Predicted Virtual Soft Shadow Maps with High Quality Filtering

Shen Li, Gael Guennebaud, Baoguang Yang, Jieqing Feng

EUROGRAPHICS 2011 / M. Chen and O. Deussen
(Guest Editors)

Volume 30 (2011), Number 2

Predicted Virtual Soft Shadow Maps with High Quality Filtering

Li Shen[1] Gaël Guennebaud[2] Baoguang Yang[3] Jieqing Feng[1,†]

1 - State Key Lab of CAD&CG, Zhejiang University ({shenli, jqfeng}@cad.zju.edu.cn) † - Corresponding author
2 - INRIA, LaBRI, Bordeaux University (gael.guennebaud@inria.fr) 3 - Autodesk (baoguang.yang@autodesk.com)

Abstract

In this paper we present a novel image based algorithm to render visually plausible anti-aliased soft shadows in a robust and efficient manner. To achieve both high visual quality and high performance, it employs an accurate shadow map filtering method which guarantees smooth penumbrae and high quality anisotropic anti-aliasing of the sharp transitions. Unlike approaches based on pre-filtering approximations, our approach does not suffer from light bleeding or losing contact shadows. Discretization artefacts are avoided by creating virtual shadow maps on the fly according to a novel shadow map resolution prediction model. This model takes into account the screen space frequency of the penumbrae via a perceptual metric which has been directly established from an appropriate user study. Consequently, our algorithm always generates shadow maps with minimal resolutions enabling high performance while guarantying high quality. Thanks to this perceptual model, our algorithm can sometimes be faster at rendering soft shadows than hard shadows. It can render game-like scenes at very high frame rates, and extremely large and complex scenes such as CAD models at interactive rates. In addition, our algorithm is highly scalable, and the quality versus performance trade-off can be easily tweaked.

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

1. Introduction

Shadows has always been, and is still a hot topic in computer graphics. During the last decade, significant advances have been made in the efficient rendering of soft shadows to make them physically more accurate. In particular, we can distinguish object space methods which are based on the construction of penumbra volumes [AAM03, FBP08], and soft shadow mapping (SSM) techniques based on back-projections [AHL*06, GBP06]. However, only a few research on soft shadows have been devoted to the rendering of very complex scenes. Moreover, in many application domains, physical accuracy is of little interest compared to the visual quality and rendering speed. By visual quality we refer to the smoothness of the shadows, i.e., to the lack of discontinuity and/or aliasing both in space and time. For instance, real world complex scenes require good antialiasing algorithms to filter out distant high frequency shadows while producing smooth penumbrae on the foreground.

In spite of all their merits, object based methods highly depend on the scene complexity making pure image based methods more attractive. Unfortunately, these later methods in general are famous to exhibit severe discretization artefacts when the shadow map resolution is insufficient, that is very likely to be the case when rendering real world complex scenes. While this issue received a particular attention in the case of hard shadows, it remains a challenging and open problem in the context of soft shadows. Back-projection based SSM techniques also exhibit other concerns such as the need of a complex and expensive data structure [YFGL09] making them impracticable for high resolution shadow maps. By assuming a locally planar oc-

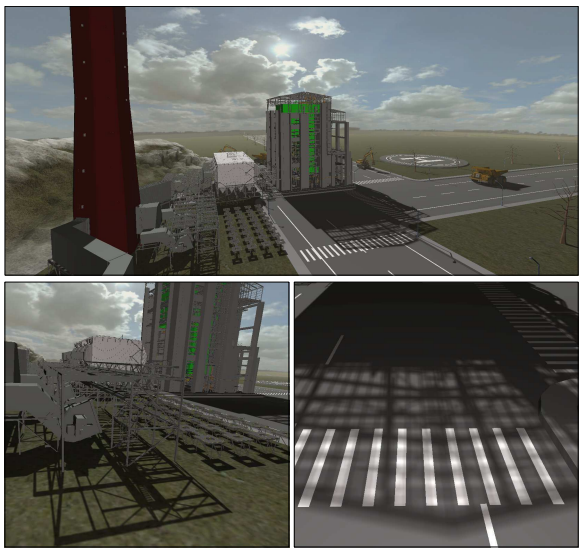


Figure 1: *A power plant scene (2M of polygons) rendered using our predicted virtual soft shadow mapping technique at about 3-7 fps. Notice how both soft and hard shadows are well rendered by our algorithm.*

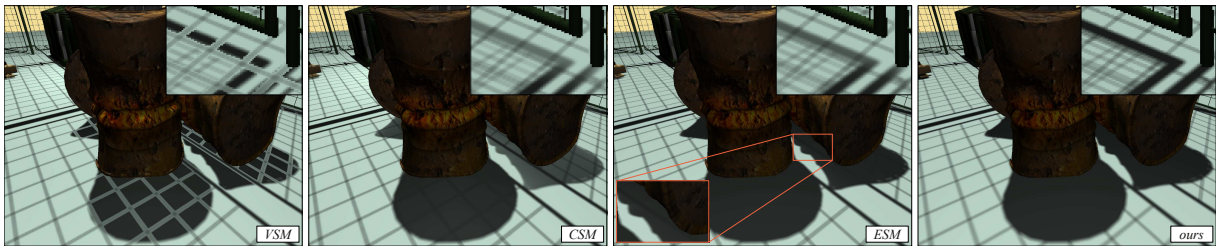


Figure 2: *Illustration of some of the lighting artefacts exhibited by prefiltering techniques. In all cases we used a single* 2048×2048 *shadow map, and enabled hardware* $\times 16$ *anisotropic filtering for VSM, CSM, and ESM.*

cluder parallel to the area light source, the relatively expensive back-projection procedure boils down to a very simple convolution between the shadow map visibility function and a low pass filter *blurring* the underlying hard shadows [SS98, Fer05]. Fortunately, in practice deviating from this ideal configuration still produces plausible shadows.

For all these reasons, such convolution based soft shadow mapping techniques are still widely used in the rendering engines of the game and movie industries. The convolution is usually carried out by means of Percentage Closer Filtering (PCF) [RSC87] which suffers from banding or noise artefacts. Recently, Annen et al. [ADM*08] showed constant time evaluation can be achieved via appropriate pre-filtering approximations. On the down side, shadowing artefacts such as loosing contact shadows cannot be avoided when rendering complex scenes (figure 2). Moreover, it requires multiple Summed Area Tables (SAT) which are prohibitively expensive to compute on high resolution shadow maps. This appealing constant time feature is further mitigated by the fact that each evaluation involves the fetching of about 256 floating point values from the texture memory. Finally, enabling anisotropic filtering with SATs remains an open issue.

For this work, our goal is to enable plausible soft shadow rendering on very large and complex scenes while avoiding visual artefacts and maintaining high performance. To this end, we first revisit convolution based soft shadow techniques and present a novel evaluation mechanism of the convolution integral which focuses on high quality. It guarantees a high smoothness of both the penumbrae and high frequency shadows by embedding an efficient analytical anisotropic antialiasing filter. To our knowledge, this is the first soft shadow algorithm being able to perform anisotropic antialiasing without falling back to a naive and expensive screen space multisampling strategy. Moreover, it does not require the precomputation of any intermediate representation that is a mandatory feature for dynamic high resolution shadow maps.

Based on this proper integration mechanism, we address the famous discretization issue by generating on the fly virtual shadow maps matching some predicted resolutions. This allows us to reach arbitrarily large shadow maps while avoiding over-sampling by locally adjusting the shadow map resolutions. Our main contribution here is the derivation of a perceptual resolution prediction metric specifically tailored to our soft shadow evaluation method. We exploit the fact that penumbrae is a low frequency phenomena masking, up to some extent, discretization errors [DHS*05]. More precisely, we conducted a user study to establish a human vision aware relation between the screen space frequency of the penumbra and the minimal required shadow map resolution. Thanks to this prediction metric, for the first time, soft shadows can sometimes be rendered faster than their respective hard versions.

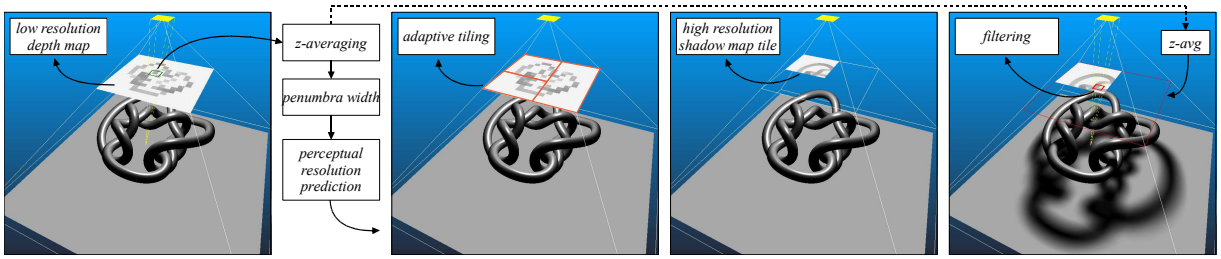
2. Related Work

The last decades have been extremely productive regarding shadow algorithms, and a complete review of them is beyond the scope of this paper. We refer the readers to Eisemann et al. [EASW09] and Scherzer et al. [SWP10] for recent overviews on real time shadow casting algorithms.

Physically based soft shadow algorithms strive at computing the visibility between a given point and an area light source. Akenine-Möller and Assarsson [AAM03] proposed penumbra-wedges as an extension of the well known shadow volume algorithm [Cro77]. This algorithm has been extended many times, both in terms of accuracy [LAA*05, SEA08], speed [FBP08], and generality to support alpha textured polygons [FGBP09]. Nevertheless, all these variants require the extraction and rasterization of object based primitives making them limited to clean polygonal meshes, and impracticable for complex scenes.

Soft Shadow Mapping (SSM) techniques replace the complex geometry of the scene by a simple shadow map (SM) [Wil78], and employ back-projection to integrate the visibility [GBP06, AHL*06]. Continuous reconstruction methods [GBP07, SS07] have been proposed to overcome the artefacts caused by the discrete nature of the SM. Yang et al. [YFGL09] showed that SSM performance can be significantly improved by exploiting the light and screen space coherence via sophisticated hierarchical algorithms. However, it requires the construction of an expensive multi-scale data structure which is impracticable for high resolution shadow maps. Moreover, discretization artefacts is still an open and challenging issue for such approaches.

Convolution based soft shadow methods render plausible penumbrae by applying a low pass filter to the occluding signal. This idea was first introduced by Soler and Sillon [SS98]. The convolutions were efficiently performed by Fast Fourier Transformations (FFT), but an explicit separation of the blockers and receivers is required. Recently, Eisemann and Décoret [ED08] proposed a variant of this approach using prefiltered occlusion textures and blending



(a) Scene analysis (b) Adaptive tiling (c) Tile rendering (d) Soft filtering

Figure 3: *Overview of our algorithm. Please see section 3, overview, for the explanations.*

heuristics for self occlusions and the fusion of multiple layers. Fernando [Fer05] proposed a more flexible approach where the convolution is evaluated by means of Percentage Closer Filtering (PCF) [RSC87] of a classic shadow map. By adjusting the size of the filter according to the local average depth of the occluders, plausible penumbrae are carried out. PCF allows to properly filter a shadow map by applying standard texture filtering techniques to the results of the shadow tests. More recently, several approximations of the shadow test function have been proposed to enable pre-filtering of shadow maps. They include Variance Shadow Maps (VSM) [DL06], Convolution Shadow Maps (CSM) [AMB*07] and Exponential Shadow Maps [AMS*08]. Lauritzen [LAU07] adapted Fernando's method to utilize a variant of VSM for the filtering step. In the same vein, Annen et al. [ADM*08] extended their CSM framework to achieve constant time soft shadows for both the depth averaging and filtering steps. Recently, Yang et al. [YDF*10] showed the depth averaging step can also be achieved using a VSM. Unfortunately, as explained in the introduction, all these pre-filtering approximations exhibit severe limitations preventing them to be used for the rendering of large scale scenes.

Therefore, in this paper we derive a novel proper evaluation method which guarantees both high smoothness and high quality anisotropic filtering.

Discretization-free shadow mapping methods aim to overcome aliasing artefacts produced by an insufficient shadow map resolution. So far, this issue has only been addressed in the context of hard shadows. Alias-free shadow maps [AL04] achieve ray-casting quality by rendering the scene to a non uniform depth image where the *texels* are located at the exact positions of the visible points. Even though GPU implementations have been proposed [Arv07, SEA08], they remain quite expensive for practical uses. Reparametrization techniques [SD02, WSP04, LGQ*08] increase the effective SM resolutions by counter balancing the view perspective. Unfortunately, such methods work well in some given configurations only. Moreover, it is unclear how such non uniform representations, including alias-free SM, can be exploited as occluders to compute soft shadows since the sampling requirements appear to be very different. A simpler and more flexible approach consists in subdividing the shadow map space into multiple high resolution depth images. For instance, Adaptive shadow maps (ASM) [FFBG01] iteratively refine the SM when aliasing is detected following a quad-tree subdivision strategy. In parallel works, Lefohn et al. [LSO07] and Giegl and Wimmer [GW07] proposed to avoid the expensive recursions by computing *a priori* the required SM resolution.

In this paper, we follow this latter approach and extend it to the case of soft shadows by deriving an appropriate resolution prediction metric.

3. Overview

An overview of our overall algorithm is given in figure 3. A low resolution shadow map covering the scene is rendered to then estimate the penumbra width for each visible point (figure 3-a). This information is then used to predict the required resolution for each portion of the shadow map plane. Based on a kd-tree traversal of the predicted resolution map (figure 3-b), high resolution shadow maps are then rendered on the fly (figure 3-c) and utilized right away to produce the actual penumbrae (figure 3-d). This later step is performed by employing a variant of PCF. Notice that the filter sizes are adjusted with respect to the average occluder depths estimated from the low resolution shadow maps. We empirically found the average occluder depths tolerate much higher approximations than the final filtering steps.

The algorithm to produce the actual soft shadows is detailed in section 4. It can be used independently of the adaptive shadow map creation procedure which is detailed in section 5 together with our resolution prediction metric.

4. Percentage Closer Soft Shadows Revisited

In order to introduce the notations, we will briefly review percentage closer soft shadows (PCSS) [Fer05]. Let $\mathbf{y} \in \mathbb{R}^3$ be the world-space position of a given screen pixel, and z_y and $\mathbf{x}$ its light space depth and 2D shadow map coordinates respectively. A shadow map stores the discrete function $z(\mathbf{x})$ which corresponds to the depth of the closest occluder for each texel $\mathbf{x}$. The occluding function o implements the shadow test as:

$$o_{z_y}(\mathbf{x}) = \begin{cases} 1, & \text{if } z_y \leq z(\mathbf{x}); \\ 0, & \text{otherwise.} \end{cases}$$

Let S be the source *characteristic* function defined such that:

$$S(.) = \begin{cases} 1, & \text{if } . \text{ is inside the light source;} \\ 0, & \text{elsewhere.} \end{cases}$$

Assuming a planar occluder of constant depth z_o, the percentage of visibility $V(\mathbf{y})$ can be expressed as a convolution between the light source S and the occluding function o_{z_y}:

$$V(\mathbf{y}) = \frac{1}{\alpha^2 A_l}(s_\alpha \otimes o_{z_y})(\mathbf{x}) \quad (1)$$

where $\alpha = \frac{z_y - z_o}{z_y}\frac{z_n}{z_o}$ is the scale factor induced by the relative position of the occluder, receiver, and shadow map plane of depth z_n, and $s_\alpha(\mathbf{x}) = S\left(\frac{\mathbf{x}}{\alpha}\right)$ is called the *soft shadowing filter*. A_l denotes the area of the light source.

In practice, scenes are composed of many occluders with various shapes. In order to produce plausible soft shadows, the size of the filter s_α has to be appropriately adjusted for each point $\mathbf{y}$. Following previous work [Fer05, ADM*08], we achieve this goal by taking $z_o = z_{avg}$ the local average of the neighbor occluding depth values stored in the shadow map. This neighborhood is chosen as the intersection of the shadow map plane and the pyramid formed by the visible point and the light source (figure 3-b). This is another convolution of the occluding depth signal $o_{z_y}(\mathbf{x})z(\mathbf{x})$ which might be relatively expensive to compute. However, as we will detail later, in our complete algorithm, this step is performed on a low resolution shadow map.

4.1. Smooth convolution evaluation

A classic approach to evaluate the convolution of equation 1 consists in accumulating the result of the shadow test against each texel falling inside the filter s_α. This approach is very simple and fast, but it exhibits banding artefacts since the result of the convolution remains constant to small variations of the filter (figure 4-a). This undesirable effect can be reduced by exploiting the 2×2 *bilinear PCF* capability of current graphics hardware. Unfortunately, such an approach can only deal with filter sizes being an integer multiple of the texel sizes. Consequently, discontinuities still appear for a quickly varying penumbra width (figure 4-b).

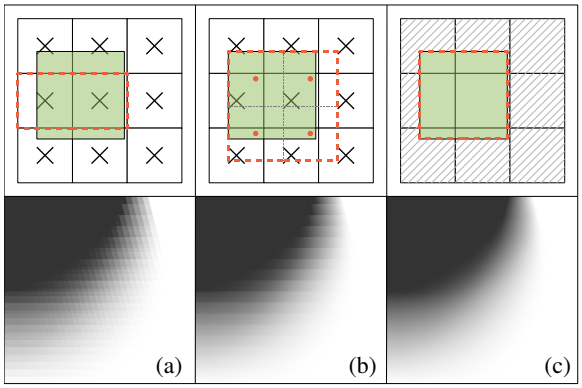


Figure 4: *Illustration (top row) and comparison (bottom row) of three different integration methods. Crosses represent texel centers, the requested filter kernel is shown in green while the the equivalent canonical one is in red. (a) Classic sampling. (b) Sampling with HW 2×2 bilinear PCF. The red dots show the texture query positions. (c) Our smooth integration method where the texels are considered as quads which are clipped by the kernel.*

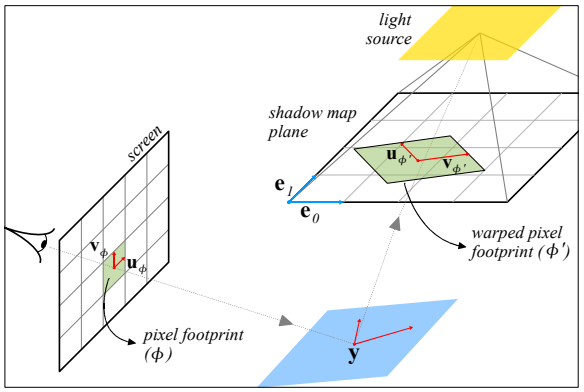


Figure 5: *Illustration of the warping of the pixel footprint ϕ through the surface patch (in blue) to the shadow map plane yielding a parallelogram (in green).*

In order to guarantee a high smoothness, we interpret each shadow map texel as a quad with a constant depth (instead of a point sample). The convolution integral boils down to the following discrete sum where the shadow test result against each texel $\mathbf{t}_{i,j}$ is weighted by the area $c_{s_\alpha}(\mathbf{t}_{i,j})$ of its intersection with the box filter s_α:

$$V(\mathbf{y}) = \frac{1}{\alpha^2 A_l}\sum_{\mathbf{t}_{i,j}} c_{s_\alpha}(\mathbf{t}_{i,j}) o_{z_y}(\mathbf{t}_{i,j}) \quad (2)$$

The superior smoothness offered by this method is depicted in figure 4-c. For a rectangular light source aligned with the shadow map grid, s_α corresponds to an axis aligned box filter, and the implementation of its coverage function c_{s_α} is straightforward. It requires only 9 floating point operations that is relatively cheap compared to the mandatory texture fetch.

4.2. Anisotropic antialiasing

When the screen space size of the penumbra becomes smaller than a pixel, the soft transition becomes hard and aliasing might occur since the shadow signal V and the sampling resolution do not satisfy the Nyquist criteria. Multi-sampling strategies [PWC*09] could be easily adapted to our case. However, such an approach is far to be satisfactory since an arbitrary large number of sub-samples can be required to totally remove the aliasing, and discontinuities are likely to occur at the transitions between soft penumbrae and the anti-aliased hard shadows.

Our solution was inspired from anti-aliased splatting techniques [ZPvBG01]. Instead of such a discrete integration, we show it is possible to perform an analytical integration over each pixel area. This integral can be expressed as a convolution of the soft visibility signal V and a low pass box filter ϕ which evaluates to 1 inside the given pixel and 0 outside. Let $\mathcal{M}$ be the mapping from screen space to shadow map space through the receiver patch. Then the anti-aliased visibility

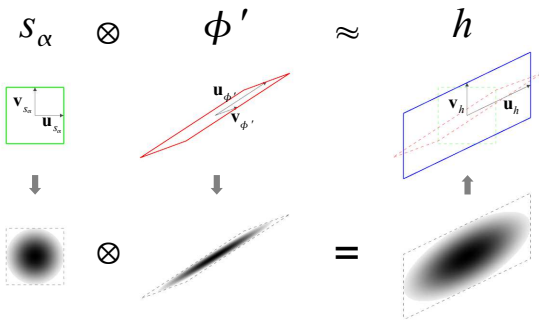


Figure 6: *The convolution between the two parallelogram filters s_α (green) and ϕ' (red) is approximated by interpreting them as ellipsoid Gaussian for which the convolution can be carried out analytically. The result is still a Gaussian which is converted back to a partially axis aligned parallelogram filter h (blue).*

$\tilde{V}(\mathbf{y})$ is given by:

$$\begin{aligned}\tilde{V}(\mathbf{y}) &= (\mathcal{M}(\phi)\otimes V)(\mathbf{y}) \\ &= \frac{1}{\alpha^2 A_l}(\mathcal{M}(\phi)\otimes s_\alpha \otimes o_{z_y})(\mathbf{y}) \qquad (3)\end{aligned}$$

In order to evaluate this expression efficiently, the key idea is to analytically compute the anti-aliased convolution filter $h = \mathcal{M}(\phi)\otimes s_\alpha$ and then reuse the previous smooth evaluation mechanism of section 4.1. To achieve this goal, some approximations have to be done. As usual with anisotropic texture filtering techniques [Hec89], we assume the current pixel corresponds to a planar patch, and we locally approximate the mapping $\mathcal{M}$ by an affine transformation neglecting the non linear part of the two involved perspective projections. These approximations yield for the warped low pass filter $\phi' \approx \mathcal{M}(\phi)$ a parallelogram of axes $\mathbf{u}_{\phi'}$, $\mathbf{v}_{\phi'}$. In practice, they are easily computed by projecting twice the pixel axes $\mathbf{u}_\phi$, $\mathbf{v}_\phi$ from screen to the shadow map plane via the receiver's patch. This step is illustrated in figure 5. Note that, even though locally approximating the perspectives projections by affine mappings might lead to small gaps in-between the warped filters of neighboring pixels, in practice this has very little effect on the final results as shown in figure 11.

The principle of our analytical computation of this convolution is depicted in figure 6. The key idea is to temporarily interpret the two filters ϕ' and s_α as ellipsoid Gaussian. We define an ellipsoid Gaussian $\mathcal{G}_\mathbf{V}$ of variance $\mathbf{V}$ as:

$$\mathcal{G}_\mathbf{V}(\mathbf{x}) = e^{-\mathbf{x}^T\mathbf{V}^{-1}\mathbf{x}} .$$

Let $\mathbf{M}_{\phi'}$ (resp. $\mathbf{M}_{s_\alpha}$) be the 2×2 affine transformations from the local parametrization space of ϕ' (resp. s_α) to the shadow map space. As an example, we have $\mathbf{M}_{\phi'} = \left[\mathbf{u}_{\phi'}\mathbf{v}_{\phi'}\right]$. A bounding parallelogram defines a unique ellipsoid. Therefore, it is easy to see that the kernels ϕ' and s_α correspond to ellipsoid Gaussian of variances $\mathbf{V}_{\phi'} = \mathbf{M}_{\phi'}\mathbf{M}_{\phi'}^T$ and $\mathbf{V}_{s_\alpha} = \mathbf{M}_{s_\alpha}\mathbf{M}_{s_\alpha}^T$ respectively.

Since the convolution of two Gaussian is also a Gaussian of summed variance, the final kernel h is temporarily approximated as a Gaussian of variance $\mathbf{V}_h = \mathbf{V}_{\phi'} + \mathbf{V}_{s_\alpha}$. Finally, in order to be able to reuse the previous integration method while maintaining high performance, we convert back this Gaussian filter to a parallelogram box filter. Since a given ellipsoid accepts an infinity of bounding parallelograms, we choose the one which has one axis aligned with the vertical shadow map axis. This is especially important to ease the *rasterization* of the kernel. This is achieved by computing the Cholesky factorization $\mathbf{v}_h = \mathbf{L}\mathbf{L}^T$ of the symmetric positive definite variance matrix $\mathbf{V}_h$ where $\mathbf{L}$ is a lower triangular matrix. The columns of $\mathbf{L}$ correspond to the axes $\mathbf{u}_h$ and $\mathbf{v}_h$ of the bounding parallelogram defining h. This procedure is illustrated in figure 6, and a complete shader code taking advantage of the characteristics of the involved matrices is provided in the appendix. It can be seen this procedure is very simple as it requires a very few floating point operations. Figure 7 shows the behavior of this pseudo convolution method on various configurations of the input filters.

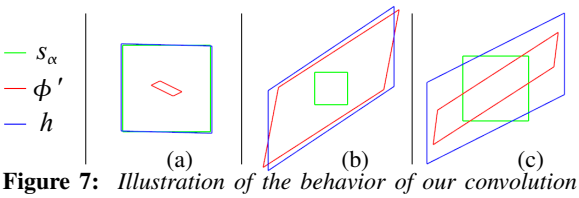
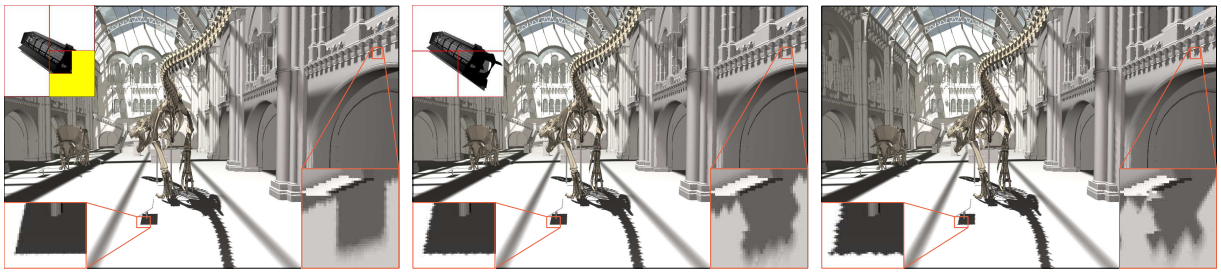


Figure 7: *Illustration of the behavior of our convolution approximation when, (a) the shadow filter s_α is much larger than the anti-aliasing filter ϕ', (b) the other way round, and (c) in an hybrid configuration.*

Recall that our smooth evaluation mechanism of section 4.1 has to compute coverage areas between the texels and the shadowing filter which is now a semi-axis-aligned parallelogram. Without any visual quality loss, in practice we approximate each vertical panel of one pixel width by an axis aligned box thus maintaining the same performance as with a fully axis-aligned filter.

5. Virtual Soft Shadow Maps

In this section we detail our adaptive virtual shadow map creation algorithm. This algorithm aims at avoiding any visible discretization artefacts by computing on the fly a set of high resolution shadow maps. Since both the computation and the use of a high resolution shadow map is expensive, the main challenge is to predict the lowest shadow map resolution which is needed to avoid any visible artefact. The case of hard shadows is simple since it is enough to guarantee a 1 : 1 mapping, i.e., to guarantee that the footprint ϕ' of a pixel in shadow map space is larger than a shadow map texel [GW07]. In the case of soft shadows, this criteria often defines a too conservative higher bound. Indeed, since penumbrae is a low frequency phenomena, it is expected that same quality can be achieved using an input of lower resolution. However the relation between the shadow map resolution, and the screen space visual quality of the produced soft shadows is still an open and difficult problem as it heavily relies on the complex human vision system. For this reason, this relation cannot be established through a theoretical frequency analysis, and instead, we propose to directly estimate it by means of a user study.



(a) (b) (c)

Figure 8: *Comparison on a complex scene composed of about 1.5M of triangles. (a) Using our algorithm and a cubic polynomial approximation of our perceptual metric* @4.8*fps. (b) Our algorithm with a linear approximation* @4.7*fps. (c) Our convolution algorithm but using a single high resolution* 4096^2 *shadow map* @11*fps.*

In the rest of this section, we first present our perceptual prediction metric (section 5.1), and show how it is used in section 5.2.

5.1. Perceptual Resolution Prediction Metric

Our shadow map resolution problem can be stated as: what is the minimal size $\rho(\dots)$ of the filter h in texels that is required to ensure a high quality penumbra? This function ρ brings into play many complex phenomena of the human vision system, and therefore it ideally depends on a huge number of parameters. In order to get a practical solution, in this study we will seek for a non optimal but conservative function $\rho(n)$ that depends only on the screen space size n of the penumbra in pixels. Recall that in our case, n corresponds to the screen space size of the full anti-aliased shadowing filter h. In order to estimate it with highest accuracy as possible, we conducted the following user study.

User study

Given a screen space penumbra size n, we run a binary search algorithm which quickly converge to an ideal value $\rho(n)$. The key idea of our system is to let the user drive the search algorithm. The search range is initialized with $[1:2n]$. At each step, a penumbra image is computed from a shadow map of resolution satisfying the constraint implied by the middle value of the current search range. This image is presented to the user beside a reference image computed using a high resolution shadow map. The user tells the system whether the proposed image is as good as the reference one, thus discarding half of the search range. This step is repeated until convergence, i.e., until the leading shadow map resolution does not change.

In order to ease the analysis of this experiment, we used for the test scene a simple disk casted on a parallel receiver plane. The light source is a square parallel to the rest of the scene. The camera is positioned at the light source center, and only the receiver plane is rendered. The produced images are similar to the ones of figure 4 but with constant penumbra widths. The choice for a disk is to make sure all possible edge orientations are taken into account. The screen space penumbra size n is set by adjusting the width of the light source. We tested power of two penumbra sizes only which appeared to be sufficient in practice. Finally, this test has been run for four different configurations: static/animated occluder and with/without a texture on the receiver plane and on a group of 12 users.

The results of this experiment are shown in figure 9. We add that the variance is extremely small for penumbra sizes up to 64 pixels, but it rapidly increases for larger penumbrae. Nevertheless, it still provides a reasonable and useful idea on the general behavior for large penumbrae. As expected, both the motion and texture masking effects significantly influence the behavior of ρ. It is worth noting that a dynamic scene requires a higher resolution to avoid noticeable flickering artefacts. Note that, the results with texture masking are provided for comparison purpose only. Indeed, taking advantage of texture masking in a proper way is much more complicated as the number of parameters which have to be taken into account explode. On the other hand, it is relatively easy to detect a static scene. As an example, since the relevant motion is between the occluders and the light source, when only the camera is moving, the whole images can be rendered using the more aggressive metric.

In practice, these results can be directly exploited using tables. In our implementation, we preferred to approximate these curves by fitting cubic polynomials using a $\log_2$ scaling to get a better fit:

$$\rho_{\text{dynamic}}(n) = P_{\text{d}}(\log_2(n)) \qquad P_{\text{d}}(x) = 0.667 + 0.7794x + 0.3967x^2 - 0.02716x^3$$

$$\rho_{\text{static}}(n) = P_{\text{s}}(\log_2(n)) \qquad P_{\text{s}}(x) = 0.5477 + 1.406x - 0.04687x^2 - 0.003944x^3$$

All the results of this paper have been obtained using the rule for dynamic scenes (i.e., ρ_{dynamic}) which is the most conservative. The plot of ρ_{dynamic} in figure 9 might suggest a linear model (after the $\log_2$ mapping). However, in prac-

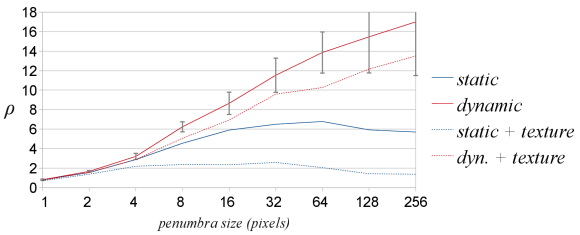


Figure 9: *Results of our user study to estimate the perceptual relation* ρ. *The standard deviation is shown for the dynamic case.*

tice we found that it is very important to have a model that accurately reproduces the subtle variations of the beginning of the curve where the variance is very small. For instance, figure 8 compares the above cubic approximation against a linear model tuned to give the same frame rate. As can be seen, such a tuned linear model yields a much lower quality.

Resolution prediction

The previous perceptual metric ρ gives us the minimal number of texels that the shadowing filter h should contain in function of its screen space size in pixels. Owing to the mapping from shadow map space to screen space, even for a squared light source, the screen space sizes n_0, n_1 of the kernel h along the $\mathbf{e}_0$ and $\mathbf{e}_1$ directions of the shadow map can significantly vary (see figure 5), and so does the respective predicted resolutions r_0 and r_1. Recall that for anti-aliasing purpose we have already computed the warped shadow map space pixel's footprint ϕ'. Therefore, n_0, n_1 can be directly computed as the lengths of the axis aligned dimension vectors of the filter h expressed in the local parametrization space of ϕ':

$$n_0 = \left\| \mathbf{M}_{\phi'}^{-1} \mathbf{e}_0 \mathbf{e}_0^T \mathbf{u}_h \right\| \ , \ n_1 = \left\| \mathbf{M}_{\phi'}^{-1} \mathbf{e}_1 \mathbf{e}_1^T \mathbf{v}_h \right\| \ . \qquad (4)$$

Note that since the axis $\mathbf{v}_h$ is already aligned with the vertical axis $\mathbf{e}_1$ of the shadow map, we have $\mathbf{e}_1 \mathbf{e}_1^T \mathbf{v}_h = \mathbf{v}_h$.

Finally, the predicted shadow map resolutions r_0 and r_1 correspond to the ratios between the required number of texels and the respective shadow map space sizes of the filter kernel h:

$$r_0 = \frac{2\mathbf{e}_0^T \mathbf{u}_h}{\rho(n_0)} \ , \ r_1 = \frac{2\mathbf{e}_1^T \mathbf{v}_h}{\rho(n_1)} \ . \qquad (5)$$

5.2. Overall algorithm and implementation details

We now have all the ingredients to assemble the full algorithm which is summarized below:

1 Create a geometry buffer: render the scene from the view point while storing depth and normal attributes. The normals are needed to warp pixel's axes (fig. 5).
2 Render a low resolution shadow map using conservative rasterization (fig. 3-a).
3 Create a *kernel shape* buffer storing for each screen pixel its respective kernel axes $\mathbf{u}_h$, $\mathbf{v}_h$ (sec. 4 and 4.2), and detect and discard fully lit pixels for future computations.
4 Predict for each visible point the required shadow map resolution (sec. 5.1) and store this information in light space in a low resolution **Shadow Map Tile Grid** (SMTG).
5 Build a pyramidal version of the SMTG and transfer it to host memory.
6 Traverse the SMTG pyramid top-down, building a KD-tree on-the-fly, to decide how the virtual shadow map tiles are constructed (fig. 3-b).
7 When a leaf node is detected based on the predicted resolutions and maximal texture resolution, render a shadow map of appropriate resolution (fig. 3-c).
For each "in penumbra" visible point mapping to the tile:
8 Apply the soft filtering procedure of section 4.1 using the filter properties stored in the *kernel shape* buffer.

Some of these steps deserve some additional details.

Initialization & filter sizes. As many other algorithms, we employ a deferred shadowing strategy to avoid multiple rasterizations of the scene to screen space (step 1). Our prediction metric needs to know in advance the shape of the filter kernels, which themselves require the computation of a local average depth occluder. Fortunately, we found this latter can be computed from a low resolution shadow map without decreasing the visual quality. On the other hand, in order to make sure that no small geometry will be neglected, it is necessary to render this initial shadow map using conservative rasterization (step 2). Otherwise, the shadows produced by very thin geometries might be missing in the final images. In our system we implemented the second algorithm presented by Hasselgren et al. [HAMO05] which consists in rasterizing bounding enlarged triangles which are then clipped by the respective bounding rectangle in the fragment shader.

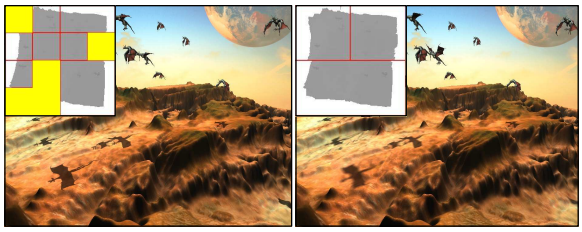


Figure 10: *A scene rendered with our adaptive algorithm, with a punctual light source at 15 FPS (left), and an area light source at 27 FPS (right). The tiles are shown in the top left corners.*

The shadow map tile grid (SMTG) is a low resolution (e.g., 64×64) two component floating-point texture associated to the shadow map plane. Each cell (i.e., texel) of the SMTG stores the minimal shadow map resolutions that are required for its respective covered region (one for each dimension). It is computed from the *kernel shape* buffer in a single rendering pass as follow. A point primitive is issued for each screen pixel. In the vertex shader, we use the `vertex ID` attribute to retrieve its respective world-space position from the geometric buffer, and to compute its ideal resolution $\mathbf{r} = [r_0, r_1]$ from the *kernel shape* buffer. Then, it is projected onto the shadow map plane to compute its target SMTG coordinates, and its value $\mathbf{r}$ is accumulated into the SMTG with *minimum* blending. Note that these point primitives do not have any attribute, and so they do not have to be stored at all.

Shadow map tile creation. For the steps 5 to 7, we follow Giegl and Wimmer [GW07]. The only difference in our implementation is that the SMTG pyramid is constructed on the GPU to reduce CPU load to the strict minimum (step 5). In order to keep the discussion self-contained, we briefly review these steps and refer to Giegl and Wimmer [GW07] for the implementation details. The SMTG is first converted to a pyramidal version that is then traversed top-down to build a KD-tree on-the-fly where each node records the maximal resolution of all the SMTG cells it covers. If the stored resolution could be satisfied by a shadow map smaller than the maximal predefined limit size, then we hit a leaf node, and a shadow map covering the tile is created. All screen pixels mapping to this tile are shadowed immediately using this shadow map.

Kernels overlapping multiple tiles are easily handled by storing for each pixel both its current percentage of visibility V and its corresponding filter area A such that the contributions coming from the current shadow map can be properly

merged as follow:

$$V_{dst} = V_{dst} * A_{dst} + V_{src} * A_{src}$$
$$A_{dst} = A_{dst} + A_{src}$$

Note that this simplicity is due to our proper integration method which, in contrast to prefiltering or 2x2 bilinear PCF, does not require any additional care on the transitions.

Fully-lit pixel optimization consists in explicitly detecting such pixels which are known in advance to yield to a full visibility factor. This is easily achieved in step 3 when calculating the average occluder depth. If no occluder texels is detected, then we can skip all further computation for this pixel by recording this information in the *kernel shape* buffer. This is possible thanks to the conservative rasterization. On the other hand, even when all texels are detected as occluders, we cannot conclude on the fully-occluded status of the pixel.

6. Results

In this section, we present performance and visual results followed by some comparisons. We implemented our technique with DirectX 10, and all our results have been obtained with a GeForce 280 GTX graphics card. In our experiments we used a resolution of 1024×1024 for the global low resolution SM, and instantiated 4096×4096 SMs for the tiles.

Figures 1 and 8 show the behavior of our algorithm on two large and complex scenes composed of about 2M and 1.5M of polygons respectively. Both are rendered at about 3 to 7 fps for an output image of 1024^2 pixels. The rendering cost of each part of the algorithm for a typical frame of these scenes is given in table 1. As can be seen, the rendering of the shadow map tiles represents a large part of the rendering cost, and the overhead of conservative rasterization to compute the low resolution shadow map is negligible. Likewise, thanks to the use of a low shadow map resolution, the z-averaging step represents a minor part of the overall algorithm. Figure 8-c also shows the great quality improvement offered by our adaptive tiling approach compared to using a single high resolution shadow map of 4096^2.

The effect of our perceptual prediction metric on the generated tiles is enlighten in figure 10 for hard and soft shadow configurations. As can be seen, since softer shadows require shadow maps of lower resolutions, fewer shadow map tiles have to be rendered in the case of soft shadows. Therefore, in this example, the soft shadow images are rendered twice as fast as the versions with hard shadows.

	fig.1	fig.8		fig.1	fig.8
Geometry buffer	9	49	SMTG & tiling	10.5	10.5
Low res. SM	15	53	High res. SMs	55.8	91
Z-avg	4.5	9	Filtering	82.5	2.4
			Total	**177**	**215**

Table 1: *Rendering time in ms of each part of our algorithm for a typical 1024^2 frame of the scenes of figures 1 and 8. Nine shadow map tiles have been created for the former and three only for the latter.*

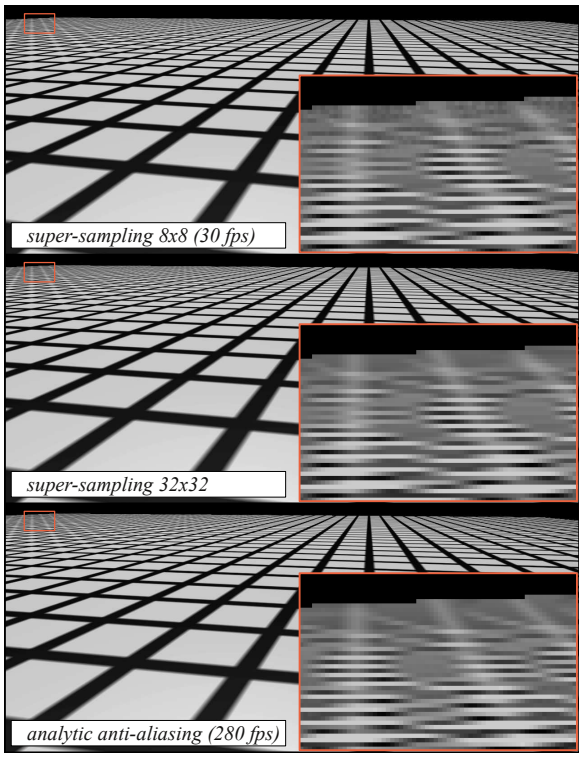


Figure 11: *Comparison of our analytic antialiasing method against a naive screen space super-sampling strategy (SM resolution: 1024×1024). The scene is composed of single but large grid parallel to the floor. Please zoom-in to see the differences.*

The performance of our analytic anisotropic anti-aliasing technique of section 4.2 is demonstrated in figure 11 where it can be seen it exhibits the same quality as a brute force 32×32 screen space super-sampling strategy. Furthermore, on this tough example it achieves a higher quality than 8×8 super-sampling while being an order of magnitude faster.

Comparisons

Since our algorithm is the first one tailored for the rendering of complex scenes, it is difficult to perform any fair comparisons. Nevertheless, table 2 gives some insights on the relative performance of most recent image based soft shadow algorithms for a large and complex scene and for which a 4096^2 shadow map appeared to be a reasonable tradeoff. Notice that for the small light source, a shadow map of 4096^2 was not completely sufficient and our adaptive method led to a higher effective resolution, hence the slow down compared to the fixed strategy. On the contrary, for the large light source, our adaptive strategy is significantly faster because for a large part of the scene a lower SM resolution was enough. The relative poor performance of Annen et al. CSSM [ADM*08] and Yang et al. [YFGL09] techniques are easily explained by their need of intermediate representations which become prohibitively expensive for high resolution SM: 80ms for a 4096^2 neighborhood buffer of two

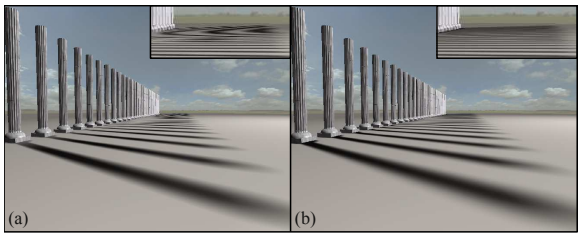


Figure 12: *Comparison of (a) CSSM [ADM*08] (4 terms using SATs, ∼ 6fps) and (b) our algorithm (∼ 28fps). Shadow map resolution:* 2048×2048. *Notice the strong aliasing of CSSM.*

components ([YFGL09]) and 180ms for two 4096^2 SATs of four CSM terms ([ADM*08]). Moreover, let us recall that the constant time evaluation mechanism of CSSM actually requires the fetching of $4^4 = 256$ scalar values from video memory (4 terms, 4 coefficients per term, 4 corners per term, and 4 scalar per corners for bilinear interpolation) that is as expensive as a 16×16 kernel in our case.

Finally, figure 12 shows the superior ability of our method to smooth out high frequency shadows compared to CSSM. Indeed, performing anisotropic filtering on a SAT is still an open problem.

7. Discussion and Conclusion

We presented a fast soft shadow rendering algorithm tailored for the high quality rendering of large and complex scenes. To maintain high performance, it follows the general principle of percentage closer soft shadowing [Fer05], and thus shares some of its limitations. In particular, as Annen et al. [ADM*08], we assume the blockers are locally planar that is obviously unlikely to be the case in real world scenes. Like most of the image based algorithms, we also neglect all the possible occluders which are visible from some parts of the light sources, but not from the light center. Fortunately, as shown in, e.g., figure 13, even in the case of the fusion of numerous occluders with very different depths (e.g., the top border of the right wall with the right branches of the trees), the produced penumbrae still look reasonable. Unlike approximate pre-filtering approaches, we cannot claim about constant time evaluation. Nevertheless, we emphasize

Light source size	10	80	memory
Our method (adaptive resolution)	12.33	21.62	68 MB
Our method (fixed resolution)	16.82	7.30	64 MB
Annen et al. 2008 (4 terms & SATs)	5.10	5.08	1088 MB
Yang et al. 2009	4.77	1.08	832 MB
Average of 64 hard shadow maps	0.60	0.61	64 MB

Table 2: *Performance comparisons in frame per second on a scene (a set of trees) of 782K triangles. Except for the first row, we used a single* 4096×4096 *shadow map which appeared to be a reasonable trade-off. The last column reports the memory consumption for the SM(s) and related intermediate representations. We had to reduce the accuracy of the SATs and the NBuffer to 16 bits per scalar to fit into the video memory.*

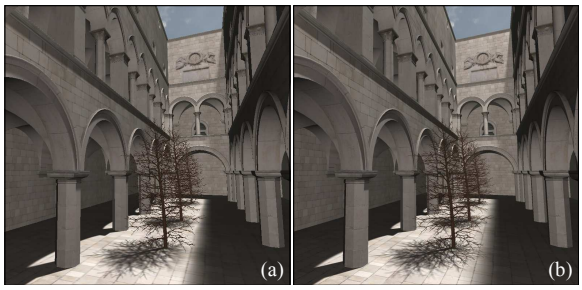


Figure 13: *Comparison of our algorithm (a) against a reference image (b) computed using the average of 1024 light samples.*

this performance issue is partly compensated by the use of a low resolution shadow map for the depth averaging step, the lack of any expensive intermediate representation, and a smart adaptive shadow map creation algorithm which limits oversampling in the shadowing filters. Quality wise, our algorithm does not exhibit odd light bleeding or losing contact shadow effects as shown in figure 2, it solves the discretization aliasing issue by allowing high resolution shadow maps, and it seamlessly handles soft and hard shadows with high quality anisotropic filtering. As a result, for complex enough scenes, our approach outperforms pre-filtering techniques both in term of speed, and quality, and we believe our algorithm is perfectly well suited for production rendering engines where reliability, visual quality and high performance are most important criteria. Nevertheless, we acknowledge that in some specific cases, prefiltering methods can still be more appropriate than ours in term of performance, making both approaches more complementary ones than competitors.

One of the key features of our algorithm is an original perceptual prediction metric which drives the generation of adaptive shadow maps. This metric has been directly established from a user study. Unfortunately, that does not bring us any hint about how to tweak the performance versus visual quality tradeoff in a consistent manner. In other words, it is indeed not possible to ask users if a given picture is "half" the quality of a reference image since this is a too subjective notion. Therefore it would be interesting to conduct a theoretical analysis of the underlying masking effect of the soft penumbrae to establish a parametric model which could be fitted to the results of our experiments. Such a model would also allow us to properly extrapolate the prediction metric for very large penumbrae. In the same vein, our perceptual metric takes only into account the static versus dynamic factor. However, we show that exploiting texture-masking could lead to additional performance boost. This remains an open and challenging problem though.

Acknowledgments

This work is supported by the Bird associated team (INRIA-DREI), the NSF of China (60933007, 60873046), and the 973 program of China (2009CB320801). We also thank Lifeng Wang for his support. The original Sponza Atrium model is made by Marko Dabrovic, RNA studio.

References

[AAM03] ASSARSSON U., AKENINE-MÖLLER T.: A geometry-based soft shadow volume algorithm using graphics hardware. *ACM Transaction on Graphics (Proceedings of SIGGRAPH 2003) 22*, 3 (2003), 511–520. 1, 2

[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 (Proceedings of SIGGRAPH 2008) 27*, 3 (2008), 34:1–34:8. 2, 3, 4, 8, 9

[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). 1, 2

[AL04] AILA T., LAINE S.: Alias-free shadow maps. In *Proceedings of Eurographics Symposium on Rendering 2004* (2004), pp. 161–166. 3

[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. 3

[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. 3

[Arv07] ARVO J.: Alias-free shadow maps using graphics hardware. *journal of graphics, gpu, and game tools 12*, 1 (2007), 47–59. 3

[Cro77] CROW F. C.: Shadow algorithms for computer graphics. *Proceedings of ACM SIGGRAPH 77 11*, 2 (1977), 242–248. 2

[DHS*05] DURAND F., HOLZSCHUCH N., SOLER C., CHAN E., SILLION F. X.: A frequency analysis of light transport. *ACM Transaction on Graphics (Proceedings of SIGGRAPH 2005) 24*, 3 (2005), 1115–1126. 2

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

[EASW09] EISEMANN E., ASSARSSON U., SCHWARZ M., WIMMER M.: Casting shadows in real time. In *ACM SIGGRAPH Asia 2009 Courses* (2009). 2

[ED08] EISEMANN E., DÉCORET X.: Occlusion textures for plausible soft shadows. *Computer Graphics Forum 27*, 1 (2008), 12–23. 2

[FBP08] FOREST V., BARTHE L., PAULIN M.: Accurate Shadows by Depth Complexity Sampling. *Computer Graphics Forum (Proceedings of Eurographics 2008) 27*, 2 (2008), 663–674. 1, 2

[Fer05] FERNANDO R.: Percentage-closer soft shadows. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Sketches* (New York, NY, USA, 2005), ACM, p. 35. 2, 3, 4, 9

[FFBG01] FERNANDO R., FERNANDEZ S., BALA K., GREENBERG D. P.: Adaptive shadow maps. In *Proceedings of ACM SIGGRAPH 2001* (2001), pp. 387–390. 3

[FGBP09] FOREST V., GUENNEBAUD G., BARTHE L., PAULIN M.: Soft Textured Shadow Volume. *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering 2009) 28*, 4 (2009), 1111–1120. 2

[GBP06] GUENNEBAUD G., BARTHE L., PAULIN M.: Real-time soft shadow mapping by backprojection. In *Eurographics Symposium on Rendering* (2006), pp. 227–234. 1, 2

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

[GW07] GIEGL M., WIMMER M.: Fitted virtual shadow maps. In *GI '07: Proceedings of Graphics Interface 2007* (2007), pp. 159–168. 3, 5, 7

[HAMO05] HASSELGREN J., AKENINE-MÖLLER T., OHLSSON L.: *GPU Gems 2*. Adison-Wesley, 2005, ch. Conservative rasterization on the gpu. 7

[Hec89] HECKBERT P.: *Fundamentals of Texture Mapping and Image Warping*. Master's thesis, UCB/CSD 89/516, CS Division, U.C. Berkeley, 1989. 5

[LAA*05] LAINE S., AILA T., ASSARSSON U., LEHTINEN J., AKENINE-MÖLLER T.: Soft shadow volumes for ray tracing. *ACM Transaction on Graphics (Proceedings of SIGGRAPH 2005) 24*, 3 (2005), 1156–1165. 2

[LAU07] LAURITZEN A.: *GPU Gems 3*. 2007, ch. Summed-Area Variance Shadow Maps. 3

[LGQ*08] LLOYD B., GOVINDARAJU N. K., QUAMMEN C., MOLNAR S., MANOCHA D.: Logarithmic perspective shadow maps. *ACM Transaction on Graphics 27* (2008). 3

[LSO07] LEFOHN A. E., SENGUPTA S., OWENS J. D.: Resolution-matched shadow maps. *ACM Transaction on Graphics 26*, 4 (2007), 20. 3

[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). 4

[RSC87] REEVES W. T., SALESIN D. H., COOK R. L.: Rendering antialiased shadows with depth maps. In *Proceedings of ACM SIGGRAPH 87* (1987), pp. 283–291. 2, 3

[SD02] STAMMINGER M., DRETTAKIS G.: Perspective shadow maps. In *Proceedings of SIGGRAPH '02* (2002), pp. 557–562. 3

[SEA08] SINTORN E., EISEMANN E., ASSARSSON U.: Sample-based visibility for soft shadows using alias-free shadow maps. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering 2008) 27*, 4 (2008), 1285–1292. 2, 3

[SS98] SOLER C., SILLION F. X.: Fast calculation of soft shadow textures using convolution. In *Proceedings of ACM SIGGRAPH 98* (1998), pp. 321–332. 2

[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. 2

[WSP04] WIMMER M., SCHERZER D., PURGATHOFER W.: Light space perspective shadow maps. In *Proceedings of Eurographics symposium on Rendering 2004* (2004). 3

[YDF*10] YANG B., DONG Z., FENG J., SEIDEL H.-P., KAUTZ J.: Variance soft shadow mapping. *Computer Graphics Forum (Proceedings of Pacific Graphics 2010) 29* (2010). 3

[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. 1, 2, 8, 9

[ZPvBG01] ZWICKER M., PFISTER H., VAN BAAR J., GROSS M.: Surface splatting. In *Proceedings of ACM SIGGRAPH 2001* (2001), pp. 371–378. 4

Appendix

Shader code to approximate the convolution of two parallelograms of axes (`u0`,`v0`) and (`u1`,`v1`) respectively, by a semi axis aligned parallelogram:

```
convolve(vec2 u0, vec2 v0, vec2 u1, vec2& v1,
         vec2& u_out, vec2& v_out) {
  m00 = u0.x*u0.x + v0.x*v0.x + u1.x*u1.x + v1.x*v1.x;
  m10 = u0.x*u0.y + v0.x*v0.y + u1.x*u1.y + v1.x*v1.y;
  m11 = u0.y*u0.y + v0.y*v0.y + u1.y*u1.y + v1.y*v1.y;

  u_out.x = sqrt(m00);
  u_out.y = m10 / u_out.x;
  v_out.x = 0;
  v_out.y = sqrt(m11 - u2.y*u2.y);
}
```