# Moment Shadow Mapping

Christoph Peters*
University of Bonn, Germany

Reinhard Klein†
University of Bonn, Germany

Percentage closer filtering, 1.73ms | Variance shadow mapping, 1.15ms | Exponential shadow mapping, 1.15ms | Moment shadow mapping, 1.46ms

**Figure 1:** *Filtered hard shadows produced by various techniques. Moment shadow mapping provides a high quality heuristic for the result of percentage closer filtering using one shadow map sample per fragment and 64 bits per shadow map texel. Variance and exponential shadow mapping are faster requiring only 32 bits per texel but produce more artifacts (red arrows). Rendering without shadows takes 0.64ms.*

## Abstract

We present moment shadow mapping, a novel technique for fast, filtered hard shadows. Like variance shadow mapping it allows for the application of all kinds of efficient texture filtering and antialiasing to its moment shadow map. However it is designed to provide a substantially higher quality. Moment shadow maps store four moments of the depth within the filter kernel. Using this information, our efficient algorithm computes the sharpest possible lower bound as approximation to the shadow intensity. The choice to compute such a bound using four moments is based upon an automated evaluation of thousands of alternatives and thus known to be optimal. To reduce memory and bandwidth requirements we present an optimized quantization scheme to allow 16-bit quantization of moment shadow maps. Our evaluation demonstrates that moment shadow mapping produces high quality results with a single shadow map sample per fragment using 64 bits per shadow map texel.

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

**Keywords:** shadow mapping, filtered hard shadows, variance shadow mapping, moment problem, automated evaluation

## 1 Introduction

The faithful rendering of fully dynamic, hard shadows is a feature of great importance in interactive 3D graphics. Shadows help the understanding of the structure of scenes and contribute strongly to the perceived realism. On the other hand, shadow computation can cost a big portion of the available frame-time budget and results often exhibit aliasing nonetheless. This is particularly true for techniques based on shadow mapping [Williams 1978], which is the prevalent approach due to its excellent hardware support and predictable run time.

*peters@cs.uni-bonn.de
†rk@cs.uni-bonn.de



Percentage closer filtering [Reeves et al. 1987] reduces the problem of aliasing by applying an appropriate reconstruction filter, thus smoothing the shadow boundary. Since the shadow depends upon the depth in shadow map space, this filtering can only be done per shaded fragment and efficient techniques for precomputation of filter kernels cannot be applied. This means that dozens of shadow map samples are needed per fragment leading to a poor run time.

A variety of heuristic techniques has been developed to overcome this problem [Donnelly and Lauritzen 2006; Annen et al. 2007; Salvi 2008; Annen et al. 2008; Lauritzen and McCool 2008]. All of them replace the depth values commonly stored in the shadow map by some depth-dependent, low-dimensional vector. Taking a single filtered sample from such a modified shadow map allows for an approximate reconstruction of the depth-dependent shadow intensity. All these techniques suffer from some characteristic artifacts and generally offer different trade-offs between quality, memory footprint and run time.

In this context, our main contribution is moment shadow mapping, a new heuristic providing a substantial quality improvement. This is achieved using a shadow map with four channels storing $z$, $z^2$, $z^3$ and $z^4$ where $z$ is the depth of occluders. Our efficient algorithm, presented in Sections 4.1 and 4.2, uses a single filtered sample from such a shadow map to compute the darkest possible shadow without ever overestimating the shadow. To reduce memory requirements, Section 4.3 describes an optimized quantization scheme enabling use of 16-bit integers for the channels of the shadow map.

The choice to store this particular data in the shadow map originates from our systematic analysis of thousands of alternatives in Section 3. We demonstrate that any choice of data immediately leads to a well-defined heuristic. Although most of these candidates do not admit efficient real-time algorithms, we introduce a numerical method efficient enough to evaluate them on a test data set.

Our evaluation in Section 3.2 reveals that many candidates perform similarly well and that moment shadow mapping is only slightly worse than the best found technique, which we dub trigonometric moment shadow mapping. This technique does admit a moderately efficient real-time algorithm but it does not compare favorable against moment shadow mapping in terms of stability and run time.

As can be seen from Figure 1 and the evaluation in Section 5, moment shadow mapping provides high quality results while inheriting all positive traits of other shadow mapping techniques, which allow for precomputation of filter kernels. This is achieved at a moderate memory consumption of 64 bits per shadow map texel.

## 2 Related Work

The prevalent techniques for rendering dynamic shadows build upon either *Monte Carlo ray tracing* [Cook et al. 1984], *shadow volumes* [Crow 1977] or *shadow mapping* [Williams 1978]. An excellent overview of the many aspects of rendering shadows in real-time can be found in [Eisemann et al. 2011]. In this work we focus on the efficient filtering of hard shadows generated with shadow mapping and applications thereof.

Shadow mapping generates hard shadows by exploiting that lit surfaces are visible to the light source. Using the light source as point of view, the scene is rendered to a *shadow map* which stores image-space depth values. Projecting the shadow map onto the scene and comparing the stored depth values to the actual depth values allows for an efficient shadow test. However, this image-based approach is prone to aliasing and to discretization artifacts known as *surface acne*. The latter can be counteracted through scene-dependent biasing of depth values [Dou et al. 2014].

A particularly problematic type of aliasing is undersampling, which leads to jagged shadow boundaries. Various techniques diminish this problem by intelligently sampling different parts of the shadow map at different resolutions. *Trapezoidal shadow maps* [Martin and Tan 2004] optimize a single projection matrix. *Sample distribution shadow maps* [Lauritzen et al. 2011] split the view frustum by exploiting knowledge from a screen-space depth buffer and then compute bounding boxes for the visible fragments to obtain a tight shadow map frustum for each split. *Virtual shadow maps for many lights* [Olsson et al. 2014] decide which part of which omni-directional shadow map needs to be rendered at which resolution in each frame.

Nonetheless, the elimination of aliasing artifacts requires filtering. *Percentage closer filtering (PCF)* [Reeves et al. 1987] achieves this by sampling the shadow map within an appropriate filter kernel, performing the shadow test for each sample and filtering the resulting shadow intensities. This can only be done per shaded fragment because the shadow test is a depth-dependent operation.

*Variance shadow mapping (VSM)* [Donnelly and Lauritzen 2006] uses a modified shadow map containing the depth and the squared depth. Given a filtered sample the mean and variance of the depth within the filter kernel can be derived. This allows for evaluation of Chebyshev's inequality, which provides a lower bound to the percentage of depth values closer than the fragment. This bound is used as approximation to PCF.

Similarly, *convolution shadow mapping (CSM)* [Annen et al. 2007] stores $M$ complex Fourier coefficients of the shifted unit step function used for the shadow test. A filtered sample provides Fourier coefficients of the depth-dependent shadow intensity within the filter kernel and a truncated Fourier series can be used to approximate it. This technique can produce arbitrarily good results at the expense of very high memory requirements.

*Exponential shadow mapping (ESM)* [Salvi 2008; Annen et al. 2008] stores $\exp(c \cdot z)$ where $c \gg 1$ is a constant and $z$ is the occluder depth. Then the Markov inequality is used to compute an approximating lower bound to the result of PCF. For various failure cases a fallback to PCF is suggested.

ESM generally produces a high quality on the hindmost receiver of partial shadow and in fully shadowed regions but behaves unstably at the boundary of shadow receivers. Conversely, VSM produces *light leaking* in fully shadowed regions if the variance within the filter kernel is high (Figure 1). To combine the strengths of both techniques, *exponential variance shadow maps (EVSM)* [Lauritzen 2008] use VSM with $\exp(c \cdot z), \exp(c \cdot z)^2$, as well as $\exp(-c \cdot z), \exp(-c \cdot z)^2$ and use the darker shadow.

To reduce light leaking of VSM in complex scenes, *layered variance shadow maps (LVSM)* [Lauritzen and McCool 2008] partition the depth interval intelligently and use separate variance shadow maps for each interval.

While the heuristic approach and the increased memory footprint are obvious drawbacks compared to PCF, all these techniques also share some advantages. Most importantly, arbitrary linear filtering operations can be applied to the corresponding shadow maps. This includes multisample antialiasing, mipmapping, anisotropic filtering and low-pass filters such as a two-pass Gaussian [Donnelly and Lauritzen 2006]. This way, a single filtered sample can be obtained efficiently and the costly sampling procedure of PCF is avoided.

Alpha blending is a linear operation as well and therefore the restriction of PCF to opaque shadow casters can be overcome. *Fourier opacity mapping* [Jansen and Bavoil 2010] builds upon CSM to generate shadows for participating media such as smoke. This approach is also compatible with other techniques such as VSM or our novel technique.

Efficient filtering of hard shadows also enables approximate soft shadows generated by area lights of moderate extent. *Percentage-Closer Soft Shadows* [Fernando 2005] first performs a blocker search in the shadow map to determine the average distance between occluder and receiver. This is used to estimate the size of the penumbra and then PCF with a corresponding kernel radius produces it. *Variance soft shadow maps* [Yang et al. 2010] generate a summed area table for a variance shadow map. This allows a heuristic blocker search and shadow filtering in constant time. To overcome shortcomings of VSM, an adaptive sampling scheme using knowledge from a hierarchical shadow map is proposed.

*Volumetric obscurance* [Loos and Sloan 2010] demonstrates that the approach of variance shadow mapping can also be transferred to the realm of screen-space ambient occlusion.

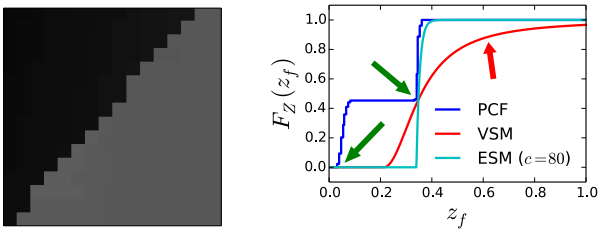
## 3 Filterable Shadow Maps

To systematically search for new heuristics for filtered hard shadows, we first need to make a few generalizing observations on existing techniques. Many of these techniques are known to have a useful interpretation in a probabilistic framework [Donnelly and Lauritzen 2006; Salvi 2008]. Since we build upon this framework in the following, we now introduce it in some detail.

All kinds of shadow maps store data directly computed from the depth of occluders in shadow map space. In general, a shadow map with $m \in \mathbb{N}$ channels is used and a map $\mathbf{b} : [0,1] \to \mathbb{R}^m$ assigns the occluder depth $z \in [0,1]$ at one shadow map texel to some vector $\mathbf{b}(z)$ stored in the shadow map. Shadow mapping represents the simplest case with $\mathbf{b}(z) := \mathbf{z}(z) := z$. Filtering is not a concern, so $z$ can be retrieved from the shadow map and the comparison $z < z_f$ reveals whether a fragment at depth $z_f \in [0,1]$ is in shadow.

In PCF the single occluder depth is replaced by many depth samples weighted according to some filter kernel. Conceptionally, this can be interpreted as picking depth values from the kernel at random and is adequately modeled by a probability distribution $Z$ on $[0,1]$ telling us the probability for each depth. The probabilities reflect the filter weights. In this formalism $Z(\mathbf{z} < z_f)$ is the probability that the randomly picked depth is less than the fragment depth, which is exactly the filtered shadow intensity computed by PCF.

The difficulty of the expression $Z(\mathbf{z} < z_f)$ lies in its dependence on $z_f$. The distribution $Z$ is uniquely identified by its left-continuous




**(a)** *The used* $16 \cdot 16$ *kernel.* **(b)** $F_Z$ *with VSM and ESM approximations.*

**Figure 2:** *A comparison of PCF, VSM and ESM on a kernel which constitutes an ideal case for VSM and ESM. Generally, the approximations are very coarse. Still two surfaces receive a correct shadow (green arrows). If there is an additional surface near the depth of the red arrow, VSM produces light leaking.*

cumulative distribution function $F_Z(z_f) := Z(\mathbf{z} < z_f)$. Being a real function it stems from an infinite-dimensional space. For filterable shadow maps we require a compressed representation of this function taking only a few bytes.

Suppose a shadow map storing $\mathbf{b}(z)$ instead of $z$ is filtered using the kernel previously used for PCF. Once more filtering can be understood as picking from the kernel at random. The expected value of the picked vector is the filtered vector:

$$b := \mathbb{E}_Z(\mathbf{b}) \in \mathbb{R}^m$$

It serves as compressed representation of $Z$. While $\mathbf{b}(z)$ provides redundant information about $z$, $Z$ is strongly underdetermined by knowledge of $b$. The benefit from this linear compression scheme is that it is compatible with hardware-accelerated texture filtering. The shadow map is *filterable*.

Now the challenge lies in approximating $F_Z(z_f)$ knowing only $b$ but not $Z$. We refer to this approximation as $G(b, z_f) \in [0, 1]$. Its derivation requires an intelligent choice of $\mathbf{b}$ and domain-specific heuristics. VSM uses $\mathbf{b}(z) := (z, z^2)^\mathsf{T}$, ESM uses $\mathbf{b}(z) := e^{c \cdot z}$ and for CSM $\mathbf{b}$ consists of $m = 2 \cdot M$ Fourier basis functions. VSM, ESM, EVSM and LVSM implement the approximation by computing lower bounds to $F_Z(z_f)$, i.e. $G(b, z_f) \leq F_Z(z_f)$. The produced shadow is never too dark. CSM is also commonly biased in such a way that its reconstruction provides a lower bound.

The reasoning behind this ubiquitous choice of lower bounds is demonstrated in Figure 2. The immediate increase of $F_Z(z_f)$ as $z_f$ passes the depth of occluders generally causes surface acne. For results without artifacts this increase has to occur between the maximal depth of the occluder and the minimal depth of the receiver. Using lower bounds to $F_Z(z_f)$ accounts for this requirement in a well-defined fashion.

Hence, we strive for a technique which guarantees lower bounds. On the other hand we want little light leaking, i.e. the lower bound should be as sharp as possible. These two requirements immediately lead to a well-defined problem [Kemperman 1968].

**Problem 1** (General moment problem)**.** Given $I \subseteq \mathbb{R}$, $b \in \mathbb{R}^m$, $z_f \in I$ and $\mathbf{b} : I \to \mathbb{R}^m$, consider the search space

$$\mathfrak{S}(b) := \{S \in \mathfrak{P}(I) \mid \mathbb{E}_S(\mathbf{b}) = b\}$$

of probability distributions $S$ on $I$ having $b$ as compressed representation. We strive to compute the sharp lower bound

$$G(b, z_f) := \inf_{S \in \mathfrak{S}(b)} F_S(z_f).$$

**Algorithm 1** Solution to Problem 1 for finite $I$.
**Input:** $I := \{z_1, \ldots, z_n\} \subset \mathbb{R}$, $\mathbf{b} : I \to \mathbb{R}^m$, $b \in \mathbb{R}^m$, $z_f \in I$
**Output:** $S \in \mathfrak{S}(b)$ minimizing $F_S(z_f)$ (or failure)

1. $A := (\mathbf{b}_j(z_i))_{j \in \{1,\ldots,m\}, i \in \{1,\ldots,n\}} \in \mathbb{R}^{m \times n}$
2. $\overline{A} := \begin{pmatrix} 1 \cdots 1 \\ A \end{pmatrix} \in \mathbb{R}^{(m+1) \times n}$, $\overline{b} := \begin{pmatrix} 1 \\ b \end{pmatrix} \in \mathbb{R}^{m+1}$
3. $p \in \mathbb{R}^n$ with $p_i := \begin{cases} 1 & \text{if } z_i < z_f \\ 0 & \text{otherwise} \end{cases}$
4. Using linear programming find $w \in \mathbb{R}^n$ minimizing $p^\mathsf{T} \cdot w$ subject to $\overline{A} \cdot w = \overline{b}$ and $w_i \geq 0$ for all $i \in \{1, \ldots, n\}$.
   (a) On success: Return $S := \sum_{i=1}^n w_i \cdot \delta_{z_i}$ (where $\delta_{z_i}$ denotes a Dirac-delta distribution with support at $z_i$)
   (b) On failure: Indicate $\mathfrak{S}(b) = \emptyset$

$\mathbb{E}_Z(\mathbf{b}) = b$ implies $Z \in \mathfrak{S}(b)$ and thus $G(b, z_f)$ is indeed a lower bound to $F_Z(z_f)$ no matter how $Z$ is chosen. At the same time $Z$ could be any of the distributions in $\mathfrak{S}(b)$ without violating our knowledge. By definition of the infimum this means that $G(b, z_f)$ provides the sharpest possible lower bound. Hence, solutions to the general moment problem provide optimal shadow mapping techniques. Note that the choice of $I$ allows us to include prior knowledge about the admissible domain of depth values.

For $I = \mathbb{R}$ and $\mathbf{b}$ as stated above, VSM and ESM both provide solutions to Problem 1. LVSM solves Problem 1 if overlap between depth intervals is disregarded. EVSM computes a lower bound, which is not sharp in the above sense. If CSM is configured to compute lower bounds, this bound is not sharp either. In conclusion Problem 1 provides a generic framework for the derivation of shadow mapping techniques in the spirit of VSM and ESM.

The general moment problem is well-studied and general statements on the structure of minimizing distributions exist [Kemperman 1968]. However, obtaining efficient closed-form solutions for arbitrary choices of $\mathbf{b}$ is not possible. We need to identify a single $\mathbf{b}$ that suits the purpose of shadow mapping well while allowing for an efficient solution.

### 3.1 Numerical Solution

As an intermediate step we describe a method to compute arbitrarily good approximations to the exact solution of Problem 1 in arbitrary cases. This solution is too inefficient for the computation of shadow intensities in real-time applications, but due to its generality it is useful for the evaluation of choices of $\mathbf{b}$.

The major difficulty with algorithmic approaches to Problem 1 is the infinite-dimensional search space $\mathfrak{S}(b)$. Obviously, this can be overcome by means of discretization of the considered distributions, that is by choosing $I \subset [0, 1]$ as finite set. As the number of samples in $I$ grows, we obtain good approximations to $I = [0, 1]$.

$\mathfrak{S}(b)$ is convex and subject to the linear constraints $S(I) = 1$ and $\mathbb{E}_S(\mathbf{b}) = b$. Within this search space the linear functional $F_S(z_f)$ needs to be optimized. For finite $I$ the set $\mathfrak{S}(b)$ is finite-dimensional and Problem 1 can be solved using linear programming. Linear programming has already been used for a special case of Problem 1 in [Prékopa 1990]. This is generalized by Algorithm 1.

**Proposition 2.** *Algorithm 1 solves Problem 1.*

*Proof sketch.* One easily verifies that the constraints in step 4 are equivalent to $S \in \mathfrak{S}(b)$. Furthermore, $p^\mathsf{T} \cdot w = F_S(z_f)$ and thus the correct functional is being minimized. □

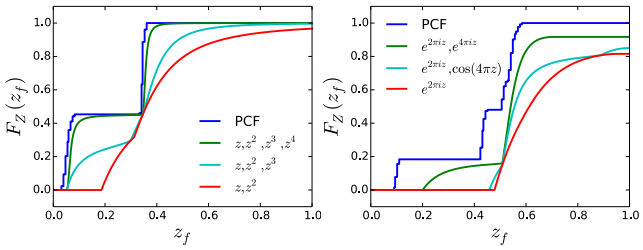



**Figure 3:** *Optimal, lower bounds for two different cumulative distribution functions (blue) obtained with different choices of* $\mathbf{b}$ *(see legend). Each plot contains approximations with* $m = 2$ *(red),* $m = 3$ *(cyan) and* $m = 4$ *(green). The approximations with* $m = 4$ *are clearly superior to those with* $m = 3$*.*

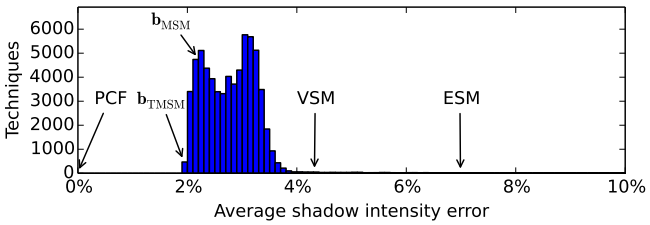



**Figure 4:** *A histogram showing how many candidate techniques produce an average error within a particular interval. For reference the error values resulting from PCF, VSM, ESM and the choices of* $\mathbf{b}$ *in Equation (1) are annotated.*

### 3.2 Choosing Shadow Map Data

In the following we develop a fully automated framework for evaluation of possible choices of $\mathbf{b} : [0, 1] \to \mathbb{R}^m$ using Algorithm 1. Since we are interested in filtered hard shadows, PCF with a $9 \cdot 9$ Gaussian filter kernel with $\sigma = 2.4$ texels and sophisticated biasing is our reference solution. Therefore, we require an error term measuring similarity to PCF.

In the end, shadow computations yield the irradiance received through direct lighting from a single light source. By definition, techniques solving Problem 1 can only produce various forms of light leaking as artifact because they never overestimate the shadow intensity. A simple but meaningful way to quantify this light leaking is the use of an $\mathcal{L}^1$-metric on the irradiance field.

To evaluate the performance on a whole scene, we use a single directional light. In this case, the radiant power received by a surface is directly proportional to the area covered in the shadow map (neglecting shadows). This enables a convenient image-based evaluation of the error.

First we use the stencil buffer to render a series of shadow maps displaying not only the foremost surface but all surfaces. Then we use Algorithm 1 to compute a shadow intensity for each fragment seen in this stack of shadow maps. Computing the average difference between these values and the shadow intensities produced by PCF yields our error term. If we disregard the minor distortions introduced by discretization of the shadow map, this error term agrees with the $\mathcal{L}^1$-error of the irradiance field divided by the total radiant power. At the same time it can be understood as weighted average error of the shadow intensity.

We evaluate on four different views of three different scenes providing challenging but realistic test cases (see supplementary material). Since we aim for a substantially higher quality than VSM and ESM, we are willing to spend some extra memory. Figure 3 indicates that techniques using shadow maps with four channels clearly outperform techniques with two or three channels. Thus, we fix $m := 4$.

It remains to choose the component functions of $\mathbf{b} : [0, 1] \to \mathbb{R}^4$. A priori it is unclear which sets of functions perform well. This is the question we are concerned with after all. To have any hope of solving Problem 1 in closed-form, we focus on a set of 37 rather elementary, smooth functions ranging from rational functions to exponentials and trigonometric functions. All of them are given in the supplementary material among with more details needed for reproduction of our results. We evaluate every possible combination of these functions, leading to $\binom{37}{4} = 66045$ candidate techniques.

In total, our evaluation requires 392 billion evaluations of Algorithm 1 which we have performed over a time span of several weeks on a cluster of computers. Due to the large number of candidates, a complete review of the results is not possible at this point. Fortunately, the data allows for a simple conclusion. Figure 4 demonstrates that thousands of techniques perform only slightly worse than the best technique. Thus, there is a large pool of promising candidates to pick from.

Intuitively, the surprisingly small variations in quality can be explained by the fact that many smooth functions can be approximated well by polynomials of degree $m = 4$ on the interval $[0, 1]$. If the first four moments are known, the expectations of all such polynomials can be computed. Thus, knowledge of the expectations of four smooth functions yields an amount of information similar to that of four moments.

It remains to select candidates with a small error that also allow for a closed-form solution of Problem 1. In this regard, two choices of $\mathbf{b}$ are particularly interesting because they are well-studied:

$$\begin{aligned} \mathbf{b}_{\text{MSM}}(z) &:= (z, z^2, z^3, z^4)^\mathsf{T} \\ \mathbf{b}_{\text{TMSM}}(z) &:= (\sin(2\pi z), \cos(2\pi z), \sin(4\pi z), \cos(4\pi z))^\mathsf{T} \end{aligned} \tag{1}$$

For these choices Problem 1 is known as *moment problem* and *trigonometric moment problem*, respectively. Solving the moment problem produces an error value of $2.19\%$. The trigonometric moment problem actually realizes the minimal measured error of $1.93\%$. Since both error values are among the smallest, we focus the remainder of our discussion on these two choices.

## 4 Moment Shadow Mapping

To find the best solution among the remaining candidates we have developed efficient algorithms for three different shadow mapping techniques. We now shortly discuss our findings for all three techniques but then focus on the technique that works best. The other two techniques are further described in the supplementary material among with proofs of all related propositions.

*Hamburger four moment shadow mapping (Hamburger 4MSM)* is our main algorithm. When there is no need to distinguish it from the other two techniques we also refer to it as *moment shadow mapping (MSM)*. We prefer this technique because it uses the fastest and most robust algorithm. It solves Problem 1 for $I = \mathbb{R}$ and $\mathbf{b} = \mathbf{b}_{\text{MSM}}$ which is known as Hamburger moment problem with four moments, hence the name.

Besides its favorable comparison to the other two techniques this technique also has a unique property, making its use more convenient for developers and designers alike. Whenever a shadow map is rendered, the near and far plane have to be fixed. Some algorithms such as ESM and CSM produce worse results as the distance

**Algorithm 2** Solution to Problem 1 for $I = \mathbb{R}$ and $\mathbf{b}(z) = (z^j)_{j=1}^m$ (Hamburger MSM).
**Input:** $m \in 2 \cdot \mathbb{N}$, $b \in \mathbb{R}^m$, $z_f \in \mathbb{R}$
**Output:** $G(b, z_f)$

1. $n := \frac{m}{2} + 1$
2. $B := (b_{j+k-2})_{j,k=1}^n \in \mathbb{R}^{n \times n}$ with $b_0 := 1$
3. $z_1 := z_f$
4. $c := B^{-1} \cdot (1, z_1, \ldots, z_1^{n-1})^\mathsf{T} \in \mathbb{R}^n$
5. Solve $\sum_{k=1}^n c_k \cdot z^{k-1} = 0$ for $z$ and let $z_2, \ldots, z_n \in \mathbb{R}$ denote the solutions.
6. $\hat{A} := (z_i^{j-1})_{j,i=1}^n \in \mathbb{R}^{n \times n}$
7. $w := \hat{A}^{-1} \cdot (1, b_1, \ldots, b_{n-1})^\mathsf{T} \in \mathbb{R}^n$
8. Return $G := \sum_{i=1, z_i < z_f}^n w_i$

between these planes increases, leaving content creators with the burden of choosing them tightly. In the supplementary document we prove that Hamburger 4MSM is the only technique described by Problem 1 where this choice does not influence the result at all.

*Hausdorff four moment shadow mapping (Hausdorff 4MSM)* is almost identical to Hamburger 4MSM except that it uses $I = [0, 1]$, i.e. it incorporates the prior knowledge that depth values have to lie within this interval. This adds a branch to the reconstruction algorithm, which produces slightly darker results for short-range shadows. Unfortunately, this can amplify quantization artifacts when using 16-bit quantization, which is the main reason why we prefer Hamburger 4MSM. Still, Hausdorff 4MSM is a viable alternative.

*Trigonometric moment shadow mapping (TMSM)* is of interest to us because Section 3.2 has shown that it performs best among all 66045 candidates. It solves Problem 1 for $I = [0, 1]$ and $\mathbf{b} = \mathbf{b}_{\text{TMSM}}$. This problem is substantially harder than the problems encountered in the other two cases and only sub-problems have been solved. For the sake of this paper we have developed a novel algorithm solving it fully. It requires the solution of a quartic equation which is costly and can lead to instabilities. Nonetheless, it is useful for direct comparisons in Section 5 and these show that the reduction of light leaking compared to MSM is small, as predicted in Section 3.2.

### 4.1 Our Main Algorithm

We now introduce and analyze an algorithm for Hamburger 4MSM solving Problem 1 for $I = \mathbb{R}$ and $\mathbf{b} = \mathbf{b}_{\text{MSM}}$. The practical implementation and use of this algorithm is discussed in Section 4.2. Closed-form solutions to the present problem are known [Kreĭn and Nudel'man 1977] and corresponding algorithms have been designed [Tari 2005]. They even generalize to an arbitrary even number of moments $m \in 2 \cdot \mathbb{N}$. However, existing algorithms are optimized for an entirely different scenario. We suggest an algorithm that is well-suited for the present real-time application. It takes the place of Chebyshev's inequality in VSM. In fact, VSM arises as simplest special case for $m = 2$, though we are more interested in $m = 4$.

The sought-after solution is known to be realized by a linear combination $S \in \mathfrak{S}(b)$ of $n := \frac{m}{2} + 1$ Dirac-$\delta$ distributions. To minimize $F_S(z_f)$ one of the Dirac-$\delta$ distributions must have support at $z_f$. It can be shown that the remaining Dirac-$\delta$ distributions have to be located at the roots of a special polynomial. Once these roots are computed, the weights of the linear combination are determined by the system of linear equations $\mathbb{E}_S(\mathbf{b}) = b$ and $G(b, z_f) = F_S(z_f)$ can be evaluated. All of this is done by Algorithm 2.

**Proposition 3.** *If Algorithm 2 produces positive-definite $B$ and $c_n \neq 0$, it solves Problem 1 correctly. If a positive-definite $B \in \mathbb{R}^{n \times n}$ is fixed, $c_n = 0$ occurs for no more than $n - 1$ different values of $z_f$.*

Obviously, the two conditions of Proposition 3 require further attention. The case $c_n = 0$ leads to undefined results but is quite unproblematic in practice because it only occurs for isolated values of $z_f$. The most elegant way to avoid this case is to use Hausdorff 4MSM instead of Hamburger 4MSM but there is little gain from doing so.

The condition on $B$ is not as strong as it might seem. Its analysis actually reveals a strength of Hamburger 4MSM [Kreĭn and Nudel'man 1977, p. 63, p.78].
**Proposition 4.** *Let $b = \mathbb{E}_Z(\mathbf{b})$ for some probability distribution $Z$ on $\mathbb{R}$. Then the matrix $B$ is symmetric and positive semi-definite. Furthermore, the following statements are equivalent:*

1. $\det B = 0$,
2. *$Z$ is the only distribution with $\mathbb{E}_Z(\mathbf{b}) = b$,*
3. *$Z$ is a linear combination of at most $\frac{m}{2}$ Dirac-$\delta$ distributions.*

This special case is highly relevant. It is common for filter kernels in shadow maps to contain only a small number of different surfaces and the depth of these surfaces is often nearly constant. This situation is approximated well by a linear combination of one Dirac-$\delta$ distribution per surface. The majority of filter kernels on a shadow map can be approximated by no more than two Dirac-$\delta$ distributions. Proposition 4 tells us that Hamburger 4MSM can achieve perfect reconstruction in this case because $Z$ is uniquely determined by the equation $\mathbb{E}_Z(\mathbf{b}) = b$.

Adding a separate branch for this case to Algorithm 2 is not difficult. Essentially, it suffices to replace $c$ by a vector in the kernel of $B$. However, we found that such a solution does not behave robustly because it is difficult to distinguish the two cases. Instead, it is better to implement the algorithm such that it behaves robustly even for nearly singular $B$. As $B$ approaches singularity, the reconstruction $G$ better approximates the ground-truth $F_Z$.

### 4.2 Implementation

Implementation of Hamburger 4MSM is a two-step procedure much like VSM. First, a *moment shadow map* needs to be rendered. This works almost identical to rendering a common shadow map except that the four-channel moment shadow map stores four moments of the depth in shadow map space, $z$, $z^2$, $z^3$, $z^4$, rather than storing only the depth, $z$. The benefit from this redundant information is that linear filtering operations such as a two-pass Gaussian blur or generation of mipmaps may be applied to the moment shadow map without losing too much information.

Second, a filtered sample from the moment shadow map and the depth of a fragment in shadow map space need to be fed into Algorithm 2 to compute a filtered shadow intensity for this fragment. Performing this computation in a numerically stable fashion is nontrivial as was previously discussed in a blog post about a failed attempt to develop moment shadow mapping [Salvi 2007]. In the following, we derive a numerically stable implementation of Algorithm 2 for $m = 4$ which we summarize in Algorithm 3.

For $m = 4$, Algorithm 2 requires the solution of the $3 \times 3$ linear systems of equations $B \cdot c = (1, z_1, z_1^2)^\mathsf{T}$ and $\hat{A} \cdot w = \hat{b}$ and the solution of the quadratic equation $c_3 \cdot z^2 + c_2 \cdot z + c_1 = 0$. The latter can be solved with the quadratic formula without running into issues. The system $\hat{A} \cdot w = \hat{b}$ does not need to be solved completely.

**Algorithm 3** Hamburger 4MSM (special case of Algorithm 2).
**Input:** Filtered sample from the moment shadow map $b \in \mathbb{R}^4$, fragment depth $z_f \in \mathbb{R}$, bias $\alpha > 0$ (e.g. $\alpha = 3 \cdot 10^{-5}$)
**Output:** Shadow intensity $G(b, z_f)$

1. $b' := (1-\alpha) \cdot b + \alpha \cdot (0.5, 0.5, 0.5, 0.5)^\mathsf{T}$
2. Use a Cholesky decomposition to solve for $c \in \mathbb{R}^3$:

$$\begin{pmatrix} 1 & b'_1 & b'_2 \\ b'_1 & b'_2 & b'_3 \\ b'_2 & b'_3 & b'_4 \end{pmatrix} \cdot c = \begin{pmatrix} 1 \\ z_f \\ z_f^2 \end{pmatrix}$$

3. Solve $c_3 \cdot z^2 + c_2 \cdot z + c_1 = 0$ for $z$ using the quadratic formula and let $z_2, z_3 \in \mathbb{R}$ with $z_2 \leq z_3$ denote the solutions.
4. If $z_f \leq z_2$: Return $G := 0$.
5. Else if $z_f \leq z_3$: Return $G := \frac{z_f \cdot z_3 - b'_1 \cdot (z_f + z_3) + b'_2}{(z_3 - z_2) \cdot (z_f - z_2)}$.
6. Else: Return $G := 1 - \frac{z_2 \cdot z_3 - b'_1 \cdot (z_2 + z_3) + b'_2}{(z_f - z_2) \cdot (z_f - z_3)}$.

Dependent upon the location of $z_f$ with respect to $z_2$ and $z_3$, the shadow intensity $G$ evaluates to 0, $w_2$ or $1 - w_1$ (assuming $z_2 \leq z_3$). Corresponding closed forms are given in Algorithm 3.

It remains to solve $B \cdot c = (1, z_1, z_1^2)^\mathsf{T}$. From Proposition 4 we know that $B$ is symmetric and positive semi-definite. Thus, a Cholesky decomposition of the form $B = L \cdot D \cdot L^\mathsf{T}$ can be employed with $L \in \mathbb{R}^{3\times3}$ being lower triangular and $D \in \mathbb{R}^{3\times3}$ being diagonal. This decomposition is known to behave robustly even for nearly singular matrices [Trefethen and Bau 1997, p. 176]. The special structure of $B$ can be exploited for further optimizations.

If Algorithm 2 is implemented like this and fed with double precision moments, it produces results which agree with those generated by Algorithm 1. However, numerical noise appears as soon as the input moments are stored in single precision. This can be explained through Proposition 4. In many relevant cases $B$ is nearly singular but still positive semi-definite. Quantization errors can give rise to negative eigenvalues making Problem 1 insolvable.

Storing moments at double precision in the moment shadow map imposes an unacceptable memory footprint. We need a method to compensate for the loss of information introduced by quantization. Above all else this method has to guarantee robust results. We found that a simple biasing on the input works best in this regard.

We use a biased moment vector $b' := (1-\alpha) \cdot b + \alpha \cdot b^\star$ where $b \in \mathbb{R}^m$ is the unbiased but quantized moment vector, $0 < \alpha \ll 1$ is the strength of the bias and $b^\star \in \mathbb{R}^m$ is an appropriately chosen constant vector. This biased moment vector corresponds to the biased distribution $Z' := (1-\alpha) \cdot Z + \alpha \cdot Z^\star$ where $\mathbb{E}_{Z^\star}(\mathbf{b}) = b^\star$ and thus $G(b', z_f)$ is a lower bound to $F_{Z'}(z_f)$. For small $\alpha$ this should not introduce a large error.

It remains to choose $b^\star = \mathbb{E}_{Z^\star}(\mathbf{b})$. Ideally, $\det B$ grows as quickly as possible as $\alpha \ll 1$ is increased. Assuming a particular distribution of $b$, this requirement can be optimized in closed-form leading to $b^\star = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)^\mathsf{T}$ or equivalently $Z^\star = \frac{1}{2} \cdot (\delta_0 + \delta_1)$.

Algorithm 3 produces robust results using a bias of $\alpha = 2 \cdot 10^{-6}$ and single precision throughout the pipeline. Note that this bias is not scene dependent. Under some circumstances a rare artifact may occur where individual pixels obtain wrong shadow intensities outside the interval $[0, 1]$. It is related to $c_3$ approaching zero. This can be made less malicious by clamping $G$ and eliminated entirely by using Hausdorff 4MSM.

### 4.3 Optimized Moment Quantization

A truly efficient implementation of 4MSM should consume no more than 16 bits per moment. With a canonical approach this leads to unacceptable artifacts. Information theory provides a natural framework to improve on this situation. To employ it we define a random variable $\mathbf{x}_b$ on $[0, 1]^m$ modeling the distribution of the moment vector $b$ within a filtered moment shadow map and analyze its differential entropy [Cover and Thomas 2001, p. 13, p. 229].
**Definition 5.** Let $p_b : \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ denote the probability density function of $\mathbf{x}_b$. Then the *differential entropy* of $\mathbf{x}_b$ is given by

$$h(\mathbf{x}_b) := -\int_{\mathbb{R}^m} p_b(b) \cdot \log_2 p_b(b) \,\mathrm{d}\, b.$$

The differential entropy $h(\mathbf{x}_b)$ measures approximately the amount of entropy which is lost in a uniformly quantized version of $\mathbf{x}_b$ due to a non-uniform distribution of $\mathbf{x}_b$ [Cover and Thomas 2001, p. 229]. More specifically a quantized version of $\mathbf{x}_b$ using four 16-bit integers to represent the unit tesseract $[0, 1]^4$ has an entropy of approximately $4 \cdot 16 + h(\mathbf{x}_b)$ bits. Note that $h(\mathbf{x}_b)$ is always negative in the present case.

Storing data with a low entropy means that the available memory is used inefficiently. Hence, we should maximize $h(\mathbf{x}_b)$ by transforming $b$ prior to storing it. It is desirable that the transformed data can still be filtered linearly. This requirement restricts our choices to affine transforms $\theta_m : \mathbb{R}^m \to \mathbb{R}^m$ maximizing $h(\theta_m(\mathbf{x}_b))$. If we store $\theta_m(\mathbf{b}(z))$ in the moment shadow map, we can take filtered samples and reconstruct $b$ using $\theta_m^{-1}$. The amount of entropy we gain this way is directly related to the stretch of the volume $|\det \theta_m|$.
**Proposition 6.** *For regular* $\theta_m$

$$h(\theta_m(\mathbf{x}_b)) = h(\mathbf{x}_b) + \log_2 |\det \theta_m|.$$

Hence, we need to maximize $|\det \theta_m|$. At the same time $\theta_m$ has to map $\mathbf{b}([0, 1])$ into the set of representable vectors $[0, 1]^m$. For an arbitrary affine transform this can be enforced by scaling and translating $\theta_m(\mathbf{b}([0, 1]))$ such that its axis-aligned bounding box is $[0, 1]^m$. This way, we can find the optimal $\theta_m$ by means of numerical optimization. The resulting transform for $m = 4$ is

$$\theta_4(b) = (0.0359558848, 0, 0, 0)^\mathsf{T} + \begin{pmatrix} -2.07224649 & 32.2370378 & -68.5710746 & 39.3703274 \\ 13.7948857 & -59.4683976 & 82.035975 & -35.3649032 \\ 0.105877704 & -1.90774663 & 9.34965551 & -6.65434907 \\ 9.79240621 & -33.76521106 & 47.9456097 & -23.9728048 \end{pmatrix} \cdot b$$

and for $m = 2$ (i.e. VSM) we obtain

$$\theta_2(b) = \begin{pmatrix} 1 & 0 \\ 4 & -4 \end{pmatrix} \cdot b.$$

This way we gain 12.3 and 2 bits of entropy, respectively. For 4MSM this improvement makes 16-bit quantization applicable. In our experiments a stronger bias of $\alpha = 3 \cdot 10^{-5}$ is required and short-range shadows suffer from slight quantization noise but the halved memory and bandwidth requirements easily compensate for these drawbacks. For VSM the optimized quantization leads to a notable reduction of quantization artifacts.

For visualization purposes we can also consider the three-dimensional case $m = 3$. In this case we gain 6.2 bits of differential entropy and the three-dimensional curves and convex hulls can be visualized (see Figure 5). Note that the original convex hull is very flat whereas the expanded convex hull fills the unit cube nicely.

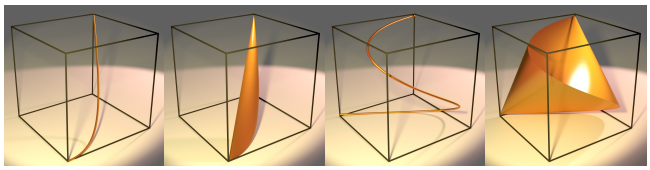
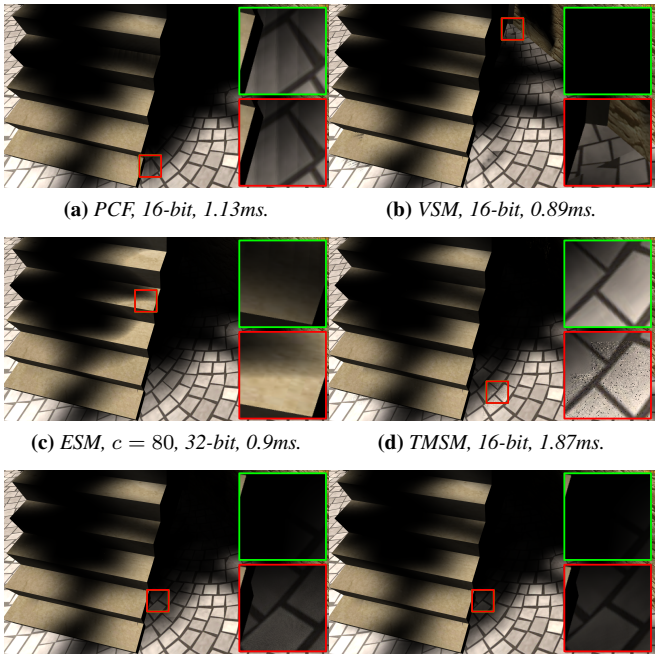



**Figure 5:** *A comparison of the curves* $\mathbf{b}$ *and* $\theta_3 \circ \mathbf{b}$ *and their hulls containing valid* $b$ *and* $\theta_3(b)$ *for* $m = 3$*. The transformed volume is* $0.41$ *and it is* $73.9$ *times larger than the untransformed volume.*



**(a)** *PCF, 16-bit, 1.13ms.* **(b)** *VSM, 16-bit, 0.89ms.*

**(c)** *ESM,* $c = 80$*, 32-bit, 0.9ms.* **(d)** *TMSM, 16-bit, 1.87ms.*

**(e)** *Hamburger 4MSM, 16-bit, 1.13ms.* **(f)** *Hausdorff 4MSM, 32-bit, 1.46ms.*

**Figure 6:** *The shadow of a flying dragon cast onto a stair. This challenging scenario provokes typical artifacts for all shown techniques. Artifacts are magnified (red) below the PCF ground truth (green). The frame time for rendering without shadows is 0.44ms.*

## 5 Results and Conclusion

4MSM provides an excellent approximation to PCF and produces less artifacts than other filterable shadow maps as shown in Figure 6. Shadows cast over extremely short ranges can suffer from slight quantization noise and light leaking (Figure 6e, 6f). The stronger bias $\alpha$ required for 16-bit quantization strengthens light leaking slightly (Figure 6e). TMSM produces just as much light leaking as Hausdorff 4MSM but suffers from robustness issues and a poor run time (Figure 6d).

With 16-bit quantization Hausdorff 4MSM is only slightly slower than Hamburger 4MSM and produces slightly darker shadows over short ranges. However, this can also amplify quantization artifacts. Hence, we consider Hamburger 4MSM preferable. Figure 1 and 8 show more results of this technique. Additionally, we recommend watching the supplementary video. Among other things it demonstrates the immensely positive effect of multisample antialiasing (MSAA) applied to the moment shadow map. This hardware feature is not applicable for PCF [Donnelly and Lauritzen 2006].

For our run time comparison in Figure 7 we use an nVidia GeForce GTX 780 and Direct3D 11 on a scene with one directional light and a shadow map uniformly covering the whole scene. We observe that the frame time for filterable shadow maps is governed by the memory per texel. VSM with 32-bit quantization and 4MSM with 16-bit quantization perform almost identical in spite of the greater complexity of Algorithm 3. VSM with 16-bit quantization (not shown) performs identical to ESM. Hamburger 4MSM outperforms PCF for large output resolutions, large filter kernels and small shadow maps. Note that PCF uses hardware support to quarter the number of taps and that ESM is implemented without fallback to PCF.

In conclusion, 4MSM provides filterable shadow maps with an unprecedented quality at a moderate cost of 64 bits per texel. Arguably, 4MSM is the best possible shadow mapping technique using this amount of memory. It should provide a good solution for many applications requiring high quality filtered hard shadows and we are hoping that it will enable new applications. In regard of 4K rendering, the demonstrated amortization of moment shadow mapping at greater output resolutions is promising. To the best of our knowledge, 4MSM is the first application of the theory of moments in a graphics context. This powerful theory holds the promise of providing many efficient heuristics for real-time graphics.

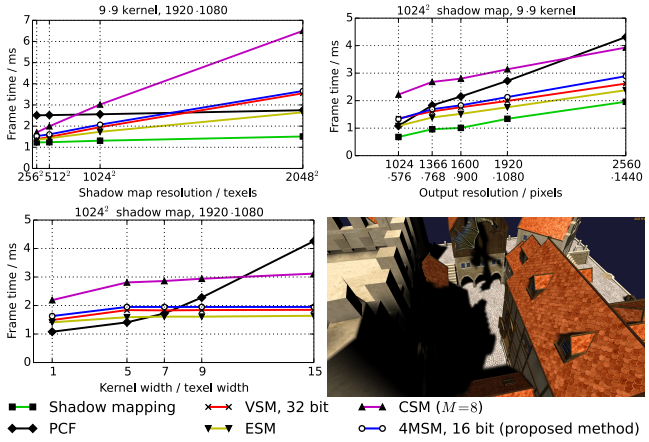



**Figure 7:** *The frame time for the shown scene plotted against the main parameters affecting the performance. Where applicable, 4x MSAA is used for the shadow map and for main scene rendering.*

## Acknowledgments

The authors would like to thank Paul Müller and Dominik Michels for helpful discussions and feedback, all reviewers for their comments on the exposition of this paper, Ralf Sarlette for supporting installation of the cluster used in Section 3.2 and the BlendSwap.com users Enrico Steffen and Zoltan Miklosi for the used models.

## 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, Grenoble, France, EGSR07, 51–60.

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, Toronto, Ont., Canada, GI '08, 155–161.

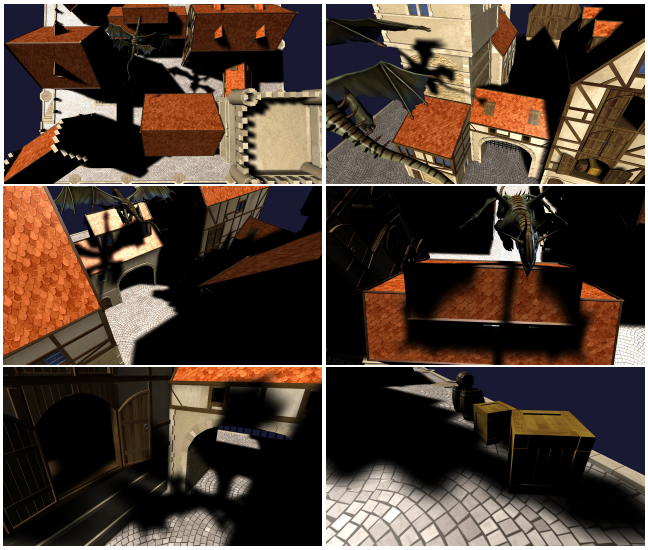


**Figure 8:** *Various views of a scene using Hamburger 4MSM with* 16*-bit quantization, a* $9 \cdot 9$ *Gaussian filter kernel and a* $1024^2$ *shadow map.*


COOK, R. L., PORTER, T., AND CARPENTER, L. 1984. Distributed ray tracing. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '84, 137–145.

COVER, T. M., AND THOMAS, J. A. 2001. *Elements of Information Theory*. John Wiley & Sons, Inc.

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

DONNELLY, W., AND LAURITZEN, A. 2006. Variance shadow maps. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D '06, 161–165.

DOU, H., YAN, Y., KERZNER, E., DAI, Z., AND WYMAN, C. 2014. Adaptive depth bias for shadow maps. In *Proceedings of the 18th Meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D '14, 97–102.

EISEMANN, E., SCHWARZ, M., ASSARSSON, U., AND WIMMER, M. 2011. *Real-Time Shadows*. An A K Peters book. CRC Press.

FERNANDO, R. 2005. Percentage-closer soft shadows. In *ACM SIGGRAPH 2005 Sketches*, ACM, New York, NY, USA, SIGGRAPH '05.

JANSEN, J., AND BAVOIL, L. 2010. Fourier opacity mapping. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D '10, 165–172.

KEMPERMAN, J. H. B. 1968. The general moment problem, a geometric approach. *Annals of Mathematical Statistics 39*, 1, 93–122.

KREĬN, M. G., AND NUDEL'MAN, A. A. 1977. *The Markov Moment Problem and Extremal Problems*, vol. 50 of *Translations of Mathematical Monographs*. American Mathematical Society.

LAURITZEN, A., AND MCCOOL, M. 2008. Layered variance shadow maps. In *Proceedings of graphics interface 2008*, Canadian Information Processing Society, Toronto, Ont., Canada, Canada, GI '08, 139–146.

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, New York, NY, USA, I3D '11, 97–102.

LAURITZEN, A. 2008. *Rendering Antialiased Shadows using Warped Variance Shadow Maps*. Master's thesis, University of Waterloo.

LOOS, B. J., AND SLOAN, P.-P. 2010. Volumetric obscurance. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D '10, 151–156.

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

OLSSON, O., SINTORN, E., KÄMPE, V., BILLETER, M., AND ASSARSSON, U. 2014. Efficient virtual shadow maps for many lights. In *Proceedings of the 18th Meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D '14, 87–96.

PRÉKOPA, A. 1990. The discrete moment problem and linear programming. *Discrete Applied Mathematics 27*, 3, 235–254.

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, New York, NY, USA, SIGGRAPH '87, 283–291.

SALVI, M., 2007. A (not so) little teaser, September. A blog post including notes on moment shadow mapping. pixelstoomany.wordpress.com/2007/09/03/a-not-so-little-teaser.

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.

TARI, Á. 2005. *Moments based bounds in stochastic models*. Ph.d. dissertation, Budapest University of Technology and Economics, Department of Telecommunications.

TREFETHEN, L. N., AND BAU, D. 1997. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104).

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

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.