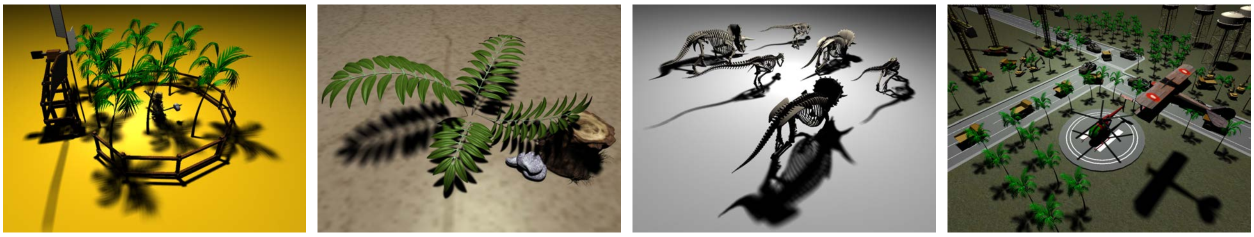
DOI: 10.1111/cgf.12156
Eurographics Symposium on Rendering 2013
Nicolas Holzschuch and Szymon Rusinkiewicz
(Guest Editors)

Volume 32 (2013), Number 4

Exponential Soft Shadow Mapping

Li Shen[1] Jieqing Feng[1,†] Baoguang Yang[2]

1 - State Key Lab of CAD&CG, Zhejiang University ({shenli, jqfeng}@cad.zju.edu.cn) † - Corresponding author
2 - Qualcomm (baoguang@qti.qualcomm.com)



(a) 117fps, 512 × 512 SM (b) 85fps, 1024 × 1024 SM (c) 79fps, 1024 × 1024 SM (d) 15fps, 4096 × 4096 SM

Figure 1: *Soft shadows generated by ESSM, tested with different scenes.*

Abstract

In this paper we present an image-based algorithm to render visually plausible anti-aliased soft shadows in real time. Our technique employs a new shadow pre-filtering method based on an extended exponential shadow mapping theory. The algorithm achieves faithful contact shadows by adopting an optimal approximation to exponential shadow reconstruction function. Benefiting from a novel overflow free summed area table tile grid data structure, numerical stability is guaranteed and error filtering response is avoided. By integrating an adaptive anisotropic filtering method, the proposed algorithm can produce high quality smooth shadows both in large penumbra areas and in high frequency sharp transitions, meanwhile guarantee cheap memory consumption and high performance.

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

1. Introduction

Soft shadow is an important component in realistic image synthesis. Real time applications, such as 3D computer games and virtual reality, usually require high quality soft shadows. However, rendering high quality soft shadows efficiently is still a challenging problem. The soft shadow value for a screen pixel is the visibility of the extended light source viewed from the point in the scene corresponding to the pixel. This visibility test is computationally intensive. Image-based approach based on shadow map [Wil78] and its extensions scale well with the scene complexity, which gradually become the main stream solution in industry. Among them, Percentage Closer Filtering (PCF) [RSC87] is the most prevalent one owing to its simplicity and efficiency. PCF smoothes the shadow edges by filtering the binary shadow test results in a given kernel. As a result, it can produce soft transition along hard shadow boundary. Recently, the extension of PCF, Percentage-Closer Soft Shadows (PCSS) [Fer05], is widely adopted to generate visually plausible soft shadows efficiently, and can be easily integrated into existing rendering systems. PCSS assumes that the occluders/receivers are planar and parallel to the light source. For each pixel to be rendered, PCSS first computes the average blocker depth by averaging the depth values of the texels in a searching kernel which are smaller than the depth of the current pixel in light space. Then the penumbra size is estimated by the average blocker depth. Finally, PCF is applied by looping all the sampling points in the penumbra.

As the complexity of both the average blocker depth evaluation and soft shadowing steps is proportional to the penumbra size, rendering efficiency measured by frame rates in general may drop dramatically due to massive texture samples when the light source is large. Furthermore, image based approaches like PCSS suffer from severe aliasing problems. There are mainly two kinds of aliasing phenomenon. When the shadow map resolution is inadequate, these algorithms tend to produce jagged edges. When the screen resolution is low, Moiré patterns can appear on high frequency shadows. We denote the two kinds of soft shadow aliasing as "shadow-map aliasing" and "screen-space aliasing" respectively in this paper.

Exponential Shadow Mapping (ESM) [AMS*08] [Sal08] is proposed to perform hard shadow anti-aliasing efficiently, which adopts the exponential function to approximate the binary shadow test function and applied pre-filtering to gain significant speed up. The ESM theory is simple and easy to implement. Its cheap memory cost and computation overhead guarantees its high performance. Unfortunately, the exponential function is a "single-bounded" function. It diverges exponentially on the left side, thus leads to incorrectly-lit artifacts due to the "non-planarity" problem. The definition of "single-bounded" function and "non-planarity" problem is proposed and investigated by Yang et al. [YDF*10]. The former indicates a function which only bound one side of the shadow test function and the latter indicates the problem that adopting a "single-bounded" function as the pre-filtering shadow function may produce incorrect results for large partially shadowed kernels. Moreover, few research focuses on how to determine the value of ESM function parameter c, which may affect the steepness of the approximated function. An inappropriate c may lead to the loss of contact shadow artifact. Furthermore, extending ESM theory to soft shadow framework is still an open problem.

To generate penumbra with arbitrary width in constant time, the summed-area table(SAT) [Cro84] is widely adopted as a pre-filtering function due to its satisfied filtering quality and efficiency [ADM*08] [LAU07] [YDF*10]. However, SAT is only suitable for simple scenes rendered by a low resolution shadow map, because its construction overhead and precision loss increase dramatically when the shadow map resolution becomes large. In order to maintain performance and avoid error shadow test results caused by precision instability of SAT, current pre-filtered soft shadow techniques can only adopt a low resolution shadow map. Hence the "shadow-map aliasing" may occur. Furthermore, SAT only supports rectangular filter kernels, which is incompatible with anisotropic filtering. Consequently, the "screen-space aliasing" is still a problem to be solved.

In this paper, We present Exponential Soft Shadow Mapping (ESSM), a PCSS based pre-filtering soft shadow technique, which employs the exponential function as shadow reconstruction function, for the sake of high quality pre-filtering and cheap space/time overhead. The main contributions of the proposed algorithm are:

1. A new formula based on ESM theory is given to estimate the average blocker depth in PCSS framework, thus ESM theory is extended to soft shadow rendering.
2. Faithful contact shadow is obtained by adopting the optimal ESM function parameter and effective depth range in light space.
3. Incorrectly-lit artifacts caused by non-planarity problem are alleviated by micro-subdivision.
4. An adaptive overflow free SAT tile grid data structure for pre-filtering is designed, which can significantly reduce both the SAT precision loss and SAT building computation cost. Thereby it overcomes the resolution restriction of pre-filtering soft shadow techniques when using SAT.
5. By introducing an adaptive approximation scheme to an anti-aliased ellipsoid soft shadow filtering kernel, efficient anisotropic shadow filtering is applied on SAT.

The experimental results show that the proposed algorithm can render both simple game-like scenes at very high frame rates and complex large scale scenes efficiently in real-time. The produced visually plausible soft shadows are smooth and well anti-aliased.

2. Related work

A complete review of shadow algorithms is beyond the scope of this paper, readers can refer to the surveys of Woo et al. [WPF90], Hasenfratz et al. [HLHS03], Scherzer et al. [SWP10], and Eisemann et al. [EAS*12] for a detailed overview. This section focuses on the most related real time image-based soft shadow methods.

2.1. Soft Shadow Mapping via Back-projection

The algorithms proposed by Atty et al. [AHL*06] and Guennebaud et al. [GBP06] rasterize the complex scene geometries into a shadow map and then perform back-projection to estimate the visibility. However light leaking and shadow over-estimation may occur, due to the piecewise constant approximation of the blocker geometry by the shadow map. Guennebaud et al. [GBP07] and Schwarz et al. [SS07] remove most of these artifacts efficiently. The former back-projects local contour edges detected by a 2D marching square algorithm. The latter calculates an occlusion bitmask that provides a discrete representation of the light source to determine which portion is occluded by the back-projected micro-quads. Unfortunately, both techniques increase the runtime and space complexity. Yang et al. [YFGL09] exploits the screen and light space coherence by a hierarchical shadow map data-structure and significantly improves the performance by a pixel-packet approach. However, the technique is still non-trivial and not efficient enough for real-time applications.

2.2. Pre-filtered Soft Shadow Mapping

Recently, soft shadow techniques based on PCSS framework are becoming increasingly popular for their efficiency and simplicity. They can produce visually plausible soft shadows with high performance, which is an indispensable requirement in 3D game engines. In general, the techniques pre-filter a certain shadow reconstruction function so that PCSS is performed in constant time consumption for arbitrary light source size.

Convolution soft shadow mapping (CSSM) [ADM*08] adopts Fourier basis to construct a "double-bounded" pre-filtering function which can cover the whole range of the binary shadow test function. CSSM can well fits PCSS framework. However it requires large amount of GPU memories to store the truncated Fourier series, which is too expensive for real-time applications.

SAT-based variance shadow mapping (SAVSM) [LAU07] adopts the one-tailed version of Chebyshev's inequality as shadow reconstruction function, but still performs brute-force sampling for average blocker depth evaluation. Then Variance soft shadow mapping (VSSM) [YDF*10] achieves constant time evaluation of average blocker depth and significantly reduces the memory cost compared to CSSM. Moreover, VSSM employs a kernel subdivision approach and successfully alleviates the incorrectly-lit artifacts caused by non-planarity problem. Unfortunately, the annoying light bleeding artifact inherited in variance shadow mapping theory limits its applications for complex scenes.

All of CSSM, SAVSM and VSSM adopt SAT as their pre-filtering solution for its superior filtering quality compared to other functions like mip-mapping. Although some improvements [HTG*05] [DL06] [LAU07] are proposed to alleviate the precision problem of SAT, the defects discussed in Sec.1 make existing pre-filtered soft shadow techniques less attractive in real time applications.

2.3. Soft Shadow anti-aliasing

When the size of the penumbra in screen space is smaller than a pixel, the soft shadow transition becomes hard. If the shadow signal and the sampling resolution do not satisfy the Nyquist criteria, aliasing may occur. To alleviate the aliasing artifacts, multi-sampling methods like brute-force super-sampling and sub-pixel multi-sampling [PWC*09] can be adopted. However, the huge computation overheads make them only suitable for interactive or off-line applications. Shen et al. [SGYF11] introduces an analytical anti-aliased soft shadow filtering kernel by performing an analytical integration over each pixel. They adopts a one-axis-aligned parallelogram to approximate the integrated kernel, which enables the algorithm to run in real-time. A brute-force sampling approach is applied to the approximated kernel to compute the anti-aliased shadow value. When the kernel size is large, the sampling process becomes very time-consuming, making the algorithm inefficient.

3. Algorithm Overview

An overview of our algorithm is described as follows:

Rendering ESSM basis: Firstly, we render a normal shadow map, and generate the exponential shadow map(esm) together with the **depth-scaled exponential shadow map**($esm-z$) afterwards using our optimal ESM function parameter and effective depth range in light space, which guarantees better contact shadow quality which will be discussed in Sec.7.

Constructing SAT tile grids: Secondly, we compute the **overflow-free metric** and adopt it as the guide to create **SAT tile grids** for esm and $esm-z$. The step is to solve SAT precision problem, and will be described in Sec.6.

Computing average blocker depth: For each visible surface point $\boldsymbol{x}$, we estimate its average blocker depth Z_{avg}, where the proposed **ESM-Z formula** is adopted, to estimate penumbra kernel w_p. It will be explained in detail in Sec.4.2. A **micro-subdivision**(Sec.5.2) method is applied to the initial filter kernel w_{avg} in advance to alleviate the incorrectly-lit artifacts caused by non-planarity problem.

Shadowing the pixel: Finally, a penumbra ellipsoid kernel w_{ap} for anisotropic anti-aliasing is computed for each screen pixel which will be adaptively decomposed into a series of rectangular sub-filter-kernels using the proposed **ellipsoid kernel approximation algorithm**, which is SAT-friendly. The step will be described in Sec.5.1. Every sub-filter-kernel is further checked if micro-subdivision is necessary to prevent non-planarity effect.

4. Exponential Soft Shadow Mapping

4.1. Review of Exponential Shadow Mapping

Let $\boldsymbol{x}$ be a surface point visible to the camera, d be the distance from $\boldsymbol{x}$ to the light source, and $z(\boldsymbol{p})$ be the depth of the occluder as seen from the light center, where $\boldsymbol{p}$ represents the position of the occluder on the shadow map plane. The shadow test for each visible point $\boldsymbol{x}$ is defined as:

$$s(\boldsymbol{x}) := f(d(\boldsymbol{x}), z(\boldsymbol{p}))$$

where $f(d,z)$ is a Heaviside step function, with a value of 0 if $d > z$ and 1 otherwise.

Exponential shadow mapping [AMS*08] [Sal08] approximates the Heaviside step function with the following exponential function (Fig.2):

$$\begin{aligned} f(d,z) &\approx e^{-c(d-z)} \\ &\approx e^{-cd}e^{cz}. \end{aligned}$$

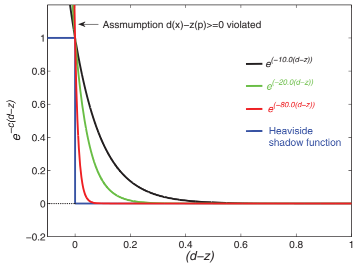


Figure 2: *The ESM shadow reconstruction functions. The blue line is the binary shadow test function (also can be expressed as Heaviside step function). d is the pixel depth in light space and z is the sampled depth value from shadow map. Obviously, the exponential function is left side unbounded. A larger constant factor c yields better approximation on the right side.*

Considering that the visibility factor s_f is calculated by filtering the shadow test function s, which can be written as a convolution [AMB*07], then the exponential approximation could be adopted to replace the shadow test function:

$$\begin{aligned} s_f(\boldsymbol{x}) &= [w * f(d(\boldsymbol{x}), z)](\boldsymbol{p}) \\ &\approx [w * e^{-cd(\boldsymbol{x})} e^{cz}](\boldsymbol{p}) \\ &\approx e^{-cd(\boldsymbol{x})} [w * e^{cz}](\boldsymbol{p}) \end{aligned}$$

where w is the soft shadow filter kernel of surface point $\boldsymbol{x}$.

Note that exponential approximation can be decomposed into two factors about d and z, consequently the exponent-transformed depth values e^{cz} can be pre-filtered to speed-up the shadow filtering operation.

Since the exponential function is single-bounded, the approximation on the left side will diverge exponentially when the assumption $d(\boldsymbol{x}) \geq z(\boldsymbol{p})$ is not satisfied, and the exponential function may return an arbitrarily large value, which will result in an erroneous filtering response.

4.2. Estimating Average Blocker Depth

The most time consuming step of integrating ESM to PCSS framework is estimating the average blocker depth. Brute-force sampling solutions [Fer05] [LAU07] are too expensive, therefore cannot achieve high performance when the search kernel is very large, which corresponds to large penumbra area in general.

Actually, the averaging step can also be formulated as a convolution and computed efficiently through pre-filtering. Assuming $\boldsymbol{x}$ is a visible surface point in the scene, its searching kernel w_{avg} for average blocker depth can be estimated by computing the intersecting between the shadow map plane and the frustum defined by $\boldsymbol{x}$ and the light source, then the local average depth Z_{avg} is:

$$Z_{avg}(\boldsymbol{x}) = [w_{avg} * z](\boldsymbol{p})$$

In fact, only the depth values which are smaller than $d(\boldsymbol{x})$ need to be averaged, their average is noted as Z_{occ}. Meanwhile, the average of the depth values which are greater than $d(\boldsymbol{x})$ is noted as Z_{unocc}, the following equation holds:

$$Z_{avg} = \frac{S_1}{S} Z_{unocc} + \frac{S_2}{S} Z_{occ}$$

where S is the number of samples in the filter kernel, S_1 and S_2 are the numbers of samples which are greater and smaller than $d(\boldsymbol{x})$ respectively. It is obvious that S_1/S and S_2/S correspond to shadow test results $s_f(\boldsymbol{x})$ and $1 - s_f(\boldsymbol{x})$ for the filter kernel w_{avg}. Therefore, the average blocker depth Z_{occ} can be re-written as:

$$Z_{occ} = (Z_{avg} - s_f(\boldsymbol{x}) Z_{unocc}) / (1.0 - s_f(\boldsymbol{x}))$$

It is straightforward to compute Z_{unocc} with the help of the shadow test function. We can just extract those samples with depth values greater than $d(\boldsymbol{x})$ inside the filter kernel w_{avg} by weighting them:

$$Z_{unocc} = \frac{[w_{avg} * [f(d(\boldsymbol{x}), z) z]](\boldsymbol{p})}{[w_{avg} * f(d(\boldsymbol{x}), z)](\boldsymbol{p})}$$

Now, we can combine the two equations above:

$$Z_{occ} = \frac{Z_{avg} - [w_{avg} * [f(d(\boldsymbol{x}), z) z]](\boldsymbol{p})}{1.0 - [w_{avg} * f(d(\boldsymbol{x}), z)](\boldsymbol{p})}$$

The denominator equals the complement of the shadow test function, i.e., $1 - s_f(\boldsymbol{x})$. For the numerator, Z_{avg} is known, the product of the shadow test function and z can be simply approximated as:

$$\begin{aligned} f(d, z) z &\approx e^{-c(d-z)} z \\ &\approx e^{-cd} e^{cz} z. \end{aligned}$$

Finally, we can approximate average blocker depth Z_{occ} as:

$$Z_{occ} = \frac{Z_{avg} - e^{-cd(\boldsymbol{x})} [w_{avg} * [e^{cz} z]](\boldsymbol{p})}{1.0 - s_f(\boldsymbol{x})}$$

The new $[e^{cz(\boldsymbol{p})} z(\boldsymbol{p})]$ basis can be computed and stored alongside the regular *esm* basis. The approach to compute average blocker depth is similar to CSSM [ADM*08]. Hence it is superior to VSSM [YDF*10], without the parallel receiver and occluder assumptions. We refer to this approach as ESM-Z. However due to the single-bounded feature of the ESM shadow test function, non-planarity problem also may occur in ESM-Z computation. In Sec. 5.2, we will discuss the non-planarity problem and its solution in detail.

5. High quality pre-filtering with anisotropic anti-aliasing

To achieve high quality soft shadows, we will remove or alleviate the artifacts caused by screen-space aliasing and non-planarity problem. The proposed filtering method is introduced in detail in this section.

5.1. Adaptive approximation of ellipsoid filter kernel

An analytical integration over each pixel area is performed to apply anisotropic anti-aliasing for soft shadows. Assuming the pixel footprint in screen space is a circular Gaussian, then its projection through receiver patch to shadow map plane is an ellipsoid Gaussian. It is also reasonable to assume that the soft shadow kernel is a circular Gaussian in order to benefit from the fact that the convolution of two Gaussians is still a Gaussian. The anti-aliased soft shadow kernel is decided finally by convolving the Gaussian soft shadow kernel with the ellipsoid Gaussian pixel footprint projected on the shadow map plane. Fig.3 gives an illustration of this convolution procedure.

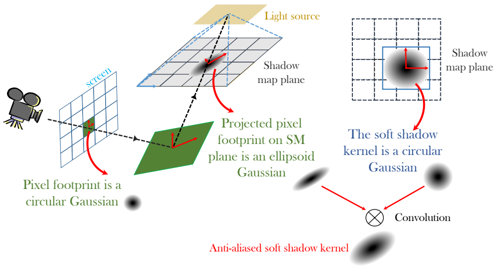

Light source
Shadow map plane
screen
Shadow map plane
Projected pixel footprint on SM plane is an ellipsoid Gaussian
The soft shadow kernel is a circular Gaussian
Pixel footprint is a circular Gaussian
Convolution
Anti-aliased soft shadow kernel

Figure 3: *Illustration of the convolution between the soft shadow filter and the projected pixel footprint on shadow map plane.*

Different from the anti-aliasing solution proposed by Shen et al. [SGYF11], an ellipsoid is computed to approximate the anti-aliased soft shadow kernel, and is further approximated adaptively by a series of rectangular filter kernels. This approximation can guarantee that the sub-filter-kernels are SAT-friendly. All of the rectangular sub-filter-kernels are evaluated and then summed up.

The ellipsoid kernel is decomposed uniformly along the vertical axis, which can facilitate the shader code implementation. The number of rectangles, namely N, can be determined by the size of the ellipsoid and a pre-defined parameter H, i.e., the height of the rectangles, for the sake of efficiency and quality trade-off. Assuming the ellipsoid covers Y texels along the vertical-axis of the shadow map, then $N = ceil(Y/H)$. A smaller H achieves better approximation and anti-aliasing quality, while a larger H guarantees fewer SAT look-ups. The width of each rectangular filter kernel is determined by the distance between the two intersection points of the rectangle's horizontal center line and the ellipsoid. We denote this approximation scheme as $Level0$ rasterization.

To achieve better anti-aliasing quality, we further perform an adaptive approximation to the ellipsoid kernel. Firstly, we uniformly decompose the ellipsoid along the vertical axis into N rectangles. For each rectangle with vertical range $[y^i_{min}, y^i_{max}]$, we then compute the intersection points of horizontal line $y = y^i_{min}$, $y = y^i_{max}$ and the ellipsoid kernel respectively. The top and the bottom rectangles both have 3 intersection points, and they are processed by the $Level0$ rasterization as described above. For those rectangles with 4 intersection points, an inscribed quadrilateral is estimated inside the ellipsoid. Then a $Level0$ rectangle can be further subdivided into 3 rectangles: one inscribed quadrilateral whose 4 vertexes all fall inside the ellipsoid and two rectangles which have 3 vertices inside, 1 outside. The filter defined on the inscribed quadrilateral can be evaluated straightforwardly by SAT, and the other two rectangles need to be further rasterized. We divide each of the two rectangles into four smaller rectangles. Two of them are completely outside and inside the ellipsoid respectively which can be evaluated trivially. The other two rectangles across the ellipsoid can be further rasterized similarly, until the user defined rasterization level is reached. When the rasterization stops, the rectangle filters partially covered by the ellipsoid are looked up and weighted by 0.5, because the area inside the ellipsoid is nearly half of the corresponding rectangle. We denote the adaptive approximation method $LevelX$ rasterization, X is the rasterization level. Figure.4 illustrate the $Level0$, $Level1$, and $Level2$ kernel rasterization scheme.

It is worth noting that the analytical integration is only required in the soft shadow estimation stage, the initial kernel in average blocker depth evaluation can be approximated by a rectangular kernel immediately.

5.2. Micro-subdivision for rectangle filters

As mentioned in Sec.4.1, the ESM shadow test function only works correctly under the assumption $\Delta_x = d(\boldsymbol{x}) - z(\boldsymbol{p}) \geq 0$. However the z values in the filter kernel will not be necessarily smaller than a given pixel depth $d(\boldsymbol{x})$. As a result, an erroneous result may be generated due to the divergence on the left side of the exponential function. Although the artifacts may be invisible in unoccluded regions because the overflow can be easily clamped to 1.0, they will occur in partially occluded regions where part of the z values inside the filter kernel is smaller than the pixel depth $d(\boldsymbol{x})$, i.e., incorrectly-lit soft shadows. The artifact is named as "non-planarity" problem and the filter kernels suffering from this problem are called "non-planarity" kernels by Yang et al. [YDF*10].

Obviously, the rectangular filter kernels obtained in the previous ellipsoid approximation step may be non-planarity

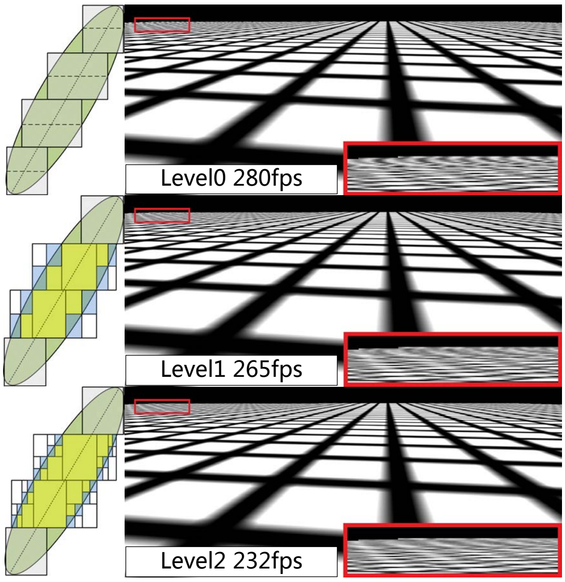


Figure 4: *There kinds of rasterization scheme. The white rectangles are outside the ellipsoid which will be discarded immediately, the yellow rectangles lie inside the ellipsoid and the blue rectangles are nearly half covered by the ellipsoid, so weighted by 0.5. The anti-aliasing quality is improved while the rasterization level increases. Here we use $H = 4$ to render the three images, the shadow map resolution is* 512×512.

kernels. Furthermore, the average blocker depth formula according to ESM shadow test function proposed in Sec.4.2 also suffers from this non-planarity problem, thus the average blocker depth evaluated should be revised.

Annen et al. [AMS*08] return back to PCF in these non-planarity regions, which will lead to expensive overhead in large soft shadow kernels. Inspired by the kernel subdivision approach in VSSM [YDF*10], we perform micro-subdivision after ellipsoid kernel approximation, which can effectively remove the artifacts caused by single-bounded shadow test functions in the average blocker depth evaluation and soft shadow computation.

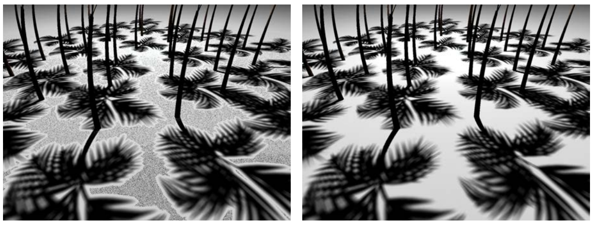
Each current filter kernel w is subdivided into n equal-sized sub-kernels $\{w_i\}_{i=1}^{n}$ uniformly. The sub-kernels can be classified into two types: the non-planarity kernels and the normal kernels. For the normal sub-kernels, the ESM shadow and ESM-Z will be evaluated straightforwardly. For the non-planarity sub-kernels, where the assumption $\Delta_x \geq 0$ is violated, a cheap 2×2 PCF is performed inside these kernels to estimate ESM shadow and ESM-Z. Assuming there are m texels inside each sub-kernel, m_i^{occ} of them have a depth value smaller than the current pixel depth $d(\boldsymbol{x})$, the corresponding average blocker depth and shadow value are d_i and s_i respectively, then average blocker depth d and the shadow value s_f over the whole kernel w can be calculated as: $d = \sum_{i=1}^{n}(d_i m_i^{occ}) / \sum_{i=1}^{n} m_i^{occ}, s_f = \sum_{i=1}^{n} s_i/n$ Considering that $m_i^{occ} = m(1 - s_i)$, then the average blocker depth d is:

$$d = \frac{1}{n(1-s_f)} \sum_{i=1}^{n} (d_i(1-s_i))$$

We adopt the threshold classification approach proposed by Annen et.al [AMS*08] to determine if a sub-kernel w_i is a non-planarity kernel, by checking whether the ESM shadow value of the current sub-kernel exceeds $1 + \varepsilon$ or not, where ε is a user defined threshold. This method requires no more additional resources and can produce high quality soft shadows efficiently in our experiments.

6. Adaptive overflow-free SAT tile grids

The precision loss problem in SAT may generate artifacts when the soft shadow kernel is relatively small, which corresponds to the case where the receiver is very close to the occluder. In this case, massive noise will occur due to wrong SAT filtering response caused by precision loss (Fig.5(a)). Moreover, when the shadow map resolution increases, the situation become more serious due to numerical overflow. Meanwhile, the overhead of SAT construction increases significantly and gradually becomes the bottleneck of the algorithm.



(a) Original SAT (b) SAT tile grids

Figure 5: *Filtering quality comparison between the original SAT and adaptive SAT tile grids. (a): Noise introduced by SAT precision loss. (b): Noise is removed due to precision preserved by our SAT tile grids.*

For an IEEE standard 754 floating point SAT, assuming the filtering kernel size is $width \times height$, then $log_2(width \times height)$ bits are consumed in the worst case, and only $23 - log_2(width \times height)$ bits are left for the data itself. Integer SAT [LAU07] can save 8 bits more for precision. Moreover, its wraparound feature automatically handles the numerical overflow problem. But as the soft shadow filter kernels may not be necessarily aligned with texel centers, bilinear sampling is obliged to achieve smooth shadow transition. Consequently, the integer values fetched from an integer SAT are converted to floating point values for bilinear interpolation and the wraparound feature is disabled. Thus numerical overflow problem is exposed and brings artifacts like noise into generated soft shadows.

6.1. Construction of SAT tile grids

An overflow free SAT tile grid construction algorithm is proposed to address the precision loss issue and accelerates SAT construction procedure. Since precision tends to be burnt and the probability of triggering numerical overflow increases while the size of SAT grows, a high resolution SAT can be divided into a low resolution SAT tile grid to avoid these problems. The precision loss in each tile can be controlled by the user and fewer bits are required for accumulation. If the corresponding tile grid is free of overflow, the precision stability of an arbitrarily high resolution SAT only depends on its tiles, which is below the user defined threshold. The overhead of introducing such an SAT tile grid is additional lookups, especially for the filter kernels which cover several SAT tiles. In this case, the filter kernel should be divided along the tile boundaries. Fortunately, this overhead is much slighter compared to the rapid rising construction cost when SAT resolution is increasing in most of the scenes, We will give the performance statics of the SAT tile grids in Sec.8.

It is natural to divide a square SAT uniformly into an SAT tile grid, each tile has the same resolution. The key to this approach is to determine the number of tiles in the grid. We adopt the 32-bit unsigned integer format proposed in [LAU07] for SAT tile grid generation. Assume a user defined precision threshold μ is given, i.e., preserving μ bits for precision of the integer values. Then $32-\mu$ bits are left for accumulation, and the value stored in the right-bottom corner of each tile should not exceed 2^{32}. This upper bound provides the principle of SAT tile grid generation.

It is easy to obtain the number of tiles with the help of a mip-mapped ESSM basis texture. First, the image pyramid is generated by hardware immediately after the float *esm* & $esm-z$ texture is rendered. Then we perform a comparison between the texel value in the pyramid and the threshold $2^{32-\mu-2\times Miplevel}$ ($Miplevel = 0$ is the most detailed level) in a top-down way, until we find a level that all the value of its texels are smaller than the threshold. The number of SAT tiles equals to the resolution of this level, and the integer SAT tile grid can be built afterwards.

We adopt the parallel recursive doubling SAT construction method proposed in [HTG*05]. The SAT is generated in two phases: first a horizontal phase, then a vertical phase. Assuming an original SAT with resolution $N \times N$ is divided into an $M \times M$ SAT tile grid, the resolution of each tile is N/M. If we read X pixels per-fragment, the original SAT generation requires a total of $2 \times ceil(log_2 N/log_2 X)$ passes for the two rendering phases, and the SAT tile grid only requires $2 \times ceil(log_2(N/M)/log_2 X)$ passes. So a $log_2 N/log_2(N/M)$ accelerating ratio is achieved in the SAT construction step. The acceleration is remarkable especially for large resolution shadow maps, which is a significant improvement in pre-filtered soft shadow techniques.

6.2. SAT tile grids for ESSM

Since the mainstream shading languages e.g. HLSL, GLSL, do not support 3-channel textures, we suggest to adopt a 2-channel 32-bit integer texture to store *esm* and $esm-z$ and a 1-channel integer texture to store the light space linear depth z. Compared to a 4-channel single texture approach, the proposed method consumes less memory and is more efficient.

Considering that the integer SAT solution [LAU07] requires a normalized value as input, we apply an apriori mapping from the *esm* and $esm-z$ basis term e^{cz}, ze^{cz} to $[0,1]$. The mapping can be written as: $esm = (e^{cz} - e^{cz_{min}})/(e^{cz_{max}} - e^{cz_{min}}), esm - z = (ze^{cz} - z_{min}e^{cz_{min}})/(z_{max}e^{cz_{max}} - z_{min}e^{cz_{min}})$ where z_{min}, z_{max} are the minimum and maximum linear depth of the scene respectively in light space, *esm* & $esm-z$ are normalized ESSM basis. z_{min} and z_{max} can be estimated by transforming the scene bounding box to light space, or precisely computed by a hierarchical shadow map which can be generated by hardware after the normal shadow map is rendered. The choice between the two ways is a trade-off between efficiency and accuracy. Finally, the ESSM basis *esm* & $esm-z$ will be converted to integer values by multiplying the user defined threshold 2^{μ} to construct SAT tile grids.

7. ESM parameter and contact shadow quality

From Fig. 2 we can see that the exponential function approximates the Heaviside step function well when the shadow receiver is far away from the shadow caster. However when the shadow receiver depth value d approaches the shadow caster depth value z, the ESM shadow value $e^{-c(d-z)}$ increases, which deviates from the correct shadow test result 0. This approximation error may lead to the loss of contact shadow, i.e., light leaking in umbra where the shadow receiver and shadow caster are very close.

A trivial solution to alleviate this artifact is to increase the value of parameter c, and generate a steeper exponential function. However, c can not be arbitrarily large, because there is an upper bound to prevent numerical overflow. As described above, the depth range between shadow receiver and occluder, i.e., $(d-z)$, may also influence the approximated shadow test result.

We adopt linear depth rather than the commonly used nonlinear depth to fully exploit the relationship between ESM shadow test result and light space effective depth range. Assuming that the light space depth of a given pixel is d_l and the light space occluder depth stored in the shadow map is z_l, then the ESM shadow value will be calculated after a linear depth normalization:

$$e^{-c(\frac{d_l-z_n}{z_f-z_n}-\frac{z_l-z_n}{z_f-z_n})} = e^{-\frac{c}{z_f-z_n}(d_l-z_l)}$$

where z_n and z_f are the minimum and maximum light space effective depth respectively. Obviously, $(d_l - z_l)$ is determined by the input object, thus a larger c and a smaller $z_f - z_n$ can lead to better approximation.

In the proposed ESSM framework, three terms are influenced by parameter c during ESM shadow and ESM-Z estimation, i.e., e^{-cd}, e^{cz} and $e^{cz}z$. When c is very large, numerical overflow may occur in the latter two terms, when e^{cz} is greater than the upper bound of the floating point values which can be represented in an exponential form: e^{R}. In addition, the light space effective depth range for linear depth normalization has a minimal value which is determined by the scene light space depth range z_{min} & z_{max}. Combining these two critical parameters, we get an optimal shadow test function approximation:

$$f(d,z) \approx e^{-\frac{R}{z_{max}-z_{min}}(d_l-z_l)}$$

here z_{min} & z_{max} can be determined in the same way as we discussed in Sec.6.2.

In this way, satisfied contact shadow quality is obtained without any performance drop.

8. Implementation Results and Discussion

Our experiments are implemented on a PC equipped with a quad-core @2.67GHz Intel i5 CPU, an NVIDIA Geforce GTX465 GPU, and 4GB physical memory. The screen resolution is 1280 × 960. All the soft shadows in the scenes are generated by a rectangular light source under the configuration: $\mu = 20$ for generating SAT tile grids, 3×3 micro-subdivision, and $H = 5$ $Level0$ rasterization. Light space depth range z_{min} & z_{max} are estimated by a hierarchical shadow map.

Fig.1 shows the ESSM algorithm quality in scenes with different complexity: Ogre (135k faces) in Fig.1(a), Plants (141k faces) in Fig.1(b), Dinosaurs (869k faces) in Fig.1(c) and Traffic (1285k faces) in Fig.1(d). Table1 provides a detailed performance statics in ESSM pipeline. The rows record runtime for: generating G-buffer data (GB), rendering ESSM basis textures and their pyramid (EB), generating SAT tile grids (STG), and shadowing pass (Shadow). The last column is the total runtime of rendering these scenes. We can see from the data that STG generating and soft shadowing are the most time consuming steps of ESSM pipeline. Table2 provides runtime of the two time-consuming steps and number of rendering passes required by the original SAT and our SAT tile grid for different shadow map resolution in the Traffic scene (Fig.1(d)).

Table 1: *Detailed runtime statics (milliseconds) of rendering scenes in Fig.1*

Scene	GB	EB	STG	Shadow	Total
Ogre	2.10	0.21	1.90	4.13	8.34
Plants	2.32	0.56	4.43	4.62	11.93
Dinosaurs	5.25	0.52	4.22	3.51	13.50
Traffic	16.4	9.6	41.3	10.4	77.7

Table 2: *SAT/Shadow runtime (milliseconds) and the rendering passes required by the Traffic scene (Fig.1(d)) with different shadow map resolution*

SMRes	512	1024	2048	4096
SAT/Shadow	1.8/4.0	6.8/4.2	28.2/5.7	121.1/8.6
STG/Shadow	1.5/4.1	4.5/4.6	18.3/6.2	41.3/10.4
SAT/STG passes	6/4	8/4	8/4	8/4

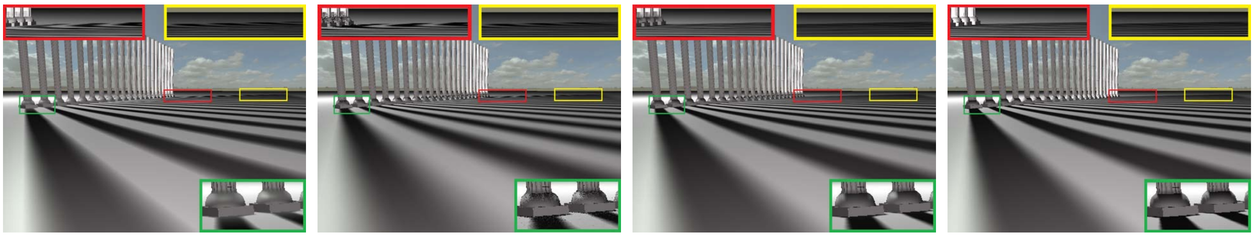
8.1. Comparison with previous pre-filtered soft shadow techniques

As a PCSS based pre-filtered soft shadowing technique, ESSM achieves step forward both in shadow quality and space-time efficiency. In this section, we will perform detailed comparison with the two most important pre-filtered soft shadow techniques: CSSM [ADM*08] and VSSM [YDF*10].

Shadow quality: Fig.6 compares the shadow quality rendered by CSSM, VSSM and ESSM. The shadow map resolution adopted in this scene is 512 × 512. CSSM renders 4 Fourier basis to approximate the shadow test function. VSSM adopts 5 × 5 uniform kernel subdivision in both the average blocker depth estimating and shadowing steps. All three techniques can generate high-quality smooth foreground penumbra, but notable aliasing artifacts occur in far-away high frequency shadows produced by CSSM and VSSM (red and yellow rectangles in Fig.6(a) and Fig.6(b)), because they are incapable of performing anisotropic filtering on SAT.

Combining the optimal shadow test function approximation with the precision stability provided by our advanced SAT tile grid scheme, ESSM can generate high quality contact shadow. VSSM introduces obvious noise in the contact shadow due to SAT precision loss (green rectangle in Fig.6(b)), and Yang et al. [YDF*10] return back to PCF in these regions, which may cause shadow discontinuity and requires much more texture samples. CSSM suffers from the loss of contact shadow artifact(green rectangle in Fig.6(a)), and requires expensive memory costs to save Fourier basis to remove this artifact, which will lead to significant decrease in performance.

Furthermore, ESSM is robust about scene depth complexity due to ESM shadow reconstruction theory. Annoying



(a) 48fps, CSSM (b) 118fps, VSSM (c) 101fps, ESSM (d) 43fps, Analytical Filtering

Figure 6: *Shadow quality comparison among CSSM, VSSM, ESSM and Shen et al. [SGYF11]'s analytical filtering algorithm. Obvious aliasing artifacts can be captured in the far-away hard shadow transition (red and yellow rectangles) in CSSM(a) and VSSM(b). CSSM suffers from the loss of contact shadow artifacts (green rectangles in (a)). VSSM introduces noise in contact shadows (green rectangles in (b)). The shadow quality of ESSM algorithm (c) is almost the same as Shen et al. [SGYF11]'s algorithm (d), but the performance is much higher.*

light leaking artifacts in VSSM (Fig.7(a)) will never occur in ESSM.

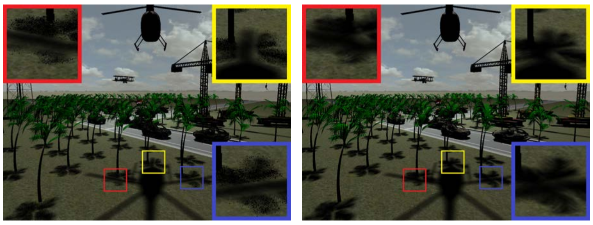
Space-time efficiency: In the aspect of performance, the frame rates of ESSM and VSSM are in the same order of magnitudes which outperform CSSM owing to their less GPU memory storage and texture fetch. Although Shen et al.'s analytical filtering algorithm [SGYF11] can also render very high quality soft shadows, their brute-force sampling strategy is awkward compared to SAT pre-filtering, which leads to low performance (Fig.6).

As traditional SAT construction is expensive and numerically instable for high resolution shadow maps, it becomes the bottleneck of VSSM and CSSM algorithms, which makes them impractical to render large scale scenes. Assume that we render a 4096×4096 shadow map, using 4 terms and 128-bit 4-channel textures for CSSM, 64-bit 2-channel texture for VSSM. CSSM requires more than 1 Gigabyte GPU memory, which reaches or even exceeds the capability of most main stream GPUs. We further compare VSSM and ESSM in the complex large scale Traffic scene(Fig.1(d)). As shown in Fig.7(a), we can find that VSSM introduces more noise spreading all over the screen and suffers from light bleeding artifacts due to the depth complexity in this scene. Compared to the performance drop caused by SAT construction in VSSM, ESSM benefits from our SAT tile grids and performs better, meanwhile can guarantee shadow quality.

From the statics in Table2, we can see that when the shadow map resolution increases, the SAT tile grids reach a good compromise of SAT construction and shadow computation. It brings much better performance compared to the original SAT construction scheme.

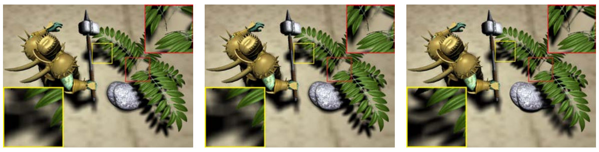
8.2. Limitations

The proposed algorithm is based on PCSS framework, thus inherits its limitations. The planar assumption of PCSS may lead to umbra underestimation because multiple blockers' depth are averaged (yellow square in Fig.8). Moreover, if occluders are close to each other or the light source is relatively far away from the scene, objects actually outside the frustum formed by the current surface point $\boldsymbol{x}$ and the light



(a) 10.5fps VSSM (b) 16.2fps ESSM

Figure 7: *A different view of the complex large scale Traffic scene, generated by VSSM and ESSM with a* 4096×4096 *shadow map. Obvious light leaking and noise artifacts appear in VSSM.*



(a) 98fps, ESSM (b) 26fps, PCSS (c) Reference

Figure 8: *Artifacts exhibited in ESSM and PCSS compared to the ground-truth. The reference image is rendered by* 32×32 *point-sample area light source. Shadow map resolution is* 512×512 *for all three images. ESSM is much faster than brute-force PCSS, while the soft shadow quality produced is almost the same.*

source may be wrongly considered as occluders, as long as their depth values are recorded inside the initial search kernel w_{avg} and are closer to the light source than $\boldsymbol{x}$. This problem may lead to incorrect visibility computation and brings shadow blurring and deblurring artifacts in dynamic scenes (red square in Fig.8, contact shadows are over-blurring or even lost).

9. Conclusions and Future Works

In this paper, we present an Exponential Soft Shadow mapping algorithm (ESSM) for rendering high quality anti-aliased visually plausible soft shadow efficiently. ESSM extends the ESM theory based on the framework of percentage-closer soft shadows by introducing a novel formula to estimate the average blocker depth for each visible point in constant time. The algorithm alleviates the loss of contact shadow artifacts and non-planarity problem caused by ESM shadow test function. Efficient anisotropic anti-aliasing is also integrated, and applied on a novel overflow free SAT tile grids data structure which alleviates artifacts brought by precision loss, meanwhile can accelerate SAT building procedure. As future work, we would like to investigate the possibility of automatically determining parameters for SAT tile grids construction, anisotropic filter approximation and kernel micro-subdivision according to scene geometry.

10. Acknowledgments

We would like to thank Dr. Hongzhi Wu for his valuable comments and suggestions. This work is supported by the National Natural Science Foundation of China under Grant Nos.60933007, 61170138 and 61173128, the 973 program of China under Grant No.2009CB320800, the Program for New Century Excellent Talents in University under Grant No.NCET-10-0728.

References

[ADM*08] ANNEN T., DONG Z., MERTENS T., BEKAERT P., SEIDEL H.-P., KAUTZ J.: Real-time, all-frequency shadows in dynamic scenes. *ACM Transaction on Graphics 27*, 3 (2008), 1–8. 2, 3, 4, 8

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

[AMB*07] ANNEN T., MERTENS T., BEKAERT P., SEIDEL H.-P., KAUTZ J.: Convolution shadow maps. In *Proceedings of Eurographics Symposium on Rendering* (2007), pp. 51–60. 4

[AMS*08] ANNEN T., MERTENS T., SEIDEL H.-P., FLERACKERS E., KAUTZ J.: Exponential shadow maps. In *Proceedings of graphics interface 2008* (2008), pp. 155–161. 2, 3, 6

[Cro84] CROW F. C.: Summed-area tables for texture mapping. *SIGGRAPH Computer Graphics 18*, 3 (1984), 207–212. 2

[DL06] DONNELLY W., LAURITZEN A.: Variance shadow maps. In *Symposium on Interactive 3D graphics and games* (2006), pp. 161–165. 3

[EAS*12] EISEMANN E., ASSARSSON U., SCHWARZ M., VALIENT M., WIMMER M.: Efficient real-time shadows. In *ACM SIGGRAPH 2012 Courses* (New York, NY, USA, 2012), SIGGRAPH '12, ACM, pp. 18:1–18:53. 2

[Fer05] FERNANDO R.: Percentage-closer soft shadows. In *ACM SIGGRAPH 2005 Sketches* (New York, NY, USA, 2005), SIGGRAPH '05, ACM. 1, 4

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

[GBP07] GUENNEBAUD G., BARTHE L., PAULIN M.: High-Quality Adaptive Soft Shadow Mapping. *Computer Graphics Forum 26*, 3 (2007), 525–534. 2

[HLHS03] HASENFRATZ J.-M., LAPIERRE M., HOLZSCHUCH N., SILLION F.: A survey of real-time soft shadows algorithms. *Computer Graphics Forum 22*, 4 (dec 2003), 753–774. 2

[HTG*05] HENSLEY J., THORSTEN S., GREG C., MONTEK S., ANSELMO L.: Fast summed-area table generation and its applications. *Computer Graphics Forum 24*, 3 (2005), 547–555. 3, 7

[LAU07] LAURITZEN A.: *GPU Gems 3*. Adison-Wesley, 2007, ch. Summed-Area Variance Shadow Maps. 2, 3, 4, 6, 7

[PWC*09] PAN M., WANG R., CHEN W., ZHOU K., BAO H.: Fast, sub-pixel antialiased shadow maps. *Comput. Graph. Forum (Proceedings of Pacific Graphics 2009) 28*, 7 (2009). 3

[RSC87] REEVES W. T., SALESIN D. H., COOK R. L.: Rendering antialiased shadows with depth maps. In *Proceedings of ACM SIGGRAPH 87* (New York, NY, USA, 1987), ACM, pp. 283–291. 1

[Sal08] SALVI M.: *ShaderX 6.0 - Advanced Rendering Techniques*. Charles River Media, 2008, ch. Rendering filtered shadows with exponential shadow maps. 2, 3

[SGYF11] SHEN L., GUENNEBAUD G., YANG B., FENG J.: Predicted Virtual Soft Shadow Maps with High Quality Filtering. *Computer Graphics Forum 30*, 2 (2011), 493–502. 3, 5, 9

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

[SWP10] SCHERZER D., WIMMER M., PURGATHOFER W.: A survey of real-time hard shadow mapping methods. In *State of the Art Reports Eurographics* (2010). 2

[Wil78] WILLIAMS L.: Casting curved shadows on curved surfaces. *Proceedings of ACM SIGGRAPH 78 12*, 3 (1978), 270–274. 1

[WPF90] WOO A., POULIN P., FOURNIER A.: A survey of shadow algorithms. *IEEE Computer Graphics and Applications 10*, 6 (1990), 13–32. 2

[YDF*10] YANG B., DONG Z., FENG J., SEIDEL H.-P., KAUTZ J.: Variance soft shadow mapping. *Computer Graphics Forum 29*, 7 (2010), 2127–2134. 2, 3, 4, 5, 6, 8

[YFGL09] YANG B., FENG J., GUENNEBAUD G., LIU X.: Packet-based hierarchal soft shadow mapping. *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering 2009) 28*, 4 (2009), 1121–1130. 2