DOI: 10.1111/cgf.12673
Eurographics Symposium on Rendering 2015
J. Lehtinen and D. Nowrouzezahrai
(Guest Editors)

Volume 34 (2015), Number 4

Stochastic Soft Shadow Mapping

G. Liktor and S. Spassov and G. Mückl and C. Dachsbacher

Karlsruhe Institute of Technology

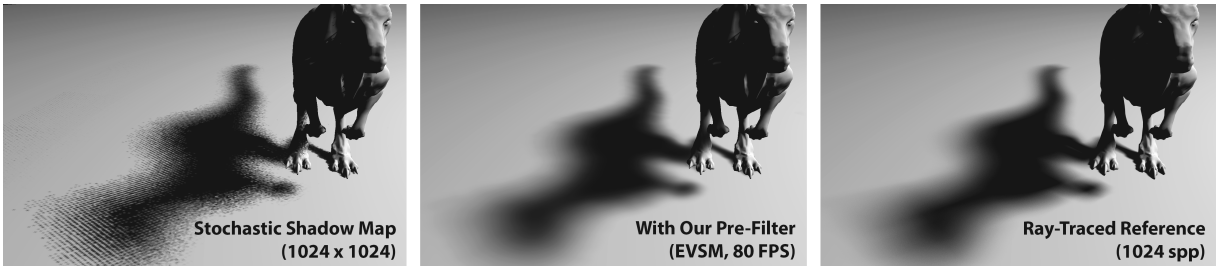


Figure 1: *Stochastic shadow maps sample the entire area of the light source and store the visibility information in a texture (**left**). Our novel algorithm reconstructs a smooth visibility function from the sparse data at real-time frame rates using multi-plane pre-filtering (**middle**). We closely approximate ray-traced soft shadows even for large area light sources (**right**).*

Abstract

In this paper, we extend the concept of pre-filtered shadow mapping to stochastic rasterization, enabling real-time rendering of soft shadows from planar area lights. Most existing soft shadow mapping methods lose important visibility information by relying on pinhole renderings from an area light source, providing plausible results only for small light sources. Since we sample the entire 4D shadow light field stochastically, we are able to closely approximate shadows of large area lights as well. In order to efficiently reconstruct smooth shadows from this sparse data, we exploit the analogy of soft shadow computation to rendering defocus blur, and introduce a multi-plane pre-filtering algorithm. We demonstrate how existing pre-filterable approximations of the visibility function, such as variance shadow mapping, can be extended to four dimensions within our framework.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

1. Introduction

Shadows provide important visual cues contributing to the perceived realism of synthetic images. While interactive rendering commonly approximates light sources as points, this is rarely accurate, since physical light sources always have a spatial extent. The exact computation of shadows involves solving an expensive visibility integral over the area of such light sources.

Monte Carlo methods, such as distributed ray tracing [CPC84], take stochastic *shadow ray* samples over the light source, but a very high sample count is needed for a smooth result. Recent advances of GPU-accelerated ray tracing [PBD*10] and reconstruction based on light field frequency analysis [EHDR11] can yield very accurate images at nearly interactive rates.

Fully ray-traced rendering still remains too expensive for real-time applications. Real-time methods are forced to simplify the problem, striving to obtain rather a *plausible*, but not necessarily *accurate* look of shadows. Many of these methods approximate soft shadows by an extension of shadow mapping [Wil78]. A plausible-looking soft shadow can be reconstructed by the filtering or reprojection of a hard shadow map, although such severe simplifications may introduce artifacts like false overshadowing or light leaks.

In this paper, we introduce *stochastic soft shadow maps*, which sample the entire 4D shadow light field instead of relying on a set of pinhole images. Our technique is designed for rendering pipelines with stochastic rasterization [AMMH07] or ray tracing, which we expect to become much more efficient at sampling higher-dimensional light fields on future hardware.

Our main contributions are:

- We **sample the entire 4D shadow light field** of a planar area light source using a **stochastic soft shadow map**.
- We **pre-filter the visibility function for fast soft shadowing at a fixed distance from the light source**. Our solution is a novel extension of existing pre-filtering methods, such as exponential or variance shadow mapping.
- We reconstruct the shadow light field at a set of **filter planes** and show that these can be used to **approximate the visibility at arbitrary locations using interpolation.**

2. Related Work

2.1. Soft Shadow Mapping Methods

Most real-time soft shadow rendering methods are based on shadow mapping [Wil78]. The accuracy of these methods is limited by the pinhole camera model used during the shadow map generation, sampling visibility from a single point on the light source. In other words, a pinhole shadow map represents only a 2D plane of the 4D shadow light field, missing important visibility information.

Percentage-closer soft shadows (PCSS) [Fer05] greatly simplify the soft shadow computation problem by assuming a single planar occluder, and that both the occluder and the receiver surfaces are parallel to the light source. The shadowing integral can be then described as a 2D convolution. In a practical scene, the average occluder depth is approximated within a search radius. Assuming a planar occluder at this depth, the penumbra size can be estimated, and the shadowing is solved by performing percentage-closer filtering (PCF) over the penumbra.

Pre-filtered convolution The footprint of the PCF filter of PCSS may become a limiting factor of the performance when the penumbra gets large, which motivated further approximations that pre-filter the visibility function prior to lighting. Convolution soft shadow maps (CSSM) [ADM*08] transform the visibility function into a Fourier basis. To reduce ringing artifacts, the method requires the storage of a high number of basis terms. Variance soft shadow maps (VSSM) [YDF*10] offer better performance by extending the theory of variance shadow maps (VSM) [DL06] to the evaluation of PCSS kernels. Exponential soft shadow maps (ESSM) [SFY13] adopt the exponential function for the occluder depth estimation and pre-filtering steps, based on exponential shadow mapping [AMS*08] [Sal08].

Backprojection methods attempt to approximate the visibility function more accurately by treating the shadow map texels as micro-occluders and projecting these at different locations on the area light source based on the shaded receiver surface [AHL*06] [GBP06]. Due to this coarse point-sampled approximation, light leaks and shadow overestimation may occur. Most of these artifacts are addressed by Guennebaud et al. [GBP07] and Schwartz et al. [SS07], but with reduced performance.

Multi-layer preconvolution A more accurate pre-filtering method is presented by Selgrad et al. [SDMS14], which captures all occluder surfaces in a multi-layer shadow map [XTP07]. Their key recognition is that for a given scene configuration with an area light and a planar occluder, an equivalent soft shadow can be obtained by using a point light source and a semitransparent occluder. The multi-layer shadow map is captured by building a per-texel fragment list of all occluders in a rasterization step. Each fragment in the list is an elementary occluder, which can be merged with neighboring fragments during pre-filtering. A final shadow map lookup then selects an appropriate filter width for each fragment layer (based on the receiver depth) and composites the pre-convolved opacity values.

The primary limitation of this technique is that it ignores partial occlusion among the different shadow map layers: each layer is assumed to be fully visible from the light source. The layers are composited heuristically, yielding plausible results for small light sources, but some inaccurate overshadowing or light leaking could still occur.

2.2. 4D Shadow Light Field Reconstruction

Our algorithm shares much similarity with Monte Carlo soft shadow reconstruction methods. A large body of recent research studied the frequency-space behavior of shadow light fields to efficiently sample and filter such phenomena, the complete overview of which falls beyond the scope of this paper. Our work has been inspired by the soft shadow reconstruction method of Egan et al. [EHDR11]. They developed a 4D sheared reconstruction filter for soft shadow rendering which could drastically reduce the per-pixel sample count. However, the evaluation of this filter is still too expensive for interactive rates. Mehta et al. [MWR12] showed that it is possible to reduce this problem to a 2D image-space filter which yields interactive results, but requires more samples per pixel for equivalent quality to sheared filtering.

We use stochastic rasterization for Monte Carlo sampling, first introduced by Akenine-Möller et al. [AMMH07]. They invented the method for the rendering of high-quality depth of field and motion blur. McGuire et al. [MESL10] presented an algorithm which implements stochastic rasterization within the modern GPU rendering pipeline. The application of stochastic rasterization to soft shadow rendering has been proposed by Nilsson et al. [NCJ*12], supporting motion blurred soft shadows as well. However, their implementation uses a fixed set of lens and time samples (virtual pinhole cameras), and can thus be regarded as an accelerated version of accumulation buffering.

In this paper, we similarly exploit the analogy of soft shadows to defocus. This concept has also been used by Lehtinen et al. [LAC*11] to model the shadow light field with a thin-lens camera and its focal plane. They aim for the actual generation of novel visibility samples to reduce noise yielding better quality than blurring, but the performance is below interactive rates.

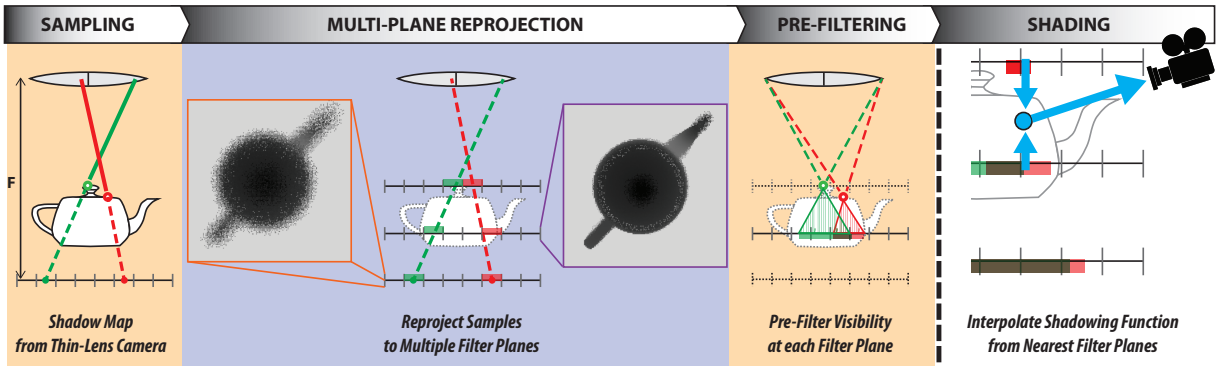


Figure 3: *We approximate soft shadows by pre-filtering the 4D shadow light field at multiple planes. Each filter plane is in the focus of an imaginary thin-lens camera (notice how the teapot appears sharper on the intersecting plane). While we visualize depth values here, in a real implementation we store samples in a pre-filterable basis, such as the first two moments (for VSMs).*

3. Overview

The basic motivation of our method is to exploit the analogy of the *thin lens camera model* to the problem of computing occlusion from a planar area light source.

Consider the special case of a planar receiver parallel to the light source. For any point on the receiver, the visibility integral can be solved numerically by shooting a set of shadow rays from that point to the light source (Fig. 2, left). Reversing the direction of the shadow rays, we can also perform this integration by rendering the occluders from an appropriately defined thin lens camera. With the lens placed on the light source and focused on the receiver, each point of the rendered image accumulates the shadow rays of a corresponding point on the receiver (Fig. 2, right). For any point on an occluder closer to the light source, its penumbra on the receiver matches its *blur circle* on the focal plane. Up to the magnification factor of the lens, this corresponds to its *circle of confusion* on the image plane.

During rendering from the thin lens camera, we store the distance to the nearest occluder along each ray. We call the resulting image a *stochastic soft shadow map*, since its generation involves stochastic sampling of a 4D shadow

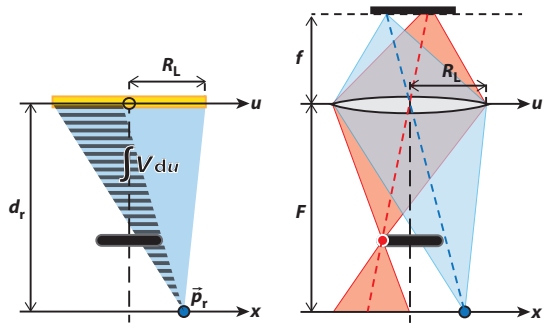


Figure 2: ***Left:*** *The soft shadow problem is solved by integrating visibility over the light source.* ***Right:*** *The same integral can be defined on the image plane of a thin lens camera.*

light field. In contrast to Selgrad et al.'s multi-layer filtering [SDMS14], where *all* occluders are captured and inter-occlusions between the layers ignored, our shadow map stores surfaces *only* if they are visible from the light source.

The accurate rendering of soft shadows as described above would require a large amount of samples to generate smooth results. *Our primary challenge is the creation of a reconstruction filter that can generate smooth results in real time from a sparse stochastic soft shadow map.*

3.1. Algorithm Preview

Fig. 3 provides a high-level overview of our technique:

1. We capture the 4D shadow light field in a stochastic soft shadow map using the thin-lens camera model (Sec. 3).
2. The shadow map samples are reprojected to multiple *filter planes* (Sec. 6) and converted to a pre-filterable basis.
3. For each filter plane, after computing the required per-sample filter radii by treating it as being in focus (Sec. 5.2), we can independently apply our visibility pre-filter (Sec. 5.3) to obtain a shadowing function at each texel. No additional information is stored.
4. During surface shading, we linearly interpolate an approximate shadowing function from the closest two filter planes based on the depth of the shaded surface and evaluate it to obtain the amount of shadowing.

In the following, we formulate soft shadow mapping as a light field reconstruction problem (Sec. 4), and show our novel method for fast approximate reconstruction using pre-filtered visibility (Sec. 5). Finally, we show how to extend the planar solution to arbitrary receivers (Sec. 6).

4. Light Field Reconstruction for Soft Shadow Mapping

We describe the depth samples in the shadow map using the *two-plane parameterization*, where each sample (ray) is defined by its origin on the plane of the light source (u) and an offset on a second plane at unit distance from the light

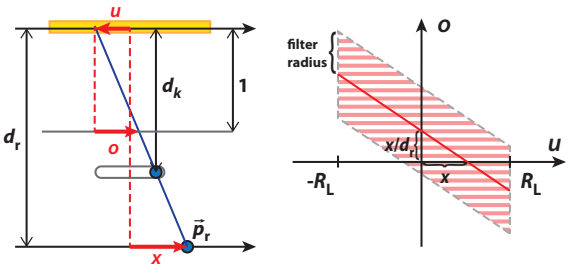


Figure 4: ***Left:*** *The two-plane parameterization of the shadow light field in flatland.* ***Right:*** *In the (u,o)-domain, the integration of the visibility corresponds to a line segment with a slope $-1/d_r$. Filtering the samples extends this line segment to a parallelogram.*

source (o) [DHS*05]. We formalize the shadowing problem in a 2D flatland scene (Fig. 4, left), by defining each receiver point $\vec{p}_r$ with its depth along the light's z-axis (d_r) and an offset relative to the projected center of the light source (x). The total irradiance at $\vec{p}_r = (d_r, x)$ can be then expressed as:

$$H(d_r,x) = \int_{-R_L}^{R_L} L(u)V(u,o,d_r)F(u,o,d_r)\,\mathrm{d}u \qquad (1)$$
$$\text{with } o = \frac{x-u}{d_r} \text{ from geometric similarities}$$

where R_L is the light source radius, $L(u)$ is the radiance emitted at point u on the light source (assuming a Lambertian source), $V(u,o,d_r)$ and $F(u,o,d_r)$ are, respectively, the visibility function and form factor between u and the receiver point. In the (u,o)-domain, the integral domain corresponds to a line segment with a slope $-1/d_r$ (Fig. 4, right).

Until Sec 6, we restrict our analysis to a single receiver plane parallel to the light source ($d_r = \text{const}$). We will use subscripts to denote normally free function arguments that are fixed in the current context. If there are no transparent surfaces, then the visibility function is binary:

$$V_{d_r}(u,x) = V(u,o,d_r) = \begin{cases} 1 & \text{if } d_k \geq d_r \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

where d_k is the depth sample at (u,x). For the purposes of this paper, we make the assumption that the light source has a constant intensity L. Further, we are only focusing on the *shadowing function* $S_{d_r}(x)$, ignoring the form factor, using the following approximation:

$$H_{d_r}(x) \approx L \cdot S_{d_r}(x)\bar{F}_{d_r}(x)$$
$$S_{d_r}(x) = \int_{-R_L}^{R_L} V_{d_r}(u,x)\,\mathrm{d}u \qquad (3)$$

Having a finite set of samples in the shadow map, it is unlikely that any of them would fall onto the corresponding (u,o)-line. Therefore, we need to approximate Eq. 3 by a two-dimensional filter $W_{d_r}(u,x)$:

$$S_{d_r}(x) \approx [V_{d_r} * W_{d_r}](u,x) \qquad (4)$$

4.1. Our Filtering Approach

The visibility function depends on nearest-occluder depths and those cannot be pre-filtered. Instead, we can filter results of depth comparisons (percentage-closer filtering). The thin-lens analogy provides an intuitive *per-sample* filter radius: if each sample is considered an *elementary occluder*, then its penumbra on the receiver corresponds to its *circle of confusion* on the shadow map (as long as the shadow map camera is focused on the receiver). By taking the approximation that each occluder in the shadow map is visible from the *entire* light source, we define our (initial) shadowing algorithm:

1. Assign to each elementary occluder (sample) a disc centered at its projection from the light source center on the shadow map, matching its circle of confusion (radius c_k).
2. Project the receiver being shaded to the shadow map and gather all samples with overlapping discs.
3. Evaluate visibility for these samples, and accumulate their contribution with a per-sample weight of $1/c_k^2$

A similar idea was described for the reconstruction of defocus blur by Shirley et al. [SAC*11]. They also propose an improved algorithm that adjusts the filter for partially occluded samples. We will revisit this problem in Sec. 5.2.

5. Pre-Filtered Visibility for Fast Shadowing

Finding the overlapping samples and performing the weighted PCF would be too expensive for large light sources. This leads us to the same challenge as fast PCSS-approximations: we need to transform the standard visibility function into a new basis which allows the use of a pre-filtering operator before the shadowing. In this paper, we demonstrate the concept using variance shadow mapping, but any other pre-filterable basis may be used instead.

5.1. Variance Shadow Mapping

Variance shadow mapping [DL06] approximates the result of the PCF kernel evaluation using a statistical approach. If we consider the shadow map sample depth a random variable D with mean μ and variance σ^2 over the filter region, then according to the Chebyshev inequality, there is an upper bound to the visibility at any given receiver depth $d_r > \mu$:

$$P(D \geq d_r) \leq \frac{\sigma^2}{\sigma^2 + (d_r - \mu)^2} \qquad (5)$$

The probability $P(D \geq d_r)$ represents the fraction of samples in the filter region, where the depth comparison yields $V = 1$. Since $\mu = E(D)$ and $\sigma^2 = E(D^2) - E(D)^2$ the visibility can be estimated by pre-filtering D and D^2.

A limitation of the technique stems from directly using the upper bound, even though it may not represent the true amount of visibility. Additionally, for $d_r \leq \mu$, no occlusion is assumed. In scenes with high depth complexity, the mean and variance are a weak estimation, causing light leaks when the filter overlaps occluders in different depth ranges.

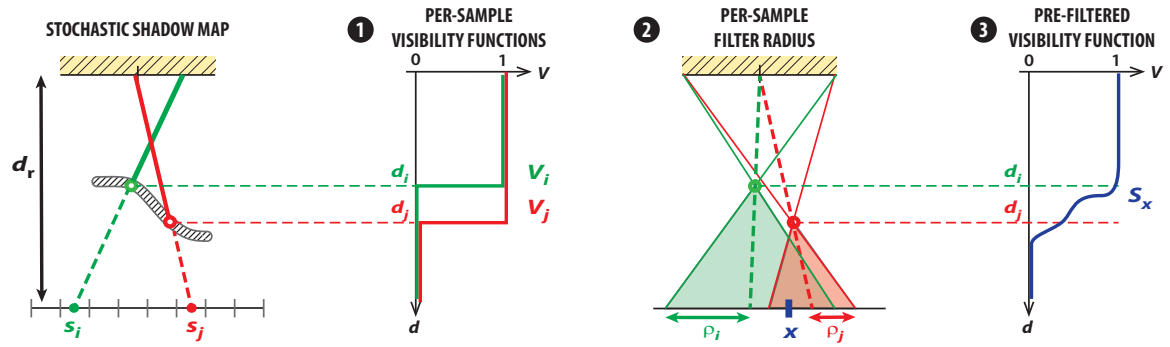


Figure 5: *Our visibility approximation concept:* ***(1)*** *The shadow map samples are transformed into pre-filterable visibility functions.* ***(2)*** *Samples are treated as elementary occluders and their penumbra on the receiver plane determines the filter sizes and weights.* ***(3)*** *Each point x on the receiver gathers overlapping samples to yield our approximate shadowing function.*

In a follow-up paper, Lauritzen and McCool [LM08] observe that the Chebyshev inequality also gives meaningful shadow estimates for the warped random variable $f(D)$ for monotonous f. They propose warping the sample depths using exponential functions to alleviate the light leaking problem. Exponential Variance Shadow Maps (EVSMs) apply VSM theory for warped depth maps e^{cD} and $-e^{-cD}$ to reduce variance and make the predictions more reliable.

5.2. Per-Sample Filter Radius

As proposed in Sec. 4.1, for each sample $s_k = [u_k, o_k, d_k]$ we can initially define the filter radius based on its circle of confusion in the shadow map image, as given by:

$$c_k = m(d_\mathrm{r}) R_\mathrm{L} \frac{|d_\mathrm{r} - d_k|}{d_k}, \tag{6}$$

where $m(d_\mathrm{r})$ represents the ratio between units (texels) in the shadow map and the virtual image on the object focal plane at depth d_r. Unfortunately, this intuitive radius does not yield pleasing results when used directly for filtering. If the shadow map is sparse (which is usually the case in a real-time scenario), there will be partially occluded surfaces that should cast continuous hard shadows but are not sampled with enough density. Fig. 6 shows a visualization of our samples (for the scene in Fig. 14), where it is clearly visible that the partial occlusion in the shadow map causes "holes" in the shadows. Increasing the sampling rate would solve the problem, but if that is not possible, *the only solution is to blur the in-focus samples to compensate against the density loss of the light field information*.

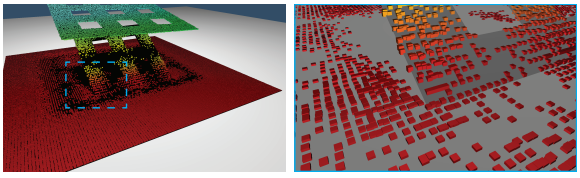


Figure 6: *World space visualization of stochastic shadow map samples: a partially occluded surface is not sampled with enough density to cast a sharp contact shadow.*

We have found it difficult to estimate the density of visibility information from the shadow map reliably. Instead, we propose to find the closest occluding sample s_l for each sample in the shadow map: a sample with the smallest circle of confusion (c_l) that overlaps the current sample, and $d_l < d_k$. If such a sample is found, we increase the radius of the partially occluded sample to $\rho_k = \max(c_k, c_l)$. The purpose of this step is to trade noise in partially occluded hard shadows for slight edge overblur, which is visually less distracting. We discuss the specifics of our implementation in Sec. 7.2.

5.3. Pre-Integration on a Single Plane

For a planar receiver parallel to the light source at a *fixed* distance d_r, our pre-filtered shadow computation can be summarized as follows (Fig. 5):

1. Convert each depth sample s_k to a pre-filterable 1D visibility function $V_k(d)$ stored as a simple *moment*-vector. The next steps involve only these vectors without function evaluations. Thus, we omit the d argument.
2. Project each sample from the center of the light source to the receiver plane to obtain x_k^p, and compute a per-sample filter radius ρ_k and weight $\alpha_k = 1/\rho_k^2$.
3. For each receiver point x on the d_r plane, the shadowing function can be computed as a weighted average of the visibility functions of all samples that overlap x:

$$S_x = \frac{\sum_k V_k \alpha_k}{\sum_k \alpha_k} \text{ for } k \text{ with } \|x - x_k^\mathrm{p}\| < \rho_k \tag{7}$$

In practice, we perform this pre-integration for the texels of a stochastic soft shadow map focused at d_r (discrete set of x). The result per-texel functions S_x can be evaluated at arbitrary d, but the dependency of ρ_k on d_r limits the validity of the approximation to $d \approx d_\mathrm{r}$. We address this in Sec. 6.

6. Multi-Plane Pre-Filtering

In the previous sections, we limited our reconstruction to a single planar receiver parallel to the light source. By using multiple such planes at different light source distances $d_{\mathrm{r},i}$,

we can obtain an approximation of soft shadows on arbitrary receivers. In the following, we will refer to these as *filter planes*. A filter plane i at depth $d_{r,i}$ represents $S_{d_{r,i}}(x)$ but can estimate the generalized $S(d_r, x)$ for arbitrary $d_r \approx d_{r,i}$.

Reprojection In a naïve implementation, the generation of each filter plane would involve the rendering of a stochastic shadow map from a thin lens camera focused on the corresponding plane. However, we can avoid multiple sampling of the occluders by *reprojecting* samples from an initial stochastic soft shadow map. By intersecting each additional filter plane with the rays defined by the samples of the initially rendered plane, we can make our algorithm independent from scene geometry. Thus, we use the same overall number of samples that a pinhole shadow map would.

6.1. Interpolated Shadowing Function

We can pre-filter visibility as described in Sec. 5 independently for each filter plane. In the general case, the receiver surfaces do not lie exactly on any of these filter planes, however, our shadowing functions can be evaluated at *any* distance from the light source, not only at the depth of the given filter plane. For this step, we can use the simple pinhole camera model: an arbitrary receiver point $\vec{p}_r = (d_r, x)$ can be projected from the center of the light source to the filter plane i to obtain $(d_{r,i}, x^p)$. The shadowing function stored at x^p can then be evaluated for the actual receiver depth d_r.

This means that pre-filtered hard shadows such as VSMs are special cases of our technique, where the pre-filtered pinhole visibility function is equivalent to one of our filter planes, sufficient to compute shadows everywhere in the scene. For larger area light sources, the results from this approximation lose accuracy with increasing distance between the filter plane and the receiver point. *This is our motivation for using multiple filter planes*. We generate a novel shadowing function by linearly interpolating the shadowing functions of the nearest two filter planes (see Fig. 7).

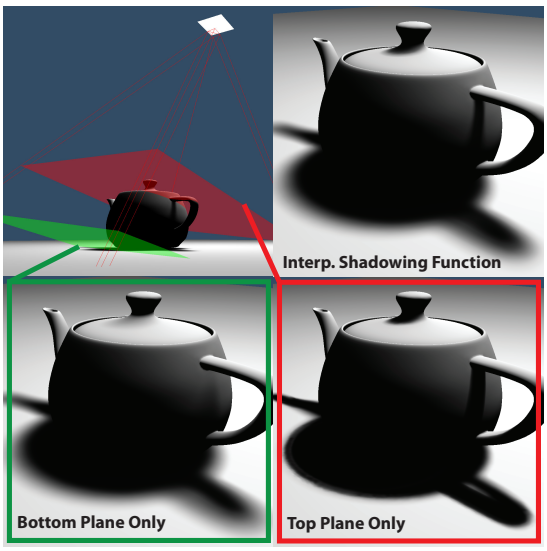


Figure 7: *Shadowing only with the bottom filter plane misses the shadows below the teapot handle. The top plane yields too sharp shadows on the ground. Evaluating an interpolated shadowing function produces plausible results.*

7. Implementation

7.1. Stochastic Sampling

Our stochastic soft shadow map is a plain 2D texture: each texel (x_k) stores the origin u_k and nearest-occluder depth d_k for the corresponding sample ray s_k. We implemented two algorithms for its generation: an emulation of stochastic rasterization [MESL10], and ray tracing from a thin lens camera using OptiX [PBD*10]. On current hardware, the latter solution is faster and we have included stochastic rasterization as a proof-of-concept only. Besides ray tracing we implemented all other steps of our algorithm in Direct3D 11.

Stratification Uniform sampling of the light source yields very noisy shadows, leading to a splotchy look in the filtered results that is particularly noticeable under animation. In our experiments, stratified sampling gave the best results: 4×4 neighboring texels sample different strata of the light area, reducing spatial and temporal artifacts. The resulting structured noise can be removed with a box-filter.

7.2. Multi-Resolution Pre-Filtering

To perform the filter radius adjustment that accounts for loss of sampling density efficiently (Sec. 5.2), each sample needs to find its closest occluding sample in the shadow map. While the exact solution would require iteration over all samples within a large search radius, we have found that using a sparse subset of samples yields a good tradeoff between performance and quality. We use Poisson-disc sampling of a fixed radius (32 texels) with 12 samples that results in a noisy approximation of c_l. We use a small Gaussian filter (2 texels radius) to remove the noise.

In the previous sections, we did not stress the difficulty arising from using a per-sample filter radius. A naïve implementation of the reconstruction filter could be based on *splatting*: every sample (texel) of the shadow map may be replaced with a disc according to its filter radius and contribute its visibility to the covered texels in the filtered shadow map.

This would be very inefficient as there is no real limitation on the radius. Furthermore, it would result in a very unbalanced number of scatter operations. Our initial step was to implement the reconstruction filter as a *gather* compute shader instead: for each texel of the filtered shadow map we iterate over a neighborhood within a radius R_G and accumulate the visibility functions from the overlapping samples.

We further increased the efficiency of our gather implementation by loading a texture tile into *groupshared memory* together with an R_G-wide ring of neighboring texels (*halo*

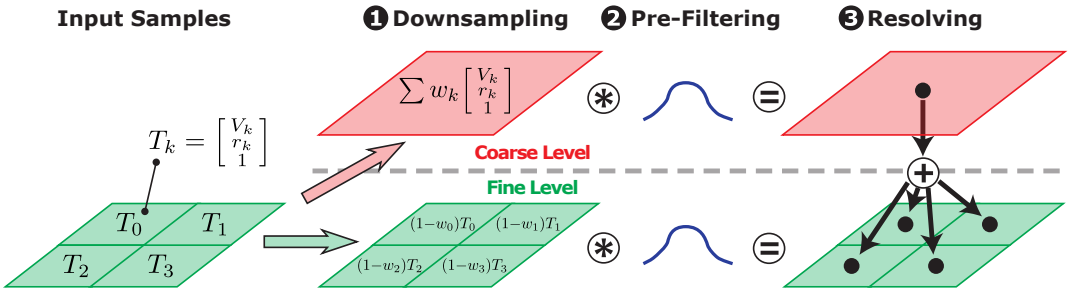


Figure 8: *Our multi-resolution scheme for pre-filtering visibility on a plane.* ***(1)*** *Samples with large filter radii are weighted and assigned to coarser levels.* ***(2)*** *The levels are pre-filtered independently.* ***(3)*** *Resolving yields our result on the finest level.*

region): this way all filter accesses can be performed without expensive global texture reads. Since the capacity of the groupshared memory is limited, R_G has to be small.

Our implementation pre-filters 16×16 samples per thread-group, with $R_G = 8$. Unfortunately, limiting R_G misses important samples in large penumbrae: samples outside the halo region are not considered, even if their filter radius possibly overlaps the tile, yielding discontinuities. Therefore, we have extended our implementation to a *multi-resolution filter*, which processes samples with large filter radii using coarse, downsampled levels (Fig. 8). Each level is processed with the same texel gather radius R_G, while the world-space texel size doubles with each coarser level (Fig. 9). The main steps of our multi-resolution filter are:

1. **Downsampling** For pairs of levels from finest to coarsest: for the coarser level we compute a *per-sample weight* w_k based on the filter radius ρ_k of that sample:

$$w_k = \begin{cases} 1 & \text{if } \rho_k > 0.5R_G \\ 0 & \text{if } \rho_k < 0.4R_G \\ 1 - \frac{0.5R_G - \rho_k}{0.1R_G} & \text{otherwise} \end{cases} \quad (8)$$

   The purpose of this weighting scheme is to eliminate samples from the finer level that might otherwise fall outside the halo region of a tile, despite overlapping that tile with their large filter radii, yielding the aforementioned discontinuities. Since a sample can be stored anywhere within its circle of confusion on the shadow map, we can only guarantee that it is found by the pre-filter if $\rho_k < 0.5R_G$. We also keep these samples on the finer level, but pre-multiplied with $(1 - w_k)$ to ensure the sum of contribution weights of any given sample always remains 1. Since the weights change continuously based on ρ_k, we avoid temporal artifacts.
2. **Pre-Filtering** We perform the algorithm from Sec. 5.3 for each level in parallel. We filter the weights as well, since they are needed for the final normalization step.
3. **Resolving** For pairs of levels from coarsest to finest: each finer sample gathers the pre-filtered (weighted) visibility from the coarser level, together with the pre-filtered weights (bilinear upsampling). On the finest level, the final shadowing function can be reconstructed by normalizing with the resolved weights.

7.3. Further Optimizations

We used the following ideas to optimize the filter performance within each individual resolution level.

Prior to filtering, we determine the minimum and maximum sample radii for each tile. With this information we can further limit the search radius for each thread group, or even completely skip tiles where all sample weights are zero. Furthermore, we have found that by using *interleaved sampling*, we can reduce the filtering cost to 25%: each thread skips every second sample in both dimensions. This results in a structured noise that can be easily removed in the resolving step, yielding nearly identical shadows to the brute-force filter (Fig. 10).

We make an additional approximation that greatly reduces texture bandwidth at the cost of slight overblurring of soft shadows: accurate projection of a depth sample from the light source center onto the filter plane (step 2. of Sec. 5.3) would require storing a per-texel 2D offset of the precise circle of confusion center. Instead, our pre-filter implementation treats each sample as if the center of its circle of confusion matches the center of the texel that it is stored in.

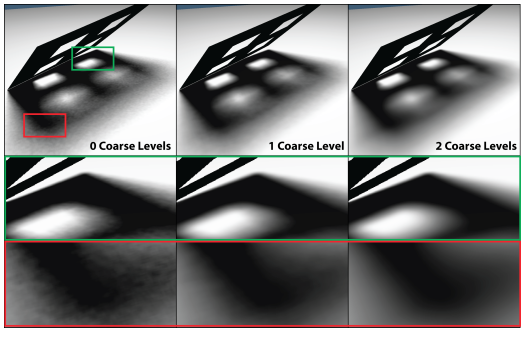


Figure 9: *Omitting samples causes noise. Coarse levels extend the effective world space search radius to avoid this.*

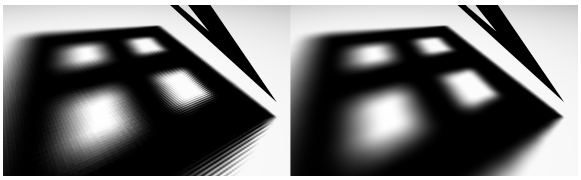


Figure 10: *The effect of stratified sampling and coarse pre-filtering is a structured noise (**left**) that can be easily removed by a 3×3 box filter (at each level) during the hierarchical upsampling step (**right**).*

7.4. Reprojection

This step synthesizes novel d_r-slices of the shadow light field, effectively re-focusing the thin lens camera, but *without* sampling the scene geometry again. As described in Sec. 6, we reproject samples in the shadow map to all filter planes. The primary implementation challenge is that multiple samples may project to the same texel while some texels will remain uncovered due to the irregularity of the light field.

We address this by first creating a *per-texel linked list* of reprojected samples, then *merge* them into a single value by averaging the visibility function and filtering radii at each texel. This approximation allows us to store only one value per texel for the filtering step. To preserve the total visibility contribution, we also set the initial weights to match the number of samples in the per-texel lists.

7.5. Selective Filtering

The complexity of our algorithm as described above would increase linearly with the number of filter planes. However, the planes may contain redundant samples that are not needed for the final shadow computation. We avoid handling such samples by introducing a pre-processing step: we first render the scene from the camera, and store the light-space depth of each visible surface in an off-screen target. This information can be used to mark *active tiles* at the finest level of each filter plane: only these regions need to be pre-filtered.

Care must be taken to propagate the marking of active tiles to coarser levels properly. Furthermore, halo regions of active tiles on all levels need to store valid information, but need not be pre-filtered. We mark these as *data tiles* across the multi-resolution hierarchy. All other texels in the filter planes can be ignored, saving bandwidth and computations.

8. Results and Discussion

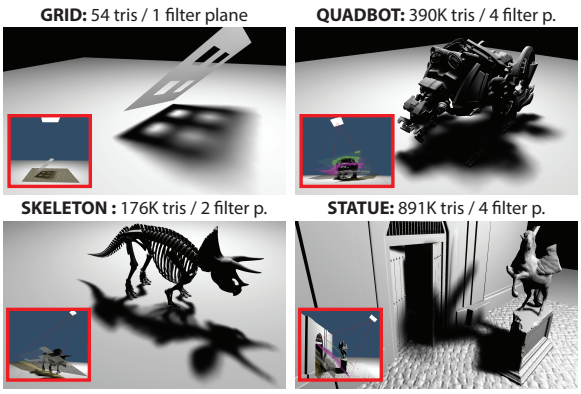
We performed our measurements on a workstation equipped with an Nvidia Geforce GTX 980 GPU, an Intel Core i7-860 CPU and 8GB RAM. Unless otherwise stated, all images were rendered at a resolution of 1920×1080 pixels, and all shadow maps were 1024×1024 texels large, with 2 coarse levels (Sec. 7.2). Our results use EVSMs to represent the visibility function (Sec. 5.1).

8.1. Performance

Fig. 11 provides measurements for different scenes. The `GRID` is an ideal setup for our algorithm where the focal plane matches the ground plane. Due to the very low number of polygons, the shadow map can be generated very fast using stochastic rasterization [MESL10]. For the rest of the scenes we used OptiX to generate the shadow map. The `QUADBOT` features a character with moderate depth complexity. The `SKELETON` challenges the reconstruction filter with fine geometry. Finally, the `STATUE` represents a higher, game-like depth complexity and a high polygon count.

We provide the timings of the `Sample` (shadow map generation), `Pre-Filter`, and `Lighting` steps. `Pre-Filter` consists of the substeps: `Reproject` (if there are at least 2 filter planes); `InitFinest` (per-sample radii on the finest levels); `GenCoarse` (downsampling); and `FilterLevels` (pre-filtering and resolving of all levels). Except for the `GRID` scene, we can observe that the time taken by `Sample` is of the same order of magnitude as `Pre-Filter`. We expect that on future architectures 4D stochastic sampling will become more feasible for real-time rendering.

The images and timings of Fig. 12 show how the algorithm performs for a range of shadow map resolutions. Our method fulfills the desired property of a shadow map, by yielding a smooth shadow even for a coarse, low-resolution shadow map. By increasing the resolution it approaches the details present in the ray-traced image.



	GRID	QUADBOT	SKELETON	STATUE
FPS	**238**	**51**	**88**	**40**
Sample	0.9	7.88	5.26	11.0
Pre-Filter	3.0	10.5	5.51	12.1
Reproject	–	2.82	0.69	3.03
InitFinest	0.86	3.01	1.68	3.32
GenCoarse	0.7	1.21	0.66	1.37
FilterLevels	1.44	3.5	2.47	4.39
Lighting	0.3	1.09	0.63	1.73

Figure 11: *The test scenes that were used for our measurements. The insets show the filter-plane configurations. All timings are in milliseconds (besides the FPS).*

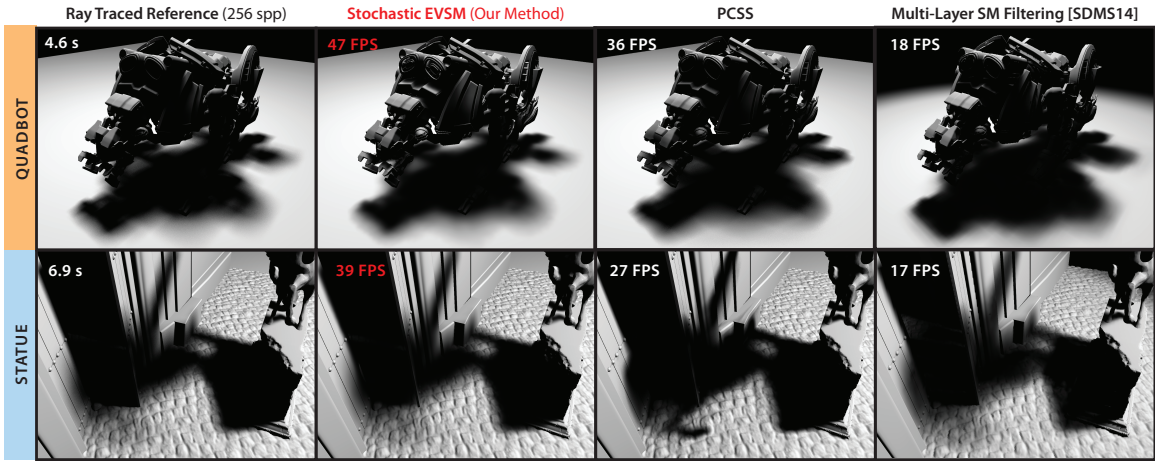


Figure 13: *Quality comparisons of our technique against ray tracing and other real-time soft shadow mapping methods.*

8.2. Quality

Fig. 13 shows quality comparisons for the QUADBOT and STATUE scenes. For the reference images we used ray tracing with OptiX. The soft shadows generated by our method closely approximate the reference for both scenes, while there is some loss of contact shadows. This is particularly noticeable below the door of the STATUE scene. For the QUADBOT we also miss some finer self-shadows on the robot model: we discuss this problem in Sec. 8.3.

Our implementation of PCSS uses 64 samples for the blocker search and 128 samples for the percentage-closer filtering. While self-shadows on the QUADBOT are plausible, the shadow on the ground appears harder than the reference. For the STATUE it misses the contact shadow of the door and renders the shadow of the statue too sharp on the ground. This is due to the planarity-assumption of PCSS.

We have also evaluated Multi-Layer Filtering [SDMS14], using the implementation generously provided by the authors. We have tested both the additive and multiplicative accumulation of fragment lists, but additive accumulation proved to be unsuitable for large light sources, because it consistently overestimated the occlusion. In this paper we therefore show images only with multiplicative accumulation. The main benefit of Multi-Layer Filtering is that it generates correct contact shadows. However, it overestimated the umbra region under the QUADBOT, and in the STATUE it yielded light leaks behind the statue.

	512 × 512	1024 × 1024	2048 × 2048
FPS	**200**	**84**	**36**
Sample	2.5	5.1	14
Pre-Filter	2.2	6.2	20
Lighting	0.3	0.63	1.7

Figure 12: *For this detailed view of the SKELETON scene, we show how the high-frequency details appear with increasing shadow map resolution. Our results are smooth (but low-frequency) even with low-resolution shadow maps.*

Light Field Modulation To our knowledge *ours is the first shadow mapping method that filters over the 4D shadow light field*. An important benefit is shown in Fig. 14: the shape of the shadow cast by the smaller grid changes on the ground, caused by the modulation of the shadow light field by the rotating top grid. Shadow mapping methods based on the pin-hole camera model are unable to emulate this.

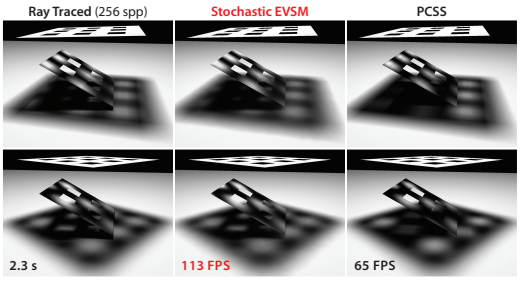


Figure 14: *Even though the light source is static, the shadow of the smaller grid seems to be moving on the ground when the larger grid is rotated.*

8.3. Limitations and Failure Cases

Our real-time pre-filter makes several assumptions which may cause visual artifacts. We directly inherit the limitations of our pre-filterable basis. In this paper we use (E)VSMs and thus share the problem of potential light leaking in contact shadows. Increasing the exponent alleviates the problem, but causes numerical problems in wide penumbra regions. In the presence of discontinuous receivers the moment-based pre-filtering method might not represent the depth distribution within the filter kernel faithfully (Fig. 15).

Other problems are coming from our simplification of the 4D shadow light field. Shadow map samples on the farther side of a depth discontinuity with a large filter radius (CoC) may erroneously spread their visibility to filter plane regions used for shading the nearer side. This happens in Fig. 16: visibility from the ground plane leaks to the pipe's interior. The discretization of the visibility function to planes has another consequence: if a receiver lies between two filter-planes, the contact shadows will get overblurred (Fig. 17).

The temporal behavior of our method significantly differs from pinhole-based shadow mapping. Our static results tend to be smooth. However, a dynamic change of the occluder depth of a sample may affect not only the footprint of a single texel, but a potentially large area on the filter plane.

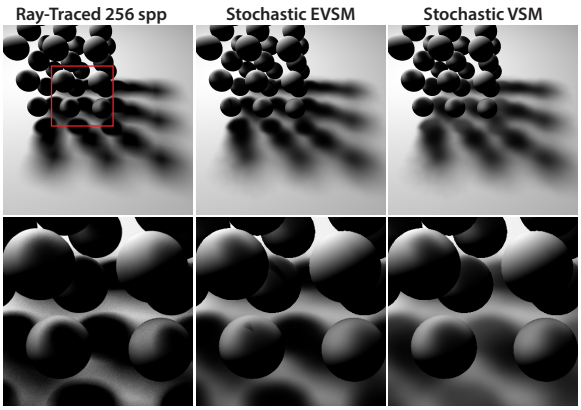


Figure 15: *The usage of VSM (right) causes light leaking in the penumbrae, which is largely alleviated when we use EVSM. The shadows on the spheres show leakage in both cases and numerical overflow with EVSM.*

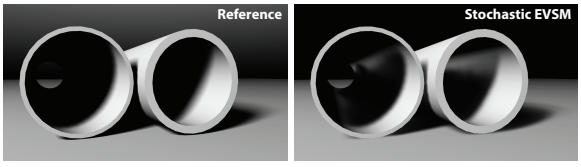


Figure 16: *Light leaks from defocused samples on the ground plane onto the interior walls of the pipes.*

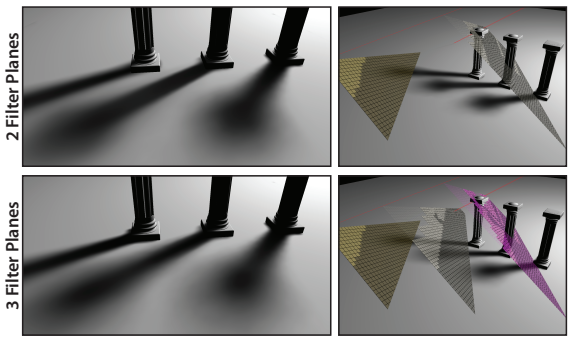


Figure 17: *With only 2 filter planes the contact shadow of the left column appears too soft. An additional filter plane improves the quality.*

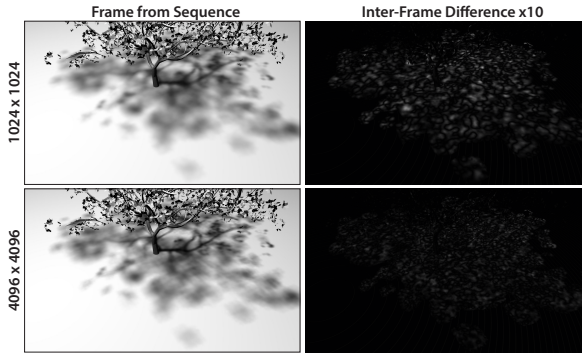


Figure 18: *A high-frequency tree exhibits a high amount of temporal artifacts (see also supplementary video). On the right, we show enhanced difference images between consecutive animation frames.* ***Top:*** *at the resolution of* 1024^2 *samples, the temporal noise has a low-frequency, splotchy look.* ***Bottom:*** *increasing the resolution also increases the frequency of the flicker and reduces its amount.*

Fig. 18 features a high-frequency occluder that easily gets undersampled while generating the stochastic soft shadow map. The difference images show not only the expected changes due to slowly moving light source, but also a high amount of noise. As can be expected, the temporal flickering reduces with increasing shadow map resolution. Note that this is a worst-case scenario for the method, and in our other scenes (e.g. `GRIDS` or `TEAPOT`) flickering is less apparent.

9. Conclusion and Future Work

We introduced a shadow mapping method that samples the entire shadow light field, and a novel multi-plane pre-filter for real-time reconstruction. We believe that on future graphics hardware, where the sampling of 4D visibility would become more practical in real-time, our approach may offer a viable alternative to standard shadow mapping.

We closely approximate ray-traced soft shadows in many cases, but some of the required simplifications can lead to inaccuracies near contact shadows. We analyzed the reasons of these artifacts in this paper. In the future we would like to develop a heuristic approach that falls back to percentage-closer filtering in the problematic regions. Provided that this involves a small percentage of pixels only, this would not significantly impact the performance.

Another open question is the placement of the filter planes. Our implementation uniformly partitions a user-defined depth range, but an optimal placement should be based on the visible receivers from the camera. Using active and data tiles greatly reduced our computational costs, however, we still allocate physical memory for each complete plane. An important challenge is to switch to a sparse tile-based allocation scheme to avoid redundant storage.

Finally, a very interesting challenge of future research is the extension of our approach to the temporal domain.

Acknowledgments

We would like to thank Kai Selgrad for providing the original implementation of the multi-layer pre-filtering method, and for his diligent support while setting up the comparisons against it. Our paper benefited from several assets generously provided by the on-line artist community: the QUADBOT was published by the Blender Foundation, the KILLEROO is courtesy of headus (metamorphosis) Pty Ltd. We also thank Joel Anderson (SKELETON), David Edwards (STATUE) and anonymous artists. We are grateful to our reviewers for their insightful feedback that helped us in improving the clarity of the paper.

References

[ADM*08] ANNEN T., DONG Z., MERTENS T., BEKAERT P., SEIDEL H.-P., KAUTZ J.: Real-time, all-frequency shadows in dynamic scenes. *ACM Transactions on Graphics (Proc. SIGGRAPH) 27*, 3 (2008), 34:1–34:8. 2

[AHL*06] ATTY L., HOLZSCHUCH N., LAPIERRE M., HASENFRATZ J.-M., HANSEN C., SILLION F.: Soft shadow maps: Efficient sampling of light source visibility. *Computer Graphics Forum 25*, 4 (2006), 725–âĂŞ741. 2

[AMMH07] AKENINE-MÖLLER T., MUNKBERG J., HASSELGREN J.: Stochastic rasterization using time-continuous triangles. In *Proc. Graphics Hardware* (2007), pp. 7–16. 1, 2

[AMS*08] ANNEN T., MERTENS T., SEIDEL H.-P., FLERACKERS E., KAUTZ J.: Exponential shadow maps. In *Proc. Graphics Interface* (2008), pp. 155–161. 2

[CPC84] COOK R. L., PORTER T., CARPENTER L.: Distributed ray tracing. In *Proc. ACM SIGGRAPH* (1984), pp. 137–145. 1

[DHS*05] DURAND F., HOLZSCHUCH N., SOLER C., CHAN E., SILLION F. X.: A frequency analysis of light transport. *ACM Transactions on Graphics (Proc. SIGGRAPH) 24*, 3 (2005), 1115–1126. 4

[DL06] DONNELLY W., LAURITZEN A.: Variance shadow maps. In *Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2006), pp. 161–165. 2, 4

[EHDR11] EGAN K., HECHT F., DURAND F., RAMAMOORTHI R.: Frequency analysis and sheared filtering for shadow light fields of complex occluders. *ACM Transactions on Graphics 30*, 2 (2011), 1–13. 1, 2

[Fer05] FERNANDO R.: Percentage-closer soft shadows. In *ACM SIGGRAPH 2005 Sketches* (2005). 2

[GBP06] GUENNEBAUD G., BARTHE L., PAULIN M.: Real-time soft shadow mapping by backprojection. In *Proc. Eurographics Symposium on Rendering* (2006), pp. 227–234. 2

[GBP07] GUENNEBAUD G., BARTHE L., PAULIN M.: High-quality adaptive soft shadow mapping. *Computer Graphics Forum 26*, 3 (2007), 525–533. 2

[LAC*11] LEHTINEN J., AILA T., CHEN J., LAINE S., DURAND F.: Temporal light field reconstruction for rendering distribution effects. *ACM Transactions on Graphics (Proc. SIGGRAPH) 30*, 4 (2011), 55:1–55:12. 2

[LM08] LAURITZEN A., MCCOOL M.: Layered variance shadow maps. In *Proc. Graphics Interface* (2008), pp. 139–146. 5

[MESL10] MCGUIRE M., ENDERTON E., SHIRLEY P., LUEBKE D.: Real-time stochastic rasterization on conventional gpu architectures. In *Proc. High Performance Graphics* (2010), pp. 173–182. 2, 6, 8

[MWR12] MEHTA S., WANG B., RAMAMOORTHI R.: Axis-aligned filtering for interactive sampled soft shadows. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 31*, 6 (2012), 163:1–163:10. 2

[NCJ*12] NILSSON J., CLARBERG P., JOHNSSON B., MUNKBERG J., HASSELGREN J., TOTH R., SALVI M., AKENINE-MÖLLER T.: Design and novel uses of higher-dimensional rasterization. In *Proc. High Performance Graphics* (2012), pp. 1–11. 2

[PBD*10] PARKER S. G., BIGLER J., DIETRICH A., FRIEDRICH H., HOBEROCK J., LUEBKE D., MCALLISTER D., MCGUIRE M., MORLEY K., ROBISON A., ET AL.: OptiX: a general purpose ray tracing engine. *ACM Transactions on Graphics (Proc. SIGGRAPH) 29*, 4 (2010), 66:1–66:13. 1, 6

[SAC*11] SHIRLEY P., AILA T., COHEN J., ENDERTON E., LAINE S., LUEBKE D., MCGUIRE M.: A local image reconstruction algorithm for stochastic rendering. In *Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2011), pp. 9–14. 4

[Sal08] SALVI M.: Rendering filtered shadows with exponential shadow maps. In *ShaderX6* (2008). 2

[SDMS14] SELGRAD K., DACHSBACHER C., MEYER Q., STAMMINGER M.: Filtering multi-layer shadow maps for accurate soft shadows. *Computer Graphics Forum 34*, 1 (2014), 205–215. 2, 3, 9

[SFY13] SHEN L., FENG J., YANG B.: Exponential soft shadow mapping. *Computer Graphics Forum 32*, 4 (2013), 107–116. 2

[SS07] SCHWARZ M., STAMMINGER M.: Bitmask soft shadows. *Computer Graphics Forum 26*, 3 (2007), 515–524. 2

[Wil78] WILLIAMS L.: Casting curved shadows on curved surfaces. In *Proc. ACM SIGGRAPH* (1978), pp. 270–274. 1, 2

[XTP07] XIE F., TABELLION E., PEARCE A.: Soft shadows by ray tracing multilayer transparent shadow maps. In *Proc. Eurographics Symposium on Rendering* (2007), pp. 265–276. 2

[YDF*10] YANG B., DONG Z., FENG J., SEIDEL H.-P., KAUTZ J.: Variance soft shadow mapping. *Computer Graphics Forum 29*, 7 (2010), 2127–2134. 2