Supplementary Document on Moment Shadow Mapping

Christoph Peters*
University of Bonn, Germany

Reinhard Klein†
University of Bonn, Germany

In this supplementary document we provide information relevant to those who want to build upon our work. This includes technical details on the evaluation of candidate techniques, descriptions of Hausdorff 4MSM and TMSM, a lot of mathematical background and HLSL code listings for Hamburger 4MSM and Hausdorff 4MSM. A still more detailed discussion of most results can be found in [Peters 2013].

The document is to be understood as appendix of the paper. It is not self-contained and includes many direct references to the paper.

Contents

6 Details on the Choice of Shadow Map Data

In the paper we describe a fully automated method to measure the performance of shadow mapping techniques and apply it to techniques based upon Algorithm 1. We have yet to provide some details needed for reproduction of our results.

The evaluated candidate techniques use all possible combinations of shadow map data generated by 37 different depth-dependent scalar functions. A complete list of these functions follows:

- Polynomials: $z^1, \ldots, z^8$
- Roots: $\sqrt{z}$, $\sqrt[3]{z}$, $\sqrt[4]{z}$
- Rational functions: $\frac{1}{(z+1)^1}, \ldots, \frac{1}{(z+1)^4}$
- Scaled exponential functions: $\exp(1 \cdot z), \ldots, \exp(4 \cdot z)$
- Shifted logarithm functions: $\log(z+1), \ldots, \log(z+4)$
- Fourier basis functions: $\sin(1 \cdot 2 \cdot \pi \cdot z), \ldots, \sin(4 \cdot 2 \cdot \pi \cdot z)$, $\cos(1 \cdot 2 \cdot \pi \cdot z), \ldots, \cos(4 \cdot 2 \cdot \pi \cdot z)$

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

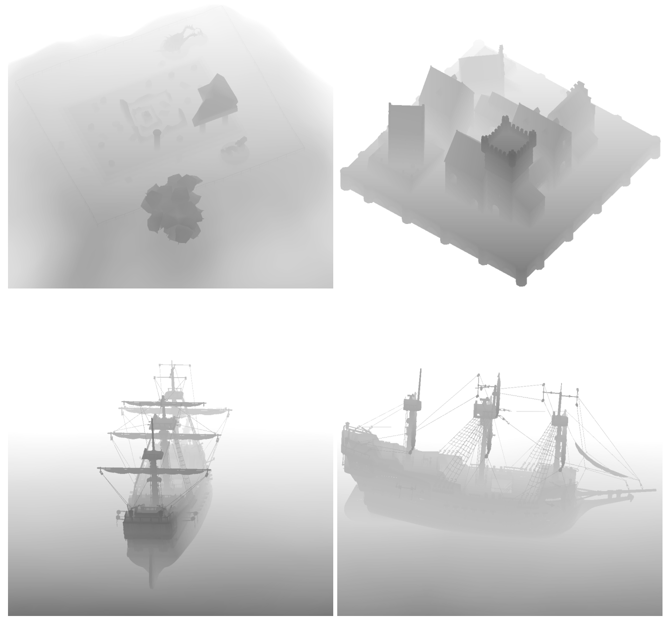

Figure 9: *All shadow maps used for evaluation of candidate techniques in the fully automated setup. Their resolution is* 1024^2.

- Trigonometric functions: $\cosh(z)$, $\sinh(z)$, $\arcsin(z)$, $\arcsin(2 \cdot z - 1)$, $\arctan(z)$
- A Gaussian probability density function: $\exp(-z^2)$

The resulting techniques are evaluated on three different scenes using a total of four different light directions. The corresponding shadow maps are shown in Figure 9. Evaluation requires a full stack of shadow maps showing not only the foremost surface but all surfaces. Due to memory restrictions we will provide these images on the project webpage or on request.

The evaluation is sensitive to the biasing of percentage closer filtering, which provides the ground truth. If the ground truth contains wrong self-shadowing, the evaluated error term punishes techniques for not reproducing this artifact. Therefore, our evaluation uses an optimized biasing derived from the slope of the surface with respect to the light. Percentage closer filtering and all candidate techniques use the same biasing. You will find a stack of images containing the values of this bias on the project webpage.

As an optimization we have used a slightly more sophisticated variant of Algorithm 1 than presented in the paper. We first run it using 251 uniform samples of $[0, 1]$. Then we refine the sampling near support points of the output distribution to enable a more precise positioning of these support points and rerun Algorithm 1 with the refined sampling. This is done iteratively. In total Algorithm 1 is invoked three times. It should be noted that the remaining discretization error is large enough to impact the ranking of candidate techniques. Still, the conclusion that many techniques are close to the optimum and that Hamburger four moment shadow mapping is among them remains valid.

The histogram in Figure 4 uses an average of the results for all shadow maps in Figure 9 weighted by the number of fragments in the stack of shadow maps. The qualitative behavior is similar on all three scenes. The scores of each candidate technique for each shadow map can be found in the supplementary data.

It should also be noted that the x-axis of the histogram is cut off for the sake of readability. A few outlier techniques perform very poorly. For example choosing

$$\mathbf{b}(z) = (\sin(4\pi z), \cos(4\pi z), \sin(8\pi z), \cos(8\pi z))^\mathsf{T}$$

produces the largest measured error of 29.6%. This poor performance is due to symmetries in this choice of data. Any moment vector can be explained using solely depth values greater than $\frac{3}{4}$ and thus the computed lower bound is always zero up to this point.

7 Moment Problems

Algorithm 2 can be used as black box. Still, it can be beneficial to understand its inner workings. This has aided us in the derivation of algorithms that are well-suited for real-time applications and can help in finding future applications of the powerful theory of moments.

In the present section we provide proof for all corresponding propositions in the paper and we describe Hausdorff moment shadow mapping and trigonometric moment shadow mapping. For statements on existence and uniqueness of solutions we generally reference the literature but provide self-contained proofs for the more algorithmic aspects of the theory.

The paper claims that the Hamburger moment problem is always solved by a linear combination of $\frac{m}{2}+1$ Dirac-δ distributions. In a more general formulation many moment problems can be solved by canonical representations [Kreĭn and Nudel'man 1977, p. 35, p. 77].

Definition 7. Let $m \in \mathbb{N}$ be even, let $I := [\alpha, \beta] \subset \mathbb{R}$ and let $\mathbf{b} : I \to \mathbb{R}^m$ with $\mathbf{b}(z) := (z^j)_{j=1}^m$. Let $w_1, \ldots, w_n > 0$, let $z_1, \ldots, z_n \in I$ be pairwise different and let

$$Z := \sum_{i=1}^n w_i \cdot \delta_{z_i} \in \mathfrak{P}(I).$$

We call

$$\varepsilon(Z) := \sum_{i=1}^n \begin{cases} 1 & \text{if } z_i \in \{\alpha, \beta\} \\ 2 & \text{if } z_i \in (\alpha, \beta) \end{cases}$$

the *index* of Z. For distributions $Z \in \mathfrak{P}(\mathbb{R})$ the index is the number of support points multiplied by two. If $b = \mathbb{E}_Z(\mathbf{b})$ and $\varepsilon(Z) \leq m+2$, we refer to Z as *canonical representation* of b.

The importance of canonical representations stems from the following proposition which is a formulation of the Markov-Krein theorem (also known as Chebyshev-Markov inequality).

Proposition 8 (Markov-Krein theorem)**.** *Let $b \in \mathbb{R}^m$ such that $\mathfrak{S}(b)$ is non-empty. Then $\mathfrak{S}(b)$ either contains exactly one canonical representation $S \in \mathfrak{S}(b)$ and for this distribution $\varepsilon(S) \leq m$ or $\mathfrak{S}(b)$ contains a unique canonical representation $S \in \mathfrak{S}(b)$ with support at any given $z_f \in (\alpha, \beta)$. In both cases $G(b, z_f) = F_S(z_f)$.*

Proof. In the case where $\mathfrak{S}(b)$ contains exactly one distribution, the claim $\varepsilon(S) \leq m$ follows from [Kreĭn and Nudel'man 1977, p. 78, Theorem III.4.1] and $G(b, z_f) = F_S(z_f)$ is trivial.

Otherwise, the existence of a unique canonical representation $S \in \mathfrak{S}(b)$ with support at $z_f \in I$ follows from [Kreĭn and Nudel'man 1977, p. 58, Theorem III.1.1] and [Kreĭn and Nudel'man 1977, p. 79, Theorem III.4.3]. The equality $G(b, z_f) = F_S(z_f)$ follows from the Markov-Krein theorem [Kreĭn and Nudel'man 1977, p.125, Theorem IV.3.1]. □

Proposition 8 tells us that Problem 1 for $\mathbf{b}(z) := (z^j)_{j=1}^m$ on I is always solved by canonical representations. The degrees of freedom of canonical representations coincide with their index which is at most $m+2$. On the other hand, we have $m+2$ constraints; S must have support at z_f if $\varepsilon(S) > m$, $\mathbb{E}_S(\mathbf{b}) = b$ and $S(I) = 1$. The remaining problem is entirely algebraic. We need to find the unique S matching the constraints.

7.1 The Hamburger Moment Problem

We now consider Problem 1 for $I = \mathbb{R}$ and $\mathbf{b}(z) := (z^j)_{j=1}^m$ which is known as Hamburger moment problem due to Hans Ludwig Hamburger. The matrix B defined in Algorithm 2 is essential for our discussion. It may be written as

$$B = \begin{pmatrix} 1 & b_1 & \cdots & b_{\frac{m}{2}} \\ b_1 & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & b_{m-1} \\ b_{\frac{m}{2}} & \cdots & b_{m-1} & b_m \end{pmatrix} \in \mathbb{R}^{n \times n}$$

where $n := \frac{m}{2}+1$. Its skew-diagonals are constant which makes it a *Hankel matrix*. According to Proposition 4 positive semi-definite B is a necessary condition for a $Z \in \mathfrak{S}(\mathbf{b})$ to exist. For Algorithm 2 to be correct, this also has to be a sufficient criterion with a few exceptions. This is indeed the case.

Proposition 9. *Let B be positive definite, $z_f \in \mathbb{R}$ and*

$$c := B^{-1} \cdot (1, z_f, \ldots, z_f^{n-1})^\mathsf{T}.$$

If $c_n \neq 0$, there exists a unique canonical representation $S \in \mathfrak{S}(b)$ with support at z_f and exactly $n-1$ other points.

Proof. The proposition follows directly from [Akhiezer and Kreĭn 1962, p. 8, Theorem 3 c)]. □

In the case $c_n = 0$ there is no single distribution realizing the infimum $G(b, z_f)$ but a series of distributions approximating it. The infimum can still be computed but since the problem occurs only for isolated values of z_f (c_n is a non-constant polynomial in z_f) and does not occur at all for the Hausdorff moment problem, we disregard this case.

Now we provide the reasoning behind Step 5 of Algorithm 2. Equivalent results are used in [Tari 2005, p. 14].

Proposition 10. *Let $z_1, \ldots, z_n \in \mathbb{R}$ be pairwise different, let $w \in \mathbb{R}^n$ and let $S := \sum_{i=1}^n w_i \cdot \delta_{z_i} \in \mathfrak{S}(b)$. Let B be a regular Hankel matrix. Then for all $i \in \{2, \ldots, n\}$*

$$(z_i^0, \ldots, z_i^{n-1}) \cdot B^{-1} \cdot (z_1^0, \ldots, z_1^{n-1})^\mathsf{T} = 0. \tag{1}$$

Furthermore, for all $i \in \{1, \ldots, n\}$

$$w_i = \frac{1}{(z_i^0, \ldots, z_i^{n-1}) \cdot B^{-1} \cdot (z_i^0, \ldots, z_i^{n-1})^\mathsf{T}}$$

and these values are positive if B is positive-definite.

Proof. We introduce the function

$$\begin{aligned} \hat{\mathbf{b}} : \mathbb{R} &\to \mathbb{R}^n \\ z &\mapsto (z^{j-1})_{j=1}^n. \end{aligned}$$

This way our claim is equivalent to $\hat{\mathbf{b}}^\mathsf{T}(z_i) \cdot B^{-1} \cdot \hat{\mathbf{b}}(z_1) = 0$. Furthermore, we consider the matrix $\hat{A}$ from Algorithm 2 which can be written as

$$\hat{A} := (\hat{\mathbf{b}}(z_1), \dots, \hat{\mathbf{b}}(z_n)) \in \mathbb{R}^{n \times n}.$$

This matrix is a square Vandermonde matrix and since $z_1, \dots, z_n$ are pairwise different, it is invertible. We can also use $\hat{\mathbf{b}}$ to express B. Using

$$\hat{\mathbf{b}}(z) \cdot \hat{\mathbf{b}}^\mathsf{T}(z) = (z^{j-1} \cdot z^{k-1})_{j,k=1}^n = (z^{j+k-2})_{j,k=1}^n$$

we obtain $B = \mathbb{E}_S(\hat{\mathbf{b}} \cdot \hat{\mathbf{b}}^\mathsf{T})$ and thus:

$$\begin{aligned}
\hat{A}^{-1} \cdot B \cdot \hat{A}^{-\mathsf{T}} &= \hat{A}^{-1} \cdot \mathbb{E}_S \left(\hat{\mathbf{b}} \cdot \hat{\mathbf{b}}^\mathsf{T}\right) \cdot \hat{A}^{-\mathsf{T}} \\
&= \hat{A}^{-1} \cdot \left(\sum_{i=1}^n w_i \cdot \hat{\mathbf{b}}(z_i) \cdot \hat{\mathbf{b}}^\mathsf{T}(z_i)\right) \cdot \hat{A}^{-\mathsf{T}} \\
&= \sum_{i=1}^n w_i \cdot \left(\hat{A}^{-1} \cdot \hat{\mathbf{b}}(z_i)\right) \cdot \left(\hat{A}^{-1} \cdot \hat{\mathbf{b}}(z_i)\right)^\mathsf{T} \\
&= \sum_{i=1}^n w_i \cdot e_i \cdot e_i^\mathsf{T} = \operatorname{diag}(w_1, \dots, w_n)
\end{aligned}$$

The inverse of this diagonal matrix $\hat{A}^\mathsf{T} \cdot B^{-1} \cdot \hat{A}$ is still a diagonal matrix and therefore the entry $i, 1$ is zero for $i \in \{2, \dots, n\}$. This entry coincides with $\hat{\mathbf{b}}^\mathsf{T}(z_i) \cdot B^{-1} \cdot \hat{\mathbf{b}}(z_1)$. Also, entry i, i is

$$\frac{1}{w_i} = \hat{\mathbf{b}}^\mathsf{T}(z_i) \cdot B^{-1} \cdot \hat{\mathbf{b}}(z_i)$$

for all $i \in \{1, \dots, n\}$. The positivity of w_i is then trivial for positive-definite B. ☐

With these propositions the correctness proof for Algorithm 2 is easily provided.

Proof of Proposition 3. By Proposition 9 there exists a distribution $S \in \mathfrak{S}(b)$ with support at $z_1 = z_f$ and exactly $n-1$ other points $z_2, \dots, z_n$. By Proposition 10 the points $z_2, \dots, z_n$ have to be roots of Equation (1). Thus, Algorithm 2 computes these points correctly.

Let $w \in \mathbb{R}^n$ such that $S = \sum_{i=1}^n w_i \cdot \delta_{z_i}$. The system of linear equations $\hat{A} \cdot w = (1, b_1, \dots, b_{n-1})^\mathsf{T}$ is equivalent to

$$\sum_{i=1}^n w_i = 1, \qquad \forall j \in \{1, \dots, n-1\} : \sum_{i=1}^n w_i \cdot z_i^j = b_j.$$

Obviously, both equations hold if $S \in \mathfrak{S}(b)$. Furthermore, the system of linear equations uniquely determines w because $\hat{A}$ is invertible (see proof of Proposition 10). Thus, Algorithm 2 computes w correctly. By Proposition 8 $G(b, z_f) = F_S(z_f)$ and thus Algorithm 2 returns the correct value.

It remains to prove that $c_n = 0$ cannot occur for more than $n-1$ different values of z_f if a regular B is fixed. Obviously, c_n is a polynomial of degree $n-1$ in z_f. Suppose this polynomial is constant zero. Then the bottom row of $B^{-1} \cdot \hat{A}$ is all zero. Contradiction. Thus, c_n as function of z_f can have at most $n-1$ roots. ☐

7.2 The Hausdorff Moment Problem

For the Hausdorff moment problem (named after Felix Hausdorff) we need to distinguish two cases. The first case governs the behavior of the solution in most practical cases and is no different from the Hamburger moment problem. It occurs if the Hamburger moment problem has a solution which has all of its support within $[0, 1]$. Only if this is not the case, special treatment is necessary.

If such a solution does not exist although B is positive-definite, the solution to the problem must have more than $\frac{m}{2} + 1$ points of support. By Proposition 8 we know that the solution is a canonical representation and therefore the additional support can only lie at α and β. More precisely, the distribution must have support at α, β and $\frac{m}{2}$ other points.

We now restrict our considerations to the case $m = 4$ which allows for a particularly simple solution.

Proposition 11. *Let $m = n = 4$ and let $z_1 := 0$, $z_2 := z \in [0, 1]$, $z_3 := z_f \in [0, 1]$ and $z_4 := 1$ such that z_1, z_2, z_3, z_4 are pairwise different. Let $w \in \mathbb{R}^n$ such that $S := \sum_{i=1}^n w_i \cdot \delta_{z_i} \in \mathfrak{S}(b)$. Then*

$$z = \frac{(b_3 - b_2) \cdot z_f + b_3 - b_4}{(b_2 - b_1) \cdot z_f + b_2 - b_3}. \tag{2}$$

Proof. We use $\overline{A}$ and $\overline{b}$ as in Algorithm 1, i.e.

$$\overline{A} := (z_i^{j-1})_{j \in \{1, \dots, 5\}, i \in \{1, \dots, 4\}} \in \mathbb{R}^{5 \times 4}, \qquad \overline{b} = \begin{pmatrix} 1 \\ b \end{pmatrix}.$$

Then

$$S \in \mathfrak{S}(b) \quad \Rightarrow \quad \overline{A} \cdot w = \overline{b} \quad \Rightarrow \quad \det(\overline{A} \mid \overline{b}) = 0.$$

Hence, we are now interested in roots of the following determinant with respect to z:

$$\det(\overline{A} \mid \overline{b}) = \det \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & z & z_f & 1 & b_1 \\ 0 & z^2 & z_f^2 & 1 & b_2 \\ 0 & z^3 & z_f^3 & 1 & b_3 \\ 0 & z^4 & z_f^4 & 1 & b_4 \end{pmatrix}$$

Obviously, two identical columns arise for $z = 0$, $z = z_f$ and $z = 1$ as well as $z_f = 0$ and $z_f = 1$. However, z and z_f cannot realize any of these roots because $0, z, z_f, 1$ are pairwise different. We perform polynomial division to eliminate them:

$$\begin{aligned}
0 =& \frac{\det(\overline{A} \mid \overline{b})}{z \cdot (z - z_f) \cdot (z - 1) \cdot z_f \cdot (1 - z_f)} \\
=& ((b_2 - b_1) \cdot z_f + b_2 - b_3) \cdot z + (b_2 - b_3) \cdot z_f + b_4 - b_3
\end{aligned}$$

Solving for z yields the claimed expression. ☐

We can now present the algorithm used for Hausdorff four moment shadow mapping and prove its correctness.

Proposition 12. *Algorithm 4 solves Problem 1 for $\mathbf{b}(z) = (z, z^2, z^3, z^4)^\mathsf{T}$ and $I = [0, 1]$. Furthermore, it only fails if $\mathfrak{S}(b)$ contains exactly one distribution or is empty.*

Proof. Suppose that $\mathfrak{S}(b)$ contains more than one distribution. First we note that we can disregard the case $z_f = 0$ because then $G(b, z_f) = 0$ is correctly computed and the case $z_f = 1$ because then the solution must have exactly three points of support with one of them being z_f and the first branch of the algorithm solves the problem.

Algorithm 4 Solution to Problem 1 for $\mathbf{b}(z) = (z, z^2, z^3, z^4)^\mathsf{T}$ and $I = [0, 1]$ (Hausdorff 4MSM).
Input: $b \in \mathbb{R}^4$, $z_f \in [0, 1]$
Output: $G(b, z_f)$

1. $B := (b_{j+k-2})_{j,k=1}^3 \in \mathbb{R}^{3\times 3}$ (with $b_0 := 1$)
2. If B is not positive-definite: Indicate failure and abort
3. $z_1 := z_f$
4. $c := B^{-1} \cdot (1, z_1, z_1^2)^\mathsf{T} \in \mathbb{R}^3$
5. Solve $c_3 \cdot z^2 + c_2 \cdot z + c_1 = 0$ for z. If exactly two distinct solutions $z_2, z_3 \in [0, 1]$ exist:
   (a) $\hat{A} := (z_i^{j-1})_{j,i=1}^3 \in \mathbb{R}^{3\times 3}$
   (b) $w := \hat{A}^{-1} \cdot (1, b_1, b_2)^\mathsf{T} \in \mathbb{R}^3$
   (c) Return $G := \sum_{i=1, z_i<z_f}^3 w_i$
6. Else:
   (a) $z := \frac{(b_3-b_2)\cdot z_f + b_3 - b_4}{(b_2-b_1)\cdot z_f + b_2 - b_3}$
   (b) $z_1 := 0$, $z_2 := z$, $z_3 := z_f$, $z_4 := 1$
   (c) $\hat{A} := (z_i^{j-1})_{j,i=1}^4 \in \mathbb{R}^{4\times 4}$
   (d) $w = \hat{A}^{-1} \cdot (1, b_1, b_2, b_3)^\mathsf{T}$
   (e) Return $G := \sum_{i=1, z_i<z_f}^4 w_i$

By Proposition 8 there exists a unique canonical representation $S \in \mathfrak{S}(b)$ with support at z_f and $\varepsilon(S) \in \{5, 6\}$. If this distribution has exactly three points of support, the correct $G(b, z_f)$ is returned in Step 5c by Proposition 3.

If S has four points of support, let $w \in \mathbb{R}^4$ and $z_1, \ldots, z_4 \in [0, 1]$ such that $S = \sum_{i=1}^4 w_i \cdot \delta_{z_i} \in \mathfrak{S}(b)$. Due to $\varepsilon(S) \leq 6$ we know that $\{0, 1\} \subset \{z_1, \ldots, z_4\}$. By Proposition 11 Algorithm 4 computes the last remaining point of support z correctly.

Then $z \in [0, 1]$ and the algorithm cannot have returned in Step 5c because otherwise the unique S would have three points of support (note that the weights generated by Step 5b have to be positive due to Proposition 10). As in the proof of Proposition 3 w is uniquely determined by $\hat{A} \cdot w = (1, b_1, b_2, b_3)^\mathsf{T}$ and thus Algorithm 4 computes it correctly. Then the correct $G(b, z_f)$ is returned in Step 6e. □

Algorithm 4 is slightly slower than Algorithm 2 but also computes a slightly sharper bound by including the knowledge that valid solutions have all their support in the interval $[0, 1]$. In practice, this leads to a darkening of shadows cast over very short ranges. Additionally, it has the advantage that the failure case for $c_n = 0$ is removed.

However, there is also a drawback when 16-bit quantization is used. Quantization artifacts mostly effect shadows cast over a very short range. With Algorithm 4 these artifacts tend to be stronger than with Algorithm 2. Therefore, we believe that Hamburger four moment shadow mapping is overall preferable.

7.3 Moment Problems in Degenerate Cases

In both presented algorithms we have not handled the case where B is not positive-definite. Our reasoning for this design decision in the paper builds upon Proposition 4.

Proof of Proposition 4. The matrix B is symmetric by definition. To see that it is also positive semi-definite we take an arbitrary vector $u \in \mathbb{R}^n$ and reuse the representation of B through $\hat{\mathbf{b}}$ from the proof of Proposition 10:

$$u^\mathsf{T} \cdot B \cdot u = u^\mathsf{T} \cdot \mathbb{E}_Z(\hat{\mathbf{b}} \cdot \hat{\mathbf{b}}^\mathsf{T}) \cdot u = \mathbb{E}_Z(u^\mathsf{T} \cdot \hat{\mathbf{b}} \cdot \hat{\mathbf{b}}^\mathsf{T} \cdot u) = \mathbb{E}_Z((u^\mathsf{T} \cdot \hat{\mathbf{b}})^2) \geq 0$$

"1. ⇒ 3.": Let $\det B = 0$. Then we can choose a $c \in \ker B$ with $c \neq 0$. In analogy to the previous computation we obtain:

$$0 = c^\mathsf{T} \cdot B \cdot c = \mathbb{E}_Z((c^\mathsf{T} \cdot \hat{\mathbf{b}})^2)$$

The expression $c^\mathsf{T} \cdot \hat{\mathbf{b}}$ is a polynomial of degree $n - 1 = \frac{m}{2}$ and thus it can have no more than $\frac{m}{2}$ roots. Z must have all of its support at these roots because although $(c^\mathsf{T} \cdot \hat{\mathbf{b}})^2$ is never negative the expectation $\mathbb{E}_Z((c^\mathsf{T} \cdot \hat{\mathbf{b}})^2)$ vanishes.

"3. ⇒ 1.": Let $z_1, \ldots, z_{\frac{m}{2}} \in \mathbb{R}$ and $w \in \mathbb{R}^{\frac{m}{2}}$ such that $Z = \sum_{i=1}^{\frac{m}{2}} w_i \cdot \delta_{z_i}$. Then

$$B = \mathbb{E}_Z(\hat{\mathbf{b}} \cdot \hat{\mathbf{b}}^\mathsf{T}) = \sum_{i=1}^{\frac{m}{2}} w_i \cdot \hat{\mathbf{b}}(z_i) \cdot \hat{\mathbf{b}}^\mathsf{T}(z_i).$$

Each of the matrices $\hat{\mathbf{b}}(z_i) \cdot \hat{\mathbf{b}}^\mathsf{T}(z_i)$ has rank one. Thus, B cannot have rank greater than $\frac{m}{2}$, i.e. it is singular.

"1. ⇒ 2.": For c as above the roots of $c^\mathsf{T} \cdot \hat{\mathbf{b}}$ uniquely determine at most $\frac{m}{2}$ possible locations for the points of support of Z. Forming a square Vandermonde matrix from the distinct roots we can uniquely determine the weights to be used at these roots. Thus $Z \in \mathfrak{S}(b)$ is unique.

"2. ⇒ 1.": Suppose $\det B \neq 0$. Then B is positive definite and Algorithm 2 can produce multiple different distributions in $\mathfrak{S}(b)$. Contradiction. □

The proof of this Proposition immediately induces an algorithm for perfect reconstruction in the degenerate case as stated in the paper. The vector $c \neq 0$ has to be chosen in the kernel of B. The roots of $c^\mathsf{T} \cdot \hat{\mathbf{b}}$ become the points of support of the constructed distribution and the weights can be computed in the usual way. While this is an effective algorithm, its small field of application to nothing but the perfectly singular case makes it useless for the present problem.

7.4 The Trigonometric Moment Problem

The trigonometric moment problem is of interest to us due to its excellent error in the automated evaluation. To see how well it performs in practice a closed-form solution is needed. For the sake of this paper we have developed such a solution.

The difficulty in doing so is that the Markov-Krein theorem does not apply. Using notions of Krein the theorem applies to functions $\mathbf{b}$ inducing M_+-systems [Kreĭn and Nudel'man 1977, p. 125]. This means that the function

$$c_0 + \sum_{j=1}^{m'} c_j \cdot \mathbf{b}_j$$

must never have more than m' roots for all $m' \in \{0, \ldots, m\}$ and non-zero choices of $c_0, \ldots, c_{m'}$. Obviously, this holds for polynomials, that is $\mathbf{b}(z) = (z^j)_{j=1}^m$. However, for

$$\mathbf{b}(z) := (\cos(1\cdot 2\pi z), \sin(1\cdot 2\pi z), \ldots, \cos(m\cdot 2\pi z), \sin(m\cdot 2\pi z))$$

such functions can have $2 \cdot \left\lceil \frac{m'}{2} \right\rceil$ roots. Whenever a cosine is added the number of possible roots increases by two. For example, $\cos(1 \cdot 2\pi z)$ has two roots on $[0, 1)$, not one. Therefore, the Markov-Krein theorem cannot be applied.

Experiments with Algorithm 1 reveal that solutions do indeed have a different structure. A full proof of this structure is beyond the scope of the present work and our derivation of the solution to the trigonometric moment problem is not rigorous. Still, the presented closed-form produces results which agree with those of Algorithm 1 in all tested cases.

Although they do not solve the problem immediately, canonical representations still play a crucial role. Once more they can be constructed by means of a special matrix. To define it we switch to a complex setting interpreting the pair $\cos(j \cdot 2\pi z), \sin(j \cdot 2\pi z)$ as complex number $\exp(i \cdot j \cdot 2\pi z) \in \mathbb{C}$. Besides, it is convenient to augment $\mathbf{b}$ with a constant function and to use zero-based indices. The Hankel matrix is replaced by a so-called Toeplitz matrix.

Definition 13. Let

$$\begin{aligned} \mathbf{b} : [0,1) &\to \mathbb{C}^{m+1} \\ z &\mapsto (\exp(i \cdot j \cdot 2\pi z))_{j=0}^{m}. \end{aligned}$$

Let $Z \in \mathfrak{P}([0, 1))$ be a distribution. Then we refer to $b := \mathbb{E}_Z(\mathbf{b})$ as *moment vector* and to

$$B := \mathbb{E}_Z(\mathbf{b} \cdot \mathbf{b}^*)$$

as associated *Toeplitz matrix* ($\mathbf{b}^*$ denotes the conjugate transpose). This matrix has constant diagonals and is Hermitian. Its entries can be written as

$$B_{j,k} = \begin{cases} \overline{b_{k-j}} & \text{if } k \geq j \\ b_{j-k} & \text{if } k < j. \end{cases}$$

The Toeplitz matrix behaves much like the Hankel matrix. For example, it is positive semi-definite.

Proposition 14. *B is positive semi-definite.*

Proof. For all $u \in \mathbb{C}^m$:

$$\begin{aligned} u^* \cdot B \cdot u = u^* \cdot \mathbb{E}_Z(\mathbf{b} \cdot \mathbf{b}^*) \cdot u &= \mathbb{E}_Z(u^* \cdot \mathbf{b} \cdot \mathbf{b}^* \cdot u) \\ &= \mathbb{E}_Z(u^* \cdot \mathbf{b} \cdot \overline{(u^* \cdot \mathbf{b})}) = \mathbb{E}_Z(|u^* \cdot \mathbf{b}|^2) \geq 0 \end{aligned}$$

□

It can also be used to test for existence of canonical representations with prescribed points of support.

Proposition 15. *Let $B \in \mathbb{C}^{(m+1)\times(m+1)}$ be positive-definite and let $z_f \in [0, 1)$. Then there exists exactly one canonical representation $S \in \mathfrak{S}(b)$ with support at z_f.*

Proof. See [Kreĭn and Nudel'man 1977, p. 149, Theorem IV.8.3]. □

Furthermore, the Toeplitz matrix allows for computation of canonical representations in much the same way as in Algorithm 2.

Proposition 16. *Let $z_0, \ldots, z_m \in [0, 1)$ be pairwise different, let $w \in \mathbb{R}^{m+1}$ and let $S := \sum_{l=0}^{m} w_l \cdot \delta_{z_l} \in \mathfrak{S}(b)$. Let B be positive-definite. Then for all $l \in \{1, \ldots, m\}$*

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

The weights are given by

$$w_l = \frac{1}{\mathbf{b}^*(z_l) \cdot B^{-1} \cdot \mathbf{b}(z_l)}$$

for all $l \in \{0, \ldots, m\}$.

Proof. Let

$$\hat{A} := (\mathbf{b}(z_0), \ldots, \mathbf{b}(z_m)) \in \mathbb{C}^{(m+1)\times(m+1)}.$$

This matrix is a Vandermonde matrix formed by $\exp(i \cdot 2\pi z_0), \ldots, \exp(i \cdot 2\pi z_m)$ and as such it is invertible. We obtain:

$$\begin{aligned} \hat{A}^{-1} \cdot B \cdot \hat{A}^{-\mathsf{T}} &= \hat{A}^{-1} \cdot \mathbb{E}_Z(\mathbf{b} \cdot \mathbf{b}^{\mathsf{T}}) \cdot \hat{A}^{-\mathsf{T}} \\ &= \sum_{l=0}^{m} w_l \cdot \hat{A}^{-1} \cdot \mathbf{b}(z_l) \cdot \mathbf{b}^{\mathsf{T}}(z_l) \cdot \hat{A}^{-\mathsf{T}} \\ &= \sum_{l=0}^{m} w_l \cdot e_l \cdot e_l^{\mathsf{T}} = \operatorname{diag}(w_0, \ldots, w_m) \end{aligned}$$

Now $\mathbf{b}^*(z_l) \cdot B^{-1} \cdot \mathbf{b}(z_0) = 0$ is an off-diagonal entry of the diagonal matrix $\hat{A}^{\mathsf{T}} \cdot B^{-1} \cdot \hat{A}$. For all $l \in \{0, \ldots, m\}$ the diagonal entries are

$$\frac{1}{w_l} = \mathbf{b}^*(z_l) \cdot B^{-1} \cdot \mathbf{b}(z_l).$$

□

Canonical representations help us in the solution of Problem 1 due to the following conjecture. This conjecture has been developed based upon observations on the output of Algorithm 1. We believe that one of the proof techniques which serves to prove the Markov-Krein theorem can also be transferred to this conjecture but do not present such a proof here.

Conjecture 17. *Let $z_f \in [0, 1]$ and let B be positive-definite. There exists a w_3 with*

$$0 \leq w_3 \leq \frac{1}{\mathbf{b}^*(1) \cdot B^{-1} \cdot \mathbf{b}(1)}$$

such that $b - w_3 \cdot \mathbf{b}(1)$ admits a canonical representation $S \in \mathfrak{S}(b - w_3 \cdot \mathbf{b}(1))$ with support at z_f and $S + w_3 \cdot \delta_1$ solves Problem 1, i.e. $G(b, z_f) = F_{S+w_3\cdot\delta_1}(z_f)$.

This means that distributions solving the trigonometric moment problem can always have support at one in addition to the support provided by a canonical representation. This behavior is intuitive in the sense that one is no different from z_f. It bounds the interval where the support has to be optimized.

The remaining problem is an optimization problem where a closed-form solution is challenging. We have to find w_3 minimizing $F_{S+w_3\cdot\delta_1}(z_f)$ with S depending upon w_3. We have been unable to find a solution for arbitrary m, but we have found a solution for $m = 2$ which is the choice discussed in the paper.

In this case, S has 3 points of support with one of them being z_f. Let

$$0 \leq z_0 < z_1 < 1, \quad z_2 := z_f, \quad z_3 := 1, \quad w_0, w_1, w_2 > 0$$

such that $S = \sum_{l=0}^{3} w_l \cdot \delta_{z_l}$ is the sought-after canonical representation of $b - w_3 \cdot \mathbf{b}(1)$. The size we need to minimize is

$$F_{S+w_3\cdot\delta_1}(z_f) = \sum_{l=0,\, z_i<z_f}^{3} w_l = \begin{cases} 0 & \text{if } z_f \leq z_0 \\ w_0 & \text{if } z_0 < z_f \leq z_1 \\ w_0 + w_1 & \text{if } z_1 < z_f. \end{cases}$$

The case $z_f \leq z_0$ is trivial. A more difficult case arises when w_0 and w_1 both contribute, i.e. $z_1 < z_f$. In this case

$$\begin{aligned} &F_{S+w_3\cdot\delta_1}(z_f) = 1 - w_2 - w_3 \\ =&1 - \frac{1}{\mathbf{b}(z_f)^* \cdot (B - w_3 \cdot \mathbf{b}(1) \cdot \mathbf{b}^*(1))^{-1} \cdot \mathbf{b}(z_f)} - w_3. \end{aligned} \quad (3)$$

The most problematic part of this expression is the inverse matrix. Fortunately, this term can be simplified. It is a linear combination of no more than two different matrices which do not depend upon w_3. More precisely:

$$\begin{aligned} &(B - w_3 \cdot \mathbf{b}(1) \cdot \mathbf{b}^*(1))^{-1} \\ =&B^{-1} + \frac{w_3 \cdot B^{-1} \cdot \mathbf{b}(1) \cdot \mathbf{b}^*(1) \cdot B^{-1}}{1 - w_3 \cdot \mathbf{b}^*(1) \cdot B^{-1} \cdot \mathbf{b}(1)} \end{aligned} \quad (4)$$

This can be verified directly by taking the product of the matrix and its claimed inverse.

Substitution into Equation (3) yields:

$$\begin{aligned} &F_{S+w_3\cdot\delta_1}(z_f) \\ =&1 - \frac{1}{\mathbf{b}(z_f)^* \cdot B^{-1} \cdot \mathbf{b}(z_f) + w_3 \cdot \frac{|\mathbf{b}(z_f)^* \cdot B^{-1} \cdot \mathbf{b}(1)|^2}{1 - w_3 \cdot \mathbf{b}^*(1) \cdot B^{-1} \cdot \mathbf{b}(1)}} - w_3 \end{aligned}$$

Now the dependence of this term upon w_3 is simple enough to compute critical points. We obtain two solutions for w_3:

$$\frac{\pm\left|\mathbf{b}^*(1) \cdot B^{-1} \cdot \mathbf{b}(z_f)\right| - \mathbf{b}^*(z_f) \cdot B^{-1} \cdot \mathbf{b}(z_f)}{\left|\mathbf{b}^*(1) \cdot B^{-1} \cdot \mathbf{b}(z_f)\right|^2 - \mathbf{b}^*(1) \cdot B^{-1} \cdot \mathbf{b}(1) \cdot \mathbf{b}^*(z_f) \cdot B^{-1} \cdot \mathbf{b}(z_f)}$$

It remains to treat the case $z_0 < z_f \leq z_1$, i.e. $F_{S+w_3\cdot\delta_1}(z_f) = w_0$. To this end, we ask for the value of w_0 as function of the unknown z_0. We note that $S + w_3 \cdot \delta_1 - w_0 \cdot \delta_{z_0}$ is a canonical representation of $b - w_0 \cdot \mathbf{b}(z_0)$. Then Proposition 16 implies

$$\mathbf{b}^*(z_f) \cdot (B - w_0 \cdot \mathbf{b}^*(z_0) \cdot \mathbf{b}(z_0))^{-1} \cdot \mathbf{b}(1) = 0.$$

Equation 4 applies analogously to this case and we obtain:

$$\mathbf{b}^*(z_f) \cdot \left(B^{-1} + \frac{w_0 \cdot B^{-1} \cdot \mathbf{b}(z_0) \cdot \mathbf{b}^*(z_0) \cdot B^{-1}}{1 - w_0 \cdot \mathbf{b}^*(z_0) \cdot B^{-1} \cdot \mathbf{b}(z_0)}\right) \cdot \mathbf{b}(1) = 0$$

Solving for w_0 yields Equation (5) in Table 1 (it is too lengthy to fit into a single column). Now we can consider the partial derivative

$$\frac{\partial}{\partial z_0} \frac{1}{w_0}$$

and look for critical points. The corresponding equation depends upon $\exp(-2 \cdot 2\pi z_0), \ldots, \exp(2 \cdot 2\pi z_0)$. Substituting $x_0 \in \mathbb{C}$ for $\exp(2\pi z_0)$ and multiplying by x_0^2 we obtain a quartic equation in x_0. Its roots can be computed in closed-form. Once z_0 is known, w_0 can be computed from Equation (5) and the remaining points of support can be constructed as canonical representation.

Thus, we are now able to compute the optimal distribution in all relevant cases. The only thing that remains to be done is to distinguish between these cases. We have found that this can be done by considering the canonical representation having support at one. Its three points of support partition $[0, 1]$ into three intervals. On the first one $G(b, z_f)$ is zero, on the second one it is w_0 and on the third one it is $1 - w_2 - w_3$.

8 Additional Notes on Moment Quantization

The paper demonstrates that the proposed quantization for four moment shadow mapping increases entropy of the stored data by 12.3 bits. Proposition 6 is crucial for this result because it states the differential entropy of the transformed random variable.

Proof of Proposition 6. The function

$$\begin{aligned} p_{\theta_m(b)} : \mathbb{R}^m &\to \mathbb{R}_{\geq 0} \\ b' &\mapsto \frac{p_b(\theta_m^{-1}(b'))}{|\det \theta_m|} \end{aligned}$$

is the probability density function of $\theta_m \circ \mathbf{x}_b$ because for all random variables $\mathbf{x}$ on $\mathbb{R}^m$:

$$\begin{aligned} \mathbb{E}_{\theta_m \circ \mathbf{x}_b}(\mathbf{x}) = \mathbb{E}_{\mathbf{x}_b}(\mathbf{x} \circ \theta_m) &= \int_{\mathbb{R}^m} \mathbf{x}(\theta_m(b)) \cdot p_b(b) \,\mathrm{d}\, b \\ &= \int_{\mathbb{R}^m} \mathbf{x}(b') \cdot p_b(\theta_m^{-1}(b')) \cdot |\det \theta_m^{-1}| \,\mathrm{d}\, b' \\ &= \int_{\mathbb{R}^m} \mathbf{x}(b') \cdot p_{\theta_m(b)}(b') \,\mathrm{d}\, b' \end{aligned}$$

Using $p_{\theta_m(b)}$ we can compute the differential entropy of $\theta_m \circ \mathbf{x}_b$:

$$\begin{aligned} h(\theta_m \circ \mathbf{x}_b) &= -\int_{\mathbb{R}^m} \frac{p_b(\theta_m^{-1}(b'))}{|\det \theta_m|} \cdot \log_2 \frac{p_b(\theta_m^{-1}(b'))}{|\det \theta_m|} \,\mathrm{d}\, b' \\ &= -\int_{\mathbb{R}^m} p_b(b) \cdot (\log_2 p_b(b) - \log_2 |\det \theta_m|) \,\mathrm{d}\, b \\ &= h(\mathbf{x}_b) + \log_2 |\det \theta_m| \end{aligned}$$

□

The optimized moment quantization exploits the fact that all valid moment vectors are convex combinations of vectors generated by $\mathbf{b}$, i.e. they lie in $\operatorname{conv} \mathbf{b}([0, 1])$. Canonical quantization wastes memory on encoding this fact separately for every single texel. More precisely:

Proposition 18. *If $\mathbf{x}_b$ is a random variable on $R \subset \mathbb{R}^m$,*

$$h(\mathbf{x}_b) \leq \log_2 \operatorname{vol} R.$$

Proof. Without loss of generality $p_b(b) \neq 0$ for all $b \in R$. Otherwise we replace R by $R \cap p_b^{-1}(\mathbb{R}_+)$ and obtain a sharper bound. By Jensen's inequality the following holds [Cover and Thomas 2001, p. 25]:

$$\begin{aligned} h(\mathbf{x}_b) &= \mathbb{E}_{\mathbf{x}_b}(-\log_2 p_b) = \mathbb{E}_{\mathbf{x}_b}\left(\log_2 \frac{1}{p_b}\right) \\ &\leq \log_2 \mathbb{E}_{\mathbf{x}_b}\left(\frac{1}{p_b}\right) = \log_2 \left(\int_R \frac{p_b(b)}{p_b(b)} \,\mathrm{d}\, b\right) = \log_2 \operatorname{vol} R \end{aligned}$$

□

Knowing the volume of $\operatorname{conv} \mathbf{b}([0, 1])$ allows for a quantitative statement on the amount of entropy $h(\mathbf{x}_b)$ that is wasted by canonical quantization.

Proposition 19. *For $m \in \mathbb{N}$ and $\mathbf{b}(z) = (z^j)_{j=1}^m$*

$$\operatorname{vol} \operatorname{conv} \mathbf{b}([0, 1]) = \prod_{j=1}^m \frac{((j-1)!)^2}{(2 \cdot j - 1)!}.$$

$$w_0 = \frac{\mathbf{b}^*(z_f) \cdot B^{-1} \cdot \mathbf{b}(1)}{\mathbf{b}^*(z_f) \cdot B^{-1} \cdot \mathbf{b}(1) \cdot \mathbf{b}^*(z_0) \cdot B^{-1} \cdot \mathbf{b}(z_0) - \mathbf{b}^*(z_f) \cdot B^{-1} \cdot \mathbf{b}(z_0) \cdot \mathbf{b}^*(z_0) \cdot B^{-1} \cdot \mathbf{b}(1)} \tag{5}$$

Table 1: *The formula for the weight at the first point of support as function of the location of this point.*

Proof. An outline of the proof can be found in [Karlin and Shapley 1953, p. 51, Theorem 15.2]. ☐

Combining both leads to $h(\mathbf{x}_b) \leq -14.6$ for $m = 4$, i.e. canonical 16-bit quantization uses at most 49.4 of the available 64 bits efficiently. The other 14.6 bits are wasted on encoding the fact that $b \in \operatorname{conv} \mathbf{b}([0,1])$. The transform θ_m improves on this situation by expanding $\operatorname{conv} \theta_m \circ \mathbf{b}$ and only 2.3 bits of wasted entropy remain. Unless our optimization has failed to find the global maximum affine linear transforms cannot produce a better result.

9 Translation and Scale Invariance

In the paper we have claimed that users need not worry about choosing the near and far planes of the shadow map tightly if they use Hamburger MSM (disregarding quantization artifacts). Even more interestingly this property is unique for Hamburger MSM.

This statement comes from the observation that the information conveyed by a choice of $\mathbf{b}$ is characterized by the function space spanned by its component functions among with the constant one function. The shadow map provides us with $b = \mathbb{E}_Z(\mathbf{b})$ and we certainly know $\mathbb{E}_Z(1) = 1$. For a different choice $\tilde{\mathbf{b}}$ spanning the same function space every component function can be written as linear combination of the component functions of $\mathbf{b}$ among with the constant one function. The same linear combinations serve to transform b and 1 into $\mathbb{E}_Z(\tilde{\mathbf{b}})$.

Changing the near and far plane of the shadow map is equivalent to scaling and shifting the domain of depth values. To achieve that this does not affect the information held by the shadow map, $\mathbf{b}$ has to be chosen such that the corresponding function space is invariant under scaling and shifting. This is achieved by polynomials and nothing else:

Proposition 20. *Let $\mathcal{V}$ be a finite-dimensional vector space of smooth functions $f : \mathbb{R} \to \mathbb{R}$. $\mathcal{V}$ is the vector space of all polynomials up to degree $\dim \mathcal{V} - 1$ if and only if the function*

$$z \mapsto f(x \cdot z + y)$$

is still in $\mathcal{V}$ for all $f \in \mathcal{V}$ and all $x, y \in \mathbb{R}$.

Proof. "⇒" If f is a polynomial of degree $\dim \mathcal{V} - 1$,

$$z \mapsto f(x \cdot z + y)$$

is still a polynomial of degree $\dim \mathcal{V} - 1$ for all $x, y \in \mathbb{R}$.

"⇐" Let $z \mapsto f(x \cdot z + y) \in \mathcal{V}$ for all $f \in \mathcal{V}$ and $x, y \in \mathbb{R}$. Since $f \in \mathcal{V}$ is smooth we can consider its derivative at $z \in \mathbb{R}$:

$$f'(z) = \lim_{h \to 0} \frac{f(z+h) - f(z)}{h}$$

The function $z \mapsto f(z+h)$ is still in $\mathcal{V}$ and so is the entire quotient. Then the limit f' is also in $\mathcal{V}$ because $\mathcal{V}$ is a finite-dimensional, real vector space. Thus, the differential is an endomorphism on $\mathcal{V}$

$$\begin{aligned} \mathrm{d} : \mathcal{V} &\to \mathcal{V} \\ f &\mapsto f'. \end{aligned}$$

Now let $\lambda \in \mathbb{C}$ be an eigenvalue of d for the eigenvector $f \in \mathcal{V} \setminus \{0\}$, i.e. $\mathrm{d}\, f = \lambda \cdot f$. Then we know for all $x, y, z \in \mathbb{R}$

$$\lambda \cdot f(x \cdot z + y) = \mathrm{d}\, f(x \cdot z + y) = \lambda \cdot x \cdot f(x \cdot z + y).$$

It follows that $\lambda = \lambda \cdot x$ and thus $\lambda = 0$.

Therefore, d has solely vanishing eigenvalues, i.e. it is nilpotent on $\mathcal{V}$. In particular, $\mathrm{d}^{\dim \mathcal{V}}\, \mathcal{V} = \{0\}$. This can only be true if all functions in $\mathcal{V}$ are polynomials of degree $\dim \mathcal{V} - 1$ or less. Then for dimensionality reasons, $\mathcal{V}$ has to be the unique vector space containing all these polynomials. ☐

10 Code Listings

Implementing four moment shadow mapping requires little modification of existing renderers using shadow mapping especially if another technique using filterable shadow maps is implemented already. The procedure consists of two major steps. First the shadow map is generated. The output color of the pixel shader should be computed using Listing 1.

Once the shadow map has been generated linear filtering may be applied, e.g. a two-pass Gaussian blur, generation of mipmaps or a resolve operation for shadow maps with multisample antialiasing. In the latter case a worthwhile optimization is to use Listing 1 during the resolve [Lauritzen et al. 2011]. This way, the multisampled shadow map requires only a depth channel and bandwidth requirements are reduced. Though, our implementation does not make use of this optimization.

In the second step fragments in the scene are shaded using the provided moment shadow map. Listing 2 demonstrates how to obtain a common moment vector from the moment shadow map with optimized quantization. This vector can then be passed to the function in Listing 3 or Listing 4 to obtain the shadow intensity for a fragment using Hamburger 4MSM or Hausdorff 4MSM, respectively. The code for Hausdorff 4MSM adds an additional branch to the algorithm. This darkens short-range shadows slightly at the cost of a minor increase in run time.

Additional Acknowledgments

We would like to thank Sebastian Merzbach, Oliver Burghard and Jonathan Klein for proofreading and helpful feedback. Furthermore, we thank the BlendSwap.com users Greg Zaal, Chris Kuhn, Oscar Baechler, Paulo Bardes and Zuck as well as the operators of the Stanford 3D scanning repository for providing additional models used in the demo, the evaluation and the supplementary video.

References

AKHIEZER, N. I., AND KREĬN, M. G. 1962. *Some Questions in the Theory of Moments*, vol. 2 of *Translations of Mathematical Monographs*. American Mathematical Society.

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

KARLIN, S., AND SHAPLEY, L. S. 1953. Geometry of moment spaces. *Memoirs of the American Mathematical Society*, 12.

Listing 1: *An HLSL function constructing a vector which encodes four moments in a way that is optimized for little information loss during quantization. It should be used to construct a four moment shadow map.*
`Out4MomentsOptimized` *is set to a vector where every entry lies between zero and one. It should be the output of the pixel shader generating the four moment shadow map.*
`FragmentDepth` *is the linear depth in shadow map space of the fragment that is currently being rendered to the moment shadow map. It is supposed to lie between zero and one.*

```
void Get4MomentsOptimized(out float4 Out4MomentsOptimized,float FragmentDepth){
   float Square=FragmentDepth*FragmentDepth;
   float4 Moments=float4(FragmentDepth,Square,Square*FragmentDepth,Square*Square);
   Out4MomentsOptimized=mul(Moments,float4x4(
                -2.07224649f,    13.7948857237f,   0.105877704f,    9.7924062118f,
                32.23703778f,   -59.4683975703f,  -1.9077466311f,-33.7652110555f,
               -68.571074599f,   82.0359750338f,   9.3496555107f,  47.9456096605f,
                39.3703274134f,-35.364903257f,    -6.6543490743f,-23.9728048165f));
   Out4MomentsOptimized[0]+=0.035955884801f;
}
```

Listing 2: *An HLSL function sampling a moment vector from a four moment shadow map with optimized quantization.*
`Out4Moments` *is set to the vector of moments. It starts with the first moment and ends with the fourth.*
`ShadowMapSampler` *is the sampler to be used. It is recommended to use anisotropic filtering.*
`Optimized4MomentShadowMap` *is the four-channel texture containing the moment vectors with optimized quantization. Typically, this is a render target whose pixel values originate from the function in Listing 1. It should use a format with 16 bit unsigned integers mapped to the interval from zero to one.*
`ShadowMapTexCoord` *is the texture coordinate at which the sample should be taken. Typically, this is the texture coordinate in the shadow map of a fragment that is to be shaded.*

```
void SampleOptimized4MomentShadowMap(out float4 Out4Moments,
    SamplerState ShadowMapSampler,Texture2D Optimized4MomentShadowMap,
   float2 ShadowMapTexCoord)
{
   float4 _4MomentsOptimized=Optimized4MomentShadowMap.Sample(
      ShadowMapSampler,ShadowMapTexCoord
   );
   _4MomentsOptimized[0]-=0.035955884801f;
   Out4Moments=mul(_4MomentsOptimized,
      float4x4(0.2227744146f, 0.1549679261f, 0.1451988946f, 0.163127443f,
               0.0771972861f,0.1394629426f,0.2120202157f,0.2591432266f,
               0.7926986636f, 0.7963415838f, 0.7258694464f, 0.6539092497f,
               0.0319417555f,-0.1722823173f,-0.2758014811f,-0.3376131734f));
}
```

Listing 3: *An implementation of Hamburger 4MSM (Algorithm 3) in HLSL.*
`OutShadowIntensity` *is set to the computed shadow intensity (one for fully shadowed areas).*
`_4Moments` *is the moment vector obtained using Listing 2.*
`FragmentDepth` *is the depth in shadow map space of the fragment that is to be shaded.*
`DepthBias` *is a small positive constant used to avoid wrong self-shadowing.*
`MomentBias` *is a small positive constant (e.g.* $3 \cdot 10^{-5}$*) used to compensate quantization errors (see Section 4.2).*

```
void GetHamburger4MSMShadowIntensity(out float OutShadowIntensity,
    float4 _4Moments, float FragmentDepth, float DepthBias, float MomentBias)
{
   // Bias input data to avoid artifacts
   float4 b=lerp(_4Moments,float4(0.5f,0.5f,0.5f,0.5f),MomentBias);
   float3 z;
   z[0]=FragmentDepth-DepthBias;
   // Compute a Cholesky factorization of the Hankel matrix B storing only non-
   // trivial entries or related products
   float L32D22=mad(-b[0],b[1],b[2]);
   float D22=mad(-b[0],b[0],b[1]);
   float SquaredDepthVariance=mad(-b[1],b[1],b[3]);
   float D33D22=dot(float2(SquaredDepthVariance,-L32D22),float2(D22,L32D22));
   float InvD22=1.0f/D22;
   float L32=L32D22*InvD22;
   // Obtain a scaled inverse image of bz=(1,z[0],z[0]*z[0])^T
   float3 c=float3(1.0f,z[0],z[0]*z[0]);
   // Forward substitution to solve L*c1=bz
   c[1]-=b.x;
   c[2]-=b.y+L32*c[1];
   // Scaling to solve D*c2=c1
   c[1]*=InvD22;
   c[2]*=D22/D33D22;
   // Backward substitution to solve L^T*c3=c2
   c[1]-=L32*c[2];
   c[0]-=dot(c.yz,b.xy);
   // Solve the quadratic equation c[0]+c[1]*z+c[2]*z^2 to obtain solutions z[1]
   // and z[2]
   float p=c[1]/c[2];
   float q=c[0]/c[2];
   float D=((p*p)/4.0f)-q;
   float r=sqrt(D);
   z[1]=-(p/2.0f)-r;
   z[2]=-(p/2.0f)+r;
   // Construct a solution composed of three Dirac-deltas and evaluate its CDF
   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]));
   OutShadowIntensity=saturate(Switch[2]+Switch[3]*Quotient);
}
```

Listing 4: *An implementation of Hausdorff 4MSM (Algorithm 4) in HLSL. Parameters are the same as in Listing 3.*

```
void GetHausdorff4MSMShadowIntensity(out float OutShadowIntensity,
    float4 _4Moments, float FragmentDepth, float DepthBias, float MomentBias)
{
   // Bias input data to avoid artifacts
   float4 b=lerp(_4Moments,float4(0.5f,0.5f,0.5f,0.5f),MomentBias);
   float3 z;
   z[0]=FragmentDepth-DepthBias;
   // Compute a Cholesky factorization of the Hankel matrix B storing only non-
   // trivial entries or related products
   float L32D22=mad(-b[0],b[1],b[2]);
   float D22=mad(-b[0],b[0],b[1]);
   float SquaredDepthVariance=mad(-b[1],b[1],b[3]);
   float D33D22=dot(float2(SquaredDepthVariance,-L32D22),float2(D22,L32D22));
   float InvD22=1.0f/D22;
   float L32=L32D22*InvD22;
   // Obtain a scaled inverse image of bz=(1,z[0],z[0]*z[0])^T
   float3 c=float3(1.0f,z[0],z[0]*z[0]);
   // Forward substitution to solve L*c1=bz
   c[1]-=b.x;
   c[2]-=b.y+L32*c[1];
   // Scaling to solve D*c2=c1
   c[1]*=InvD22;
   c[2]*=D22/D33D22;
   // Backward substitution to solve L^T*c3=c2
   c[1]-=L32*c[2];
   c[0]-=dot(c.yz,b.xy);
   // Solve the quadratic equation c[0]+c[1]*z+c[2]*z^2 to obtain solutions z[1]
   // and z[2]
   float p=c[1]/c[2];
   float q=c[0]/c[2];
   float D=((p*p)/4.0f)-q;
   float r=sqrt(D);
   z[1]=-(p/2.0f)-r;
   z[2]=-(p/2.0f)+r;
   // Use a solution made of four deltas if the solution with three deltas is invalid
   if(z[1]<0.0f || z[2]>1.0f){
      float zFree=((b[2]-b[1])*z[0]+b[2]-b[3])/((b[1]-b[0])*z[0]+b[1]-b[2]);
      float w1Factor=(z[0]>zFree)?1.0f:0.0f;
      OutShadowIntensity=(b[1]-b[0]+(b[2]-b[0]-(zFree+1.0f)*(b[1]-b[0]))
         *(zFree-w1Factor-z[0])/(z[0]*(z[0]-zFree)))/(zFree-w1Factor)+1.0f-b[0];
   }
   // Use the solution with three deltas
   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]));
      OutShadowIntensity=Switch[2]+Switch[3]*Quotient;
   }
   OutShadowIntensity=saturate(OutShadowIntensity);
}
```

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

LAURITZEN, A., SALVI, M., AND LEFOHN, A. 2011. Sample distribution shadow maps. In *Proceedings of the 2011 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D '11, 97–102.

PETERS, C. 2013. *Moment Shadow Mapping*. Master's thesis, University of Bonn.

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