal sampling rates for our sheared filter shape.

$$\Omega_x^* = \Omega_x^{\max} + \Omega_y^{\max} \left(\frac{d_1}{d_{2\max}} - 1 \right)^{-1} \quad (17)$$

$$\Omega_y^* = \Omega_y^{\max} \left(\frac{d_1}{d_{2\max}} - 1 \right) \left[\left(\frac{d_1}{d_{2\max}} - 1 \right)^{-1} - \left(\frac{d_1}{d_{2\min}} - 1 \right)^{-1} \right] \quad (18)$$

In the preceding equations, Ω_x^* and Ω_y^* are the required sampling rates in the x and y dimensions, respectively. These values are derived by measuring the distance between replicas along Ω_x and Ω_y (the exact derivation is omitted for brevity). To compute the number of samples requested by a receiver point we calculate the 4D product $(\Omega_x^*)^2 (\Omega_y^*)^2$ and divide by the 4D volume of the sheared filter in (x_1, x_2, y_1, y_2) (because the subspaces are orthogonal this is simply the product of the filter areas in (x_1, y_1) and (x_2, y_2)). The Ω_x^{\max} bandlimit is the extent of the occluder wedge along Ω_x (see Figure 5(a)). Looking at the shape of the original occluder spectrum, $F(\Omega_v, \Omega_y)$ (see Figure 4(b)), we find that the transformation from (Ω_x, Ω_y) to (Ω_v, Ω_y) results in a Ω_x^{\max} being equal to $\Omega_g^{\max} \frac{d_{2\max}}{d_1}$.

ACKNOWLEDGMENTS

Many of the initial ideas and motivation for this work were conceived in early discussions with M. Meyer and J. Anderson at Pixar. The authors thank E. Grinspun for his support and collaboration in early efforts on the problem. They also thank the reviewers for their insightful comments. R. Overbeck provided much needed code and support for the Adaptive Wavelet Rendering comparisons. T.-H. Lin wrote code for the flatland ray tracer.

REFERENCES

AGRAWALA, M., RAMAMOORTHY, R., HEIRICH, A., AND MOLL, L. 2000. Efficient image-based methods for rendering soft shadows. In *Proceedings of the SIGGRAPH Conference*. 375–384.

- ANNEN, T., DONG, Z., MERTENS, T., BEKAERT, P., SEIDEL, H.-P., AND KAUTZ, J. 2008. Real-Time, all-frequency shadows in dynamic scenes. *ACM Trans. Graph.* 27, 3, 1–8.
- ARIKAN, O., FORSYTH, D. A., AND O'BRIEN, J. F. 2005. Fast and detailed approximate global illumination by irradiance decomposition. In *Proceedings of the SIGGRAPH Conference*. 1108–1114.
- ASSARSSON, U. AND AKENINE-MÖLLER, T. 2003. A geometry-based soft shadow volume algorithm using graphics hardware. *ACM Trans. Graph.* 22, 3, 511–520.
- BALA, K., DORSEY, J., AND TELLER, S. 1999. Radiance interpolants for accelerated bounded-error ray tracing. *ACM Trans. Graph.* 18, 3, 213–256.
- BEN-ARTZI, A., RAMAMOORTHY, R., AND AGRAWALA, M. 2006. Efficient shadows from sampled environment maps. *J. Graph. Tools* 11, 1, 13–36.
- BRACEWELL, R., CHANG, K., JHA, A., AND WANG, Y. 1993. Affine theorem for two-dimensional fourier transform. *Electron. Lett.* 29, 304.
- CHAI, J., TONG, X., CHAN, S., AND SHUM, H. 2000. Plenoptic sampling. In *Proceedings of the SIGGRAPH Conference*, K. Akeley, Ed. ACM Press, 307–318.
- CHEN, W.-C., BOUGUET, J.-Y., CHU, M. H., AND GRZESZCZUK, R. 2002. Light field mapping: Efficient representation and hardware rendering of surface light fields. *ACM Trans. Graph.* 21, 3, 447–456.
- COOK, R. L., PORTER, T., AND CARPENTER, L. 1984. Distributed ray tracing. *Comput. Graph.* 18. ACM, 137–145.
- DURAND, F. 1999. 3D Visibility: Analytical study and applications. Ph.D. thesis, Université Joseph Fourier, Grenoble I.
- DURAND, F., HOLZSCHUCH, N., SOLER, C., CHAN, E., AND SILLION, F. X. 2005. A frequency analysis of light transport. *ACM Trans. Graph.* 24, 3, 1115–1126.
- EGAN, K., TSENG, Y.-T., HOLZSCHUCH, N., DURAND, F., AND RAMAMOORTHY, R. 2009. Frequency analysis and sheared reconstruction for rendering motion blur. *ACM Trans. Graph.* 28, 3, 93:1–93:13.
- GORTLER, S., GRZESZCZUK, R., SZELISKI, R., AND COHEN, M. 1996. The lumigraph. In *Proceedings of the SIGGRAPH Conference*. 43–54.
- HACHISUKA, T., JAROSZ, W., WEISTROFFER, R., DALE, K., HUMPHREYS, G., ZWICKER, M., AND JENSEN, H. 2008. Multidimensional adaptive sampling and reconstruction for ray tracing. *ACM Trans. Graph.* 27, 3, 33:1–33:10.
- HART, D., DUTRÉ, P., AND GREENBERG, D. P. 1999. Direct illumination with lazy visibility evaluation. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. 147–154.
- HASENFRATZ, J.-M., LAPIERRE, M., HOLZSCHUCH, N., AND SILLION, F. 2003. A survey of real-time soft shadows algorithms. *Comput. Graph. Forum* 22, 4, 753–774.
- ISAKSEN, A., McMILLAN, L., AND GORTLER, S. J. 2000. Dynamically reparameterized light fields. In *Proceedings of the SIGGRAPH Conference*, K. Akeley, Ed. ACM Press, 297–306.
- JENSEN, H. W. AND CHRISTENSEN, N. J. 1995. Efficiently rendering shadows using the photon map. In *Proceedings of the Compugraphics Conference*. 285–291.
- JOHNSON, G. S., HUNT, W. A., HUX, A., MARK, W. R., BURNS, C. A., AND JUNKINS, S. 2009. Soft irregular shadow mapping: Fast, high-quality, and robust soft shadows. In *Proceedings of the Symposium on Interactive 3D Graphics and Games (I3D'09)*. 57–66.
- LACEWELL, D., BURLEY, B., BOULOS, S., AND SHIRLEY, P. 2008. Raytracing prefiltered occlusion for aggregate geometry. In *Proceedings of the IEEE Symposium on Interactive Raytracing*.
- LAINÉ, S., AILA, T., ASSARSSON, U., LEHTINEN, J., AND AKENINE-MÖLLER, T. 2005. Soft shadow volumes for ray tracing. *ACM Trans. Graph.* 24, 3, 1156–1165.
- LANMAN, D., RASKAR, R., AGRAWAL, A., AND TAUBIN, G. 2008. Shield fields: Modeling and capturing 3D occluders. *ACM Trans. Graph.* 27, 5.
- LEVOY, M. AND HANRAHAN, P. 1996. Light field rendering. In *Proceedings of the SIGGRAPH Conference*. 31–42.
- NG, R., RAMAMOORTHY, R., AND HANRAHAN, P. 2003. All-Frequency shadows using non-linear wavelet lighting approximation. *ACM Trans. Graph.* 22, 3, 376–381.
- OVERBECK, R., RAMAMOORTHY, R., AND MARK, W. R. 2007. A real-time beam tracer with application to exact soft shadows. In *Proceedings of the EuroGraphics Symposium on Rendering*.
- OVERBECK, R. S., DONNER, C., AND RAMAMOORTHY, R. 2009. Adaptive wavelet rendering. *ACM Trans. Graph.* 28, 5, 1–12.
- RAMAMOORTHY, R., KOUDELKA, M., AND BELHUMEUR, P. 2005. A fourier theory for cast shadows. *IEEE Trans. Patt. Anal. Mach. Intell.* 27, 2, 288–295.
- RAMAMOORTHY, R., MAHAJAN, D., AND BELHUMEUR, P. 2007. A first-order analysis of lighting, shading, and shadows. *ACM Trans. Graph.* 26, 1, 2:1–2:21.
- ROBISON, A. AND SHIRLEY, P. 2009. Image space gathering. In *Proceedings of the Conference on High Performance Graphics (HPG'09)*. ACM, New York, 91–98.
- SHINYA, M. 1993. Spatial anti-aliasing for animation sequences with spatio-temporal filtering. In *Proceedings of the SIGGRAPH Conference*. 289–296.
- SINTORN, E., EISEMANN, E., AND ASSARSSON, U. 2008. Sample-Based visibility for soft shadows using alias-free shadow maps. *Comput. Graph. Forum* 27, 4, 1285–1292.
- SOLER, C. AND SILLION, F. 1998. Fast calculation of soft shadow textures using convolution. In *Proceedings of the SIGGRAPH Conference*, M. Cohen, Ed. ACM Press, 321–332.
- SOLER, C., SUBR, K., DURAND, F., HOLZSCHUCH, N., AND SILLION, F. 2009. Fourier depth of field. *ACM Trans. Graph.* 28, 2, 18:1–18:18.
- STEWART, J., YU, J., GORTLER, S. J., AND McMILLAN, L. 2003. A new reconstruction filter for undersampled light fields. In *Proceedings of the 14th Eurographics Workshop on Rendering (EGRW'03)*. 150–156.
- SUN, B. AND RAMAMOORTHY, R. 2009. Affine double and triple product wavelet integrals for rendering. *ACM Trans. Graph.* 28, 2, 1–17.
- VAN DER LINDEN, J. 2003. Multiple light field rendering. In *Proceedings of GRAPHITE*. 197–ff.
- WALTER, B., ARBREE, A., BALA, K., AND GREENBERG, D. P. 2006. Multidimensional lightcuts. *ACM Trans. Graph.* 25, 3, 1081–1088.
- YANG, B., FENG, J., GUENNEBAUD, G., AND LIU, X. 2009. Packet-Based hierarchical soft shadow mapping. *Comput. Graph. Forum* 28, 4, 1121–1130.
- ZHOU, K., HU, Y., LIN, S., GUO, B., AND SHUM, H.-Y. 2005. Precomputed shadow fields for dynamic scenes. In *Proceedings of the SIGGRAPH Conference*. 1196–1201.
- ZWICKER, M., YEA, S., VETRO, A., FORLINES, C., MATUSIK, W., AND PFISTER, H. 2007. Display pre-filtering for multi-view video compression. In *Proceedings of the 15th International Conference on Multimedia (MULTIMEDIA'07)*. 1046–1053.

Received October 2010; revised November 2010; accepted December 2010