# Moment-Based Methods for Real-Time Shadows and Fast Transient Imaging

**Christoph Peters**

# Moment-Based Methods for Real-Time Shadows and Fast Transient Imaging

Dissertation

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

**M.Sc. Christoph Jonathan Peters**

aus Köln

Bonn, September 2016

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn.

Dekan: Prof. Dr. Johannes Beck

1. Referent: Prof. Dr. Reinhard Klein
2. Referent: Prof. Dr. Elmar Eisemann
3. Referent: Prof. Dr. Michael Wimmer

Tag der Disputation: 5. Mai 2017
Veröffentlichungsjahr: 2017

# Contents

# Abstract

We apply the theory of moments to develop computationally efficient methods for real-time rendering of shadows and reconstruction of transient images from few measurements. Given moments of an unknown probability distribution, i.e. the expectations of known, real random variables, the theory of moments strives to characterize all distributions that could have led to these moments. Earlier works in computer graphics only use the most basic results of this powerful theory.

When filtering shadows based on shadow maps, the distribution of depth values within the filter region has to be estimated. Variance shadow mapping does this using two power moments. While this linear representation admits direct filtering, it leads to a very coarse reconstruction. We generalize this approach to use an arbitrary set of general moments and benchmark thousands of possible choices. Based on the results, we propose the use of moment shadow mapping which produces high-quality antialiased shadows efficiently by storing four power moments in 64 bits per shadow map texel.

Techniques for shadow map filtering have been applied to a variety of problems. We combine these existing approaches with moment shadow mapping to render shadows of translucent occluders using alpha blending, soft shadows using summed-area tables and prefiltered single scattering using six power moments. All these techniques have a high overhead per texel of the moment shadow map but a low overhead per shaded pixel. Thus, they scale well to the increasingly high resolutions of modern displays.

Transient images help to analyze light transport in scenes. Besides two spatial dimensions, they are resolved in time of flight. Earlier cost-efficient approaches reconstruct them from measurements of amplitude modulated continuous wave lidar systems but they typically take more than a minute of capture time. We pose this reconstruction problem as trigonometric moment problem. The maximum entropy spectral estimate and the Pisarenko estimate are known closed-form solutions to such problems which yield continuous and sparse reconstructions, respectively. By applying them, we reconstruct complex impulse responses with $m$ distinct returns from measurements at as few as $m$ non-zero frequencies. For $m = 3$ our experiments with measured data confirm this. Thus, our techniques are computationally efficient and simultaneously reduce capture times drastically. We successfully capture 18.6 transient images per second which leads to transient video. As an important byproduct, this fast and accurate reconstruction of impulse responses enables removal of multipath interference in range images.

## Acknowledgments

The work presented here is the result of a collaborative effort and I am grateful to everyone who took part in it. First and foremost I would like to thank my doctoral advisor Reinhard Klein. With his excellent master lectures and his open work group, he has paved my way from a bachelor in mathematics to graphics research. Over the years, we have had many productive and enjoyable discussions and I could always rely on his support.

I am just as grateful to my other coauthors. Our work on transient imaging would not have been possible without Matthias Hullin's exceptional expertise in this field. Jonathan Klein's master thesis gave the initial incentive for this project, he has helped conduct many experiments, provided software frameworks and our shared discussions are beyond counting. Cedrick Münstermann's bachelor thesis showed that prefiltered single scattering with moments is worthwhile and Nico Wetzstein's bachelor thesis demonstrated the promise of moment soft shadow mapping.

Beyond that, I would like to express my gratitude to Dominik Michels, whose council helped me find my way in academia, Paul Müller, who shared his extensive technical insight at many occasions, Sebastian Merzbach, who offered proofing and feedback for most of my works, Michael Weinmann, who has been a frequent source of feedback, Jaime Martín, who helped with measurements and Ralf Sarlette, who set up the computational resources used in Section 3.4 and helped with other hardware and experimental setups many times.

Furthermore, my gratitude goes to the anonymous reviewers of my submissions. In many cases, their comments helped to elevate the quality of the manuscripts before publication. I received additional valuable feedback from Roland Ruiters, Matt Pettineo, Andreas Kolb, Damien Lefloch, Hamed Sarbolandi, Zdravko Velinov, Martin Albrecht and Lutz Heyne. I am also grateful to the organizers of the venues I have attended and to our secretaries Simone von Neffe and Michaela Mettler.

Finally, I would like to thank my family and friends as well as my colleagues. The way to this dissertation has been long and stressful at times but they made it easy to keep up a positive attitude.

Part of this work was supported by the German Research Foundation (HU 2273/2-1) and by X-Rite Inc. through the Chair for Digital Material Appearance.

## Third-Party Data Sets

In real-time graphics, the evaluation of techniques on diverse and complex scenes is of paramount importance. In this regard, our work has benefited tremendously from online repositories of 3D models. Most of our models originate from BlendSwap.com and some additional models were retrieved from the Stanford 3D scanning repository[1]. We are grateful to the operators of these repositories and to everyone who contributed his works under permissive licenses.

In Figure 4.8 and many other figures throughout Part I, we use models by Enrico Steffen[2] and Zoltan Miklosi[3]. In Figures 1.1c and 6.2 we use models by the Blender foundation[4]. The model in Figure 7.4 is created by Eugene Kiver[5].

Additional models were used for the evaluation of candidate shadow mapping techniques (Fig. 3.5). The model in Figure 3.5c and 3.5d is a work by Chris Kuhn and Greg Zaal[6]. The scene in Figure 3.5a is a composite including works by Oscar Baechler[7] and Paulo Bardes[8] and the bunny and dragon from the Stanford 3D scanning repository.

Figures 8.1 and 8.4 use a transient image provided by Velten et al. [2013].

## Used Software

All projects described in this dissertation rely on software developed by third parties. In many cases this software is freely available under open-source licenses and I would like to thank all contributors.

For creation of most figures and for countless numerical experiments, the SciPy stack has been used, i.e. Python, NumPy, SciPy and Matplotlib. 3D models have been prepared using Blender and some vector graphics are created in Inkscape.

The software libraries GLPK and Gurobi have been used to solve linear programming problems and Ceres Solver has solved some non-linear optimization problems. To process massive computational workloads in parallel,

[1] graphics.stanford.edu/data/3Dscanrep/

[2] blendswap.com/blends/view/50534

[3] blendswap.com/blends/view/73418

[4] durian.blender.org, mango.blender.org

[5] blendswap.com/blends/view/59269

[6] blendswap.com/blends/view/66638

[7] blendswap.com/blends/view/4394

[8] blendswap.com/blends/view/3914 (retrieved on 1st of September 2016).

I used HTCondor. The shown real-time renderings employ Direct3D 11, DirectXTex, Eigen, FreeImage and FreeType. Some profiling has been done with NVIDIA Nsight.

All publications and this dissertation have been prepared in LyX using MiKTeX. As development environment I used Eclipse with PyDev as well as Microsoft Visual Studio. Notepad++ has been of great use at many occasions. Some computations have been aided by wxMaxima. Subversion has provided version control. Supplementary videos have been encoded using FFmpeg.

Chapter 1

# Introduction

The plausible rendition of shadows is one of the most intensely studied problems in rendering. Given their strong influence on the perceived realism of rendered scenes, this is unsurprising. Even laymen can immediately notice a lack of shadows. Shadows also provide important visual cues for geometric relations. Especially contact of two objects can hardly be conveyed without the use of shadows.

The prevalent approach for rendering dynamic shadows in modern real-time applications is shadow mapping [Williams 1978]. A shadow map is an image rendered from the point of view of the light source where each texel stores the depth $z$ of the foremost surface. By comparing the depth of a fragment $z_f$ to the corresponding depth from the shadow map, it can be decided whether the surface is visible to the light source and thus lit. While this image-based approach maps to rasterization hardware nicely and scales well with scene complexity, it is prone to aliasing. Applying texture filtering directly to the shadow map is not meaningful because it would smooth the shadow-casting geometry rather than the shadow signal.

The correct way to apply filtering to the shadow intensities is to sample the shadow map within a filter region to obtain depth values $z_0, \ldots, z_{n-1}$ [Reeves et al. 1987]. Then these depth values are converted to shadow intensities and filtered using weights from a filter kernel $w_0, \ldots, w_{n-1} \in [0, 1]$:

$$\sum_{l=0}^{n-1} w_l \cdot \begin{cases} 0 & \text{if } z_f \leq z_l \\ 1 & \text{if } z_f > z_l \end{cases} \tag{1.1}$$

Since this procedure depends on the fragment depth $z_f$, it can only be performed per fragment making it quite costly.

Variance shadow maps are an alternative approach that makes the shadow map directly filterable [Donnelly and Lauritzen 2006]. Rather than storing $z$ only, they store $z$ and $z^2$ in a two-channel texture. Then this texture is filtered within the filter region leading to values of the form

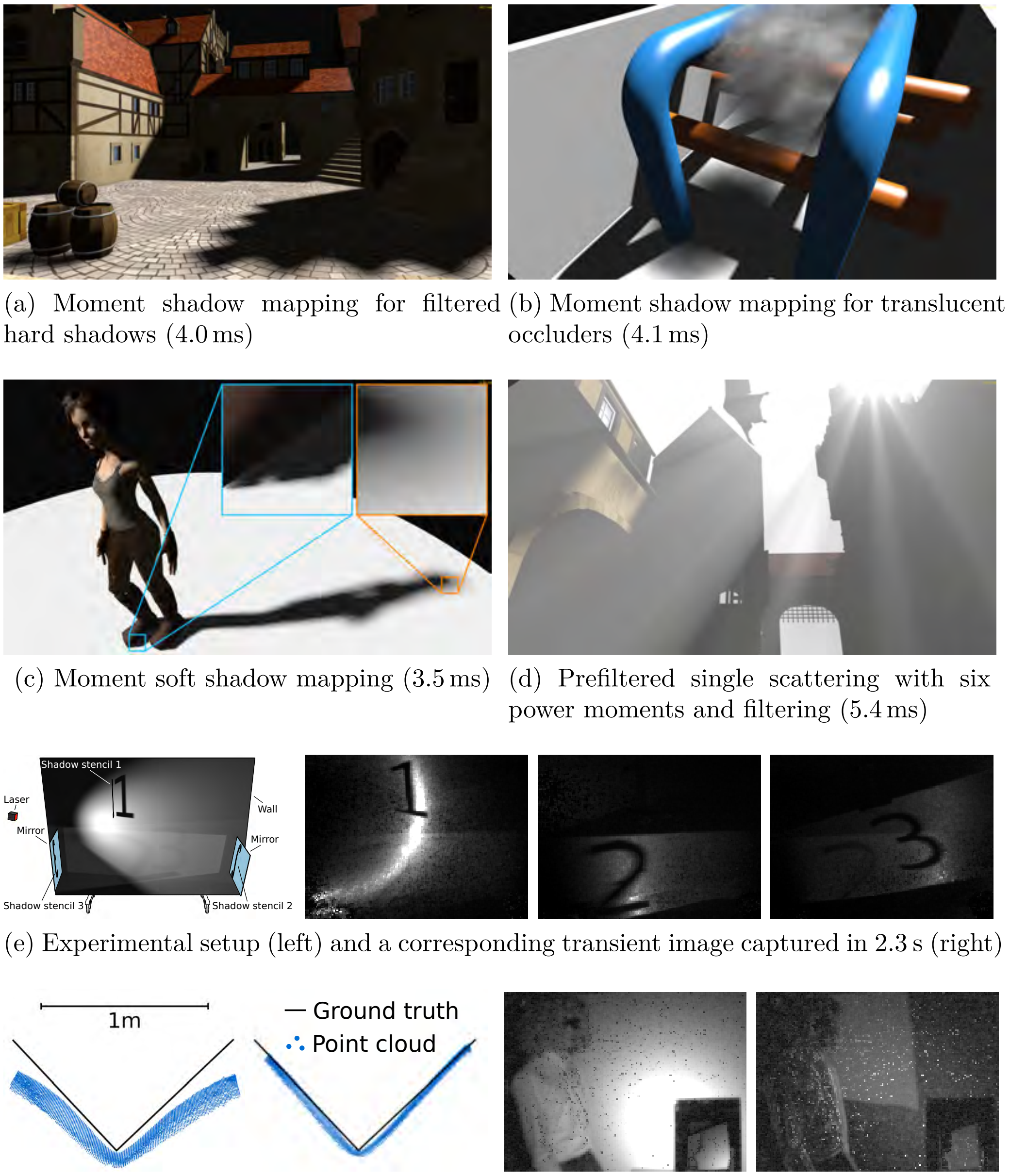
$$b_1 := \sum_{l=0}^{n-1} w_l \cdot z_l \qquad \text{and} \qquad b_2 := \sum_{l=0}^{n-1} w_l \cdot z_l^2.$$

These values are power moments of the distribution of depth values within the filter region. The first power moment $b_1$ is simply the mean. By combining it with the second power moment, we compute the variance $b_2 - b_1^2$. For a given $z_f$ we then use mean and variance to compute a lower bound to the expression in Equation (1.1) which serves as an approximation. In many relevant cases this works well but under some circumstances the shadow intensity will be underestimated substantially leading to objectionable artifacts known as light leaking. A lot of follow-up work has picked up this idea using shadow maps with different numbers of channels and different contents to reduce light leaking [Annen et al. 2007; Salvi 2008; Annen et al. 2008b; Lauritzen and McCool 2008].

In our work we generalize this idea. We consider shadow maps with $m \in \mathbb{N}$ channels storing $\mathbf{a}_1(z), \ldots, \mathbf{a}_m(z)$ where $\mathbf{a}_1, \ldots, \mathbf{a}_m : [-1, 1] \to \mathbb{R}$ are arbitrary continuous functions. We then demand that the shadow intensity is always underestimated but never more than necessary. This way, we immediately obtain a well-defined shadow mapping technique. Furthermore, we demonstrate how to compute its result in a discretized setting. This method is not fast enough for real-time rendering but provides a practical way to compare different choices of functions $\mathbf{a}_1, \ldots, \mathbf{a}_m$. We evaluate 66054 candidate techniques on real scenes.

Through this evaluation, we find that storing $z$, $z^2$, $z^3$ and $z^4$ in a shadow map with four channels is one among many choices which lead to minimal light leaking. For this particular case, we develop a highly optimized algorithm to evaluate the lower bound to the shadow intensity. This leads to our main technique for shadow mapping; moment shadow mapping (Fig. 1.1a). Thanks to an optimized quantization scheme, it only takes 64 bits per texel of the shadow map. At this memory consumption, it outperforms competing techniques quality-wise.

Just like previously proposed filterable shadow maps, this new type of shadow maps is useful for more than filtered hard shadows. We render shadows for translucent occluders by simply rendering to the moment shadow map with alpha blending (Fig. 1.1b). Approximate soft shadows for rectangular area lights are rendered very efficiently using summed-area tables of



(a) Moment shadow mapping for filtered hard shadows (4.0 ms)

(b) Moment shadow mapping for translucent occluders (4.1 ms)

(c) Moment soft shadow mapping (3.5 ms)

(d) Prefiltered single scattering with six power moments and filtering (5.4 ms)

(e) Experimental setup (left) and a corresponding transient image captured in 2.3 s (right)

(f) Elimination of diffuse multipath interference, single frequency (left) and ours (right)

(g) Real-time separation of direct (left) and indirect illumination (right)

Figure 1.1: An overview of the major applications of our work. The shown timings are full frame times for rendering at a resolution of 3840·2160 with 4× multisample antialiasing on an NVIDIA GeForce GTX 970.

a moment shadow map (Fig. 1.1c). By combining our work with prefiltered single scattering [Klehm et al. 2014b], we render shadows of directional lights in homogeneous participating media using only two texture reads per pixel (Fig. 1.1d). Compared to techniques based on sampling of a common shadow map, all these techniques have a higher overhead per texel of the shadow map but a low overhead per shaded fragment. Thus, they scale well to the ever-increasing resolutions of modern displays.

In all of this work, we draw on the theory of moments. Broadly speaking, this theory is concerned with the reconstruction of a probability distribution $\sum_{l=0}^{n-1} w_l \cdot \delta_{z_l}$ described by points of support $z_0, \ldots, z_{n-1} \in \mathbb{R}$ and probabilities $w_0, \ldots, w_{n-1} \in [0, 1]$ from its general moments $a_j := \sum_{l=0}^{n-1} w_l \cdot \mathbf{a}_j(z_l)$ where $j \in \{0, \ldots, m\}$. Such moment problems are very well-understood and a large body of mathematical literature provides diverse closed-form solutions to these inverse problems.

In the second part of our work, we apply related techniques to a problem from an entirely different field; transient imaging. As input data we use measurements from amplitude modulated continuous wave (AMCW) lidar systems. These imagers consist of an active illumination and a special sensor. The illumination is modulated with a periodic signal while the sensitivity of the sensor is modulated with a shifted version of the same signal. Thus, the contribution of light from the active illumination to the measurement at a pixel depends on the time of flight of the light through a periodic function.

The main application of AMCW lidar is range imaging. Four measurements with different shifts between the modulation of the illumination and the sensitivity of the sensor are captured. Assuming that all light reaching a pixel has the same time of flight, these measurements suffice to reconstruct the phase shift of the returning signal, which is proportional to range. Unfortunately, the assumption of a unique time of flight does not hold in presence of global illumination effects. Light scatters through the scene on many paths of varying length and a superposition of this light reaches each pixel. This effect is known as multipath interference and it leads to strong, systematic errors (Fig. 1.1f).

Transient images capture this complicated reality more completely. Rather than storing a single time of flight, they store a time-resolved impulse response per pixel. Equivalently, they can be regarded as videos with high temporal resolution recording the return of light to the sensor after illuminating the scene with an infinitesimally short light pulse. Capturing transient images directly is possible but the involved hardware is very expensive and capture times are in the magnitude of hours [Velten et al. 2013; Gkioulekas et al. 2015]. Still, this work has served to demonstrate their usefulness for applications such as non-line-of-sight imaging [Velten et al. 2012] and decomposition of illumination into direct lighting, indirect lighting and subsurface scattering [Wu et al. 2014].

Faster and more cost-efficient approaches for the acquisition of transient images use AMCW lidar [Heide et al. 2013; Kadambi et al. 2013]. The sensor effectively provides the correlation between a periodic signal and the transient image. By using many different modulation signals, enough information can be extracted to reconstruct the impulse responses approximately. In our work, we observe that a specific measurement procedure turns this reconstruction problem into a so called trigonometric moment problem. Again the theory of moments provides efficient solutions.

We use the maximum entropy spectral estimate to reconstruct continuous impulse responses (Fig. 1.1e) and the Pisarenko estimate for sparse impulse responses. Both solutions incorporate all measurements as hard constraint. Thus, few measurements suffice to reconstruct complex impulse responses. This is particularly true, if the impulse response is temporally sparse as it would be when specular global illumination dominates. Under such idealizing assumptions, a perfect reconstruction can be accomplished. Compared to related work, we reduce the necessary capture time heavily to the point where we record transient images at video frame rates (18.6 Hz, Fig. 1.1g). Once the full impulse response is available, the direct return can be extracted efficiently to reduce multipath interference in range imaging (Figs. 1.1g, 1.1f).

Until now, the theory of moments has barely found any attention in the graphics community. With our work we hope to establish moment-based methods as a standard tool in graphics research and practice. They are theoretically well-founded, computationally efficient and can often extract surprisingly much information from few moments, especially in presence of sparsity. Thus, they are a good match for many problems in graphics where solutions need not be exact but robust, fast and plausible in the most common cases.

## 1.1 Publications

Most of the work described in this dissertation has been previously presented at conferences and published in proceedings. In particular, we describe results from the following three publications:

- Peters, C. and Klein, R. (2015). Moment shadow mapping. In *Proceedings of the 19th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, i3D '15, pages 7–14. ACM, doi: 10.1145/2699276.2699277,
- 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 2015)*, 34(6), doi: 10.1145/2816795.2818103,
- Peters, C., Münstermann, C., Wetzstein, N., and Klein, R. (2016). Beyond hard shadows: Moment shadow maps for single scattering, soft shadows and translucent occluders. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, i3D '16, pages 159–170. ACM, doi: 10.1145/2856400.2856402.

The majority of the results in the first publication [Peters and Klein 2015] goes back to my master thesis [Peters 2013]. It introduces the relation between moment problems and filterable shadow maps, evaluates candidate techniques and describes moment shadow mapping for filtered hard shadows.

The second publication [Peters et al. 2015], which deals with transient imaging, has also been presented as an invited poster at the International Conference on Computational Photography 2016 in Evanston, Illinois. Besides, I held invited talks about this work at the University of Siegen on 19th of February 2016 and at the headquarters of pmdTechnologies AG in Siegen on 5th of April 2016.

The third publication [Peters et al. 2016], which transfers moment shadow mapping to three new applications, has been invited for an extended version to the Journal of Computer Graphics Techniques. This extension has not been submitted yet but some additional research has been conducted which is included in this dissertation. An overview of these and other previously unpublished results is provided in Appendix A.

Additionally, I disseminated the work on moment shadow mapping [Peters and Klein 2015; Peters et al. 2016] through a one hour lecture at the Game Developers Conference Europe in Cologne on 15th of August 2016.

I am also a coauthor on the following publication but its contents are not covered by this dissertation:

- Klein, J., Peters, C., Martín, J., Laurenzis, M., and Hullin, M. B. (2016). Tracking objects outside the line of sight using 2D intensity images. *Scientific Reports*, 6(32491), doi: 10.1038/srep32491

## 1.2 Outline

The thesis consists of two major parts. Part I discusses our work on real-time rendering of shadows using moment shadow maps and the many applications thereof. Part II discusses fast transient imaging and range imaging based on AMCW lidar systems. The common foundation of both parts in the theory of moments is laid out in Chapter 2.

Our generalized view on filterable shadow maps and the evaluation of candidate techniques is discussed in Chapter 3 [Peters and Klein 2015]. We then introduce moment shadow mapping and apply it to filtered hard shadows in Chapter 4 [Peters and Klein 2015]. Next, we discuss the three advanced applications of moment shadow mapping [Peters et al. 2016]. Shadows for translucent occluders are described in Chapter 5, soft shadows in Chapter 6 and single scattering in Chapter 7 [Peters et al. 2016].

All of our work on transient imaging and range imaging using AMCW lidar systems [Peters et al. 2015] is described in Chapter 8. Finally, we draw some conclusions and look at possible future work in Chapter 9.

The appendix provides some less important proofs and derivations (Appendix B) and describes additional implementation details including HLSL code listings (Appendix C). For easy reference, we also provide an index on page 211 and a nomenclature on page 213.

Chapter 2

# Moment Problems

Before we focus on specific moment problems in practical applications, it is useful to define them more rigorously and to derive some fundamental statements. In particular, we define different kinds of moments, discuss the solubility of corresponding moment problems and analyze their behavior in boundary cases.

It should be noted that none of this is to be considered as a contribution of our work. Almost all solutions to moment problems discussed in this thesis go back to the literature and adequate references are given throughout the text. The only notable exception is an alternative to moment shadow mapping named trigonometric moment shadow mapping that we discuss in Appendix B.4. Other than that, our contribution lies mostly in connecting this mathematical work to a variety of practical applications and in crafting tailor-made algorithms. We also rephrase some existing proofs in hopes of making the theory of moments more accessible to an audience of graphics researchers. Usually, we refer to the literature for proofs of existence and uniqueness but provide constructive proofs whenever they help the understanding of algorithms.

## 2.1 Moments

Moments provide information about finite measures. Such measures are the fundamental primitives of our work and all methods that we consider attempt to characterize them in one way or another. A measure $M$ assigns non-negative values to measurable[1] subsets $\mathbb{A}$ of a measurable space $\mathbb{X} \subseteq \mathbb{R}^d$.

[1] For a rigorous definition of measurable sets and functions we refer to the literature [Georgii 2008, Chapter 1].

The quantity $M(\mathbb{A})$ can be seen as a weighted, $d$-dimensional volume of $\mathbb{A}$ and is called the measure of $\mathbb{A}$. The measure $M$ is called finite if $M(\mathbb{X}) < \infty$. The precise definitions do not concern us because throughout this thesis only two special cases are truly relevant; measures with finite support and measures given by density functions.

**Definition 2.1** (Dirac-$\delta$ distribution)**.** Let $\mathbb{X} \subseteq \mathbb{R}^d$ and $x_0 \in \mathbb{X}$. The Dirac-$\delta$ distribution with support at $x_0$ is the finite measure $\delta_{x_0}$ defined by

$$\forall \mathbb{A} \subseteq \mathbb{X} : \; \delta_{x_0}(\mathbb{A}) := \begin{cases} 1 & \text{if } x_0 \in \mathbb{A}, \\ 0 & \text{otherwise.} \end{cases}$$

Integrating a function $\mathbf{a} : \mathbb{X} \to \mathbb{R}$ with respect to the measure $\delta_{x_0}$ means evaluating it at $x_0$:

$$\int \mathbf{a}(x) \, \mathrm{d}\delta_{x_0}(x) := \mathbf{a}(x_0)$$

We endow measures with vector-like operations, so a measure with support $w_0, \ldots, w_{n-1} > 0$ at the $n \in \mathbb{N}$ points $x_0, \ldots, x_{n-1} \in \mathbb{X}$ can be written as $M := \sum_{l=0}^{n-1} w_l \cdot \delta_{x_l}$. The corresponding integral of $\mathbf{a}$ is given by

$$\int \mathbf{a}(x) \, \mathrm{d}M(x) := \sum_{l=0}^{n-1} w_l \cdot \mathbf{a}(x_l).$$

**Definition 2.2** (Density function)**.** Let $\mathbb{X} \subseteq \mathbb{R}^d$ and let $D : \mathbb{X} \to \mathbb{R}_{\geq 0}$ be an integrable, non-negative function. The measure $M$ with density $D$ associates each measurable set $\mathbb{A} \subseteq \mathbb{X}$ with the integral

$$M(\mathbb{A}) := \int_{\mathbb{A}} D(x) \, \mathrm{d}x.$$

Correspondingly, the integral of a measurable function $\mathbf{a} : \mathbb{X} \to \mathbb{R}$ with respect to the measure $M$ is given by

$$\int \mathbf{a}(x) \, \mathrm{d}M(x) := \int_{\mathbb{X}} \mathbf{a}(x) \cdot D(x) \, \mathrm{d}x.$$

By means of finite measures, these two distinct cases are cleanly unified in one notation. At the same time, this notation opens up our work to an intuitive interpretation based on probability theory. Special finite measures are the foundation of this discipline.

**Definition 2.3** (Probability distribution)**.** Let $P$ be a finite measure over $\mathbb{X} \subseteq \mathbb{R}^d$. If $P(\mathbb{X}) = 1$, the measure $P$ is called a probability distribution. For a measurable set $\mathbb{A} \subseteq \mathbb{X}$, $P(\mathbb{A})$ is interpreted as the probability of the event $x \in \mathbb{A}$ and is also written as $P(\mathbf{x} \in \mathbb{A})$ where $\mathbf{x}(x) := x$. In general, a measurable function $\mathbf{a} : \mathbb{X} \to \mathbb{R}$ is called a random variable and its integral with respect to $P$ is called its expectation. We write

$$\mathcal{E}_P\left(\mathbf{a}\right) := \int \mathbf{a}(x)\,\mathrm{d}P(x),$$

i.e. integration and expectation are synonymous.

Note that $\frac{1}{M(\mathbb{X})} \cdot M$ is a probability distribution for any non-zero, finite measure $M$ on $\mathbb{X}$. Therefore, statements about probability distributions apply similarly to general finite measures. With these definitions at hand, we can now define precisely what we mean by general moments.

**Definition 2.4** (General moments)**.** Let $\mathbb{I} \subseteq \mathbb{R}$ and let $M$ be a finite measure over $\mathbb{I}$. Let $m \in \mathbb{N}$ and let $\mathbf{a}_1, \ldots, \mathbf{a}_m : \mathbb{I} \to \mathbb{R}$ be measurable functions. Then for $j \in \{1, \ldots, m\}$, the numbers

$$a_j := \int \mathbf{a}_j(x)\,\mathrm{d}M(x) \in \mathbb{R}$$

are called general moments of $M$. Let $\mathbf{a}_0(x) := 1$ for all $x \in \mathbb{I}$. The zeroth moment is defined by

$$a_0 := \int \mathbf{a}_0(x)\,\mathrm{d}M(x) = M(\mathbb{I}) \geq 0.$$

If $M$ is a probability distribution, $a_0 = 1$. Therefore, the number of used moments $m$ does not count the zeroth moment throughout our work. The vector $(a_0, \ldots, a_m)^\mathsf{T} \in \mathbb{R}^{m+1}$ is referred to as vector of general moments and the function $\mathbf{a} : \mathbb{I} \to \mathbb{R}^{m+1}$ with

$$\mathbf{a}(x) := (\mathbf{a}_0(x), \ldots, \mathbf{a}_m(x))^\mathsf{T}$$

is referred to as moment-generating function.

For the case of probability distributions, a general moment is simply the expectation of some random variable. Integration leads us from a measure to its general moments. Though, in the end we are more interested in going the other way, i.e. reconstructing the measure from its general moments. To answer when this is possible at all, we need means to tell when a vector in $\mathbb{R}^{m+1}$ is a vector of moments for some measure $M$.

**Proposition 2.5** (Solubility of general moment problems)**.** *A vector $a \in \mathbb{R}^{m+1}$ is a vector of general moments with respect to the moment-generating function $\mathbf{a} : \mathbb{I} \to \{1\} \times \mathbb{R}^m$ for some probability distribution $P$ on $\mathbb{I}$ if and only if $a$ lies in the convex hull* $\operatorname{conv} \mathbf{a}(\mathbb{I})$.

*Proof.* "⇒" Let $a = \mathcal{E}_P(\mathbf{a})$.

We only consider the case where $P$ has finite support. The general case can be reduced to this case [Mulholland and Rogers 1958, p. 178]. Let $n \in \mathbb{N}$, $x_0, \ldots, x_{n-1} \in \mathbb{I}$ and $w_0, \ldots, w_{n-1} \geq 0$ such that $P = \sum_{l=0}^{n-1} w_l \cdot \delta_{x_l}$. Then we know $\sum_{l=0}^{n-1} w_l = P(\mathbb{I}) = 1$ and thus

$$a = \mathcal{E}_P(\mathbf{a}) = \int \mathbf{a}(x)\, \mathrm{d}P(x) = \sum_{l=0}^{n-1} w_l \cdot \mathbf{a}(x_l) \tag{2.1}$$

is a convex combination of points in $\mathbf{a}(\mathbb{I})$, i.e. $a \in \operatorname{conv} \mathbf{a}(\mathbb{I})$.

"⇐" Let $a \in \operatorname{conv} \mathbf{a}(\mathbb{I})$.

By Caratheodory's theorem [Schrijver 1986, p. 94] $a$ can be written as a convex combination of points in $\mathbf{a}(\mathbb{I})$. This convex combination induces a probability distribution with $a = \mathcal{E}_P(\mathbf{a})$ as in Equation (2.1). □

General moments can be used to model many inverse problems. On the other hand, the use of an arbitrary moment-generating function limits us in deriving useful statements. Ultimately, we are interested in efficient algorithms that reconstruct information about measures given only some of their moments. To this end, we turn our attention towards very specific moment-generating functions, namely polynomials and Fourier basis functions.

**Definition 2.6** (Power moments)**.** Let $m \in \mathbb{N}$ be even. Let $\mathbf{b} : \mathbb{R} \to \mathbb{R}^{m+1}$ such that for all $j \in \{0, \ldots, m\}$ and $x \in \mathbb{R}$

$$\mathbf{b}_j(x) := x^j.$$

Let $M$ be a finite measure on $\mathbb{R}$. Then

$$b_j := \int \mathbf{b}_j(x)\, \mathrm{d}M(x) = \int x^j\, \mathrm{d}M(x) \in \mathbb{R}$$

is called the $j$-th power moment of $M$.

Of course, power moments are a special case of general moments and therefore Proposition 2.5 applies. However, the specific properties of the moment-generating function allow a far more specific criterion.

**Definition 2.7** (Hankel matrix)**.** Let $b \in \mathbb{R}^{m+1}$ with $m \in \mathbb{N}$ even. The associated Hankel matrix of $b$ is defined by

$$B(b) := (b_{j+k})_{j,k=0}^{\frac{m}{2}} = \begin{pmatrix} b_0 & b_1 & \cdots & b_{\frac{m}{2}} \\ b_1 & b_2 & \iddots & \vdots \\ \vdots & \iddots & \iddots & b_{m-1} \\ b_{\frac{m}{2}} & \cdots & b_{m-1} & b_m \end{pmatrix} \in \mathbb{R}^{(\frac{m}{2}+1)\times(\frac{m}{2}+1)}.$$

We define $\hat{\mathbf{b}} : \mathbb{R} \to \mathbb{R}^{\frac{m}{2}+1}$ with $\hat{\mathbf{b}}_j(x) := x^j$ for all $j \in \{0, \ldots, \frac{m}{2}\}$ and $x \in \mathbb{R}$ and call it the Hankel-matrix-generating function.

**Proposition 2.8** (Solubility of power moment problems [Akhiezer and Kreĭn 1962, p. 2 ff., 8 ff.])**.** *A vector $b \in \mathbb{R}^{m+1}$ admits a measure $M$ on $\mathbb{R}$ with $b = \int \mathbf{b}(x)\,\mathrm{d}M(x)$ if and only if $B(b)$ is positive semi-definite. Then*

$$B(b) = \int \hat{\mathbf{b}}(x) \cdot \hat{\mathbf{b}}^\mathsf{T}(x)\,\mathrm{d}M(x).$$

*Proof.* "⇒" Suppose $M$ is a measure on $\mathbb{R}$ such that $b = \int \mathbf{b}(x)\,\mathrm{d}M(x)$.

We observe that

$$B(b) = \int \begin{pmatrix} x^0 & x^1 & \cdots & x^{\frac{m}{2}} \\ x^1 & x^2 & \iddots & \vdots \\ \vdots & \iddots & \iddots & x^{m-1} \\ x^{\frac{m}{2}} & \cdots & x^{m-1} & x^m \end{pmatrix} \mathrm{d}M(x) = \int \hat{\mathbf{b}}(x) \cdot \hat{\mathbf{b}}^\mathsf{T}(x)\,\mathrm{d}M(x).$$

Now for an arbitrary $u \in \mathbb{R}^{\frac{m}{2}+1}$ it follows that

$$u^\mathsf{T} \cdot B(b) \cdot u = \int u^\mathsf{T} \cdot \hat{\mathbf{b}}(x) \cdot \hat{\mathbf{b}}^\mathsf{T}(x) \cdot u\,\mathrm{d}M(x) = \int \left(\sum_{j=0}^{\frac{m}{2}} u_j \cdot x^j\right)^2 \mathrm{d}M(x) \geq 0.$$

Thus, $B(b)$ is positive semi-definite.

"⇐" For the proof of sufficiency we refer to the literature [Akhiezer and Kreĭn 1962, p. 8 f., 11 f.]. □

Proposition 2.8 provides a concise and practical definition of power moments. If the Hankel matrix $B(b)$ is positive semi-definite, $b$ consists of power moments for some measure, otherwise it does not. It also introduces an important foundation for the theory of moments. The Hankel matrix $B(b)$ plays a crucial role whenever we solve a power moment problem.

Since we admit arbitrary finite measures on $\mathbb{R}$, the moment problem described above is referred to as Hamburger moment problem. In some situations, we are only interested in measures on some interval $\mathbb{I} \subset \mathbb{R}$. If this interval is half-open (e.g. $\mathbb{I} = [0, \infty)$), we are dealing with a Stieltjes moment problem. If it is compact (e.g. $\mathbb{I} = [-1, 1]$), the problem is referred to as Hausdorff moment problem (see Appendix B.3). Solubility criteria similar to the one in Proposition 2.8 exist for both cases [Kreĭn and Nudel'man 1977, p. 62 f., 175 f.].

Moment problems are studied similarly well for one other type of moments. This time, Fourier basis functions serve as moment-generating function.

**Definition 2.9** (Trigonometric moments)**.** For $m \in \mathbb{N}$ let $\mathbf{c} : \mathbb{R} \to \mathbb{C}^{m+1}$ such that for all $j \in \{0, \ldots, m\}$ and $x \in \mathbb{R}$

$$\mathbf{c}_j(x) := \exp(j \cdot i \cdot x) = \exp(i \cdot x)^j$$

where $i$ denotes the imaginary unit. Let $M$ be a finite measure on $\mathbb{R}$. Then

$$c_j := \int \mathbf{c}_j(x) \, \mathrm{d}M(x) = \int \exp(j \cdot i \cdot x) \, \mathrm{d}M(x) \in \mathbb{C}$$

is called the $j$-th trigonometric moment of $M$.

Essentially, trigonometric moments are Fourier coefficients of a measure. Equivalently, they may be seen as power moments of a measure on the complex unit circle. Therefore, they have a lot in common with power moments. The counterpart of the Hankel matrix is the Toeplitz matrix.

**Definition 2.10** (Toeplitz matrix)**.** Let $c \in \mathbb{C}^{m+1}$. For all $j \in \{1, \ldots, m\}$ let $c_{-j} := \overline{c_j}$ where $\overline{c_j}$ denotes the complex conjugate. The associated Toeplitz matrix of $c$ is defined by

$$C(c) := (c_{j-k})_{j,k=0}^{m} = \begin{pmatrix} c_0 & \overline{c_1} & \cdots & \overline{c_m} \\ c_1 & c_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \overline{c_1} \\ c_m & \cdots & c_1 & c_0 \end{pmatrix} \in \mathbb{C}^{(m+1)\times(m+1)}.$$

**Proposition 2.11** (Solubility of trigonometric moment problems [Kreĭn and Nudel'man 1977, p. 64 f.])**.** *A vector $c \in \mathbb{C}^{m+1}$ admits a measure $M$ on $\mathbb{R}$ with $c = \int \mathbf{c}(x) \, \mathrm{d}M(x)$ if and only if $C(c)$ is positive semi-definite. Then*

$$C(c) = \int \mathbf{c}(x) \cdot \mathbf{c}^*(x) \, \mathrm{d}M(x)$$

*where $\mathbf{c}^* = \overline{\mathbf{c}^\mathsf{T}}$ denotes the conjugate transpose.*

*Proof.* "⇒" Suppose $M$ is a measure on $\mathbb{R}$ such that $c = \int \mathbf{c}(x)\,\mathrm{d}M(x)$.

For all $x \in \mathbb{R}$ consider entry $j, k \in \{0, \ldots, m\}$ of the matrix $\mathbf{c}(x) \cdot \mathbf{c}^*(x)$:

$$\begin{aligned}(\mathbf{c}(x) \cdot \mathbf{c}^*(x))_{j,k} &= \mathbf{c}_j(x) \cdot \overline{\mathbf{c}_k}(x) \\ &= \exp(j \cdot i \cdot x) \cdot \overline{\exp(k \cdot i \cdot x)} \\ &= \exp(j \cdot i \cdot x) \cdot \exp(-k \cdot i \cdot x) \\ &= \exp((j-k) \cdot i \cdot x)\end{aligned}$$

We can use this equation to rewrite the Toeplitz matrix:

$$\begin{aligned}c_{j-k} &= \int (\mathbf{c}(x) \cdot \mathbf{c}^*(x))_{j,k}\,\mathrm{d}M(x) \\ \Rightarrow \;\; C(c) &= \int \mathbf{c}(x) \cdot \mathbf{c}^*(x)\,\mathrm{d}M(x)\end{aligned}$$

Now for all $u \in \mathbb{C}^{m+1}$ we get

$$u^* \cdot C(c) \cdot u = \int (u^* \cdot \mathbf{c}(x)) \cdot (\mathbf{c}^*(x) \cdot u)\,\mathrm{d}M(x) = \int |u^* \cdot \mathbf{c}(x)|^2\,\mathrm{d}M(x) \geq 0.$$

"⇐" For the case with positive-definite $C(c)$, we provide a constructive proof in Appendix B.5. For the general case (including singular $C(c)$), we refer to the literature [Kreĭn and Nudel'man 1977, p. 64 f.]. □

It is interesting to note that this proof is completely analogous to the proof of Proposition 2.8. The reason for the differences between the Hankel matrix $B(b)$ and the Toeplitz matrix $C(c)$ is simply that complex conjugation turns the entries of $\mathbf{c}$ into their reciprocal whereas it does nothing at all to the entries of $\mathbf{b}$.

## 2.2 Boundary Cases

It is natural to ask what happens as the Hankel or Toeplitz matrix stops being positive semi-definite? Analyzing this boundary case turns out to be very fruitful. It provides us with the first algorithms that actually solve moment problems, albeit only in special cases, and simultaneously reveals the greatest strength of the theory of moments with respect to the applications at hand.

We find that the boundary case is present whenever the underlying measure is sufficiently sparse. In this case, the ground truth can be reconstructed perfectly. The following proposition makes this claim more precise.

**Proposition 2.12** (The boundary case for power moment problems [Kreĭn and Nudel'man 1977, p. 63, 78])**.** *Let $b \in \mathbb{R}^{m+1}$ such that $B(b)$ is positive semi-definite. The following statements are equivalent:*

1. *$B(b)$ is singular,*
2. *There exists exactly one measure $M$ on $\mathbb{R}$ with $b = \int \mathbf{b}(x)\,\mathrm{d}M(x)$,*
3. *A measure of the form $M := \sum_{l=0}^{\frac{m}{2}-1} w_l \cdot \delta_{x_l}$ with $x_0, \ldots, x_{\frac{m}{2}-1} \in \mathbb{R}$ and $w_0, \ldots, w_{\frac{m}{2}-1} > 0$ exists such that $b = \int \mathbf{b}(x)\,\mathrm{d}M(x)$.*

*Suppose $B(b)$ is singular and let $q \in \ker B(b)$ with $q \neq 0$. Then $x_0, \ldots, x_{\frac{m}{2}-1}$ are roots of the polynomial $\sum_{j=0}^{\frac{m}{2}} q_j \cdot x^j$.*

*Proof.* "1. ⇒ 3. and 2." Let $q \in \ker B(b)$ with $q \neq 0$.

By Proposition 2.8 there exists a measure $M$ with $b = \int \mathbf{b}(x)\,\mathrm{d}M(x)$ and we can use it to represent the Hankel matrix $B(b)$:

$$0 = q^{\mathsf{T}} \cdot B(b) \cdot q = \int q^{\mathsf{T}} \cdot \hat{\mathbf{b}}(x) \cdot \hat{\mathbf{b}}^{\mathsf{T}}(x) \cdot q\,\mathrm{d}M(x) = \int \left( \sum_{j=0}^{\frac{m}{2}} q_j \cdot x^j \right)^2 \mathrm{d}M(x) \tag{2.2}$$

The integrand $\left(\sum_{j=0}^{\frac{m}{2}} q_j \cdot x^j\right)^2$ is non-negative. Furthermore, $\sum_{j=0}^{\frac{m}{2}} q_j \cdot x^j$ is a non-zero polynomial of degree $\frac{m}{2}$ or less and cannot have more than $\frac{m}{2}$ roots. Since the integral evaluates to zero, $M$ must have all of its support at these roots which proves 3. and the claim about the locations of $x_0, \ldots, x_{\frac{m}{2}-1}$.

Equation (2.2) uniquely determines the locations of the points of support $x_0, \ldots, x_{\frac{m}{2}-1} \in \mathbb{R}$. Suppose, the first $n \in \{1, \ldots, \frac{m}{2}\}$ of these points of support are distinct. Then the system of linear equations

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_0^1 & x_1^1 & \cdots & x_{n-1}^1 \\ \vdots & \vdots & & \vdots \\ x_0^{n-1} & x_1^{n-1} & \cdots & x_{n-1}^{n-1} \end{pmatrix} \cdot \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{n-1} \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{pmatrix}$$

uniquely determines the corresponding weights $w_0, \ldots, w_{n-1}$ because the matrix in this system is a square Vandermonde matrix constructed from pairwise different values. Also, these weights have to be non-negative because otherwise this would contradict existence of $M$. Thus, we have proven 2..

"3. ⇒ 1." Let $M = \sum_{l=0}^{\frac{m}{2}-1} w_l \cdot \delta_{x_l}$ such that $b = \int \mathbf{b}(x)\,\mathrm{d}M(x)$.

We note that the matrix $\hat{\mathbf{b}}(x_l) \cdot \hat{\mathbf{b}}^\mathsf{T}(x_l)$ has rank one for all $l \in \{0, \ldots, \frac{m}{2} - 1\}$. Thus, the rank of

$$B(b) = \int \hat{\mathbf{b}}(x) \cdot \hat{\mathbf{b}}^\mathsf{T}(x)\,\mathrm{d}M(x) = \sum_{l=0}^{\frac{m}{2}-1} w_l \cdot \hat{\mathbf{b}}(x_l) \cdot \hat{\mathbf{b}}^\mathsf{T}(x_l)$$

cannot be greater than $\frac{m}{2}$. It follows that $B(b) \in \mathbb{R}^{(\frac{m}{2}+1)\times(\frac{m}{2}+1)}$ is singular.

"¬1. ⇒ ¬2." Suppose $\det B(b)$ is positive definite.

Let $M$ be a measure on $\mathbb{R}$ with $b = \int \mathbf{b}(x)\,\mathrm{d}M(x)$. Let $x_0 \in \mathbb{R}$ such that $M(\{x_0\}) = 0$, i.e. $M$ does not have support at $x_0$. There exists an $\varepsilon > 0$ such that $B(b - \varepsilon \cdot \mathbf{b}(x_0))$ is still positive semi-definite. Let $N$ be a measure on $\mathbb{R}$ with $b - \varepsilon \cdot \mathbf{b}(x_0) = \int \mathbf{b}(x)\,\mathrm{d}N(x)$. Then

$$b = \int \mathbf{b}(x)\,\mathrm{d}M(x) = \int \mathbf{b}(x)\,\mathrm{d}N(x) + \varepsilon \cdot \mathbf{b}(x_0) = \int \mathbf{b}(x)\,\mathrm{d}(N + \varepsilon \cdot \delta_{x_0})(x)$$

and thus we have constructed two different measures representing the vector of power moments $b$. □

While Proposition 2.12 may seem rather abstract, its relevance for our work is tremendous. It tells us exactly in which situation the boundary case $\det B(b) = 0$ occurs, it tells us that there is a unique measure corresponding to $b$ and it even tells us how to reconstruct it perfectly. The algorithm is implicit in the proof and we summarize it in Algorithm 2.1. When we apply the theory of moments to shadow mapping, this boundary case is relevant because it is very common that a filter region in a shadow map only touches one or two shadow casting surfaces. Whenever this is the case, the distribution of depth values can be approximated well using two points of support and we are close to the boundary case for $m = 4$. The reconstruction will not be perfect but highly accurate.

Again a completely analogous result, which is just as relevant, can be obtained for trigonometric moments.

**Proposition 2.13** (Boundary case for trigonometric moment problems [Kreĭn and Nudel'man 1977, p. 65, 78])**.** *Let $c \in \mathbb{C}^{m+1}$ such that $C(c)$ is positive semi-definite. The following statements are equivalent:*

1. *$C(c)$ is singular,*
2. *There exists exactly one measure $M$ on $(0, 2 \cdot \pi]$ with $c = \int \mathbf{c}(x)\,\mathrm{d}M(x)$,*

**Algorithm 2.1** Perfect reconstruction of a measure from power moments in the boundary case.
**Input:** $b \in \mathbb{R}^{m+1}$ such that $B(b)$ is positive semi-definite but singular.
**Output:** The unique measure $M$ with $b = \int \mathbf{b}(x) \, \mathrm{d}M(x)$.

```
1. Compute $q \in \ker B(b)$ with $q \neq 0$.
2. Solve $\sum_{j=0}^{\frac{m}{2}} q_j \cdot x^j = 0$ for $x$ to obtain all $n \in \{1, \ldots, \frac{m}{2}\}$ pairwise
   different, real roots $x_0, \ldots, x_{n-1} \in \mathbb{R}$.
3. Solve the system of linear equations
   $$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_0^1 & x_1^1 & \cdots & x_{n-1}^1 \\ \vdots & \vdots & & \vdots \\ x_0^{n-1} & x_1^{n-1} & \cdots & x_{n-1}^{n-1} \end{pmatrix} \cdot \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{n-1} \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{pmatrix}.$$
4. Return $M := \sum_{l=0}^{n-1} w_l \cdot \delta_{x_l}$.
```

3. *There exists* $x_0, \ldots, x_{m-1} \in (0, 2 \cdot \pi]$ *and* $w_0, \ldots, w_{m-1} > 0$ *such that* $M := \sum_{l=0}^{m-1} w_l \cdot \delta_{x_l}$ *yields* $c = \int \mathbf{c}(x) \, \mathrm{d}M(x)$.

*Suppose* $C(c)$ *is singular and let* $q \in \ker C(c)$ *with* $q \neq 0$. *Then* $x_0, \ldots, x_{m-1}$ *are solutions of the equation* $q^* \cdot \mathbf{c}(x) = 0$.

*Proof.* The proof is completely analogous to the proof of Proposition 2.12. For completeness we provide it in Appendix B.6 nonetheless. □

For shadow mapping, we end up using power moments rather than trigonometric moments and Proposition 2.13 is not very relevant. However, it is extremely relevant for transient imaging. In many application scenarios, the indirect returns observed by an AMCW lidar system are dominated by specular interreflections. Thus, the impulse responses are sparse. Unless the number of returns is greater than the number of acquired trigonometric moments, we can expect a nearly perfect reconstruction. In fact, this works so well that Proposition 2.13 gives rise to one of the two reconstruction methods that we use for transient imaging. It is known as Pisarenko estimate and we describe it in Algorithm 8.2 on page 131 which is analogous to Algorithm 2.1.

## 2.3 Overview of Moment Problems and Solutions

We now turn our attention to more general instances of moment problems. By moment problems we do not mean a single well-defined problem but a rather big and diverse class of problems. Their common goal is to derive some statement about the set of measures that match a given sequence of moments. Some literature deals with infinite sequences of power moments or trigonometric moments [Kreĭn and Nudel'man 1977, p. 64, 66]. Moment problems dealing with a finite number of moments, are sometimes referred to as truncated moment problems.

Throughout this thesis, we employ solutions to a variety of truncated moment problems. They will be described in the context of their respective applications. As an outline and for reference, we now provide an overview of all described solutions to moment problems including some historical comments on their origins.

**Criteria for Solubility** It is natural to begin the discussion of moment problems with the question of solubility as we did above. If a given sequence of moments does not match any measure, there is nothing more to say about the set of matching measures. Criteria for solubility of the general, power and trigonometric moment problem are given in Propositions 2.5, 2.8 and 2.11, respectively.

**Construction of Canonical Representations** Simply put, a canonical representation of a sequence of power or trigonometric moments is a matching measure with support at a minimal number of points [Kreĭn and Nudel'man 1977, p. 77]. In some contexts these representations are of interest by themselves but more often they serve as a tool in the construction of other quantities. For the boundary cases, we have seen their construction in Propositions 2.12 and 2.13. In the general case, there is a whole family of canonical representations. They are uniquely determined, once a single point of support is prescribed. We describe their construction for power moments in Section 4.1.1 and for trigonometric moments in Appendix B.4.

**Chebyshev-Markov Inequalities** Chebyshev-Markov inequalities provide sharp upper and lower bounds for the cumulative distribution function of a matching measure. For general moments, this can be solved numerically as demonstrated in Section 3.2.1. For power moment problems, we find in Section 4.1.1 that the bounds are always realized by canonical representations

[Kreĭn and Nudel'man 1977, p. 125]. For trigonometric moment problems the solution is more complicated. Canonical representations are part of it but do not fully solve the problem. For the case $m = 2$, we present a solution in Appendix B.4, which is the only truly novel mathematical result in this thesis and the underlying publications.

These inequalities are at the core of all of our novel shadow mapping techniques. The results that we utilize go back to the work of P. L. Chebyshev [Tchebichef 1874]. The proofs for his work and some generalizations were completed by his student A. A. Markov [Markov 1884]. Later the theory was further extended and formalized by M. G. Kreĭn, A. A. Nudel'man and N. I. Akhiezer [Akhiezer and Kreĭn 1962; Akhiezer 1965; Kreĭn and Nudel'man 1977] as well as S. Karlin and W. J. Studden [Karlin and Studden 1966]. These later works are the primary source for our work.

**Reconstruction from Three Power Moments** In Section 7.2, we will find use for a simple method that reconstructs a probability distribution with two points of support from three power moments $b_1, b_2, b_3$.

**Maximum Entropy Spectral Estimate** For transient imaging we do not work with conservative estimates and we are primarily interested in the density of the measure. The maximum entropy spectral estimate, which we introduce in Section 8.2.2, reconstructs a density function that matches all given trigonometric moments and simultaneously minimizes the so called Burg entropy. This prior serves to ensure that the reconstruction does not produce features that are not warranted by the given measurements. The cumulative distribution function is also efficiently evaluable (Section 8.4.4). The Burg entropy and the maximum entropy spectral estimate were introduced by J. P. Burg during his doctoral studies. Our primary sources are his Ph.D. thesis [Burg 1975] and an article that reformulates some of the proofs [Landau 1987].

**Pisarenko Estimate** The Pisarenko estimate is the counterpart of Algorithm 2.1 for trigonometric moments and we provide a detailed discussion in Section 8.4.1 and Appendix B.6.

**Uncertainty Bounds** In general, the maximum entropy spectral estimate will not provide a perfect reconstruction of the ground truth. Still, we can provide rather strong guarantees on the quality of the approximation. To this end we utilize work by Karlsson and Georgiou [2013], which provides sharp upper and lower bounds for the density of matching measures after application of a specific smoothing kernel.

# Part I

# Moment Shadow Mapping

Chapter 3

# Filterable Shadow Maps

Having fully dynamic shadows in real-time applications is commonplace nowadays. Unfortunately, these shadows are a frequent source of artifacts. A widely used technique is percentage-closer filtering [Reeves et al. 1987] where a common shadow map is sampled in a neighborhood of the relevant fragment. A shadow intensity is computed from each individual sample and the results are filtered to diminish shadow map aliasing. This way, a bandlimiting filter is applied during reconstruction of the shadow signal at the cost of many texture reads per shaded fragment.

Percentage-closer filtering provides no remedies for aliasing that arises during initial sampling of the shadow map [Eisemann et al. 2011, p. 75 ff.]. Shadows are binary in nature and thus discontinuous. From a signal-theoretic standpoint, this means that they contain arbitrarily high frequencies and with common shadow maps there is no way to eliminate these high frequencies before sampling. In consequence, the sampling rate is never high enough. The best option is to ensure a constantly high sampling rate throughout the scene coupled with a large filter for percentage-closer filtering. However, dense sampling is hard to accomplish since the shadow map is rendered in light space but sampled in screen space and larger filters make the sampling procedure more expensive.

For textures that admit linear filtering, powerful solutions to these problems are implemented in graphics hardware. When rendering a scene to a texture, multisample antialiasing serves to diminish high-frequency signal components. Afterwards, further low-pass filters (e.g. a two-pass Gaussian blur) can be applied efficiently. During sampling bilinear filtering, mipmapping and anisotropic filtering are available. If they were applicable to shadow maps, these features could diminish aliasing tremendously.

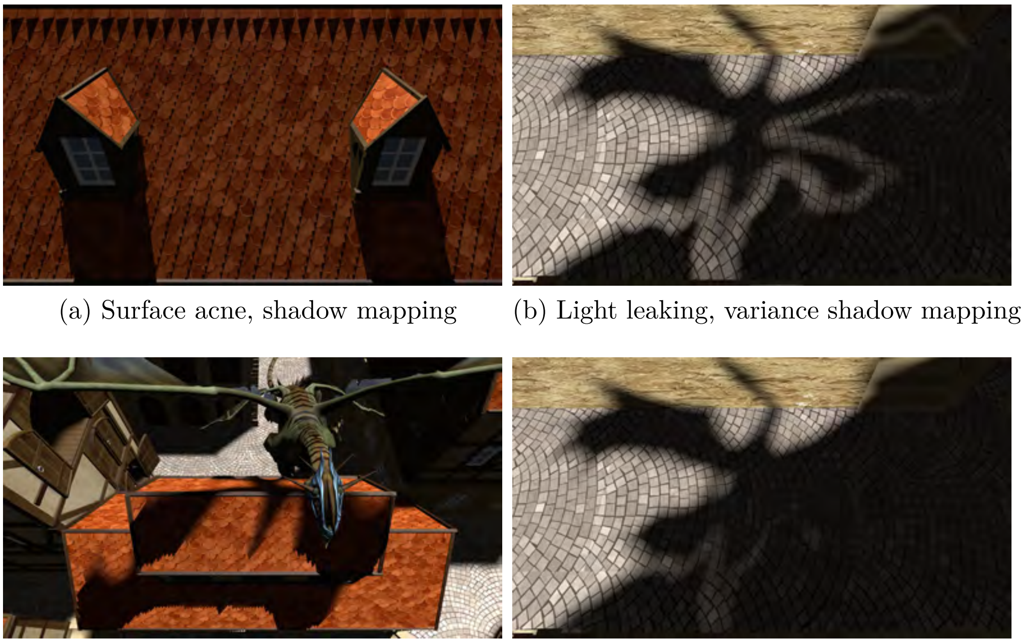
Filterable shadow maps offer just that. Following the introduction of variance shadow mapping [Donnelly and Lauritzen 2006], a variety of such techniques has been proposed. All of them provide an equally good solution to the problem of aliasing but introduce various new artifacts such as light leaking or ringing. In the present chapter, we provide a generalized view onto these works that culminates in the systematic evaluation of 66045 potential new techniques. The most promising candidates are turned into practical new shadow mapping techniques in Chapter 4, followed by a variety of applications in Chapters 5, 6 and 7.

## 3.1 Related Work

The prevalent approaches for rendering dynamic, hard shadows are based on ray tracing, shadow volumes or shadow mapping. Monte-Carlo ray tracing is the standard approach in offline rendering and yields accurate hard or soft shadows for arbitrary light sources [Cook et al. 1984]. For real-time applications it used to be prohibitively expensive. Recent works use shadow maps as basis for acceleration structures to make real-time ray tracing of hard or soft shadows practical but the cost remains comparatively high [Wang et al. 2014; Wyman et al. 2015]. Shadow volumes [Crow 1977] enable pixel perfect hard shadows by rasterizing geometry for the shadow boundaries. Extensions to soft shadows exist [Akenine-Möller and Assarsson 2002]. However, all these techniques scale poorly with scene complexity.

Currently, shadow mapping [Williams 1978] serves as basis for the vast majority of practically relevant techniques. A shadow map is a texture rendered from the point of view of the light source, i.e. each view ray of the used camera corresponds to a light ray of the light source. Each texel stores the depth of the first opaque surface thus describing the segment of the light ray that is lit. A single sample reveals whether an arbitrary point is lit. Besides the aliasing problems discussed above, shadow mapping is prone to one more artifact known as surface acne. Since the surface samples available from the shadow map will always be slightly offset with respect to the surface samples on screen, a simple comparison of depth values will often falsely classify a surface sample as shadowed (Fig. 3.1a). This can be avoided by subtracting a depth bias from the fragment depth but excessive biasing leads to missing shadows at contact points. Normal vectors help to determine the bias adaptively [Dou et al. 2014].

In any case the sampling of the scene in the shadow map should closely resemble the sampling on screen to avoid undersampling and oversampling.



(a) Surface acne, shadow mapping

(b) Light leaking, variance shadow mapping

(c) Missing shadows at boundaries, exponential shadow mapping, $c_{\mathrm{esm}} = 80$

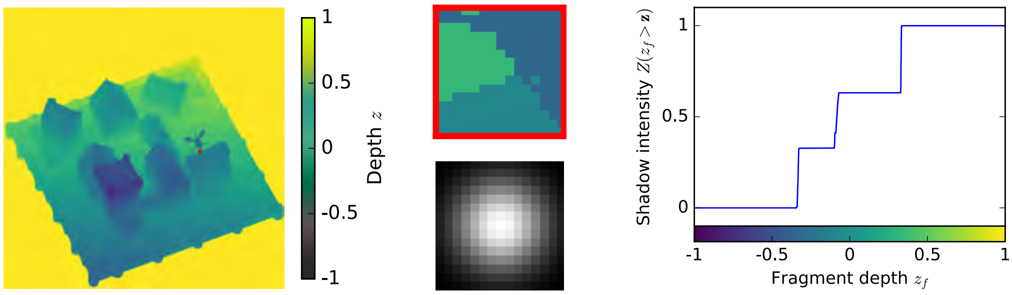
(d) Light leaking, exponential variance shadow mapping, $c_{\mathrm{evsm}}^{-} = c_{\mathrm{evsm}}^{+} = 5.54$

Figure 3.1: Failure cases of various shadow mapping techniques. Each example is chosen to provoke a specific artifact.

Light space perspective shadow maps [Martin and Tan 2004; Wimmer et al. 2004] optimize a single perspective transform based on the view frustum. Parallel-split shadow maps [Zhang et al. 2006] subdivide the view frustum along multiple planes and use one shadow map per resulting cascade. Sample distribution shadow maps [Lauritzen et al. 2011] analyze the screen space depth buffer to place these planes optimally. Virtual shadow maps for many point lights [Olsson et al. 2014, 2015] use omni-directional shadow maps and decide which part of which shadow map needs to be rendered at which resolution per frame. All these techniques greatly diminish problems with aliasing but still filtering is needed for visually pleasing results.

Percentage-closer filtering [Reeves et al. 1987] is the reference for shadow map filtering and we will now interpret it in a probabilistic framework. The technique takes $n \in \mathbb{N}$ samples $z_0, \ldots, z_{n-1} \in [-1, 1]$ from a neighborhood of the relevant fragment in the shadow map[1] and weights each of them

[1] We follow the convention that the near-clipping plane of the shadow map projection corresponds to $z = -1$ and the far-clipping plane corresponds to $z = 1$. The practical advantages of this choice are elaborated in Section 4.1.5.



(a) Shadow map

(b) Filter region and kernel

(c) Shadow intensity graph (i.e. cumulative distribution function)

Figure 3.2: A visualization of percentage-closer filtering. The depth-dependent shadow intensity (3.2c) coincides with the cumulative distribution function of the depth distribution $Z$. This distribution is constructed from the depth values in the filter region, weighted by a filter kernel (3.2b). The depth samples stem from the shadow map (3.2a). Note how the three distinct surfaces in the filter region correspond to the three steep increases of the shadow intensity.

with a weight $w_0, \ldots, w_{n-1} \in [0, 1]$ based on a low-frequent filter kernel. Given the biased depth of the shaded fragment $z_f \in [-1, 1]$, it then turns each depth into a shadow intensity and filters these. We define the depth distribution within the filter region as

$$Z := \sum_{l=0}^{n-1} w_l \cdot \delta_{z_l}. \tag{3.1}$$

Using the random variable $\mathbf{z}(z) := z$ we find that the filtered shadow-intensity can be written as

$$Z(z_f > \mathbf{z}) = \sum_{l=0}^{n-1} w_l \cdot \begin{cases} 0 & \text{if } z_f \leq z_l, \\ 1 & \text{if } z_f > z_l. \end{cases}$$

Thus, percentage-closer filtering evaluates the cumulative distribution function of the depth distribution within the filter region (see Figure 3.2).

To avoid sampling filter regions in the shadow map per fragment, a compact representation of the depth distribution $Z$ needs to be precomputed. Deep shadow maps [Lokovic and Veach 2000] construct the cumulative distribution function of $Z$ explicitly and then approximate it by a piecewise linear function while maintaining quality guarantees. This does provide a compact, precomputed representation of $Z$ but this representation still cannot be filtered directly because it depends on $Z$ in a non-linear manner.

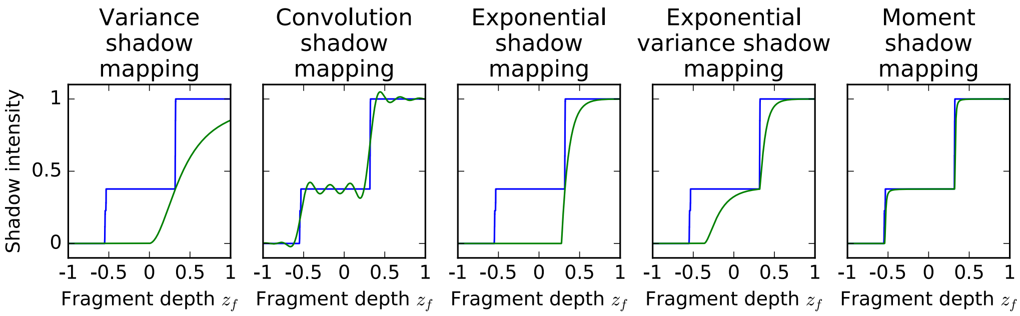


Figure 3.3: The shadow intensity produced by various types of filterable shadow maps (green) next to the ground truth produced by percentage-closer filtering (blue). The parameters used for the techniques correspond to the lower-quality option in Table 3.1. In particular, $c_{\text{esm}} = 11.09$, $c^+_{\text{evsm}} = c^-_{\text{evsm}} = 5.54$ and $\alpha_b = 6 \cdot 10^{-5}$ (see Section 4.1.4).

Variance shadow maps [Donnelly and Lauritzen 2006] offer a linear representation. Since they can be filtered directly, we refer to them as filterable shadow maps. Unlike common shadow maps, variance shadow maps have two channels. The first one stores the depth $z$ like a common shadow map but the second one stores $z^2$. Initially this information is redundant but if we filter the variance shadow map within our filter region, we obtain two power moments $b_1 = \mathcal{E}_Z(\mathbf{z})$ and $b_2 = \mathcal{E}_Z(\mathbf{z}^2)$. Additionally, we know $b_0 = 1$ because the filter weights are normalized. The two power moments correspond to the mean $\mu := b_1$ and variance $\sigma^2 := b_2 - b_1^2$ of the depth distribution. By Cantelli's inequality (referred to as one-tailed version of Chebyshev's inequality by Donnelly and Lauritzen), we know for all $z_f > \mu$

$$Z(z_f > \mathbf{z}) = 1 - Z(\mathbf{z} - \mu \geq z_f - \mu) \geq 1 - \frac{\sigma^2}{\sigma^2 + (z_f - \mu)^2}.$$

This lower bound is used as approximation to the shadow intensity. Failure cases arise when $\sigma$ is large because then the lower bound converges to one slowly such that light leaks through occluders (Figs. 3.1b and 3.3). Layered variance shadow maps [Lauritzen and McCool 2008] avoid this case by using multiple variance shadow maps for subintervals of the depth interval $[-1, 1]$.

Convolution shadow mapping [Annen et al. 2007] approximates the cumulative distribution function by a truncated Fourier series. Just like the power moments, the Fourier coefficients can be filtered linearly. However, a large number of coefficients is needed for an acceptable approximation, especially in large scenes and ringing is a problem (Fig. 3.3).

Exponential shadow maps [Salvi 2008; Annen et al. 2008b] approximate the shadow intensity by an exponential function that is scaled to be very steep using a fixed factor $c_{\text{esm}} \gg 1$ (e.g. $c_{\text{esm}} = 80$). The exponential shadow map stores $\mathcal{E}_Z(\exp(c_{\text{esm}} \cdot \mathbf{z}))$ and then by Markov's inequality

$$Z(z_f \geq \mathbf{z}) = 1 - Z(\exp(c_{\text{esm}} \cdot \mathbf{z}) \geq \exp(c_{\text{esm}} \cdot z_f)) \geq 1 - \frac{\mathcal{E}_Z(\exp(c_{\text{esm}} \cdot \mathbf{z}))}{\exp(c_{\text{esm}} \cdot z_f)}.$$

This works very well for the hindmost receiver of partial shadow and for all receivers of full shadow but the bound is meaningless for smaller values of $z_f$ (Fig. 3.3). Therefore, artifacts occur at boundaries of shadow casters (Fig. 3.1c).

Exponential variance shadow maps [Lauritzen and McCool 2008] combine exponential shadow mapping and variance shadow mapping to cancel out the complementary weaknesses of these techniques. For fixed $c^+_{\text{evsm}}, c^-_{\text{evsm}} > 1$ they store

$$\begin{gathered}\mathcal{E}_Z\left(\exp(c^+_{\text{evsm}} \cdot \mathbf{z})\right) \text{ and } \mathcal{E}_Z\left(\exp(c^+_{\text{evsm}} \cdot \mathbf{z})^2\right) \text{ as well as}\\ \mathcal{E}_Z\left(-\exp(-c^-_{\text{evsm}} \cdot \mathbf{z})\right) \text{ and } \mathcal{E}_Z\left(\exp(-c^-_{\text{evsm}} \cdot \mathbf{z})^2\right).\end{gathered}$$

Having two power moments for the random variables $\exp(c^+_{\text{evsm}} \cdot \mathbf{z})$ and $-\exp(-c^-_{\text{evsm}} \cdot \mathbf{z})$, variance shadow mapping is used to compute two lower bounds to the shadow intensity and the larger one functions as approximation. This technique produces decent filtered hard shadows in most cases. Though, some light leaking remains, especially when $c^+_{\text{evsm}}$ is chosen small enough to enable the use of half-precision floating point textures (Figs. 3.1d and 3.3).

Since bandwidth is the limiting factor, the run time of all sorts of filterable shadow maps is primarily determined by the amount of memory that is used per texel in the shadow map, which is in turn dependent on the quality requirements. An overview of typical values is given in Table 3.1.

Applications of filterable shadow maps are not restricted to filtered hard shadows. For example, variance shadow maps have been used for approximations to screen-space ambient occlusion [Loos and Sloan 2010]. The same goal is accomplished with first-order moments when different surfaces are separated [Hendrickx et al. 2015]. Shadows for translucent occluders can be rendered efficiently as described in Section 5.1. Filtered hard shadows resemble soft shadows and in Section 6.1 we discuss techniques exploiting this fact for approximate soft shadows. Even single scattering due to directional lights in homogeneous participating media can be accelerated using filterable shadow maps as shown in Section 7.1.1.

| Technique | Lower quality | Higher quality |
|---|---|---|
| Variance shadow mapping | $2 \cdot 16 = 32$ | $2 \cdot 32 = 64$ |
| Layered variance shadow mapping | $8 \cdot 16 = 128$ | $16 \cdot 16 = 256$ |
| Convolution shadow mapping | $16 \cdot 8 = 128$ | $32 \cdot 8 = 256$ |
| Exponential shadow mapping | $1 \cdot 16 = 16$ | $1 \cdot 32 = 32$ |
| Exponential variance shadow mapping | $4 \cdot 16 = 64$ | $4 \cdot 32 = 128$ |
| Moment shadow mapping (ours) | $4 \cdot 16 = 64$ | $4 \cdot 32 = 128$ |

Table 3.1: The typical memory consumption in bits per texel for various kinds of filterable shadow maps in two typical configurations. The formulas are to be understood as "number of channels · bits per scalar = total bits per texel". For layered variance shadow mapping and convolution shadow mapping other configurations are practical.

## 3.2 Generalized Filterable Shadow Maps

The common ground of all filterable shadow maps is that they take the depth of the shadow casting geometry as input for a fixed, vector-valued function to determine what to store in the shadow map. Let us say that this function produces a vector with $m \in \mathbb{N}$ dimensions and denote the component-functions by $\mathbf{a}_1, \ldots, \mathbf{a}_m : [-1, 1] \to \mathbb{R}$. Then what a filterable shadow map really stores after filtering, are the general moments of the depth distribution in Equation (3.1)

$$a_j = \mathcal{E}_Z(\mathbf{a}_j) \quad \text{for} \quad j \in \{1, \ldots, m\}.$$

Since we assume a filter with normalized weights, we additionally know $a_0 = \mathcal{E}_Z(1) = 1$. The filterable shadow map is fully characterized by its moment-generating function

$$\mathbf{a} : [-1, 1] \to \{1\} \times \mathbb{R}^m \quad \text{with} \quad \mathbf{a}(z) := (1, \mathbf{a}_1(z), \ldots, \mathbf{a}_m(z))^\mathsf{T}.$$

Going from a depth-distribution $Z$ to its compact representation in the shadow map $a := \mathcal{E}_Z(\mathbf{a})$ works the same for each possible choice of $\mathbf{a}$. The inverse problem of reconstructing $Z$ from $a$ is far less trivial. Solutions employed by the related work are diverse. However, most of them share one crucial property: They may underestimate the shadow intensity but they will never overestimate it. This is true for variance, layered variance, exponential and exponential variance shadow mapping. Convolution shadow mapping does not guarantee it by default but the biasing employed in practice serves to avoid overestimation of the shadow intensity.

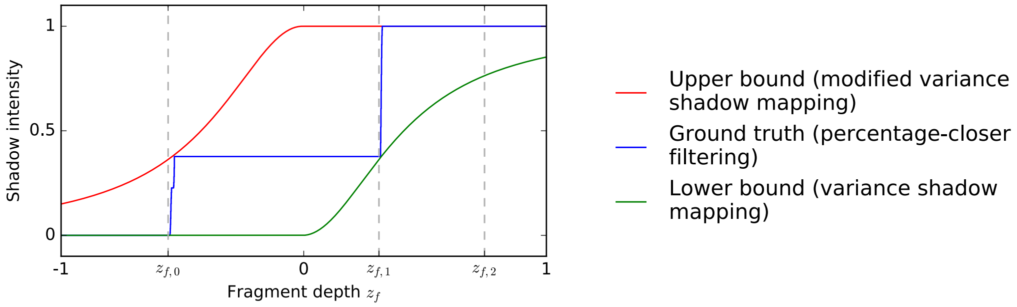


Figure 3.4: An example demonstrating why it is preferable to underestimate the shadow intensity. Although the whole graph is shown, the shadow intensity is only evaluated at three surfaces at biased depth $z_{f,0}$, $z_{f,1}$ and $z_{f,2}$. The surfaces at $z_{f,0}$ and $z_{f,1}$ coincide with increases in shadow intensity because they are visible in the shadow map. For them the lower bound is in close agreement with the ground truth. On the other hand, the upper bound overestimates the shadow intensity in a manner that would lead to wrong self-shadowing. Between $z_{f,0}$ and $z_{f,1}$ the lower bound provides a very poor approximation but is never used. Only at $z_{f,2}$ the approximation error of the lower bound will lead to visible light leaking.

There are two motivations behind this design decision: It avoids surface acne and it tends to produce strong approximation errors in regions where the shadow intensity is never evaluated. As an example, we consider Figure 3.4. It demonstrates that overestimation of shadow intensities is likely to cause wrong self-shadowing while underestimation avoids it. Even if $z_{f,0}$ and $z_{f,1}$ were slightly shifted to the right, variance shadow mapping would still provide a correct result whereas percentage-closer filtering would lead to surface acne. While this is just one example, it is representative of a very common situation. Whenever the filter region overlaps the silhouette of one shadow caster, it will cover two surfaces and yield a similar depth distribution. In fact, this case is so common that Donnelly and Lauritzen [2006] proved that variance shadow mapping produces the correct result at $z_{f,0}$ and $z_{f,1}$.

The downside is the light leaking found at $z_{f,2}$. Although we want to underestimate the shadow intensity, we do not want to underestimate it more than necessary. Given all the knowledge that we have, it should be at least possible that the ground truth agrees with the lower bound. This simple requirement immediately leads to a well-defined reconstruction technique for each and every moment-generating function. To implement it, we have to solve the following problem.

**Problem 3.1** (Chebyshev-Markov inequality for general moments [Kreĭn and Nudel'man 1977, p. 118 ff.])**.** Suppose we know that all shadow map depth values lie in a set $\mathbb{I} \subseteq \mathbb{R}$ and for some moment-generating function $\mathbf{a} : \mathbb{I} \to \{1\} \times \mathbb{R}^m$ we are given the vector of general moments $a := \mathcal{E}_Z(\mathbf{a})$ of an unknown depth distribution $Z$ on $\mathbb{I}$. Let $\mathbb{P}(\mathbb{I})$ denote the set of probability distributions on $\mathbb{I}$. For a given $z_f \in \mathbb{R}$ compute the optimal lower bound

$$G_{\mathbb{I},\mathbf{a}}(a, z_f) := \inf_{\substack{S \in \mathbb{P}(\mathbb{I}) \\ \mathcal{E}_S(\mathbf{a}) = a}} S(z_f > \mathbf{z}).$$

We note that this problem has a well-defined solution if and only if $a \in \operatorname{conv} \mathbf{a}(\mathbb{I})$ because by Proposition 2.5 this is necessary and sufficient for distributions $S$ on $\mathbb{I}$ with $\mathcal{E}_S(\mathbf{a}) = a$ to exist. If indeed $a = \mathcal{E}_Z(\mathbf{a})$, this is guaranteed. Then the optimal lower bound will be a value in [0, 1] with

$$G_{\mathbb{I},\mathbf{a}}(a, z_f) \leq Z(z_f > \mathbf{z}).$$

Thus, it is indeed a lower bound for the shadow intensity. At the same time, it is the sharpest possible lower bound because given our knowledge about $Z$ we cannot rule out the possibility that the optimal $S$ equals $Z$.

Choosing $\mathbb{I}$ as proper subset of $\mathbb{R}$ introduces apriori knowledge into the approximation that leads to a sharper lower bound. With respect to our conventions $\mathbb{I} = [-1, 1]$ yields the sharpest possible bound. Variance shadow mapping and exponential shadow mapping both employ solutions to Problem 3.1 for $\mathbb{I} = \mathbb{R}$. For layered variance shadow maps this is true as long as the depth intervals of the layers do not overlap. The lower bound computed by exponential variance shadow maps is not optimal because $\exp(c^+_{\text{evsm}} \cdot \mathbf{z})$ and $-\exp(-c^-_{\text{evsm}} \cdot \mathbf{z})$ are treated as independent random variables.

### 3.2.1 Numerical Solution

As of yet, Problem 3.1 only provides a generic way to describe the reconstruction for filterable shadow maps but it is not an actual technique to work with. In the present section we develop an algorithmic solution to the problem, which solves arbitrary problem instances as long as $\mathbb{I}$ is a finite set. However, it is significantly too slow for use in real-time rendering. Instead it will serve as foundation for our evaluation of candidate techniques.

We observe that the set of distributions $\mathbb{P}(\mathbb{I})$ is a linear space subjected to linear inequality constraints that enforce non-negative probabilities. The equation $\mathcal{E}_S(\mathbf{a}) = a$ additionally enforces linear equality constraints. Within this constrained search space we strive to minimize the functional $S(z_f > \mathbf{z})$

**Algorithm 3.1** Solution to Problem 3.1 for finite $\mathbb{I}$.
**Input:** $\mathbb{I} = \{z_0, \ldots, z_{n-1}\}$, $\mathbf{a} : \mathbb{I} \to \{1\} \times \mathbb{R}^m$, $a \in \{1\} \times \mathbb{R}^m$, $z_f \in \mathbb{R}$
**Output:** $G_{\mathbb{I},\mathbf{a}}(a, z_f) \in [0,\,1]$ or error "$a \notin \operatorname{conv} \mathbf{a}(\mathbb{I})$"

```
1. $A := (\mathbf{a}(z_0), \ldots, \mathbf{a}(z_{n-1})) \in \mathbb{R}^{(m+1)\times n}$
2. Construct $p \in \mathbb{R}^n$ with $p_l := \begin{cases} 0 & \text{if } z_f \leq z_l, \\ 1 & \text{if } z_f > z_l. \end{cases}$
3. Using a linear programming solver, compute $w \in \mathbb{R}^n$ with $w_0, \ldots, w_{n-1} \geq 0$ such that $A \cdot w = a$ and $p^\mathsf{T} \cdot w$ is minimal.
    a) On success: Return $p^\mathsf{T} \cdot w$.
    b) On failure: Indicate $a \notin \operatorname{conv} \mathbf{a}(\mathbb{I})$.
```

which is also linear in $S$. The form of this problem is exactly that of a linear programming problem. As soon as the search space is made finite-dimensional, standard solvers for linear programming are applicable. We accomplish this by discretizing the interval $\mathbb{I}$. The result is Algorithm 3.1. Prékopa [1990] uses a similar approach for a special case.

**Proposition 3.2.** *Algorithm 3.1 is correct.*

*Proof.* Suppose the algorithm terminates successfully. Let $S := \sum_{l=0}^{n-1} w_l \cdot \delta_{z_l}$. We note that

$$\mathcal{E}_S(\mathbf{a}) = \sum_{l=0}^{n-1} w_l \cdot \mathbf{a}(z_l) = A \cdot w.$$

In particular, $S(\mathbb{I}) = \sum_{l=0}^{n-1} w_l = (A \cdot w)_0$. Thus, $S$ is a probability distribution in $\mathbb{P}(\mathbb{I})$ if and only if $w_0, \ldots, w_{n-1} \geq 0$ and $(A \cdot w)_0 = a_0$. Furthermore, $\mathcal{E}_S(\mathbf{a}) = a$ if and only if $A \cdot w = a$. Overall, the constraints of the linear programming problem agree with the constraints for the search space in Problem 3.1. The functional also agrees:

$$S(z_f > \mathbf{z}) = \sum_{\substack{l=0 \\ z_f > z_l}}^{n-1} w_l = \sum_{l=0}^{n-1} p_l \cdot w_l = p^\mathsf{T} \cdot w$$

If Problem 3.1 has a solution, the linear programming solver will find and return it. Otherwise, the problem is infeasible and the algorithm will correctly indicate $a \notin \operatorname{conv} \mathbf{a}(\mathbb{I})$. □

Since we make excessive use of Algorithm 3.1, it is worthwhile to optimize it as good as we can. To this end, we observe that the reconstructed distribution $S = \sum_{l=0}^{n-1} w_l \cdot \delta_{z_l}$ never has more than $m + 1$ points of support because it corresponds to a vertex of the polytope that is the search space [Schrijver 1986, p. 96]. Most entries of $w \in \mathbb{R}^n$ vanish. The number of potential points of support $n \in \mathbb{N}$ is the most crucial parameter for the run time of Algorithm 3.1 but we only really need the points of support of the ground truth solution. Thus, we initially run Algorithm 3.1 with only $n = 251$ uniformly distributed samples. Then we refine the sampling near the points of support of the solution and rerun the algorithm using the previous output as initialization. This refinement is repeated one more time such that we solve three linear programming problems in total. The result is still a globally optimal solution to Problem 3.1 but for a potentially suboptimal set $\mathbb{I}$.

To solve the linear programming problems we use the implementation of the simplex algorithm in GLPK 4.54[2]. In some cases this implementation fails for numerical reasons and we fall back to Gurobi[3] which turned out to be slower in our application.

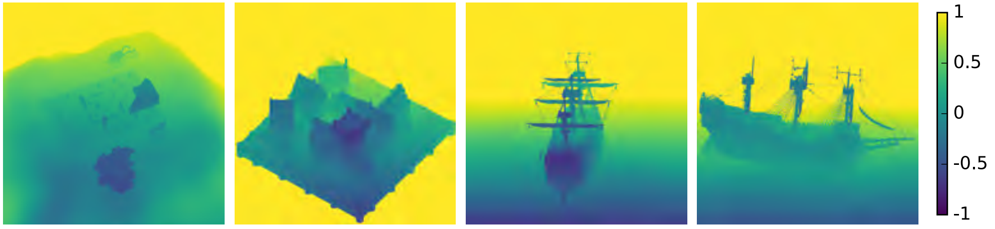
## 3.3 Benchmark of Candidate Techniques

Problem 3.1 defines an infinite supply of potential new shadow mapping techniques and Algorithm 3.1 allows us to test them. To evaluate their suitability for real-world applications we now want to test a selection of these techniques on real scenes. We care about robustness so the test cases should be challenging without being unrealistic. Thus, we prepare three scenes of moderate scale but high depth complexity and illuminate them using a single directional light source. The used shadow maps are shown in Figure 3.5.

Our ground truth is defined by percentage-closer filtering. To obtain reasonably complex depth distributions, we utilize a $9 \cdot 9$ Gaussian filter kernel with a standard deviation of 2.4 texels. Of course, percentage-closer filtering is prone to surface acne and we must avoid that in the ground truth. Otherwise, candidate techniques would be rewarded for reproducing this artifact. Therefore, we use a sophisticated slope-based depth bias for percentage-closer filtering and all candidate techniques.

[2] See www.gnu.org/software/glpk/ (retrieved on 1st of September 2016).

[3] See www.gurobi.com/ (retrieved on 1st of September 2016).



(a) Temple (b) Seaport (c) Ship (back) (d) Ship (side)

Figure 3.5: The shadow maps used in our benchmark of candidate techniques. Their resolution is $1024^2$. Sources of the models are detailed in the acknowledgments on page 9.

Having the ground truth, we need means of comparison. All artifacts produced by the candidate techniques are known to be some sort of light leaking because they never overestimate the shadow intensity. For the sake of simplicity, we quantify this light leaking through a 1-norm. It is applied to the error in the irradiance field of the directional light on the scene surfaces. Conveniently, the irradiance without the shadow term is proportional to the area covered in the shadow map. This enables a convenient image-based evaluation of the error term.

We render a stack of shadow maps showing not only the foremost surface but all surfaces. Then we iterate over all fragments in these images. For each fragment, we sample the shadow map to compute the depth distribution, which allows us to evaluate the shadow intensity with percentage-closer filtering and the candidate technique. The arithmetic mean of the differences between the two results is our error term. Except for discretization errors, this value is proportional to the error in the irradiance field. A value of zero is only accomplished by the ground truth whereas a value of one would mean that all surfaces should be fully shadowed but are not shadowed at all. The image data required to reproduce this benchmark are available[4].

Finally, we need to define the candidate techniques themselves, i.e. we have to define functions $\mathbf{a} : [-1, 1] \rightarrow \{1\} \times \mathbb{R}^m$ mapping a depth $z$ to the data stored in the shadow map $\mathbf{a}(z)$. A priori it is unclear which functions may perform well. This is the question we are concerned with after all. The component functions $\mathbf{a}_1, \ldots, \mathbf{a}_m$ can be defined independently. We pick them from a set of 37 rather elementary, smooth functions to have any hope of deriving an efficient closed-form solution afterwards. These

[4] cg.cs.uni-bonn.de/aigaion2root/attachments/BenchmarkData.zip (retrieved on 1st of September 2016).

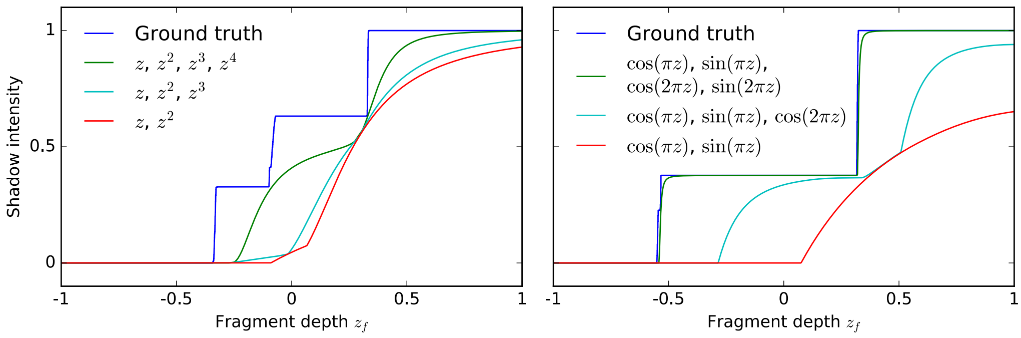


Figure 3.6: Lower bounds as defined by Problem 3.1 for different moment-generating functions shown in the legend. Note that the results obtained with $m = 4$ (green) are significantly better than those with $m = 3$ (cyan).

functions have been defined with a depth in [0, 1] so it is convenient to remap $z \in [-1, 1]$ to this interval via $y := \frac{z+1}{2}$. The used component functions are:

- Polynomials $y$, $y^2$, $\ldots$, $y^8$,
- Roots $\sqrt{y}$, $\sqrt[3]{y}$, $\sqrt[4]{y}$,
- Rational functions $\frac{1}{(y+1)^1}$, $\ldots$, $\frac{1}{(y+1)^4}$,
- Scaled exponential functions $\exp(1 \cdot y)$, $\ldots$, $\exp(4 \cdot y)$,
- Shifted logarithm functions $\log(y + 1)$, $\ldots$, $\log(y + 4)$,
- Fourier basis functions $\sin(1 \cdot 2 \cdot \pi \cdot y)$, $\ldots$, $\sin(4 \cdot 2 \cdot \pi \cdot y)$, $\cos(1 \cdot 2 \cdot \pi \cdot y)$, $\ldots$, $\cos(4 \cdot 2 \cdot \pi \cdot y)$,
- Trigonometric functions $\cosh y$, $\sinh y$, $\arcsin y$, $\arcsin(2 \cdot y - 1)$, $\arctan y$,
- The probability density of a Gaussian $\exp(-y^2)$.

The last quantity to determine is the number of component functions $m \in \mathbb{N}$ that we use at once. There is clearly a need for techniques that provide a higher quality than variance shadow mapping and exponential shadow mapping but use less memory than layered variance shadow mapping and convolution shadow mapping. Currently the only technique in this category is exponential variance shadow mapping. Thus, we want $m > 2$. Looking at the examples in Figure 3.6, $m = 4$ looks significantly more promising than $m = 3$ and therefore we fix this number.

We consider all combinations of the 37 component functions above leading to a total of $\binom{37}{4} = 66045$ candidate techniques. Performing the benchmark for all of them requires 392 billion evaluations of Algorithm 3.1. To process this massive workload in parallel, we use a cluster of 18 computers controlled by HT Condor[5].

## 3.4 Results and Discussion

The original publication on moment shadow mapping [Peters and Klein 2015] includes a complete benchmark of all 66045 candidate techniques. It took about one month of computing time on the cluster. In retrospect, this data turned out to be compromised significantly by the default numerical tolerances in the used linear programming solvers[6]. Therefore, we repeated the experiment with smaller tolerances on a random sample of 6605 candidate techniques. We discuss the issues in the original experiment and the implications below.

For now, we focus on the results of the repeated experiment shown in Figure 3.7. Our key observation is that more than 13000 candidate techniques perform nearly identical to the best technique in the sample. Visually the results of these candidate techniques are nearly indistinguishable. Only the shadow intensities reconstructed for complicated depth distributions, such as the one in Figure 3.2, differ by a few percent. Such distributions are rare. This is a very positive result because it means that there is a large pool of promising candidate techniques to choose from.

Our explanation for this phenomenon is based on a generalization of Propositions 2.12 and 2.13. These Propositions imply that the depth distribution can be reconstructed perfectly from four power moments or two complex trigonometric moments if it only contains two surfaces at constant depth. As discussed before, this situation is very important because it corresponds to the case where the filter region overlaps the silhouette of one shadow caster. The generalization of these propositions provides the same guarantee whenever the basis $\mathbf{a}_0, \ldots, \mathbf{a}_m$ spans a so-called Chebyshev system, i.e. no function in the span has more than $m$ roots [Kreĭn and Nudel'man 1977, p. 31, 78]. Apparently this holds for many candidates, at least approximately. Thus, they perform nearly identical in the most common cases. In

[5] research.cs.wisc.edu/htcondor/ (retrieved on 1st of September 2016).

[6] The relevant parameters in GLPK are `tol_bnd` and `tol_dj`. By default they are set to $10^{-7}$ but in the repeated experiment we use $10^{-11}$. In Gurobi the relevant parameter is named `GRB_DoubleParam_FeasibilityTol`.

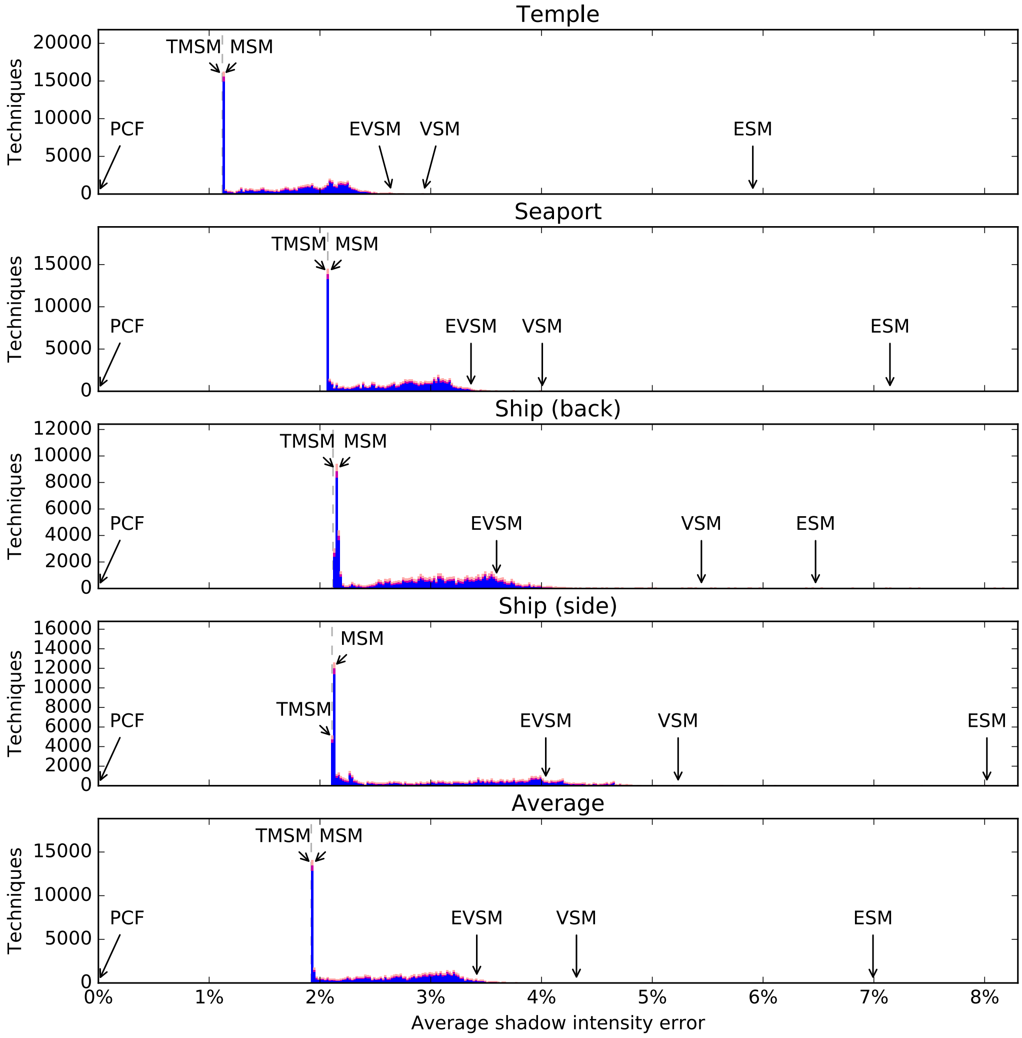


Figure 3.7: Histograms showing how many of the 66045 candidate techniques produce an error within particular bins. The counts are estimated based on a random sample of 6605 candidate techniques and the transparent red bars show confidence intervals for confidence level 95%. The upper four plots refer to the shadow maps in Figure 3.5. The plot at the bottom shows an arithmetic mean of these results weighting each shadow map by the number of shaded fragments. The errors of some noteworthy techniques are annotated: Percentage-closer filtering (PCF), trigonometric moment shadow mapping ($\mathbf{a}(z) = \mathbf{c}(\pi \cdot z)$, TMSM), Hausdorff moment shadow mapping ($\mathbf{a} = \mathbf{b}$, MSM), exponential variance shadow mapping ($c^+_{\text{evsm}} = 40$, $c^-_{\text{evsm}} = 5.54$, EVSM), variance shadow mapping (VSM) and exponential shadow mapping ($c_{\text{esm}} = 80$, ESM). The dashed line indicates the error of the best candidate technique in the sample.

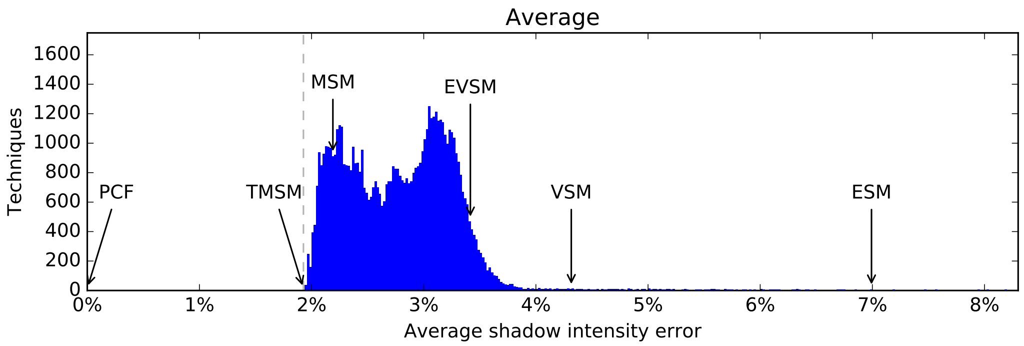


Figure 3.8: The counterpart of the bottom graph in Figure 3.7 when operating the linear programming solvers with default tolerances. This previously published data set [Peters and Klein 2015] includes all 66045 candidate techniques. Note that many candidates perform significantly worse.

more complex situations, the amount of information conveyed by the four general moments is still similar and results agree roughly.

Other candidates perform significantly worse but many are still competitive with regard to exponential variance shadow mapping and other related work. Note that the x-axis of the histograms is cut off. The worst measured average score is 23.5% and it is realized by

$$\mathbf{a}(z) = (1, \cos(1 \cdot \pi \cdot z), \cos(2 \cdot \pi \cdot z), \cos(3 \cdot \pi \cdot z), \cos(4 \cdot \pi \cdot z))^\mathsf{T}.$$

The reason for this poor performance is that this function is even, i.e. $\mathbf{a}(z) = \mathbf{a}(-z)$, and thus the lower bound has to be zero for $z_f \leq 0$.

Figure 3.8 shows results of the original experiment, which allow some interesting conclusions in spite of their flaws. The data also suggest that thousands of candidate techniques perform nearly as good as the best one but many measured errors are larger leading to a drastically spread out histogram. This difference is entirely due to tolerance thresholds in GLPK governing the accuracy with which the equality constraints for the general moments are maintained. The exact meaning of these thresholds is not stated in the documentation but using the default value of $10^{-7}$, we observed outputs which violate the equality constraints with an absolute error of as much as $10^{-3}$ on a value of 0.028.

GLPK uses these tolerances in such a way that its computed minimum is less than the true minimum of the linear programming problem. Thus light leaking is introduced, especially for short-range shadows. Given the relatively small distortions in the general moments, the strength of this light leaking is surprising. It is a strong indication that moment problems are ill-conditioned in practically relevant cases. Indeed, precision of the stored moments is an issue of major importance and in our derivation of moment shadow mapping Sections 4.1.3, 4.1.4 and 4.1.5 are dedicated to it.

Another interesting observation is that the smallest measured error is nearly unchanged across the two experiments. This supports the hypothesis that the full set of candidate techniques does not contain one that is significantly better than the best in the smaller sample. We expect the best candidate in the original experiment to be particularly robust with regard to errors in the given general moments. This candidate is

$$\mathbf{a}(z) = \mathbf{c}(\pi \cdot z) = (1, \cos(\pi \cdot z), \sin(\pi \cdot z), \cos(2 \cdot \pi \cdot z), \sin(2 \cdot \pi \cdot z))^\mathsf{T}$$

which corresponds to two complex trigonometric moments. Since we have already attributed some useful algebraic properties to trigonometric moments, a closed-form solution could be practical.

The power moments obtained with $\mathbf{a}(z) = \mathbf{b}(z) = (1, z, z^2, z^3, z^4)^\mathsf{T}$ are another promising candidate for which a closed-form solution seems feasible. In the repeated experiment, its average error of 1.93% is only insignificantly worse than the least measured error of 1.92%. In the original experiment, the average error of 2.19% is clearly worse than the least average error of 1.93% that is realized by trigonometric moments. Therefore, we expect increased light leaking when the power moments are given with low precision.

# Chapter 4

# Moment Shadow Maps

Knowing that power moments and trigonometric moments are among the many good choices that we could make, we now approach the development of fast and robust algorithms. We have implemented and evaluated shaders for three novel techniques:

**Hamburger Moment Shadow Mapping** solves Problem 3.1 for $\mathbf{a}(z) = \mathbf{b}(z) = (1, z, z^2, z^3, z^4)$ and $\mathbb{I} = \mathbb{R}$, i.e. it uses four power moments and does not incorporate the prior knowledge that valid depths lie in $[-1, 1]$. It even generalizes to any even number of power moments and an application using six power moments is given in Section 7.4. In nearly all situations it is the most compelling novel technique because it is fast and robust. When there is no need to distinguish it from the other two techniques, we refer to it as moment shadow mapping. However, it is slightly worse than the other two techniques in terms of light leaking.

**Hausdorff Moment Shadow Mapping** is very similar to Hamburger moment shadow mapping. It takes the same four power moments as input but solves Problem 3.1 for $\mathbb{I} = [-1, 1]$, i.e. it does incorporate the knowledge about the valid domain of depth values. In practice, this introduces an additional branch to the algorithm. Most of the time the result is computed in the same way as for Hamburger moment shadow mapping but in some situations the computed shadow will be slightly darker. This affects mostly shadows cast over a very short range. The downside is that it amplifies artifacts arising due to the quantization of the moment shadow map. Overall we find Hamburger moment shadow mapping preferable but we still provide a detailed derivation of Hausdorff moment shadow mapping in Appendix B.3.

**Trigonometric Moment Shadow Mapping** solves Problem 3.1 for $\mathbf{a}(z) = \mathbf{c}(\pi \cdot z)$ and $\mathbb{I} = [-1, 1]$, i.e. it uses two complex trigonometric moments. In the benchmark this technique has performed very well even in presence of errors in the trigonometric moments. Indeed, our experiments in Section 4.2.1 confirm that it is more robust to such errors than Hamburger moment shadow mapping. Unfortunately, the derivation of an efficient algorithm is substantially more difficult than for Hamburger moment shadow mapping. We have been able to derive a novel closed-form solution but it is far more expensive than Hamburger moment shadow mapping because it involves the solution of a complex, quartic equation. We do not recommend the use of this technique for any serious application but still provide a derivation in Appendix B.4.

In the following, we provide a detailed derivation of Hamburger moment shadow mapping. The basic principles of this technique are the same as for Hausdorff and trigonometric moment shadow mapping. Though, the latter two techniques require additional considerations which are quite specific to these techniques and, at times, more intricate.

## 4.1 Hamburger Moment Shadow Mapping

Solutions to Problem 3.1 for power moments go back to the work of Markov [1884]. They require computation of the roots of a polynomial of degree $\frac{m}{2}$, so they are closed forms up to $m = 8$. More recent work [Tari 2005] investigates the development of efficient and robust algorithms. However, this work has an entirely different application in mind and thus the algorithms are not optimized adequately for our application. In the following, we derive and implement an algorithm that is tailor-made for the present use case. It employs the same principles as earlier algorithms but in a different fashion. We formulate it for an arbitrary even number of power moments $m \in \mathbb{N}$ but only discuss the robust implementation for $m = 4$ and $m = 6$ (see Section 7.4).

### 4.1.1 Algorithm

The key insight for the solution is that optimal distributions $S$ on $\mathbb{R}$ have a very special structure. They can always be chosen to have no more than $\frac{m}{2} + 1$ points of support (see Figure 4.1). One of these points is the point at which the cumulative probability is to be minimized. Thus, the infinite-dimensional search space $\mathbb{P}(\mathbb{R})$ can be reduced to a search space with a mere $m + 1$ dimensions for the locations and probabilities of the points of sup-

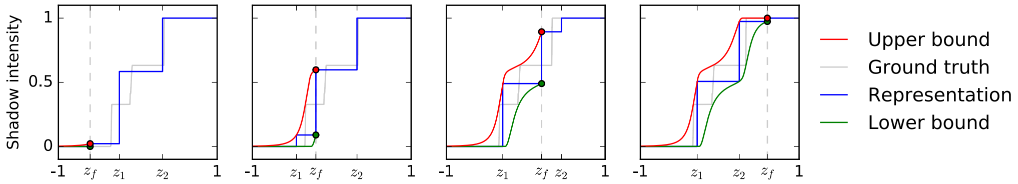


Figure 4.1: Examples of optimal solutions to Problem 3.1 for a single ground truth. The ground truth and the four representations share the same power moments $b_0, b_1, \ldots, b_4$. The upper and lower bound touch the representations at the respective $z_f$. Note that all representations use exactly three depth values $z_f, z_1, z_2$.

port. This dimensionality matches the number of constraints provided by $\mathcal{E}_S(\mathbf{b}) = b$ and therefore the minimization problem is simplified to solving a system of $m + 1$ equations.

This result is known as Chebyshev-Markov inequality [Kreĭn and Nudel'man 1977, p. 125] or Markov-Kreĭn theorem [Karlin and Studden 1966, p. 82]. It has been formulated for a more general class of moment-generating functions and objective functions but to avoid the additional definitions we only state the relevant special cases.

**Theorem 4.1** (Markov-Kreĭn for $\mathbb{I} = \mathbb{R}$)**.** *Let $b \in \mathbb{R}^{m+1}$ such that $b_0 = 1$ and $B(b)$ is positive definite. Let $z_f \in \mathbb{R}$ such that*

$$(B^{-1}(b) \cdot \hat{\mathbf{b}}(z_f))_{\frac{m}{2}} \neq 0. \tag{4.1}$$

*Then there exists exactly one probability distribution $S$ with $\mathcal{E}_S(\mathbf{b}) = b$ having support at $z_f$ and exactly $\frac{m}{2}$ additional points. It solves Problem 3.1, i.e.*

$$S(z_f > \mathbf{z}) = G_{\mathbb{R},\mathbf{b}}(b, z_f) = \inf_{\substack{S' \in \mathbb{P}(\mathbb{R}) \\ \mathcal{E}_{S'}(\mathbf{b}) = b}} S'(z_f > \mathbf{z}).$$

*The corresponding optimal upper bound is attained when we include the support at $z_f$, i.e.*

$$S(z_f \geq \mathbf{z}) = G_{\mathbb{R},\mathbf{b}}(b, z_f) + S(z_f = \mathbf{z}) = \sup_{\substack{S' \in \mathbb{P}(\mathbb{R}) \\ \mathcal{E}_{S'}(\mathbf{b}) = b}} S'(z_f \geq \mathbf{z}).$$

*Proof.* We refer to the literature for the proof of existence [Akhiezer and Kreĭn 1962, p. 8 f.] and optimality [Kreĭn and Nudel'man 1977, p. 125 f.]. □

Theorem 4.1 describes the sought-after solution in nearly all cases. There are two exceptions though. First, it demands that $B(b)$ is positive definite. We know that $B(b)$ has to be positive semi-definite from Proposition 2.8. If $B(b)$ is singular, Proposition 2.12 tells us that there is exactly one matching distribution, which is necessarily the solution to Problem 3.1. Algorithm 2.1 can compute it.

The condition formulated in Inequality (4.1) excludes a case where there really is no minimizing distribution. The infimum is rather realized by the limit of an infinite sequence of distributions spreading out their support towards infinity. In practice, this case is only a minor problem because $(B^{-1}(b) \cdot \hat{\mathbf{b}}(z_f))_{\frac{m}{2}}$ is a polynomial of degree $\frac{m}{2}$ in $z_f$. Thus, it can only have $\frac{m}{2}$ roots, i.e. out of all real numbers $\frac{m}{2}$ will cause problems. Upon close observation our results do indeed exhibit a few instabilities at individual fragments which are related to this case. They are rare enough to be ignored. Alternatively, one can use Hausdorff moment shadow mapping where another code branch takes over. We also demonstrate how to compute the exact result for this case in another context in Section 7.2.

The remaining problem is to compute the (existent and unique) distribution $S$ described in Theorem 4.1. More precisely, we have to compute its points of support. Once they are known, the corresponding probabilities are uniquely determined by the system of linear equations $\mathcal{E}_S(\mathbf{b}) = b$. We now reduce computation of the $\frac{m}{2}$ unknown points of support to polynomial root finding for a polynomial of degree $\frac{m}{2}$.

**Proposition 4.2** ([Akhiezer and Kreĭn 1962, p. 8 f.]). *Let $z_0, \ldots, z_{\frac{m}{2}} \in \mathbb{R}$ be pairwise different and let $w_0, \ldots, w_{\frac{m}{2}} > 0$ with $\sum_{l=0}^{\frac{m}{2}} w_l = 1$. Let $S := \sum_{l=0}^{\frac{m}{2}} w_l \cdot \delta_{z_l}$ and $b := \mathcal{E}_S(\mathbf{b})$. Then for all $l \in \{1, \ldots, \frac{m}{2}\}$*

$$\hat{\mathbf{b}}^\mathsf{T}(z_l) \cdot B^{-1}(b) \cdot \hat{\mathbf{b}}(z_0) = 0.$$

*Proof.* We note that $B(b)$ is regular by Proposition 2.12. Let

$$A := (\hat{\mathbf{b}}(z_0), \ldots, \hat{\mathbf{b}}(z_{\frac{m}{2}})) \in \mathbb{R}^{(\frac{m}{2}+1)\times(\frac{m}{2}+1)}.$$

This matrix is a square Vandermonde matrix and since $z_0, \ldots, z_{\frac{m}{2}}$ are pairwise different, it is invertible. We recall from Proposition 2.8 that

$B(b) = \mathcal{E}_S\left(\hat{\mathbf{b}} \cdot \hat{\mathbf{b}}^\mathsf{T}\right)$ and thus:

$$\begin{aligned} A^{-1} \cdot B(b) \cdot A^{-\mathsf{T}} &= A^{-1} \cdot \mathcal{E}_S\left(\hat{\mathbf{b}} \cdot \hat{\mathbf{b}}^\mathsf{T}\right) \cdot A^{-\mathsf{T}} \\ &= A^{-1} \cdot \left(\sum_{l=0}^{\frac{m}{2}} w_l \cdot \hat{\mathbf{b}}(z_l) \cdot \hat{\mathbf{b}}^\mathsf{T}(z_l)\right) \cdot A^{-\mathsf{T}} \\ &= \sum_{l=0}^{\frac{m}{2}} w_l \cdot \left(A^{-1} \cdot \hat{\mathbf{b}}(z_l)\right) \cdot \left(A^{-1} \cdot \hat{\mathbf{b}}(z_l)\right)^\mathsf{T} \\ &= \sum_{l=0}^{\frac{m}{2}} w_l \cdot e_l \cdot e_l^\mathsf{T} = \operatorname{diag}(w_0, \ldots, w_{\frac{m}{2}}) \end{aligned}$$

Note that $e_l := (0, \ldots, 0, 1, 0, \ldots, 0)^\mathsf{T} \in \mathbb{R}^{\frac{m}{2}+1}$ denotes the $l$-th canonical basis vector. Then the inverse matrix $A^\mathsf{T} \cdot B^{-1}(b) \cdot A$ is still a diagonal matrix and thus for all $l \in \{1, \ldots, \frac{m}{2}\}$

$$\hat{\mathbf{b}}^\mathsf{T}(z_l) \cdot B^{-1}(b) \cdot \hat{\mathbf{b}}(z_0) = (A^\mathsf{T} \cdot B^{-1}(b) \cdot A)_{l,0} = 0.$$

□

Proposition 4.2 provides the last missing piece. Putting everything together we get Algorithm 4.1 which constitutes the core of all of our recommended shadow mapping techniques.

**Theorem 4.3.** *If it does not abort in Step 1 or 3, Algorithm 4.1 solves Problem 3.1 correctly.*

*Proof.* If the algorithm fails, there is nothing to prove. Thus, consider the case that it terminates without failure. In this case, the conditions of Theorem 4.1 are met and there exists a unique distribution

$$S := \sum_{l=0}^{\frac{m}{2}} w_l \cdot \delta_{z_l}$$

with $z_0 = z_f$, $z_1, \ldots, z_{\frac{m}{2}} \in \mathbb{R}$, $w_0, \ldots, w_{\frac{m}{2}} > 0$ and $\mathcal{E}_S(\mathbf{b}) = b$. Then by Proposition 4.2 $z_1, \ldots, z_{\frac{m}{2}}$ are the roots of $\hat{\mathbf{b}}^\mathsf{T}(z) \cdot B^{-1}(b) \cdot \hat{\mathbf{b}}(z_0)$ which is exactly the polynomial in Step 4, i.e. the algorithm computes these points correctly.

The system of linear equations in Step 6 is equivalent to

$$\mathcal{E}_S\left(\hat{\mathbf{b}}\right) = (b_0, b_1, \ldots, b_{\frac{m}{2}})^\mathsf{T}$$

**Algorithm 4.1** Hamburger moment shadow mapping, i.e. the solution to Problem 3.1 for $\mathbf{a} = \mathbf{b}$ and $\mathbb{I} = \mathbb{R}$.
**Input:** Power moments $b \in \mathbb{R}^{m+1}$ and the fragment depth $z_f \in \mathbb{R}$.
**Output:** The lower bound $G_{\mathbb{R},\mathbf{b}}(b, z_f)$ as defined in Problem 3.1 or failure.

```
1. If $B(b)$ is not positive definite: Indicate failure.
2. Solve $B(b) \cdot q = \hat{\mathbf{b}}(z_f)$ for $q \in \mathbb{R}^{\frac{m}{2}+1}$.
3. If $q_{\frac{m}{2}} = 0$: Indicate failure.
4. Solve the polynomial equation $\sum_{j=0}^{\frac{m}{2}} q_j \cdot z^j = 0$ for $z$ and denote the distinct solutions by $z_1, \ldots, z_{\frac{m}{2}} \in \mathbb{R}$.
5. Set $A := (\hat{\mathbf{b}}(z_f), \hat{\mathbf{b}}(z_1), \ldots, \hat{\mathbf{b}}(z_{\frac{m}{2}})) \in \mathbb{R}^{(\frac{m}{2}+1)\times(\frac{m}{2}+1)}$.
6. Solve $A \cdot w = (b_0, b_1, \ldots, b_{\frac{m}{2}})^\mathsf{T}$ for $w \in \mathbb{R}^{\frac{m}{2}+1}$.
7. Return $\sum_{l=1,\, z_l<z_f}^{\frac{m}{2}} w_l$.
```

and thus $\mathcal{E}_S(\mathbf{b}) = b$ implies that the probabilities are recovered correctly as well. Note that $A$ cannot be singular because it is a square Vandermonde matrix constructed from pairwise different points of support.

Finally, the algorithm returns

$$\sum_{l=1,\, z_l<z_f}^{\frac{m}{2}} w_l = S(z_f > \mathbf{z}) = G_{\mathbb{R},\mathbf{b}}(b, z_f)$$

by Theorem 4.1. □

Additional branches utilizing Algorithm 2.1 and 7.1 could be added to compute a correct result for the two failure cases. However, we found that it is more practical to implement the Algorithm such that it behaves stable near these cases.

### 4.1.2 Implementation

In terms of the framework, Hamburger moment shadow mapping is no different from other filterable shadow maps. The best practice for creation of the moment shadow map is to first create a common shadow map as depth buffer with hardware-accelerated multisample antialiasing. Computation of the moments $z$, $z^2$, $z^3$, $z^4$ is done during a custom resolve into a four-channel texture that is no longer multisampled. Subsequently, a low-frequent filter such as a two-pass Gaussian may be applied. When shading a fragment, this texture is sampled with appropriate filtering and Algorithm 4.1 is used to approximate the filtered shadow intensity.

However, the implementation of Algorithm 4.1 is non-trivial for numerical reasons. Care has to be taken to implement each step in a numerically stable manner. At the same time, the implementation has to be very efficient because it is invoked per fragment. Only single precision arithmetic should be utilized. We note that there has been an earlier attempt by M. Salvi to implement moment shadow mapping, which is vaguely documented in a blog post [Salvi 2007]. It was unsuccessful due to the aforementioned challenges and M. Salvi moved on to derive exponential shadow mapping [Salvi 2008]. Pertaining his attempt to implement moment shadow mapping, he writes:

> “Moreover even though inequalities that handle three or four moments exist, they are mathematical monsters and we don’t want to evaluate them on a per pixel basis. In the end I decided to give it a go only to find out that this incredibly slow and inaccurate extension to variance shadow maps was only marginally improving light bleeding problems, and in some cases the original technique was looking better anyway due to good numerical stability that was sadly lacking in my own implementation.”

We now describe our solutions to these problems for $m = 4$ and summarize them in Algorithm 4.2. A robust implementation for $m = 6$ is described in Section 7.4. Shader code is given in Appendices C.1 and C.3.3.

To solve the $3 \times 3$ linear system $B(b) \cdot q = \hat{\mathbf{b}}(z_f)$, we exploit that $B(b)$ is symmetric and positive definite. These are exactly the conditions under which a Cholesky decomposition is usable. This way, the system is solved in a manner that is efficient and backward stable in all cases [Trefethen and Bau 1997, p. 176].

Simply solving the quadratic equation $\sum_{j=0}^{2} q_j \cdot z^j = 0$ with the quadratic formula works well.

Concerning the solution of the $3 \times 3$ linear system $A \cdot w = (1, b_1, b_2)^\mathsf{T}$, we observe that we do not require the full solution. For convenience, we assume $z_1 < z_2$. If $z_f \leq z_1$, the output is zero, if $z_1 < z_f \leq z_2$, the output is $w_1$ and if $z_2 < z_f$ the output is $w_1 + w_2 = 1 - w_0$. Starting from Cramer’s rule

**Algorithm 4.2** Robust implementation of Hamburger four moment shadow mapping, i.e. Algorithm 4.1 for $m = 4$.
**Input:** Power moments $b' \in \mathbb{R}^5$ with $b'_0 = 1$, fragment depth $z_f \in \mathbb{R}$.
**Output:** The lower bound $G_{\mathbb{R},\mathbf{b}}(b', z_f)$ as defined in Problem 3.1.

1. Use a Cholesky decomposition followed by forward and back substitution to solve for $q \in \mathbb{R}^3$:
$$\begin{pmatrix} 1 & b'_1 & b'_2 \\ b'_1 & b'_2 & b'_3 \\ b'_2 & b'_3 & b'_4 \end{pmatrix} \cdot q = \begin{pmatrix} 1 \\ z_f \\ z_f^2 \end{pmatrix}$$
2. Solve $q_2 \cdot z^2 + q_1 \cdot z + q_0 = 0$ for $z$ using the quadratic formula and let $z_1, z_2 \in \mathbb{R}$ with $z_1 < z_2$ denote the solutions.
3. If $z_f \leq z_1$: Return 0.
4. Else if $z_f \leq z_2$: Return
$$\frac{z_f \cdot z_2 - b'_1 \cdot (z_f + z_2) + b'_2}{(z_2 - z_1) \cdot (z_f - z_1)}.$$
5. Else: Return
$$1 - \frac{z_1 \cdot z_2 - b'_1 \cdot (z_1 + z_2) + b'_2}{(z_f - z_1) \cdot (z_f - z_2)}.$$

and simplifying reveals the closed forms given in Algorithm 4.2. We also experimented with the approach described in Section 7.4.2 but found that it is slightly slower without noticeable advantage.

If such an implementation of Algorithm 4.1 is implemented in double precision arithmetic and provided with double precision power moments, its results agree with the results of Algorithm 3.1. However, artifacts appear as soon as the power moments are provided in single precision.

### 4.1.3 Biasing

Rounding errors in the given power moments may invalidate the vector of moments. From Proposition 2.12 we know that the vector of moments lies on the topological boundary of the set of valid vectors of moments whenever the depth distribution has only one or two points of support. The majority of all relevant situations is approximated well by such depth

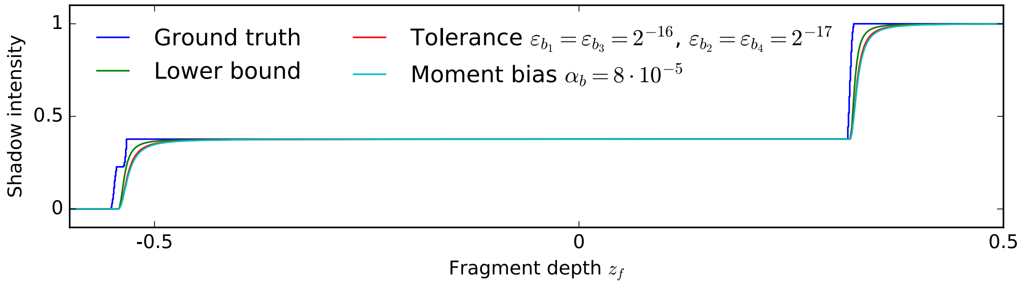


Figure 4.2: Various reconstructions for a single ground truth (blue). The green graph uses the exact moments, the red graph shows the lower bound obtained when the moments are only constrained to tolerance intervals $\left[b_j - \varepsilon_{b_j},\, b_j + \varepsilon_{b_j}\right]$ and the cyan graph uses the exact moments with a moment bias as defined in Equations (4.2) and (4.3). The value of $\alpha_b = 8 \cdot 10^{-5}$ is sufficient to compensate the rounding errors admitted by the tolerances. Notice how the tolerance and the moment bias both increase light leaking slightly but in a very similar manner.

distributions because the filter region in the shadow map covers no more than two surfaces. Thus, vectors of moments are commonly very close to the boundary and vulnerable to rounding errors. We need a robust and efficient method to counteract these rounding errors.

In Section 3.2 we argue that it is beneficial to always underestimate the shadow intensity to avoid surface acne. However, even when the vector of moments remains valid, rounding errors in the power moments may lead to overestimation. If we still want to guarantee underestimation, we have to incorporate knowledge about the rounding errors into the reconstruction.

As a means of analysis, we note that this is easily accomplished with the linear programming approach in Algorithm 3.1. Suppose we are given a vector of moments $b \in \mathbb{R}^{m+1}$ with rounding errors. Rather than knowing the power moment $j \in \{1, \ldots, m\}$ exactly, we only know that it lies in an interval $\left[b_j - \varepsilon_{b_j},\, b_j + \varepsilon_{b_j}\right]$ for an implementation-dependent constant $\varepsilon_{b_j} > 0$. Linear programming enables the replacement of the equality constraints by such inequality constraints. As shown in Figure 4.2, this has the effect of smoothing the lower bound in a conservative manner.

Of course, we require a more efficient scheme for real-time rendering. A lot of effort went into experiments with many different approaches but the most robust results were obtained with the simplest solution. We bias the rounded vector of moments by interpolating towards a fixed vector of biasing moments using a fixed weight.

More precisely, we fix a moment bias $0 < \alpha_b \ll 1$ and a vector of biasing moments $b^\star \in \operatorname{conv} \mathbf{b}([-1,1])$. The vector of moments with rounding errors $b$ is biased by setting

$$b' := (1-\alpha_b)\cdot b + \alpha_b \cdot b^\star. \tag{4.2}$$

This bias cannot increase the estimate of the shadow intensity by more than $\alpha_b$. To see that this is true suppose that $S, Z^\star$ are distributions on $\mathbb{R}$ with $\mathcal{E}_S(\mathbf{b}) = b$ and $\mathcal{E}_{Z^\star}(\mathbf{b}) = b^\star$. Then the biased vector $b'$ represents $(1-\alpha_b)\cdot S + \alpha_b \cdot Z^\star$ and thus

$$G_{\mathbb{R},\mathbf{b}}(b', z_f) \leq ((1-\alpha_b)\cdot S + \alpha_b \cdot Z^\star)(z_f < \mathbf{z}) \leq S(z_f < \mathbf{z}) + \alpha_b.$$

Since this relation holds for each choice of $S$, we obtain

$$G_{\mathbb{R},\mathbf{b}}(b', z_f) \leq G_{\mathbb{R},\mathbf{b}}(b, z_f) + \alpha_b.$$

On the other hand, a decrease of the lower bound is possible to an arbitrary extent. This behavior nicely matches up with our goal to guarantee underestimation in spite of rounding errors. Without using an unnecessarily high value of $\alpha_b$, we do not get a strong guarantee but Figure 4.2 demonstrates that the results nicely resemble those obtained with linear programming. We have observed this behavior across many randomly generated examples.

It remains to choose $b^\star$. Our main concern is robust behavior in all situations. Thus, we optimize for the worst case by choosing a $b^\star \in \operatorname{conv} \mathbf{b}([-1,1])$ that has maximal distance to the topological boundary of $\operatorname{conv} \mathbf{b}(\mathbb{R})$. The details of this optimization are described in Appendix B.1. The result is

$$b^\star = (1,\, 0,\, 0.375,\, 0,\, 0.375)^\mathsf{T}. \tag{4.3}$$

Using single precision throughout the pipeline our implementation yields robust results consistently with a moment bias of $\alpha_b = 3 \cdot 10^{-7}$.

### 4.1.4 Quantization

The most important factor for the run time of a filterable shadow map is still the amount of memory per texel because bandwidth is the bottleneck. Therefore, it is crucial to store the power moments $b_1$, $b_2$, $b_3$ and $b_4$ in as little memory as possible. The most practical options are to use either 64 or 128 bits per texel. Using 128 bits in the form of four single precision floating point values works well but when we use four 16-bit fixed precision values, a strong moment bias is required to avoid artifacts and light leaking increases. The quantized power moments simply do not provide enough information.

Information theory enables us to improve on this situation. We recall that the vector of moments $b$ always lies in the convex hull $\operatorname{conv}\mathbf{b}([-1,1])$. There is a closed-form expression for the $m$-dimensional volume of this convex hull [Karlin and Shapley 1953, p. 57]:

$$\operatorname{vol}\operatorname{conv}\mathbf{b}([-1,1]) = \prod_{j=1}^{m} \frac{2^j \cdot ((j-1)!)^2}{(2 \cdot j - 1)!} = \frac{64}{1575}$$

The naïve way to store the four power moments in four fixed precision numbers is to cover the axis-aligned bounding box of $\mathbf{b}([-1,1])$. It has volume

$$\operatorname{vol}\,[-1,\,1] \times [0,\,1] \times [-1,\,1] \times [0,\,1] = 4$$

because odd moments lie in $[-1,1]$ and even moments in $[0,\,1]$. From an information theoretic standpoint it follows that we dedicate approximately

$$\log_2 \frac{\operatorname{vol}\,[-1,\,1] \times [0,\,1] \times [-1,\,1] \times [0,\,1]}{\operatorname{vol}\operatorname{conv}\mathbf{b}([-1,1])} \approx 6.62$$

bits to encode the information $b \in \operatorname{conv}\mathbf{b}([-1,1])$ for every single texel. In other words, only one percent of the $2^{64}$ possible vectors encodes a meaningful vector of moments. This unnecessary redundancy reduces the entropy of the stored data.

Though, we are not willing to give up on the possibility to filter moment shadow maps linearly. Therefore, our only option is to apply an invertible, affine transform to the vector of moments before storing it. Since the zeroth moment is always one, it does not need to be stored and we define

$$\begin{aligned} b^+ &:= (b_1,\, b_2,\, b_3,\, b_4)^\mathsf{T} \in \mathbb{R}^m, \\ \mathbf{b}^+(z) &:= (\mathbf{b}_1(z),\, \mathbf{b}_2(z),\, \mathbf{b}_3(z),\, \mathbf{b}_4(z))^\mathsf{T} \in \mathbb{R}^m. \end{aligned}$$

We are looking for an affine transform $\Theta_m^\star : \mathbb{R}^m \to \mathbb{R}^m$ with

$$\Theta_m^\star(\operatorname{conv}\mathbf{b}^+([-1,1])) \subseteq [0,\,1]^m\,,$$

i.e. each vector of moments maps to a point in the unit tesseract $[0,\,1]^4$ that we can conveniently represent with four fixed precision values. The objective is to maximize

$$\operatorname{vol}\Theta_m^\star(\operatorname{conv}\mathbf{b}^+([-1,1])) = \operatorname{vol}\operatorname{conv}\mathbf{b}^+([-1,1]) \cdot |\det\Theta_m^\star|$$

so we are looking for the transform with maximal determinant.

Since we only need to determine this transform once, numerical optimization is feasible. Though, there is no need to consider the $(m+1)\times m$-dimensional search space consisting of all affine transforms. Given a linear transform, we can shift and scale each component function $(\Theta_m^\star(\mathbf{b}))_j$ such that it maps $[-1,1]$ to $[0,\,1]$. To this end, we simply compute minima and maxima of quartic polynomials. We sample the remaining $(m-1)\times m$-dimensional search space with random initializations and then perform a local optimization using Ceres Solver[1]. This recovers the following transform which we expect to be globally optimal:

$$\begin{aligned}\Theta_4^\star(b^+) = &\begin{pmatrix}0.94835322 & 0.07453389 & 0.32881232 & 0.94980125\end{pmatrix}^\mathsf{T}\\ &+\begin{pmatrix}-0.78917548 & -2.89102075 & 1.27119753 & 2.46064546\\ -0.91449110 & 2.63455252 & 1.41327106 & -2.21030645\\ 0.94159908 & 0.53380326 & -0.49488033 & -0.41589682\\ 0 & 0.54849905 & 0 & -1.49830030\end{pmatrix}\cdot b^+\end{aligned}$$

Its determinant is 4.85 which means that entropy increases by 4.28 bits compared to the naïve approach.

Application of this transform does have a measurable impact on the run time. Without having a proper explanation for this phenomenon, we found that results degrade only slightly if odd and even moments are transformed separately. In particular, the following transform still increases entropy by 4.21 bits but can be evaluated twice as fast due to the vanishing entries:

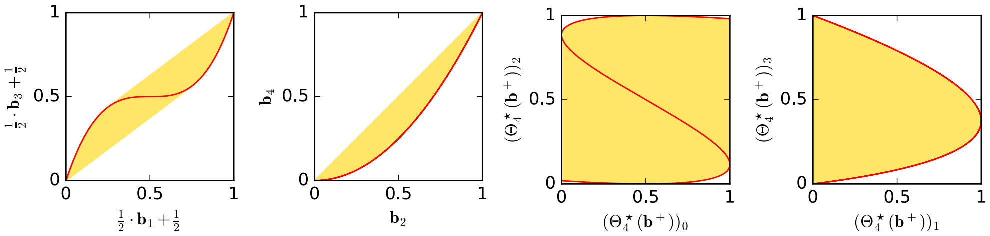
$$\Theta_4^\star(b^+) = \begin{pmatrix}\frac{3}{2} & 0 & -2 & 0\\ 0 & 4 & 0 & -4\\ \frac{1}{2}\cdot\sqrt{3} & 0 & -\frac{2}{9}\cdot\sqrt{3} & 0\\ 0 & \frac{1}{2} & 0 & \frac{1}{2}\end{pmatrix}\cdot b^+ + \begin{pmatrix}\frac{1}{2}\\ 0\\ \frac{1}{2}\\ 0\end{pmatrix} \tag{4.4}$$

It is our recommended solution. The way in which it expands $\operatorname{conv}\mathbf{b}^+([-1,1])$ is visualized in Figure 4.3. We have also adapted the biasing to the fact that rounding errors apply to $\Theta_4^\star(b^+)$ rather than $b^+$ (see Appendix B.1). In this setting the optimal biasing is

$$b^\star = (1,\,0,\,0.628,\,0,\,0.628)^\mathsf{T}$$

and for 16-bit quantization we found a moment bias of $\alpha_b = 6\cdot 10^{-5}$ to be sufficient in all experiments.

[1] See ceres-solver.org (retrieved on 1st of September 2016).



(a) Naïve quantization.

(b) Optimized quantization $\Theta_4^\star$.

Figure 4.3: A visualization of the quantization transform in Equation (4.4). The red line is the curve indicated by the axis labels. The yellow area is its convex hull. Note that the convex hull has a substantially larger area for the optimized quantization and that the graphs for the optimized quantization arise from the ones for naïve quantization through a simple affine transform.

### 4.1.5 Scaling and Translation of Depth Values

Using the optimized quantization transform is reasonable when using 16 bits per power moment but pointless when the power moments are stored in single precision because for evaluation of Algorithm 4.2 they have to be transformed back in single precision. Still we want to minimize the negative effect of rounding errors.

One option that we do have is to redefine the range of depth values by applying a linear transform $x \cdot z + y$ with $x, y \in \mathbb{R}$ and $x \neq 0$. Doing so transforms the moments linearly. For all $j \in \{0, \ldots, m\}$

$$\mathcal{E}_Z\left((x \cdot \mathbf{z} + y)^j\right) = \mathcal{E}_Z\left(\sum_{k=0}^{j} \binom{j}{k} \cdot (x \cdot \mathbf{z})^k \cdot y^{j-k}\right) = \sum_{k=0}^{j} \binom{m}{k} \cdot x^k \cdot y^{j-k} \cdot b_k.$$

This linear transform corresponds to a lower triangular matrix with diagonal entries $x^0, \ldots, x^m$. Therefore, its determinant is $\prod_{j=1}^{m} x^j$.

If we apply such a transform, the resulting $j$-th power moment can have a magnitude up to $(|x| + |y|)^j$. Since we are using floating point numbers, the relative precision remains unchanged if we divide by this magnitude to get back to a maximal magnitude of one. Combining this with the previous transform yields a determinant of

$$\prod_{j=1}^{m} \left(\frac{x}{|x| + |y|}\right)^j.$$

Obviously, the magnitude of this determinant is maximized by choosing $y = 0$ and invariant under changes of $x$.

We conclude that our decision to define depth on the interval $[-1,1]$ is optimal. For $m=4$, the volume of $\operatorname{conv}\mathbf{b}([-1,1])$ is $2\cdot 2^2\cdot 2^3\cdot 2^4=1024$ times larger than the volume of $\operatorname{conv}\mathbf{b}([0,\,1])$ while precision remains the same. Indeed, experiments confirm that a stronger moment bias is required when using depth values defined in [0, 1]. In terms of light leaking, this makes a notable difference.

Numerical issues aside, the above considerations reveal a unique property of Hamburger moment shadow mapping. A linear transform applied to the depth values does not change the information conveyed by the power moments. In theory, this means that the definition of the near and far clipping planes of the shadow map camera does not change the result of moment shadow mapping at all as long as no geometry is clipped. In Appendix B.2 we demonstrate that Hamburger moment shadow mapping is the only technique based on Problem 3.1 with this property. In practice, it is still advisable to choose the clipping planes tightly to minimize the negative effect of the biasing.

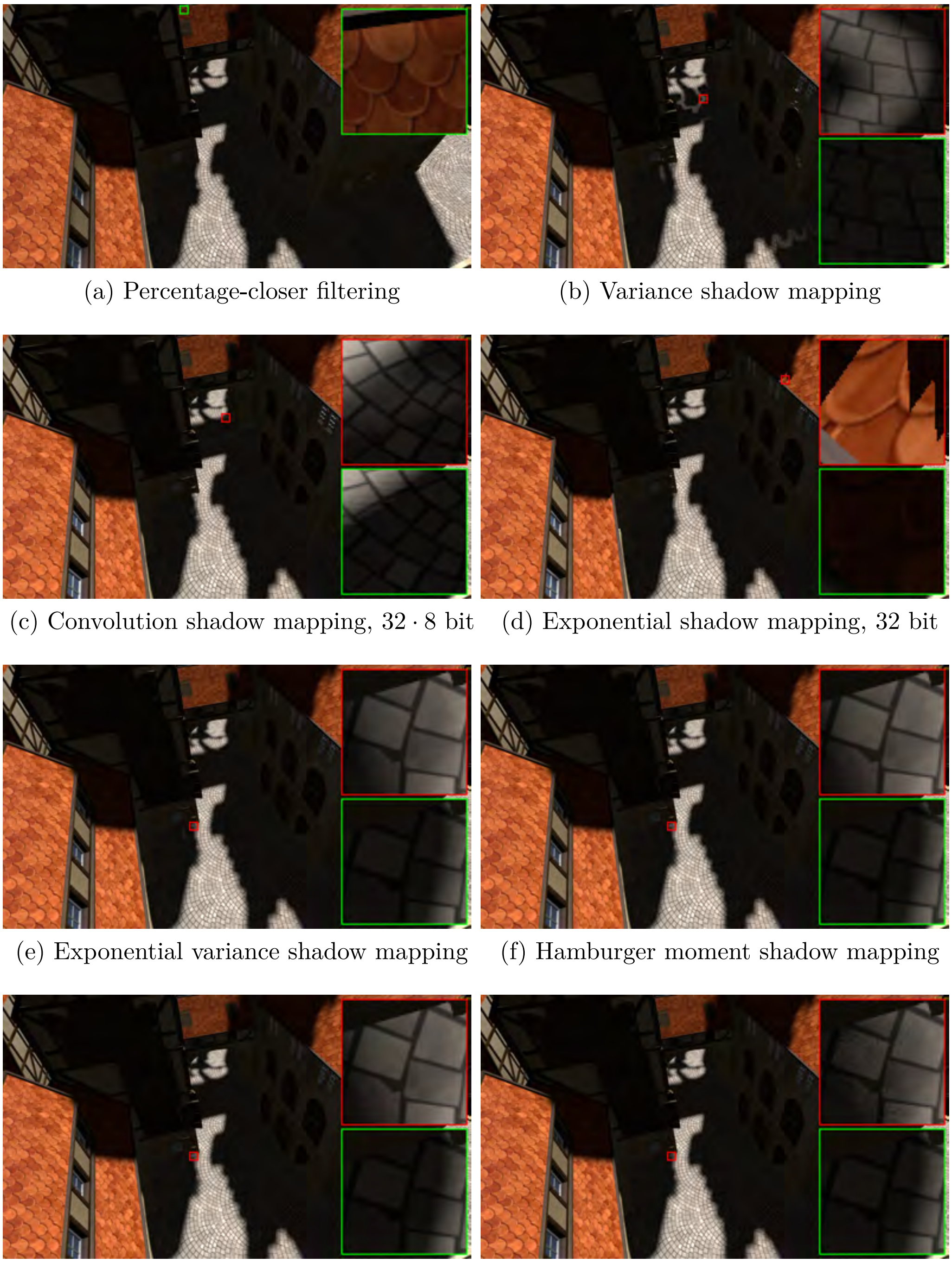
## 4.2 Results and Discussion

Our implementation uses forward rendering in Direct3D 11. Output images are in sRGB. It is noteworthy that the conversion from linear colors to sRGB, which is necessary for correct display on common monitors, considerably strengthens light leaking as it increases small values. Therefore, we scale the shadow intensity slightly as proposed by Annen et al. [2007]. For example, a computed shadow intensity of 98% may be mapped to full shadow by dividing by 0.98. The amount of scaling is indicated per figure.

All scenes in our experiments use a single directional light. The frustum for the corresponding shadow map is simply fitted to a bounding box around relevant parts of the scene. Thus, undersampling occurs frequently and the techniques for filtered hard shadows are crucial for the quality. Of course, techniques such as sample distribution shadow maps discussed in Section 3.1 can greatly diminish this undersampling.

### 4.2.1 Qualitative Evaluation

Figure 4.4 shows how the implemented techniques for filtered hard shadows perform in a moderately challenging situation with a $1024^2$ shadow map and a $9\cdot 9$ Gaussian filter with a standard deviation of 2.4 texels. All techniques expose some typical artifacts. Percentage-closer filtering yields surface acne on some of the steep roofs but performs well otherwise (Fig. 4.4a).



(a) Percentage-closer filtering

(b) Variance shadow mapping

(c) Convolution shadow mapping, $32 \cdot 8$ bit

(d) Exponential shadow mapping, 32 bit

(e) Exponential variance shadow mapping

(f) Hamburger moment shadow mapping

(g) Hausdorff moment shadow mapping

(h) Trigonometric moment shadow mapping

Figure 4.4: Results of all implemented techniques. Artifacts are magnified (red) above the percentage-closer filtering ground truth (green). Unless stated otherwise, the shadow maps use 16 bits per channel. Shadow intensities are divided by 98% and we use $c_{\text{esm}} = 80$ and $c^{+}_{\text{evsm}} = c^{-}_{\text{evsm}} = 5.54$.

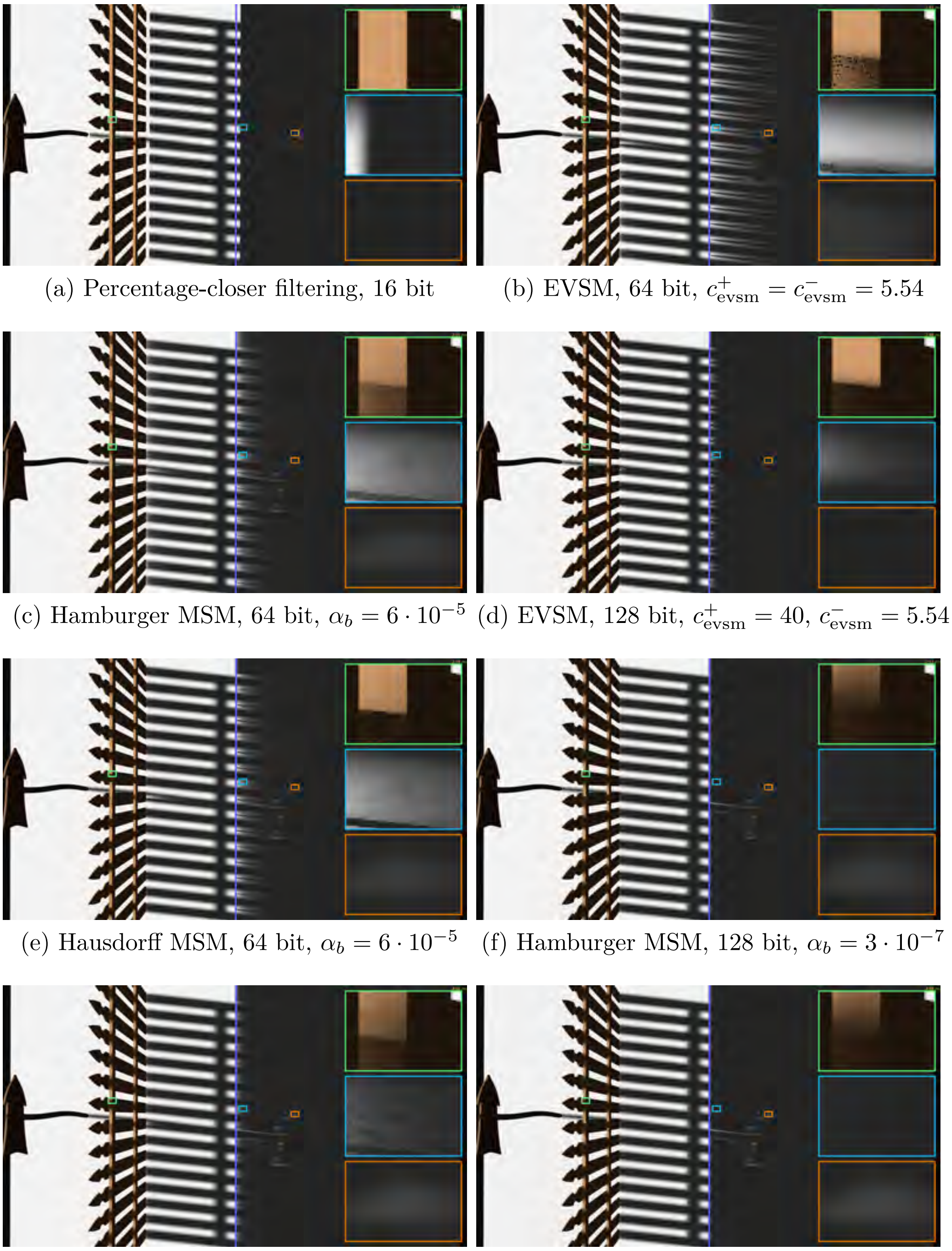
The silhouettes of shadow casters lead to light leaking for variance shadow mapping (Fig. 4.4b). Convolution shadow mapping exhibits ringing in spite of its high memory consumption (Fig. 4.4c). Exponential shadow mapping fails near boundaries of shadow casters (Fig. 4.4d). Note that our implementation does not fall back to percentage-closer filtering in these cases as proposed by Annen et al. [2008b].

In comparison to the other filterable shadow maps, the techniques using four channels produce substantially better results. The only artifact is some light leaking in shadows cast over a very short range. The results of Hausdorff and Hamburger moment shadow mapping are nearly identical (Figs. 4.4f, 4.4g). Light leaking of exponential variance shadow maps is slightly weaker in some places and stronger in others (Fig. 4.4e). Results of trigonometric moment shadow mapping are consistently better (Fig. 4.4h).

Figure 4.5 shows a comparison of these techniques in a more challenging case. The three magnified regions receive shadow from the fence only (green), the fence and the wall (cyan) or the fence, the wall and the direction sign (orange). The depth bias for percentage-closer filtering is set large enough to avoid surface acne but in consequence contact shadows vanish (Fig. 4.5a). For exponential variance shadow maps, the shadow of the fence leads to light leaking that stretches on over a long distance when using 64 bits per texel. Besides short-range shadows exhibit strong noise (Fig. 4.5b). Using 128 bits per texel diminishes the leaking and the noise (Fig. 4.5d).

Hamburger moment shadow mapping with 64 bits also produces light leaking but it is a lot weaker and so is noise in short-range shadows (Fig. 4.5c). Hausdorff moment shadow mapping with 64 bits behaves mostly identical but darkens short-range shadows (Fig. 4.5e). Hamburger moment shadow mapping with 128 bits eliminates light leaking due to the fence almost entirely (Fig. 4.5f). The results of Hausdorff moment shadow mapping with 128 bits are virtually indistinguishable and hence not shown. Unlike exponential variance shadow mapping, Hamburger moment shadow mapping with 128 bits fails for points on the ground receiving shadow from three different surfaces but does capture short-range shadows on the fence correctly (Figs. 4.5d, 4.5f). The results of trigonometric moment shadow mapping with 64 bits are similar in nature to those of Hausdorff moment shadow mapping with 64 bits but consistently better (Figs. 4.5e, 4.5g). When using 128 bits the two techniques produce nearly identical results (Figs. 4.5f, 4.5h). These observations agree with our predictions from Section 3.4.

Overall, the 64-bit variants of moment shadow mapping are superior to exponential variance shadow mapping with 64 bits. At 128 bits the tech-



(a) Percentage-closer filtering, 16 bit

(b) EVSM, 64 bit, $c^+_{\text{evsm}} = c^-_{\text{evsm}} = 5.54$

(c) Hamburger MSM, 64 bit, $\alpha_b = 6 \cdot 10^{-5}$

(d) EVSM, 128 bit, $c^+_{\text{evsm}} = 40$, $c^-_{\text{evsm}} = 5.54$

(e) Hausdorff MSM, 64 bit, $\alpha_b = 6 \cdot 10^{-5}$

(f) Hamburger MSM, 128 bit, $\alpha_b = 3 \cdot 10^{-7}$

(g) Trigonometric MSM, 64 bit, $\alpha_c = 6 \cdot 10^{-5}$

(h) Trigonometric MSM, 128 bit, $\alpha_c = 9 \cdot 10^{-7}$

Figure 4.5: A bird's-eye view of a thin wall lit by a directional light and shadowed by a fence and a direction sign. This scenario provokes light leaking in exponential variance shadow mapping (EVSM) and all variants of moment shadow mapping (MSM). The shadow intensity is not scaled.

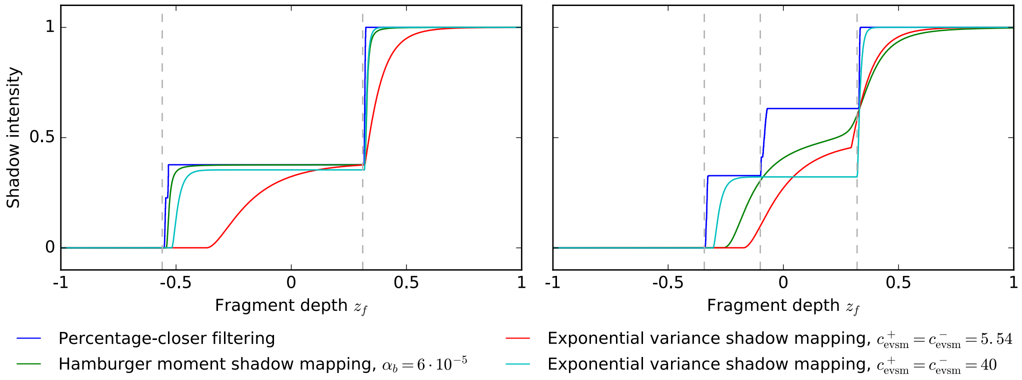


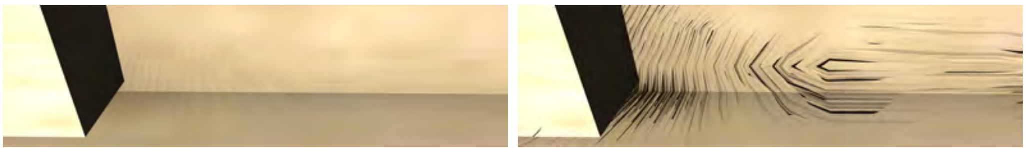
Figure 4.6: The reconstructed shadow intensities of several techniques in situations with two (left) or three (right) shadow casting surfaces.

niques yield artifacts under different circumstances and which technique is preferable depends on the use case. The different behavior can be better understood through Figure 4.6. As the exponents $c^+_{\text{evsm}}, c^-_{\text{evsm}}$ grow, the reconstruction of exponential variance shadow mapping approaches a function with two steps representing the foremost and the hindmost shadow caster. Thus, there is little light leaking behind the hindmost shadow caster but intermediate shadow casters are missed. Smaller exponents may not diminish the light leaking of variance shadow mapping sufficiently.

Moment shadow mapping only generates a reconstruction with two steps if it matches the ground truth (Proposition 2.12). For a ground truth consisting of three surfaces, it generates a smoother function that still provides a correct approximation at these three surfaces. However, the slow convergence to one leads to light leaking behind the hindmost shadow caster. Note that this behavior is entirely analogous to variance shadow mapping in presence of two surfaces. In general, $m$ power moments provide such a guarantee for $\frac{m}{2} + 1$ surfaces because whenever the fragment depth $z_f$ matches the depth of one of the $\frac{m}{2} + 1$ surfaces, Algorithm 4.1 will reconstruct the other $\frac{m}{2}$ surface depths correctly (Theorem 4.1). For $\frac{m}{2}$ surfaces or fewer, the reconstruction can be nearly perfect at all depths according to Proposition 2.12 but biasing brings back some light leaking.

Figure 4.7 demonstrates why we find Hamburger moment shadow mapping preferable to Hausdorff moment shadow mapping at 64 bits per texel. Wrong self-shadowing in Hausdorff moment shadow mapping can be very high-frequent due to quantization errors. The slightly less accurate bound of Hamburger moment shadow mapping, does not produce this artifact.

Additional results of moment shadow mapping are shown in Figure 4.8.



(a) Hamburger moment shadow mapping (b) Hausdorff moment shadow mapping

Figure 4.7: An extreme close-up of a concave corner demonstrating quantization artifacts in 64-bit Hausdorff moment shadow mapping, which are a lot weaker in 64-bit Hamburger moment shadow mapping.

Throughout this complex scene, we obtain plausible filtered hard shadows. Noticeable light leaking does occur at some points, e.g. in the shadow of the barrel on the bottom right, but it is not very strong and restricted to short-range shadows.

### 4.2.2 Run Time

To assess the speed of the various techniques, we measure frame times on an NVIDIA GeForce GTX 970 in a scene with 327397 triangles. The results are shown in Figure 4.9. Hausdorff moment shadow mapping and exponential shadow mapping perform very similar to Hamburger moment shadow mapping and variance shadow mapping, respectively. Therefore, their frame times are not shown separately.

The time spent per texel of the shadow map depends heavily on the amount of memory used per texel (Fig. 4.9a). Overall, this is the dominating factor for the frame time of most filterable shadow maps. For example, Hamburger moment shadow mapping with 64 bits is only slightly slower than exponential variance shadow mapping in spite of the more complex algorithm. Trigonometric moment shadow mapping is an exception due to its high algorithmic complexity. The cost per shaded fragment is so high that the higher quality of 64-bit trigonometric moment shadow maps cannot justify the additional cost as long as run time is any concern (Fig. 4.9b).

For sufficiently large shadow maps, percentage-closer filtering is faster than all sorts of filterable shadow maps (Fig. 4.9a). However, it should be noted that the effective resolution of the filterable shadow maps is higher due to the used 4× multisample antialiasing which cannot be utilized for common shadow maps. With respect to the cost per shaded fragment, filterable shadow maps are far less expensive than percentage-closer filtering because they only require a single texture sample (Fig. 4.9b). Thus, the cost of creating the filterable shadow map is amortized at higher output resolutions.

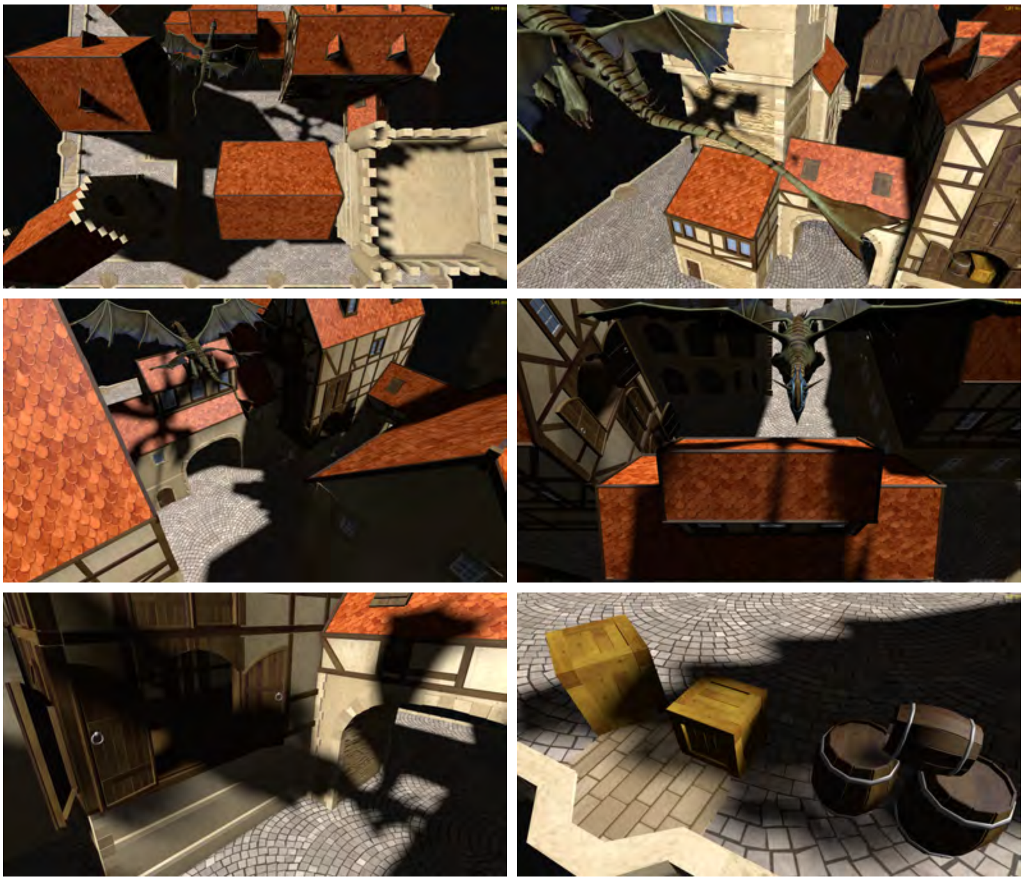
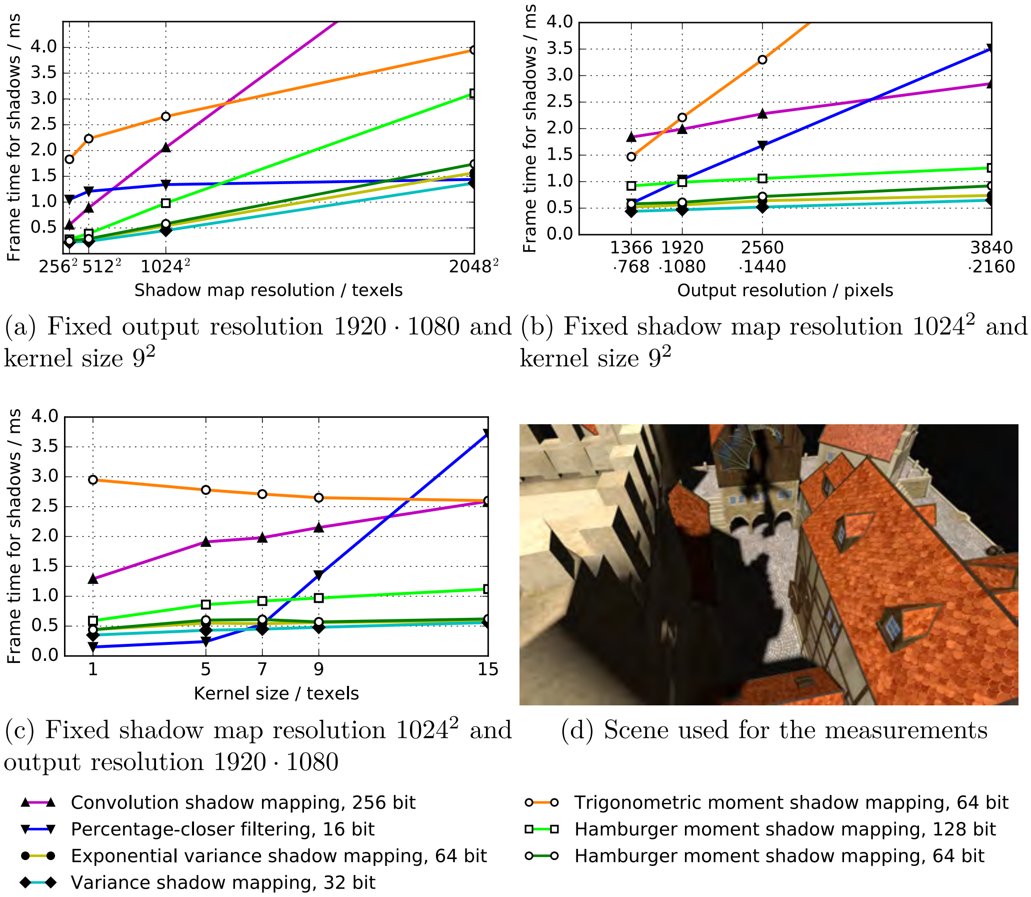


Figure 4.8: Screenshots of shadows computed with 64-bit Hamburger moment shadow mapping. They are captured at a resolution of $3840 \cdot 2160$. Shadow intensities are divided by 98%.

With filterable shadow maps, a two-pass Gaussian blur enables efficient smoothing of hard shadows with large kernels. The run time barely increases as the filter grows. On the other hand, the run time of percentage-closer filtering grows quadratically with the kernel size (Fig. 4.9c). Note that our implementation of percentage-closer filtering does use hardware-acceleration to take four shadow map samples at once.

### 4.2.3 Conclusions

At identical shadow map resolution, filterable shadow maps are more efficient than percentage-closer filtering at high output resolutions. With regard to ever increasing monitor resolutions and the recent trend towards head-mounted displays, their use is becoming more attractive than ever.



(a) Fixed output resolution 1920 · 1080 and kernel size $9^2$

(b) Fixed shadow map resolution $1024^2$ and kernel size $9^2$

(c) Fixed shadow map resolution $1024^2$ and output resolution 1920 · 1080

(d) Scene used for the measurements

Figure 4.9: Frame time measurements for various shadow mapping techniques. The frame times for scene rendering without shadows have been subtracted. All measurements use 4× multisample antialiasing for the output and where applicable also for the shadow map.

Even at lower resolutions, a 64-bit filterable shadow map with multisample antialiasing can be less expensive than percentage-closer filtering with a higher shadow map resolution while aliasing is the same.

Until recently, exponential variance shadow mapping has been the most practical shadow mapping technique using 64 bits per texel. At this memory consumption, Hamburger moment shadow mapping produces substantially less light leaking while taking only slightly more computation time. The improved robustness opens up filterable shadow maps for broader ranges of applications without the need for artist intervention to circumvent objectionable artifacts. Arguably, surface acne in percentage-closer filtering is more likely to require artist intervention than light leaking in moment shadow mapping (Fig. 4.4a).

Trigonometric moment shadow mapping provides an excellent quality at 64 bits per texel but for reasons that we consider fundamental, it cannot be implemented efficient enough to make it competitive. At 128 bits per texel the qualitative advantage vanishes as predicted in Section 3.4.

When using 128 bits per texel, moment shadow mapping is not necessarily better than exponential variance shadow mapping. In rare situations with three shadow-casting surfaces, light leaking is significantly stronger. On the other hand, exponential variance shadow mapping produces many artifacts near boundaries of shadow casters and leaks more light in common situations.

Variance shadow mapping and exponential shadow mapping remain attractive when artifacts are more tolerable because they only use 32 bits per texel. Convolution shadow mapping has the benefit of scaling to arbitrarily high quality albeit at a high cost.

Chapter 5

# Translucent Occluders

Prior art uses filterable shadow maps for far more than just filtered hard shadows. Being able to apply arbitrary linear filtering operations means that information for large filter kernels can be precomputed, which opens up applications in soft shadows (see Chapter 6) and single scattering (see Chapter 7).

Alpha blending is another linear operation that enables use of filterable shadow maps for translucent occluders. In the following, we demonstrate generation of shadows for translucent occluders by simply rendering to a moment shadow map with alpha blending enabled. The same moment shadow map is used for opaque and translucent occluders. Thus, there is almost no overhead. Light leaking is increased slightly by the more complex depth distributions but we demonstrate that moment shadow maps perform better than other filterable shadow maps. Our approach handles neither caustics nor subsurface scattering and requires sorted fragments.

## 5.1 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 it 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. McGuire and Enderton [2011] extend this method to colored objects. To avoid the costly filtering step, more recent work uses a variance shadow map and adds heuristic caustics [McGuire and Mara 2016]. In this case, a separate common shadow map for opaque occluders serves to avoid light leaking.

Fourier opacity mapping [Jansen and Bavoil 2010] introduced 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 employ ray marching to render single scattering.

## 5.2 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 of this choice is that we require 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]). Our experiments rely on sorted geometry.

We now demonstrate that alpha blending produces the vector of moments of a depth distribution $Z$ modeling transmittance of translucent occluders correctly. Given $n_s \in \mathbb{N}$ surfaces along a light ray at depths $z_0 < z_1 < \ldots < z_{n_s-1}$ with opacities $\alpha_0, \ldots, \alpha_{n_s-1} \in [0, 1]$, the amount of light transmitted to depth $z_f \in \mathbb{R}$ is the product of the relevant transmittance factors $\prod_{k=0,\, z_k<z_f}^{n_s-1}(1-\alpha_k)$. This transmittance is precisely modeled by the depth

distribution

$$Z := \sum_{j=0}^{n_s-1} \left( \prod_{k=0}^{j-1} 1 - \alpha_k \right) \cdot \alpha_j \cdot \delta_{z_j}$$

because at depth $z_j$ the fraction $\alpha_j$ of the remaining light is blocked.

Suppose we render these surfaces back to front to a moment shadow map using standard alpha blending. It is safe to assume $z_{n_s-1} = \alpha_{n_s-1} = 1$ because we clear the moment shadow map accordingly. Through alpha blending, the vector of moments $\mathbf{b}(z_j)$ for $z_j$ is first multiplied by $\alpha_j$ and subsequently by $(1 - \alpha_{j-1})$, …, $(1 - \alpha_0)$. Thus, we obtain the vector of moments

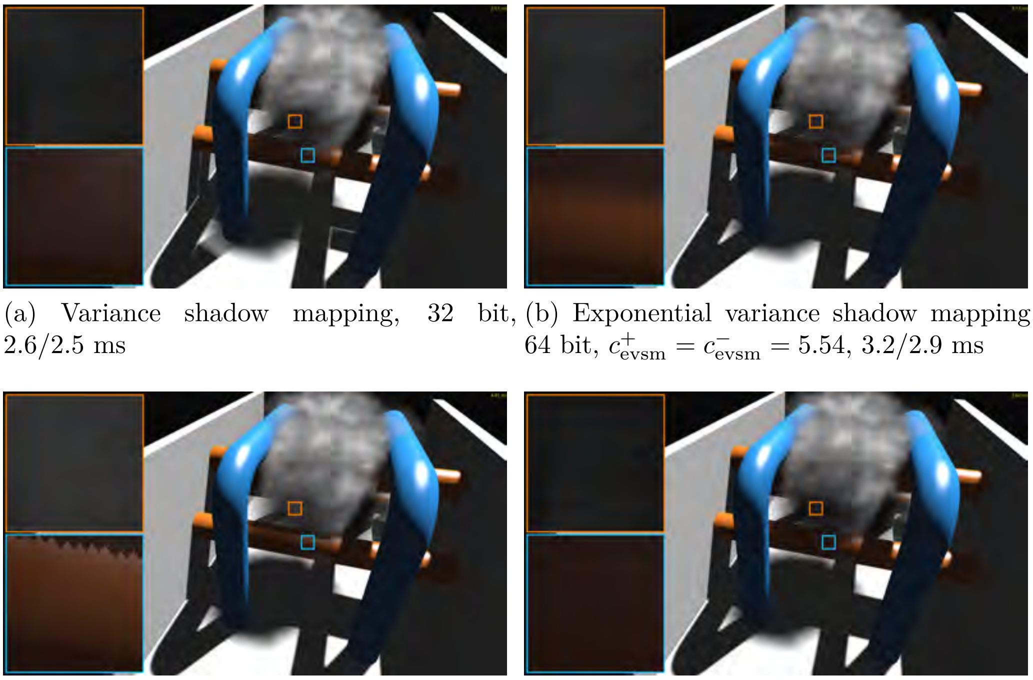
$$b := \sum_{j=0}^{n_s-1} \left( \prod_{k=0}^{j-1} 1 - \alpha_k \right) \cdot \alpha_j \cdot \mathbf{b}(z_j) = \mathcal{E}_Z\left(\mathbf{b}\right)$$

which is exactly the sought-after result. Approximation errors are only introduced during reconstruction of the shadow intensity from the power moments through Hamburger moment shadow mapping. Note that $b_0$ still does not need to be stored because it corresponds to total alpha and due to $\alpha_{n_s-1} = 1$ we know $b_0 = 1$.

Since alpha blending is required, translucent occluders have to be rendered to the moment shadow map directly rather than generating the entire moment shadow map from a depth buffer. Besides, we need to work around a limitation of current graphics APIs. The opacity value used for alpha blending cannot be independent from the values written to RGBA textures. Hence, we use two RG textures, each with 16 bits per channel, instead of a single RGBA texture. Rendering is done using hardware support for multiple render targets, so performance is only mildly reduced. Of course, it is still beneficial to use the optimized quantization transform.

## 5.3 Results and Discussion

While we have formulated the approach above for moment shadow maps, it is applicable to any kind of filterable shadow map and related works utilize that [Delalandre et al. 2011; McGuire and Mara 2016]. Figure 5.1 compares results obtained with different filterable shadow maps. All shown techniques underestimate the shadow intensity, so darker results are necessarily closer to the ground truth. We observe that moment shadow mapping yields the darkest self-shadowing in the smoke and the least light leaking on the pipes (Fig. 5.1d). Overall it performs best, although the run time increase in comparison to 64-bit exponential variance shadow maps is a bit higher



(a) Variance shadow mapping, 32 bit, 2.6/2.5 ms

(b) Exponential variance shadow mapping, 64 bit, $c^+_{\text{evsm}} = c^-_{\text{evsm}} = 5.54$, 3.2/2.9 ms

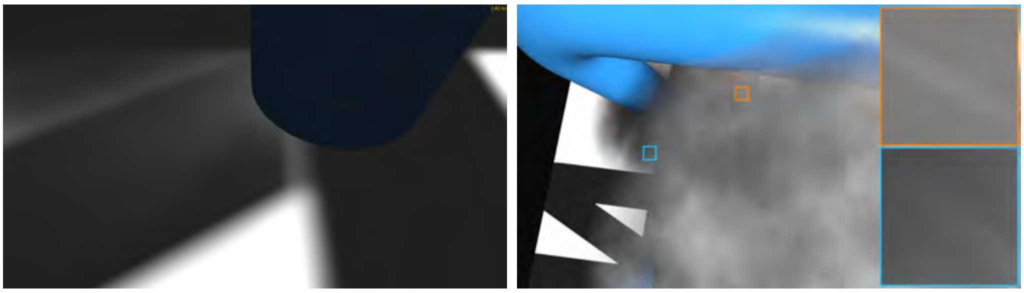
(c) Exponential variance shadow mapping, 128 bit, $c^+_{\text{evsm}} = 40$, $c^-_{\text{evsm}} = 5.54$, 4.4/3.5 ms

(d) Hamburger moment shadow mapping, 64 bit, $\alpha_b = 6 \cdot 10^{-5}$, 3.7/3.3 ms

Figure 5.1: A scene with walls, colored pipes and smoke consisting of 30 textured planes. Various filterable shadow maps are used to compute the shadows. Results exhibit different amounts of self-shadowing within the smoke and partial shadow of the smoke on the opaque surfaces. The shadow map resolution is $1024^2$ and images are rendered at $3840 \cdot 2160$ with 4× multisample antialiasing. Timings are full frame times for rendering to the filterable shadow map with/without alpha blending. Shadow intensities are divided by 98%.

than usual due to the high shading rate. Using 128-bit moment shadow maps is not beneficial here because the negative effect of the biasing is less significant when depth distributions are complex in the first place.

64-bit exponential variance shadow maps yield slightly weaker self-shadowing in the smoke and there is strong light leaking at the boundary of the pipe (Fig. 5.1b). The higher exponent in 128-bit exponential variance shadow mapping actually makes both artifacts worse (Fig. 5.1c). With variance shadow mapping the shadows cast by the smoke are reconstructed almost as well as with 64-bit exponential variance shadow mapping but there is unacceptable light leaking on opaque surfaces (Fig. 5.1a).



(a) Light leaking on surfaces

(b) Volumetric light leaking

Figure 5.2: Screenshots of two typical artifacts of shadows for translucent occluders rendered with 64-bit Hamburger moment shadow mapping. The complex depth distributions increase light leaking on surfaces and in the volume. To make the artifacts more visible, shadow intensities are not scaled and the image on the right does not use a two-pass Gaussian for filtering of the shadows.

Figure 5.2 demonstrates artifacts encountered with moment shadow mapping for translucent occluders. The many layers of the smoke in our test scene add to the complexity of depth distributions and thus light leaking on the surfaces increases (Fig. 5.2a). 64-bit exponential variance shadow maps exhibit very similar artifacts. Dividing shadow intensities by 95% resolves the issue in this example. More complex depth distributions degrade the approximation quality at all depths. Therefore, the silhouette of the blue pipe in the background leads to increased light leaking along elongated stripes for the self-shadows of the smoke in the foreground in Figure 5.2b.

Note that rounding errors may accumulate through alpha blending. In some experiments with 64-bit moment shadow maps we observed corresponding artifacts. Accumulation of rounding errors is particularly strong when there are many overlapping layers with a very low opacity. We found that a simple alpha test discarding fragments with an opacity below 1% removed these artifacts. If an alpha test is not an option, one may use 128-bit moment shadow maps or a method for transparency other than alpha blending.

In spite of these artifacts, we believe that the technique is robust enough for use in production. It is particularly attractive due to its simple implementation and its low overhead (e.g. 0.4 ms for the scene in Figure 5.1d). The combination with stochastic shadow maps [Enderton et al. 2010] seems compelling for situations where sorting the translucent geometry is not practical. The technique can also be extended to colored translucent occluders by using one moment shadow map per color channel.

Chapter 6

# Soft Shadows

So far, all shown results use a directional light, i.e. a point light at infinite distance. In reality, light sources always have some extent and correspondingly cast soft shadows with smooth penumbra regions where the light is partially occluded. To some extent the assumption of a point light is inherent in shadow mapping because the shadow map is rendered from a single point of view. On the other hand, filtered hard shadows appear similar to soft shadows. The most notable difference is that they remain soft at contact points.

The currently most practical approximation to soft shadows in performance-sensitive real-time applications exploits this similarity by adapting the size of the filter kernel in percentage-closer filtering to the distance between occluder and receiver [Fernando 2005]. As long as shadow casters of different depth do not overlap, this yields convincing results at the cost of excessive sampling of the shadow map.

In the following, we extend earlier work [Yang et al. 2010] to combine this approach with moment shadow maps. The sampling step is made more efficient by using a summed-area table. Our technique produces robust soft shadows using only two queries to this four-channel summed-area table.

## 6.1 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 produce accurate soft shadows, they are too costly for most interactive applications.

More practical methods, including ours, are derived from 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 fragment it performs a blocker search, sampling the shadow map to determine the average depth of the occluding geometry. Then the adequate size of the penumbra is 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]. Aliasing is a problem near contact points. Therefore, Story and Wyman [2016] propose to blend over to hard shadows computed with irregular z-buffers [Wyman et al. 2015].

Summed-area variance shadow maps [Lauritzen 2007] try to avoid the excessive sampling by means of 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. The integral over an arbitrary rectangle is queried by sampling the summed-area table at the four corners of this rectangle (see Figure 6.1a). 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. Filtering is done using a box kernel which corresponds to a rectangular area light.

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. 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.

## 6.2 Summed-Area Tables with Four Moments

Our technique follows the same basic steps as variance soft shadow mapping but never resorts to smaller filter kernels. 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 begin our discussion of the individual steps with the generation of the summed-area table.

The summed-area table is created in two passes. The first one creates horizontal prefix sums and the second one creates vertical prefix sums on the output of the first pass. Both passes are implemented in a compute shader using one thread per row/column as recommended by Klehm et al. [2014a].

For small variance shadow maps the precision provided by summed-area tables with single-precision floating point values may be sufficient but for moment shadow maps it is generally insufficient (cf. Section 4.1.3). 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 $n_t \in \mathbb{N}$ texels (e.g. $n_t = 784$ for a $28 \cdot 28$ kernel). The transformed power moments $\Theta_4^\star(b^+)$ (see Equation (4.4) on page 66) stored in the moment shadow map initially lie in $[0,\,1]^4$. If we multiply them by $\frac{2^{32}-1}{n_t}$ and round to integers afterwards, the sum of power moments 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

$$\log_2 \frac{2^{32}-1}{n_t}$$

which evaluates to 22.4 bits for the example above. This precision is only slightly worse than the precision of single precision floats and we found that a moment bias of $\alpha_b = 6 \cdot 10^{-7}$ is sufficient. For larger kernels, higher values are needed.

In our implementation 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 $n_t$ texels or fewer, 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.

Having a summed-area table, mipmapping becomes unnecessary. Therefore, the memory requirements compared to 64-bit moment shadow mapping only increase by 50%:

$$\frac{128\,\text{bit}}{64\,\text{bit}\cdot\frac{4}{3}}=\frac{6}{4}=150\%$$

## 6.3 Blocker Search

During 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 Algorithm 4.1 to turn the biased moments and the unbiased fragment depth $z_f$ into a matching depth distribution $Z := \sum_{l=0}^{2} w_l \cdot \delta_{z_l}$ consisting of three depth values $z_0 = z_f$, $z_1, z_2 \in \mathbb{R}$ with probabilities $w_0, w_1, w_2 > 0$.

Our assumption is that this reconstruction matches up with the ground truth. If the search region contains one or two surfaces, this assumption is justified by Proposition 2.12. When the search region contains three surfaces but one of them is at depth $z_f$, the distribution is still uniquely determined by the power moments according to Theorem 4.1 and is reconstructed correctly. For this reason, it is beneficial to use the unbiased fragment depth. More complicated cases are rare.

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

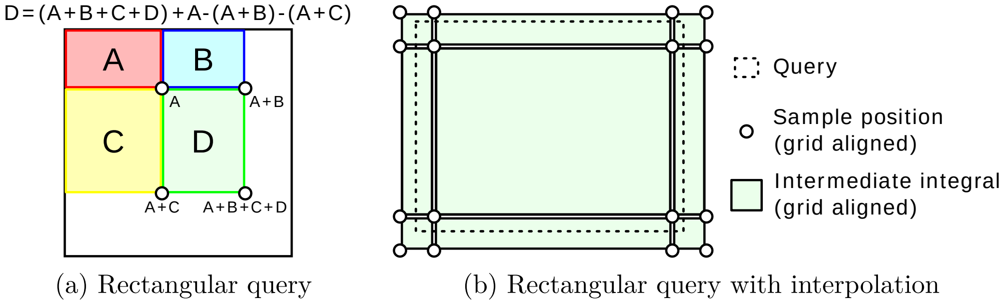
$$\frac{\sum_{l=1,\, z_l<z_f}^{2} w_l \cdot z_l}{\sum_{l=1,\, z_l<z_f}^{2} w_l}$$

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 $z_0 = z_f$ to indicate that a small filter kernel is to be used.

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

$$\frac{\varepsilon_{z_0} \cdot z_0 + \sum_{l=1,\, z_l<z_f}^{2} w_l \cdot z_l}{\varepsilon_{z_0} + \sum_{l=1,\, z_l<z_f}^{2} w_l}$$

where $\varepsilon_{z_0} > 0$ is a small constant. We found that this parameter is not crucial for the quality. Greater values move all shadow casters slightly


D=(A+B+C+D)+A-(A+B)-(A+C)
A
B
C
D
A
A+B
A+C
A+B+C+D
Query
Sample position (grid aligned)
Intermediate integral (grid aligned)

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

Figure 6.1: (6.1a) Summed-area tables enable computation of the integral over a rectangle $D$ from four samples. (6.1b) To compute the integral over an arbitrary rectangle, we load values of 16 texels and compute the integrals over the nine shown rectangles.

towards the receivers thus making shadows harder. For small values, the average blocker depth may be too far away leading to an unnecessarily large filter kernel. However, this typically affects fully lit fragments, so the final result does not change. We use $\varepsilon_{z_0} = 10^{-3}$ in all our experiments.

## 6.4 Filtering

Once average blocker depth is available, the penumbra estimation [Fernando 2005] provides an adequate filter size. Combining it with the texture coordinate of the fragment in 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 unknown multiples of $2^{32}$. Such problems can be avoided by operating exceptionally on integrals over regions containing fewer than $n_t$ texels. Figure 6.1b 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 are 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 6.6.2). As an alternative we tried randomized dithering but found that the noise is too strong at hard shadow boundaries.

Having the four filtered power moments, Hamburger moment shadow mapping (Algorithm 4.2) yields the final shadow intensity. Due to the potentially large filter size, it is important to use a sufficient depth bias. We recommend increasing it in proportion to the filter size. Additionally, an adaptive depth bias may be used [Dou et al. 2014].

## 6.5 Optimization

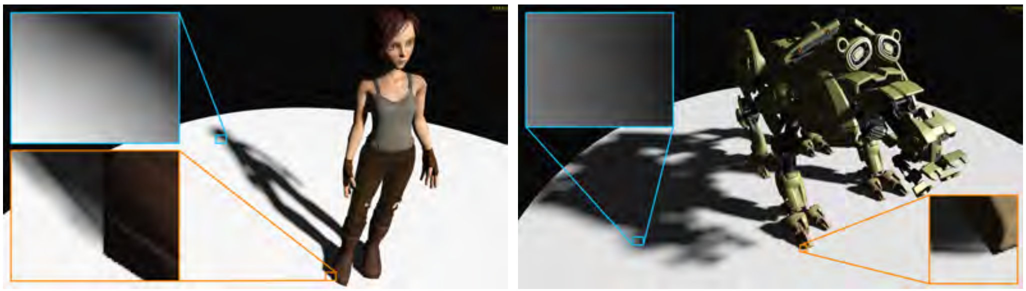
The most efficient way to optimize the technique is to reduce the number of texture loads. To avoid the cost of interpolation during the blocker search, we extend the search region to match the texel grid. Having grid-aligned search regions leads to small discontinuities in the soft shadows but is easily 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 umbra. We compute the shadow intensity $\sum_{l=1,\, z_l<z_f}^{2} w_l$ from available quantities. If it surpasses a threshold $1-\varepsilon_u$ where $\varepsilon_u > 0$, we assume that the fragment lies in the umbra and immediately return a maximal shadow intensity. In our experiments a value of $\varepsilon_u = 0.01$, coupled with division of the shadow intensity by 99% or less, yields robust results while reducing the need for texture loads in large, connected regions.

We also tried skipping the filtering step for fully lit fragments but the lower bound provided by moment shadow mapping leads to too many false positives. It is possible to use the upper bound instead but then only few fragments are classified as fully lit. Therefore, we do not recommend this approach and do not use it in our experiments.

## 6.6 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 power moments. Thus, we expect it to be faster than the actual technique [Yang et al. 2010] but



(a) Sintel

(b) Quadbot

Figure 6.2: Moment soft shadow mapping with a search region of $27^2$ rendering shadows for two models above a plane. Note that the shadows are contact hardening.

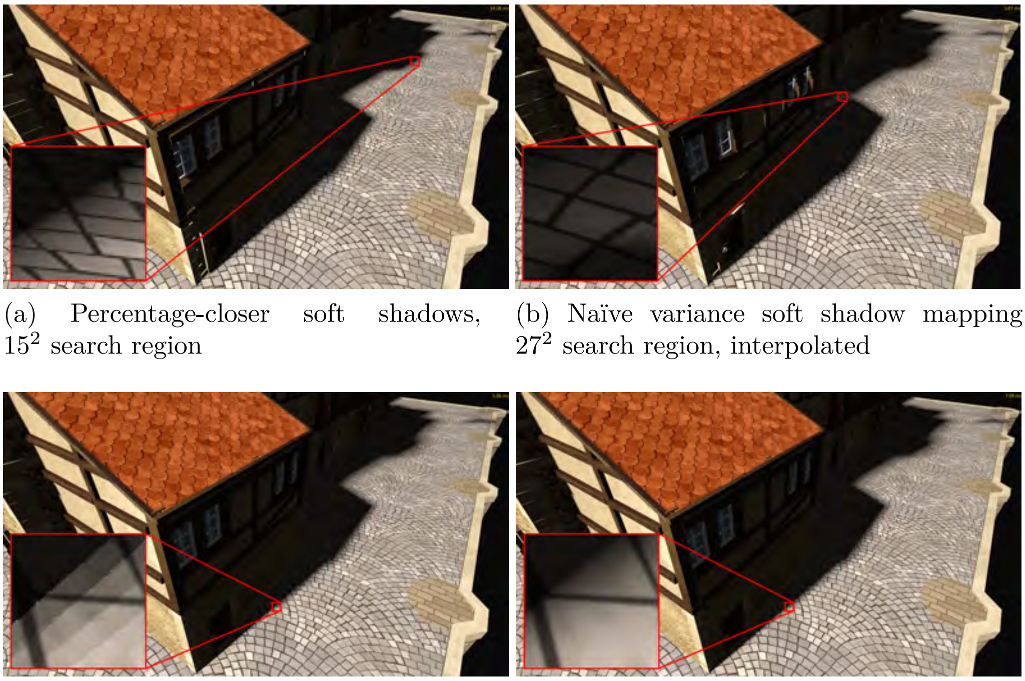
with more artifacts. All techniques skip filtering if the result of the blocker search allows it. We divide shadow intensities by 98% throughout this chapter.

### 6.6.1 Qualitative Evaluation

Figure 6.2 shows two examples where moment soft shadow mapping produces plausible soft shadows. It works well for character models (Fig. 6.2a) but also for complex models with many fine details (Fig. 6.2b). As expected, shadows harden at contact-points. Note that short-range shadows exhibit slight light leaking. Since precision in the summed-area table is high, the light leaking is only slightly stronger than for single-precision moment shadow maps (see Figure 4.5f).

Figure 6.3 compares all implemented techniques for soft shadows. Percentage-closer soft shadows generates a good result but to get an acceptable run time the search region has to be limited to $15^2$ and therefore long-range shadows are too hard (Fig. 6.3a). The other techniques support large search regions efficiently. Our naïve implementation of variance soft shadow mapping produces objectionable light leaking (Fig. 6.3b). 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. 6.3c). Interpolation eliminates this artifact (Fig. 6.3d).

A failure case is shown in Figure 6.4. 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



(a) Percentage-closer soft shadows, $15^2$ search region

(b) Naïve variance soft shadow mapping, $27^2$ search region, interpolated

(c) Moment soft shadow mapping, $27^2$ search region, not interpolated

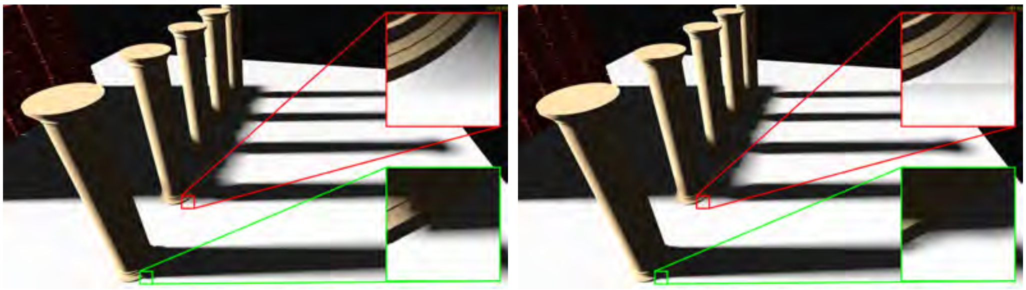
(d) Moment soft shadow mapping, $27^2$ search region, interpolated

Figure 6.3: Various techniques for soft shadows in a scene where shadows are cast over a long range.

short-range shadow of a pillar becomes soft due to the shadow of the more distant brick wall (Fig. 6.4a). This artifact occurs whenever the search region contains more than two occluding surfaces. Thus, it coincides with increased light leaking making the artifact more noticeable for moment soft shadow mapping (Fig. 6.4b). A stronger depth bias strengthens this light leaking.

The effect of an insufficient depth bias is shown in Figure 6.5. Lighting that is nearly parallel to a surface, coupled with large filter regions, is likely to cause wrong self-shadowing. Moment soft shadow mapping is less susceptible to this artifact than percentage-closer soft shadows but when using large search regions, it is an issue. In our implementation the depth bias is proportional to the filter size but not adaptive with respect to the surface. We believe that an adaptive depth bias will offer robust results without parameter tweaking, even for large search regions [Dou et al. 2014].

Overall, we found moment soft shadow mapping to be more robust than percentage-closer soft shadows. With percentage-closer soft shadows miss-



(a) Percentage-closer soft shadows

(b) Moment soft shadow mapping

Figure 6.4: An example of wrong occluder fusion. The hardness of the contact shadows in the two magnified insets should be the same. Both shown techniques exhibit this artifact but light leaking of moment soft shadow mapping strengthens it.

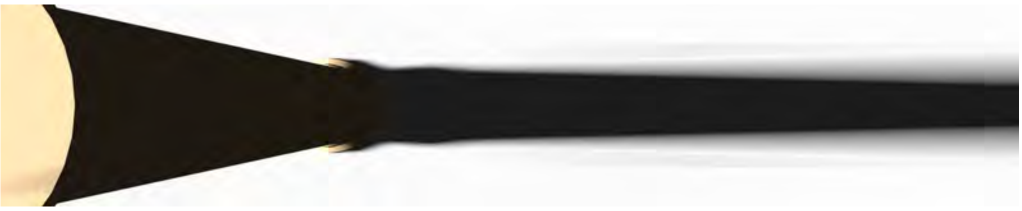
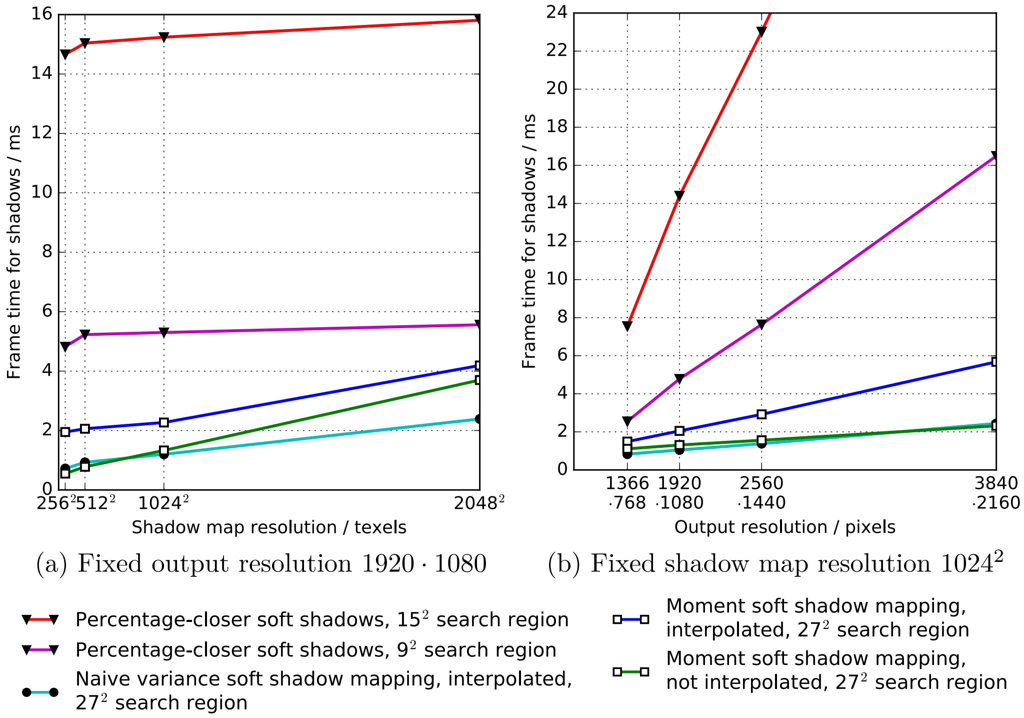


Figure 6.5: The shadow of a column rendered with moment soft shadow mapping using a $27^2$ search region. The ground is lit at an angle of incidence of 80°. Fragments that are just outside the penumbra still use a large filter size and wrong self-shadowing occurs due to an insufficiently biased fragment depth.

ing contact shadows due to a strong depth bias are hard to avoid (Fig. 6.4a bottom). While moment soft shadow mapping does not solve this problem entirely, it does diminish it by using lower bounds. The light leaking, which is not present in percentage-closer soft shadows, is weak thanks to the high precision in the summed-area table.

### 6.6.2 Run Time

Figure 6.6 shows run time measurements recorded in our Direct3D 11 implementation running on an NVIDIA GeForce GTX 970. As for filtered hard shadows, the cost per texel of the shadow map increases with the memory per texel (Fig. 6.6a). However, it is less crucial here due to the high cost per shaded fragment. Even for a $2048^2$ shadow map, all techniques using filterable shadow maps are faster than percentage-closer soft shadows with



(a) Fixed output resolution $1920 \cdot 1080$

(b) Fixed shadow map resolution $1024^2$

Figure 6.6: Frame times for rendering soft shadows with various techniques for the scene in Figure 4.9d. The frame time for rendering without shadows has been subtracted. All techniques with filterable shadow maps use 4× multisample antialiasing for the shadow map. The output always uses 4× multisample antialiasing.

a small $9^2$ search region. We provide more insights on the time it takes to generate the summed-area table in Appendix C.3.2.

The cost per shaded fragment is immense for percentage-closer soft shadows, especially when using a $15^2$ search region (Fig. 6.6b). Note that such a search region is still not large enough to generate sufficiently soft shadows over long range (Fig. 6.3a). Without interpolation during filtering, moment soft shadow mapping has a slightly lower cost per fragment than naïve variance soft shadow mapping with interpolation. Enabling interpolation increases this cost significantly, but moment soft shadow mapping is still consistently faster than percentage-closer soft shadows with a $9^2$ search region.

### 6.6.3 Conclusions

Percentage-closer soft shadows is easily the most widely used real-time technique for dynamic soft shadows of moderately large area lights. Moment soft shadow mapping is substantially faster, scales better to large search regions and large output resolutions and is less susceptible to wrong self-shadowing, which plagues percentage-closer soft shadows. The only newly introduced artifact is light leaking, which is weak due to the high precision in the summed-area table.

Therefore, we believe that it may become the new technique of choice for affordable soft shadows. A notable limitation is that the summed-area table supports exclusively rectangular light sources. This may not be ideal, e.g. for sun shadows, but the smooth penumbra regions are still plausible. Rendering soft shadows at $1920 \cdot 1080$ with a $1024^2$ shadow map in 2.1 ms used to require more compromises in terms of quality. More natural kernels may be constructed from multiple rectangles at an increased cost. While other techniques are more accurate (e.g. Wang et al. [2014]), their run time makes them uncompetitive for most real-time applications.

Chapter 7

# Single Scattering

Another common simplification in rendering is 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 visible shafts of light known as crepuscular rays. These are perceived as aesthetic and provide a tool to direct the attention of the viewer.

Unsurprisingly, industrial practitioners like to use this effect but it still comes at a high cost. Scattering occurs everywhere within the 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. 2014b] 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 7.3) or six moments (Section 7.4). Besides we demonstrate how to apply filtering during the necessary resampling step (Section 7.2).

## 7.1 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 and Dachsbacher [2010] apply more aggressive 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 boolean results of 128 shadow tests along a view ray are 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.

### 7.1.1 Prefiltered Single Scattering

Prefiltered single scattering [Klehm et al. 2014b,a] 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. 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.

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

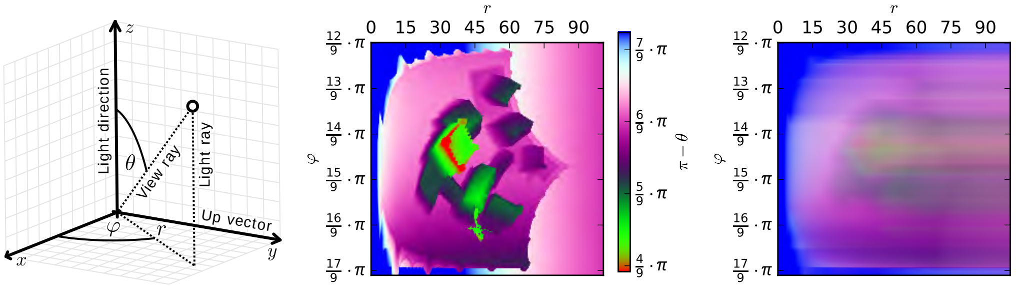
$$\exp(-\sigma_t \cdot s) \cdot L_s + f(\omega_l, \omega_p) \cdot E_l \cdot \int_0^s \exp(-\sigma_t \cdot t) \cdot V(p - t \cdot \omega_p) \, \mathrm{d}t.$$

The first summand is 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 3.1 provide means to store such functions in a way that enables the application of filters. We utilize them to turn integration of single scattering into an integration over rows of a shadow map.

To this end, 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. 2014b]. We have implemented this linear rectification but for reasons given in Section 7.2 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 7.1. Then the horizontal texture coordinate in the shadow map corresponds to the distance to the origin after projecting to the $x$-$y$-plane, $r$. This agrees with the distance between light ray and camera. For the



(a) Rectified coordinates $r$, $\varphi$ and $\theta$

(b) Rectified moment shadow map (first three channels)

(c) Rectified and prefiltered moment shadow map

Figure 7.1: Prefiltered single scattering resamples the shadow map into a coordinate system where each view ray maps to a row. Computation of weighted prefix sums over these rows effectively precomputes the single scattering result for every possible view ray.

other two coordinates, we convert to spherical coordinates. The vertical texture coordinate 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 $\varphi$ and $\theta$ 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 $\varphi$ and thus maps to a single texel. In terms of epipolar geometry $\varphi$ indexes epipolar slices containing the light direction and going through the camera. Single scattering computations for different epipolar slices are independent. To fit the shadow map tightly, bounds for $r$, $\varphi$ and $\theta$ are computed such that the entire view frustum is covered (see Appendix C.3.1). Along the dimension of $\theta$ we add a guard band to avoid light leaking. The length of the interval of values for $\theta$ is stretched by 10%.

We generate a filterable shadow map $a(u, v)$ in this coordinate system indexed with integer texel indices $u, v$. Each texel stores a representation of the visibility function along a light ray (e.g. Fourier coefficients [Klehm et al. 2014a] or power 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 $\theta$, a heuristic is required. Sophisticated heuristics have been proposed [Klehm et al. 2014a] but we simply compute the distance for the arithmetic mean of the minimal and maximal values of $\theta$. Then transmittance-weighted prefix sums are computed as

$$a_\Sigma(u,v) := \frac{\sum_{j=0}^{u} w_{j,v} \cdot a(j,v)}{\sum_{j=0}^{u} w_{j,v}}$$

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

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

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

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.

## 7.2 Rectification with Filtering

The linear rectification proposed by Klehm et al. [2014b] tends to allocate major parts of the shadow map for geometry near the camera while farther geometry is compressed. 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 with rasterization hardware. 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 (Fig. 7.5a on page 108). 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 power moments

**Algorithm 7.1** Construction of a distribution with two points of support from three power moments.
**Input:** A vector of power moments $b \in \mathbb{R}^4$ with $b_0 = 1$ and $b_2 - b_1^2 > 0$.
**Output:** A distribution $Z$ on $\mathbb{R}$ such that $\mathcal{E}_Z(\mathbf{z}^j) = b_j$ for all $j \in \{0, 1, 2, 3\}$.

```
1. Set $q_2 := b_2 - b_1^2$.
2. Set $q_1 := b_1 \cdot b_2 - b_3$.
3. Set $q_0 := -b_1 \cdot q_1 - b_2 \cdot q_2$.
4. Solve $q_2 \cdot z^2 + q_1 \cdot z + q_0 = 0$ to get solutions $z_1, z_2 \in \mathbb{R}$.
5. Set $w_2 := \frac{b_1 - z_1}{z_2 - z_1}$ and $w_1 := 1 - w_2$.
6. Return $w_1 \cdot \delta_{z_1} + w_2 \cdot \delta_{z_2}$.
```

back into a distribution of depth values because we need to distort depth in a non-linear fashion. It is appropriate to expect simple distributions because we are working with small filter regions. In most relevant cases the filter region will not cover more than two different surfaces.

In Section 4.1.1 we explained how to reconstruct a distribution with three depth values $z_0, z_1, z_2$ from four power moments where $z_0 = z_f$ is prescribed. This leaves us with the question how to choose $z_0$. To avoid an arbitrary choice and to obtain a more efficient solution we let $z_0$ go to infinity. As this happens, $w_0$ approaches zero and we can discard the depth value $z_0$. The remaining distribution $w_1 \cdot \delta_{z_1} + w_2 \cdot \delta_{z_2}$ is still compatible with the power moments $b_0, b_1, b_2, b_3$. Only the fourth power moment $b_4$ does not match. Under the assumption of two or fewer surfaces at nearly 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.

The described distribution is constructed by Algorithm 7.1. It fails for inputs with non-positive variance $\sigma^2 := q_2 = b_2 - b_1^2$ but this case is adequately handled by simply returning $\delta_{b_1}$.

**Proposition 7.1.** *Given a valid input, Algorithm 7.1 works correctly.*

*Proof.* We recall from Proposition 2.12 that a distribution with two points of support corresponds to a singular $3 \times 3$ Hankel matrix. Implicitly, the algorithm computes the missing fourth power moment $b_4$ such that the Hankel matrix is singular and then proceeds like Algorithm 2.1.

For all $b_4 \in \mathbb{R}$ the multilinearity of the determinant implies

$$\begin{aligned}\det B\begin{pmatrix}b\\b_4\end{pmatrix} &= b_4\cdot\det\begin{pmatrix}b_0 & b_1 & 0\\ b_1 & b_2 & 0\\ b_2 & b_3 & 1\end{pmatrix} + \det B\begin{pmatrix}b\\0\end{pmatrix}\\ &= b_4\cdot(b_2-b_1^2)+\det B\begin{pmatrix}b\\0\end{pmatrix}.\end{aligned}$$

Thus, we can choose the unique $b_4 \in \mathbb{R}$ that makes the Hankel matrix singular and find ourselves in the situation of Proposition 2.12.

To prove that $q := (q_0, q_1, q_2)^\mathsf{T}$ lies in the kernel, we observe that the first two rows of the Hankel matrix are linearly independent due to

$$\det\begin{pmatrix}b_0 & b_1\\ b_1 & b_2\end{pmatrix} = b_2 - b_1^2 > 0.$$

Thus, the third row is a linear combination of the first two rows and it suffices to show that $q$ is orthogonal to the first two rows:

$$\begin{aligned}(1, b_1, b_2)\cdot q &= (-b_1\cdot q_1 - b_2\cdot q_2) + b_1\cdot q_1 + b_2\cdot q_2 = 0\\ (b_1, b_2, b_3)\cdot q &= (b_2 - b_1^2)\cdot q_1 + (b_3 - b_1\cdot b_2)\cdot q_2 = q_2\cdot q_1 - q_1\cdot q_2 = 0\end{aligned}$$

The remainder of Algorithm 7.1 is completely analogous to Algorithm 2.1 and correctness follows from Proposition 2.12. □

Once we have obtained the distribution, we convert its depths $z_1, z_2$ to inclinations $\theta_1, \theta_2$ as described in Section 7.1.1 and normalize to the interval $[-1, 1]$ clamping out-of-range values. Then we convert both values to vectors of moments and linearly combine them using the weights $w_1, w_2$. The result is stored in $a(u, v)$. At this point we can also generate more than four power moments or other general moments.

Using this scheme is entirely optional. Our implementation creates the rectified, filterable shadow map $a(u, v)$ using a pixel shader. When filtering is enabled, the pixel shader takes a filtered sample from a moment shadow map, otherwise it just loads a texel from a common shadow map. The sample from the moment shadow map is only slightly more expensive (see Section 7.5.2).

## 7.3 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 power moments with the usual quantization transform (see Section 4.1.4). When it comes to the computation of the shadow intensity during evaluation of single scattering, we can proceed as for hard shadows using Algorithm 4.2.

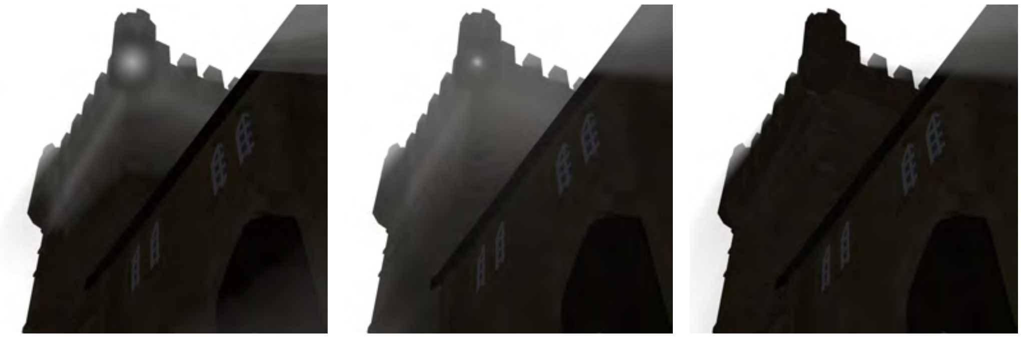
However, some assumptions made for surface shadows are inadequate for single scattering. For surface shadows we always underestimate the shadow intensity to avoid surface acne. For single scattering this is generally not necessary but we may want to avoid other artifacts. Our solution is to take a weighted combination of the sharp lower bound and the sharp upper bound. By Theorem 4.1 the upper bound is obtained by adding $w_0$ from Algorithm 4.1 to the lower bound (see Figure 4.1 on page 57). Thus, the overhead for computing both bounds is small.

### 7.3.1 Adaptive Overestimation

Having sharp upper and lower bounds, we need a weight $\beta \in [0, 1]$ to interpolate between the two. For $\beta = 0$ single scattering is underestimated, for $\beta = 1$ it is overestimated. A simple approach would set $\beta = \frac{1}{2}$ such that the worst-case error is minimal. However, the weight can be set arbitrarily per pixel and a more sophisticated choice avoids artifacts.

Figure 7.2a shows an artifact of prefiltered single scattering [Klehm et al. 2014a]. Light leaking is strongly increased at the epipole (i.e. when looking into the light). The inclination of the corresponding view ray is $\theta = \pi$ which corresponds to a minimal depth in the rectified shadow map. Thus, no occluder can have a smaller depth. Near the epipole, inclinations are still large and the leaking only falls off slowly.

If we use moment shadow maps with $\beta = \frac{1}{2}$, this problem persists (Fig. 7.2b) but if we underestimate the single scattering it vanishes as expected (Fig. 7.2c). On the other hand, a constant choice of $\beta = 0$ degrades the approximation quality elsewhere. In particular, at the antipodal point of the epipole, the inclination reaches $\theta = 0$ and no depth values in the rectified shadow map can be greater than the fragment depth. Thus, underestimation of the single scattering leads to no single scattering, which is likely incorrect.



(a) Convolution shadow map, $32 \cdot 8$ bit [Klehm et al. 2014a]

(b) Moment shadow map, $4 \cdot 16$ bit, $\beta = \frac{1}{2}$

(c) Moment shadow map, $4 \cdot 16$ bit, $\beta = 0$

Figure 7.2: A view of the pinnacles of a tower. The directional light is hidden behind it but when single scattering is not underestimated, the light is clearly visible due to leaking artifacts for large $\theta$.

To avoid both artifacts while maintaining the best approximation quality in intermediate cases, we set $\beta$ dependent on the direction of the view ray $\omega_p$. The weight $\beta$ is supposed to be zero for $\omega_p = -\omega_l$, one for $\omega_p = \omega_l$ and near the epipoles it should vary slowly. In our experiments, we found that the following choice reliably removes the artifacts discussed above while providing a smooth and plausible result:

$$\beta = \frac{1 + \omega_l^\mathsf{T} \cdot \omega_p}{2}$$

## 7.4 Prefiltered Single Scattering with Six Moments

Four moment shadow mapping works well for surface shadows, because individual points rarely receive shadow from more than two different surfaces. Prefiltered single scattering on the other hand computes the average shadow received by an entire view ray. Such a view ray may be shadowed by many different surfaces at different depths. Overall, depth distributions are significantly more complex. Using only four power moments for their representation is often insufficient.

Fortunately, Algorithm 4.1 is formulated for an arbitrary even number of power moments. As a reasonable tradeoff between computational complexity and quality, we investigate the use of six power moments. Similar to our derivation in Section 4.1, the robust implementation of this method is non-trivial and we now discuss the necessary steps to avoid numerical noise. Shader code for all steps is provided in Appendix C.3.3.

### 7.4.1 Computation of Roots

Solving the $4 \times 4$ linear system $B(b) \cdot q = \hat{\mathbf{b}}(z_f)$ using a Cholesky decomposition still works well. Next we need to solve the cubic equation $\sum_{j=0}^{3} q_j \cdot z^j = 0$ for $z$. We have experimented with various iterative and closed-form solutions and settled for a variation of a closed-form solution proposed by Blinn [2007].

The algorithm presented in the article uses two different branches for computation of the roots of minimal and maximal magnitude to avoid cancellation. In our application, we found that this overhead is unnecessary. Using one of the two branches to compute all three roots yields results that are free of artifacts. Other closed-form solutions suffered from artifacts for $|q_3| \ll 1$ and iterative solutions had a high computational overhead. Among all attempted solutions, the one based on Blinn's work is the fastest.

### 7.4.2 Computation of Bounds

The final step is to approximate the average visibility, which is proportional to the strength of single scattering. It is a linear combination of the weights $w_0, \ldots, w_3$ subject to the moment constraints

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ z_0 & z_1 & z_2 & z_3 \\ z_0^2 & z_1^2 & z_2^2 & z_3^2 \\ z_0^3 & z_1^3 & z_2^3 & z_3^3 \end{pmatrix} \cdot \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}.$$

The weights in the linear combination are

$$v_0 := \beta \quad \text{and} \quad \forall l \in \{1,2,3\} : \; v_l := \begin{cases} 0 & \text{if } z_f > z_l, \\ 1 & \text{if } z_f \leq z_l. \end{cases}$$

Written as a dot product, the linear combination is

$$\begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{pmatrix}^{\mathsf{T}} \cdot \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}^{\mathsf{T}} \cdot \underbrace{\begin{pmatrix} 1 & z_0 & z_0^2 & z_0^3 \\ 1 & z_1 & z_1^2 & z_1^3 \\ 1 & z_2 & z_2^2 & z_2^3 \\ 1 & z_3 & z_3^2 & z_3^3 \end{pmatrix}^{-1} \cdot \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix}}_{=:u}.$$

Since the matrix in this last expression is a Vandermonde matrix, the vector $u \in \mathbb{R}^4$ holds the coefficients of the interpolation polynomial $\sum_{j=0}^{3} u_j \cdot z^j$ taking value $v_l$ for $z = z_l$ where $l \in \{0, 1, 2, 3\}$. We construct its Newton form using divided differences and then transform back to the canonical basis of polynomials [Greenbaum and Chartier 2012, p. 181 ff.]. This works efficiently and sufficiently robust for our purposes.

### 7.4.3 Quantization and Biasing

In presence of complex depth distributions, even perfect accuracy in all computations cannot yield a perfect reconstruction. Therefore, we expect the effect of strong biasing to be less drastic for single scattering and using moment shadow maps with low precision is attractive.

Again, rounding errors should be diminished by means of an affine transform that is applied to the power moments before storing them in the moment shadow map. As in Section 4.1.4 we use numerical optimization to determine it. We have experimented with general transforms and with transforms that operate on odd and even moments separately. The best found transform belongs to the latter category. It increases the entropy of the stored data by 12.5 bits per texel and is given by

$$\Theta_6^\star(b^+) = \begin{pmatrix} \begin{pmatrix} 2.5 & -1.87499864 & 1.26583039 \\ -10 & 4.20757543 & -1.47644883 \\ 8 & -1.83257679 & 0.71061660 \end{pmatrix}^\mathsf{T} \cdot \begin{pmatrix} b_1 \\ b_3 \\ b_5 \end{pmatrix} + \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \end{pmatrix} \\ \begin{pmatrix} 4 & 9 & -0.57759806 \\ -4 & -24 & 4.61936648 \\ 0 & 16 & -3.07953907 \end{pmatrix}^\mathsf{T} \cdot \begin{pmatrix} b_2 \\ b_4 \\ b_6 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0.018888946 \end{pmatrix} \end{pmatrix}.$$

An optimal biasing is determined as for the case with four moments (see Section 4.1.3 and Appendix B.1). The vector of biasing moments is

$$b^\star := (0,\ 0.5566,\ 0,\ 0.489,\ 0,\ 0.47869382)^\mathsf{T}.$$

For storage of the moments in our Direct3D 11 implementation we use two textures with either three channels storing 10-bit fixed-precision numbers (64 bits per texel, $\alpha_b = 4 \cdot 10^{-3}$), two and four channels storing 16-bit fixed-precision numbers (96 bits per texel, $\alpha_b = 8 \cdot 10^{-5}$) or three channels storing single-precision floating point numbers (192 bits per texel, $\alpha_b = 5 \cdot 10^{-6}$).

## 7.5 Results and Discussion

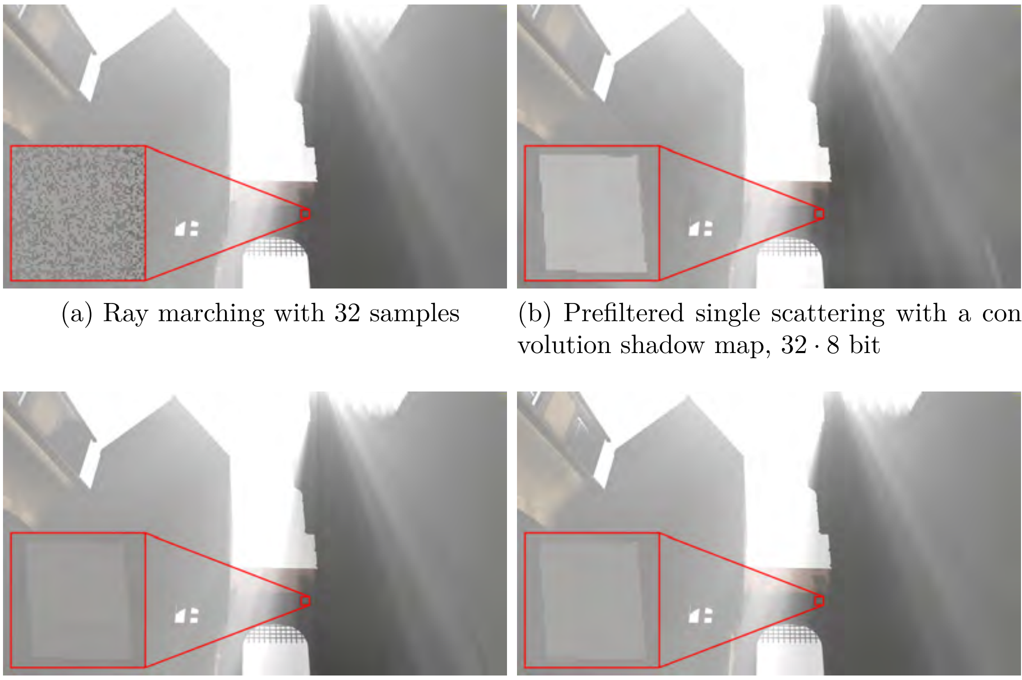
We apply all scattering techniques in a deferred rendering pass with the depth buffer as input. Multisample antialiasing is disabled. Note that prefiltered single scattering is fast enough to be applied during forward rendering thus avoiding problems with multisampling and transparencies. Though, we have not tested this. Unless stated otherwise, the shadow map resolution is $1024^2$. The rectified shadow map is generated from the shadow map for surface shadows and has the same resolution.

### 7.5.1 Qualitative Evaluation

For comparison we have implemented ray marching with equidistant, jittered samples and prefiltered single scattering using convolution shadow maps with 32 real coefficients [Klehm et al. 2014a]. The compute shader generating the transmittance-weighted prefix sums is described in detail in Appendix C.3.2. For common shadow maps we use 16-bit textures and for convolution shadow maps we use 8 bits per channel.

Figure 7.3 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, noise is acceptable using 32 ray samples (Fig. 7.3a). Techniques based on prefiltered single scattering do not produce noise but more systematic errors. When using prefiltered single scattering with convolution shadow maps (Fig. 7.3b), ringing due to the truncation of the Fourier series is strong. Techniques based on moment shadow mapping do not exhibit ringing (Figs. 7.3c, 7.3d). An artifact that is shared by all techniques with prefiltering is magnified (Figs. 7.3b, 7.3c, 7.3d). 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. Although, the differences are subtle, using six moments gives the best result in this example (Fig. 7.3d). Note that Figures 7.3a, 7.3b and 7.3c exhibit some surface acne from percentage-closer filtering which is not present in Figure 7.3d because the available moment shadow map is used for shadowing.

Figure 7.4 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. 7.4a). Ringing in prefiltered single scattering with convolution shadow maps leads to overly dark concentric circles that change with camera movements (Fig. 7.4b). A characteristic artifact of prefiltered single scattering with moment shadow maps are exces-



(a) Ray marching with 32 samples

(b) Prefiltered single scattering with a convolution shadow map, $32 \cdot 8$ bit

(c) Prefiltered single scattering with four moments and without filtering, 64 bit

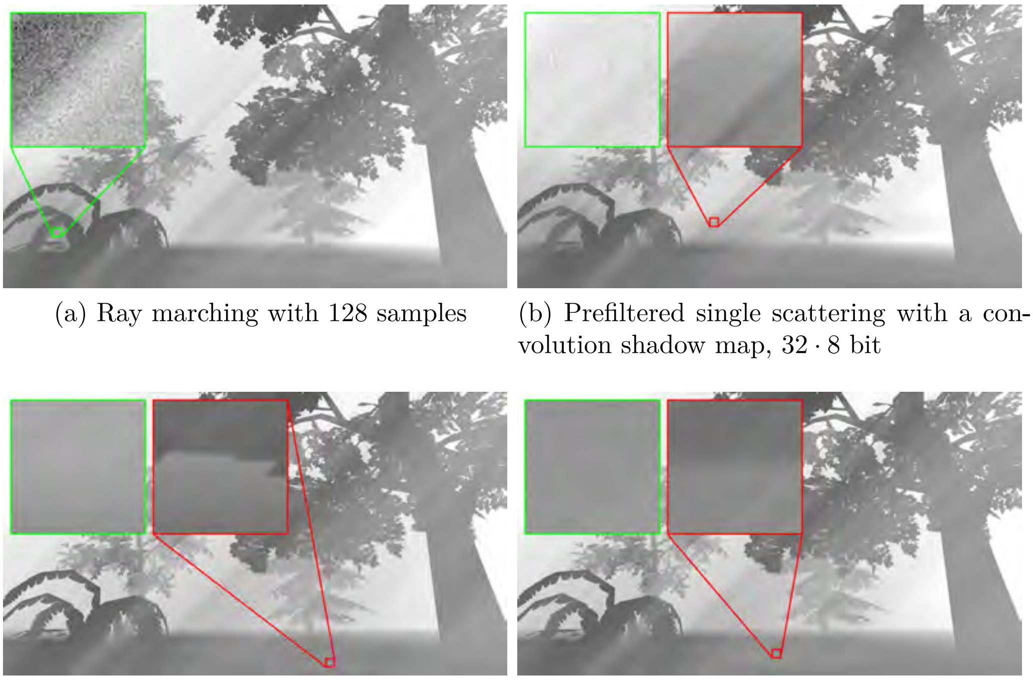
(d) Prefiltered single scattering with six moments and filtering, 64 bit

Figure 7.3: A comparison of various single scattering techniques. Note the ringing produced by convolution shadow maps and the window that appears brighter than the surrounding wall (magnified). Contrasts in the magnified insets are stretched by a factor of four.

sively sharp boundaries of shadow volumes (Fig. 7.4c). This occurs because the approximation quality can change quickly as depth distributions become more complex. Using six moments reduces this artifact heavily (Fig. 7.4d). It is also slightly diminished by a greater moment bias $\alpha_b$.

Figure 7.5 demonstrates the positive effect of filtering during resampling in an extreme case. Without filtering crepuscular rays exhibit fine structures, which change rapidly as the camera moves or rotates (Fig. 7.5a). Especially for slowly moving cameras this aliasing can be a very distracting artifact. Applying bilinear filtering to a moment shadow map during resampling (Section 7.5b) makes the shadows lose some detail but aliasing is reduced to a point where it is unproblematic (Fig. 7.5b).

Figure 7.6 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



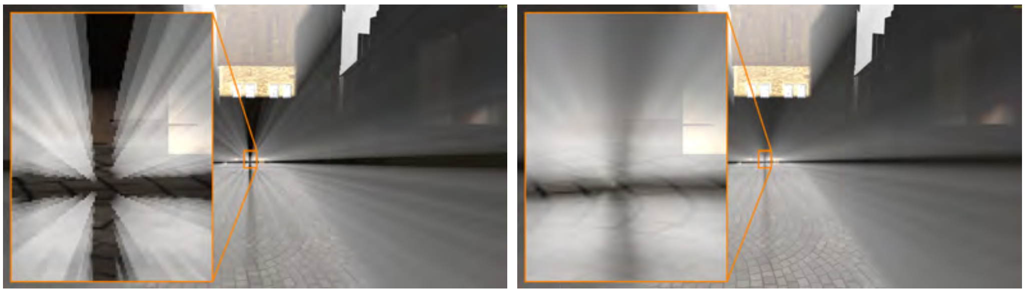
(a) Ray marching with 128 samples

(b) Prefiltered single scattering with a convolution shadow map, $32 \cdot 8$ bit

(c) Prefiltered single scattering with four moments and without filtering, 64 bit

(d) Prefiltered single scattering with six moments and without filtering, 64 bit

Figure 7.4: A challenging scenario for single scattering techniques involving shadows of trees. The scene itself is shaded black. Various artifacts are shown in magnified insets (red) next to the ray marching result (green). Contrasts in the magnified insets are stretched by a factor of four.

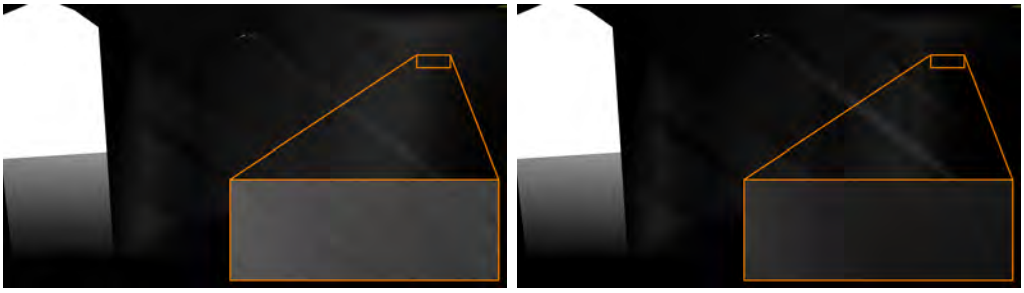


(a) Not filtered

(b) Filtered

Figure 7.5: A view into the shadow of a gate rendered using prefiltered single scattering with six moments (64 bit). Resampling a common shadow map without filtering yields heavy aliasing that is not temporally stable. Resampling a four moment shadow map with filtering (Section 7.2) produces a much smoother result.

Figure 7.6: 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. Magnified insets stretch contrasts by a factor of four.



(a) Six moments in $6 \cdot 10$ bit, $\alpha_b = 4 \cdot 10^{-3}$ (b) Six moments in $6 \cdot 16$ bit, $\alpha_b = 8 \cdot 10^{-5}$

Figure 7.7: An indoors scene where light leaks through walls. The scene is shaded black. Note that the use of 10 bits per power moment leads to quantization noise and that the increased moment bias $\alpha_b$ strengthens light leaking.

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.

Figure 7.7 demonstrates the slight quality improvement obtained by using 96 rather than 60 bits (plus four unused bits) for storage of six moments. The high moment bias required with 60 bits increases light leaking, which is problematic for indoors environments with thin walls. Besides, quantization errors manifest as splotches in dark regions. In some scenarios, the higher quality resulting from 96 bits may be required but in most situations the lower contrast of single scattering and the surface shading will cover up the artifacts.

### 7.5.2 Run Time

We measure frame times on an NVIDIA GeForce GTX 970 and show the results in Figure 7.8. The frame times for rendering without single scattering have been subtracted. Since the single scattering is applied in a deferred pass, this means that the shown timings are almost entirely scene-independent.

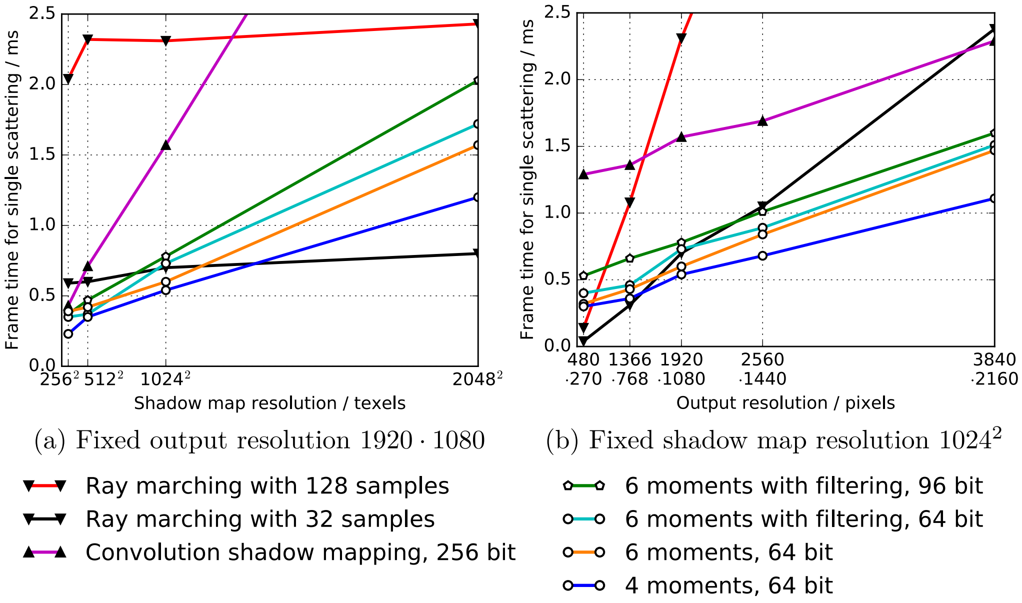
As expected, the run time of ray marching only depends weakly on the shadow map resolution because no post-processing is applied to the common shadow map (Fig. 7.8a). On the other hand, the cost of shadow map sampling increases rapidly with the output resolution, especially when using 128 samples (Fig. 7.8b). At this number of samples, ray marching can only compete with prefiltered single scattering with four or six moments at very low resolutions such as $480 \cdot 270$. Note that these resolutions may be sufficient when a bilateral upscaling is used but undersampling artifacts have to be expected.

Prefiltered single scattering with convolution shadow maps gets more expensive rapidly with growing shadow map resolution because it stores 256 bits per texel. When using moment shadow maps, the cost per texel is much lower (Fig. 7.8a). Using six moments stored in 64 bits is slightly more costly than using four moments stored in 64 bits, which is likely due to the use of two textures for six moments in our implementation. Filtering during resampling adds to this cost slightly and so does the use of 96 bits per texel.

Looking at the cost per pixel in the output, we observe that it is similar for all techniques using six moments but lower for prefiltered single scattering with four moments (Fig. 7.8b). This is a strong indication that arithmetic operations are the bottleneck for techniques using six moments. Still, the cost is moderate and even at a resolution of $3840 \cdot 2160$ the techniques finish in about 1.5 ms.

### 7.5.3 Conclusions

Prefiltered single scattering with moment shadow maps outperforms the approach with convolution shadow maps clearly. It is faster, does not suffer from ringing and enables an explicit control over leaking artifacts by interpolating between lower and upper bounds. Compared to ray marching, the high performance at large output resolutions means that no upscaling is needed. These traits make our techniques highly attractive for performance-sensitive real-time applications.



(a) Fixed output resolution $1920 \cdot 1080$

(b) Fixed shadow map resolution $1024^2$

Figure 7.8: 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.

In most cases, the variants with six moments stored in 64 bits should provide the best tradeoff between quality and run time. When single scattering is only used as a subtle effect, four moments may provide sufficient quality. If a moment shadow map for the scene is already available, the overhead for filtering is small and it should be used. Otherwise, it depends upon the scene whether the reduced aliasing justifies the cost for creation of a moment shadow map.

# Part II

# Fast Transient Imaging

# Chapter 8

# Fast Transient Imaging

Many consumers have access to amplitude modulated continuous wave (AMCW) lidar systems such as Microsoft Kinect for Xbox One or the depth sensors by pmdTechnologies used e.g. in mobile devices with Google's Project Tango. Typically, these cameras are used for range imaging where the time of flight that light takes from an active illumination into the scene and back to the camera is measured indirectly. The active illumination is modulated with a high-frequent, periodic signal and the phase shift of the signal scattered back to the sensor is measured.

In presence of global illumination effects, the assumption of a unique time of flight no longer holds since light may reach points in the scene on many different paths of different length. Assuming existence of a unique phase shift during reconstruction leads to systematic distortions in range images, which are often far greater than precision errors due to sensor noise. This effect is known as multipath interference.

Transient images model this complex behavior more completely. In such an image each pixel stores a time-resolved impulse response indicating how much light returned after a particular time of flight. This enables applications such as separation of direct and indirect illumination [Wu et al. 2014] or non-line-of-sight imaging [Velten et al. 2012].

It has been shown that an AMCW lidar system can be used to estimate a transient image using measurements at many modulation frequencies [Heide et al. 2013]. While this approach is among the most cost-efficient ways to measure transient images, it has difficulties reconstructing complex impulse responses, measurement takes a minute and reconstruction takes even longer.

Our key observation is that AMCW lidar systems can be configured such that they measure trigonometric moments of the impulse response. We then use two efficient closed-form solutions, which reconstruct an impulse response matching the given trigonometric moments exactly. Our approach drastically reduces the number of required measurements and still successfully reconstructs complex impulse responses. The technique scales well from measurement of high-quality transient images to quick heuristic measurements. The latter is useful for reduction of multipath interference in range imaging.

The first used solution is known as maximum entropy spectral estimate [Burg 1975] (see Section 8.2.2). It reconstructs a smooth, positive density function matching given trigonometric moments $c_0, c_1, \ldots, c_m \in \mathbb{C}$. If the ground truth is close to an impulse response with $m$ or fewer points of support, so is the maximum entropy spectral estimate. In the limit case, which we already discussed in Section 2.2, the Pisarenko estimate (see Section 8.4.1) reconstructs the $m$ points of support of the ground truth perfectly from the trigonometric moments $c_1, \ldots, c_m \in \mathbb{C}$. In more difficult situations, the maximum entropy spectral estimate models the uncertainty by broad peaks in the reconstructed density. Closed-form bounds for its error are described in Section 8.4.2.

To measure trigonometric moments, we require sinusoidal modulation and we describe a novel method to accomplish this on our prototype hardware in Section 8.3.1. Using this method, we are capable of measuring up to 18.6 transient images of reasonable quality per second as demonstrated in Section 8.5.3. Such images can also be used for improved range imaging as demonstrated in Section 8.5.2. Higher quality measurements can take a few seconds because it is advisable to average many captures for an improved signal-to-noise ratio.

## 8.1 Related Work

In recent years transient imaging has been introduced as an exciting new imaging modality. Such images can be understood as video recording the return of light to a camera at an extreme frame rate when the scene is lit by an infinitesimally short light pulse. The first general hardware setup for their measurement uses a femtosecond laser and a streak camera [Velten et al. 2011, 2013]. The laser sends repeated short light pulses into the scene while the streak camera directs light returning at different times to different rows of the image sensor. This way a transient image can be captured one row at a time with a temporal resolution around 2 ps. Capture takes roughly one hour. Later work uses interferometry and several hours of capture time to push temporal resolution to 33 fs within a small capture volume [Gkioulekas et al. 2015].

Transient images add a fundamentally new dimension to images, thus enabling new applications. Velten et al. [2012] reconstruct geometry solely by analyzing the light it reflects onto a diffuse wall. Similarly, Naik et al. [2011] reconstruct surface reflectance around a corner. Wu et al. [2014] separate images into direct illumination, subsurface scattering and indirect illumination by analyzing impulse responses.

While these applications demonstrate the usefulness of transient images, they are limited by the high cost and long measurement times of the involved hardware. A drastically faster and more cost-efficient approach uses AMCW lidar systems [Heide et al. 2013]. These cameras apply a modulation signal at the light source and the sensor. Effectively this means that they measure the correlation of a transient image with a time-dependent, periodic signal. Using measurements at many different modulation frequencies, the authors reconstruct the transient image by solving an inverse problem with soft, linear constraints enforcing compatibility with the measurements and additional temporal and spatial regularization priors. The authors capture a transient image within a minute but the regularization priors tend to lose high-frequency temporal details and reconstruction takes several minutes.

Subsequent works explore various measurement procedures, priors and reconstruction algorithms. Kadambi et al. [2013] use a broadband modulation and sample it at many phase shifts. The arising inverse problem is solved with various linear and non-linear priors. Kirmani et al. [2013] assume sinusoidal modulation at several multiples of a common base frequency. These measurements are used as soft constraint to reconstruct impulse responses as distributions with two points of support. Lin et al. [2014] use a similar input but employ an inverse Fourier transform with subsequent corrective post-processing. Bhandari et al. [2014b] use measurements at many frequencies and orthogonal matching pursuit to estimate a distribution with few points of support. Qiao et al. [2015] use a logarithmic prior to reward sparsity. Kadambi et al. [2016] consider measurements as a function of frequency and derive times of flight from the frequencies in this signal. Bhandari et al. [2014a] present a method using sinusoidal modulation at $2 \cdot m + 1$ frequencies to reconstruct a distribution with $m \in \mathbb{N}$ points of support. With measurements from Microsoft Kinect for Xbox One they successfully separate two returns using measurements at 21 frequencies.

At the other end of the spectrum there are works using far fewer measurements to reconstruct range images. In this context reconstruction of impulse responses only servers as intermediate step to model multipath interference. Using measurements at two modulation frequencies, Dorrington et al. [2011] fit a distribution with two points of support to the measurements using non-linear optimization. Godbaz et al. [2012] use measurements at four frequencies to estimate parameters for a similar model in closed form. Using linear programming in a manner that is quite similar to Algorithm 3.1 on page 46, Freedman et al. [2014] minimize an $\mathcal{L}^1$-prior while allowing a tolerance on the measurements. For application in real time the authors store the results in a four-dimensional look-up table to process three frequency measurements quickly. Gupta et al. [2015] observe that diffuse multipath interference tends to cancel out at high frequencies and thus propose to reconstruct range from few measurements at high frequency.

The above works all capture transient images using an active illumination to generate a repetitive event. Transient images of non-repetitive events have been recorded in a single capture using compressed sensing, although this approach sacrifices spatial resolution for temporal resolution [Gao et al. 2014]. When range imaging is the primary concern, it is also possible to reduce multipath interference without additional measurements. To this end, diffuse interreflections in the scene are modeled explicitly and the estimated multipath interference is then subtracted from the measurement [Fuchs 2010; Jimenez et al. 2012]. While this saves measurement time, it requires substantial post-processing time. Ground truth data for transient imaging can be generated with specialized Monte Carlo renderers [Jarabo et al. 2014].

## 8.2 Reconstruction of Impulse Responses

In the present section we demonstrate how to cast the inverse problem encountered in transient imaging with AMCW lidar systems into a trigonometric moment problem. We propose the maximum entropy spectral estimate as solution and analyze its properties. Though, before dealing with the inverse problem, we need to describe the forward model.

### 8.2.1 Signal Formation Model

Suppose $G$ is a finite measure on $\mathbb{R}$ modeling the impulse response for a single pixel in a transient image. The measure $G([\alpha, \beta])$ tells how much of the returning light has a time of flight in $[\alpha, \beta] \subset \mathbb{R}$. When diffuse interactions scatter light, the impulse response is adequately modeled by a density function as in Definition 2.2 on page 22 (Figs. 8.1a, 8.1b). In situations where specular interactions dominate, a measure with finite support as in Definition 2.1 may be more adequate. Finite measures capture both situations and combinations thereof in a single notation.

The active illumination of the AMCW lidar system is modulated by a $T$-periodic signal $\boldsymbol{s}_i(\tau)$ and the pixel receives the convolved signal

$$(\boldsymbol{s}_i * G)(\sigma) = \int \boldsymbol{s}_i(\sigma - \tau)\,\mathrm{d}G(\tau).$$

The sensor is modulated with another $T$-periodic signal $\boldsymbol{s}_s(\tau)$. Throughout the exposure time it integrates over the resulting signal $(\boldsymbol{s}_i * G) \cdot \boldsymbol{s}_s$. For simplicity we assume that it integrates exactly one period. Thus, the measurement at the pixel is

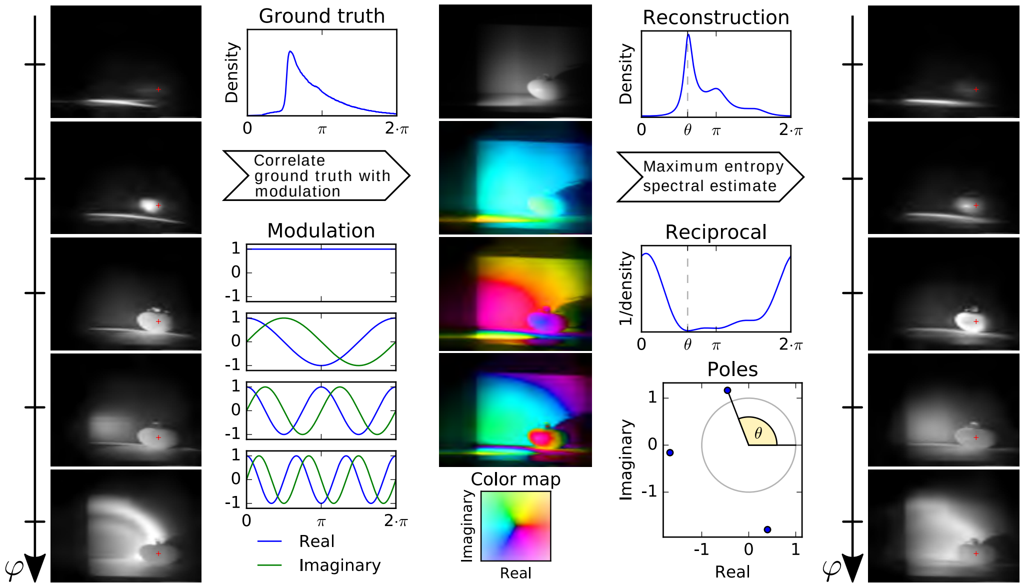
$$\begin{aligned}\frac{1}{T} \cdot \int_0^T (\boldsymbol{s}_i * G)(\sigma) \cdot \boldsymbol{s}_s(\sigma)\,\mathrm{d}\sigma &= \frac{1}{T} \cdot \int_0^T \int \boldsymbol{s}_i(\sigma - \tau)\,\mathrm{d}G(\tau) \cdot \boldsymbol{s}_s(\sigma)\,\mathrm{d}\sigma \\ &= \int \frac{1}{T} \cdot \int_0^T \boldsymbol{s}_i(\sigma - \tau) \cdot \boldsymbol{s}_s(\sigma)\,\mathrm{d}\sigma\,\mathrm{d}G(\tau) \\ &= \int \boldsymbol{s}_i \star \boldsymbol{s}_s(\tau)\,\mathrm{d}G(\tau)\end{aligned}$$

where $\boldsymbol{s}_i \star \boldsymbol{s}_s$ denotes periodic cross-correlation.

We conclude that the sensor measures the correlation between the impulse response $G$ and the effective modulation $\boldsymbol{s}_e := \boldsymbol{s}_i \star \boldsymbol{s}_s$ (Fig. 8.1b). At this point, it is interesting to note that all information that can possibly be captured by an AMCW lidar system is contained in the impulse response $G$. Vice versa, many AMCW lidar systems allow customization of the modulation and thus a lot of information about $G$ can be inferred by using many different modulations. Therefore, transient imaging and AMCW lidar are closely linked imaging modalities.

Our method assumes measurements with a specific set of modulation functions. We now introduce these assumptions and later demonstrate their practical implementation in Section 8.3. First we fix a base frequency $f \in \mathbb{R}$. In most of our experiments this is 23 MHz. Furthermore, we fix the number of non-zero frequencies $m \in \mathbb{N}$ to measure at. This is one of two major factors allowing tradeoffs between capture time and quality. Our experiments use $m \in \{3, \ldots, 8\}$. Now for all $j \in \{0, \ldots, m\}$ we sequentially use the effective modulation functions

$$\boldsymbol{s}_e(\tau) = \cos(j \cdot 2 \cdot \pi \cdot f \cdot \tau) \quad \text{and} \quad \boldsymbol{s}_e(\tau) = \sin(j \cdot 2 \cdot \pi \cdot f \cdot \tau).$$



(a) Transient image (b) Signal formation (c) Moment images (d) Reconstructed impulse response (e) Result

Figure 8.1: A schematic visualization of AMCW lidar signal formation and our signal reconstruction. A lit scene implicitly defines a transient image resolved in time of flight $\tau$ or equivalently phase $\varphi = 2 \cdot \pi \cdot f \cdot \tau$ (8.1a, data provided by Velten et al. [2013]). Per pixel, an AMCW lidar system correlates this signal with $m + 1 = 4$ periodic modulation functions (8.1b). This yields $m$+1 images holding complex trigonometric moments per pixel (8.1c). These images are the only input of our closed-form reconstruction. The signal is reconstructed as continuous density (8.1d top) which is the reciprocal of a Fourier series (8.1d middle) and the absolute, squared reciprocal of a polynomial of degree $m$ on the unit circle (8.1d bottom). Reconstruction per pixel yields the full time-resolved transient image (8.1e).

For convenience let $\varphi := 2 \cdot \pi \cdot f \cdot \tau$ denote the phase of $\tau$ with respect to the base frequency $f$. Then $\boldsymbol{s}(\varphi) := \boldsymbol{s}_e\left(\frac{\varphi}{2\cdot\pi\cdot f}\right)$ is a $2 \cdot \pi$-periodic version of the effective modulation. The base frequency should be chosen such that its wavelength is longer than all interesting light paths. Otherwise phase ambiguity arises which we model by defining the measure

$$F(\mathbb{A}) := \sum_{l=-\infty}^{\infty} G\left(\frac{\mathbb{A} + l \cdot 2 \cdot \pi}{2 \cdot \pi \cdot f}\right)$$

for all measurable sets $\mathbb{A} \subseteq (0, 2 \cdot \pi]$. $F$ measures sets of phases rather than sets of times of flight. Since all our modulation functions have period $T := f^{-1}$, we can only reconstruct $F$ but not the time-resolved $G$.

To further simplify notions, we combine the two real measurements at frequency $j \in \{0, \ldots, m\}$ into a single complex phasor (Figs. 8.1b, 8.1c):

$$\begin{aligned} c_j &:= \int \cos(j \cdot \varphi)\,\mathrm{d}F(\varphi) + i \cdot \int \sin(j \cdot \varphi)\,\mathrm{d}F(\varphi) \qquad (8.1) \\ &= \int \exp(i \cdot j \cdot \varphi)\,\mathrm{d}F(\varphi) \in \mathbb{C} \end{aligned}$$

Rewritten like this, the definition of the AMCW lidar measurement $c_j$ agrees exactly with Definition 2.9 on page 26, which introduces trigonometric moments. We recall the definition of the trigonometric moment-generating function $\mathbf{c} : \mathbb{R} \to \mathbb{C}^{m+1}$:

$$\forall j \in \{0, \ldots, m\},\ \varphi \in \mathbb{R} :\ \mathbf{c}_j(\varphi) := \exp(i \cdot j \cdot \varphi)$$

It is a vector-valued function encompassing all used modulations as function of $\varphi$. Then

$$c := \int \mathbf{c}(\varphi)\,\mathrm{d}F(\varphi) \in \mathbb{C}^{m+1} \qquad (8.2)$$

is a vector holding all the measurements defined in Equation (8.1) and simultaneously it is the vector of trigonometric moments for $F$.

### 8.2.2 Reconstruction via Trigonometric Moments

With the above derivation, we are in a position to apply the theory of moments to reconstruct impulse responses. To make the most of our measurements, we incorporate them into the reconstruction as a hard constraint; that is, we only admit distributions $F$ fulfilling Equation (8.2). However, this does not lead to a well-posed problem by itself. In general, many different distributions share the trigonometric moments $c$.

We are left with uncertainty about the exact temporal distribution of light. A good reconstruction should reflect this uncertainty. It should not localize density unless the data enforces such a localization. Any other behavior would be arbitrary and could lead to wrong conclusions. We implement this requirement by asking for the distribution of minimal Burg entropy.

**Definition 8.1** (Burg entropy [Burg 1975])**.** Let $F_D$ be a finite measure on $(0,\,2\cdot\pi]$ with density $D:\;(0,\,2\cdot\pi]\to\mathbb{R}$. Then the Burg entropy[1] of $F_D$ is defined by

$$\mathcal{H}_{\text{Burg}}(F_D):=\mathcal{H}_{\text{Burg}}(D):=\int_0^{2\cdot\pi}-\log D(\varphi)\,\mathrm{d}\varphi.$$

For measures which do not have a density, Burg entropy is not defined.

By minimizing the Burg entropy, we punish small densities heavily because $-\log D\to\infty$ as $D\to 0$. On the other hand large densities are rewarded only slightly because $\log D$ grows slowly. In terms of minimal Burg entropy, a measure is optimal if it achieves moderate densities over large intervals. In this sense, uncertainty is rewarded.

This prior is of particular interest to us because it admits a closed-form solution to the trigonometric moment problem.

**Theorem 8.2** (Maximum entropy spectral estimate [Burg 1975])**.** *We recall from Definition 2.10 on page 26 that*

$$C(c)=(c_{j-k})_{j,k=0}^{m}=\begin{pmatrix}c_0 & \overline{c_1} & \cdots & \overline{c_m}\\ c_1 & c_0 & \ddots & \vdots\\ \vdots & \ddots & \ddots & \overline{c_1}\\ c_m & \cdots & c_1 & c_0\end{pmatrix}\in\mathbb{C}^{(m+1)\times(m+1)}$$

*denotes the Toeplitz matrix. Suppose that $C(c)$ is positive definite. For all $\varphi\in(0,\,2\cdot\pi]$ let*

$$D(\varphi):=\frac{1}{2\cdot\pi}\cdot\frac{e_0^*\cdot C^{-1}(c)\cdot e_0}{|e_0^*\cdot C^{-1}(c)\cdot\mathbf{c}(\varphi)|^2}\in\mathbb{R}\tag{8.3}$$

*where $e_0:=(1,0,\dots,0)^{\mathsf{T}}\in\mathbb{C}^{m+1}$. Then $D(\varphi)$ is positive and the measure $F_D$ with density $D$ fulfills the moment constraints $c=\int\mathbf{c}(\varphi)\,\mathrm{d}F_D(\varphi)$. Among all such measures it has minimal Burg entropy $\mathcal{H}_{Burg}(F_D)$.*

[1] Burg entropy should not be confused with the more widely used Boltzmann-Shannon entropy which is given by

$$\int_0^{2\cdot\pi}-D(\varphi)\cdot\log D(\varphi)\,\mathrm{d}\varphi.$$

**Algorithm 8.1** Levinson's algorithm [Burg 1975, p. 14 ff.].
**Input:** $c \in \mathbb{C}^{m+1}$ with $C(c)$ Hermitian and positive definite.
**Output:** $q := C^{-1}(c) \cdot e_0 \in \mathbb{C}^{m+1}$

```
1. $q_0 := \frac{1}{c_0}$
2. For $l \in \{1, \dots, m\}$:
    a) $d := \sum_{k=0}^{l-1} q_k \cdot c_{l-k}$
    b) $(q_0, \dots, q_l) := \frac{1}{1-|d|^2} \cdot ((q_0, \dots, q_{l-1}, 0) - d \cdot (0, \overline{q_{l-1}}, \dots, \overline{q_0}))$
3. Return $(q_0, \dots, q_m)^\mathsf{T}$.
```

*Proof.* See [Burg 1975, p. 8 ff.] or Appendix B.5. □

The requirement of a positive-definite Toeplitz matrix $C(c)$ is justified by Propositions 2.11 and 2.13. If the Toeplitz matrix has a negative eigenvalue, there cannot be any solution to the trigonometric moment problem according to Proposition 2.11. Thus, the measurements $c$ must be faulty. This is a useful test for validation of the measurement procedure. If it is positive semi-definite but singular, Proposition 2.13 implies that there is a unique reconstruction with finite support. We investigate this case in Section 8.4.1. Otherwise, the maximum entropy spectral estimate[2] provides a valid reconstruction.

In spite of its remarkable properties of matching all measurements exactly while minimizing the prior, Equation (8.3) can be evaluated easily. The term mostly consists of dot products and basic arithmetic operations. To compute $e_0^* \cdot C^{-1}(c)$ we need to solve a system of linear equations of size $(m+1) \times (m+1)$. This system has a very special structure which can be exploited by fast algorithms solving it in time $O(m^2)$ or even superfast algorithms solving it in $O(m \cdot \log^2 m)$ [Ammar and Gragg 1988]. For our values of $m$ Levinson's algorithm with its asymptotic run time of $O(m^2)$ performs best. It is given in Algorithm 8.1 and for the correctness proof we refer to the work of Burg [1975, p. 14 ff.].

[2] The name may seem counter-intuitive because Burg entropy is minimized, not maximized. This discrepancy is due to a dual interpretation where the defined density is the power spectrum of a stochastic process with maximal Boltzmann-Shannon entropy. Burg [1975] was primarily interested in this dual problem.

### 8.2.3 Properties of the Reconstruction

At a very general level the maximum entropy spectral estimate provides an alternative to a common inverse Fourier transform for a truncated series of Fourier coefficients. While a common inverse Fourier transform would simply set all unknown Fourier coefficients to zero, this solution chooses them to minimize Burg entropy. Still, it matches the given Fourier coefficients (i.e. trigonometric moments) exactly. The major advantage is that the reconstruction is known to be a positive density. Therefore, when the application provides this prior knowledge, the maximum entropy spectral estimate should be preferred over an inverse Fourier transform.

Effectively, Equation (8.3) defines the reciprocal of a positive Fourier series. Everything except for the denominator with $\mathbf{c}(\varphi)$ is a constant. Exploiting $|y|^2 = y \cdot \overline{y}$ for $y \in \mathbb{C}$, this last expression can be rewritten as

$$|e_0^* \cdot C^{-1}(c) \cdot \mathbf{c}(\varphi)|^2 = e_0^* \cdot C^{-1}(c) \cdot \mathbf{c}(\varphi) \cdot \mathbf{c}^*(\varphi) \cdot C^{-1}(c) \cdot e_0. \tag{8.4}$$

The product $\mathbf{c}(\varphi) \cdot \mathbf{c}^*(\varphi)$ is an $(m+1) \times (m+1)$ Toeplitz matrix with entries $\exp(-i \cdot m \cdot \varphi), \ldots, \exp(i \cdot m \cdot \varphi)$ on its diagonals (cf. Proposition 2.11 on page 26). Thus, the expression is a positive Fourier series with frequency components ranging from $-m$ to $m$ (Fig. 8.1d middle).

It can be regarded as the Fourier series of minimal degree such that its reciprocal produces the prescribed trigonometric moments. If $C(c)$ is positive definite, this Fourier series has no root. Still, it can be close to zero. Whenever this happens, the reconstructed density exhibits a sharp peak. In practice, this situation is very common and the maximum entropy spectral estimate handles it much better than the inverse Fourier transform.

Another useful interpretation of the maximum entropy spectral estimate extends it to the whole complex plane. Each phase $\varphi \in (0, 2 \cdot \pi]$ is associated with a point on the unit circle $x := \exp(i \cdot \varphi) \in \mathbb{C}$. With this identity $\mathbf{b}(x) = \mathbf{c}(\varphi)$ where $\mathbf{b} : \mathbb{C} \to \mathbb{C}^{m+1}$ with $\mathbf{b}_j(y) := y^j$ is the extension of the power-moment-generating function to the complex plane.

In this notation,

$$e_0^* \cdot C^{-1}(c) \cdot \mathbf{c}(\varphi) = e_0^* \cdot C^{-1}(c) \cdot \mathbf{b}(x) \tag{8.5}$$

is a complex polynomial of degree $m$ or less. If its degree is $m$, it has roots $x_0, \ldots, x_{m-1} \in \mathbb{C}$ which are known to lie outside the unit circle by Lemma B.9 on page 179 (Fig. 8.1d bottom). Otherwise $m$ can be reduced accordingly. The roots determine the polynomial uniquely except for a constant factor, i.e. the density of the maximum entropy spectral estimate is proportional to

$$D(\varphi) \propto \frac{1}{\left|\prod_{l=0}^{m-1}(x - x_l)\right|^2} = \prod_{l=0}^{m-1} \frac{1}{|\exp(i \cdot \varphi) - x_l|^2}. \tag{8.6}$$

A root near the unit circle leads to a large density at the corresponding phase whereas farther roots correspond to smaller values. Any placement of roots outside the unit circle is possible and thus the maximum entropy spectral estimate achieves considerable expressive power.

## 8.3 Measurement Procedure

To use the maximum entropy spectral estimate with measured data, we need to acquire measurements at specific frequencies with sinusoidal modulation as explained in Section 8.2.1. In the following, we present methods to achieve this robustly. Our experiments use a modified version of the hardware setup by Heide et al. [2013], but we are confident that the proposed methods are applicable to a wide range of hardware including the setup by Shrestha et al. [2016], more recent sensors by pmdTechnologies as well as Microsoft Kinect for Xbox One [Bamji et al. 2015].

### 8.3.1 Achieving Sinusoidal Modulation

It is difficult to achieve exact sinusoidal modulation by adjusting the electronics providing the modulation signal. Fortunately, there is a robust workaround leading to a modulation that is arbitrarily close to a sinusoid. For this to work, it has to be possible to adjust the phase shift between the light modulation $\boldsymbol{s}_i$ and the sensor modulation $\boldsymbol{s}_s$. Doing so shifts the effective modulation $\boldsymbol{s}$.

Harmonic cancellation [Payne et al. 2010] uses $n_\varphi \in \mathbb{N}$ equidistant phase shifts and builds a linear combination of the resulting measurements. This is equivalent to generating a linear combination of the phase-shifted modulations. The used linear combination is

$$\sum_{k=0}^{n_\varphi - 1} \sin\left((k+1) \cdot \frac{\pi}{n_\varphi + 1}\right) \cdot \boldsymbol{s}\left(\varphi - k \cdot \frac{\pi}{n_\varphi + 1}\right).$$

This new effective modulation is free of harmonic frequencies up to harmonic $2 \cdot n_\varphi - 1$. Use of harmonic cancellation does not increase measurement times because each phase shift is used for a fraction of the exposure time that reflects the weight in the linear combination [Payne et al. 2010].

We have adopted harmonic cancellation in our prototype hardware but only get robust results up to $n_\varphi = 3$ due to timing issues. Therefore, we propose an alternate scheme. We do not use equidistant phase shifts but split up the exposure time evenly. The $k$-th interval of the exposure time uses phase shift $\arccos\left(1 - \frac{2 \cdot k+1}{n_\varphi}\right)$. Thus, the effective modulation becomes

$$\frac{1}{n_\varphi} \cdot \sum_{k=0}^{n_\varphi - 1} \boldsymbol{s}\left(\varphi - \arccos\left(1 - \frac{2 \cdot k + 1}{n_\varphi}\right)\right). \tag{8.7}$$

For this scheme to work, four-bucket sampling should be used. This means that values are measured with phase shifts of 0, $\frac{\pi}{2}$, $\pi$ and $\frac{3}{2} \cdot \pi$ (in addition to other phase shifts). By subtracting pairs of measurements with relative phase shift $\pi$, the resulting values correspond to an effective modulation with the symmetry $\boldsymbol{s}(\varphi - \pi) = -\boldsymbol{s}(\varphi)$ for all $\varphi \in \mathbb{R}$. By exploiting this property, we can prove that the arccos-phase sampling in Equation (8.7) converges to the desired result.

**Proposition 8.3.** *Let $\boldsymbol{s} : \mathbb{R} \to \mathbb{R}$ be $2 \cdot \pi$-periodic, continuous and for all $\varphi \in \mathbb{R}$ let $\boldsymbol{s}(\varphi - \pi) = -\boldsymbol{s}(\varphi)$. Then for all $\varphi \in \mathbb{R}$*

$$\lim_{n_\varphi \to \infty} \frac{1}{n_\varphi} \cdot \sum_{k=0}^{n_\varphi - 1} \boldsymbol{s}\left(\varphi - \arccos\left(1 - \frac{2 \cdot k + 1}{n_\varphi}\right)\right) = \frac{\pi}{2} \cdot \boldsymbol{s} * \sin(\varphi), \tag{8.8}$$

*i.e. by the convolution theorem Equation (8.7) converges to a scaled and shifted sinusoid.*

*Proof.* Equation (8.8) is a Riemann-sum in the sense that

$$\lim_{n_\varphi \to \infty} \frac{1}{n_\varphi} \cdot \sum_{k=0}^{n_\varphi - 1} \boldsymbol{s}\left(\varphi - \arccos\left(1 - \frac{2 \cdot k + 1}{n_\varphi}\right)\right) = \int_0^1 \boldsymbol{s}(\varphi - \arccos(1 - 2 \cdot k))\,\mathrm{d}k.$$

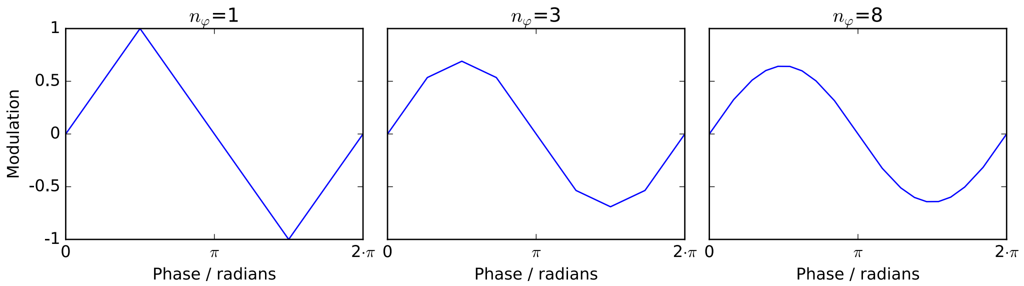


Figure 8.2: The modulation arising from arccos-phase sampling for a triangular original modulation $\boldsymbol{s}$. For $n_\varphi = 8$ it is close to a sinusoid.

Now we apply integration by substitution with $k := \frac{1-\cos\psi}{2}$:

$$\begin{aligned}
&\int_0^1 \boldsymbol{s}(\varphi - \arccos(1 - 2\cdot k))\,\mathrm{d}k \\
&= \int_0^\pi \boldsymbol{s}\left(\varphi - \arccos\left(1 - 2\cdot\frac{1-\cos\psi}{2}\right)\right)\cdot\frac{\sin\psi}{2}\,\mathrm{d}\psi \\
&= \frac{1}{2}\cdot\int_0^\pi \boldsymbol{s}(\varphi-\psi)\cdot\sin\psi\,\mathrm{d}\psi \\
&= \frac{1}{4}\cdot\left(\int_0^\pi \boldsymbol{s}(\varphi-\psi)\cdot\sin\psi\,\mathrm{d}\psi + \int_\pi^{2\cdot\pi} \boldsymbol{s}(\varphi-\psi)\cdot\sin\psi\,\mathrm{d}\psi\right) \\
&= \frac{\pi}{2}\cdot\boldsymbol{s}*\sin(\varphi)
\end{aligned}$$

□

Looking at the constant factor in Equation (8.8) we observe that the sinusoidal component in $\boldsymbol{s}$ is reduced by a factor of $\frac{\pi}{4} \approx 0.79$. Thus, harmonic cancellation and arccos-phase sampling have the same asymptotic effect on the demodulation contrast [Payne et al. 2010].

In our experiments we use arccos-phase sampling with $n_\varphi = 8$. Figure 8.2 demonstrates that this yields a high-quality approximation to a sinusoid.

### 8.3.2 Calibration

The above methods make the effective modulation sinusoidal. Since we use direct digital synthesis for generation of the modulation signal, we can also rely on the accuracy of the frequency ratios. Thus, the only remaining degrees of freedom for the modulation signal are the phase shift and the amplitude. Measurements show that these need to be calibrated per pixel.

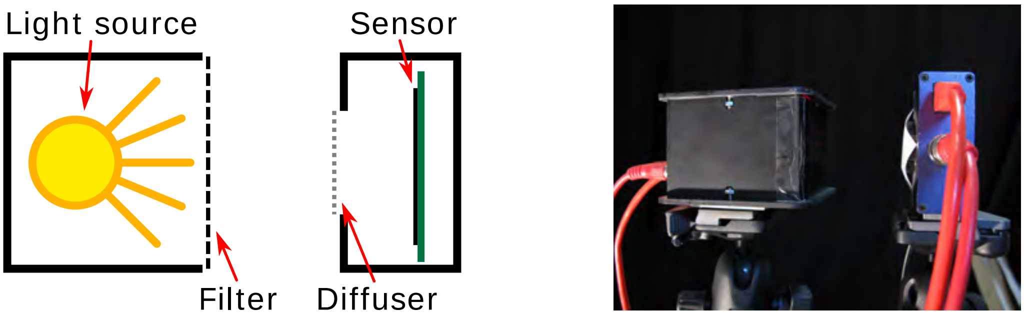


Figure 8.3: We point the laser at the sensor at short range using a neutral density filter with 5.8‰ translucency to avoid overexposure and a diffuser to ensure a uniform light distribution on the sensor. The filter has low reflectivity to avoid undesired interreflections.

To this end, we point the light source at the sensor as shown in Figure 8.3. This whole setup is designed to avoid multipath interference. We expect that most light only passes through the filter once before it reaches the sensor. Other light paths should be attenuated due to the low reflectivity of the filter and either way they should be rather short.

In this calibration setup we assume that the impulse response for each pixel is given by $c_0 \cdot \delta_0$, i.e. an amount of light $c_0 > 0$ arrives at the pixel immediately. If the setup were properly calibrated, we would measure the trigonometric moments

$$c_j = c_0 \cdot \int \exp(i \cdot j \cdot \varphi) \, \mathrm{d}\delta_0(\varphi) = c_0$$

for all $j \in \{0, \ldots, m\}$. The actual measured values will deviate from this. The appropriate correction factor is the measured $c_0$ divided by the measured trigonometric moment.

These factors are computed per pixel and frequency from a high-quality calibration measurement and stored. Multiplying them onto the trigonometric moments obtained from a subsequent measurement simultaneously compensates for non-normal phase shifts and amplitudes in the modulation.

Note that the setup in Figure 8.3 differs from the more common calibration setup of pointing the light source and the camera at a white wall [Heide et al. 2013; Lin et al. 2014]. We also experimented with this setup but found that it makes it hard to avoid multipath interference leading to systematic errors. If the impulse response in the calibration is not a Dirac-$\delta$, calibration procedures effectively perform a deconvolution with the actual impulse response leading to systematic distortions of the reconstruction.

### 8.3.3 The Zeroth Moment

The zeroth moment is defined by $c_0 := \int 1 \, \mathrm{d}F(\varphi)$, so it captures total brightness due to the active illumination without any modulation. All related work using AMCW lidar systems, except for Godbaz et al. [2012], only incorporates measurements with zero-mean modulation, meaning that the captured data are literally orthogonal to the zeroth moment. This misses important information.

Consider a uniform impulse response $F$ on $(0, 2 \cdot \pi]$. All of its trigonometric moments except for the zeroth moment are zero. If the zeroth moment is not measured, an arbitrarily strong uniform component can be added to the impulse response without changing the data. In this sense, the zeroth moment governs sparsity of the distribution, as demonstrated in Figure 8.4. If it takes its minimal value, Proposition 2.13 implies that the ground truth has to be a unique measure with finite support.

The best practice for capturing the zeroth moment is to capture two images without sensor modulation, one with and one without active illumination. Their difference provides the zeroth moment. Since our prototype hardware cannot measure without modulation, we instead perform measurements at 900 kHz. This corresponds to a wavelength of 333.1 m so the sinusoidal modulation wave should be nearly constant across relevant lengths of light paths.

Alternatively, we can exploit Proposition 2.11 to estimate $c_0$ based on prior knowledge about the sparsity of impulse responses. The zeroth moment $c_0$ constitutes the main diagonal of the Toeplitz matrix $C(c)$. If we have not measured $c_0$ yet, we set the main diagonal of $C(c)$ to zero and compute its smallest eigenvalue $\lambda_m$ which will be negative. We then fix the estimated uniform component $\varepsilon_\lambda > 0$ and set $c_0 := \varepsilon_\lambda - \lambda_m$ to ensure that the Toeplitz matrix is positive definite with smallest eigenvalue $\varepsilon_\lambda$. Smaller values of $\varepsilon_\lambda$ lead to sparser reconstructions.

This method is also suited for correcting invalid measurements if we have measured the zeroth moment. Whenever we encounter a Toeplitz matrix with a smallest eigenvalue less than $\varepsilon_\lambda$, we replace $c_0$ by $\varepsilon_\lambda - \lambda_m$ as defined above. This changes the measurement in a minimal way to make it valid. Alternatively, $c_1, \ldots, c_m$ can be scaled down to avoid changing overall brightness. We refer to this procedure as biasing (cf. Section 4.1.3). For scenes with sparse impulse responses, sensor noise makes it indispensable.

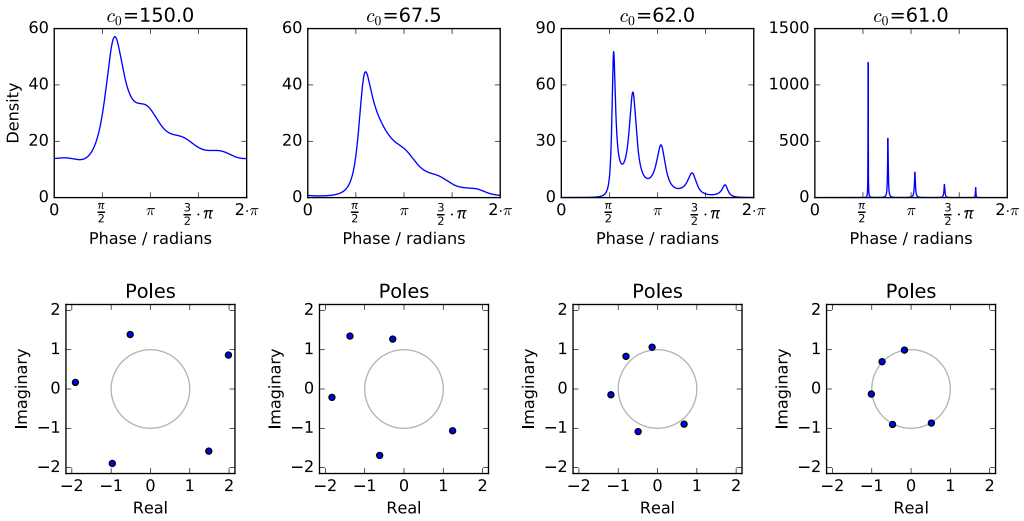


Figure 8.4: Top: Various reconstructions arising from the maximum entropy spectral estimate for the transient pixel in Figure 8.1b with $m = 5$ and different values of the zeroth moment $c_0$. The zeroth moment governs sparsity of the reconstruction. The best result is obtained with the ground truth $c_0 = 67.5$. For $c_0 = 61$ the Toeplitz matrix $C(c)$ is nearly singular and the reconstruction is nearly sparse. In between it changes continuously. The strong changes illustrate the importance of measuring $c_0$ accurately. Bottom: The roots of the polynomial in the denominator of the maximum entropy spectral estimate (see Equation (8.6)). The roots approach the unit circle as the reconstruction approaches a measure with finite support.

## 8.4 Analysis of Impulse Responses

The maximum entropy spectral estimate provides efficient random access to the density of a transient image. Though, in many application scenarios we would like to infer other information immediately. In this section we present various efficient methods to infer information about a transient image without computing it completely. We also present upper bounds for the error of the reconstruction.

### 8.4.1 Perfect Reconstruction of Sparse Responses

The maximum entropy spectral estimate fails if the ground truth $F$ has $m$ or fewer points of support. This special situation is fully described by Proposition 2.13 on page 29. The reason why the maximum entropy spectral estimate fails is that the ground truth $F$ is uniquely determined by

**Algorithm 8.2** Perfect reconstruction of a measure from trigonometric moments in the boundary case.
**Input:** $c \in \mathbb{C}^{m+1}$ such that $C(c)$ is positive semi-definite but singular.
**Output:** The unique measure $F$ with $c = \int \mathbf{c}(\varphi)\,\mathrm{d}F(\varphi)$.

```
1. Compute $q \in \ker C(c)$ with $q \neq 0$.
2. Solve $\sum_{j=0}^{m} \overline{q_j} \cdot x^j = 0$ for $x$ to obtain all pairwise different roots
   $x_0, \ldots, x_{n-1} \in \mathbb{C}$.
3. Solve the system of linear equations
   $$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_0^1 & x_1^1 & \cdots & x_{n-1}^1 \\ \vdots & \vdots & & \vdots \\ x_0^{n-1} & x_1^{n-1} & \cdots & x_{n-1}^{n-1} \end{pmatrix} \cdot \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{n-1} \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix}.$$
4. For $l \in \{0, \ldots, n-1\}$ set $\varphi_l := \arg x_l \in (0, 2 \cdot \pi]$, i.e.
   $$x_l = |x_l| \cdot \exp(i \cdot \varphi_l).$$
5. Return $F := \sum_{l=0}^{n-1} w_l \cdot \delta_{\varphi_l}$.
```

its trigonometric moments. There is no other valid reconstruction and in particular there is no reconstruction having a finite density. Equation (8.3) is not applicable because the Toeplitz matrix $C(c)$ is singular.

While biasing may be used to avoid this failure case, handling it explicitly offers an attractive alternate reconstruction. Since the ground truth is uniquely determined by the measurements, it can be reconstructed perfectly. Algorithm 8.2 computes it efficiently. Its correctness is a direct consequence of Proposition 2.13 and the proof thereof. The polynomial solved in this Algorithm is the limit of the polynomial in Equation (8.5) for $c_0$ approaching its minimal value (Fig. 8.4 bottom).

Although we have formulated this result for the case that $C(c)$ is singular, it is also applicable in the general case. We simply separate the distribution into a uniform component and a sparse component. To this end we compute the smallest eigenvalue $\lambda_m$ of $C(c)$ and a corresponding eigenvector $q$. Then $\lambda_m$ gives the strength of the uniform component and the sparse component can be computed from $q$ as in Algorithm 8.2 using $c_0 - \lambda_m$ in place of $c_0$.

If specular interactions dominate, the uniform component becomes small and measurement of the zeroth moment may be skipped. This method is known as Pisarenko estimate and optimized algorithms exist for its computation [Cybenko and Van Loan 1986]. It is closely related to the work by Bhandari et al. [2014a], except that their method requires more than twice as many measurements and does not necessarily find a distribution compatible with all of them. It can also be understood as closed-form implementation of the work by Freedman et al. [2014] without error tolerance because their technique minimizes $c_0$. The Pisarenko estimate realizes the theoretical best case of reconstructing the $2 \cdot m$ real parameters describing a distribution with $m$ points of support from $m$ complex phasors.

While the Pisarenko estimate does not reflect uncertainty as reasonably as the maximum entropy spectral estimate, it provides a more explicit reconstruction. This eases analysis of transient images and provides excellent results if specular interactions dominate as demonstrated in Figure 8.5. The reconstructed data directly provides insight into the strength and time of flight of all returns per pixel.

A possible application is fast separation of direct and indirect illumination. The direct component can be identified as first return with a weight above a relative threshold. Its weight provides the strength of the direct return. The sum of the other weights provides indirect returns.

### 8.4.2 Error Bounds

From Proposition 2.13 we know that we can obtain a perfect reconstruction if the ground truth has $m$ or fewer points of support. Intuitively, we expect that the reconstruction is still very close to the ground truth when the ground truth has a small uniform component. This motivates the search for bounds on the error of the reconstruction. We suppose that our measurements are correct and ask for the maximal possible distance between the unknown ground truth and our reconstruction.

For the trigonometric moment problem this question has been answered [Karlsson and Georgiou 2013]. The authors observe that no meaningful statements can be made if densities are considered directly. The construction used in Algorithm 4.1 works analogously for the trigonometric moment problem (see Algorithm B.2). We can prescribe an arbitrary phase where the reconstruction must have support in the form of a Dirac-$\delta$ and obtain a reconstruction with a total of $m+1$ points of support. A Dirac-$\delta$ has infinite density, so the density of reconstructions at any point can be anything between zero and infinity.

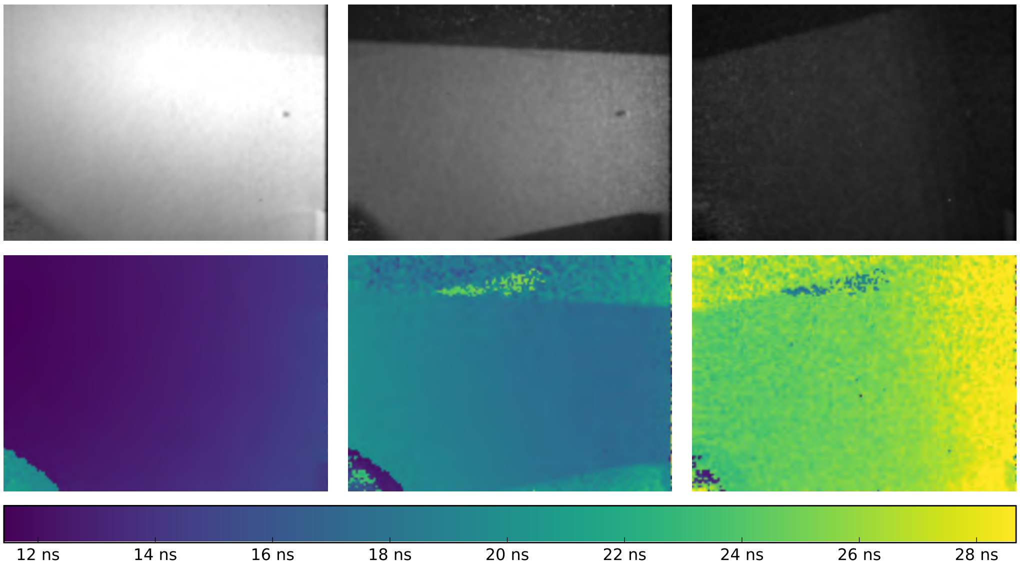


Figure 8.5: A scene where direct illumination and two mirrors are used to illuminate a wall three times (cf. Figure 8.7). Four trigonometric moments are measured by averaging 30 takes to improve the signal to noise ratio. Top: The strength of the strongest three returns computed with the Pisarenko estimate. Bottom: The corresponding time of flight. The different light paths through the mirrors are separated clearly. Phase noise increases as the strength of the return weakens.

To get a meaningful result, Karlsson and Georgiou [2013] propose to smooth densities before analysis. This is done using the Poisson kernel (Fig. 8.6a)

$$P_r(\varphi) := \frac{1}{2 \cdot \pi} \cdot \frac{1 - r^2}{|1 - r \cdot \exp(i \cdot \varphi)|^2},$$

where $r \in [0,\, 1)$ governs the sharpness of the kernel. For $r = 0$ it is constant, for $r \to 1$ it converges to $\delta_0$.

After smoothing, sharp lower and upper bounds can be computed in closed form.

**Proposition 8.4** (Error bounds [Karlsson and Georgiou 2013]). *Let $F$ be a finite measure such that $c = \int \mathbf{c}(\varphi)\, \mathrm{d}F(\varphi)$. Then for all $\varphi \in (0,\, 2 \cdot \pi]$ and* *all* $r \in (0,\, 1)$

$$\begin{aligned}
&\frac{1}{2\cdot\pi}\cdot\left(\Re Q(r\cdot\exp(i\cdot\varphi)) - \sqrt{R(r\cdot\exp(i\cdot\varphi))}\right)\\
\leq\ &(P_r * F)(\varphi) = \int P_r(\varphi-\psi)\,\mathrm{d}F(\psi)\\
\leq\ &\frac{1}{2\cdot\pi}\cdot\left(\Re Q(r\cdot\exp(i\cdot\varphi)) + \sqrt{R(r\cdot\exp(i\cdot\varphi))}\right)
\end{aligned}$$

*with* $\Re$ *denoting the real part,*

$$\begin{aligned}
Q(x) &:= \frac{\frac{2}{1-|x|^2} + \mathbf{e}^*(x)\cdot C^{-1}(c)\cdot\mathbf{d}(x)}{\mathbf{d}^*(x)\cdot C^{-1}(c)\cdot\mathbf{d}(x)} \in \mathbb{C},\\
R(x) &:= |Q(x)|^2 - \frac{\mathbf{e}^*(x)\cdot C^{-1}(c)\cdot\mathbf{e}(x)}{\mathbf{d}^*(x)\cdot C^{-1}(c)\cdot\mathbf{d}(x)} \in \mathbb{R}
\end{aligned}$$

*and* $\mathbf{d}, \mathbf{e} :\ \mathbb{C}\setminus\{0\} \to \mathbb{C}^{m+1}$ *defined by*

$$\forall j \in \{0,\ldots,m\}:\ \mathbf{d}_j(x) := x^{-j-1},\, \mathbf{e}_j(x) := x^{-j-1}\cdot\left(c_0 + 2\cdot\sum_{k=1}^{j} c_k\cdot x^k\right).$$

*These bounds are sharp.*

*Proof.* Let $\varphi \in (0,\, 2\cdot\pi]$. Karlsson and Georgiou [2013, proof of Proposition 12] prove that a $w_z \in \mathbb{C}$ with

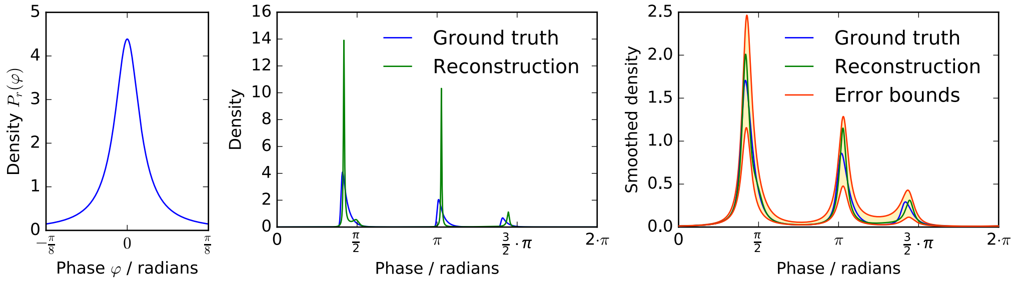
$$\Re w_z = 2\cdot\pi\cdot(P_r * F)(\varphi)$$

exists if and only if

$$|w_z - Q(r\cdot\exp(i\cdot\varphi))|^2 \leq R(r\cdot\exp(i\cdot\varphi)).$$

The claimed inequalities follow immediately and since this is an equivalence, they are sharp. □

The bounds rely solely on the knowledge that $F$ fulfills the moment constraints $c = \int \mathbf{c}(\varphi)\,\mathrm{d}F(\varphi)$. Assuming correct measurements, the ground truth is known to fulfill these constraints and we can compute an area containing its smoothed density. The same holds for our reconstruction. If the ground truth is reasonably close to a sparse distribution, this area is pleasantly small as demonstrated in Figure 8.6. In this case, we can be certain that the reconstruction is close to the ground truth and we can give specific bounds on possible locations of local maxima in the unknown ground truth.



(a) Poisson kernel (b) Densities before smoothing (c) Error bounds

Figure 8.6: An example of error bounds using a synthetic ground truth (cf. Figure 8.12b) with $m = 5$ and $r = 0.93$. The impulse responses (8.6b) have to be smoothed using a Poisson kernel (8.6a) before meaningful error bounds can be derived. After smoothing (8.6c), we obtain sharp bounds on the smoothed density using solely the knowledge of the trigonometric moments $c_0, \ldots, c_5 \in \mathbb{C}$. These bounds apply to the ground truth and the maximum entropy spectral estimate such that the maximal reconstruction error is bounded.

### 8.4.3 Estimating Range

Range imaging with AMCW lidar systems is typically done in real time with a limited computing time budget. Therefore, it is worthwhile to implement highly optimized methods. The natural candidates for estimates of range are local maxima of the maximum entropy spectral estimate in Equation (8.3).

Critical points of this function coincide with critical points of its scaled reciprocal. For convenience we define $q := C^{-1}(c) \cdot e_0$ and rewrite the scaled reciprocal as in Equation (8.4):

$$|e_0^* \cdot C^{-1}(c) \cdot \mathbf{c}(\varphi)|^2 = q^* \cdot \mathbf{c}(\varphi) \cdot \mathbf{c}^*(\varphi) \cdot q$$

$$= \sum_{j,k=0}^{m} \overline{q_j} \cdot \exp(i \cdot j \cdot \varphi) \cdot \overline{\exp(i \cdot k \cdot \varphi)} \cdot q_k = \sum_{j,k=0}^{m} \overline{q_j} \cdot q_k \cdot \exp(i \cdot (j-k) \cdot \varphi)$$

To compute critical points, we take the derivative with respect to $\varphi$:

$$\frac{\partial}{\partial \varphi} |e_0^* \cdot C^{-1}(c) \cdot \mathbf{c}(\varphi)|^2 = \sum_{j,k=0}^{m} \overline{q_j} \cdot q_k \cdot i \cdot (j-k) \cdot \exp(i \cdot (j-k) \cdot \varphi)$$

Now we substitute $x = \exp(i \cdot \varphi)$ and multiply by $i^{-1} \cdot x^m \neq 0$ to arrive at

$$\sum_{j,k=0}^{m} \overline{q_j} \cdot q_k \cdot (j-k) \cdot x^{m+j-k}.$$

This is a polynomial of degree $2 \cdot m$ so it can have up to $2 \cdot m$ roots on the unit circle which are the critical points of the maximum entropy spectral estimate. Since the reconstructed density is a smooth, periodic function, at most half of the critical points correspond to local maxima. A reasonable estimate for range is that it corresponds to the first local maximum above a threshold.

If specular interactions are known to be the primary cause of multipath interference, an even faster approach uses the Pisarenko estimate introduced in Section 8.4.1. This way, the degree of the arising polynomial equation is halved. It is interesting to note that both methods can reconstruct $m$ distinct peaks. The maximum entropy spectral estimate only requires the additional measurement of the zeroth moment to estimate smoothness of the impulse response and it only constructs $m$ peaks when the data demands it (see Figure 8.4).

### 8.4.4 Cumulative Transient Images

The density defined in Equation (8.3) describes the maximum entropy spectral estimate completely. However, densities are not very informative by themselves. A large density at a point could mean that a return carries a lot of light, but it could also mean that a small amount of light is strongly localized in time.

Integrals over densities are far more informative. For an interval $[\alpha,\, \beta] \subseteq [0,\, 2 \cdot \pi]$ the measure

$$F_D([\alpha,\, \beta]) = \int_\alpha^\beta D(\varphi)\,\mathrm{d}\varphi$$

provides the amount of light that returned with a phase shift in $[\alpha,\, \beta]$. The total amount of light is given by the zeroth moment $c_0$ so we can immediately relate these quantities. The cumulative distribution function $F_D([0,\, \varphi])$ provides a useful visualization of transient images that we refer to as cumulative transient image. Burg [1975, p. 109 ff.] computes it using a partial fraction expansion and we take a similar approach here.

To construct a closed form for the indefinite integral of Equation (8.3), we rewrite it as a rational function of $x = \exp(i \cdot \varphi)$. Let $q := C^{-1}(c) \cdot e_0$ and let $x_0, \ldots, x_{m-1} \in \mathbb{C}$ be the roots of the polynomial $q^* \cdot \mathbf{b}(x)$ as in Equation

(8.6). Then

$$\begin{aligned}
D(\varphi) &= \frac{1}{2\cdot\pi}\cdot\frac{q_0}{\left|\overline{q_m}\cdot\prod_{l=0}^{m-1}(x-x_l)\right|^2}\\
&= \frac{q_0}{2\cdot\pi\cdot|q_m|^2}\cdot\frac{1}{\prod_{l=0}^{m-1}(x-x_l)\cdot\prod_{l=0}^{m-1}(x^{-1}-\overline{x_l})}\\
&= \frac{q_0}{2\cdot\pi\cdot|q_m|^2}\cdot\frac{x^m}{\prod_{l=0}^{m-1}(x-x_l)\cdot\prod_{l=0}^{m-1}(1-x\cdot\overline{x_l})}\\
&= \frac{q_0\cdot(-1)^m}{2\cdot\pi\cdot|q_m|^2\cdot\prod_{l=0}^{m-1}\overline{x_l}}\cdot\frac{x^m}{\prod_{l=0}^{m-1}(x-x_l)\cdot\prod_{l=0}^{m-1}(x-\overline{x_l}^{-1})}.
\end{aligned}$$

For all $l\in\{0,\dots,m-1\}$ let

$$x_{m+l} := \overline{x_l}^{-1} = \frac{x_l}{|x_l|^2} \qquad\text{and}\qquad \eta := \frac{q_0\cdot(-1)^m}{2\cdot\pi\cdot|q_m|^2\cdot\prod_{k=0}^{m-1}\overline{x_k}}$$

such that

$$D(\varphi) = \eta\cdot\frac{x^m}{\prod_{l=0}^{2\cdot m-1}(x-x_l)}.$$

For simplicity we assume that the poles $x_0,\dots,x_{2\cdot m-1}$ are pairwise different. Thus, we have written $D$ as rational function of $x\in\mathbb{C}$ with $2\cdot m$ simple poles. Its partial fraction decomposition takes the form [Fischer and Lieb 2012, p. 78 f.]

$$D(\varphi) = \sum_{l=0}^{2\cdot m-1}\frac{r_l}{x-x_l} = \sum_{l=0}^{2\cdot m-1}\frac{r_l}{\exp(i\cdot\varphi)-x_l}$$

with residues

$$r_l = \lim_{x\to x_l}(x-x_l)\cdot\eta\cdot\frac{x^m}{\prod_{k=0}^{2\cdot m-1}(x-x_k)} = \frac{\eta\cdot x_l^m}{\prod_{k=0,\,k\neq l}^{2\cdot m-1}(x_l-x_k)}.$$

Integrating the individual summands, we find the indefinite integral of $D(\varphi)$

$$-\sum_{l=0}^{2\cdot m-1}\frac{r_l}{x_l}\cdot(\varphi+i\cdot\ln(\exp(i\cdot\varphi)-x_l)). \tag{8.9}$$

When evaluating this expression, care has to be taken to pick branches of the complex logarithm that are continuous on the relevant domain.

## 8.5 Results and Discussion

In the following, we present results measured with our prototype hardware. The setup is similar to the one presented by Heide et al. [2013] but additionally features the arccos-phase sampling described in Section 8.3.1. The sensor is a pmd PhotonICs 19k-S3 by pmdTechnologies with a resolution of 163·120 pixels taken from the reference design CamBoard nano.

To improve the signal to noise ratio of the data, we average multiple takes for the results in Figures 8.5, 8.8c, 8.8d, 8.9, 8.10c, 8.10d, 8.11 and 8.13. Note that our prototype hardware suffers from some systematic outliers due to synchronization issues. 5-10% of all captured images differ significantly from other images captured with the same configuration. When multiple takes are given for averaging, we automatically discard such outliers before averaging. In videos we fill in data missing due to outliers using data from the previous frame.

To further reduce noise, we smooth trigonometric moment images using a Gaussian filter with a standard deviation of 0.6 pixels. For the maximum entropy spectral estimate we use biasing as described in Section 8.3.3 to ensure that $\lambda_m \geq 4 \cdot 10^{-3} \cdot c_0$. Unless otherwise noted, this only affects few pixels.

We have implemented evaluation of Equation (8.3) on the GPU in a pixel shader and measure the run time on an NVIDIA GeForce GTX 780. The shader reconstructs $2.9 \cdot 10^5$, $2.2 \cdot 10^5$ and $1.1 \cdot 10^5$ transient frames per second for $m = 3$, $m = 4$ and $m = 8$, respectively. This includes repeated computation of $C^{-1}(c) \cdot e_0$, although this vector could be precomputed. This means that Equation (8.3) is evaluated $163 \cdot 120 \cdot 1.1 \cdot 10^5 = 2.2 \cdot 10^9$ times per second for $m = 8$.

### 8.5.1 Transient Imaging

Our first experiment uses the scene shown in Figure 8.7 to provide a challenging test case with complex specular interactions. The impulse responses encountered for a single pixel consist of up to three distinct returns with high dynamic range. Light initially sweeps from left to right, then the right mirror reflects it to the left and finally the left mirror reflects it to the right. These three returns have different times of flight and the reconstruction has to separate them.

Figure 8.8 shows results obtained with the maximum entropy spectral estimate for different measurement times using an exposure time of 1.92 ms.

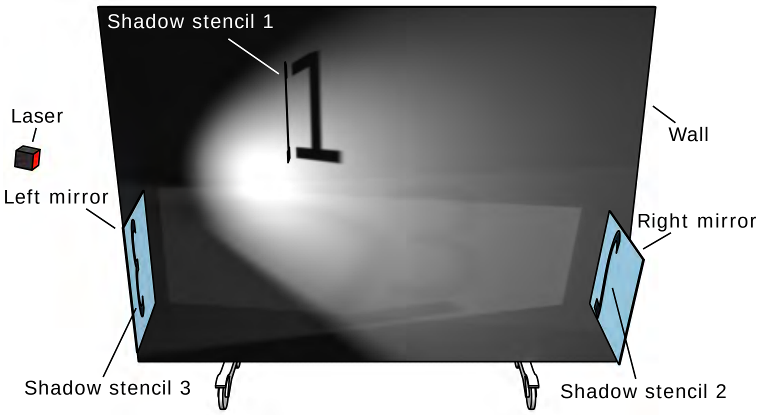
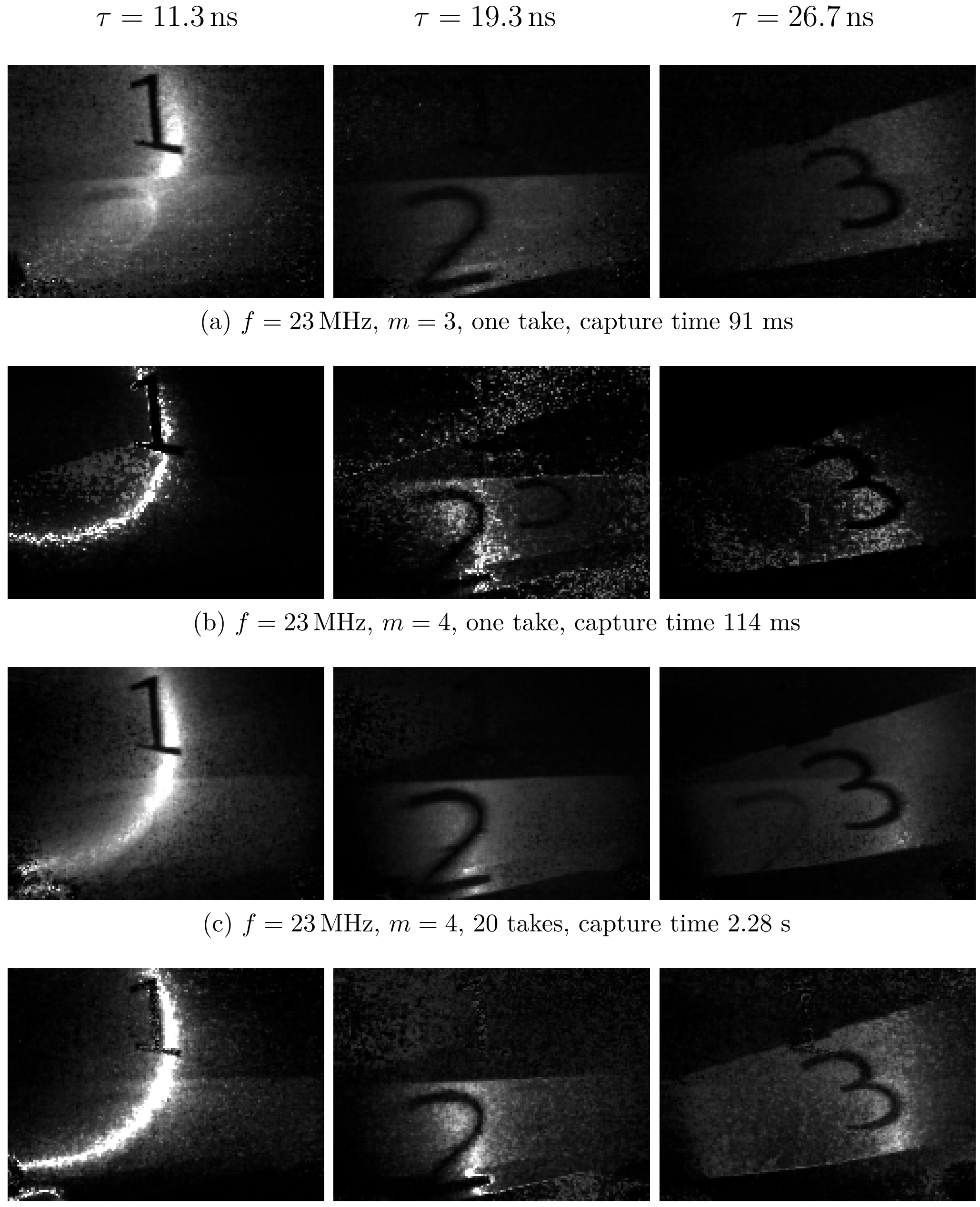


Figure 8.7: A schematic visualization of our experimental setup. The diffuse wall is lit by the active illumination of the AMCW lidar system. Part of the light is reflected by mirrors on both sides such that light sweeps over the wall three times. There are cardboard stencils on both mirrors that cast the shadow of a number onto the wall. Additionally, there is a stencil perpendicular to the wall. This way, all three returns include a shadow showing their index.

With $m = 3$ (i.e. measurements at four frequencies including the measurement for the zeroth moment at 900 kHz) and a single take we already successfully separate the three distinct returns (Fig. 8.8a). However, the reconstruction includes significant uncertainty expressed by means of broad peaks with low density. This behavior depends on the shape of the impulse response. Therefore, the number two is slightly visible for $\tau = 11.3\,\text{ns}$.

Additional frequency measurements yield sharper peaks. At $m = 4$ some visible artifacts remain but all important features are reconstructed (Figs. 8.8b, 8.8c). Using $m = 8$ further reduces these artifacts, leading to a reconstruction with sharp peaks (Fig. 8.8d). On the other hand, the additional measurements also introduce additional noise and potential contradictions, thus making biasing mandatory for 90% of all pixels (see Section 8.3.3).

The additional takes used for Figure 8.8c and 8.8d reduce the noise in the input and the output alike. Figures 8.8b and 8.8c show the same measurement with one and 20 averaged takes, respectively. While the reconstructed features are essentially identical, the averaging leads to a result with substantially less noise. In spite of the single take, Figure 8.8a appears less noisy than Figure 8.8b because the greater uncertainty causes smoothing of impulse responses.


$\tau = 11.3\,\mathrm{ns}$
$\tau = 19.3\,\mathrm{ns}$
$\tau = 26.7\,\mathrm{ns}$

(a) $f = 23\,\mathrm{MHz}$, $m = 3$, one take, capture time 91 ms

(b) $f = 23\,\mathrm{MHz}$, $m = 4$, one take, capture time 114 ms

(c) $f = 23\,\mathrm{MHz}$, $m = 4$, 20 takes, capture time 2.28 s

(d) $f = 11.5\,\mathrm{MHz}$, $m = 8$, 40 takes, capture time 8.21 s

Figure 8.8: A transient image of the scene shown in Figure 8.7 captured with different tradeoffs between capture time and quality. The images show the maximum entropy spectral estimate for different times of flight $\tau$. Note how the three returns are separated.

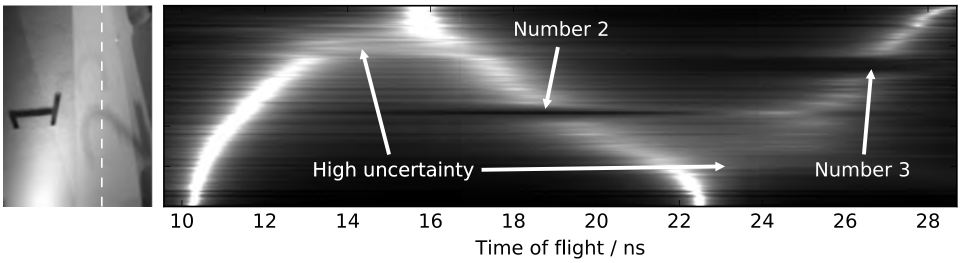


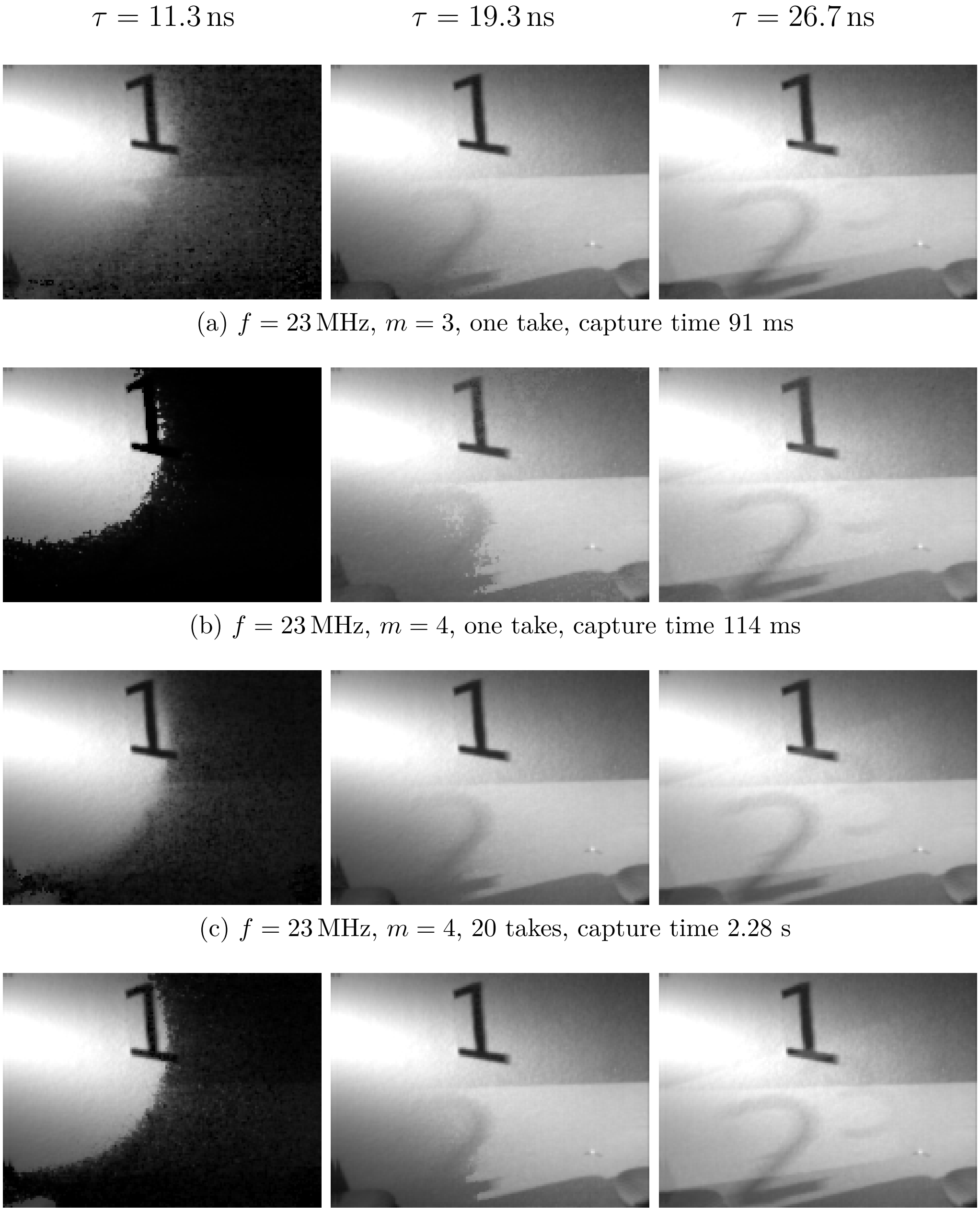
Figure 8.9: A streak image for the data set in Figure 8.8c. Each row shows the time-resolved density for one pixel on the scanline highlighted on the left. Note how separate returns merge in regions of high uncertainty.

If two returns are temporally close, uncertainty may cause them to merge into one as shown in Figure 8.9. It is not possible to specify generic lower or upper bounds on the required distance of returns for successful separation. It rather depends on the complexity of the impulse response and the amount and quality of input data. Under the assumption of perfect data and perfect sparsity, returns can be arbitrarily close (see Section 8.4.1), but in presence of uniform components and noise they have to be farther apart.

Density images such as those shown in Figure 8.8 generally exhibit rather strong noise because slight changes in the sharpness or phase of a peak lead to strong changes in density at a fixed point in time. To analyze whether this noise is systematic, we consider cumulative transient images (see Section 8.4.4). Figure 8.10 shows the cumulative transient images for the above experiment. We note that uncertainty in the reconstruction translates to smeared out or misshaped wave fronts. However, the total brightness contributed by the waves is always reconstructed correctly.

For an example with diffuse interactions we point the camera at a corner but only illuminate the left wall of this corner directly. Most of the right wall is lit indirectly. As shown in Figure 8.11, the measurement of the zeroth moment helps us to adequately reconstruct the corresponding transient image. While the wave on the left wall is very sharp, the right wall receives a smooth wave due to diffuse interreflections.

The comparison of our proposed methods to related work in Figure 8.12 uses synthetic data. For visualization we once more use the cumulative distribution function because sparse reconstructions cannot be represented by a density function. The first example in Figure 8.12a constitutes an ideal case for all techniques and thus all techniques obtain an excellent reconstruction without noise. However, the Dirac-$\delta$ model [Godbaz et al. 2012] and SPUMIC [Kirmani et al. 2013] are quite sensitive to noise.


$\tau = 11.3\,\mathrm{ns}$
$\tau = 19.3\,\mathrm{ns}$
$\tau = 26.7\,\mathrm{ns}$

(a) $f = 23\,\mathrm{MHz}$, $m = 3$, one take, capture time 91 ms

(b) $f = 23\,\mathrm{MHz}$, $m = 4$, one take, capture time 114 ms

(c) $f = 23\,\mathrm{MHz}$, $m = 4$, 20 takes, capture time 2.28 s

(d) $f = 11.5\,\mathrm{MHz}$, $m = 8$, 40 takes, capture time 8.21 s

Figure 8.10: Frames of the cumulative transient images corresponding to Figure 8.8. Each image shows the total amount of light that returned earlier than $\tau$, reconstructed using Equation (8.9).

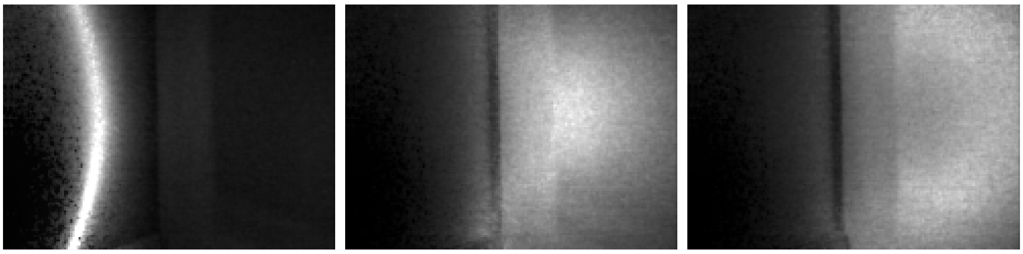
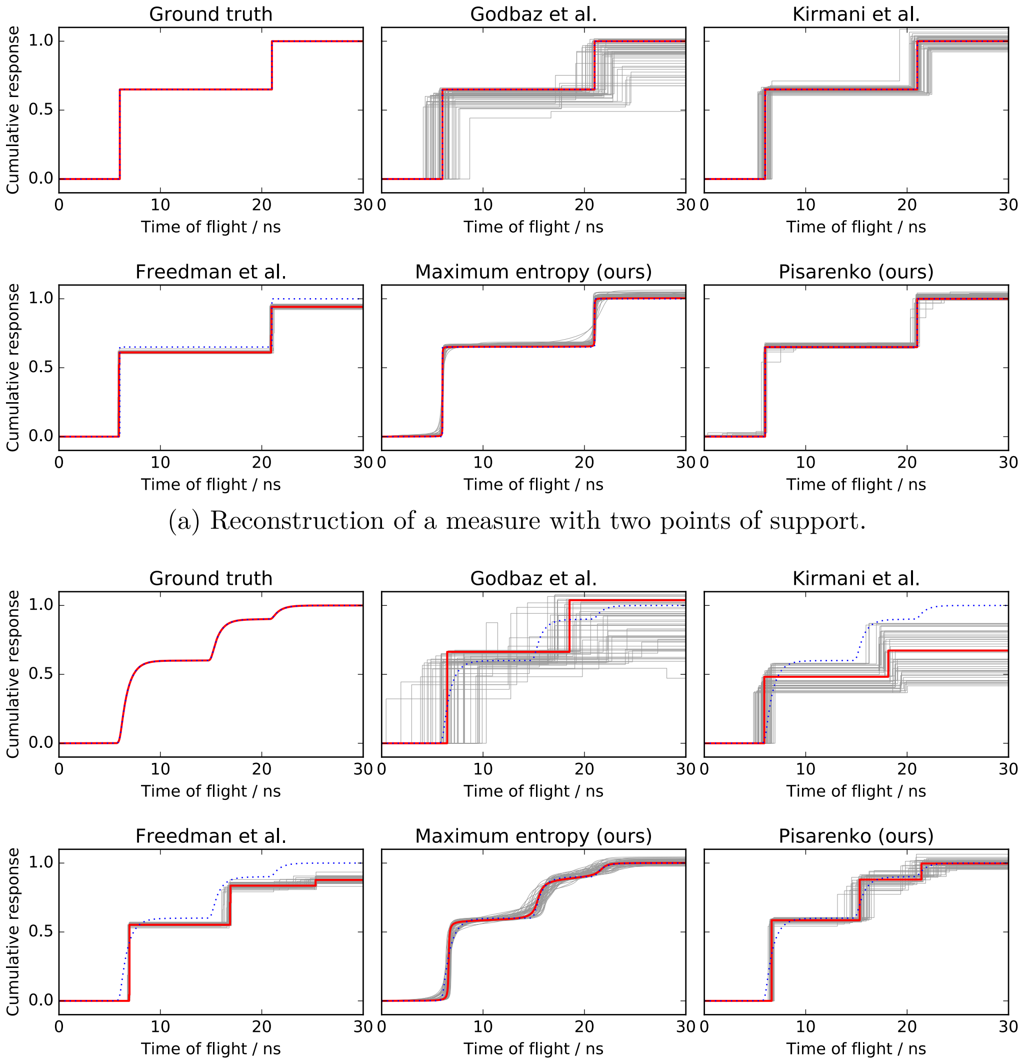


Figure 8.11: Frames of a transient image showing a corner where only the left wall is lit directly. Thanks to the measurement of the zeroth moment the maximum entropy spectral estimate reconstructs a sharp peak for directly lit parts and a smooth peak for indirectly lit parts. This measurement uses $f = 23\,\mathrm{MHz}$, $m = 4$ and 40 takes at an exposure time of 1.92 ms per capture. The capture takes 4.56 seconds.

The second example in Figure 8.12b provides a more realistic test case consisting of three continuous returns modeled by exponentially modified Gaussians [Heide et al. 2014]. The Dirac-$\delta$ model and SPUMIC, both targeted at two sparse returns, only capture the first return adequately and become even more sensitive to noise. SRA [Freedman et al. 2014] has a bias towards stronger sparsity and less overall brightness. Therefore, it loses the third return but successfully reduces the impact of noise. The Pisarenko estimate provides a better reconstruction but is more sensitive to noise. The maximum entropy spectral estimate adequately reconstructs the continuous return. Noise mostly affects sharpness of the peaks.

### 8.5.2 Range Imaging

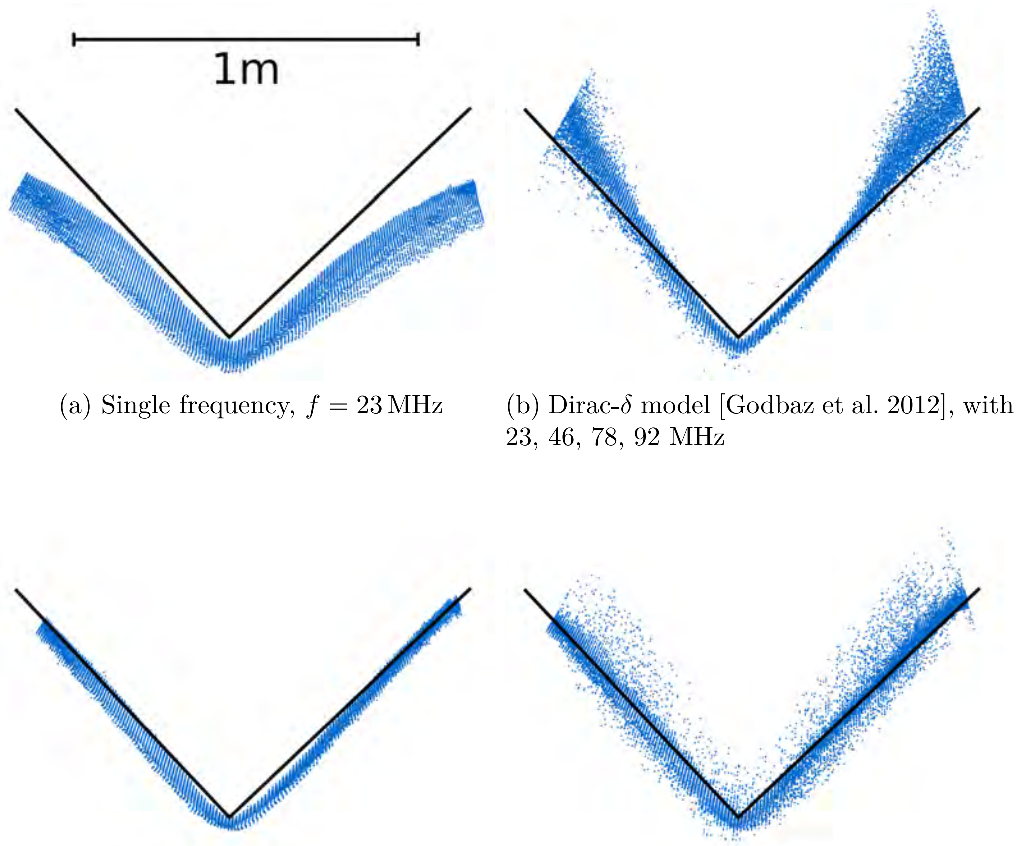
As benchmark for range imaging, we place the camera and the light source next to each other and capture a diffuse corner. This is a prime example of diffuse multipath interference. Figure 8.13 shows our results. A naïve reconstruction using measurements at a single frequency exhibits severe distortions. Range is overestimated because long indirect paths contribute to the estimate. Using the Dirac-$\delta$ model [Godbaz et al. 2012] reduces these systematic distortions but does not behave robustly. The maximum entropy spectral estimate (see Section 8.4.3) provides robust results and reduces distortions due to multipath interference heavily. The Pisarenko estimate suffers from severe outliers. This is understandable because its inherent assumption of a sparse impulse response is inadequate for diffuse multipath interference. This demonstrates the benefit of including the zeroth moment in the reconstruction.



(a) Reconstruction of a measure with two points of support.

(b) Reconstruction of a measure that localizes density around three points.

Figure 8.12: Reconstruction results of various techniques using synthetic data with sinusoidal modulation. Red graphs use measurements without noise, gray graphs originate from measurements with a simulated signal to noise ratio of 70:1 due to Gaussian noise. Each plot contains the ground truth as dotted blue line. The Dirac-$\delta$ model [Godbaz et al. 2012] uses measurements at 11, 22, 33, 44 MHz, Kirmani et al. [2013] uses 11, 22, . . . , 66 MHz and Freedman et al. [2014] uses 23, 46, 69 MHz and $\varepsilon = 0.05$. Our proposed techniques use measurements at 0 (maximum entropy spectral estimate only), 23, 46, 69 MHz.



(a) Single frequency, $f = 23\,\mathrm{MHz}$

(b) Dirac-$\delta$ model [Godbaz et al. 2012], with 23, 46, 78, 92 MHz

(c) Maximum entropy spectral estimate (proposed method), $f = 23\,\mathrm{MHz}$, $m = 3$

(d) Pisarenko estimate (proposed method), $f = 23\,\mathrm{MHz}$, $m = 4$

Figure 8.13: Point clouds of a diffuse corner reconstructed with various techniques. The images show an orthographic top view and the black line is the ground truth. All reconstructions use the same data set with base frequency $f = 23\,\mathrm{MHz}$ and 4 averaged takes (without outliers). The capture time is 365 ms for all results except (8.13a) where it is 91 ms.

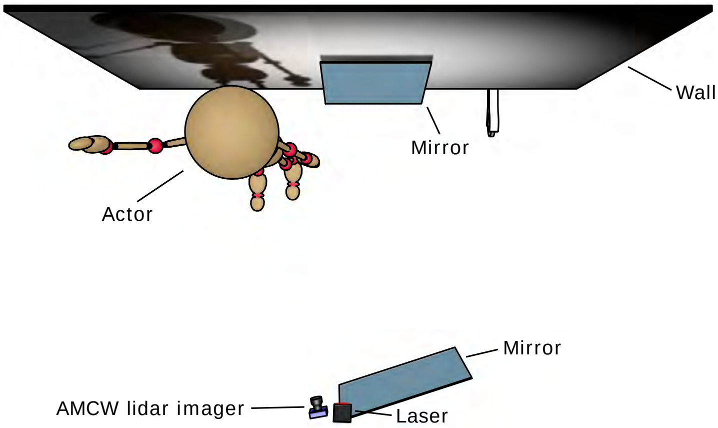


Figure 8.14: A bird's eye view of the experimental setup used for transient video. Light from the active illumination of the AMCW lidar system reaches the diffuse wall directly but also indirectly via two mirrors. An actor moves through the scene from left to right.

### 8.5.3 Transient Video

Our hardware is capable of capturing transient images at video frame rate. If we choose $m = 3$ and use a Pisarenko estimate or a maximum entropy spectral estimate with biasing of the zeroth moment, we require measurements at three frequencies. With four-bucket sampling this amounts to twelve images per transient image. Using an exposure of 0.5 ms and $f = 23\,\text{MHz}$ we can capture such sets of images at 18.6 Hz. The result is a transient video, i.e. each pixel in each frame is an impulse response resolved in time of flight.

Figure 8.14 shows our experimental setup. A mirror is placed in front of a lit wall to reflect part of the light away from the wall and into another mirror. The latter mirror reflects the light back onto the wall such that part of it receives light at two different times of flight. An actor enters from the left, waves and leaves to the right. Therefore, the scene exhibits interesting features in the dimension of time of flight $\tau$ as well as common time $t$.

Figure 8.15 shows three frames of the transient images for three frames of the transient video. Although the images are quite noisy due to the short exposure time, all important features are reconstructed correctly. The light first returns from the actor, then from the direct interaction with the wall and finally it returns after being reflected by both mirrors and the wall.

We can use the Pisarenko estimate for separation of direct and indirect illumination as proposed in Section 8.4.1. Figure 8.16 shows the same frames

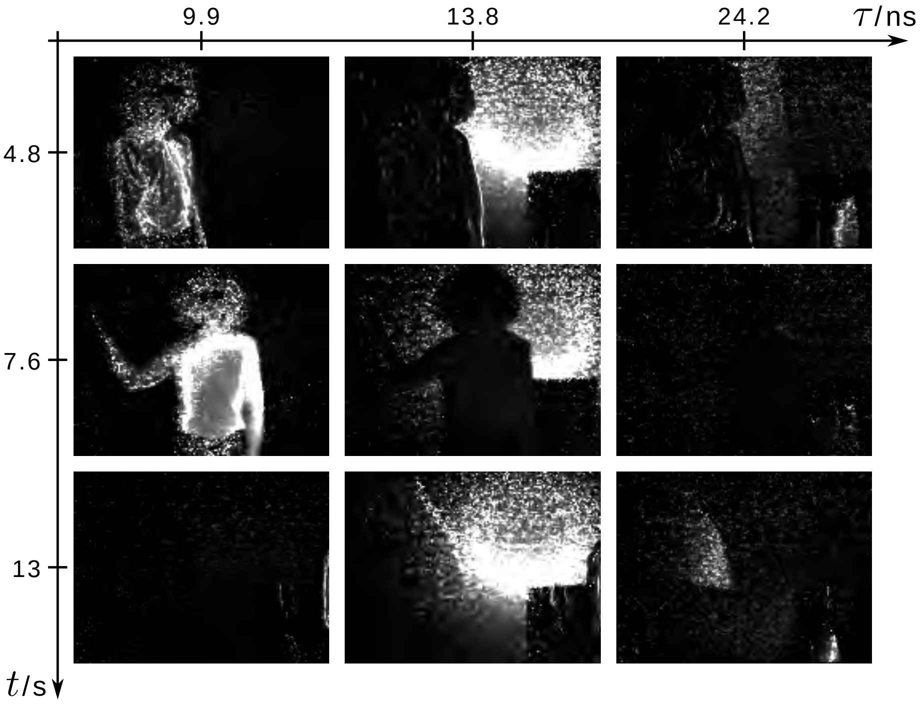


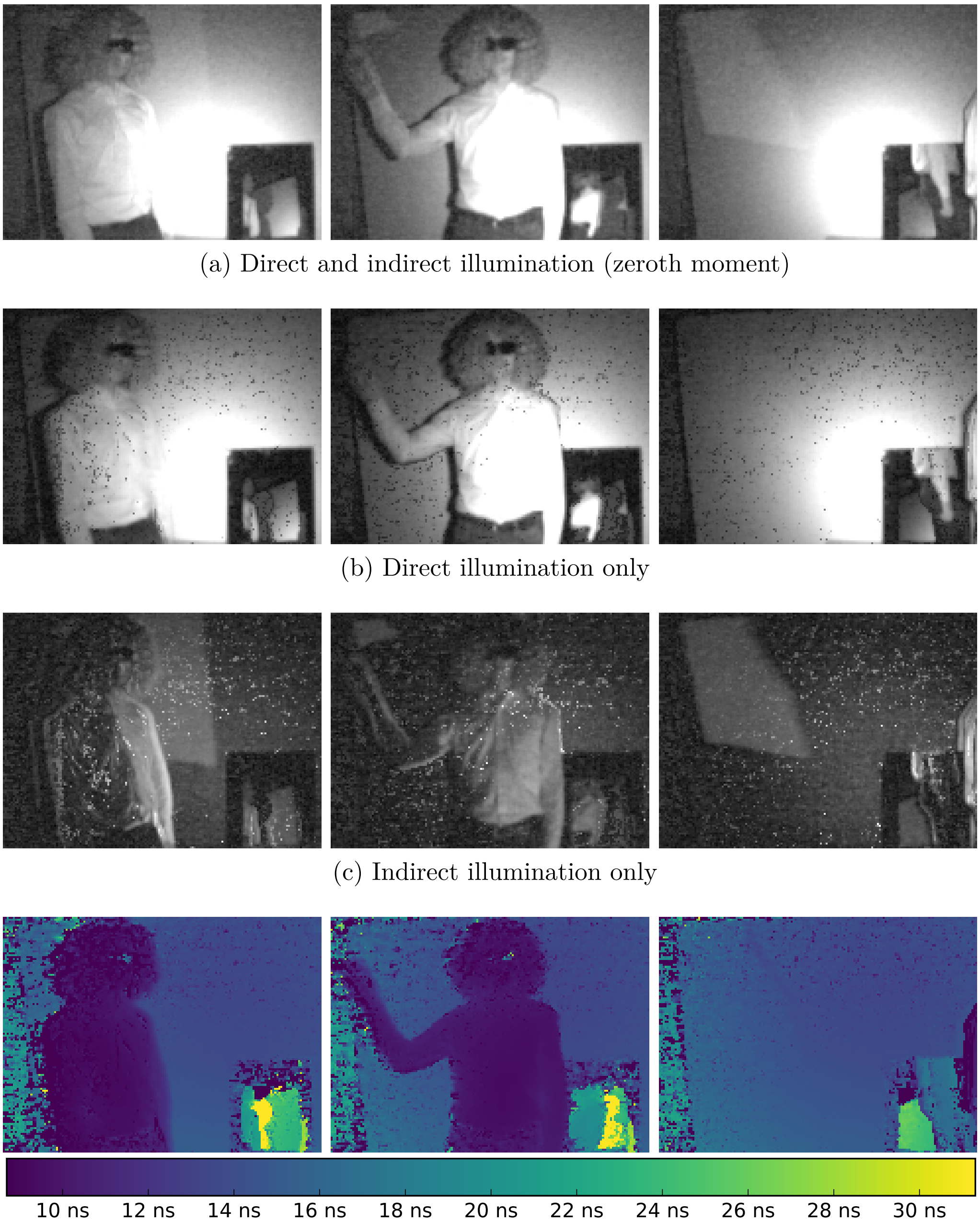
Figure 8.15: A transient video visualized through densities. It can be interpreted as four-dimensional image, parameterized over time of flight, time and two spatial dimensions. The light wave progresses through the scene as time of flight increases whereas the actor moves through the scene as time increases.

as Figure 8.15 but this time direct and indirect illumination are separated. The indirect component exhibits a few outliers but generally separates the lighting due to the mirrors correctly from other lighting. Thanks to this correct separation, the time of flight for the direct return is free of multipath interference.

The sum of both components is the zeroth moment shown in Figure 8.16a. This image has not been measured directly but has been computed from three frequency measurements as described in Section 8.3.3. This is of interest by itself because it provides a method to compute images including solely active illumination.

### 8.5.4 Conclusions

Our proposed reconstruction algorithms provide powerful ways to transfer AMCW lidar measurements from the frequency domain to the time domain.



(a) Direct and indirect illumination (zeroth moment)

(b) Direct illumination only

(c) Indirect illumination only

(d) Time of flight for the direct return

Figure 8.16: Separation of direct and indirect illumination in a transient video. Note that the lighting due to the mirrors is separated from direct illumination and does not distort the time of flight.

By using the data as hard constraints and making reasonable assumptions about the structure of the solution, they achieve good reconstructions with a minimal amount of data. At the same time, they are theoretically well-understood and provide computationally efficient solutions for a wide range of problems.

While our results demonstrate the robustness of the algorithms, their quality is limited by our prototype hardware. The used sensor began shipping in 2012 and cannot be considered state of the art. Among with the customization of the hardware, this leads to longer capture times, lower signal-to-noise ratios and temperature drifts that can invalidate the calibration. Since the setup lacks active cooling, we performed the calibration and all measurements after a warm-up phase but this further degrades the signal-to-noise ratio.

AMCW lidar systems have seen massive improvements in recent years. Newer sensors provide higher resolutions, better signal-to-noise ratios, faster capture and greater robustness. These sensors are distributed to the mass market with products such as Microsoft Kinect for Xbox One, Microsoft HoloLens and mobile devices using sensors by pmdTechnologies with Google's Project Tango technology.

This hardware does support the required frequency ranges [Bamji et al. 2015][3]. Application of our methods is a matter of implementing them at the proprietary, lower levels of the software. The biggest problem in this process will likely be the generation of a sinusoidal modulation. Since different hardware imposes different limitations, our novel approach may not be applicable and there is potential for future work.

The most immediate use from implementing our techniques will be the elimination of multipath interference in range images. Another advantage is that mixed pixels at silhouettes of objects can be resolved. Beyond that, the related work on transient imaging shows the potential of this imaging modality in computational photography, non-line-of-sight imaging and computer vision. With wide-spread availability of the required hard- and software, research in this field could gain massive traction.

The temporal resolution of all transient imaging techniques using AMCW lidar is limited by the maximal modulation frequency of current hardware (ca. 100 MHz). Recent developments in engineering may raise this upper limit substantially [Gupta et al. 2015]. Eventually, the temporal resolution

[3] www.infineon.com/dgdl/Infineon-REAL3+Image+Sensor+Family-PB-v01_00-EN.PDF?fileId=5546d462518ffd850151a0afc2302a58 (retrieved on 1st of September 2016).

might approach that of techniques with more expensive equipment [Velten et al. 2013; Gkioulekas et al. 2015]. With access to finer time scales, transient imaging enables acquisition of subsurface scattering [Wu et al. 2014] and fluorescence [Gao et al. 2014].

# Closure

# Chapter 9

# Conclusions

Our work draws directly on abstract mathematical literature but culminates in methods of immediate practical use. They are easy to apply in spite of their theoretical foundation. Substantial effort has been necessary to ensure robust behavior in all situations while maintaining a low computational complexity, especially for moment shadow mapping. However, these efforts lead to techniques that are ready for widespread industrial application.

Indeed, the publications have sparked considerable interest among practitioners. At least one game development studio[1] is already using moment shadow mapping in production. The lecture on moment shadow mapping [Peters and Klein 2015; Peters et al. 2016] at the Game Developers Conference Europe 2016 was well received. There is clearly considerable interest in the games industry.

Several ongoing trends will make moment shadow mapping highly attractive in the next years. In our evaluation we have seen that 64-bit moment shadow maps are clearly superior to all other filterable shadow maps using 64 bits per texel. Whether they outperform common shadow maps primarily depends on the output resolution. There is a trend towards higher resolutions. Displays with a resolution of $3840 \cdot 2160$ have become affordable, Sony is about to release a console to support these resolutions and Microsoft will do so in 2017. Head-mounted displays are gaining importance rapidly and require stereoscopic rendering at high resolutions and extreme frame rates.

[1] www.readyatdawn.com. Use of moment shadow mapping has been confirmed by Matt Pettineo in private correspondence and publicly at twitter.com/MyNameIsMJP/status/731613160503955456 (retrieved on 1st of September 2016).

These developments make moment shadow mapping attractive because the overhead per shaded fragment is low compared to approaches based on common shadow maps. While the cost per texel of the shadow map is higher, use of multisample antialiasing diminishes the need for high-resolution shadow maps and ultimately the required shadow map resolution is not linked to the output resolution but to the intended hardness of the shadow.

Our techniques for translucent occluders, soft shadows and single scattering inherit the same traits and are thus equally attractive. So far percentage-closer soft shadows has been the most widely used technique for soft shadows in performance-sensitive real-time applications. Moment soft shadow mapping is faster in nearly all situations. It introduces slight light leaking but diminishes aliasing and surface acne.

For single scattering, widely used solutions apply ray marching at low resolutions and then perform bilateral upscaling. In stereoscopic rendering, such upscaling leads to discrepancies between the two images which contribute to simulator sickness. Prefiltered single scattering with six moments offers a comparably low run time at full resolution. The publication of these three extensions to moment shadow mapping is too recent to judge their actual impact in the industry but we hope that the recent lecture at the Game Developers Conference Europe 2016 fosters it.

Our work on fast transient imaging is likely to find industrial use as well. After presenting and discussing it at the headquarters of pmdTechnologies AG, I know that there are plans to implement it. The technique has the necessary traits to become the default way of processing AMCW lidar measurements. It is well-understood, robust in presence of diffuse and specular multipath interference, enables various tradeoffs between quality and measurement time and provides efficient algorithms for a wide range of problems besides the reconstruction of a transient image.

With its use in Microsoft HoloLens and Google's Project Tango, AMCW lidar is on its way to become the standard depth-sensing modality for mixed-reality applications. In these applications, depth sensing is not only used for acquisition but also for accurate positional tracking. Avoiding the systematic depth distortions that arise from multipath interference in adverse environments is of utmost importance.

## 9.1 Future Work

Beyond the specific applications of the theory of moments described in this dissertation, we hope that our work establishes this powerful set of tools in computer graphics. To the best of our knowledge, its extensive use in a graphics context is novel. Variance shadow mapping [Donnelly and Lauritzen 2006] uses second-order moments and some earlier works on removal of multipath interference use methods that are related to the Pisarenko estimate [Kirmani et al. 2013; Bhandari et al. 2014a]. However, we are not aware of any earlier work in graphics that uses multiple higher-order moments as hard constraint to reconstruct a finite measure.

Speaking very broadly, any application requiring a compact, filterable representation and fast reconstruction of a non-negative function may benefit from the theory of moments. This is particularly true, if the function is sparse, i.e. if most of the area below its graph is found near few points. In this case, few moments guarantee a surprisingly good reconstruction with sharp bounds on the approximation error (see Section 8.4.2).

In real-time rendering, this situation often arises through filtering. Each individual point on a surface is associated with a host of attributes such as depth (in view or light space), normal and opacity. Aliasing arises if these attributes are only evaluated at a single point. However, none of these attributes can be filtered directly. Moments offer a generic way to store the distribution of these attributes within a filter region compactly and in such a way that it can be filtered directly. Then all the powerful hardware-accelerated filtering efficiently resolves issues with aliasing.

Specific directions for future work would be the application to normal distributions for specular antialiasing [Toksvig 2005], order-independent transparency [Enderton et al. 2010] and volumetric obscurance [Loos and Sloan 2010; Hendrickx et al. 2015]. Low-dimensional representations for spherical distributions of radiance hold the potential to improve heuristic global illumination approaches [Papaioannou 2011].

Concerning our work on transient imaging, we are anticipating its implementation in commercial products. If cheap and fast transient imaging becomes widely available in mobile devices, it can aid tasks in computational photography, computer vision [Wu et al. 2014] and non-line-of-sight imaging [Velten et al. 2012]. To enable the implementation on a broad range of hardware, future work should investigate further ways to robustly construct sinusoidal modulation signals from the signals supported by the hardware. Registration between trigonometric moment images captured at slightly different times in dynamic scenes is another promising endeavor [Lefloch et al. 2013].

We are looking forward to a long-lasting era of moments in graphics.

# Appendix

Appendix A

# Overview of Previously Unpublished Contributions

The contents of this dissertation go beyond the contents of the three publications listed in Section 1.1. There are many minor contributions that have not been published in a peer-reviewed paper yet. Most of them are designated to become part of the invited extension of the most recent paper [Peters et al. 2016] in the Journal of Computer Graphics Techniques. For the convenience of readers who have already read the original publications, we provide a complete overview of these contributions in the present Section.

In addition to these contributions, the related work sections have been extended to include more recent publications.

## A.1 Moment Shadow Mapping

Here we describe previously unpublished contributions pertaining to contents of the first paper on moment shadow mapping [Peters and Klein 2015].

**Evaluation of Candidate Techniques (Section 3.4)** The evaluation of candidate techniques in the paper has been distorted systematically by inappropriately large default tolerances in the used linear programming solvers. Therefore, we repeated the evaluation with lower error tolerances on a subset of 6605 randomly selected candidate techniques.

**Accounting for Rounding Errors Explicitly (Section 4.1.3)** We have found that linear programming can incorporate the rounding errors in the provided power moments as inequality constraints. This observation and the comparisons it leads to (see Figure 4.2) are novel.

**Biasing for the Worst Case (Section 4.1.3)** The vector of biasing moments $b^\star$ in the paper is optimized for a particular average case. However, it cannot guarantee good results in all cases. It corresponds to a depth distribution with support at the two boundaries of the domain of depth values. If biasing is applied to a vector of moments with the same property, biasing may fail to result in a valid vector of moments. To achieve greater robustness, the new vector of biasing moments is optimized for the worst case and incorporates the effect of the quantization transform.

**Signed Depth (Section 4.1.5)** In the paper depth is consistently defined as a quantity in the interval [0, 1]. Throughout the dissertation we define it in the interval $[-1, 1]$ because we found that this provides substantially increased numerical stability. When storing the power moments in single-precision floats, light leaking is reduced due to a weaker moment bias $\alpha_b$.

**Optimized Quantization Transform (Section 4.1.4)** The globally optimal quantization transform proposed in the original paper uses a $4 \times 4$ matrix without vanishing entries. As a consequence of the use of signed depth values, it is now possible to use a quantization transform which is only slightly worse while having eight vanishing entries. This reduces the computational complexity.

**sRGB and Overdarkening (Section 4.2)** Imagery shown in the original paper used linear colors. Throughout the dissertation, all shown images are in sRGB. This conversion does strengthen the perceived light leaking considerably and we discuss this effect. As a countermeasure we use overdarkening as proposed by Annen et al. [2007].

**Comparisons to Exponential Variance Shadow Mapping (Section 4.2)** The original paper discussed exponential variance shadow mapping but it was not included in competitive comparisons. We now provide extensive comparisons to exponential variance shadow mapping with 64 and 128 bits per texel.

**More Robust Trigonometric Moment Shadow Mapping (Appendix B.4.2)** The implementation of trigonometric moment shadow mapping in the paper used Cramer's rule to compute $C^{-1}(c) \cdot \mathbf{c}(\pi \cdot z)$. The implementation used for evaluation in the dissertation uses a complex Cholesky decomposition. This results in improved robustness and makes 64-bit trigonometric moment shadow mapping superior to 64-bit Hamburger moment shadow mapping in terms of light leaking. Still, a few robustness issues remain and the technique is too slow for serious use.

**Optimizations (Section 4.2.2)** All evaluated techniques are implemented more efficiently now. A list of all major optimizations follows:

- Common shadow maps are now created as depth buffer rather than by rendering to a render target,
- Filterable shadow maps are created during a custom resolve of a multisampled depth buffer rather than by rendering to a render target,
- Mipmap hierarchies for filterable shadow maps are only created when they are needed, i.e. only for the texture that arises after all other filtering operations,
- The optimized quantization transform with the eight vanishing entries is used for moment shadow mapping.

The new run time measurements include higher output resolutions.

## A.2 Applications of Moment Shadow Mapping

Next we discuss previously unpublished contributions which apply specifically to our three novel applications of moment shadow mapping [Peters et al. 2016]. Note that the point about sRGB and overdarkening above also applies here.

### A.2.1 Translucent Occluders

**More Comparisons (Section 5.3)** We now compare our approach to analogous approaches with variance shadow mapping and exponential variance shadow mapping using 64 or 128 bits per texel.

**Discussion of Rounding Errors (Section 5.3)** The potential of increased rounding errors due to alpha blending is discussed now.

### A.2.2 Soft Shadows

**Blocker Search (Section 6.3)** In the original paper we recommend using the biased fragment depth as input to the moment-based blocker search. This does in fact degrade the quality of the results in situations with three surfaces in the search region. Therefore, we now use the unbiased fragment depth.

**Adaptive Depth Biasing (Section 6.4)** In the original paper the depth bias was a global constant. Now we increase the depth bias in proportion to the filter size to diminish the artifacts shown in Figure 6.5.

**Optimizations (Section 6.6.2)** The implementation of moment soft shadow mapping and naïve variance soft shadow mapping has become more efficient because it no longer generates a mipmap hierarchy for the summed-area table. Mipmaps are not needed for filtering because the summed-area table supports arbitrary rectangular filter regions. They may still be used to improve cache efficiency but we opted against this because the overhead for their generation is significant. The optimizations listed for moment shadow mapping above are inherited.

### A.2.3 Single Scattering

**Adaptive Overestimation (Section 7.3.1)** For results in the paper we consistently used $\beta = 0.5$. Now we describe adaptive overestimation as an efficient way to diminish leaking at epipoles.

**Improved Six Moment Shadow Mapping (Section 7.4)** Our implementation of six moment shadow mapping has become more efficient and more robust. Just like four moment shadow mapping, it now uses signed depth and a quantization transform where half of the entries vanish. Additionally, the cubic equation is solved more efficiently now. This solution also eliminates the artifacts that used to arise when the leading coefficient was small.

**Further Optimizations (Section 7.5.2)** The implementation of all variants of prefiltered single scattering has become faster for various reasons:

- Mipmap hierarchies are only created when they are needed, i.e. for the prefiltered convolution or moment shadow map,

- Rectification is now done in a separate pixel shader pass rather than doing it on the fly as part of the compute shader that generates prefix sums. This way, parallelism for the computationally intense steps is improved,
- The compute shader generating the prefix sums has been optimized (see Section C.3.2).

## A.3 Transient Imaging

Since our publication on transient imaging [Peters et al. 2015] is in a journal already, we have not revisited as many aspects as for moment shadow mapping and its applications. Nonetheless, there is one noteworthy contribution.

**Efficient Computation of Cumulative Transient Images (Sections 8.2.3 and 8.4.4)** In the paper we propose to compute cumulative transient images using numerical quadrature. Since impulse responses are often strongly localized, this required roughly $10^5$ samples for sound results. Here we describe a closed-form solution for computation of cumulative transient images. The only step which cannot be done with a closed form for $m > 4$ is polynomial root finding. To build up to this solution, we interpret the maximum entropy spectral estimate in terms of its poles.

Appendix B

# Derivations and Proofs

## B.1 Optimal Biasing for Moment Shadow Mapping

In Section 4.1.3 we explain our biasing strategy for moment shadow mapping. Though, we have not described the details of determining the vector of biasing moments $b^\star \in \mathbb{R}^{m+1}$. We now deliver a discussion of this procedure for the various cases.

Our motivation in Section 4.1.3 explains that we are looking for a vector of biasing moments that corresponds to a depth distribution on $[-1, 1]$, i.e. $b^\star \in \operatorname{conv} \mathbf{b}([-1, 1])$ by Proposition 2.5. At the same time we want to maximize the minimal distance to the topological boundary[1] of the domain of valid vectors of moments $\partial \operatorname{conv} \mathbf{b}(\mathbb{R})$. Speaking formally, we need to compute a $b^\star \in \operatorname{conv} \mathbf{b}([-1, 1])$ maximizing the functional

$$R_\partial(b^\star) := \inf_{b \in \partial \operatorname{conv} \mathbf{b}(\mathbb{R})} \|b - b^\star\|_2.$$

Although we only need to solve it once, this optimization problem is too complex to be approached with complete brute force. To make it tractable, we first reduce the dimensionality of the set that could be containing the optimal $b^\star$.

**Proposition B.1.** *The functional $R_\partial$ is concave on* $\operatorname{conv} \mathbf{b}(\mathbb{R})$*. Maximizing it is accomplished with a vector $b^\star \in$* $\operatorname{conv} \mathbf{b}([-1, 1])$ *where all odd power moments vanish, i.e.*

$$b^\star = (b_0^\star, 0, b_2^\star, 0, \ldots, b_m^\star)^\mathsf{T}.$$

[1] We define this boundary using the subspace topology in $\{1\} \times \mathbb{R}^m$.

*Furthermore, the entry $b^\star_m$ can be chosen maximal in the sense that any greater value would immediately violate $b^\star \in \operatorname{conv} \mathbf{b}([-1,1])$.*

*Proof.* To prove the concavity of $R_\partial$, we represent it using signed distances to hyperplanes. Consider the set of hyperplanes having $\mathbf{b}(\mathbb{R})$ in their non-negative half-space:

$$\mathbb{H} := \{(\nu, d) \in \mathbb{R}^{m+1} \times \mathbb{R} \mid \nu_0 = 0,\ \|\nu\|_2 = 1,\ \forall z \in \mathbb{R} :\ \nu^\mathsf{T} \cdot \mathbf{b}(z) + d \geq 0\}$$

For all $b^\star \in \operatorname{conv} \mathbf{b}(\mathbb{R})$ and $(\nu, d) \in \mathbb{H}$ we know $\nu^\mathsf{T} \cdot b^\star + d \geq 0$ and we can rewrite $R_\partial$ as minimal signed distance to a hyperplane that is tangent to the convex hull:

$$R_\partial(b^\star) = \inf_{(\nu,d)\in\mathbb{H}} \nu^\mathsf{T} \cdot b^\star + d$$

Now we verify the concavity for arbitrary $b^\star, b'^\star \in \operatorname{conv} \mathbf{b}(\mathbb{R})$ and $\lambda \in [0, 1]$:

$$\begin{aligned}
&R_\partial(\lambda \cdot b^\star + (1-\lambda) \cdot b'^\star) \\
=& \inf_{(\nu,d)\in\mathbb{H}} \nu^\mathsf{T} \cdot (\lambda \cdot b^\star + (1-\lambda) \cdot b'^\star) + d \\
=& \inf_{(\nu,d)\in\mathbb{H}} \lambda \cdot (\nu^\mathsf{T} \cdot b^\star + d) + (1-\lambda) \cdot (\nu^\mathsf{T} \cdot b'^\star + d) \\
\geq& \inf_{(\nu,d)\in\mathbb{H}} \lambda \cdot (\nu^\mathsf{T} \cdot b^\star + d) + \inf_{(\nu,d)\in\mathbb{H}} (1-\lambda) \cdot (\nu^\mathsf{T} \cdot b'^\star + d) \\
=& \lambda \cdot R_\partial(b^\star) + (1-\lambda) \cdot R_\partial(b'^\star)
\end{aligned}$$

Let $b^\star \in \operatorname{conv} \mathbf{b}(\mathbb{R})$. By Proposition 2.5 there exists a distribution $Z$ on $\mathbb{R}$ such that $b^\star = \mathcal{E}_Z(\mathbf{b})$. We consider the flipped vector

$$b'^\star := \mathcal{E}_Z(\mathbf{b}(-\mathbf{z})) = (b^\star_0,\ -b^\star_1,\ b^\star_2,\ -b^\star_3, \ldots, b^\star_m)^\mathsf{T}.$$

By Proposition 2.5, this vector is still in $\operatorname{conv} \mathbf{b}(\mathbb{R})$. Applying the same transform to the nearest point on the boundary yields $R_\partial(b^\star) = R_\partial(b'^\star)$. Now we exploit the concavity of $R_\partial$ to conclude

$$R_\partial((b^\star_0,\ 0,\ b^\star_2,\ 0, \ldots, b^\star_m)^\mathsf{T}) = R_\partial\left(\frac{b^\star + b'^\star}{2}\right) \geq \frac{R_\partial(b^\star) + R_\partial(b'^\star)}{2} = R_\partial(b^\star).$$

Combining Propositions 2.5 and 2.8, we know that $b \in \mathbb{R}^{m+1}$ is in $\operatorname{conv} \mathbf{b}(\mathbb{R})$ if and only if the Hankel matrix $B(b)$ is positive semi-definite. For all

$b \in \operatorname{conv} \mathbf{b}(\mathbb{R})$ and $\varepsilon_m \geq 0$ we observe

$$\begin{aligned}\det B(b + \varepsilon_m \cdot e_m) &= \det B(b) + \det \begin{pmatrix} b_0 & \cdots & b_{\frac{m}{2}-1} & 0 \\ b_1 & \iddots & \vdots & \vdots \\ \vdots & \iddots & \vdots & 0 \\ b_{\frac{m}{2}} & \cdots & b_{m-1} & \varepsilon_m \end{pmatrix} \\ &= \det B(b) + \varepsilon_m \cdot \det B((b_0,\, b_1,\, \ldots,\, b_{m-2})^\mathsf{T}) \geq \det B(b).\end{aligned}$$

Thus, the determinant of $B(b)$ grows monotonically as the last power moment $b_m$ is increased. The determinant of the main minors of $B(b)$ is completely independent of the last power moment. In consequence, the matrix remains positive semi-definite and thus

$$b \in \operatorname{conv} \mathbf{b}(\mathbb{R}) \qquad \Rightarrow \qquad b + \varepsilon_m \cdot e_m \in \operatorname{conv} \mathbf{b}(\mathbb{R}).$$

Now for a point $b^\star \in \operatorname{conv} \mathbf{b}([-1, 1])$, consider an open ball of radius $R_\partial(b^\star)$ centered around $b^\star$. By definition, it lies completely within $\operatorname{conv} \mathbf{b}(\mathbb{R})$ and hence the same holds for a ball of radius $R_\partial(b^\star)$ around $b^\star + \varepsilon_m \cdot e_m$. We conclude

$$R_\partial(b^\star + \varepsilon_m \cdot e_m) \geq R_\partial(b^\star).$$

□

Proposition B.1 reduces the dimensionality of the relevant search space considerably. We know that $b_0^\star$ has to be one, odd moments have to be zero and the last moment $b_m^\star$ has to be maximal. For $m = 4$ only $b_2^\star$ remains to be optimized, for $m = 6$ it is $b_2^\star$ and $b_4^\star$. The concavity of $R_\partial$ could be exploited for an efficient optimization but in such a small search space it is easier to use brute force.

We begin with the case $m = 4$ and start by determining the maximal admissible value for $b_4^\star$. Let $Z$ be a distribution on $[-1, 1]$ and $b^\star := \mathcal{E}_Z(\mathbf{b})$. Then we know

$$b_4^\star = \mathcal{E}_Z(\mathbf{b}_4) = \mathcal{E}_Z\left(\mathbf{z}^4\right) = \mathcal{E}_Z\left(|\mathbf{z}|^2 \cdot |\mathbf{z}|^2\right) \leq \mathcal{E}_Z\left(|\mathbf{z}|^2 \cdot 1\right) = b_2^\star.$$

This bound is sharp with equality if $Z = \frac{b_2^\star}{2} \cdot (\delta_{-1} + \delta_1) + (1 - b_2^\star) \cdot \delta_0$:

$$\mathcal{E}_Z(\mathbf{b}_4) = \frac{b_2^\star}{2} \cdot (1 + 1) + 0 = \mathcal{E}_Z(\mathbf{b}_2)$$

At the same time this distribution realizes $b_0^\star = 1$ and $b_1^\star = b_3^\star = 0$. Thus, the optimal vector of biasing moments for the case $m = 4$ takes the form

$$b^\star = (1,\, 0,\, b_2^\star,\, 0,\, b_2^\star)^\mathsf{T}. \tag{B.1}$$

A brute-force search leads to the optimum

$$b^\star = (1,\, 0,\, 0.375,\, 0,\, 0.375)^\mathsf{T}.$$

To incorporate the quantization transform $\Theta_4^\star$ into our considerations, we change the functional that is maximized to

$$\inf_{b \in \partial \operatorname{conv} \mathbf{b}(\mathbb{R})} \|\Theta_4^\star(b) - \Theta_4^\star(b^\star)\|_2.$$

Thanks to the special structure of the transform in Equation (4.4), all arguments used above carry over and the optimal solution still has the structure shown in Equation (B.1). Another brute-force optimization with the new functional then leads to

$$b^\star = (1,\, 0,\, 0.628,\, 0,\, 0.628)^\mathsf{T}.$$

Next we derive the bias for $m = 6$ using the optimal quantization from Section 7.4.3. Again we start by maximizing $b_6^\star$ in closed form. According to Kreĭn and Nudel'man [1977, p. 62 f.] $b^\star \in \mathbb{R}^{m+1}$ is in $\operatorname{conv} \mathbf{b}([-1,1])$ if and only if $B(b^\star)$ is positive semi-definite and

$$\begin{pmatrix} b_0^\star - b_2^\star & b_1^\star - b_3^\star & b_2^\star - b_4^\star \\ b_1^\star - b_3^\star & b_2^\star - b_4^\star & b_3^\star - b_5^\star \\ b_2^\star - b_4^\star & b_3^\star - b_5^\star & b_4^\star - b_6^\star \end{pmatrix}$$

is also positive semi-definite. We substitute in our prior knowledge about the structure of an optimal $b^\star$ and then demand that all main minors have positive determinant:

$$\begin{aligned}
&\det\left(1 - b_2^\star\right) \geq 0 \\
\Rightarrow\ & b_2^\star \leq 1 \\
&\det\begin{pmatrix} 1 - b_2^\star & 0 \\ 0 & b_2^\star - b_4^\star \end{pmatrix} \geq 0 \\
\Rightarrow\ & b_4^\star \leq b_2^\star \\
&\det\begin{pmatrix} 1 - b_2^\star & 0 & b_2^\star - b_4^\star \\ 0 & b_2^\star - b_4^\star & 0 \\ b_2^\star - b_4^\star & 0 & b_4^\star - b_6^\star \end{pmatrix} \geq 0 \\
\Rightarrow\ & (b_4^\star - b_6^\star) \cdot (b_2^\star - b_4^\star) \cdot (1 - b_2^\star) - (b_2^\star - b_4^\star)^3 \geq 0 \\
\Rightarrow\ & (b_4^\star - b_6^\star) \cdot (1 - b_2^\star) \geq (b_2^\star - b_4^\star)^2 \\
\Rightarrow\ & b_6^\star \leq b_4^\star - \frac{(b_2^\star - b_4^\star)^2}{1 - b_2^\star}
\end{aligned}$$

Thus, an optimal bias has the structure

$$b^\star = \left(1,\, 0,\, b_2^\star,\, 0,\, b_4^\star,\, 0,\, b_4^\star - \frac{(b_2^\star - b_4^\star)^2}{1 - b_2^\star}\right)^\mathsf{T}.$$

Performing brute-force optimization on $b_2^\star$, $b_4^\star$ leads to

$$b^\star := (0,\, 0.5566,\, 0,\, 0.489,\, 0,\, 0.47869382)^\mathsf{T}.$$

## B.2 Scale and Translation Invariance of Hamburger Moment Shadow Mapping

In Section B.2 we have claimed that Hamburger moment shadow mapping is the only shadow mapping technique based on Problem 3.1 for which changes of the near and far clipping planes do not change the outcome unless geometry is clipped. Below we make this statement precise and provide a proof.

We observe that the information conveyed by a filterable shadow map only depends upon the space $\langle \mathbf{a}_0, \ldots, \mathbf{a}_m \rangle$ spanned by the component functions of its moment-generating function. If two different moment-generating functions span the same space, a linear transform maps the corresponding general moments onto each other. To ensure that a technique is invariant under scaling and translation of depth values, this space has to be invariant under these operations. Only spaces of polynomials have this property:

**Proposition B.2.** *Let $\mathbb{V}$ be an $(m+1)$-dimensional vector space of functions $\mathbf{f} : \mathbb{R} \to \mathbb{R}$. The space $\mathbb{V}$ has the property*

$$\forall \mathbf{f} \in \mathbb{V},\, x \in \mathbb{R} \setminus \{0\},\, y \in \mathbb{R} :\, z \mapsto \mathbf{f}(x \cdot z + y) \in \mathbb{V} \tag{B.2}$$

*if and only if $\mathbb{V} = \langle \mathbf{b}_0, \ldots, \mathbf{b}_m \rangle$, i.e. $\mathbb{V}$ consists of all polynomials up to degree $m$.*

*Proof.* "$\Leftarrow$" Let $\mathbb{V} = \langle \mathbf{b}_0, \ldots, \mathbf{b}_m \rangle$.

Then $\mathbf{f}(x \cdot z + y)$ is the concatenation of a linear polynomial with a polynomial of degree $m$ or less and thus still a polynomial of degree $m$ or less.

"$\Rightarrow$" Suppose Statement (B.2) holds.

Let $\mathbf{f} \in \mathbb{V}$ and for $\varepsilon > 0$ consider the divided difference

$$\frac{\mathbf{f}(z+\varepsilon) - \mathbf{f}(z)}{\varepsilon}.$$

Due to Statement (B.2), $\mathbf{f}(z+\varepsilon)$ is a function in $\mathbb{V}$ and since $\mathbb{V}$ is a vector space, the entire divided difference is also in $\mathbb{V}$. Since $\mathbb{V}$ is finite-dimensional, this is also true for the limit

$$\mathbf{f}'(z)=\lim_{\varepsilon\to 0}\frac{\mathbf{f}(z+\varepsilon)-\mathbf{f}(z)}{\varepsilon}.$$

In particular, $\mathbf{f}$ is differentiable. The differential $\mathrm{d}$ acts as a linear operator $\mathrm{d}:\mathbb{V}\to\mathbb{V}$. Let $\lambda\in\mathbb{C}$ be an eigenvalue of $\mathrm{d}$ with eigenvector $\mathbf{v}\in\mathbb{V}$ where $\mathbf{v}(0)=1$.

Suppose $\lambda\neq 0$: Then the initial value problem $\mathbf{v}'=\lambda\cdot\mathbf{v}$ with $\mathbf{v}(0)=1$ has the unique solution $\mathbf{v}(z)=\exp(\lambda\cdot z)$. However, the functions

$$\exp(\lambda\cdot x\cdot z)=\exp(\lambda\cdot z)^x$$

are linearly independent for different values of $x\in\mathbb{N}$. Still all of them lie in $\mathbb{V}$ contradicting our knowledge that $\mathbb{V}$ has dimension $m+1$.

Thus, $\mathrm{d}$ only has vanishing eigenvalues and therefore it is a nilpotent operator. In consequence, $\mathrm{d}^{m+1}\,\mathbf{f}=0$ for all $\mathbf{f}\in\mathbb{V}$ and therefore $\mathbf{f}$ has to be a polynomial of degree $m$ or less. □

## B.3 Hausdorff Moment Shadow Mapping

Hamburger moment shadow mapping is not among the candidate techniques evaluated in Section 3.4 because it does not incorporate the prior knowledge that $\mathbb{I}=[-1,1]$. Incorporating this knowledge yields Hausdorff moment shadow mapping which gives identical results in most cases but better results in others. Generation of the moment shadow map, quantization and biasing are the same as for Hamburger moment shadow mapping. Only the solution to Problem 3.1 works differently.

We recall that Algorithm 4.1 constructs a distribution with $\frac{m}{2}+1$ points of support. If all points of support lie in $[-1,1]$, this distribution provides the optimal solution but to describe all cases, we need another version of the Markov-Kreĭn theorem.

**Theorem B.3** (Markov-Kreĭn for $\mathbb{I}=[-1,1]$)**.** *Let $b\in\operatorname{conv}\mathbf{b}([-1,1])$ such that $B(b)$ is positive definite and let $z_f\in\mathbb{R}$. Then there exists exactly one probability distribution $S\in\mathbb{P}([-1,1])$ with $\mathcal{E}_S(\mathbf{b})=b$ having support at $z_f$ and either at $\frac{m}{2}$ additional points or at $-1$, $1$ and $\frac{m}{2}-1$ additional points. It solves Problem 3.1, i.e.*

$$S(z_f>\mathbf{z})=G_{[-1,1],\mathbf{b}}(b,z_f)=\inf_{\substack{S'\in\mathbb{P}([-1,1])\\ \mathcal{E}_{S'}(\mathbf{b})=b}}S'(z_f>\mathbf{z}).$$

*As in Theorem 4.1, the corresponding optimal upper bound is attained when we include the support at $z_f$.*

*Proof.* We refer to the literature for proofs of existence [Kreĭn and Nudel'man 1977, p. 58, 79] and optimality [Kreĭn and Nudel'man 1977, p. 125 f.]. □

In the case with support at $z_f$ and $\frac{m}{2}$ additional points, Algorithm 4.1 is capable of constructing the relevant distribution but we need a new algorithm for the other case. For simplicity we restrict this derivation to the case $m = 4$. Again the difficulty lies in the computation of the points of support. However, only $\frac{m}{2} - 1 = 1$ point is unknown.

**Proposition B.4.** *Let $z_0 := -1$, $z_1 \in [-1,1]$, $z_2 = z_f \in [-1,1]$ and $z_3 = 1$ be pairwise different and let $w_0, w_1, w_2, w_3 > 0$. Let $S := \sum_{l=0}^{3} w_l \cdot \delta_{z_l}$ and let $b := \mathcal{E}_S(\mathbf{b})$ with $b_0 = 1$. Then*

$$z_1 = \frac{(b_1 - b_3) \cdot z_f + b_4 - b_2}{(1 - b_2) \cdot z_f + b_3 - b_1}. \tag{B.3}$$

*Proof.* Since $b = \mathcal{E}_S(\mathbf{b})$, the following matrix has to be singular:

$$(A \mid b) := \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ -1 & z_1 & z_f & 1 & b_1 \\ 1 & z_1^2 & z_f^2 & 1 & b_2 \\ -1 & z_1^3 & z_f^3 & 1 & b_3 \\ 1 & z_1^4 & z_f^4 & 1 & b_4 \end{pmatrix}$$

Then $\det(A \mid b)$ is a polynomial in $z_1$ and $z_f$. Obviously, it has roots for $z_1 \in \{-1, z_f, 1\}$ as well as $z_f \in \{-1, 1\}$ but since $-1, z_1, z_f, 1$ are pairwise different, none of them is relevant. We perform polynomial division to remove them and obtain

$$\begin{aligned} 0 &= \frac{\det(A \mid b)}{2 \cdot (z_1 + 1) \cdot (z_1 - z_f) \cdot (z_1 - 1) \cdot (z_f + 1) \cdot (z_f - 1)} \\ &= z_1 \cdot ((1 - b_2) \cdot z_f + b_3 - b_1) + (b_3 - b_1) \cdot z_f + b_2 - b_4. \end{aligned}$$

This expression cannot be zero for all values of $z_1$ because otherwise

$$b \in \langle \mathbf{b}(-1), \mathbf{b}(z_f), \mathbf{b}(1) \rangle$$

and that would contradict $w_1 \neq 0$. Thus, we can solve for $z_1$ and obtain Equation (B.3). □

**Algorithm B.1** Hausdorff moment shadow mapping, i.e. the solution to Problem 3.1 for $m = 4$, $\mathbf{a} = \mathbf{b}$ and $\mathbb{I} = [-1, 1]$.
**Input:** Power moments $b \in \operatorname{conv} \mathbf{b}([-1, 1])$, fragment depth $z_f \in [-1, 1]$.
**Output:** The lower bound $G_{[-1,1],\mathbf{b}}(b, z_f)$ as defined in Problem 3.1 or failure.

```
1. If $B(b)$ is not positive definite: Indicate failure.
2. Solve $B(b) \cdot q = \hat{\mathbf{b}}(z_f)$ for $q \in \mathbb{R}^3$.
3. Solve $q_2 \cdot z^2 + q_1 \cdot z + q_0 = 0$ for $z$. If solutions $-1 \leq z_1 < z_2 \leq 1$ exist:
    a) Set $A := (\hat{\mathbf{b}}(z_f), \hat{\mathbf{b}}(z_1), \hat{\mathbf{b}}(z_2)) \in \mathbb{R}^{3\times 3}$.
    b) Solve $A \cdot w = (b_0, b_1, b_2)^\mathsf{T}$ for $w \in \mathbb{R}^3$.
    c) Return $\sum_{l=1,\, z_l<z_f}^{2} w_l$.
4. Otherwise:
    a) Set
       $$z_1 := \frac{(b_1 - b_3) \cdot z_f + b_4 - b_2}{(1 - b_2) \cdot z_f + b_3 - b_1}.$$
    b) Solve the following system of linear equations for $w \in \mathbb{R}^4$:
       $$\begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & z_1 & z_f & 1 \\ 1 & z_1^2 & z_f^2 & 1 \\ -1 & z_1^3 & z_f^3 & 1 \end{pmatrix} \cdot w = \begin{pmatrix} 1 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}$$
    c) Return $w_0 + \begin{cases} w_1 & \text{if } z_1 < z_f, \\ 0 & \text{otherwise.} \end{cases}$
```

Combining this result with Algorithm 4.1 we obtain Algorithm B.1.

**Proposition B.5.** *If it does not indicate failure, Algorithm B.1 solves Problem 3.1 correctly.*

*Proof.* We distinguish two cases for the two branches of the algorithm.

*Case 1:* The algorithm terminates in Step 3c):
We observe that the algorithm performs the exact same steps as Algorithm 4.1 except that it ensures that $z_f, z_1, z_2 \in [-1, 1]$. Thus, the optimal distribution on $\mathbb{R}$ is in fact a distribution on $[-1, 1]$ and also provides the optimal solution there.

*Case 2:* The algorithm terminates in Step 4c):
By Theorem B.3, the optimal distribution $S$ on $[-1, 1]$ must have support at $-1$, $z_f$, $1$ and one additional point $z_1$ because if it had exactly three points of support, Case 1 would be present. Thus, we are in the situation of Proposition B.4 and Step 4a) computes $z_1$ correctly. Let

$$S := w_0 \cdot \delta_{-1} + w_1 \cdot \delta_{z_1} + w_2 \cdot \delta_{z_f} + w_3 \cdot \delta_1.$$

The system of linear equations in Step 4b) is equivalent to $\mathcal{E}_S(\mathbf{b}_j) = b_j$ for all $j \in \{0, 1, 2, 3\}$. Since $-1, z_1, z_f, 1$ are pairwise different, this Vandermonde system determines the weights $w_0, w_1, w_2, w_3$ uniquely. Finally, $S(z_f > \mathbf{z})$ is returned. □

The efficient computation of the shadow intensity in the new branch is non-trivial but we describe a fast solution in Appendix C.2.

## B.4 Trigonometric Moment Shadow Mapping

In terms of quality, trigonometric moment shadow mapping is the best shadow mapping technique to date when using 64 bits per shadow map texel (Section 4.2.1). On the other hand, our algorithm for it has a substantial computational complexity (Section 4.2.2). At 64 bits per texel, it is significantly slower than the superior Hamburger moment shadow mapping with 128 bits per texel. Thus, this technique is ultimately a dead end[2]. It is interesting to study but not useful in practice for reasons that we consider inevitable.

[2] We derived trigonometric moment shadow mapping in hopes that it would be applicable to shadow mapping and transient imaging. Though, shortly after completing the algorithm for trigonometric moment shadow mapping, we came across the maximum entropy spectral estimate, which provides a more appropriate solution for transient imaging in every regard.

### B.4.1 Derivation

In the following, we derive our novel algorithm for trigonometric moment shadow mapping. More precisely, we solve Problem 3.1 for $\mathbb{I} = [-1, 1]$ and

$$\mathbf{a}(z) := (1,\, \cos(\pi \cdot z),\, \sin(\pi \cdot z),\, \cos(2 \cdot \pi \cdot z),\, \sin(2 \cdot \pi \cdot z))^\mathsf{T}.$$

It is far more involved than the algorithm for Hamburger moment shadow mapping and since the end result is not useful, we do not provide a rigorous derivation. However, we have validated the results by means of extensive comparisons to Algorithm 3.1 and could not find any discrepancies.

To solve the problem, we view the given general moments as complex trigonometric moments. Rather than working with the $\mathbf{a}(z)$ above, we work with $\mathbf{c}(\pi \cdot z)$. Correspondingly, we set $m := 2$ because four real general moments fit into two complex trigonometric moments. It is convenient to define

$$\varphi := \pi \cdot z \in [-\pi,\, \pi]\,. \tag{B.4}$$

Rather than working with depth distributions, we will be working with phase distributions that are related to depth distributions by Equation (B.4).

We note that the Toeplitz matrix $C(c)$ has to be positive semi-definite. If it is singular, Algorithm 8.2 provides the solution. Thus, we only consider the case of a positive-definite Toeplitz matrix $C(c)$.

As for the solution of Hamburger moment shadow mapping, distributions with $m + 1 = 3$ points of support play a crucial role. Kreĭn and Nudel'man [1977, p. 149] prove their existence for positive-definite $C(c)$ and arbitrary prescribed points of support. Their construction is completely analogous to the construction for Hamburger moment shadow mapping in Proposition 4.2.

**Proposition B.6.** *Let $\varphi_0, \ldots, \varphi_m \in (-\pi,\, \pi]$ be pairwise different and let $w_0, \ldots, w_m > 0$ with $\sum_{l=0}^{m} w_l = 1$. Let $S = \sum_{l=0}^{m} w_l \cdot \delta_{\varphi_l}$ and $c = \mathcal{E}_S(\mathbf{c})$. Then for all $l \in \{1, \ldots, m\}$*

$$\mathbf{c}^*(\varphi_l) \cdot C^{-1}(c) \cdot \mathbf{c}(\varphi_0) = 0.$$

*Furthermore, the weight $l \in \{0, \ldots, m\}$ is given by*

$$w_l = \frac{1}{\mathbf{c}^*(\varphi_l) \cdot C^{-1}(c) \cdot \mathbf{c}(\varphi_l)}.$$

*Proof.* We note that $C(c)$ is regular by Proposition 2.13. Let

$$A := (\mathbf{c}(\varphi_0),\, \ldots,\, \mathbf{c}(\varphi_m)) \in \mathbb{C}^{(m+1)\times(m+1)}.$$

This matrix is a square Vandermonde matrix and since $\exp(i\cdot\varphi_0), \ldots, \exp(i\cdot\varphi_m)$ are pairwise different, it is invertible. We recall from Proposition 2.11 on page 26 that $C(c) = \mathcal{E}_S(\mathbf{c}\cdot\mathbf{c}^*)$ and thus:

$$\begin{aligned}
A^{-1}\cdot C(c)\cdot A^{-*} &= A^{-1}\cdot\mathcal{E}_S(\mathbf{c}\cdot\mathbf{c}^*)\cdot A^{-*}\\
&= A^{-1}\cdot\left(\sum_{l=0}^{m} w_l\cdot\mathbf{c}(\varphi_l)\cdot\mathbf{c}^*(\varphi_l)\right)\cdot A^{-*}\\
&= \sum_{l=0}^{m} w_l\cdot\left(A^{-1}\cdot\mathbf{c}(\varphi_l)\right)\cdot\left(A^{-1}\cdot\mathbf{c}(\varphi_l)\right)^*\\
&= \sum_{l=0}^{m} w_l\cdot e_l\cdot e_l^* = \operatorname{diag}(w_0,\ldots,w_m)
\end{aligned}$$

Then the inverse matrix $A^*\cdot C^{-1}(c)\cdot A$ is still a diagonal matrix and thus for all $l\in\{1,\ldots,m\}$

$$\mathbf{c}^*(\varphi_l)\cdot C^{-1}(c)\cdot\mathbf{c}(\varphi_0) = (A^*\cdot C^{-1}(c)\cdot A)_{l,0} = 0.$$

For the weights we observe

$$\mathbf{c}^*(\varphi_l)\cdot C^{-1}(c)\cdot\mathbf{c}(\varphi_l) = (A^*\cdot C^{-1}(c)\cdot A)_{l,l} = w_l^{-1}.$$

□

This proposition immediately leads to Algorithm B.2. From the proposition it is clear that the algorithm provides the correct result if it terminates. The fact that it terminates follows from the existence of the output distribution [Kreĭn and Nudel'man 1977, p. 149].

For Hamburger moment shadow mapping, the analog to Algorithm B.2 already solves Problem 3.1 by the Markov-Kreĭn Theorem 4.1. This is however not the case for the trigonometric moment problem. The Markov-Kreĭn theorem does not apply and optimal solutions to Problem 3.1 do indeed have a more complicated structure. Based on observations on the output of Algorithm 3.1, we make the following conjecture.

**Conjecture B.7.** *Let $c\in\mathbb{C}^{m+1}$ such that the Toeplitz matrix $C(c)$ is positive definite. Then a phase distribution $S$ on $(-\pi,\,\pi]$ minimizing $S((-\pi,\,\varphi_f))$ may be chosen to have support at $\varphi_f$, $\pi$ and at most $m$ additional points in $(-\pi,\,\pi]$. The support at either $\varphi_f$ or $\pi$ may vanish.*

**Algorithm B.2** Reconstruction of a phase distribution with $m+1$ points of support matching $m$ trigonometric moments.
**Input:** Trigonometric moments $c \in \mathbb{C}^{m+1}$ such that $C(c)$ is positive definite and a prescribed point of support $\varphi_f := \pi \cdot z_f \in (-\pi,\, \pi]$.
**Output:** A distribution $S$ on $(-\pi,\, \pi]$ with $\mathcal{E}_S(\mathbf{c}) = c$ and support at $\varphi_f$.

```
1. Solve $C(c) \cdot q = \mathbf{c}(\varphi_f)$ for $q \in \mathbb{C}^{m+1}$.
2. Solve $\sum_{j=0}^{m} \overline{q_j} \cdot x^j = 0$ for $x$ and denote the distinct solutions by $x_1, \ldots, x_m \in \mathbb{C}$.
3. For all $l \in \{1, \ldots, m\}$ set $\varphi_l := \arg x_l \in (-\pi,\, \pi]$, i.e.
       $|x_l| \cdot \exp(i \cdot \varphi_l) = x_l.$
4. Set $A := (\mathbf{c}(\varphi_f),\, \mathbf{c}(\varphi_1),\, \ldots,\, \mathbf{c}(\varphi_m)) \in \mathbb{C}^{(m+1)\times(m+1)}$.
5. Solve $A \cdot w = c$ for $w \in \mathbb{R}^{m+1}$.
6. Return $\sum_{l=0}^{m} w_l \cdot \delta_{\varphi_l}$.
```

For Hamburger moment shadow mapping solutions have a similar structure but without support at $\pi$. Intuitively, this difference is plausible. The trigonometric moment-generating function $\mathbf{c}$ is periodic. There is no reason why the minimizing distribution should have support at one end of the interval $(-\pi,\, \varphi_f)$ but not at the other. And for minimization this support has to be placed at $\pi$ rather than $-\pi$, which does not make a difference for the trigonometric moments since $\mathbf{c}(-\pi) = \mathbf{c}(\pi)$.

Unfortunately, this different structure introduces one additional degree of freedom, namely the probability $w_3 := S(\{\pi\})$. Determining this degree of freedom is non-trivial and we have only found a closed-form solution for $m = 2$. In the following we let $S := \sum_{l=0}^{3} w_l \cdot \delta_{\varphi_l}$ denote an optimal solution with $w_0, w_1, w_2 \in [0,\, 1]$, $\varphi_1, \varphi_2 \in (-\pi,\, \pi]$ with $\varphi_1 < \varphi_2$, $\varphi_0 := \varphi_f := \pi \cdot z_f \in [-\pi,\, \pi]$ and $\varphi_3 := \pi$.

Suppose we know the optimal value for $w_3$. Then we can remove its contribution to the trigonometric moments by computing $c - w_3 \cdot \mathbf{c}(\varphi_3)$. The corresponding distribution $S - w_3 \cdot \delta_{\varphi_3}$ has support at no more than $m+1 = 3$ points of support. If there are two or fewer points of support, Algorithm 8.2 serves to reconstruct it. Otherwise, $\varphi_f$ has to be one of the points of support and Algorithm B.2 is applicable.

The challenge lies in determining the $w_3$ that minimizes

$$S((-\pi,\,\varphi_f)) = \begin{cases} 0 & \text{if } \varphi_f \leq \varphi_1, \\ w_1 & \text{if } \varphi_1 < \varphi_f \leq \varphi_2, \\ w_1 + w_2 & \text{if } \varphi_2 < \varphi_f. \end{cases} \tag{B.5}$$

The case $\varphi_f \leq \varphi_1$ is trivial. In the case $\varphi_2 < \varphi_f$ we need to minimize $w_1 + w_2$. By Proposition B.6 we can write this as

$$w_1 + w_2 = 1 - w_0 - w_3 = 1 - \frac{1}{\mathbf{c}^*(\varphi_f) \cdot C^{-1}(c - w_3 \cdot \mathbf{c}(\varphi_3)) \cdot \mathbf{c}(\varphi_f)} - w_3. \tag{B.6}$$

The following Lemma allows us to make the dependence on $w_3$ more explicit.

**Lemma B.8.** *Let $c \in \mathbb{C}^{m+1}$, $w_3 \in [0,\,1]$ and $\varphi_3 \in (-\pi,\,\pi]$ such that $C(c)$ and $C(c - w_3 \cdot \mathbf{c}(\varphi_3))$ are both invertible. Then*

$$C^{-1}(c - w_3 \cdot \mathbf{c}(\varphi_3)) = C^{-1}(c) + \frac{w_3 \cdot C^{-1}(c) \cdot \mathbf{c}(\varphi_3) \cdot \mathbf{c}^*(\varphi_3) \cdot C^{-1}(c)}{1 - w_3 \cdot \mathbf{c}^*(\varphi_3) \cdot C^{-1}(c) \cdot \mathbf{c}(\varphi_3)}.$$

*Proof sketch.* The proof can be completed by multiplying the Toeplitz matrix $C(c) - w_3 \cdot \mathbf{c}(\varphi_3) \cdot \mathbf{c}^*(\varphi_3)$ by its claimed inverse. Expanding the product, the first term is the identity matrix $I$. The other three terms all contain the matrix factor $w_3 \cdot \mathbf{c}(\varphi_3) \cdot \mathbf{c}^*(\varphi_3) \cdot C^{-1}(c)$. After factoring it out, it is evident that the scalar prefactor of this matrix evaluates to zero. □

For the sake of more compact expressions, we introduce the function

$$Q(\varphi, \psi) := \mathbf{c}^*(\varphi) \cdot C^{-1}(c) \cdot \mathbf{c}(\psi).$$

Applying Lemma B.8 to Equation (B.6), we obtain

$$w_1 + w_2 = 1 - \frac{1}{Q(\varphi_f, \varphi_f) + w_3 \cdot \frac{|Q(\varphi_3, \varphi_f)|^2}{1 - w_3 \cdot Q(\varphi_3, \varphi_3)}} - w_3.$$

The dependence on $w_3$ is now explicit enough to compute critical points. We take the derivative with respect to $w_3$ and compute two roots for $w_3$:

$$w_3 = \frac{\pm|Q(\varphi_3, \varphi_f)| - Q(\varphi_f, \varphi_f)}{|Q(\varphi_3, \varphi_f)|^2 - Q(\varphi_f, \varphi_f) \cdot Q(\varphi_3, \varphi_3)}$$

Inserting + for ± minimizes $w_1 + w_2$ whereas − maximizes this probability.

The most involved case is the one where $\varphi_1 < \varphi_f \leq \varphi_2$ and we need to minimize $w_1$. To this end, we ask for the value of $w_1$ as function of the unknown $\varphi_1$. We note that $S - w_1 \cdot \delta_{\varphi_1}$ is a distribution with at most three points of support representing $c - w_1 \cdot \mathbf{c}(\varphi_1)$. Assuming a positive-definite Toeplitz matrix, Proposition B.6 implies

$$\mathbf{c}^*(\varphi_f) \cdot C^{-1}(c - w_1 \cdot \mathbf{c}(\varphi_1)) \cdot \mathbf{c}(\varphi_3) = 0.$$

By Lemma B.8, we can rewrite this as

$$Q(\varphi_f, \varphi_3) + \frac{w_1 \cdot Q(\varphi_f, \varphi_1) \cdot Q(\varphi_1, \varphi_3)}{1 - w_1 \cdot Q(\varphi_1, \varphi_1)} = 0.$$

Solving for $w_1$ yields

$$w_1 = \frac{Q(\varphi_f, \varphi_3)}{Q(\varphi_f, \varphi_3) \cdot Q(\varphi_1, \varphi_1) - Q(\varphi_f, \varphi_1) \cdot Q(\varphi_1, \varphi_3)}. \tag{B.7}$$

The numerator in Equation (B.7) is independent of $\varphi_1$. To find extrema with respect to $\varphi_1$, it suffices to find extrema of the denominator. This denominator is a linear combination of $\exp(-2 \cdot i \cdot \varphi_1), \ldots, \exp(2 \cdot i \cdot \varphi_1)$. Substituting $x := \exp(i \cdot \varphi_1)$ and multiplying by $x^2 \neq 0$, this becomes a quartic polynomial. Its roots can be computed in closed form albeit at considerable cost. Having all critical points, the $\varphi_1$ minimizing $w_1$ can be found through Equation (B.7) and Algorithm B.2 serves to complete $S$.

Thus, we now have a closed-form expression for the minimizing $S$ in all cases. However, unless we want to try each case, we need a means of determining which case is present. Experiments with Algorithm 3.1 showed that there is a very simple way to do this. Running Algorithm B.2 with input $c$ and $\varphi_f = \pi$, we obtain a distribution with support at points $\varphi_1$, $\varphi_2$ and $\pi$. Using these phases for $\varphi_1$, $\varphi_2$ in Equation (B.5) correctly distinguishes the various cases.

### B.4.2 Implementation

Implementing the algorithm derived above is cumbersome. Rather than discussing the details here, we note that an HLSL implementation has been published in the supplementary material of the original paper [Peters and Klein 2015]. The only change made to this code since then, is that the current implementation uses a Cholesky-decomposition whenever it needs to invert a Toeplitz matrix. Doing so considerably improves robustness.

Trigonometric moments use the available range of values extensively and we do not require an optimized quantization transform as for moment shadow mapping (Section 4.1.4). However, we do apply biasing. The natural choice for the vector of biasing moments is $e_0 = (1, 0, 0)^\mathsf{T}$ because this is the only vector that is invariant under cyclic shifts of the depth-domain. For 64-bit trigonometric moment shadow mapping we use $\alpha_c := 6 \cdot 10^{-5}$ and at 128 bits we use $\alpha_c = 9 \cdot 10^{-7}$. The biased vector of trigonometric moments is defined by

$$c' := (1 - \alpha_c) \cdot c + \alpha_c \cdot e_0.$$

## B.5 Maximum Entropy Spectral Estimate

The maximum entropy spectral estimate described in Theorem 8.2 is at the core of our work on transient imaging. It was first introduced by Burg [1975]. Here we provide a correctness proof in a manner that is consistent with other proofs in our work. The underlying ideas go back to the work of Burg [1975] and Landau [1987].

Before we can prove the theorem itself, we need two lemmata. The first one helps us to prove that the reconstructed density $D$ is well-defined and aids in the computation of its trigonometric moments. It is concerned with the poles shown in Figure 8.1d.

**Lemma B.9.** *Let $c \in \mathbb{C}^{m+1}$ such that $C(c)$ is a positive-definite Toeplitz matrix. Let $\mathbf{b} : \mathbb{C} \to \mathbb{C}^{m+1}$ with $\mathbf{b}_j(x) := x^j$. Then all roots of the polynomial $e_0^* \cdot C^{-1}(c) \cdot \mathbf{b}$ lie outside the unit circle, i.e. they have magnitude greater one.*

*Proof.* This proof is analogous to the proof of Landau [1987, Proposition 1, p. 51 f.]. Let $x_0 \in \mathbb{C}$ be a root, i.e. $e_0^* \cdot C^{-1}(c) \cdot \mathbf{b}(x_0) = 0$. Let $p \in \mathbb{C}^m$ hold the conjugate coefficients of the polynomial resulting from division of $e_0^* \cdot C^{-1}(c) \cdot \mathbf{b}(x)$ by the linear factor $(x - x_0)$, i.e.:

$$\begin{aligned}
&\forall x \in \mathbb{C} : \; e_0^* \cdot C^{-1}(c) \cdot \mathbf{b}(x) = (x - x_0) \cdot \begin{pmatrix} p \\ 0 \end{pmatrix}^* \cdot \mathbf{b}(x) \\
\Leftrightarrow\; &\forall x \in \mathbb{C} : \; e_0^* \cdot C^{-1}(c) \cdot \mathbf{b}(x) = \begin{pmatrix} 0 \\ p \end{pmatrix}^* \cdot \mathbf{b}(x) - x_0 \cdot \begin{pmatrix} p \\ 0 \end{pmatrix}^* \cdot \mathbf{b}(x) \\
\Leftrightarrow\; &C^{-1}(c) \cdot e_0 = \begin{pmatrix} 0 \\ p \end{pmatrix} - \overline{x_0} \cdot \begin{pmatrix} p \\ 0 \end{pmatrix} \\
\Leftrightarrow\; &\overline{x_0} \cdot \begin{pmatrix} p \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ p \end{pmatrix} - C^{-1}(c) \cdot e_0 \qquad \text{(B.8)}
\end{aligned}$$

Now we consider the dot product and norm induced by the Hermitian, positive-definite matrix $C(c)$:

$$\begin{aligned} \langle\cdot,\cdot\rangle_{C(c)} :\ & \mathbb{C}^{m+1}\times\mathbb{C}^{m+1}\to\mathbb{C} & \qquad \|\cdot\|_{C(c)} :\ & \mathbb{C}^{m+1}\to\mathbb{R} \\ & u,v\mapsto u^*\cdot C(c)\cdot v & & u\mapsto\sqrt{u^*\cdot C(c)\cdot u} \end{aligned}$$

With respect to this dot product, the vectors $\begin{pmatrix}0\\p\end{pmatrix}$ and $C^{-1}(c)\cdot e_0$ are orthogonal:

$$\left\langle\begin{pmatrix}0\\p\end{pmatrix},\,C^{-1}(c)\cdot e_0\right\rangle_{C(c)} = \begin{pmatrix}0\\p\end{pmatrix}^*\cdot C(c)\cdot C^{-1}(c)\cdot e_0 = \begin{pmatrix}0\\p\end{pmatrix}^*\cdot e_0 = 0 \quad \text{(B.9)}$$

Furthermore, $\begin{pmatrix}0\\p\end{pmatrix}$ and $\begin{pmatrix}p\\0\end{pmatrix}$ have the same norm due to the special structure of the Toeplitz matrix $C(c)$:

$$\begin{aligned} \left\|\begin{pmatrix}0\\p\end{pmatrix}\right\|^2_{C(c)} &= \sum_{j,k=1}^{m}\overline{p_{j-1}}\cdot C_{j,k}(c)\cdot p_{k-1} = \sum_{j,k=0}^{m-1}\overline{p_j}\cdot C_{j+1,k+1}(c)\cdot p_k \\ &= \sum_{j,k=0}^{m-1}\overline{p_j}\cdot C_{j,k}(c)\cdot p_k = \left\|\begin{pmatrix}p\\0\end{pmatrix}\right\|^2_{C(c)} \end{aligned} \quad \text{(B.10)}$$

To complete the proof, we apply the norm on both sides of Equation (B.8):

$$\begin{aligned} \left\|\overline{x_0}\cdot\begin{pmatrix}p\\0\end{pmatrix}\right\|^2_{C(c)} &= \left\|\begin{pmatrix}0\\p\end{pmatrix} - C^{-1}(c)\cdot e_0\right\|^2_{C(c)} \\ \overset{\text{(B.9)}}{\Leftrightarrow} |x_0|^2\cdot\left\|\begin{pmatrix}p\\0\end{pmatrix}\right\|^2_{C(c)} &= \left\|\begin{pmatrix}0\\p\end{pmatrix}\right\|^2_{C(c)} + \left\|C^{-1}(c)\cdot e_0\right\|^2_{C(c)} \\ \overset{\text{(B.10)}}{\Leftrightarrow} |x_0|^2 = 1 + &\frac{\|C^{-1}(c)\cdot e_0\|^2_{C(c)}}{\left\|\begin{pmatrix}p\\0\end{pmatrix}\right\|^2_{C(c)}} > 1 \end{aligned}$$

□

The next Lemma is concerned with the correctness of Algorithm B.3 which implements the inverse of the map implemented by Levinson's Algorithm. This is potentially useful by itself but the main reason why we care about it is that it establishes that the relation between $c$ and $q := C^{-1}(c)\cdot e_0$ is bijective.

**Algorithm B.3** Inverse of Levinson's Algorithm 8.1.
**Input:** $q := C^{-1}(c) \cdot e_0 \in \mathbb{C}^{m+1}$ where $C(c)$ is a Hermitian, positive-definite Toeplitz matrix.
**Output:** $c = C(c) \cdot e_0 \in \mathbb{C}^{m+1}$.

```
1. $L := 0 \in \mathbb{C}^{(m+1)\times(m+1)}$
2. $q^{(m)} := q \in \mathbb{C}^{m+1}$
3. For $l \in \{m, m-1, \ldots, 0\}$:
    a) $p^{(l)} := (\overline{q_l^{(l)}}, \ldots, \overline{q_0^{(l)}})^\mathsf{T} \in \mathbb{C}^{l+1}$
    b) For $j \in \{0, \ldots, l\}$:
        i. $L_{l,j} := \overline{p_j^{(l)}}$
    c) $q^{(l-1)} := (q_0^{(l)}, \ldots, q_{l-1}^{(l)})^\mathsf{T} - \frac{q_l^{(l)}}{p_l^{(l)}} \cdot (p_0^{(l)}, \ldots, p_{l-1}^{(l)})^\mathsf{T} \in \mathbb{C}^l$
4. Compute $c := L^{-1} \cdot e_0$ by forward substitution.
5. Return $c$.
```

**Lemma B.10.** *Algorithm B.3 is correct and terminates in time $O(m^2)$.*

*Proof.* This proof follows the ideas of Landau [1987, Proposition 3, p. 53]. The run time of $O(m^2)$ can be seen directly from the structure of the algorithm. To prove correctness, we consider main minors of the Toeplitz matrix. For all $l \in \{0, \ldots, m\}$ let

$$C^{(l)}(c) := (C_{j,k}(c))_{j,k=0}^{l} \in \mathbb{C}^{(l+1)\times(l+1)},$$

i.e. $C^{(l)}(c)$ is the top-left part of $C(c)$. We will prove for all $l \in \{0, \ldots, m\}$ that:

1. If $C^{(l)}(c) \cdot q^{(l)} = e_0 \in \mathbb{C}^{l+1}$, then $C^{(l)}(c) \cdot p^{(l)} = e_l \in \mathbb{C}^{l+1}$,
2. $C^{(l)}(c) \cdot q^{(l)} = e_0 \in \mathbb{C}^{l+1}$,
3. $L \cdot C(c) \cdot e_0 = e_0$ at the end of the algorithm.

Step 1: Suppose that $C^{(l)}(c) \cdot q^{(l)} = e_0$ and consider the permutation matrix

$$R := \begin{pmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \iddots & \iddots & 0 \\ 0 & 1 & \iddots & \vdots \\ 1 & 0 & \cdots & 0 \end{pmatrix} \in \mathbb{C}^{(l+1)+(l+1)}.$$

When multiplied from the left, this matrix reverts the order of rows, when multiplied from the right, it reverts the order of columns. In some sense, it commutes with Hermitian Toeplitz matrices because for all $j, k \in \{0, \ldots, l\}$ we know $j - (l - k) = k - (l - j)$ and thus

$$(C^{(l)}(c) \cdot R)_{j,k} = C_{j,l-k}(c) = C_{k,l-j}(c) = \overline{C_{l-j,k}(c)} = \overline{(R \cdot C^{(l)}(c))}_{j,k}.$$

It follows that

$$C^{(l)}(c) \cdot p^{(l)} = C^{(l)}(c) \cdot R \cdot \overline{q^{(l)}} = \overline{R \cdot C^{(l)}(c) \cdot q^{(l)}} = \overline{R \cdot e_0} = e_l.$$

Step 2: We proceed by induction over $l$.

*Base case, $l = m$:* By definition of the input $C^{(l)}(c) \cdot q^{(l)} = e_0$.

*Induction hypothesis:* $C^{(l)}(c) \cdot q^{(l)} = e_0$.

*Induction step, $l \to l - 1$:* We know $C^{(l)}(c) \cdot q^{(l)} = e_0$ and by Step 1 also $C^{(l)}(c) \cdot p^{(l)} = e_l$. The division by $p_l^{(l)}$ is well-defined because $(C^{(l)}(c))^{-1}$ is positive definite and thus

$$p_l^{(l)} = e_l^* \cdot p^{(l)} = e_l^* \cdot (C^{(l)}(c))^{-1} \cdot e_l > 0. \tag{B.11}$$

We observe that the last entry of $q^{(l)} - \frac{q_l^{(l)}}{p_l^{(l)}} \cdot p^{(l)}$ is zero by construction whereas the other entries are stored in $q^{(l-1)}$. It follows that for all $j \in \{0, \ldots, l-1\}$

$$\begin{aligned} e_j^* \cdot C^{(l-1)}(c) \cdot q^{(l-1)} &= e_j^* \cdot C^{(l)}(c) \cdot \left( q^{(l)} - \frac{q_l^{(l)}}{p_l^{(l)}} \cdot p^{(l)} \right) \\ &= e_j^* \cdot \left( e_0 - \frac{q_l^{(l)}}{p_l^{(l)}} \cdot e_l \right) = e_j^* \cdot e_0. \end{aligned}$$

Hence, $C^{(l-1)}(c) \cdot q^{(l-1)} = e_0$.

Step 3: The matrix $L$ is lower triangular by construction and has non-zero diagonal entries by Equation (B.11). Thus, forward substitution is applicable. Now for all $l \in \{0, \dots, m\}$ we consider

$$e_l^* \cdot L \cdot c = \begin{pmatrix} p^{(l)} \\ 0 \end{pmatrix}^* \cdot c = ((C^{(l)}(c))^{-1} \cdot e_l)^* \cdot C^{(l)}(c) \cdot e_0 = e_l^* \cdot e_0.$$

We conclude that $L \cdot C^{(l)}(c) \cdot e_0 = e_0$ and therefore $L^{-1} \cdot e_0 = C^{(l)}(c) \cdot e_0$ is the correct output. □

Having proven these lemmata, we are now ready to prove correctness of the maximum entropy spectral estimate. We repeat the corresponding theorem here.

**Theorem 8.2** (Maximum entropy spectral estimate [Burg 1975]). *We recall from Definition 2.10 on page 26 that*

$$C(c) = (c_{j-k})_{j,k=0}^m = \begin{pmatrix} c_0 & \overline{c_1} & \cdots & \overline{c_m} \\ c_1 & c_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \overline{c_1} \\ c_m & \cdots & c_1 & c_0 \end{pmatrix} \in \mathbb{C}^{(m+1)\times(m+1)}$$

*denotes the Toeplitz matrix. Suppose that $C(c)$ is positive definite. For all $\varphi \in (0,\, 2 \cdot \pi]$ let*

$$D(\varphi) := \frac{1}{2 \cdot \pi} \cdot \frac{e_0^* \cdot C^{-1}(c) \cdot e_0}{|e_0^* \cdot C^{-1}(c) \cdot \mathbf{c}(\varphi)|^2} \in \mathbb{R} \tag{8.3}$$

*where $e_0 := (1, 0, \dots, 0)^\mathsf{T} \in \mathbb{C}^{m+1}$. Then $D(\varphi)$ is positive and the measure $F_D$ with density $D$ fulfills the moment constraints $c = \int \mathbf{c}(\varphi)\, \mathrm{d}F_D(\varphi)$. Among all such measures it has minimal Burg entropy $\mathcal{H}_{Burg}(F_D)$.*

*Proof.* Since $C^{-1}(c)$ is positive definite, $e_0^* \cdot C^{-1}(c) \cdot e_0 > 0$. Furthermore, the denominator is non-zero by Lemma B.9 because

$$e_0^* \cdot C^{-1}(c) \cdot \mathbf{c}(\varphi) = e_0^* \cdot C^{-1}(c) \cdot \mathbf{b}(\exp(i \cdot \varphi)) \neq 0.$$

Therefore, $D$ is well-defined and positive. We complete the remainder of the proof in the following steps:

1. Use Lemma B.9 to prove
$$\int \mathbf{c}(\varphi) \cdot \mathbf{c}^*(\varphi) \, \mathrm{d}F_D(\varphi) \cdot C^{-1}(c) \cdot e_0 = e_0, \tag{B.12}$$
2. Use Lemma B.10 to conclude that $\int \mathbf{c}(\varphi) \, \mathrm{d}F_D(\varphi) = c$,
3. Prove that the Burg entropy is minimal.

Step 1 (using the approach of Burg [1975, p. 9 ff.]): We consider entry $j \in \{0, \ldots, m\}$ of the vector in Equation (B.12) individually:

$$\begin{aligned}
&e_j^* \cdot \int \mathbf{c}(\varphi) \cdot \mathbf{c}^*(\varphi) \, \mathrm{d}F_D(\varphi) \cdot C^{-1}(c) \cdot e_0 \\
=&\int_0^{2\cdot\pi} D(\varphi) \cdot (e_j^* \cdot \mathbf{c}(\varphi)) \cdot (\mathbf{c}^*(\varphi) \cdot C^{-1}(c) \cdot e_0) \, \mathrm{d}\varphi \\
=&\int_0^{2\cdot\pi} \frac{1}{2 \cdot \pi} \cdot \frac{e_0^* \cdot C^{-1}(c) \cdot e_0}{e_0^* \cdot C^{-1}(c) \cdot \mathbf{c}(\varphi)} \cdot (e_j^* \cdot \mathbf{c}(\varphi)) \, \mathrm{d}\varphi \\
=&\frac{1}{2 \cdot \pi} \cdot \int_{|x|=1} \frac{e_0^* \cdot C^{-1}(c) \cdot e_0}{e_0^* \cdot C^{-1}(c) \cdot \mathbf{b}(x)} \cdot x^j \, \mathrm{d}x \\
=&\frac{1}{2 \cdot \pi \cdot i} \cdot \oint_{|x|=1} \frac{e_0^* \cdot C^{-1}(c) \cdot e_0}{e_0^* \cdot C^{-1}(c) \cdot \mathbf{b}(x)} \cdot x^{j-1} \, \mathrm{d}x
\end{aligned}$$

Note that $\oint_{|x|=1}$ denotes a counter-clockwise contour integral over the boundary of the unit circle. Such an integral includes the derivative of the arc-length parametrization of the contour as factor. In the present case this is $\frac{\mathrm{d}}{\mathrm{d}\varphi} \exp(i \cdot \varphi) = i \cdot \exp(i \cdot \varphi)$ which is why we divide the integrand by $i \cdot x$.

For $j \geq 1$ the integrand is a holomorphic function within the unit circle because by Lemma B.9 the denominator has no roots within the unit circle. By Cauchy's integral theorem such an integral evaluates to zero. For the case $j = 0$ we employ Cauchy's integral formula:

$$\begin{aligned}
&\frac{1}{2 \cdot \pi \cdot i} \cdot \oint_{|x|=1} \frac{e_0^* \cdot C^{-1}(c) \cdot e_0}{e_0^* \cdot C^{-1}(c) \cdot \mathbf{b}(x)} \cdot \frac{1}{x} \, \mathrm{d}x \\
=&\frac{e_0^* \cdot C^{-1}(c) \cdot e_0}{e_0^* \cdot C^{-1}(c) \cdot \mathbf{b}(0)} = \frac{e_0^* \cdot C^{-1}(c) \cdot e_0}{e_0^* \cdot C^{-1}(c) \cdot e_0} = 1
\end{aligned}$$

Thus, Equation (B.12) holds.

Step 2 (using the approach of Landau [1987, p. 55]): We recall from Proposition 2.11 that

$$C' := \int \mathbf{c}(\varphi) \cdot \mathbf{c}^*(\varphi) \,\mathrm{d}F_D(\varphi) \in \mathbb{C}^{(m+1)\times(m+1)}$$

is a positive semi-definite Toeplitz matrix. By Proposition 2.13 it is even positive definite because $D$ is a positive density. According to Equation (B.12)

$$C' \cdot C^{-1}(c) \cdot e_0 = e_0$$

and hence

$$C^{-1}(c) \cdot e_0 = C'^{-1} \cdot e_0.$$

By Lemma B.10 this uniquely determines $c = C(c) \cdot e_0 = C' \cdot e_0$. We conclude

$$c = \int \mathbf{c}(\varphi) \cdot \mathbf{c}^*(\varphi) \cdot e_0 \,\mathrm{d}F_D(\varphi) = \int \mathbf{c}(\varphi) \,\mathrm{d}F_D(\varphi).$$

Step 3: To prove the optimality of $D$, let $E : (0, 2 \cdot \pi] \to \mathbb{R}$ be another non-negative density function with well-defined Burg entropy $\mathcal{H}_{\mathrm{Burg}}(E) < \infty$ and $\int_0^{2\cdot\pi} E(\varphi) \cdot \mathbf{c}(\varphi) \,\mathrm{d}\varphi = c$. Using Jensen's inequality we obtain:

$$\begin{aligned}
&\frac{\mathcal{H}_{\mathrm{Burg}}(D) - \mathcal{H}_{\mathrm{Burg}}(E)}{2 \cdot \pi} \\
&= \frac{1}{2 \cdot \pi} \cdot \left( \int_0^{2\cdot\pi} \log E(\varphi) \,\mathrm{d}\varphi - \int_0^{2\cdot\pi} \log D(\varphi) \,\mathrm{d}\varphi \right) \\
&= \frac{1}{2 \cdot \pi} \cdot \int_0^{2\cdot\pi} \log \frac{E(\varphi)}{D(\varphi)} \,\mathrm{d}\varphi \\
&\leq \log \frac{1}{2 \cdot \pi} \cdot \int_0^{2\cdot\pi} \frac{E(\varphi)}{D(\varphi)} \,\mathrm{d}\varphi \\
&= \log \int_0^{2\cdot\pi} E(\varphi) \cdot \frac{|e_0^* \cdot C^{-1}(c) \cdot \mathbf{c}(\varphi)|^2}{e_0^* \cdot C^{-1}(c) \cdot e_0} \,\mathrm{d}\varphi \\
&= \log e_0^* \cdot C^{-1}(c) \cdot \int_0^{2\cdot\pi} E(\varphi) \cdot \frac{\mathbf{c}(\varphi) \cdot \mathbf{c}^*(\varphi)}{e_0^* \cdot C^{-1}(c) \cdot e_0} \,\mathrm{d}\varphi \cdot C^{-1}(c) \cdot e_0 \\
&= \log \frac{e_0^* \cdot C^{-1}(c) \cdot C(c) \cdot C^{-1}(c) \cdot e_0}{e_0^* \cdot C^{-1}(c) \cdot e_0} = \log 1 = 0
\end{aligned}$$

Thus, $\mathcal{H}_{\mathrm{Burg}}(D) \leq \mathcal{H}_{\mathrm{Burg}}(E)$ and therefore $D$ must have globally minimal Burg entropy. □

## B.6 Pisarenko Estimate

We use the Pisarenko estimate (Algorithm 8.2) as alternative to the maximum entropy spectral estimate in transient imaging. Its use is appropriate whenever the ground truth is known to be sparse and it is the limit of the maximum entropy spectral estimate for this case. At the same time it is the analog of Algorithm 2.1 for trigonometric moments.

The correctness of the Pisarenko estimate is a direct consequence of Proposition 2.13 which we now repeat and prove.

**Proposition 2.13** (Boundary case for trigonometric moment problems [Kreĭn and Nudel'man 1977, p. 65, 78]). *Let $c \in \mathbb{C}^{m+1}$ such that $C(c)$ is positive semi-definite. The following statements are equivalent:*

1. *$C(c)$ is singular,*
2. *There exists exactly one measure $M$ on $(0,\, 2\cdot\pi]$ with $c = \int \mathbf{c}(x)\,\mathrm{d}M(x)$,*
3. *There exists $x_0, \ldots, x_{m-1} \in (0,\, 2\cdot\pi]$ and $w_0, \ldots, w_{m-1} > 0$ such that $M := \sum_{l=0}^{m-1} w_l \cdot \delta_{x_l}$ yields $c = \int \mathbf{c}(x)\,\mathrm{d}M(x)$.*

*Suppose $C(c)$ is singular and let $q \in \ker C(c)$ with $q \neq 0$. Then $x_0, \ldots, x_{m-1}$ are solutions of the equation $q^* \cdot \mathbf{c}(x) = 0$.*

*Proof.* "1. ⇒ 3. and 2." Let $q \in \ker C(c)$ with $q \neq 0$.

By Proposition 2.11 there exists a measure $M$ with $c = \int \mathbf{c}(x)\,\mathrm{d}M(x)$ and we can use it to represent the Toeplitz matrix $C(c)$:

$$0 = q^* \cdot C(c) \cdot q = \int q^* \cdot \mathbf{c}(x) \cdot \mathbf{c}^*(x) \cdot q \,\mathrm{d}M(x) = \int \left|q^* \cdot \mathbf{c}(x)\right|^2 \,\mathrm{d}M(x) \quad \text{(B.13)}$$

The integrand $|q^* \cdot \mathbf{c}(x)|^2$ is non-negative. Furthermore, we can employ the substitution $y := \exp(i \cdot x)$ to obtain

$$q^* \cdot \mathbf{c}(x) = \sum_{j=0}^{m} \overline{q_j} \cdot \exp(j \cdot i \cdot x) = \sum_{j=0}^{m} \overline{q_j} \cdot y^j.$$

Written like this, the integrand is the absolute square of a non-zero polynomial of degree $m$ or less and cannot have more than $m$ roots. Since the integral evaluates to zero, $M$ must have all of its support at these roots which proves 3. and the claim about the location of $x_0, \ldots, x_{m-1}$.

Since $\exp(i \cdot x)$ maps $(0,\, 2 \cdot \pi]$ onto the unit circle bijectively, Equation (B.13) uniquely determines the points of support $x_0, \ldots, x_{m-1} \in (0,\, 2 \cdot \pi]$. Suppose, the first $n \in \{0, \ldots, m\}$ of these points of support are distinct. Then the system of linear equations

$$\begin{pmatrix} 1 & \cdots & 1 \\ \exp(1 \cdot i \cdot x_0) & \cdots & \exp(1 \cdot i \cdot x_{n-1}) \\ \vdots & & \vdots \\ \exp((n-1) \cdot i \cdot x_0) & \cdots & \exp((n-1) \cdot i \cdot x_{n-1}) \end{pmatrix} \cdot \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{n-1} \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{pmatrix}$$

uniquely determines the corresponding weights $w_0, \ldots, w_{n-1}$ because the matrix in this system is a square Vandermonde matrix constructed from pairwise different values. Also, these weights have to be non-negative because otherwise this would contradict existence of $M$. Thus, we have proven 2..

"3. $\Rightarrow$ 1." Let $M = \sum_{l=0}^{m-1} w_l \cdot \delta_{x_l}$ such that $c = \int \mathbf{c}(x)\,\mathrm{d}M(x)$.

We note that the matrix $\mathbf{c}(x_l) \cdot \mathbf{c}^*(x_l)$ has rank one for all $l \in \{0, \ldots, m-1\}$. Thus, the rank of

$$C(c) = \int \mathbf{c}(x) \cdot \mathbf{c}^*(x)\,\mathrm{d}M(x) = \sum_{l=0}^{m-1} w_l \cdot \mathbf{c}(x_l) \cdot \mathbf{c}^*(x_l)$$

cannot be greater than $m$. It follows that $C(c) \in \mathbb{C}^{(m+1)\times(m+1)}$ is singular.

"$\neg$1. $\Rightarrow$ $\neg$2." Suppose $\det C(c)$ is positive definite.

Let $M$ be a measure on $(0,\, 2 \cdot \pi]$ with $c = \int \mathbf{c}(x)\,\mathrm{d}M(x)$. Let $x_0 \in (0,\, 2 \cdot \pi]$ such that $M(\{x_0\}) = 0$, i.e. $M$ does not have support at $x_0$. There exists an $\varepsilon > 0$ such that $C(c - \varepsilon \cdot \mathbf{c}(x_0))$ is still positive semi-definite. Let $N$ be a measure on $\mathbb{R}$ with $c - \varepsilon \cdot \mathbf{c}(x_0) = \int \mathbf{c}(x)\,\mathrm{d}N(x)$. Then

$$c = \int \mathbf{c}(x)\,\mathrm{d}M(x) = \int \mathbf{c}(x)\,\mathrm{d}N(x) + \varepsilon \cdot \mathbf{c}(x_0) = \int \mathbf{c}(x)\,\mathrm{d}(N + \varepsilon \cdot \delta_{x_0})(x)$$

and thus we have constructed two different measures representing the trigonometric moments $c$. □

Appendix C

# Implementation Details

## C.1 Hamburger Moment Shadow Mapping

In Section 4.1 we describe the implementation of Hamburger moment shadow mapping in detail. Listing C.1 provides a complete and tested implementation of Algorithm 4.2 in HLSL. Note that the implementation of the Cholesky decomposition is highly specialized to only compute non-trivial entries. Also note how the end result is computed by the same branch in all cases. A conditional provides the appropriate inputs.

Listing C.1: An HLSL implementation of Hamburger moment shadow mapping with four power moments. `b` provides four biased power moments, `Depth` provides the depth at which the shadow intensity is to be evaluated. The function returns the approximate shadow intensity.

```
float Hamburger4MSM(float4 b,float Depth){
   float3 z;
   z[0]=Depth;
   // Cholesky decomposition
   float L21D11=mad(-b[0],b[1],b[2]);
   float D11=mad(-b[0],b[0],b[1]);
   float SquaredDepthVariance=mad(-b[1],b[1],b[3]);
   float D22D11=dot(float2(SquaredDepthVariance,-L21D11),
                    float2(D11,                  L21D11));
   float InvD11=1.0f/D11;
   float L21=L21D11*InvD11;
   float D22=D22D11*InvD11;
   float InvD22=1.0f/D22;
   // Solution of a linear system with Cholesky
   float3 c=float3(1.0f,z[0],z[0]*z[0]);
   c[1]-=b.x;
   c[2]-=b.y+L21*c[1];
   c[1]*=InvD11;
   c[2]*=InvD22;
   c[1]-=L21*c[2];
   c[0]-=dot(c.yz,b.xy);
   // Quadratic equation solved with the quadratic formula
   float InvC2=1.0f/c[2];
   float p=c[1]*InvC2;
   float q=c[0]*InvC2;
   float D=(p*p*0.25f)-q;
   float r=sqrt(D);
   z[1]=-p*0.5f-r;
   z[2]=-p*0.5f+r;
   // Conditional computation of the shadow intensity
   float4 Switch=
      (z[2]<z[0])?float4(z[1],z[0],1.0f,1.0f):(
      (z[1]<z[0])?float4(z[0],z[1],0.0f,1.0f):
                  float4(0.0f,0.0f,0.0f,0.0f));
   float Quotient=(Switch[0]*z[2]-b[0]*(Switch[0]+z[2])+b[1])
                 /((z[2]-Switch[1])*(z[0]-z[1]));
   return saturate(Switch[2]+Switch[3]*Quotient);
}
```

## C.2 Hausdorff Moment Shadow Mapping

When implementing Hausdorff moment shadow mapping efficiently there is one non-trivial aspect that has not been discussed yet. We need to compute $w_0$ or $w_0 + w_1$ where

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & z_1 & z_f & 1 \\ 1 & z_1^2 & z_f^2 & 1 \\ -1 & z_1^3 & z_f^3 & 1 \end{pmatrix} \cdot \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} 1 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}.$$

Let $v_1 \in \{0, 1\}$ such that $w_0 + v_1 \cdot w_1$ is our intended end result. With the same approach as in Section 7.4.2 this can be turned into $\sum_{j=0}^{3} u_j \cdot b_j$ with

$$u := \begin{pmatrix} 1 & -1 & 1 & -1 \\ 1 & z_1 & z_1^2 & z_1^3 \\ 1 & z_f & z_f^2 & z_f^3 \\ 1 & 1 & 1 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 1 \\ v_1 \\ 0 \\ 0 \end{pmatrix}.$$

This time $u$ represents a polynomial $\sum_{j=0}^{3} u_j \cdot z^j$ taking the value one at $-1$, value $v_1$ at $z_1$ and having roots at $z_f$ and 1. Thus, it is the product of a linear polynomial with the linear factors $(z - z_f) \cdot (z - 1)$. We first construct this linear polynomial and then perform multiplication by the linear factors on the coefficients.

The complete implementation in HLSL is given in Listing C.2. Most of this code is identical to Listing C.1 but there is an extra branch implementing computation of $z_1$ and the procedure that we just described.

Listing C.2: An HLSL implementation of Hausdorff moment shadow mapping with four power moments. The signature is the same as for Hamburger moment shadow mapping.

```
float Hausdorff4MSM(float4 b,float Depth){
   float3 z;
   z[0]=Depth;
   // Cholesky decomposition
   float L21D11=mad(-b[0],b[1],b[2]);
   float D11=mad(-b[0],b[0],b[1]);
   float SquaredDepthVariance=mad(-b[1],b[1],b[3]);
   float D22D11=dot(float2(SquaredDepthVariance,-L21D11),
                    float2(D11,                  L21D11));
   float InvD11=1.0f/D11;
   float L21=L21D11*InvD11;
   float D22=D22D11*InvD11;
   float InvD22=1.0f/D22;
   // Solution of a linear system with Cholesky
   float3 c=float3(1.0f,z[0],z[0]*z[0]);
   c[1]-=b.x;
   c[2]-=b.y+L21*c[1];
   c[1]*=InvD11;
   c[2]*=InvD22;
   c[1]-=L21*c[2];
   c[0]-=dot(c.yz,b.xy);
   // Quadratic equation solved with the quadratic formula
   float InvC2=1.0f/c[2];
   float p=c[1]*InvC2;
   float q=c[0]*InvC2;
   float D=(p*p*0.25f)-q;
   float r=sqrt(D);
   z[1]=-p*0.5f-r;
   z[2]=-p*0.5f+r;
   // The solution uses four points of support
   [branch] if(z[1]<-1.0f || z[2]>1.0f){
      float zFree=((b[0]-b[2])*z[0]+b[3]-b[1])
                  /(z[0]+b[2]-b[0]-b[1]*z[0]);
      float w1Factor=(z[0]>zFree)?1.0f:0.0f;
      // Construct an interpolation polynomial
      float2 Normalizers=float2(
         w1Factor/((zFree-z[0])*mad(zFree,zFree,-1.0f)),
         0.5f/((zFree+1.0f)*(z[0]+1.0f)));
      float4 Polynomial;
      Polynomial[0]=mad(zFree,Normalizers.y,Normalizers.x);
      Polynomial[1]=Normalizers.x-Normalizers.y;
      // Multiply by (z-z[0])
      Polynomial[2]=Polynomial[1];
      Polynomial[1]=mad(Polynomial[1],-z[0],Polynomial[0]);
      Polynomial[0]*=-z[0];
      // Multiply by (z-1)
      Polynomial[3]=Polynomial[2];
      Polynomial.yz=Polynomial.xy-Polynomial.yz;
      Polynomial[0]*=-1.0f;
      // Evaluate the shadow intensity
      return saturate(dot(Polynomial,float4(1.0f,b.xyz)));
   }
   // The solution uses three points of support
   else{
      float4 Switch=
         (z[2]<z[0])?float4(z[1],z[0],1.0f,1.0f):(
         (z[1]<z[0])?float4(z[0],z[1],0.0f,1.0f):
                     float4(0.0f,0.0f,0.0f,0.0f));
      float Quotient=(Switch[0]*z[2]-b[0]*(Switch[0]+z[2])+b[1])
                    /((z[2]-Switch[1])*(z[0]-z[1]));
      return saturate(Switch[2]+Switch[3]*Quotient);
   }
}
```

## C.3 Prefiltered Single Scattering

For a detailed discussion of implementation details in prefiltered single scattering we refer to Klehm et al. [2014a]. However, in the following we discuss several details that are specific to our approach.

### C.3.1 Bounds for Rectified Coordinates

In Section 7.1.1 we state that we compute sharp bounds for $r$, $\varphi$ and $\theta$ such that the entire view frustum is covered. We provide details on this procedure in the following and note that a reference implementation has been published as supplementary material [Peters et al. 2016].

Single scattering should be accumulated over the entire view ray. Thus, we do not consider the near clipping plane. Let $q_0, q_1, q_2, q_3 \in \mathbb{R}^3$ be the coordinates of the four vertices of the far clipping plane of the camera used for main scene rendering in light view space. Without loss of generality let the camera position be in the origin of the coordinate system. Then the maximal value for $r$ is given by

$$r_{\max} := \max_{j \in \{0,\ldots,3\}} \sqrt{(q_j)_0^2 + (q_j)_1^2}.$$

Note that $(q_j)_k$ denotes the $k$-th entry of the vector $q_j$ for $k \in \{0, 1, 2\}$. Since we ignore the near plane, $r_{\min} := 0$.

To compute $\varphi_{\min}$ and $\varphi_{\max}$ we compute the maximal pairwise angle enclosed by the vectors $((q_j)_0, (q_j)_1)^\mathsf{T} \in \mathbb{R}^2$ for $j \in \{0, 1, 2, 3\}$. The azimuth of one of the two involved vectors provides $\varphi_{\min}$, the other provides $\varphi_{\max}$. A special case arises if the light direction or flipped light direction lies within the view frustum. This can be checked using the four side clipping planes. In both cases we have to set $\varphi_{\min} = 0$ and $\varphi_{\max} = 2 \cdot \pi$ to indicate that the boundary of the far clipping plane surrounds the camera position in the shadow map.

The computation of $\theta_{\min}$ and $\theta_{\max}$ is more intricate because the extremal inclination may be realized at the vertices, on the edges or within the area of the far clipping plane. It is convenient to exploit that inclinations depend monotonically on the $z$-coordinate of normalized vectors, i.e.

$$\theta_j := \arccos \frac{(q_j)_2}{\|q_j\|_2}.$$

Taking the minimal and maximal values of $\theta_0, \ldots, \theta_3$ yields the extrema at vertices. The inclination is extremal within the area of the far clipping plane if and only if the light direction or flipped light direction lie within the view frustum (see above). In this case an extremal inclination is $\theta_{\min} = 0$ or $\theta_{\max} = \pi$, respectively.

To find extrema on an edge of the far plane connecting corner points $j, k \in \{0, \ldots, 3\}$, we take the derivative of the $z$-coordinate of normalized vectors on the edge to find critical points:

$$\frac{\partial}{\partial t} \frac{(q_j + t \cdot (q_k - q_j))_2}{\|q_j + t \cdot (q_k - q_j)\|_2} = 0$$

This equation has the unique solution

$$t = \frac{(q_j)_2 \cdot q_j^\mathsf{T} \cdot (q_k - q_j) - (q_k - q_j)_2 \cdot \|q_j\|_2^2}{(q_k - q_j)_2 \cdot q_j^\mathsf{T} \cdot (q_k - q_j) - (q_j)_2 \cdot \|q_k - q_j\|_2^2}.$$

If $t \in [0, 1]$, we may need to adapt $[\theta_{\min}, \theta_{\max}]$ to include the inclination at this point on the ray.

Note that this whole algorithm only needs to be executed once per frame to get single scattering for one directional light.

## C.3.2 Transmittance-Weighted Prefix Sums

In terms of arithmetic, generation of transmittance-weighted prefix sums is an inexpensive operation. We expect the corresponding compute shader to be bandwidth limited. We benchmark our implementation on an NVIDIA GeForce GTX 970 with a bandwidth of 196 $\frac{\mathrm{GB}}{\mathrm{s}}$. Thus, computation of prefix sums over a $1024^2$ texture with 64 bits per texel should take

$$\frac{2 \cdot 1024^2 \cdot 8\,\mathrm{B}}{196\,\frac{\mathrm{GB}}{\mathrm{s}}} = 85.6\,\mu\mathrm{s}.$$

For a texture with 128 bits per texel we expect it to take $171.2\,\mu$s.

We use GPU timings obtained with NVIDIA Nsight[1] to compare this expectation to the actual run times. For the 128-bit textures used in moment

[1] developer.nvidia.com/nvidia-nsight-visual-studio-edition (retrieved on 1st of September 2016).

soft shadow mapping, the simple scheme using one thread per row or per column [Klehm et al. 2014a] is clearly bandwidth limited with a run time around 190 $\mu$s. However, the same approach is less efficient for textures with 64 bits per texel. Our initial implementation for 64-bit shadow maps with six moments took 620 $\mu$s.

Our optimized implementation can process a 64-bit moment shadow map with four moments in 110 $\mu$s which is close to the theoretical optimum of 85.6 $\mu$s. A 64-bit moment shadow map with six moments takes 180 $\mu$s so it is not quite optimal but still only twice as expensive as the theoretical optimum and we were unable to optimize it further.

This is accomplished using thread groups of $8 \times 8$ threads. For a texture of height $n_y \in \mathbb{N}$ we spawn $\frac{n_y}{8}$ such thread groups such that eight threads run per row. At any point in time each thread group operates on one $8 \times 8$ block in the texture, going through from left to right. Each thread will independently compute all prefix sums for its row. However, it will only write out the one prefix sum that corresponds to its location in the block.

This means that the overall amount of texture reads and additions is multiplied by a factor of eight but so is parallelism. Besides the writes to the output texture are coalesced. We tried caching the texture reads into thread-group-shared memory but the standard texture caches turned out to be more efficient. If the shadow map consists of multiple textures, they are processed in parallel by separate thread groups. Our HLSL implementation of this approach for a single four-channel texture is shown in Listing C.3.

Of course other GPUs may exhibit different behavior and we do not recommend using our code in production without performing benchmarks on the targeted hardware.

Listing C.3: A compute shader for HLSL shader profile cs_5_0 or higher to compute transmittance weighted prefix sums for a rectified moment shadow map with four channels. The shader reads from `RectifiedMomentShadowMap` and writes prefix sums to `OutPrefilteredMomentShadowMap`. Transmittance weighting uses the extinction coefficient `Extinction`. `DistanceMax` provides the maximal value for $r$. `InclinationMin` and `InclinationMax` provide the extremal values for $\theta$.

```
Texture2D<float4> RectifiedMomentShadowMap;
RWTexture2D<float4> OutPrefilteredMomentShadowMap;
cbuffer ConstantBuffer{
   float Extinction;
   float DistanceMax;
   float InclinationMin;
   float InclinationMax;
};
[numthreads(8,8,1)]
void Main(uint3 ThreadID:SV_DispatchThreadID){
   uint Width,Height;
   OutPrefilteredMomentShadowMap.GetDimensions(Width,Height);
   float TexelLength=DistanceMax
         /(sin(0.5f*(InclinationMin+InclinationMax))*float(Width));
   float TexelTransmittance=exp(-Extinction*TexelLength);
   uint DividedWidth=Width/8;
   float Weight=1.0f;
   float TotalWeight=0.0f;
   float4 PrefixSum=float4(0.0f,0.0f,0.0f,0.0f);
   [loop] for(uint x=0;x<DividedWidth;++x){
      uint BlockX=x*8;
      float4 StoredPrefixSum=PrefixSum;
      float StoredTotalWeight=TotalWeight;
      [unroll] for(uint i=0;i!=8;++i){
         PrefixSum+=Weight*RectifiedMomentShadowMap.Load
                           (uint3(BlockX+i,ThreadID.y,0));
         TotalWeight+=Weight;
         Weight*=TexelTransmittance;
         bool Store=(i==ThreadID.x);
         StoredPrefixSum=Store?PrefixSum:StoredPrefixSum;
         StoredTotalWeight=Store?TotalWeight:StoredTotalWeight;
      }
      uint2 iOutputTexel=uint2(BlockX+ThreadID.x,ThreadID.y);
      float4 Output=StoredPrefixSum/StoredTotalWeight;
      OutPrefilteredMomentShadowMap[iOutputTexel]=Output;
   }
}
```

### C.3.3 Hamburger Moment Shadow Mapping with Six Moments

The most challenging aspect about the implementation of six moment shadow mapping is the robust computation of cubic roots. Our solution based on the work of Blinn [2007] is shown in Listing C.4.

Listing C.4: An HLSL function for computing the roots of a cubic polynomial in closed form robustly. It assumes that the polynomial has three distinct, real roots. The polynomial is given by $\sum_{j=0}^{3}$`Coefficient[`$j$`]`$\cdot z^j$. The order of the returned roots is undefined.

```
float3 SolveCubic(float4 Coefficient){
   Coefficient.xyz/=Coefficient.w;
   Coefficient.yz/=3.0f;
   float3 Delta=float3(
      mad(-Coefficient.z,Coefficient.z,Coefficient.y),
      mad(-Coefficient.y,Coefficient.z,Coefficient.x),
      dot(float2(Coefficient.z,-Coefficient.y),Coefficient.xy));
   float Discriminant=dot(float2(4.0f*Delta.x,-Delta.y),Delta.zy);
   float2 Depressed=float2(
      mad(-2.0f*Coefficient.z,Delta.x,Delta.y),
      Delta.x);
   float Theta=atan2(sqrt(Discriminant),-Depressed.x)/3.0f;
   float2 CubicRoot;
   sincos(Theta,CubicRoot.y,CubicRoot.x);
   float3 Root=float3(
      CubicRoot.x,
      dot(float2(-0.5f,-0.5f*sqrt(3.0f)),CubicRoot),
      dot(float2(-0.5f, 0.5f*sqrt(3.0f)),CubicRoot));
   return mad(2.0f*sqrt(-Depressed.y),Root,-Coefficient.z);
}
```

Implementing the remainder of the algorithm robustly and efficiently requires the use of a Cholesky decomposition and Newton interpolation polynomials as described in Section 7.4. All of this is done in our HLSL implementation in Listing C.5.

Listing C.5: An implementation of Hamburger moment shadow mapping with six power moments in HLSL. `b` provides six biased moments, `Depth` provides the depth at which the shadow intensity is to be estimated. `Overestimation` blends linearly between guaranteed underestimation (zero) and guaranteed overestimation (one). The approximate shadow intensity is returned.

```
float Hamburger6MSM(float b[6],float Depth,float Overestimation){
   float4 z;
   z[0]=Depth;
   // Cholesky decomposition
   float InvD11=1.0f/mad(-b[0],b[0],b[1]);
   float L21D11=mad(-b[0],b[1],b[2]);
   float L21=L21D11*InvD11;
   float D22=mad(-L21D11,L21,mad(-b[1],b[1],b[3]));
   float L31D11=mad(-b[0],b[2],b[3]);
   float L31=L31D11*InvD11;
   float InvD22=1.0f/D22;
   float L32D22=mad(-L21D11,L31,mad(-b[1],b[2],b[4]));
   float L32=L32D22*InvD22;
   float D33=mad(-b[2],b[2],b[5])-dot(float2(L31D11,L32D22),
                                       float2(L31,   L32));
   float InvD33=1.0f/D33;
   // Solution of a linear system with Cholesky
   float4 c;
   c[0]=1.0f;
   c[1]=z[0];
   c[2]=c[1]*z[0];
   c[3]=c[2]*z[0];
   c[1]-=b[0];
   c[2]-=mad(L21,c[1],b[1]);
   c[3]-=b[2]+dot(float2(L31,L32),c.yz);
   c.yzw*=float3(InvD11,InvD22,InvD33);
   c[2]-=L32*c[3];
```

```
    c[1]-=dot(float2(L21,L31),c.zw);
    c[0]-=dot(float3(b[0],b[1],b[2]),c.yzw);
    // Solve the cubic equation
    z.yzw=SolveCubic(c);
    // Determine the contribution to the end result
    float4 WeightFactor;
    WeightFactor[0]=Overestimation;
    WeightFactor.yzw=(z.yzw>z.xxx)?float3(0.0f,0.0f,0.0f):
                                   float3(1.0f,1.0f,1.0f);
    // Construct an interpolation polynomial
    float f0=WeightFactor[0];
    float f1=WeightFactor[1];
    float f2=WeightFactor[2];
    float f3=WeightFactor[3];
    float f01=(f1-f0)/(z[1]-z[0]);
    float f12=(f2-f1)/(z[2]-z[1]);
    float f23=(f3-f2)/(z[3]-z[2]);
    float f012=(f12-f01)/(z[2]-z[0]);
    float f123=(f23-f12)/(z[3]-z[1]);
    float f0123=(f123-f012)/(z[3]-z[0]);
    float4 Polynomial;
    // f012+f0123*(z-z2)
    Polynomial[0]=mad(-f0123,z[2],f012);
    Polynomial[1]=f0123;
    // *(z-z1)  +f01
    Polynomial[2]=Polynomial[1];
    Polynomial[1]=mad(Polynomial[1],-z[1],Polynomial[0]);
    Polynomial[0]=mad(Polynomial[0],-z[1],f01);
    // *(z-z0)  +f0
    Polynomial[3]=Polynomial[2];
    Polynomial[2]=mad(Polynomial[2],-z[0],Polynomial[1]);
    Polynomial[1]=mad(Polynomial[1],-z[0],Polynomial[0]);
    Polynomial[0]=mad(Polynomial[0],-z[0],f0);
    // Evaluate the shadow intensity
    return saturate(dot(Polynomial,float4(1.0f,b[0],b[1],b[2])));
}
```

# Bibliography

Akenine-Möller, T. and Assarsson, U. (2002). Approximate soft shadows on arbitrary surfaces using penumbra wedges. In *EGWR02: 13th Eurographics Workshop on Rendering*, pages 297–306. Eurographics Association, doi: 10.2312/EGWR/EGWR02/297-306.

Akhiezer, N. I. (1965). *The Classical Moment Problem and Some Related Questions in Analysis.* University Mathematical Monographs. Oliver & Boyd.

Akhiezer, N. I. and Kreĭn, M. G. (1962). *Some Questions in the Theory of Moments*, volume 2 of *Translations of Mathematical Monographs.* American Mathematical Society, isbn: 978-0-8218-1552-6.

Ammar, G. S. and Gragg, W. B. (1988). Superfast solution of real positive definite Toeplitz systems. *SIAM Journal on Matrix Analysis and Applications*, 9(1):61–76, doi: 10.1137/0609005.

Annen, T., Dong, Z., Mertens, T., Bekaert, P., Seidel, H.-P., and Kautz, J. (2008a). Real-time, all-frequency shadows in dynamic scenes. *ACM Trans. Graph. (Proc. SIGGRAPH 2008)*, 27(3):34:1–34:8, doi: 10.1145/1360612.1360633.

Annen, T., Mertens, T., Bekaert, P., Seidel, H.-P., and Kautz, J. (2007). Convolution shadow maps. In *EGSR07: 18th Eurographics Symposium on Rendering*, pages 51–60. Eurographics Association, doi: 10.2312/EGWR/EGSR07/051-060.

Annen, T., Mertens, T., Seidel, H.-P., Flerackers, E., and Kautz, J. (2008b). Exponential shadow maps. In *GI '08: Proceedings of graphics inter-*

*face 2008*, pages 155–161. Canadian Information Processing Society, doi: 10.20380/GI2008.20.

Bamji, C. S., O'Connor, P., Elkhatib, T., Mehta, S., Thompson, B., Prather, L. A., Snow, D., Akkaya, O. C., Daniel, A., Payne, A. D., Perry, T., Fenton, M., and Chan, V.-H. (2015). A 0.13 $\mu$m cmos system-on-chip for a 512 × 424 time-of-flight image sensor with multi-frequency photo-demodulation up to 130 mhz and 2 gs/s adc. *IEEE Journal of Solid-State Circuits*, 50(1):303–319, doi: 10.1109/JSSC.2014.2364270.

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):178:1–178:10, doi: 10.1145/1882261.1866200.

Bhandari, A., Feigin, M., Izadi, S., Rhemann, C., Schmidt, M., and Raskar, R. (2014a). Resolving multipath interference in Kinect: An inverse problem approach. In *IEEE SENSORS*, pages 614–617. doi: 10.1109/ICSENS.2014.6985073.

Bhandari, A., Kadambi, A., Whyte, R., Barsi, C., Feigin, M., Dorrington, A., and Raskar, R. (2014b). Resolving multipath interference in time-of-flight imaging via modulation frequency diversity and sparse regularization. *Opt. Lett.*, 39(6):1705–1708, doi: 10.1364/OL.39.001705.

Blinn, J. F. (2007). How to solve a cubic equation, part 5: Back to numerics. *IEEE Computer Graphics and Applications*, 27(3):78–89, doi: 10.1109/MCG.2007.60.

Burg, J. P. (1975). *Maximum Entropy Spectral Analysis.* Ph.D. dissertation, Stanford University, Department of Geophysics.

Chen, J., Baran, I., Durand, F., and Jarosz, W. (2011). Real-time volumetric shadows using 1D min-max mipmaps. In *Proceedings of the 15th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, i3D '11, pages 39–46. ACM, doi: 10.1145/1944745.1944752.

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*, SIGGRAPH '84, pages 137–145. ACM, doi: 10.1145/800031.808590.

Crow, F. C. (1977). Shadow algorithms for computer graphics. In *Proceedings of the 4th annual conference on Computer graphics* *and interactive techniques*, SIGGRAPH '77, pages 242–248. ACM, doi: 10.1145/563858.563901.

Crow, F. C. (1984). Summed-area tables for texture mapping. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '84, pages 207–212. ACM, doi: 10.1145/800031.808600.

Cybenko, G. and Van Loan, C. (1986). Computing the minimum eigenvalue of a symmetric positive definite Toeplitz matrix. *SIAM Journal on Scientific and Statistical Computing*, 7(1):123–131, doi: 10.1137/0907009.

Dachsbacher, C. and Stamminger, M. (2003). Translucent shadow maps. In *EGSR03: 14th Eurographics Symposium on Rendering*. The Eurographics Association, doi: 10.2312/EGWR/EGWR03/197-201.

Delalandre, C., Gautron, P., Marvie, J.-E., and François, G. (2011). Transmittance function mapping. In *Proceedings of the 15th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, i3D '11, pages 31–38. ACM, doi: 10.1145/1944745.1944751.

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*, pages 99–107. Eurographics Association, doi: 10.2312/EGGH/EGGH02/099-108.

Donnelly, W. and Lauritzen, A. (2006). Variance shadow maps. In *Proceedings of the 2006 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, i3D '06, pages 161–165. ACM, doi: 10.1145/1111411.1111440.

Dorrington, A. A., Godbaz, J. P., Cree, M. J., Payne, A. D., and Streeter, L. V. (2011). Separating true range measurements from multi-path and scattering interference in commercial range cameras. *Proc. SPIE*, 7864:786404-1–786404-10, doi: 10.1117/12.876586.

Dou, H., Yan, Y., Kerzner, E., Dai, Z., and Wyman, C. (2014). Adaptive depth bias for shadow maps. *Journal of Computer Graphics Techniques (JCGT)*, 3(4):146–162.

Eisemann, E., Schwarz, M., Assarsson, U., and Wimmer, M. (2011). *Real-Time Shadows*. An A K Peters book. CRC Press, isbn: 978-1-56881-438-4.

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*, i3D '10, pages 157–164. ACM, doi: 10.1145/1730804.1730830.

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*, i3D '10, pages 119–125. ACM, doi: 10.1145/1730804.1730823.

Fernando, R. (2005). Percentage-closer soft shadows. In *ACM SIGGRAPH 2005 Sketches*. ACM, doi: 10.1145/1187112.1187153.

Fischer, W. and Lieb, I. (2012). *A Course in Complex Analysis: From Basic Results to Advanced Topics*. Vieweg+Teubner Verlag, isbn: 978-3-83481-576-7.

Freedman, D., Smolin, Y., Krupka, E., Leichter, I., and Schmidt, M. (2014). SRA: Fast removal of general multipath for ToF sensors. In *Computer Vision - ECCV 2014*, volume 8689 of *Lecture Notes in Computer Science*, pages 234–249. Springer International Publishing, doi: 10.1007/978-3-319-10590-1_16.

Fuchs, S. (2010). Multipath interference compensation in time-of-flight camera images. In *20th International Conference on Pattern Recognition (ICPR)*, pages 3583–3586. doi: 10.1109/ICPR.2010.874.

Gao, L., Liang, J., Li, C., and Wang, L. V. (2014). Single-shot compressed ultrafast photography at one hundred billion frames per second. *Nature*, 516, doi: 10.1038/nature14005.

Georgii, H.-O. (2008). *Stochastics, Introduction to Probability and Statistics*. De Gruyter, isbn: 978-3-11-020676-0.

Gkioulekas, I., Levin, A., Durand, F., and Zickler, T. (2015). Micron-scale light transport decomposition using interferometry. *ACM Trans. Graph. (Proc. SIGGRAPH 2015)*, 34(4):37:1–37:14, doi: 10.1145/2766928.

Godbaz, J. P., Cree, M. J., and Dorrington, A. A. (2012). Closed-form inverses for the mixed pixel/multipath interference problem in AMCW lidar. *Proc. SPIE*, 8296:829618-1–829618-15, doi: 10.1117/12.909778.

Greenbaum, A. and Chartier, T. P. (2012). *Numerical methods – Design, analysis and computer implementation of algorithms.* Princeton University Press, isbn: 978-0-69115-122-9.

Guennebaud, G., Barthe, L., and Paulin, M. (2006). Real-time soft shadow mapping by backprojection. In *EGSR06: 17th Eurographics Symposium on Rendering*, pages 227–234. Eurographics Association, doi: 10.2312/EGWR/EGSR06/227-234.

Gupta, M., Nayar, S. K., Hullin, M. B., and Martín, J. (2015). Phasor imaging: A generalization of correlation-based time-of-flight imaging. *ACM Trans. Graph.*, 34(5):156:1–156:18, doi: 10.1145/2735702.

Heide, F., Hullin, M. B., Gregson, J., and Heidrich, W. (2013). Low-budget transient imaging using photonic mixer devices. *ACM Trans. Graph. (Proc. SIGGRAPH 2013)*, 32(4):45:1–45:10, doi: 10.1145/2461912.2461945.

Heide, F., Xiao, L., Kolb, A., Hullin, M. B., and Heidrich, W. (2014). Imaging in scattering media using correlation image sensors and sparse convolutional coding. *Opt. Express*, 22(21):26338–26350, doi: 10.1364/OE.22.026338.

Hendrickx, Q., Scandolo, L., Eisemann, M., and Eisemann, E. (2015). Adaptively layered statistical volumetric obscurance. In *Proceedings of the 7th Conference on High-Performance Graphics*, HPG '15, pages 77–84. ACM, doi: 10.1145/2790060.2790070.

Jansen, J. and Bavoil, L. (2010). Fourier opacity mapping. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, i3D '10, pages 165–172. ACM, doi: 10.1145/1730804.1730831.

Jarabo, A., Marco, J., Muñoz, A., Buisan, R., Jarosz, W., and Gutierrez, D. (2014). A framework for transient rendering. *ACM Trans. Graph. (Proc. Siggraph Asia 2014)*, 33(6):177:1–177:10, doi: 10.1145/2661229.2661251.

Jimenez, D., Pizarro, D., Mazo, M., and Palazuelos, S. (2012). Modelling and correction of multipath interference in time of flight cameras. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 893–900. doi: 10.1109/CVPR.2012.6247763.

Kadambi, A., Schiel, J., and Raskar, R. (2016). Macroscopic interferometry: Rethinking depth estimation with frequency-domain time-of-flight. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 893–902.

Kadambi, A., Whyte, R., Bhandari, A., Streeter, L., Barsi, C., Dorrington, A., and Raskar, R. (2013). Coded time of flight cameras: Sparse deconvolution to address multipath interference and recover time profiles. *ACM Trans. Graph. (Proc. SIGGRAPH Asia 2013)*, 32(6):167:1–167:10, doi: 10.1145/2508363.2508428.

Karlin, S. and Shapley, L. S. (1953). Geometry of moment spaces. *Memoirs of the American Mathematical Society*, (12), doi: 10.1090/memo/0012.

Karlin, S. and Studden, W. J. (1966). *Tchebycheff systems: with applications in analysis and statistics.* Pure and applied mathematics. Interscience Publishers.

Karlsson, J. and Georgiou, T. (2013). Uncertainty bounds for spectral estimation. *IEEE Transactions on Automatic Control*, 58(7):1659–1673, doi: 10.1109/TAC.2013.2251775.

Kim, T.-Y. and Neumann, U. (2001). Opacity shadow maps. In *EGWR01: 12th Eurographics Workshop on Rendering.* The Eurographics Association, doi: 10.2312/EGWR/EGWR01/177-182.

Kirmani, A., Benedetti, A., and Chou, P. (2013). SPUMIC: Simultaneous phase unwrapping and multipath interference cancellation in time-of-flight cameras using spectral methods. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. doi: 10.1109/ICME.2013.6607553.

Klehm, O., Seidel, H.-P., and Eisemann, E. (2014a). Filter-based real-time single scattering using rectified shadow maps. *Journal of Computer Graphics Techniques (JCGT)*, 3(3):7–34.

Klehm, O., Seidel, H.-P., and Eisemann, E. (2014b). Prefiltered single scattering. In *Proceedings of the 18th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, i3D '14, pages 71–78. ACM, doi: 10.1145/2556700.2556704.

Klein, J., Peters, C., Martín, J., Laurenzis, M., and Hullin, M. B. (2016). Tracking objects outside the line of sight using 2D intensity images. *Scientific Reports*, 6(32491), doi: 10.1038/srep32491.

Kreĭn, M. G. and Nudel'man, A. A. (1977). *The Markov Moment Problem and Extremal Problems*, volume 50 of *Translations of Mathematical Monographs.* American Mathematical Society, isbn: 978-0-8218-4500-4.

Landau, H. J. (1987). Maximum entropy and the moment problem. *Bull. Amer. Math. Soc. (N.S.)*, 16(1):47–77.

Lauritzen, A. (2007). *GPU Gems 3*, chapter Summed-Area Variance Shadow Maps, pages 157–182. Addison-Wesley, isbn: 978-0-32151-526-1.

Lauritzen, A. and McCool, M. (2008). Layered variance shadow maps. In *Proceedings of graphics interface 2008*, pages 139–146. Canadian Information Processing Society, doi: 10.20380/GI2008.18.

Lauritzen, A., Salvi, M., and Lefohn, A. (2011). Sample distribution shadow maps. In *Proceedings of the 15th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, i3D '11, pages 97–102. ACM, doi: 10.1145/1944745.1944761.

Lefloch, D., Nair, R., Lenzen, F., Schäfer, H., Streeter, L., Cree, M., Koch, R., and Kolb, A. (2013). Technical foundation and calibration methods for time-of-flight cameras. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, volume 8200 of *Lecture Notes in Computer Science*, pages 3–24. Springer Berlin Heidelberg, doi: 10.1007/978-3-642-44964-2_1.

Liktor, G., Spassov, S., Mückl, G., and Dachsbacher, C. (2015). Stochastic soft shadow mapping. *Computer Graphics Forum*, 34(4):1–11, doi: 10.1111/cgf.12673.

Lin, J., Liu, Y., Hullin, M. B., and Dai, Q. (2014). Fourier analysis on transient imaging with a multifrequency time-of-flight camera. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3230–3237. doi: 10.1109/CVPR.2014.419.

Lokovic, T. and Veach, E. (2000). Deep shadow maps. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 385–392. ACM Press/Addison-Wesley Publishing Co., doi: 10.1145/344779.344958.

Loos, B. J. and Sloan, P.-P. (2010). Volumetric obscurance. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, i3D '10, pages 151–156. ACM, doi: 10.1145/1730804.1730829.

Markov, A. (1884). *On certain applications of algebraic continued fractions.* Ph.D. dissertation, St. Petersburg University. Russian.

Martin, T. and Tan, T.-S. (2004). Anti-aliasing and continuity with trapezoidal shadow maps. In *EGSR04: 15th Eurographics Symposium on Rendering*, pages 153–160. Eurographics Association, doi: 10.2312/EGWR/EGSR04/153-160.

Max, N. L. (1986). Atmospheric illumination and shadows. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, pages 117–124. ACM, doi: 10.1145/15922.15899.

McGuire, M. and Enderton, E. (2011). Colored stochastic shadow maps. In *Proceedings of the 15th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, i3D '11, pages 89–96. ACM, doi: 10.1145/1944745.1944760.

McGuire, M. and Mara, M. (2016). A phenomenological scattering model for order-independent transparency. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, i3D '16, pages 149–158. ACM, doi: 10.1145/2856400.2856418.

Mulholland, H. P. and Rogers, C. A. (1958). Representation theorems for distribution functions. *Proc. London Math. Soc.*, s3-8(2):177–223, doi: 10.1112/plms/s3-8.2.177.

Naik, N., Zhao, S., Velten, A., Raskar, R., and Bala, K. (2011). Single view reflectance capture using multiplexed scattering and time-of-flight imaging. *ACM Trans. Graph. (Proc. SIGGRAPH Asia 2011)*, 30(6):171:1–171:10, doi: 10.1145/2070781.2024205.

Olsson, O., Billeter, M., Sintorn, E., Kämpe, V., and Assarsson, U. (2015). More efficient virtual shadow maps for many lights. *IEEE Transactions on Visualization and Computer Graphics*, 21(6):701–713, doi: 10.1109/TVCG.2015.2418772.

Olsson, O., Sintorn, E., Kämpe, V., Billeter, M., and Assarsson, U. (2014). Efficient virtual shadow maps for many lights. In *Proceedings of the 18th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, i3D '14, pages 87–96. ACM, doi: 10.1145/2556700.2556701.

Papaioannou, G. (2011). Real-time diffuse global illumination using radiance hints. In *Proceedings of the 3rd Conference on High-Performance Graphics*, HPG '11, pages 15–24. ACM, doi: 10.1145/2018323.2018326.

Payne, A. D., Dorrington, A. A., Cree, M. J., and Carnegie, D. A. (2010). Improved measurement linearity and precision for AMCW time-of-flight range imaging cameras. *Appl. Opt.*, 49(23):4392–4403, doi: 10.1364/AO.49.004392.

Peters, C. (2013). *Moment Shadow Mapping.* Master's thesis, University of Bonn.

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 2015)*, 34(6), doi: 10.1145/2816795.2818103.

Peters, C. and Klein, R. (2015). Moment shadow mapping. In *Proceedings of the 19th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, i3D '15, pages 7–14. ACM, doi: 10.1145/2699276.2699277.

Peters, C., Münstermann, C., Wetzstein, N., and Klein, R. (2016). Beyond hard shadows: Moment shadow maps for single scattering, soft shadows and translucent occluders. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, i3D '16, pages 159–170. ACM, doi: 10.1145/2856400.2856402.

Prékopa, A. (1990). The discrete moment problem and linear programming. *Discrete Applied Mathematics*, 27(3):235–254, doi: 10.1016/0166-218X(90)90068-N.

Qiao, H., Lin, J., Liu, Y., Hullin, M. B., and Dai, Q. (2015). Resolving transient time profile in ToF imaging via log-sum sparse regularization. *Opt. Lett.*, 40(6):918–921, doi: 10.1364/OL.40.000918.

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*, SIGGRAPH '87, pages 283–291. ACM, doi: 10.1145/37401.37435.

Salvi, M. (2007). A (not so) little teaser. Blog post including notes on moment shadow mapping, retrieved on 1st of September 2016. pixelstoomany.wordpress.com/2007/09/03/a-not-so-little-teaser.

Salvi, M. (2008). *ShaderX$^6$*, chapter Rendering filtered shadows with exponential shadow maps, pages 257–274. Cengage Learning Inc., isbn: 9781584505440.

Salvi, M., Vidimče, K., Lauritzen, A., and Lefohn, A. (2010). Adaptive volumetric shadow maps. *Computer Graphics Forum*, 29(4):1289–1296, doi: 10.1111/j.1467-8659.2010.01724.x.

Schrijver, A. (1986). *Theory of linear and integer programming.* Wiley-Interscience Series in Discrete Mathematics. Wiley, isbn: 978-0-471-90854-8.

Schwärzler, M., Luksch, C., Scherzer, D., and Wimmer, M. (2013). Fast percentage closer soft shadows using temporal coherence. In *Proceedings of the 17th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, i3D '13, pages 79–86. ACM, doi: 10.1145/2448196.2448209.

Shen, L., Feng, J., and Yang, B. (2013). Exponential soft shadow mapping. *Computer Graphics Forum*, 32(4), doi: 10.1111/cgf.12156.

Shrestha, S., Heide, F., Heidrich, W., and Wetzstein, G. (2016). Computational imaging with multi-camera time-of-flight systems. *ACM Trans. Graph. (Proc. SIGGRAPH 2016)*, 35(4):33:1–33:11, doi: 10.1145/2897824.2925928.

Story, J. and Wyman, C. (2016). HFTS: Hybrid frustum-traced shadows in "the division". In *ACM SIGGRAPH 2016 Talks*, SIGGRAPH '16, pages 13:1–13:2. ACM, doi: 10.1145/2897839.2927424.

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

Tchebichef, M. P. (1874). Sur les valeurs limites des intégrales. *Journal de Mathématiques Pures et Appliquées*, 2(19):157–160.

Toksvig, M. (2005). Mipmapping normal maps. *Journal of Graphics Tools*, 10(3):65–71, doi: 10.1080/2151237X.2005.10129203.

Tóth, B. and Umenhoffer, T. (2009). Real-time volumetric lighting in participating media. In *Eurographics 2009 - Short Papers.* The Eurographics Association, doi: 10.2312/egs.20091048.

Trefethen, L. N. and Bau, D. (1997). *Numerical Linear Algebra.* SIAM, isbn: 978-0-898-71957-4.

Velten, A., Lawson, E., Bardagjy, A., Bawendi, M., and Raskar, R. (2011). Slow art with a trillion frames per second camera. In *ACM SIGGRAPH 2011 Posters*, pages 13:1–13:1. doi: 10.1145/2037715.2037730.

Velten, A., Willwacher, T., Gupta, O., Veeraraghavan, A., Bawendi, M. G., and Raskar, R. (2012). Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nature Communications*, 3, doi: 10.1038/ncomms1747.

Velten, A., Wu, D., Jarabo, A., Masia, B., Barsi, C., Joshi, C., Lawson, E., Bawendi, M., Gutierrez, D., and Raskar, R. (2013). Femto-photography: Capturing and visualizing the propagation of light. *ACM Trans. Graph. (Proc. SIGGRAPH 2013)*, 32(4):44:1–44:8, doi: 10.1145/2461912.2461928.

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, doi: 10.1111/cgf.12348.

Williams, L. (1978). Casting curved shadows on curved surfaces. In *Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '78, pages 270–274. ACM, doi: 10.1145/800248.807402.

Wimmer, M., Scherzer, D., and Purgathofer, W. (2004). Light space perspective shadow maps. In *EGSR04: 15th Eurographics Symposium on Rendering*, pages 143–152. Eurographics Association, doi: 10.2312/EGWR/EGSR04/143-151.

Wu, D., Velten, A., O'Toole, M., Masia, B., Agrawal, A., Dai, Q., and Raskar, R. (2014). Decomposing global light transport using time of flight imaging. *International Journal of Computer Vision*, 107(2):123–138, doi: 10.1007/s11263-013-0668-2.

Wyman, C. (2011). Voxelized shadow volumes. In *Proceedings of the 3rd Conference on High-Performance Graphics*, HPG '11, pages 33–40. ACM, doi: 10.1145/2018323.2018329.

Wyman, C. and Dai, Z. (2013). Imperfect voxelized shadow volumes. In *Proceedings of the 5th Conference on High-Performance Graphics*, HPG '13, pages 45–52. ACM, doi: 10.1145/2492045.2492050.

Wyman, C., Hoetzlein, R., and Lefohn, A. (2015). Frustum-traced raster shadows: Revisiting irregular z-buffers. In *Proceedings of the 19th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, i3D '15, pages 15–23. ACM, doi: 10.1145/2699276.2699280.

Wyman, C. and Ramsey, S. (2008). Interactive volumetric shadows in participating media with single-scattering. In *IEEE Symposium on Interactive Ray Tracing*, RT 2008, pages 87–92. doi: 10.1109/RT.2008.4634627.

Yang, B., Dong, Z., Feng, J., Seidel, H.-P., and Kautz, J. (2010). Variance soft shadow mapping. *Computer Graphics Forum*, 29(7), doi: 10.1111/j.1467-8659.2010.01800.x.

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*, pages 311–318. ACM, doi: 10.1145/1128923.1128975.

# Index

# Nomenclature

| | |
|---|---|
| $\bar{\cdot}$ | Complex conjugate, page 26 |
| $\cdot^*$ | Conjugate transpose, page 26 |
| $\mathbf{a}$ | General-moment-generating function, page 23 |
| $a$ | Vector of general moments, page 23 |
| $\alpha_b$ | Moment bias, page 63 |
| $\mathbf{b}$ | Power-moment-generating function, page 24 |
| $b$ | Vector of power moments, page 24 |
| $B(b)$ | Hankel matrix, page 25 |
| $\hat{\mathbf{b}}$ | Hankel-matrix-generating function, page 25 |
| $b^\star$ | Vector of biasing moments, page 64 |
| $\mathbf{c}$ | Trigonometric-moment-generating function, page 26 |
| $c$ | Vector of trigonometric moments, page 26 |
| $C(c)$ | Toeplitz matrix, page 26 |
| $D$ | Density of a measure, page 122 |
| $\delta_x$ | Dirac-delta distribution, page 22 |
| $e.$ | Canonical basis vector, page 59 |
| $\mathcal{E}_P\left(\mathbf{a}\right)$ | Expectation, page 23 |

| | |
|---|---|
| $F$ | Measure modeling the phase-resolved impulse response for a transient pixel, page 121 |
| $F_D$ | Measure with density $D$, page 122 |
| $G_{\mathbb{I},\mathbf{a}}(a, z_f)$ | Optimal, lower bound, page 45 |
| $G$ | Measure modeling the time-resolved impulse response for a transient pixel, page 118 |
| $\mathcal{H}_{\text{Burg}}$ | Burg entropy, page 122 |
| $\mathbb{I}$ | Range of shadow map depth values, page 45 |
| $i$ | Imaginary unit ($i^2 = -1$), page 26 |
| $m$ | Number of considered moments, page 23 |
| $\mathbb{P}(\mathbb{I})$ | Set of probability distributions on $\mathbb{I}$, page 45 |
| $P_r$ | Poisson kernel, page 133 |
| $\int \cdot \, \mathrm{d}M(x)$ | Integral with respect to a measure, page 22 |
| $\Theta_m^\star$ | Optimized quantization transform, page 65 |
| $Z$ | Depth distribution, page 40 |
| $\mathbf{z}$ | Depth random variable $\mathbf{z}(z) = z$, page 40 |
| $z_f$ | Depth of fragment being shaded, page 39 |