Using Perception-Based Filtering to Hide Shadow Artifacts

Felix Kreuzer*
Supervised by: Michael Hecher†

Institute of Computer Graphics and Algorithms
TU Wien / Austria

Abstract

Combining filtering techniques with shadow mapping is a common tool to simulate soft shadows in real-time applications. A positive side-effect of such approaches is that the filtering also blurs aliasing artifacts caused by low resolution shadow maps, thereby improving the visual quality of the shadow. In this work we investigate the correlation between filter radius and shadow map resolution to optimize computational performance while mostly preserving the visual quality of the soft shadow. We present the results of a user study and offer a ready-to-use function to compute for shadow map aliasing artifacts a respective filter size that makes it unrecognizable.

Keywords: Soft Shadows, Shadow Mapping

1 Introduction

Shadows are crucial for identifying spatial relations between objects. In real-time graphics shadows are frequently implemented by using *shadow maps* [9]. A shadow map stores distances of the visible points in the scene from the point of view of the light source. When the scene is rendered from the camera viewpoint, those values can be compared to the respective distances of the points visible to the camera. If they are farther away form the light source the point is shaded. This basic approach produces hard shadow silhouettes and would theoretically need a sampling density that matches the size of the texel drawn on the screen. Modern implementations commonly average multiple shadow map samples per texel in order to produce softly blurred shadow transitions called penumbrae. Such soft shadows provide a higher degree of visual quality and increased artistic freedom.

Real-time rendering applications require shadow maps to be generated and filtered for every frame and consume lot of performance on the Graphics Processing Unit. Keeping the shadow map's resolution to a minimum reduces memory transfers and generation costs, and increases cache hits.

Our aim is to find a perceptually sound method to determine a minimal shadow map resolution. We exploit the low-pass filtering property of soft shadow penumbrae and introduce a linear function which allows shadow map resolutions and aliasing artifacts to be reduced to a minimum.

We can summarize our contributions to real-time soft shadowing as follows:

- We investigate the relatively complex problem of artifacts generated by arbitrary aligned shadow maps in soft shadow algorithms and break down the huge parameter space, which can be hardly investigated in a user study, into a simplified version.
- A novel approach to dynamically adjusting shadow map sizes for real-time soft shadowing algorithms. By reducing the number of depth samples in a shadow map we can increase performance in shadow map generation since there are less fragments to process and fewer texture lookups. This also improves cache-efficiency because shadow samples are tightly packed and redundant samples are being avoided.
- Our method is flexible and can be applied to several existing soft shadow mapping algorithms.

2 Related Work

Shadow mapping was first introduced by Williams [9] in 1987 and has evolved ever since.

Nowadays a variety of filter based extensions to the traditional shadow mapping algorithm exist such as the following:

- Percentage Closer Filtering (*PCF*) [6] addresses the problem of anti-aliasing in shadow maps. Traditional shadow maps contain depth information, hence prefiltering cannot be achieved directly. The solution is a screen space averaging approach. By increasing the filter size it can be used to simulate soft shadows with a constant penumbra.
- Variance Shadow Maps [3] approximate the depth values by storing mean and variance of the depth distribution. Instead of averaging multiple samples like in PCF, the probability of a fragment being lit is calculated through the moments using Chebshev's inequality. Storing mean and variance of the depth

*falichs@gmail.com
†hecher@cg.tuwien.ac.at

distribution instead of actual depth values allows pre-filtering of the shadow map.

- Convolution Shadow Maps [1] use Fourier expansion to store and reconstruct depth values. This approach allows shadow maps to be pre-filtered but requires a lot of memory and expensive memory transfers to retrieve the Fourier coefficients.

- Exponential Shadow Maps [2] adopt an exponential function to approximate the shadow test. The main benefits are pre-filter-ability and cheap memory and computational costs.

- Percentage Closer Soft Shadowing (*PCSS*) [4] extends the capabilities of PCF by evaluating the filter radius for each fragment based on the distance from shadow occluder to receiver. This approach features a more plausible penumbra behavior in regions where occluder and receiver merge (*contact hardening*).

Hecher et al. [5] present a comparison of some of these algorithms using a comprehensive perceptual study.

In this article we will focus on PCF and PCSS as representative examples of filter based methods, but our findings can be applied to any of the above.

3 Investigating Shadow Map Sampling and Filtering

Our first goal is to find a relation between depth sampling resolution and shadow map filtering. We will further investigate this relation in Section 4 and propose a formula for practical use in Section 6.

Shadow maps are generated per frame by sampling depth values of a scene from the light sources perspective. The sampling process makes use of the hardware graphics pipeline by transforming scene geometry into the perspective view space of the light source and storing the nearest depth values per viewport fragment.

Later, when the scene's per-pixel lighting is calculated, the fragment shader projects each fragment to light-space and queries the shadow map to compare depth values. In case of soft shadows multiple queries are performed in the neighboring vicinity of the fragment in question and filtered by averaging in order to achieve a penumbra effect on shadow borders.

This blurring filter hides high frequency detail on the shadow boundary, leading us to the hypotheses that with increasing softness of the shadow (i.e., a bigger filter size) a less detailed shadow map resolution is required to produce visually sound results. Figure 1 demonstrates this observation by comparing the visual impact of varying filter sizes with different resolutions of the same shadow silhouette. We can see that the resolution required for displaying a perceptually sound soft shadow seems to be directly effected by the filter size.

We will take a closer look on this relation between resolution and filter size in the following sections. Our final goal will be to exploit filtering in order to minimize the shadow map's resolution. This would not only potentially reduce a shadow map's memory footprint, but also improve memory transfer performance during Percentage Closer Filtering, since the probability of hitting the right samples in a cache-segment will be higher.

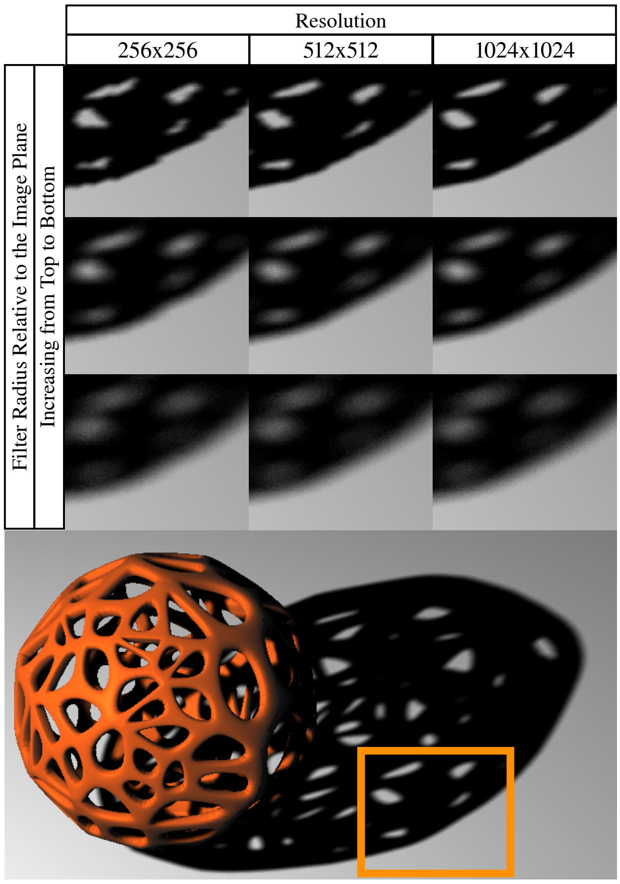


Figure 1: *By doubling the penumbra radius we can reduce the shadow map's resolution by half without noticeable negative impact on the visual quality.*

4 Study Design

Our goal is to hide shadow artifacts by finding an optimal parameterization for shadow mapping based algorithms. This means we need to consider several potentially important parameters related to viewer, light source, shadow map, shadow casting objects and shadow receiving objects. Additionally, because the major target of this research are real-time applications, we also need to take performance into account. In this section we discuss how we can map this relatively complex task to a simple 2D setup which reduces the parameter space greatly and allows us to find a solution to this problem.

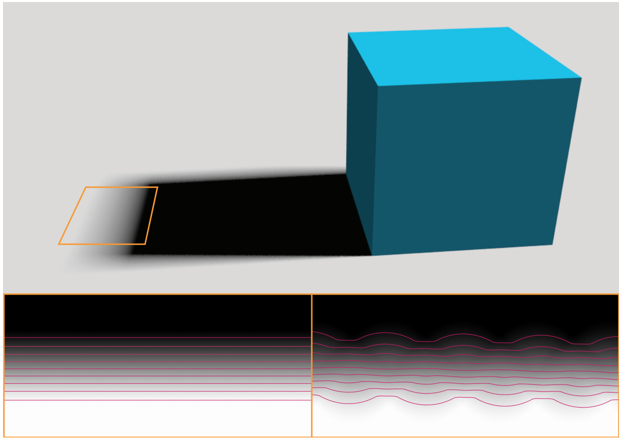


Figure 2: *A comparison of soft shadow iso-contours produced by an artifact-free shadow map (left) and a shadow map with a regular artifact pattern (right).*

4.1 Reducing the Parameter Space

Looking at all the factors involved in computing soft shadows, we find that naively sampling this huge parameter space in a user study would be next to impossible. Too many configurations are necessary to allow for a meaningful evaluation of all the setups that produce perceptually artifact-free images. We therefore need to reduce the complexity of the problem.

Let's continue our investigation from the previous section and take a closer look at the factors influencing the visibility of artifacts. Our first observation from Section 3 is that by increasing penumbra size the visibility of artifacts can be reduced and increasing the resolution of the shadow map has the same effect. Changing any of the other parameters mentioned above might impact the perceivability of shadow artifacts, as they can influence the projected shadow map pixels in the scene (by changing light source, shadow caster and/or shadow receiver position), the projection of the artifact onto the viewer's image plane (by changing the viewer's position) or the contrast of the produced artifacts (by changing the surface color and/or intensity of the light source). So the problem can be separated into three parts. The first part involves the artifact projected from the light source into the scene, the second part how the artifact is projected onto the image plane of the viewer and the third part the contrast and color of the artifact.

Looking at the first part from an analytical standpoint, we make the following observation: The result of computing the soft shadow from a shadow map using filter-based approaches, can be basically seen as a set of iso-contours representing the filtered hard shadow (see Figure 2). These iso-contours form patterns depending on the angle of the light source and the structure of the shadow receiving surface. Artifact-prone and artifact-free solutions will have different contour patterns and we argue that the user evaluates the dissimilarity in their curvature and slope to identify artifacts. Because changing the scene setup (viewer, lights source, objects) can only effect these two factors (curvature and slope), we can simplify the parameter space to said variables. Hence, the problem becomes a simple 2D evaluation of filtering differently sized artifacts with increasing filter size. So we do not have to consider the scene setup at all. The question then is when do these differences become indistinguishable to users?

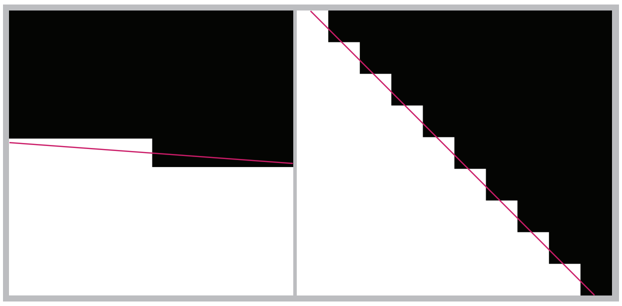


Figure 3: *An illustration of the two silhouette artifact patterns used in our user study. The left image shows a single* step, *the right image a regular* stair *pattern. The red lines represent the actual silhouette of the sampled geometry.*

4.2 Selecting the Stimuli

Now that we were able to greatly simplify the problem, we have to sample the remaining parameters in a meaningful way.

Filter Size We treat the filter size as the dependent variable in our experiments. Our goal is to understand how much an artifact has to be blurred so users cannot recognize it anymore.

Artifact Size Selecting meaningful artifact sizes is actually not that trivial, as we have to consider that the monitors the experiments will be conducted on have a specific pixel resolution. Choosing the artifact size too big or too small can bias the user in his or her decision in whether the original stimulus actually was an artifact (e.g. if the artifact is below pixel size or so big that the filter necessary to hide it needs to be bigger than the screen). We therefore choose the minimal artifact size to be at least five pixels on the screen and at most 5% of the screen size (in out case 30 pixels). In-between we set two additional sample points at 10 and 20 pixels.

Silhouette Patterns The patterns formed by the shadow map depend on the angles of the object silhouette on the shadow map. If a silhouette is horizontally or vertically aligned with the shadow map, artifacts are not visible. In the case of diagonal 45 degree silhouette artifacts are visible at regular intervals, to which we will refer to as *stair pattern*. Cases in-between result in irregular or a mixture

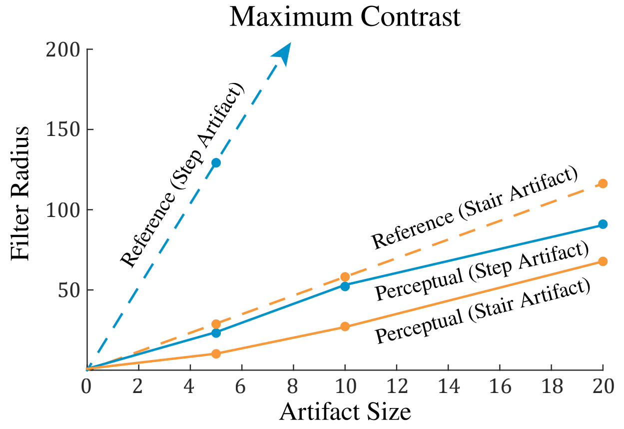


Figure 4: *A comparison between reference filter size and perceptual filter size for the investigated artifact patterns (stair and step). The filter size necessary to hide artifacts from users (perceptual filter) is significantly lower than for the reference filter size.*

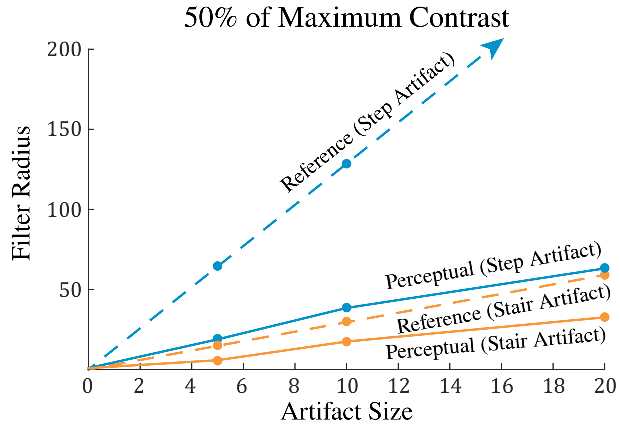


Figure 5: *Reducing the contrast makes it slightly harder to spot artifacts as can be seen in comparison to the results in Figure 4.*

of irregular and regular patterns. We decided to investigate cases where a single artifact (which we will call *step pattern*) is generated and the 45 degree case.

Light-Shadow Contrast The last independent variable we want to investigate is the contrast between lit and shaded areas. We decided to include the worst case scenario, which is the contrast between completely black and white screen pixels. Additionally we reduce the intensity of the lit part by 50% to have an additional sample case.

To summarize, we need to find the right filter size for all artifact size, artifact pattern and light-shadow contrast combinations, resulting in a total of 24 stimuli.

4.3 Task

We decided to use the QUEST procedure [8] to find the threshold at which users can no longer infer from the filtered stimuli whether the original had artifacts in it or not.

We used the Matlab Psycho Toolbox to control the QUESTs. The threshold guess was set to 3% of the artifact size, which is also used as an initial guess and the standard deviation guess. As a probability threshold we used 0.82. The gamma parameter was set to 0.5 and the delta parameter to 0.01. As beta parameter we used 3.5, which was optimized using data obtained by one of the authors performing a beta analysis over 60 trials of the experiment.

5 Evaluation

Due to limited time and resources the study was conducted with a relatively small population of ten users (nine male, one female), all of them were experts in computer-graphics aged 28.2 years on average(standard deviation 3.1 years). Based on the user study's results we can observe the impact of the different parameters on the optimal filter size.

Impact of Artifact Size As already indicated by our observations conducted in Section 3, the artifact size proportionally corresponds to the filter radius that is required to hide them. Figure 4 shows a dependence between filter size and artifact size that is nearly linear. While the stair-artifacts of size 10 are hidden using a filter size of ~ 25 pixels or larger, the 20 pixel artifacts on the other end require a filter of at least 50 pixels.

Impact of Patterns The study shows that the single artifact pattern needs a larger filter size in general to be hidden from the user. Figure 4 shows that the stair patterns of size 10 pixels are perceived as a straight contour when the filter size is at or above 25 pixels. Using the same filter size single step-artifacts of the same size are still identifiable by the users. We assume that the regularity of the 45 degree pattern is beneficial to the user's perception of a straight contour.

Comparison to Reference Filter Size In order to measure the actual benefits of the perceptual approach, we need to compare it to a reference solution. We decided to use the same setup we employed in our user study, with the assumption that in the worst case artifacts will be noticed if at least one pixel differs from the expected outcome. In other words if the rasterization of two filtered solutions, one with and one without artifacts, produces the same image (assuming a typical 8 bit representation for intensities), the perfect user will not be able to spot any artifacts. This corresponds to finding the minimal filter size where this condition is met. We will refer to this filter size as the *reference filter size*.

Impact of Light-Shadow Contrast Reducing the contrast between the lit- and the umbra region makes it slightly harder for the user to identify artifacts as can be seen in Figure 5.

In the next section we will use the data gathered by the user study to fit a linear function which describes the perceptually optimal relation between filter- and artifact size.

6 Applications

While the previous Sections investigated the impact of changes of given parameters on the optimal filter radius, real-time shadowing setups need to approach the problem from the opposite direction, that is, to calculate the optimal parameters for a certain filter radius.

Given an arbitrary scene, we have to assume the worst case of artifact pattern and the worst case of light-shadow intensity to appear, hence the only parameter left to find is the right artifact size which is governed by the shadow map's sampling resolution.

Fitting a linear function to the results shown in Figure 4 (blue line) we get Equation 1, which allows us to calculate the optimal filter radius r to a shadow map by multiplying the pixel size a (artifact size) with the slope c of the linear function.

$$r = a \cdot c \quad (c = 3.47) \tag{1}$$

In order to conduct a practical evaluation of our observations, we implemented a real-time rendering environment and applied Equation 1 to dynamically resize the resolution of a shadow map. We demonstrate the effectiveness of reduced shadow map resolution by applying PCF as well as PCSS filtering. In the case of PCF the filter always needs to have a radius of 3.47 times the pixel size. To use or findings for PCSS we first need to map the penumbra onto the shadow map of the light source to obtain the filter radius. Then the optimal shadow map can be computed by dividing the shadow map with the maximally allowed artifact size. Because the penumbra size can differ within parts of the scene and shadow map artifacts should be avoided, we need to use the maximum shadow map size calculated for each surface point visible for the viewer that lies in shadow.

These rendered results are shown in Figure 7. For each filtering method we show two situations in particular, small and large filter size and compare the reduced shadow map resolution rendering to the unreduced reference rendering side-by-side. While the results look almost identical, we observe an increase in frame-rates on a Geforce GTX 960 GPU by up to 100%.

Poisson Disc Sampling A common approach to reduce the amount of shadow map samples needed to compute soft shadows is to utilize randomly rotated Poisson disc kernels. We will shortly discuss why we expect our findings to be usable in Poisson disc based algorithms as well.

Let's first consider that Poisson disc sampling introduces noise in the soft shadow, which makes it harder for the user to perceive the iso-contours discussed in Section 4.1 (the noise obfuscates the contours). We therefore expect our findings to be compatible with such algorithms, as artifacts should be even less noticeable when they are used.

7 Limitations And Future Work

Although our implementations show promising results there are some noticeable limitations: In cases where the penumbra width is large, high frequency geometry details might be omitted. An example is shown in Figure 6. One solution to overcome this problem, would be to use a second shadow map dedicated for high frequency geometry.

In cases where the penumbra width is very small, i.e. for hard shadows or contact shadows, the penumbra width has to be enlarged, because the required resolution would tend to be infinitely large. Schwärzler et al. describe a possible solution to this problem in their adaptive light source subdivision approach [7].

Due to limited resources we conducted our user study on a small group of ten expert users. We expect to further reduce the shadow map's footprint by questioning inexperienced users.

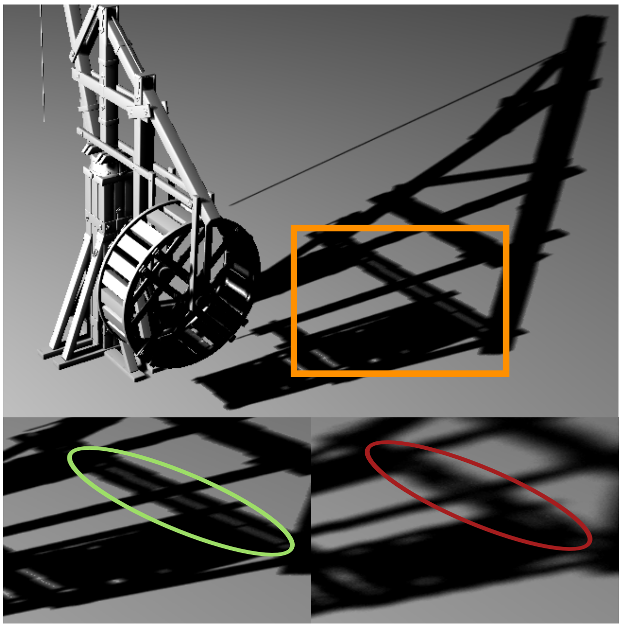


Figure 6: *Lowering the resolution might have the unwanted side-effect of detail being omitted.*

8 Conclusions

We investigated how soft shadow filtering can be exploited to hide shadow mapping related artifacts. Reducing the complex feature space of shadow perception allowed us to design a user study to find out at which point filtered artifacts become unnoticeable to users. By interpreting the results of the user study we were able to describe the connection between shadow filter width and shadow map resolution from a perceptual point-of-view with a function that can be used in practical shadow mapping setups. When we applied this function to common shadow filtering algorithms, we were able to save resources by using perceptually optimized algorithms (as can be seen by comparing the shadow map resolutions and *fps*-timings in Figure 7).

Our findings can be used to dynamically adjust shadow map resolution in real-time, or to calculate a feasible shadow map resolution tailored to a desired penumbra width.

References

[1] Thomas Annen, Tom Mertens, Philippe Bekaert, Hans-Peter Seidel, and Jan Kautz. Convolution shadow maps. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 51–60. Eurographics Association, 2007.

[2] Thomas Annen, Tom Mertens, Hans-Peter Seidel, Eddy Flerackers, and Jan Kautz. Exponential shadow maps. In *Proceedings of graphics interface 2008*, pages 155–161. Canadian Information Processing Society, 2008.

[3] William Donnelly and Andrew Lauritzen. Variance shadow maps. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 161–165. ACM, 2006.

[4] Randima Fernando. Percentage-closer soft shadows. In *ACM SIGGRAPH 2005 Sketches*, page 35. ACM, 2005.

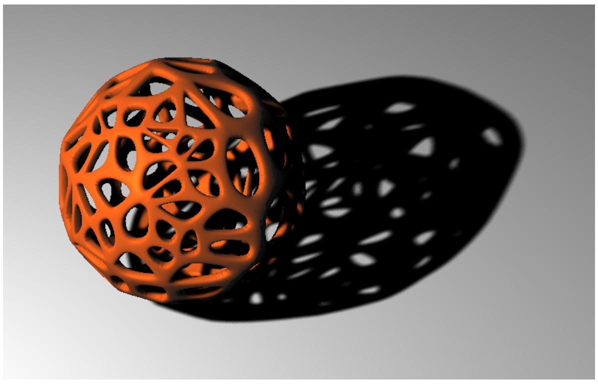
[5] Michael Hecher, Matthias Bernhard, Oliver Mattausch, Daniel Scherzer, and Michael Wimmer. A comparative perceptual study of soft shadow algorithms. *ACM Transactions on Applied Perception*, 11(5):5:1–5:21, June 2014.

[6] William T Reeves, David H Salesin, and Robert L Cook. Rendering antialiased shadows with depth maps. In *ACM Siggraph Computer Graphics*, volume 21, pages 283–291. ACM, 1987.

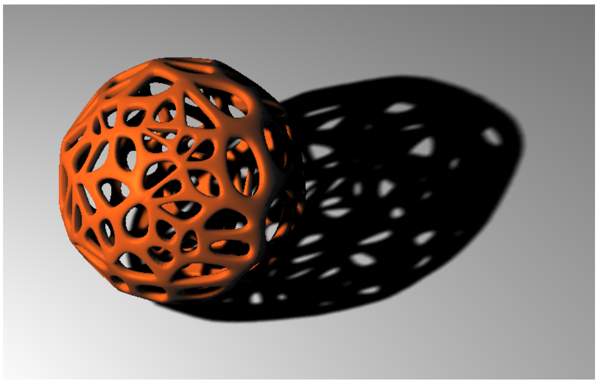
[7] Michael Schwarzler, Oliver Mattausch, Daniel Scherzer, and Michael Wimmer. Fast accurate soft shadows with adaptive light source sampling. In *Proceedings of the 17th International Workshop on Vision, Modeling, and Visualization (VMV 2012)*, pages 39–46. Eurographics Association, November 2012.

[8] Andrew B. Watson and Denis G. Pelli. Quest: A bayesian adaptive psychometric method. *Perception & Psychophysics*, 33(2):113–120, 1983.

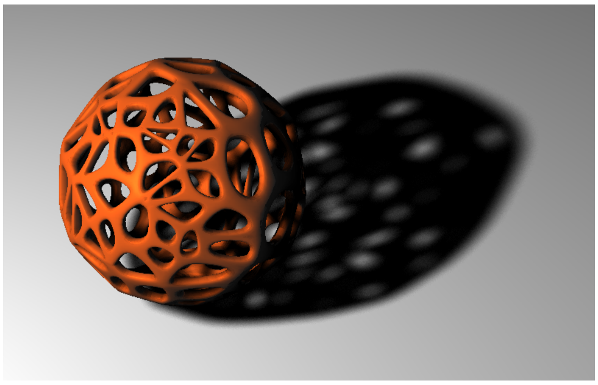
[9] Lance Williams. Casting curved shadows on curved surfaces. In *ACM Siggraph Computer Graphics*, volume 12, pages 270–274. ACM, 1978.



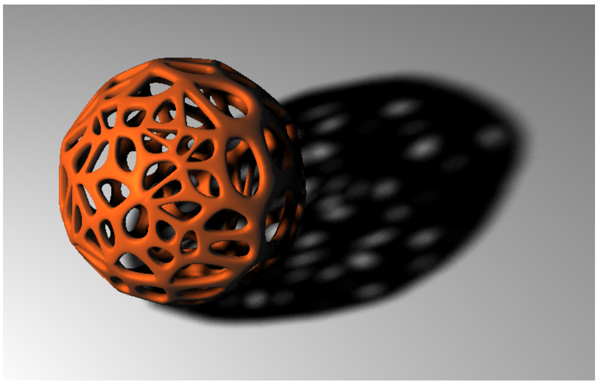
(a) PCF 2048×2048, 193.4 fps



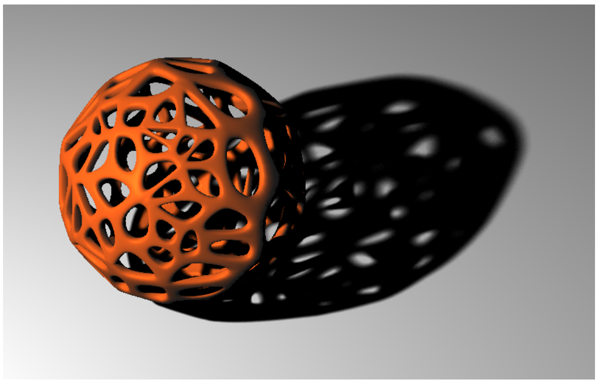
(b) PCF 598×1244, 294.2 fps



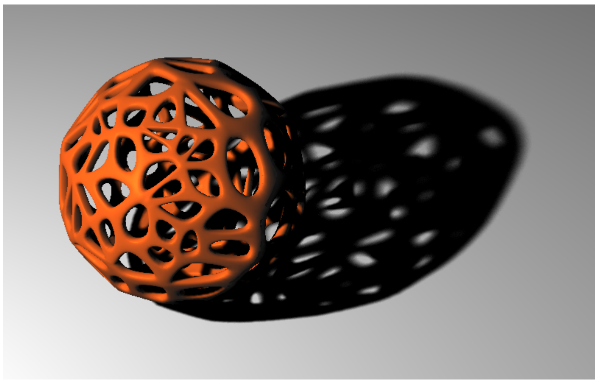
(c) PCF 2048×2048, 157.4 fps



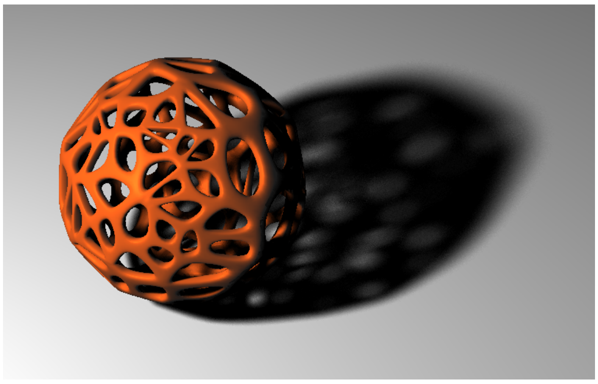
(d) PCF 199×414, 290.5 fps



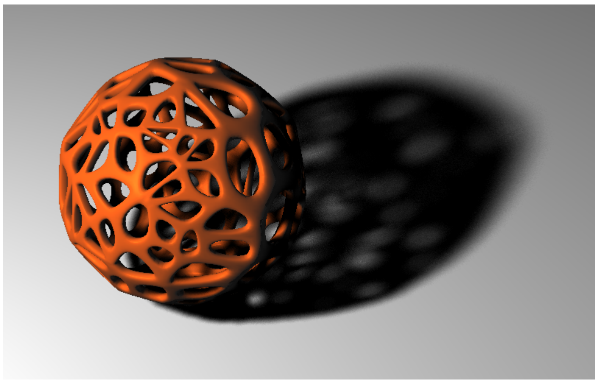
(e) PCSS 2048×2048, 192.4 fps



(f) PCSS 582×1211, 307.5 fps



(g) PCSS 2048×2048, 163.2 fps



(h) PCSS 190×395, 303.4 fps

Figure 7: *Side-by-side comparison of the same scene rendered with large and reduced shadow map resolution. The shadow map resolutions and achieved frame-rates can be seen in the respective captions.*