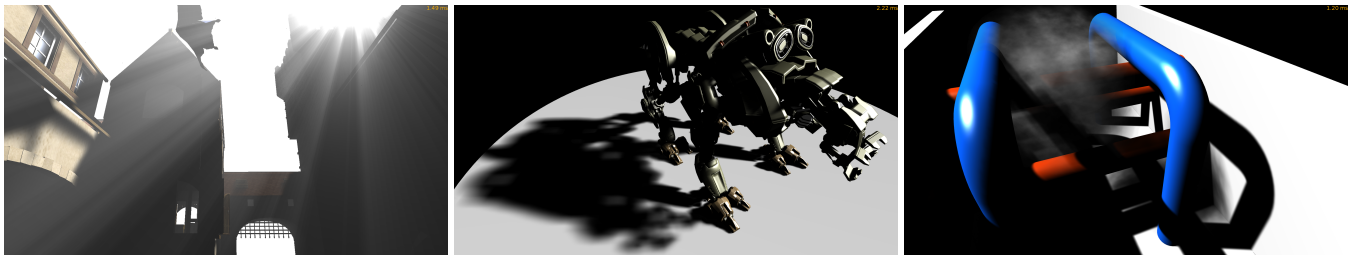
Beyond Hard Shadows: Moment Shadow Maps for Single Scattering, Soft Shadows and Translucent Occluders

Christoph Peters* Cedrick Münstermann* Nico Wetzstein* Reinhard Klein*
University of Bonn, Germany



Single scattering, 6 moments, 1.49 ms | Moment soft shadow mapping, 2.22 ms | Translucent occluders, 1.2 ms

Figure 1: *Inspired by previous techniques using other filterable shadow maps we transfer moment shadow mapping to three new applications. The combination with prefiltered single scattering produces realistic crepuscular rays, the combination with percentage-closer soft shadows leads to plausible soft shadows and extending moment shadow maps to translucent occluders is accomplished by rendering to a four moment shadow map with alpha blending. All three techniques provide an excellent run time. Timings are for full scene rendering at* $1920 \cdot 1080$.

Abstract

Building upon previous works, we transfer the recently proposed moment shadow mapping to three new applications. Like variance shadow maps and convolution shadow maps, moment shadow maps can be filtered directly. Classically, this is used to filter hard shadows but previous works explore other applications. Prefiltered single scattering uses convolution shadow maps to render single scattering in homogenous participating media, variance soft shadow mapping uses variance shadow maps for approximate soft shadows and Fourier opacity mapping uses convolution shadow maps for translucent occluders. We combine these three techniques with moment shadow mapping to arrive at better heuristics with less computational overhead.

Keywords: Moment shadow mapping, single scattering, soft shadows, translucent occluders, filterable shadow maps, participating media, real-time rendering

Concepts: •**Computing methodologies** → *Rasterization; Image-based rendering;*

1 Introduction

The plausible rendition of shadows is one of the oldest and most extensively studied problems in rendering. Considering their impact on the perception of geometric relations and overall realism, this is unsurprising. The problem also comes in manifold variants. In the most classic setting hard shadows of opaque occluders on surfaces are considered. However, scattering effects can make it necessary to integrate the results of shadowing computations along view rays, for area lights the shadows become soft and translucent occluders lead to partial shadow.

Most state-of-the-art techniques for rendering these effects rely on shadow mapping [Williams 1978]. Since shadow maps use an image-based representation of occluding geometry, aliasing is an issue but filtering of a common shadow map can only work with brute force [Reeves et al. 1987]. Alternatively, shadow maps can be made filterable by storing additional information in additional channels [Donnelly and Lauritzen 2006; Annen et al. 2007; Salvi 2008; Annen et al. 2008b; Lauritzen 2008; Peters and Klein 2015].

Being able to filter a shadow map directly means that all kinds of integrals can be precomputed. This has applications beyond the realm of hard shadows. Single scattering results can be precomputed [Klehm et al. 2014a], rendering of approximate soft shadows can be accelerated [Annen et al. 2008a; Yang et al. 2010; Shen et al. 2013] and translucent occluders can be rendered to shadow maps easily [Jansen and Bavoil 2010]. All these techniques avoid excessive sampling of shadow maps thus providing a reduced cost per shaded fragment.

Moment shadow maps are a more recently proposed kind of filterable shadow maps [Peters and Klein 2015]. Storing four scalars per texel (moments to be precise), they provide a high-quality heuristic with some unique traits. So far this has only been applied for filtering hard shadows. In the present paper we explore new applications. We combine moment shadow maps with prefiltered single scattering [Klehm et al. 2014a], use them for soft shadows in the spirit of variance soft shadow mapping [Yang et al. 2010] and take inspiration from Fourier opacity mapping [Jansen and Bavoil 2010] to demonstrate their use for shadows of translucent occluders.

Single scattering in homogenous participating media is discussed in Section 2. Exchanging the convolution shadow maps [Annen et al. 2007] used in the original prefiltered single scattering with moment shadow maps is straightforward and immediately yields convincing results. We further improve the quality by using six moments

*{peters,muenste,wetzstei,rk}@cs.uni-bonn.de

I3D '16, February 27–28, 2016, Redmond, WA
ISBN: 978-1-4503-4043-4/16/03
DOI: http://dx.doi.org/10.1145/2856400.2856402

Supplementary material can also be found on the project webpage: http://cg.cs.uni-bonn.de/en/publications/paper-details/peters-2016-msm-applications/

stored with 10 bits per moment. Besides we introduce a method to use filtered samples from a moment shadow map during resampling to diminish aliasing artifacts due to the rectification. The result is convincing single scattering with little aliasing and a predictable and short run time. Bandwidth requirements of the original technique are reduced heavily and ringing artifacts are eliminated.

Our soft shadow method, based on the framework of percentage-closer soft shadows [Fernando 2005], is introduced in Section 3. Like variance soft shadow mapping [Yang et al. 2010] it uses a summed-area table [Crow 1984] storing four moments in four 32-bit integers. Our heuristic blocker search robustly determines the average depth of occluding geometry from a single query to the summed-area table. The filter size is adapted accordingly and another query to the summed-area table provides the information for filtering. No further texture loads are needed.

Finally, we transfer the idea of Fourier opacity mapping [Jansen and Bavoil 2010] to moment shadow maps in Section 4. Rendering shadows for translucent occluders is as simple as rendering to the moment shadow map with alpha blending. Our evaluation shows that this technique increases the risk of light leaking but provides an excellent solution for scenes with few translucent occluders at almost no additional cost.

All proposed techniques work together naturally requiring nothing more than a single moment shadow map as input. The techniques are designed to be easy to implement and give robust results without complicated treatment of special cases.

1.1 Related Work

Before discussing the three major applications of this paper, we discuss related work relevant for all of them. Related work that is specific to the individual applications will be discussed in Sections 2.1, 3.1 and 4.1. For now we focus on shadow mapping basics and the various kinds of filterable shadow maps.

The major approaches for rendering shadows in real time are shadow mapping [Williams 1978] and shadow volumes [Crow 1977]. In recent years there has been a clear trend to shadow mapping because it scales better with scene complexity. A shadow map is an image rendered from the point of view of a light source storing depth values. Comparing the stored depth values to the depth of fragments allows for an efficient shadow test.

To avoid aliasing the sampling of the scene in the shadow map should closely resemble the sampling on screen and various techniques help to accomplish this [Martin and Tan 2004; Zhang et al. 2006; Lauritzen et al. 2011]. Still filtering is needed to get acceptable results. Since direct filtering of shadow maps would be meaningless, percentage-closer filtering [Reeves et al. 1987] instead samples the shadow map, turns each sample into a shadow intensity and filters these. This works well but requires many samples. Besides aliasing due to initial sampling of the shadow map remains.

Variance shadow mapping [Donnelly and Lauritzen 2006] takes a different approach by storing more information. A variance shadow map has two channels storing depth and depth squared. Filtered samples from such a shadow map provide two moments of the distribution of depth values within the filter region which can be used to evaluate a lower bound to the shadow intensity.

The idea of making shadow maps filterable has inspired further research. Convolution shadow maps [Annen et al. 2007] expand the shadow test function into a Fourier basis. This can yield an arbitrarily good approximation at the cost of high memory and bandwidth requirements. Exponential shadow maps [Salvi 2008; Annen et al. 2008b] approximate the shadow test function by an exponential that

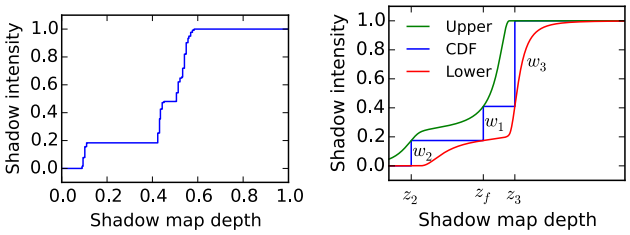


Figure 2: *Left: The shadow intensity as function of depth as reconstructed by percentage closer filtering. Right: The sharp lower and upper bounds produced by four moment shadow mapping among with one distribution realizing these bounds at z_f, z_2 and z_3.*

is scaled to be very steep. Exponential variance shadow maps [Lauritzen 2008] store two moments of two exponentially warped depth values and apply variance shadow mapping for both.

All these techniques store a compressed representation of the distribution of depth values within the filter region or equivalently of the filtered shadow test function. More generally, any monotonous function can be stored. Interpreted at this general level different methods become interchangeable, not only in the context of filtered hard shadows but also in other applications.

1.2 Moment Shadow Mapping

Since we are concerned with the transfer of moment shadow mapping [Peters and Klein 2015] to new applications, we now describe its inner workings to the extent necessary. Similar to variance shadow mapping, moment shadow mapping stores the depth, depth squared, depth cubed and depth to the power of four in a four channel texture. Filtered samples from this texture provide four moments of the distribution of depth values within the filter region. These four moments are used to compute a sharp lower bound to the shadow intensity which is then used as an approximation. Thus, the resulting shadow is never darker than it should be and given only the four moments a better bound cannot be obtained.

The algorithm for evaluating this bound at a depth $z_f \in [0,1]$ internally constructs a distribution of depth values that realizes the bound as shown in Figure 2. In the case of four moments it takes the form $\sum_{i=1}^{3} w_i \cdot \delta_{z_i}$ where δ_{z_i} denotes a Dirac-δ distribution at z_i, $z_1 := z_f$, $z_2, z_3 \in \mathbb{R}$ and $w_1, w_2, w_3 > 0$ with $\sum_{i=1}^{3} w_i = 1$. The corresponding cumulative distribution function is simply a piecewise constant function with three steps at z_1, z_2, z_3. By construction it matches the moments $b_1, b_2, b_3, b_4 \in \mathbb{R}$ in the sense that

$$\forall j \in \{1, \dots, 4\} : \; b_j = \sum_{i=1}^{3} w_i \cdot z_i^j. \tag{1}$$

Among all such distributions its cumulative distribution function is known to be minimal just before each of the steps. Since the true distribution of depth values also matches the moments, the desired lower bound for the shadow intensity is given by $\sum_{i=2,\, z_i < z_f}^{3} w_i$.

An optimized affine transform helps to store the moments in four 16-bit integers albeit quantization errors require an increased biasing that strengthens light leaking. In absence of precision errors moment shadow mapping can obtain a perfect reconstruction if the filter region contains only two surfaces at constant depth. In the context of filtered hard shadows this approximates most relevant cases well. The authors also formulate moment shadow mapping for any even number of moments (variance shadow maps arise as special case) but do not apply this to hard shadows.

2 Single Scattering

In rendering it is common to assume that all relevant light interactions happen at surfaces. This neglects volumetric scattering occurring in participating media such as smoke, dusty air, moist air and so forth. Light can be reflected towards the camera in midair. When this effect is coupled with proper computation of shadows it provides great artistic value. Holes in geometry lead to shafts of light known as crepuscular rays. These are perceived as aesthetic and provide a great tool to direct attention of the viewer.

Unsurprisingly the creative industry likes to use this effect but it still comes at a high cost. Scattering can occur anywhere within a volume. At the same time the visibility of the light source can change arbitrarily along a view ray. This visibility term makes the integration expensive. One way to evaluate it is based on classic shadow mapping coupled with ray marching but this leads to many shadow map reads with poor cache coherence.

Prefiltered single scattering [Klehm et al. 2014a] accelerates this procedure by precomputing the relevant integrals into a convolution shadow map. In the following we improve upon this idea by using moment shadow maps with four (Section 2.4) or six moments (Section 2.5). Besides we demonstrate how to apply filtering during the necessary resampling step (Section 2.3).

2.1 Related Work

Just like surfaces participating media exhibits global illumination effects known as multiple scattering. In real-time rendering these are commonly ignored to accommodate tight frame-time budgets. What remains is single scattering; light coming directly from a light source is scattered into the view ray. We focus on this effect.

Early works for rendering single scattering rely on shadow volumes [Max 1986] but more recent works employ shadow mapping. Dobashi et al. [2002] render translucent planes with shadow mapping. This is equivalent to ray marching and on modern hardware more efficient implementations use programmable shaders with interleaved sampling [Tóth and Umenhoffer 2009]. For acceleration it has been proposed to use shadow volumes to identify regions containing shadows and to render single scattering at a lower resolution followed by bilateral upscaling [Wyman and Ramsey 2008].

Engelhardt et al. [2010] apply more drastic subsampling in screen space placing samples intelligently along few epipolar lines through the light source. Voxelized shadow volumes [Wyman 2011] provide a more cache-friendly way to store shadow information for scattering. The results of 128 shadow tests along a view ray can be queried at once. With proper parallelization this can be extended to area lights [Wyman and Dai 2013].

Baran et al. [2010] exploit the simplicity of the shadow test function to perform ray marching at amortized logarithmic time. Building upon this work Chen et al. [2011] propose use of a 1D min-max-mipmap to traverse ray segments that are fully lit or fully shadowed in a single step. Both techniques use epipolar coordinates and the latter technique is easily mapped to graphics hardware.

Prefiltered single scattering [Klehm et al. 2014a; Klehm et al. 2014b] introduces the concept of filterable shadow maps to single scattering. The authors generate a convolution shadow map in a rectified coordinate system where rows correspond to epipolar planes containing the camera position while being parallel to the directional light. Computing weighted prefix sums along rows effectively precomputes the result of single scattering for the whole epipolar slice at once. While this method works fast independent of scene complexity, the Fourier series used in convolution shadow maps introduces ringing and memory requirements are high.

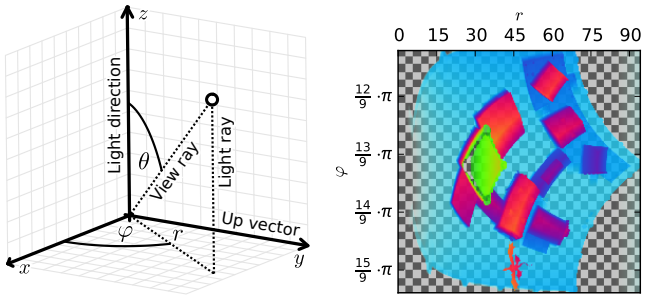


Figure 3: *Left: The definition of the rectified coordinates r, θ, φ. Right: A resampled moment shadow map using rectified coordinates with appropriate bounds. The checkerboard serves to visualize the alpha channel.*

2.2 Prefiltered Single Scattering

Our work is an extension of prefiltered single scattering and in the following we provide enough details on this technique to make our discussion self-contained. Still we recommend that a reader interested in implementing our method additionally refers to the more detailed discussion by Klehm et al. [2014b].

Besides the restriction to single scattering prefiltered single scattering makes a few additional simplifying assumptions that we inherit. The participating media has to be homogenous, i.e. its physical properties must be the same everywhere. Namely, these properties are the extinction coefficient σ_t defining transmittance, the phase function $f(\omega_l, \omega_p)$ which is the volumetric pendant of a BRDF and the scattering albedo ρ. We also assume homogenous lighting from a single directional light with direction ω_l and maximal irradiance E_l. Multiple directional lights can be handled by superposition.

Now consider a surface element at distance s from the camera with outgoing radiance L_s towards the camera. The camera lies in direction ω_p at position p. Let $V(q)$ be a visibility function for the light source mapping a position in 3D-space to one if it is lit and to zero if it is shadowed. Then the radiance received at the camera is

$$\exp(-\sigma_t \cdot s) \cdot L_s + f(\omega_l, \omega_p) \cdot E_l \cdot \int_0^s \exp(-\sigma_t \cdot t) \cdot V(p - t \cdot \omega_p) \mathrm{d}t.$$

The first summand is simply the radiance from the surface that remains after absorption and out-scattering. The second summand models single scattering. At each lit segment along the view ray a differential radiance of $f(\omega_l, \omega_p) \cdot E_l$ is added but only the part $\exp(-\sigma_t \cdot t)$ of it is transmitted to the camera.

The cost of computing single scattering lies in the integration which includes the visibility term. When we view it as one-dimensional function along a light ray the visibility function is a simple Heaviside step function. It is one up to the first occluder and then it is zero. The filterable shadow maps described in Section 1.1 provide means to store such functions in a way that enables the application of filters. We use this to turn integration of single scattering into an integration over rows of a shadow map.

For this to work the parametrization of the shadow map has to meet two requirements. View rays have to map to rows in the shadow map and the depth of view rays within the shadow map has to be constant. In most cases such a parametrization can be constructed as simple perspective transform [Klehm et al. 2014a]. We have implemented this linear rectification but for reasons given in Section 2.3 we opted for the other proposed solution; a non-linear rectification transform applied by means of resampling [Baran et al. 2010].

To convert coordinates from world space to rectified coordinates we first convert to light view space and move the origin of the coordinate system into the camera location. In this space the light direction corresponds to the z-axis and the other axes are chosen arbitrarily but orthogonal as shown in Figure 3. Then u-coordinates in the shadow map correspond to the distance to the origin after projecting to the x-y-plane, r. This coincides with the distance of the light ray to the camera. For the other two coordinates we convert to spherical coordinates. The v-coordinate in the shadow map corresponds to the azimuth $\varphi \in [0, 2 \cdot \pi]$. Depth stored in the shadow map corresponds to the flipped inclination $\pi - \theta \in [0, \pi]$ which is the angle to the negative light direction.

Since φ and θ are independent of the distance to the camera, view rays map to shadow map rows and have constant depth as required. At the same time the parametrization is valid for a shadow map because each light ray has constant r and φ and thus maps to a single texel. In terms of epipolar geometry φ indexes epipolar slices containing the light direction and going through the camera. Single scattering computations for different epipolar slices are independent. Bounds for r, φ and θ are computed such that the entire view frustum is covered. Along the dimension of θ we add a guard band to avoid unnecessary leaking artifacts.

We generate a filterable shadow map $b(u, v)$ in this coordinate system and index it with integer texel indices u, v. Each texel stores a representation of the visibility function (e.g. moments) and filtering a row corresponds to filtering along all view rays in the corresponding epipolar slice. To precompute the relevant integrals we need to know the world-space distance $\Delta(v)$ between successive view ray samples per slice. Since this quantity depends upon the inclination θ, a heuristic is required. Sophisticated heuristics have been proposed [Klehm et al. 2014b] but we simply compute the distance for the arithmetic mean of the minimal and maximal values of θ. Then transmittance weighted prefix sums are computed as

$$\bar{b}(u, v) := \frac{\sum_{i=1}^{u} w_{i,v} \cdot b(i, v)}{\sum_{i=1}^{u} w_{i,v}},$$

$$w_{u,v} := \left[-\frac{1}{\sigma_t} \cdot \exp(-\sigma_t \cdot t) \right]_{(u-\frac{1}{2})\cdot\Delta(v)}^{(u+\frac{1}{2})\cdot\Delta(v)}.$$

To compute the scattering for a view ray ending at some location $q \in \mathbb{R}^3$ we sample the prefiltered shadow map $\bar{b}$ at the appropriate location, reconstruct a shadow intensity between zero and one as one would do for filtered hard shadows, subtract it from one to get visibility and then multiply by the maximal possible scattering

$$f(\omega_l, \omega_p) \cdot E_l \cdot \left[-\frac{1}{\sigma_t} \cdot \exp(-\sigma_t \cdot t) \right]_0^{\|q-p\|_2}.$$

This procedure only requires a single lookup in the prefiltered shadow map per pixel on screen. Thus, the run time of the technique is independent of the scene complexity.

2.3 Rectification of Moment Shadow Maps

The linear rectification proposed by Klehm et al. tends to allocate major parts of the shadow map for geometry near the camera while farther geometry is compressed [Klehm et al. 2014a]. This can be alleviated by moving away the near clipping plane or by using split shadow maps but neither solution is quite satisfactory. Besides non-linear rectification still has to be implemented for the case where an epipole is near the field of view.

On the other hand, the non-linear rectification described above requires resampling of shadow maps to be implemented efficiently.

Algorithm 1 Input: Moments $b_1, b_2, b_3 \in \mathbb{R}$.
Output: Depth values $z_2, z_3 \in \mathbb{R}$ and weights $w_2, w_3 > 0$ such that $b_j = w_2 \cdot z_2^j + w_3 \cdot z_3^j$ for all $j \in \{1, 2, 3\}$.

- $c_2 := b_2 - b_1^2$, $c_1 := b_1 \cdot b_2 - b_3$, $c_0 := -b_1 \cdot c_1 - b_2 \cdot c_2$.
- Solve $c_2 \cdot z^2 + c_1 \cdot z + c_0 = 0$ to get solutions $z_2, z_3 \in \mathbb{R}$.
- $w_3 := \frac{b_1 - z_2}{z_3 - z_2}$, $w_2 := 1 - w_3$.
- Return z_2, z_3 and w_2, w_3.

Since common shadow maps cannot be filtered during resampling, this introduces considerable aliasing artifacts. Straight silhouettes exhibit staircase artifacts that lead to visible stripes in the crepuscular rays. These stripes are not stable with regard to movements or rotations of the camera which makes them quite noticeable.

Ideally the shadow map could be filtered during resampling. We have accomplished this using moment shadow maps. Instead of taking a sample without filtering from a common shadow map we take a filtered sample from a moment shadow map. We then turn the obtained moments back into a distribution of depth values because we need to distort depth in a non-linear fashion. It is adequate to expect simple distributions because we are working with small filter kernels. In most relevant cases the sample will not cover more than two different surfaces.

In Section 1.2 we explained how to reconstruct a distribution with three depth values z_1, z_2, z_3 from four moments where $z_1 = z_f$ is prescribed. This leaves us with the question how to choose z_1. To avoid an arbitrary choice and to obtain a more efficient solution we let z_1 go to infinity. As this happens w_1 approaches zero and we can discard the depth value z_1. The remaining distribution $w_2 \cdot \delta_{z_2} + w_3 \cdot \delta_{z_3}$ is still compatible with the first three moments. Only the fourth moment no longer matches. Under the assumption of two different surfaces at constant depth we can be certain that the reconstruction is adequate. Otherwise it provides a reasonable approximation that is certainly better than a single shadow map sample.

Algorithm 1 computes the distribution described above. Its derivation is non-trivial and we provide it in the supplementary material among with an HLSL implementation. Note that the algorithm fails if the variance $\sigma^2 = c_2$ is not positive. This case is adequately handled by returning $z_2 = z_3 = b_1$, $w_2 = 1$ and $w_3 = 0$.

Once we have obtained the distribution we convert z_2 and z_3 to inclinations θ_2, θ_3 as described in Section 2.2 and normalize to the interval $[0, 1]$ clamping out-of-range values. Then we convert both values to moments and linearly combine the two resulting vectors of moments using the weights w_2 and w_3. This vector is what we store in $b(u, v)$. At this point we can also generate more than four moments. Note that this whole procedure does not require an additional pass. The vectors can be generated on the fly during generation of the prefix sums.

Using this scheme is entirely optional. It does make the implementation a bit more involved and increases bandwidth requirements but also greatly enhances the visual stability of the results.

2.4 Prefiltered Single Scattering with Four Moments

Exchanging convolution shadow maps for moment shadow maps in prefiltered single scattering is straightforward. Instead of storing values of the Fourier basis in the shadow map, we store four moments. As usual it is best to apply an optimized affine transform before storage to be more robust with regard to quantization errors [Peters and Klein 2015]. When it comes to the computation of the shadow intensity during evaluation of single scattering we can proceed as for hard shadows (see Section 1.2).

However, some assumptions made for hard shadows are inadequate for single scattering. For hard shadows we always underestimate the shadow intensity to avoid surface acne. For single scattering this is not necessary. We care more about small approximation error. Our solution is to take a weighted combination of a sharp lower bound and a sharp upper bound. This only incurs a small additional cost. If we have represented the moments as in Equation (1), the sharp upper bound is simply given by $w_1 + \sum_{i=2,\, z_i<z_f}^{3} w_i$, i.e. the lower bound and the upper bound only differ by w_1 (Figure 2).

To get the desired weighted combination we fix a weighting factor $\beta \in [0, 1]$ and compute $\beta \cdot w_1 + \sum_{i=2,\, z_i<z_f}^{3} w_i$. For $\beta = 0$ this yields the lower bound, i.e. single scattering is too bright. For $\beta = 1$ it yields the upper bound and single scattering is too dark. Intermediate values blend linearly. This is an intuitive parameter that may be exposed to artists to control the effect of approximation errors. For example large values seem preferable for indoor environments to avoid light leaking through walls. In our experiments we always use $\beta = 0.5$ to achieve a minimal worst-case error.

2.5 Prefiltered Single Scattering with Six Moments

Moment shadow mapping with four moments is a particularly good match for hard shadows because as long as filter kernels only contain two surfaces, the reconstruction can be nearly perfect. For single scattering our kernels correspond to epipolar planes stretching through the whole scene. Thus the case of two surfaces within the kernel is rarely present. Depth distributions generally exhibit a greater complexity and it is unrealistic to expect an adequate reconstruction from four moments in all cases.

Therefore we propose to use six moments. While the original algorithm is formulated for any even number of moments [Peters and Klein 2015], this has never been tried in a graphics context and the implementation is challenging due to numerical issues. Here we briefly outline our solutions and provide all more technical details and HLSL implementations in the supplementary material.

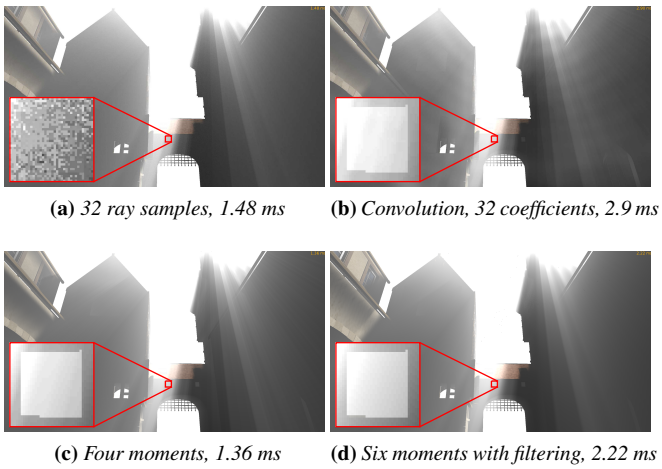
To avoid loss of information during quantization we optimize an affine transform as described by Peters and Klein [2015]. For six moments the found transform increases entropy per texel by 30.5 bits. This is almost certainly not the global maximum but considerable computational resources were invested to get close. The algorithm requires solution of a symmetric and positive semi-definite 4×4 system of linear equations. As for four moments a Cholesky decomposition solves this robustly and efficiently. To solve the resulting cubic equation we use a closed-form solution followed by one iteration of Newton's method to counteract instabilities.

This gives us the depth values z_2, z_3, z_4 for the distribution. For stability reasons we avoid computing the corresponding weights w_2, w_3, w_4 explicitly. Instead we immediately compute the relevant sum of weights by constructing a Newton polynomial. This is stable and works in quadratic time.

The resulting algorithm still requires considerable biasing ($\alpha = 4 \cdot 10^{-3}$). While this would be a serious problem for hard shadows, it does not prevent us from using the algorithm for single scattering.

2.6 Results and Discussion

Our implementation uses Direct3D 11 and applies all discussed single scattering techniques during a deferred rendering pass with the depth buffer as input. Our method should be fast enough to be applied during forward rendering to avoid problems with transparencies and multisampling but we have not tested this. Throughout the paper all our timings refer to an nVidia GeForce GTX 980 graphics card and unless specified otherwise the output resolution is $1920 \cdot 1080$ and all shadow maps have resolution 1024^2.



(a) *32 ray samples, 1.48 ms* **(b)** *Convolution, 32 coefficients, 2.9 ms*

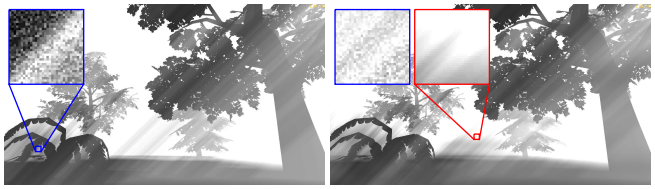
(c) *Four moments, 1.36 ms* **(d)** *Six moments with filtering, 2.22 ms*

Figure 4: *A comparison of various single scattering techniques. Frame times are for full scene rendering. Note the ringing produced by convolution shadow maps and the window that appears brighter than the surrounding wall (magnified). Contrasts in magnified insets have been stretched by a factor of four.*

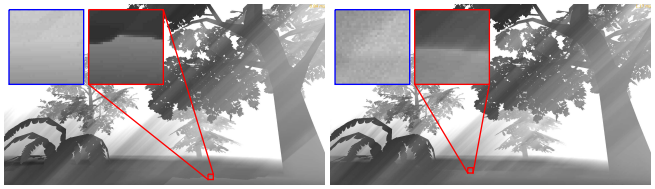
For comparison we have implemented ray marching with equidistant, jittered samples and prefiltered single scattering using convolution shadow maps with 32 real coefficients. We generate prefix sums in a compute shader using one thread per row as recommended by Klehm et al. [Klehm et al. 2014b]. For common shadow maps we use 16-bit textures and for convolution shadow maps we use 8 bits per channel. For moment shadow mapping we found that the artifacts introduced by stronger biasing are much less objectionable compared to hard shadows. Since biasing helps to compensate quantization artifacts we evaluate using 16 bits per channel for four moments and 10 bits per channel for six moments (splitting the six moments onto two textures with three channels). Using 16 bits per channel for six moments did not offer a significant advantage.

Figure 4 shows a comparison of these techniques. The scene mostly consists of occluders with a large area thus providing a simple case for ray marching. Therefore, 32 ray samples already produce acceptable results with moderate noise (Fig. 4a). Techniques based on prefiltered single scattering do not produce noise but more systematic errors. Prefiltered single scattering with convolution shadow maps (Fig. 4b) produces strong ringing due to the truncation of the Fourier series. An artifact that is shared by all techniques with prefiltering is magnified (Figs. 4b, 4c, 4d). A window allows a view onto the interior of the building, which is entirely shadowed. Thus, there should be no additional single scattering but approximation errors let the window appear brighter anyway. If the shadow map is resampled without filtering, erroneous fine structures appear (Figs. 4b, 4c) which are reduced heavily by filtering (Fig. 4d). Note that Figures 4a, 4b and 4c exhibit some surface acne from percentage-closer filtering which is not present in Figure 4d because the available moment shadow map is used for shadowing.

Figure 5 shows a more challenging test case where all techniques exhibit some characteristic artifacts. In spite of the increased number of samples ray marching still produces strong noise (Fig. 5a). Ringing in prefiltered single scattering with convolution shadow maps makes parts of crepuscular rays disappear entirely (Fig. 5b). A characteristic artifact of prefiltered single scattering with mo-

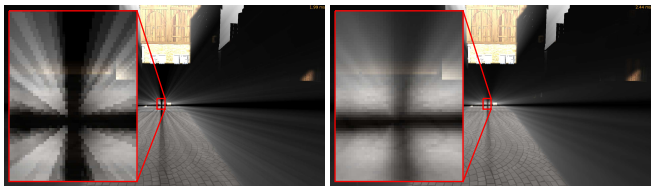


(a) *128 ray samples, 2.44 ms* **(b)** *Convolution, 32 coefficients, 2.82 ms*



(c) *Four moments, 0.84 ms* **(d)** *Six moments, 1.17 ms*

Figure 5: *A challenging scenario where all shown techniques exhibit typical artifacts. The scene is not shaded but frame times still refer to full scene rendering. The magnifiers to the left show ray marching results. Magnifiers stretch contrasts by a factor of four.*



(a) *Six moments, 1.99 ms* **(b)** *Six moments with filtering, 2.44 ms*

Figure 6: *An example of the positive effect of filtering during resampling. Aliasing is reduced heavily.*

ment shadow maps are excessively sharp boundaries of shadow volumes (Fig. 5c). This occurs because the approximation quality can change quickly as distributions of depth values become more complex. Using six moments reduces this artifact heavily (Fig. 5d).

Figure 6 demonstrates the positive effect of filtering in an extreme case. Without filtering during resampling crepuscular rays exhibit fine structures which change rapidly as the camera moves or rotates (Fig. 6a). Especially for slowly moving cameras this aliasing can be a very distracting artifact (see supplementary video). Applying trilinear filtering to a moment shadow map during resampling (Section 2.3) makes the shadows lose some detail and comes at a cost but aliasing is reduced to a point where it is unproblematic.

Figure 7 demonstrates a case where approximation errors can be problematic. As a dragon enters the view, the single scattering is not only attenuated below but also above it. Especially for moving objects this can be quite noticeable. However this artifact is not associated with particular locations in the scene but rather with particular view rays. If the moving object were seen in a close up the artifact would likely be much weaker because the distribution of depth values along the view ray would be less complex.

It should also be noted that we inherit some artifacts and limitations from prefiltered single scattering. Most importantly, strong leaking occurs at the epipole because independent of depth it corresponds to $\theta = 0$ [Klehm et al. 2014b]. Thus, the light is visible as a bright dot through walls (see supplementary video). Near the epipole light leaking is generally stronger than elsewhere. Warping may help to

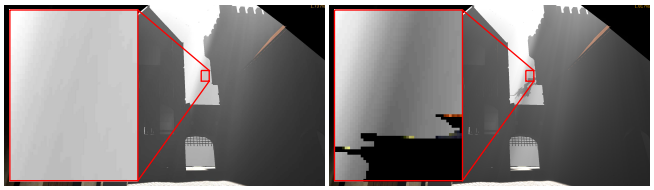


Figure 7: *A scene rendered using prefiltered single scattering with six moments. As a dragon enters, crepuscular rays are not only darkened below but also above it. Contrasts in magnified insets have been stretched by a factor of four.*

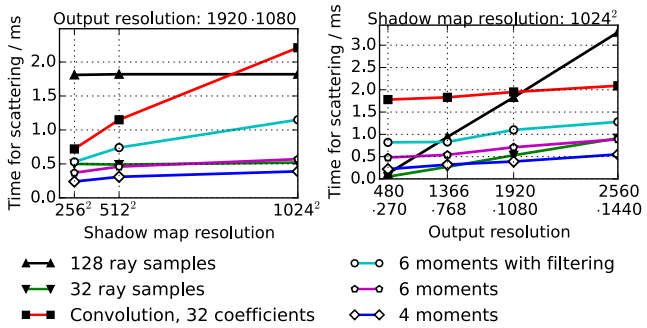


Figure 8: *The contribution to the frame time due to single scattering techniques as function of shadow map resolution and output resolution. Note that shadow map generation is not included in these timings because the same (moment) shadow map is used for single scattering and for rendering of shadows.*

reduce this [Lauritzen 2008] but we have not tried this. Besides there are the limitations mentioned in Section 2.2, i.e. homogenous media, directional lights only and no multiple scattering.

Figure 8 provides a systematic analysis of the run time of the various techniques. These timings exclude the cost for scene rendering and generation of the shadow map. Thus they are almost entirely scene independent. Overall techniques with prefiltering have a higher cost per texel of the shadow map whereas ray marching has a high cost per pixel on screen. Prefiltered single scattering with six moments offers high-quality results and using a shadow map with resolution 1024^2 at output resolution $1920 \cdot 1080$ it takes 0.71 ms. This should make it highly attractive for applications with tight frame time budgets such as games. Using four moments the technique is even faster but in this case single scattering should only be used as subtle effect due to stronger artifacts.

Prefiltered single scattering with convolution shadow maps using 32 coefficients is substantially more expensive than our novel variants. This is largely due to bandwidth requirements. Storing 32 coefficients takes 256 bits per texel whereas our techniques only take 64 bits per texel. For low output resolutions and scenes with low complexity ray marching with 32 samples can perform similarly well but it makes it hard to avoid objectionable noise everywhere.

Filtering increases the cost for generating the prefiltered moment shadow map significantly. For a 1024^2 shadow map and an output resolution of $1920 \cdot 1080$ the total cost goes up from 0.71 ms to 1.1 ms. Thus, it is not necessarily the most attractive option but in many scenes the quality improvement is substantial as demonstrated in Figure 6 and in the supplementary video.

In conclusion the combination of prefiltered single scattering with moment shadow maps appears to provide an attractive solution for many interactive applications. The run time is short compared to previous techniques and the produced artifacts are perceptionally unproblematic in most cases. Compared to convolution shadow maps ringing is eliminated entirely and the technique is accelerated. The variant with six moments and without filtering provides an excellent trade-off between quality and run time. Filtering provides a substantial quality improvement but comes at a cost. Four moments should only be used for subtle scattering.

3 Soft Shadows

Hard shadows assume an infinitesimally small light source that is occluded either entirely or not at all. Obviously, this is an unrealistic assumption. Natural light sources always have some extent and correspondingly cast soft shadows with smooth penumbra regions where the light is partially occluded. Filtered hard shadows can have a similar look but lack important characteristics. Most notably they remain soft near the occluder.

The currently most practical approximation to soft shadows in performance-sensitive real-time applications is to filter hard shadows with a filter size that depends linearly upon the distance between occluder and receiver [Fernando 2005]. This yields convincing results at the cost of excessive sampling of the shadow map.

In the following we demonstrate how to implement such an approximation using moment shadow maps. The technique is largely analog to variance soft shadow mapping [Yang et al. 2010] but behaves much more robustly thanks to the additional information in the moment shadow map. Our technique is the first to provide robust approximate soft shadows using only two queries to a four-channel summed-area table.

3.1 Related Work

Various techniques attempt to compute physically based shadows from shadow maps. Backprojection takes a single shadow map as discrete geometry representation and estimates the occluded area on the light source [Guennebaud et al. 2006]. Stochastic soft shadow mapping transfers depth of field techniques using filterable shadow maps [Liktor et al. 2015]. GEARS accelerates exact ray-triangle intersection tests using a shadow map [Wang et al. 2014]. While these techniques can produce accurate soft shadows, they are too costly for many interactive applications.

More practical methods (including ours) build upon percentage-closer soft shadows [Fernando 2005]. This technique can only get accurate results under the assumption of a single planar occluder which is parallel to the planar light source. Per pixel it performs a blocker search sampling the shadow map to determine the average depth of occluding geometry. Then the adequate size of the penumbra can be estimated by exploiting the planarity assumption. For a directional light the penumbra size is simply proportional to the distance between receiver and occluder. Finally percentage-closer filtering with a corresponding filter size generates the penumbra. Percentage-closer soft shadows generates plausible results with few noticeable artifacts but the cost is high due to excessive sampling. Temporal coherency may be exploited to amortize this cost over multiple frames [Schwärzler et al. 2013].

Summed-area variance shadow maps [Lauritzen 2007] try to avoid this sampling using a summed-area table. A summed-area table [Crow 1984] is a prefiltered representation of a texture where each texel stores the integral over a rectangle from the left top to its location. By sampling a summed-area table at the four corners of an arbitrary rectangle the integral over this rectangle can be queried (see Figure 9a). A summed-area table of a variance shadow map enables filtering with an arbitrary filter size in constant time but the blocker search still requires sampling.

Variance soft shadow mapping [Yang et al. 2010] accelerates the blocker search using a heuristic based on a single query to a summed-area variance shadow map. To improve performance and quality a hierarchical shadow map identifies the umbra and fully lit regions early and where appropriate smaller kernels are used to avoid artifacts. In some cases a fallback to percentage-closer filtering is needed. Convolution soft shadow mapping [Annen et al. 2008a] uses either a summed-area table or mipmaps to filter based on convolution shadow maps. The blocker search uses a second set of filterable textures. Exponential soft shadow mapping [Shen et al. 2013] uses summed-area tables over smaller regions of an exponential shadow map to avoid catastrophic precision loss. Again additional textures are needed for the blocker search. The authors use kernel subdivision to better approximate Gaussian filter kernels.

3.2 Summed Area Tables with Four Moments

Our technique follows the same basic steps as variance soft shadow mapping. We generate a summed-area table of a moment shadow map, use it to estimate average blocker depth during the blocker search, estimate the appropriate filter size and use the summed-area table to perform the filtering. We first discuss generation of the summed-area table.

This can be done in much the same way as generation of prefix sums for prefiltered single scattering (see Section 2.6). We use a compute shader with one thread per row to generate horizontal prefix sums. Then a second compute shader takes the result as input and generates vertical prefix sums thus producing the final summed-area table. As for single scattering we have not evaluated other schemes.

For small variance shadow maps the precision provided by summed-area tables with single-precision floating point values can be sufficient but for moment shadow maps it is generally insufficient. We instead use 32-bit integers and modular arithmetic because this allows us to exploit prior knowledge about maximal kernel sizes [Lauritzen 2007].

Suppose that the largest used kernel covers $N \in \mathbb{N}$ texels (e.g. $N = 256$ for a $16 \cdot 16$ kernel). Moments $b_1, \ldots, b_4$ initially lie in the interval $[0, 1]$. If we multiply them by $\frac{2^{32}-1}{N}$ and round to integers afterwards, the sum of moments stored in the largest relevant kernel is known to lie in $\{0, \ldots, 2^{32} - 1\}$. Thus, this number can be represented by a 32-bit unsigned integer. At the same time we still have a precision of $\log_2 \frac{2^{32}-1}{N}$ which evaluates to 24 bits for the above example. This is on par with with single precision floats.

To make use of this we generate such an integer moment shadow map and generate an integer summed-area table for it. During this step overflows will occur frequently but they can be safely ignored because they only subtract multiples of 2^{32}. When we query the summed-area table in a kernel containing N texels or less, we know that the result has to lie in $\{0, \ldots, 2^{32} - 1\}$ and thus computing it in integer arithmetic necessarily leads to the correct result.

In practice we initially create the moment shadow map with single precision floats. This way we do not need to implement a custom resolve for multisample antialiasing. Conversion to integers happens on the fly during creation of horizontal prefix sums.

3.3 Blocker Search

For the blocker search we perform a single look-up in the summed-area table to query four moments for the search region. We then use these moments and the biased fragment depth to reconstruct a distribution consisting of three depth values as described in Section 1.2. The inherent assumption of our blocker search is that the resulting distribution $\sum_{i=1}^{3} w_i \cdot \delta_{z_i}$ is correct. If the filter kernel does not contain more than two distinct surfaces, this assumption is well-justified. For three surfaces it is still justified as long as the blocker search is performed for one of them.

Since this distribution is correct by assumption, we can derive average blocker depth in analogy to percentage closer soft shadows:

$$\frac{\sum_{i=2,\, z_i<z_f}^{3} w_i \cdot z_i}{\sum_{i=2,\, z_i<z_f}^{3} w_i}$$

However, this formulation is not robust. The divisor is exactly the shadow intensity computed for the search region. It can be arbitrarily small or even exactly zero. In this case the expression becomes meaningless. A small shadow intensity implies an unoccluded fragment. For such fragments the blocker search should return $z_1 = z_f$ to indicate that a small filter kernel is to be used.

This requirement can be incorporated into the above formula robustly by setting average blocker depth to

$$\frac{\varepsilon_{z_1} \cdot z_1 + \sum_{i=2,\, z_i<z_f}^{3} w_i \cdot z_i}{\varepsilon_{z_1} + \sum_{i=2,\, z_i<z_f}^{3} w_i}$$

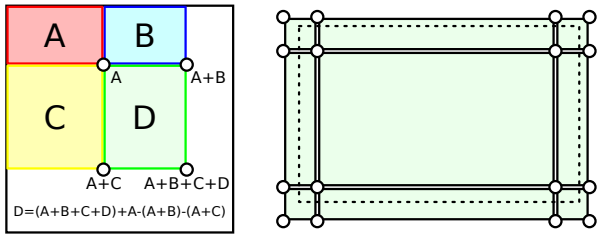
where $\varepsilon_{z_1} > 0$ is a small constant. Greater values move average blocker depth towards the receiver slightly thus making shadows harder. The choice $\varepsilon_{z_1} = 0.01$ worked well in all experiments.

3.4 Filtering

Once average blocker depth is available the penumbra estimation [Fernando 2005] provides an adequate filter size. From this and the texture coordinate of the fragment within the shadow map we can compute the left top and right bottom texture coordinates of the filter region. In general these will not lie in the center of texels. This necessitates interpolation for our integer summed-area tables.

Conversely to what one might expect it is incorrect to apply bilinear interpolation directly to samples at the four corners of the filter region because the underlying values of adjacent texels differ by an unknown multiple of 2^{32}. Such problems can be avoided by operating exceptionally on integrals over regions containing less than N texels. Figure 9b demonstrates how the filter region can be partitioned into nine such regions. For each of these regions it is safe to convert the held moments back to floating point values. Then the results from the individual regions can be summed weighting them by the area of their intersection with the filter region. This works reliably but since $4 \cdot 4 \cdot 4 \cdot 32 = 2048$ bits need to be loaded per fragment, the cost is significant (see Section 3.6). As an alternative we tried randomized dithering but found that the noise is too strong at hard shadow boundaries.

The filtered moments from the rectangular filter kernel are used as input to Hamburger moment shadow mapping to compute the final shadow intensity. Since precision of the moments is high, little biasing is required. The same value as for moment shadow mapping with single-precision floats can be used (i.e. $\alpha = 5 \cdot 10^{-6}$).



(a) *Rectangular query* **(b)** *Query with interpolation*

Figure 9: (9a) *Summed-area tables enable computation of the integral over a rectangle D with four samples.* (9b) *To compute the integral over an arbitrary rectangle, we load values of* 16 *texels and compute the integrals over the nine shown rectangles.*

3.5 Optimization

The most efficient way to optimize the technique is to reduce the number of texture loads. To avoid the cost of interpolation during the blocker search, we extend the search region to match the texel grid. This can lead to slight discontinuities in the soft shadows in some situations but is easily justified by the considerable speedup.

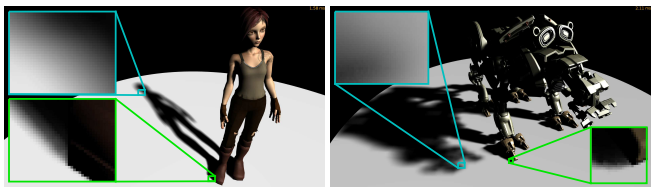
Another way to avoid texture loads is to skip filtering when the blocker search reveals that the fragment lies in the fully lit region or in the umbra. To check this we compute the shadow intensity $\sum_{i=2,\, z_i<z_f}^{3} w_i$ from available quantities. If it surpasses a threshold $1 - \varepsilon_u$ where $\varepsilon_u > 0$, we assume that the fragment lies in the umbra. We skip filtering and instead output one for full shadow. This works robustly using $\varepsilon_u = 0.01$ and eliminates the need for texture loads in big connected regions.

In the same way we can skip filtering if the shadow intensity for the search region is less than or equal to a threshold $\varepsilon_l \geq 0$. Unfortunately, this parameter is more problematic. Fine structures casting shadows over short distances contribute little to a shadow computed with a kernel as big as the search region. The shadowed fragments may falsely be classified as fully lit. Whether it is possible to find an adequate value for ε_l is scene dependent. Therefore, we do not use this optimization in our experiments.

3.6 Results and Discussion

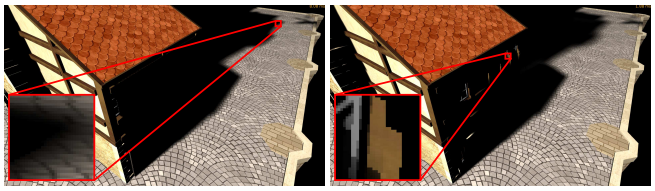
We compare moment soft shadow mapping against percentage-closer soft shadows and a naïve implementation of variance soft shadow mapping in a forward renderer using a single directional light. For percentage-closer soft shadows we benefit from hardware-accelerated $2 \cdot 2$ percentage-closer filtering to take four samples at once in the filtering step. The blocker search loads all texels in the search region to avoid artifacts for fine structures. Our implementation of variance soft shadow mapping uses neither a hierarchical shadow map nor kernel subdivision. It is essentially identical to moment soft shadow mapping but with two instead of four moments. Thus, we expect it to be faster than the actual technique but with more artifacts. All techniques skip filtering if the result of the blocker search allows it. As before timings refer to a Direct3D 11 implementation running on an nVidia GeForce GTX 980.

Figure 10 shows two examples where moment soft shadow mapping produces plausible soft shadows. It works well for character models (Fig. 10a) but also for complex models with many fine details (Fig. 10b). As expected shadows harden at contact-points with the occluder.

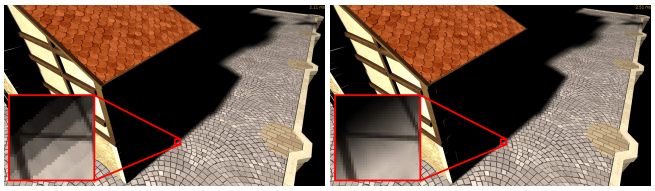


(a) *Sintel, 39k triangles, 1.58 ms* **(b)** *Quadbot, 459k triangles, 2.11 ms*

Figure 10: *Moment soft shadow mapping on two complex models above a plane. Timings are for full scene rendering at* $1920 \cdot 1080$ *with a* 1024^2 *shadow map and* $4\times$ *multisample antialiasing. Note that the shadows are contact hardening.*



(a) *Percentage-closer soft shadows,* $15 \cdot 15$ *search region, 8.08 ms* **(b)** *Naïve variance soft shadow mapping, interpolated, 1.88 ms*

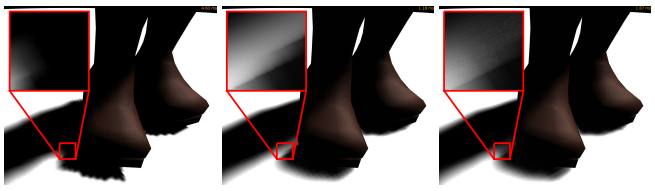


(c) *Moment soft shadow mapping, not interpolated, 2.11 ms* **(d)** *Moment soft shadow mapping, interpolated, 2.51 ms*

Figure 11: *A comparison of various techniques for soft shadows running at* $1920 \cdot 1080$ *with a* 1024^2 *shadow map. Timings are for full scene rendering.*

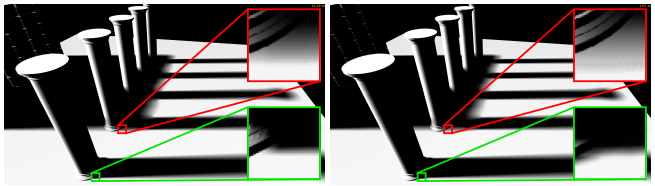
Figure 11 compares all implemented techniques for soft shadows. Percentage-closer soft shadows (Fig. 11a) generates a good result but to get an acceptable run time the search region has to be limited making long-range shadows too hard. For techniques based on summed-area tables we use a $27 \cdot 27$ search region and thus long-range shadows are as soft as they should be. Our naïve implementation of variance soft shadow maps produces objectionable light leaking (Fig. 11b). Note that kernel subdivision would fix this but at an increased cost. Moment soft shadow mapping without interpolation produces visible stripe patterns at hard shadow boundaries (Fig. 11c). Interpolation eliminates this artifact (Fig. 11d).

Figure 12 demonstrates a failure case with light leaking. The right boot casts a shadow onto the ground over short range. This area also receives shadow from the silhouette of the left boot. This silhouette becomes visible as bright line when using either variance or moment shadow maps. However, with moment shadow maps the effect is quite weak. The high precision in the summed-area table requires little biasing making moment soft shadow mapping perform as good as Hamburger moment shadow mapping with single precision shadow maps.



(a) *Percentage-closer soft shadows* **(b)** *Naïve variance soft shadow mapping* **(c)** *Moment soft shadow mapping*

Figure 12: *The shadow of Sintel's boots. Moment soft shadow mapping can produce light leaking for short-range shadows.*



(a) *Percentage-closer soft shadows* **(b)** *Moment soft shadow mapping*

Figure 13: *An example of wrong occluder fusion. The hardness of the contact shadows in the top and bottom magnifiers should be the same. Both shown techniques exhibit this artifact but light leaking of moment shadow maps strengthens it.*

Another failure case is shown in Figure 13. Like all techniques based on the framework of percentage-closer soft shadows, moment soft shadow mapping does not fuse occluders at different depths correctly. It rather replaces them by a single occluder at the average occluder depth. Therefore, the short-range shadow of a pillar becomes soft due to the shadow of the more distant wall (Fig. 13a). This artifact occurs whenever a kernel contains more than two surfaces. Thus it coincides with increased light leaking making the artifact more noticeable for moment soft shadow mapping (Fig. 13b). Still shadows are smooth and therefore the artifact is not very distracting.

Figure 14 provides timings. As usual techniques using filterable shadow maps have a high cost per texel of the shadow map whereas percentage-closer soft shadows has a high cost per pixel of the output. Especially the blocker search in percentage-closer soft shadows is costly. Subsampling can accelerate this but introduces artifacts for fine structures. Even for a $9 \cdot 9$ blocker search techniques with filterable shadow maps perform substantially better than percentage-closer soft shadows at a shadow map resolution of 1024^2. We also find that the cost of interpolation for moment soft shadow mapping is significant but not catastrophic. Using a 1024^2 shadow map at $1920 \cdot 1080$ the cost goes up from 2.7 ms to 3.48 ms.

Overall moment soft shadow mapping should be robust and fast enough for use in production. Percentage-closer soft shadows found use in many shipped titles. The only artifact that moment soft shadow mapping adds is light leaking which is weak due to the little bias. On the other hand it enables antialiasing of the shadow map which is particularly important for good results at contact points (Figure 12) and lifts limitations with regard to the size of the search region. Besides the dependence of the run time on scene complexity is reduced making it more predictable. Under realistic assumptions moment soft shadow mapping is substantially faster.

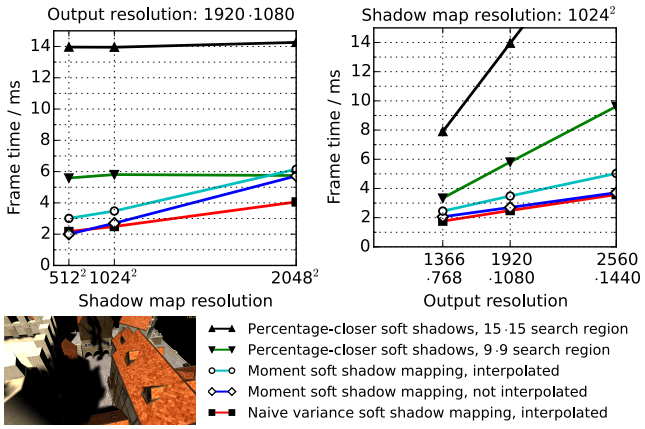


Figure 14: *Frame times for full scene rendering of the scene at the bottom left. For main scene rendering and for filterable shadow maps we use 4× multisample antialiasing. Note how the gradients of the graphs in the two plots indicate the cost per texel of the shadow map and per pixel of the output, respectively.*

4 Translucent Occluders

As our last contribution we use moment shadow mapping for translucent occluders by simply rendering to a moment shadow map with alpha blending. In presence of complicated translucent occluders light leaking increases but for simple scenarios it provides an adequate solution at almost no cost. We handle neither caustics nor subsurface scattering and require sorted fragments.

4.1 Related Work

Transmittance from the light to a surface can depend upon the depth in complex ways when translucent occluders are present. Deep shadow maps [Lokovic and Veach 2000] approximate this by a piecewise linear function of depth and compress this representation. This guarantees a high quality but maps to graphics hardware poorly. Loosing these guarantees enables an implementation using bounded memory that maps to graphics hardware better [Salvi et al. 2010]. To provide a less intricate method opacity shadow maps sample the function at predefined depths [Kim and Neumann 2001]. Translucent shadow maps [Dachsbacher and Stamminger 2003] provide an image-based solution for subsurface scattering.

Stochastic shadow maps [Enderton et al. 2010] randomly discard fragments in the shadow map in proportion to their translucency. Percentage-closer filtering then leads to a translucent shadow with little noise. This has been extended to colored objects [McGuire and Enderton 2011].

Fourier opacity mapping [Jansen and Bavoil 2010] introduces the idea of using filterable shadow maps, namely convolution shadow maps [Annen et al. 2007]. The authors represent the absorption function of translucent occluders by a convolution shadow map. Since absorption can be accumulated additively, no sorting is needed when generating the convolution shadow map. Translucent shadow maps [Delalandre et al. 2011] take a similar approach but represent transmittance by the convolution shadow map and combine their technique with ray marching to render single scattering.

4.2 Moment Shadow Maps for Translucent Occluders

Our approach is like translucent shadow maps in that the moment shadow map represents transmittance. Representing an absorption function would require an additional channel for the total absorption. Besides we want to use a single moment shadow map for opaque and translucent occluders but opaque occluders correspond to infinite absorption.

The disadvantage resulting from this choice is that we need to implement a method for order independent transparency when rendering to the moment shadow map. We consider this orthogonal to our contribution and any existing method should work (e.g. stochastic transparency [Enderton et al. 2010]). In our experiments translucent geometry has been sorted back to front manually.

Given $n \in \mathbb{N}$ surfaces along a light ray at depths $z_1 < \ldots < z_n$ with opacities $\alpha_1, \ldots, \alpha_n \in [0, 1]$ the amount of light transmitted to depth z is the product of the relevant transmittance factors $\prod_{j=1,\, z_j<z}^{n}(1-\alpha_j)$. The corresponding shadow intensity is the cumulative probability at z of the distribution

$$Z := \sum_{i=1}^{n} \left(\prod_{j=1}^{i-1} 1-\alpha_j \right) \cdot \alpha_i \cdot \delta_{z_i}$$

because at surface i the fraction α_i of the remaining light is blocked. The weights in this sum can also be understood as the visibility function [Enderton et al. 2010]. Thus, by rendering vectors of moments $(z_i, z_i^2, z_i^3, z_i^4)^\mathsf{T}$ with any method for order independent transparency, we get

$$b := \sum_{i=1}^{n} \left(\prod_{j=1}^{i-1} 1-\alpha_j \right) \cdot \alpha_i \cdot (z_i, z_i^2, z_i^3, z_i^4)^\mathsf{T}$$

which is exactly the vector of moments of the distribution Z.

We now have means to get the exact moments of a distribution of depth values that correctly models transmittance. Errors are only introduced when the shadow intensity is underestimated during reconstruction. Note that without loss of generality $z_n = \alpha_n = 1$ because we clear the moment shadow map with a corresponding background color. In consequence total alpha (resp. the zeroth moment) does not need to be stored because it would always be one.

To implement this technique we need to work around a limitation of current graphics APIs. The alpha value used for alpha blending cannot be independent from the values written to RGBA textures. Thus, we can no longer store moment shadow maps in RGBA textures. Instead we use two two-channel textures, each with 16 bits per channel. We render to them as multiple render targets so performance is only mildly reduced. Of course the optimized affine transform still has to be applied for 16-bit quantization.

Another minor slowdown arises because we can no longer generate the moment shadow map from a depth buffer. We have to use a pixel shader to compute the moments z_i, z_i^2, z_i^3, z_i^4 and the opacity α_i. Since the opacity has to be made available, we also may need to handle materials separately that would otherwise be rendered to the shadow map together.

4.3 Results and Discussion

Figure 15 shows results of our technique. Throughout the scene we obtain plausible shadows. In particular the smoke is darker at the bottom due to self shadowing. The orange pipes, the wall and the ground all receive partial shadow from the translucent smoke. Shadows of opaque occluders blend in naturally. Treating translucent occluders as opaque reduces the frame time to 1 ms, so the overhead for handling the translucency is small. Rendering translucent shadows with variance shadow maps takes 0.71 ms but light leaking is unacceptable.

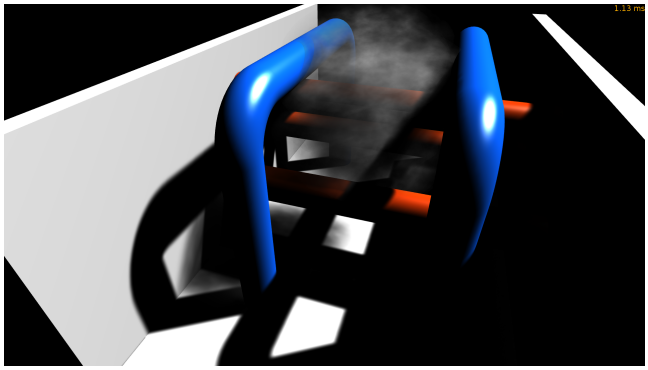
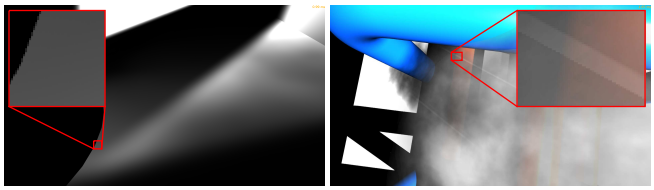


Figure 15: *A test scene with walls, pipes and smoke (represented by 15 translucent, textured planes). Rendering takes 1.13 ms at* $1920 \cdot 1080$ *with a* 1024^2 *moment shadow map. Note how everything casts shadow onto everything and how translucent occluders lead to partial shadow.*



(a) *Light leaking on the ground* **(b)** *Volumetric light leaking*

Figure 16: *Close-ups showing two kinds of light leaking.*

Figure 16 highlights failure cases. Since translucent occluders cause partial shadow over large areas, they can also contribute to the complexity of depth distributions in large areas. Therefore, light leaking is no longer restricted to the silhouettes of objects (Fig. 16a). Approximation errors of partial shadows depend on the complexity of depth distributions within the kernel. Thus silhouettes of opaque occluders (e.g. the blue pipe) effect partial shadows at locations closer to the light leading to erroneous stripes (Fig. 16b).

In spite of these approximation errors moment shadow maps for translucent occluders should prove sufficient for many applications. The low overhead makes them particularly attractive. Ringing, which is an issue for Fourier opacity maps and translucent shadow maps, cannot occur. No filtering is needed to avoid noise, which is an advantage over stochastic shadow maps. The technique can be extended to colored, translucent occluders by using one moment shadow map per color channel.

5 Conclusions

We have transferred moment shadow mapping to three new applications and all three benefited. Prefiltered single scattering with six moments can produce noise-free, antialiased and plausible single scattering efficiently. Moment soft shadow mapping creates results that are very similar to those of percentage-closer soft shadows but large blocker search regions barely effect the run time. For translucent occluders we obtain plausible shadows with little effort in terms of implementation and run time. The three techniques work together naturally with a single moment shadow map as input.

All these techniques scale favorably to high output resolutions but less favorably to large shadow maps. Though, antialiasing applied to moment shadow maps generally reduces the need for high shadow map resolutions. Their scaling behavior makes the techniques especially attractive for 4K rendering and virtual reality but even at $1920 \cdot 1080$ the run time is often better compared to techniques based on common shadow maps.

With our work we hope to establish moments as a standard tool in graphics. They provide a generic way to compress, filter and reconstruct one-dimensional distributions. This can be useful for more than just shadows. For example, trigonometric moments have recently been used for transient imaging [Peters et al. 2015]. We are looking forward to future applications in all fields of graphics.

Acknowledgments

We would like to thank Jonathan Klein and Zdravko Velinov for helpful discussions and feedback and Ralf Sarlette for hardware support. The models in Figure 10 are courtesy of the Blender foundation (see sintel.org and tearsofsteel.org). The models in Figure 7 are courtesy of Enrico Steffen and Zoltan Miklosi, the scene in Figure 5 is courtesy of Eugene Kiver.

References

ANNEN, T., MERTENS, T., BEKAERT, P., SEIDEL, H.-P., AND KAUTZ, J. 2007. Convolution shadow maps. In *EGSR07: 18th Eurographics Symposium on Rendering*, Eurographics Association, 51–60.

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. (Proc. SIGGRAPH 2008) 27*, 3 (Aug.), 34:1–34:8.

ANNEN, T., MERTENS, T., SEIDEL, H.-P., FLERACKERS, E., AND KAUTZ, J. 2008. Exponential shadow maps. In *GI '08: Proceedings of graphics interface 2008*, Canadian Information Processing Society, 155–161.

BARAN, I., CHEN, J., RAGAN-KELLEY, J., DURAND, F., AND LEHTINEN, J. 2010. A hierarchical volumetric shadow algorithm for single scattering. *ACM Trans. Graph. (Proc. SIGGRAPH Asia 2010) 29*, 6 (Dec.), 178:1–178:10.

CHEN, J., BARAN, I., DURAND, F., AND JAROSZ, W. 2011. Real-time volumetric shadows using 1d min-max mipmaps. In *Symposium on Interactive 3D Graphics and Games*, ACM, 39–46.

CROW, F. C. 1977. Shadow algorithms for computer graphics. In *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, ACM, SIGGRAPH '77, 242–248.

CROW, F. C. 1984. Summed-area tables for texture mapping. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, ACM, SIGGRAPH '84, 207–212.

DACHSBACHER, C., AND STAMMINGER, M. 2003. Translucent shadow maps. In *Eurographics Workshop on Rendering*, The Eurographics Association.

DELALANDRE, C., GAUTRON, P., MARVIE, J.-E., AND FRANÇOIS, G. 2011. Transmittance function mapping. In *Symposium on Interactive 3D Graphics and Games*, ACM, 31–38.

DOBASHI, Y., YAMAMOTO, T., AND NISHITA, T. 2002. Interactive rendering of atmospheric scattering effects using graphics hardware. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, Eurographics Association, 99–107.

DONNELLY, W., AND LAURITZEN, A. 2006. Variance shadow maps. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games*, ACM, 161–165.

ENDERTON, E., SINTORN, E., SHIRLEY, P., AND LUEBKE, D. 2010. Stochastic transparency. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, 157–164.

ENGELHARDT, T., AND DACHSBACHER, C. 2010. Epipolar sampling for shadows and crepuscular rays in participating media with single scattering. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, 119–125.

FERNANDO, R. 2005. Percentage-closer soft shadows. In *ACM SIGGRAPH 2005 Sketches*, ACM.

GUENNEBAUD, G., BARTHE, L., AND PAULIN, M. 2006. Real-time soft shadow mapping by backprojection. In *EGSR06: 17th Eurographics Symposium on Rendering*, Eurographics Association, 227–234.

JANSEN, J., AND BAVOIL, L. 2010. Fourier opacity mapping. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, 165–172.

KIM, T.-Y., AND NEUMANN, U. 2001. Opacity shadow maps. In *Eurographics Workshop on Rendering*, The Eurographics Association.

KLEHM, O., SEIDEL, H.-P., AND EISEMANN, E. 2014. Prefiltered single scattering. In *Proceedings of the 18th Meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, 71–78.

KLEHM, O., SEIDEL, H.-P., AND EISEMANN, E. 2014. Filter-based real-time single scattering using rectified shadow maps. *Journal of Computer Graphics Techniques (JCGT) 3*, 3, 7–34.

LAURITZEN, A., SALVI, M., AND LEFOHN, A. 2011. Sample distribution shadow maps. In *Proceedings of the 2011 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, 97–102.

LAURITZEN, A. 2007. *GPU Gems 3*. Addison-Wesley, ch. Summed-Area Variance Shadow Maps, 157–182.

LAURITZEN, A. 2008. *Rendering antialiased shadows using warped variance shadow maps*. Master's thesis, University of Waterloo.

LIKTOR, G., SPASSOV, S., MÜCKL, G., AND DACHSBACHER, C. 2015. Stochastic soft shadow mapping. *Computer Graphics Forum 34*, 4, 1–11.

LOKOVIC, T., AND VEACH, E. 2000. Deep shadow maps. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., SIGGRAPH '00, 385–392.

MARTIN, T., AND TAN, T.-S. 2004. Anti-aliasing and continuity with trapezoidal shadow maps. In *EGSR04: 15th Eurographics Symposium on Rendering*, Eurographics Association, 153–160.

MAX, N. L. 1986. Atmospheric illumination and shadows. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, SIGGRAPH '86, 117–124.

MCGUIRE, M., AND ENDERTON, E. 2011. Colored stochastic shadow maps. In *Symposium on Interactive 3D Graphics and Games*, ACM, 89–96.

PETERS, C., AND KLEIN, R. 2015. Moment shadow mapping. In *Proceedings of the 19th Symposium on Interactive 3D Graphics and Games*, ACM, 7–14.

PETERS, C., KLEIN, J., HULLIN, M. B., AND KLEIN, R. 2015. Solving trigonometric moment problems for fast transient imaging. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 34*, 6 (Nov.).

REEVES, W. T., SALESIN, D. H., AND COOK, R. L. 1987. Rendering antialiased shadows with depth maps. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, ACM, SIGGRAPH '87, 283–291.

SALVI, M., VIDIMČE, K., LAURITZEN, A., AND LEFOHN, A. 2010. Adaptive volumetric shadow maps. *Computer Graphics Forum 29*, 4, 1289–1296.

SALVI, M., 2008. Probabilistic approaches to shadow maps filtering, February. A talk in the tutorial "Core Techniques and Algorithms in Shader Programming" at Game Developers Conference 2008.

SCHWÄRZLER, M., LUKSCH, C., SCHERZER, D., AND WIMMER, M. 2013. Fast percentage closer soft shadows using temporal coherence. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, 79–86.

SHEN, L., FENG, J., AND YANG, B. 2013. Exponential soft shadow mapping. *Computer Graphics Forum 32*, 4.

TÓTH, B., AND UMENHOFFER, T. 2009. Real-time volumetric lighting in participating media. In *Eurographics 2009 - Short Papers*, The Eurographics Association.

WANG, L., ZHOU, S., KE, W., AND POPESCU, V. 2014. GEARS: A general and efficient algorithm for rendering shadows. *Computer Graphics Forum 33*, 6, 264–275.

WILLIAMS, L. 1978. Casting curved shadows on curved surfaces. In *Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, ACM, SIGGRAPH '78, 270–274.

WYMAN, C., AND DAI, Z. 2013. Imperfect voxelized shadow volumes. In *Proceedings of the 5th High-Performance Graphics Conference*, ACM, 45–52.

WYMAN, C., AND RAMSEY, S. 2008. Interactive volumetric shadows in participating media with single-scattering. In *Interactive Ray Tracing, 2008. RT 2008. IEEE Symposium on*, 87–92.

WYMAN, C. 2011. Voxelized shadow volumes. In *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics*, ACM, 33–40.

YANG, B., DONG, Z., FENG, J., SEIDEL, H.-P., AND KAUTZ, J. 2010. Variance soft shadow mapping. In *Computer Graphics Forum*, vol. 29, 2127–2134.

ZHANG, F., SUN, H., XU, L., AND LUN, L. K. 2006. Parallel-split shadow maps for large-scale virtual environments. In *Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*, ACM, 311–318.