Revectorization-Based Accurate Soft Shadow using Adaptive Area Light Source Sampling

Márcio C. F. Macedo* Antônio L. Apolinário Jr.†

Federal University of Bahia, Brazil

Abstract

Physically-based accurate soft shadows are typically computed by the evaluation of a visibility function over several point light sources which approximate an area light source. This visibility evaluation is computationally expensive for hundreds of light source samples, providing performance far from real-time. One solution to reduce the computational cost of the visibility evaluation is to adaptively reduce the number of samples required to generate accurate soft shadows. Unfortunately, adaptive area light source sampling is prone to temporal incoherence, generation of banding artifacts and is slower than uniform sampling in some scene configurations. In this paper, we aim to solve these problems by the proposition of a revectorization-based accurate soft shadow algorithm. We take advantage of the improved accuracy obtained with the shadow revectorization to generate accurate soft shadows from a few light source samples, while producing temporally coherent soft shadows at interactive frame rates. Also, we propose an algorithm which restricts the costly accurate soft shadow evaluation for penumbra fragments only. The results obtained show that our approach is, in general, faster than the uniform sampling approach and is more accurate than the real-time soft shadow algorithms.

Index Terms: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture;

1 Introduction

Shadows enhance the realism of computer-generated scenes in several applications, such as movies, games, training simulators, among others. To improve the user's perception of the virtual scene, shadows must be accurate and temporally coherent. However, the interactive or real-time computation of accurate shadows is challenging, mostly due to the computational cost of the shadow rendering problem.

Shadow mapping [41] simplifies the shadow rendering problem, allowing the generation of hard shadows in real-time. However, hard shadows lack realism because they do not simulate the penumbra effect, since the area light source is approximated by a point light source. Hard shadow filtering is useful to fake the penumbra effect [31] and to produce visually plausible soft shadows [11] in real-time. Unfortunately, such soft shadows are not accurate because they were computed on the basis of a single point light source.

Rendering physically correct, accurate soft shadows is computationally expensive because one must evaluate a visibility function over the surface of one or more area light sources. In practice, the area light source is commonly approximated by hundreds of point light sources distributed uniformly over the area light source surface. Also, the visibility function is typically a shadow test performed with shadow mapping [41]. Even with these simplifications, accurate shadow rendering still remains expensive, easily achieving non-interactive performance for scenarios with moderated complexity.

An alternative approach to reduce the computational cost of the accurate shadow rendering relies on the adaptive sampling of the area light source [33]. By adaptively selecting the light source samples according to their contribution in the final rendering, one can reduce the number of samples required for the proper accurate soft shadow rendering. This approach is useful to improve the rendering performance when the camera is far away from the scene, because a few samples are required to provide an accurate visual result, but slower than uniform sampling when the adaptivity does not reduce significantly the number of samples. Moreover, the adaptive approach does not provide temporally coherent, accurate soft shadows and is prone to banding artifacts due to the selection of insufficient samples.

In this paper, we introduce a revectorization-based approach to compute accurate soft shadows. We take advantage of the Revectorization-Based Shadow Mapping (RBSM) [22] to propose a new solution which computes temporally coherent, accurate soft shadows from a few light source samples.

In this sense, our main contributions include:

- A refinement criteria which take advantage of the effect of shadow revectorization to generate less light source samples and shadows with similar accuracy than related work;
- A revectorization-based accurate soft shadow rendering algorithm which evaluates the visibility functions for the light source samples using RBSM;
- A new approach to speed up the computation of accurate soft shadows by discarding non-penumbra fragments from the revectorization-based accurate soft shadow evaluation;
- A temporally coherent adaptive solution for accurate soft shadow rendering;

2 Related Work

Since the rendering equation has been formalized [16, 18], an exhaustive amount of works has been proposed to solve the problem of accurate shadow computation by evaluating a visibility function over an area light source. Here, we classify the works according to the main strategy used to determine the visibility between a surface point and a point light source. For general information about shadow computation, we refer the reader to the book [10].

Ray Tracing: One of the most common algorithms to compute accurate shadows is ray tracing [40]. In this algorithm, for each pixel of the image to be rendered, a primary ray is traced from the view direction to the first surface point visible to the camera, then a secondary ray (called shadow ray) is traced from the hit point to the direction of the light source. If the shadow ray hits the light source, the surface point is out of shadow, otherwise, the surface point is in umbra. A clear drawback of this technique is that ray tracing can reproduce only hard shadows (i.e., shadows without penumbra), because only one secondary ray is sent to evaluate the visibility condition of the surface point. By distributing several shadow rays

*e-mail: marciocfmacedo@gmail.com
†e-mail: antonio.apolinario@ufba.br

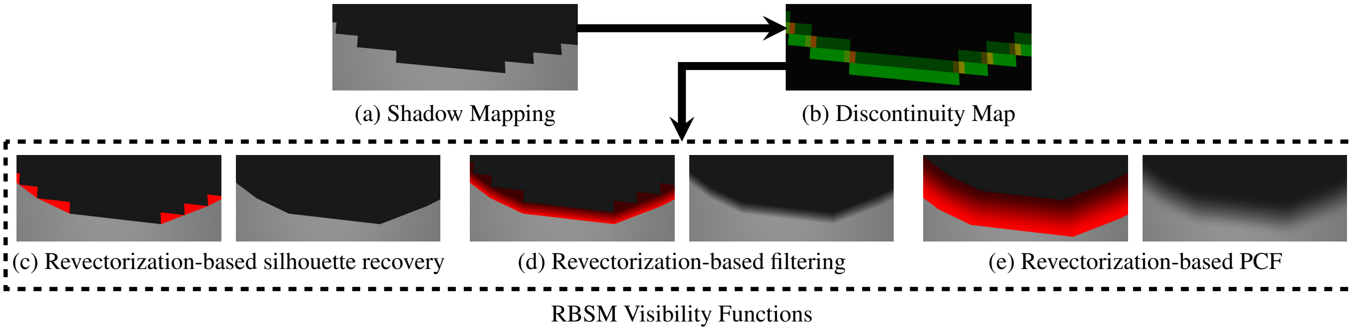


Figure 1: In RBSM, jagged hard shadows produced by shadow mapping (a) are embedded in a discontinuity map (b), which allows the generation of high-quality anti-aliased hard shadows (c) and fake penumbras (d) on the basis of the shadow revectorization theory. To enable control of the fake penumbra size, the traditional percentage-closer filtering (PCF) may be used with one of the RBSM visibility functions (e). Images were generated for the Sphere model using a 1024^2 shadow map resolution.

per area light source and averaging their results, one can render soft shadows (i.e., shadows with penumbra) with ray tracing [8]. One problem with this approach is that the use of regular or random sampling patterns to evaluate the area light source generates aliasing or noise artifacts along the shadow boundary. Since then, several strategies (e.g., stochastic sampling [7], stratified sampling [24], uniform jitter sampling [26], Poisson disk sampling [39], adaptive sampling [13,23], line sampling [4]) have been used with ray tracing to alleviate aliasing, each one of them which its own advantages and drawbacks [29,30]. However, regardless of the sampling strategy used, ray tracing demands seconds to produce accurate shadows, making this technique unsuitable to generate shadows for interactive or real-time applications.

Shadow Volumes: An alternative to compute accurate shadows faster than ray tracing relies on the use of shadow volumes [9]. For a scene described by polygons, this technique projects a ray from a point light source for each vertex located at the object's silhouette. Then, these projections are combined into a single polygon mesh called shadow volume. A surface point that lies inside the shadow volume is determined to be in shadow. Similar to ray tracing, the original shadow volume algorithm is able to compute only hard shadows. Approaches based on multiple shadow volume evaluation [6] and penumbra wedges [1,2,12,19,21] have been proposed to generate accurate soft shadows much faster than the alternatives based on ray tracing, at the cost of prohibitively large memory footprints. In this case, more accurate and faster solutions do exist, such as [25,38].

Shadow Mapping: A common approach to generate hard shadows in real-time is shadow mapping [41]. In shadow mapping, the scene is rendered twice, first from the light source viewpoint and then from the camera viewpoint. The illumination condition of each surface point is determined by a depth comparison between the depth of the camera-visible fragment as seen from the light source and the depth of the nearest blocker stored in the shadow map texel corresponding to that fragment.

Many algorithms (e.g., [11,28,35,43]) have adapted the shadow mapping algorithm to produce visually plausible soft shadows in real-time. While these algorithms have been used for games and other applications where real-time performance is required, they do not produce accurate soft shadows, typically suffering from aliasing, light leaking and banding artifacts.

To generate accurate soft shadows from the shadow mapping representation, many techniques have sampled the area light source, generating several hard shadows in real-time and averaging them using the accumulation buffer [14]. These algorithms handle scenarios with a linear light source [15], static scenes [37], and dynamic scenes [36]. All of these techniques generate high-quality, accurate soft shadows at only interactive or non real-time frame rates.

Closest to our solution, the adaptive sampling solution proposed in [33] uses a screen-space refinement criteria to determine how much light source samples are needed to generate accurate soft shadows. Although this technique works well when the camera is far away from the scene, because a few light source samples are needed to provide visually accurate soft shadows, the refinement step consumes too much processing time to determine the number of samples, making this approach inefficient when the camera is relatively close to penumbra regions, where a large number of light source samples are required to provide accurate soft shadows. Here, we propose a new solution based on RBSM which requires only a few light source samples to provide accurate soft shadows independently of the camera position or scene configuration. Moreover, temporal coherency and performance strategies are employed to enhance the robustness of our solution.

The main advantage of RBSM compared to related work is that this technique generates hard shadows of higher quality than shadow mapping, while keeping almost the same memory consumption and demanding a slightly increased processing time. The shadow silhouette mapping approach [34], for instance, has the same goal of RBSM, however, is slower than the non-real-time shadow volume technique [9], while its performance depends on the resolution of the scene geometry. More recent techniques [20,27,42] speed up the computation of accurate hard shadows, but increasing memory consumption and processing time of the shadow mapping.

3 Revectorization-Based Shadow Mapping

Shadow mapping [41] is a technique which generates hard shadows in real-time, but suffers from aliasing artifacts along shadow boundaries because the shadow map has finite resolution (Fig. 1-(a)). Inspired in [5], RBSM [22] reduces shadow aliasing artifacts generated with shadow mapping by revectorizing jagged hard shadows. To do so, the region of transition between illuminated and shadowed regions, exactly the region where the jagged hard shadows are located, is represented by discontinuities (Fig. 1-(b)). Similarly to the morphological anti-aliasing [17], these discontinuities are oriented and normalized towards the end of the shadow edge, allowing the definition of different visibility functions to determine the shadow revectorization (Fig. 1-(c, d, e)).

Let us consider the surface point $\mathbf{p}$ distant to the light source by $\mathbf{p}_z$. Also, let us assume $\mathbf{t}_{x,y}$ the shadow map texel positioned at the 2D position x,y and $z(\mathbf{t}_{x,y})$ a function which retrieves the depth of the blocker of $\mathbf{p}$ as seen from a point light source. The shadow test $s(\mathbf{p}_z, z(\mathbf{t}_{x,y}))$ is a binary visibility function defined as [41]

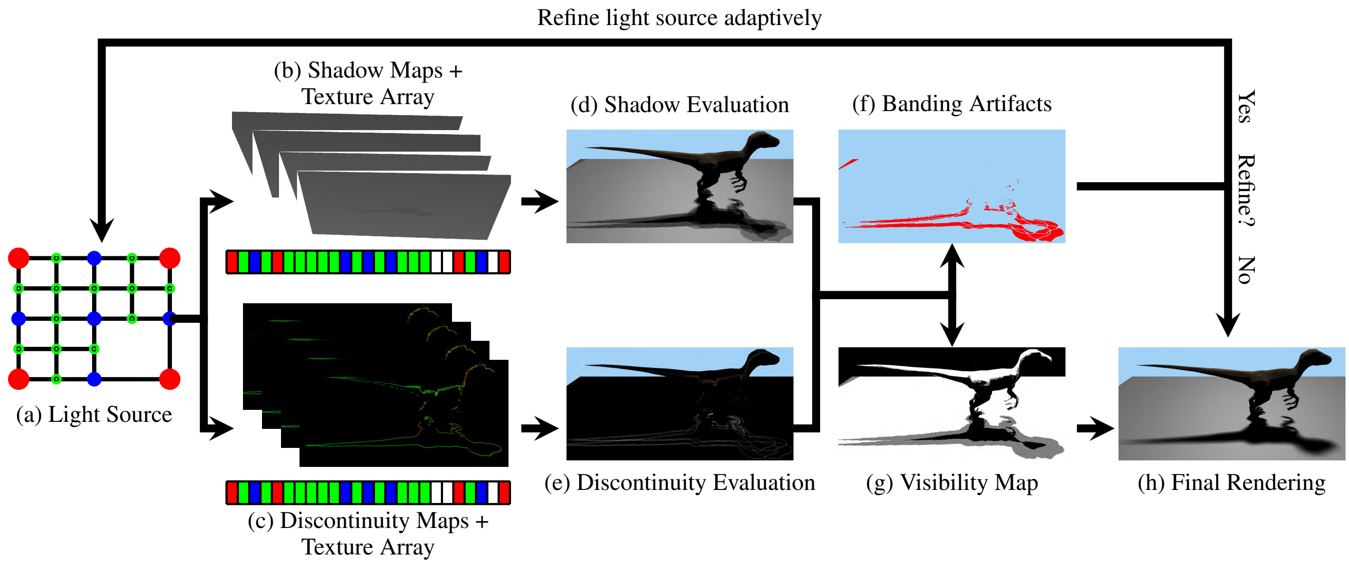


Figure 2: Given an area light source (a), we first generate four shadow (b) and discontinuity maps (c) for the neighbours point light sources located at the light source corners and store those maps into separate texture arrays. Then, the set of shadow and discontinuity maps (b, c) are evaluated (d, e) to detect the presence of banding artifacts (f) and build a visibility map (g) in the camera view. According to a refinement criteria, we determine whether the area light source must be adaptively refined and the algorithm reiterated for each four new neighbour samples. Otherwise, the accurate soft shadow is computed (h) on the penumbra fragments detected with the visibility map. Images were generated for the Raptor model using a 1024^2 shadow map resolution. The light source in (a) is refined to the third level of the adaptive structure, where each sample color represents a different level in the adaptive structure. As can be seen in (b, c), shadow and discontinuity maps are stored in the texture arrays according to the position of the sample (indicated by the colors) in the light source.

$$s(\mathbf{p}_z, z(\mathbf{t}_{x,y})) = \begin{cases} 0 & \text{if } \mathbf{p}_z > z(\mathbf{t}_{x,y}), \\ 1 & \text{otherwise,} \end{cases} \tag{1}$$

where 0 indicates that the point $\mathbf{p}$ is in umbra and 1 otherwise.

Because the shadow map has finite resolution, the shadow test produces jagged hard shadows, as shown in Fig. 1-(a). To revectorize them, we need to find discontinuities in the scene. A discontinuity is located in the jagged shadow boundary, where the shadow tests are different between neighbour shadow map texels (Fig. 1-(b)).

To locate discontinuities in the camera view, let us define the shadow test evaluation for a 4-connected neighbourhood with respect to a shadow map texel as $\mathbf{N}$

$$\mathbf{N} = \left[s(z(\mathbf{t}_{x-o,y})), s(z(\mathbf{t}_{x+o,y})), s(z(\mathbf{t}_{x,y+o})), s(z(\mathbf{t}_{x,y-o}))\right], \tag{2}$$

where o is an offset value equivalent to one shadow map sample and $\mathbf{p}_z$ is the same for every 4-connected shadow test evaluation.

Given the shadow test (1) and the neighbourhood evaluation (2), discontinuity $\mathbf{d}$ is formally defined as the absolute difference in the shadow tests of a shadow map texel and its 4-connected neighbors [22]

$$\mathbf{d} = \left[|\mathbf{N}_1 - s|, |\mathbf{N}_2 - s|, |\mathbf{N}_3 - s|, |\mathbf{N}_4 - s|\right]. \tag{3}$$

After the discontinuity computation for every visible fragment in the camera view (Fig. 1-(b)), we compute an oriented and normalized distance for each fragment belonging to a discontinuity to both shadow edge ends according to a traversal over the discontinuity map. Then, we use this relative position of the fragment to determine the shadow revectorization. In fact, RBSM has a visibility function which uses screen-space information (e.g., discontinuity location, fragment's distance to the shadow boundary, shadow aliasing shape) to estimate the new, anti-aliased shadow boundary for hard (Fig. 1-(c)) and filtered hard shadows (Fig. 1-(d)). Moreover, RBSM is extensible such that the traditional percentage-closer filtering (PCF) technique [31] may be used to take the average results of the RBSM visibility function over a sampled region to produce high-quality revectorization-based filtering over a user-defined penumbra size (Fig. 1-(e)) [22].

For accurate soft shadow computation, rather than naively replacing shadow mapping by RBSM when evaluating the visibility function, we propose a more efficient solution which takes advantage that RBSM might require less light source samples than shadow mapping to provide anti-aliased shadows. On the basis of this assumption, we propose a new algorithm which uses the concepts of discontinuity and shadow revectorization to adaptively select the appropriate number of light source samples required to approximate the area light source and generate artifact-free accurate soft shadows.

4 Revectorization-Based Accurate Soft Shadows

In this section, we describe our approach to compute accurate soft shadows. Our approach is built upon the adaptive solution proposed in [33], but we take advantage of RBSM in the refinement criteria to select less light source samples than [33]. An overview of the proposed algorithm is presented in Fig. 2 and a high-level pseudocode is listed in Algorithm 1.

4.1 Adaptive Light Source Sampling

Let us define the area light source L as an adaptive structure where each node consists of a quad Q formed by four neighbour point light sources. The main goal of the adaptive sampling is to generate only the light source samples $\mathbf{l} \in L$ which will contribute significantly to the final soft shadow appearance, generating visually accurate soft shadows. Hence, the light source refinement criteria must be view-dependent, considering whether the neighbour samples produce

Algorithm 1 Revectorization-based accurate soft shadowing

1: GB ← RENDERGBUFFER;
2: **for** each new quad Q of L with index q **do**
3:     **for** each new light source sample $\mathbf{l}[i]$ of Q **do**
4:         SM[i] ← RENDERSHADOWMAP($\mathbf{l}[i]$);
5:         DM[i] ← RENDERDISCONTINUITYMAP(SM[i], GB);
6:     **end for**
7:     SS ← COMPUTESHADOWSUM(SM, GB);
8:     SD ← ISSOFTDISCONTINUITY(DM, GB);
9:     VM_q ← UPDATEVISIBILITYMAP(SS, SD, VM_{q-1}, GB);
10:    **if** HASBANDINGARTIFACT(SS, SD, GB) **then**
11:        REFINEADAPTIVESTRUCTURE(L);
12:    **end if**
13: **end for**
14: RENDERACCURATESOFTSHADOW(SM, DM, VM, L, GB);

artifact-free soft shadows in the final rendering [33].

We start the light source sampling by building the first level of the adaptive structure, where a single leaf node represents the quad formed by the light source samples located at the corners of the area light source (red circles in Fig. 2-(a)).

To use the RBSM (Section 3) in our refinement criteria, we need to compute both shadow test (Eq. 1) and discontinuity (Eq. 3) for each new point light source sample inside the area light source quad. To do so, we render the scene from the viewpoint of the point light source and store the depth buffer as a shadow map [41] (Fig. 2-(b)). Also, we render the scene from the camera viewpoint, compute the discontinuity and store it in a discontinuity map (Fig. 2-(c)). As shown in Fig. 2, both maps are stored in separate texture arrays, whose sizes are equivalent to the maximum number of samples that can be selected from the area light source (Lines 3-6 of Algorithm 1). To optimize the discontinuity map rendering, world-space position and the surface's normal of the visible fragments in the camera viewpoint are stored in a G-Buffer [32], which is used to guarantee that the discontinuities are computed for visible fragments only (Lines 1 and 5 of Algorithm 1).

After the shadow and discontinuity map rendering, we need to determine whether the samples located in the same quad are sufficient for accurate soft shadow rendering. To do so, we project both shadow and discontinuity maps of the four neighbour samples into the same camera view and compare them (Fig. 2-(d, e)) to detect whether banding artifacts are produced by the use of those samples (Fig. 2-(f)). This comparison is done in a two-pass strategy with the scene rendered from the camera viewpoint.

In the first pass, for each fragment $\mathbf{p}$ in the camera view, we estimate the shadow sum $\text{SS}(\mathbf{p}_z)$, which is the sum of the four shadow test values computed from the neighbour four shadow maps of the light quad (Fig. 2-(d) and Line 7 of Algorithm 1)

$$\text{SS}(\mathbf{p}_z) = \sum_{i=1}^{4} s(\mathbf{p}_z, z_i(\mathbf{t}_{x,y})). \tag{4}$$

Additionally, we label a fragment as *soft discontinuity* if at least D non-null discontinuities (i.e., $\mathbf{d} \neq 0$) were computed for the fragment (Fig. 2-(e) and Line 8 of Algorithm 1). Considering $D \in [1,4]$, $D = 1$ generates a really small number of samples for rendering, making the approach susceptible to banding artifacts. On the other hand, $D = 4$ generates several samples, as a few fragments are classified as discontinuity for all the four neighbour light source samples. $D = 2$ or 3 generates a moderate number of samples. We have used $D = 2$ for all the scenarios shown in this paper because this value generates less samples than $D = 3$, while being much less susceptible to banding artifacts than $D = 1$.

In the next pass, both shadow sum and fragment classification (Fig. 2-(d, e)) are used to locate the fragments that potentially produce banding artifacts in the camera view (Fig. 2-(f), Line 10 of Algorithm 1). Based on the previously computed shadow sum (4), fully lit (i.e., $\text{SS}(\mathbf{p}_z) = 4$) and fully shadowed fragments (i.e., $\text{SS}(\mathbf{p}_z) = 0$) are discarded from rendering because they are not located in the penumbra and cannot cause banding artifacts. Fragments classified as *soft discontinuity* are discarded from rendering as well, because RBSM will guarantee high-quality anti-aliasing for them. The remaining fragments (whose shadow sum lies between 1 and 3) are compared against their 8-connected neighbour fragments in the camera view. If the fragment has at least one neighbour fragment which has a different shadow sum or which is a *soft discontinuity*, the fragment is discarded. The only fragments rendered in the scene are the ones whose shadow sums state that the fragments are in the penumbra and the shadow sums are the same for all the 8-connected neighbours. That is the case of the fragments located in penumbra regions which are not sufficiently smooth, due to the high distance between light source samples (and their shadow maps). As shown in Fig. 2-(f), those fragments produce banding artifacts in the final rendering, rather than a single, smooth penumbra region [33].

Hardware occlusion query [3] is used to check if a single pixel was rendered on the screen. If this condition is true, the area light source is further refined according to the adaptive structure (Line 11 of Algorithm 1). Then, the algorithm is iterated for the new light quads (Fig. 2).

To optimize the performance of our solution, while we perform the light source sampling, we build and update a visibility map VM. This map is a texture which stores the final illumination condition of each fragment (i.e., whether the fragment is lit, penumbra or umbra) (Fig. 2-(g)). With such a map, we are able to restrict the costly accurate soft shadow rendering for penumbra fragments only.

The algorithm to compute the visibility map is fairly simple, yet effective. First, we clear the visibility map, indicating that no classification has been assigned to any visible fragment. Then, we use the estimated shadow sum to update the stored visibility condition of the fragment. Let us redefine the shadow sum as $\text{SS}_q(\mathbf{p}_z)$ and the visibility map as $\text{VM}_q(\mathbf{p}_z)$, where q refers to the quad index. For the first quad of the adaptive structure, the visibility classification $\text{VM}_0(\mathbf{p}_z)$ of the fragment $\mathbf{p}$ is computed as

$$\text{VM}_0(\mathbf{p}_z) = \begin{cases} \text{umbra} & \text{if } \text{SS}_0(\mathbf{p}_z) = 0, \\ \text{penumbra} & \text{else if } 1 \leq \text{SS}_0(\mathbf{p}_z) \leq 3, \\ \text{lit} & \text{otherwise.} \end{cases} \tag{5}$$

Additionally, we define $\text{VM}_0(\mathbf{p}_z)$ as penumbra for the fragments classified as *soft discontinuity*.

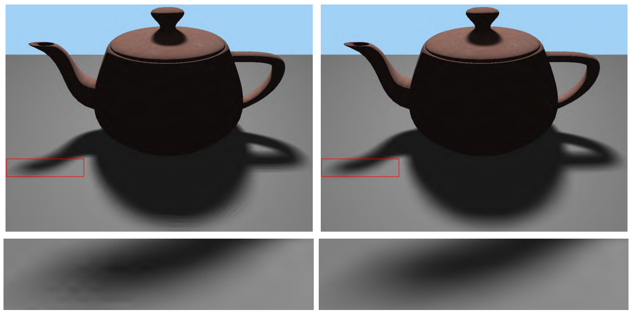
For the next quad, assuming that the adaptive structure has more than one level, we update the visibility map by classifying the fragment as penumbra if there is a difference between the illumination condition previously estimated in the visibility map and the one given by the current shadow sum. In other words

$$\text{VM}_q(\mathbf{p}_z) = \begin{cases} \text{penumbra} & \text{if } \text{VM}_{q-1}(\mathbf{p}_z) = \text{lit and } \text{SS}_q(\mathbf{p}_z) = 0, \\ & \text{or } \text{VM}_{q-1}(\mathbf{p}_z) = \text{umbra and } \text{SS}_q(\mathbf{p}_z) = 4. \end{cases} \tag{6}$$

Finally, in the final rendering step (Fig. 2-(h)), after the refinement criteria has been satisfied, we access the visibility map to determine the visibility condition of the fragment in the camera view. Lit and umbra fragments are illuminated accordingly, and for penumbra fragments only, we proceed with the computation of the final soft shadow intensity.

4.2 Final Rendering

Given the n light source samples $\mathbf{l}$ distributed over the surface of the area light source L, the final soft shadow intensity of a point $\mathbf{p}$



(a) Revectorization-based filtering

(b) Revectorization-based PCF

Figure 3: (a) For a relatively large penumbra size, the use of the revectorization-based filtering visibility function generates banding artifacts for a few light source samples. (b) The control over the filter size provided by revectorization-based PCF allows the generation of artifact-free soft shadows, at the cost of increased processing time. Images were generated for the Teapot model using a 1024^2 shadow map resolution and 25 light source samples.

(Fig. 2-(h), Line 14 of Algorithm 1) can be computed according to the visibility function $f(\mathbf{p})$

$$f(\mathbf{p}) = \frac{\sum_{i=1}^{n} \omega_i v(\mathbf{p}, \mathbf{l}_i)}{\sum_{i=1}^{n} \omega_i}, \tag{7}$$

where $v(\mathbf{p}, \mathbf{l}_i)$ denotes the visibility function of the RBSM technique [22] for a point $\mathbf{p}$ with respect to the point light source $\mathbf{l}_i$, and the function $f(\mathbf{p}) \in [0,1]$ estimates how much of the area light source is visible to the point $\mathbf{p}$, where $f(\mathbf{p})$ = 0 indicates that the entire area light source is not visible to $\mathbf{p}$, and $f(\mathbf{p})$ = 1 indicates the full visibility of the area light source. ω_i is the weight assigned to the point light source $\mathbf{l}_i$ which compensates for the irregular distribution of samples, computed as [33]

$$\omega = \frac{1}{(2^{\alpha}+1)^2}, \tag{8}$$

where α is the level of the adaptive structure.

As stated in Section 3, RBSM is the basis of three distinct techniques which produce different shadow outputs. Here, we assume $v(\mathbf{p}, \mathbf{l}_i)$ as the RBSM filtering visibility function (Fig. 1-(d)) because the use of filtering for shadow revectorization is efficient to solve banding artifacts [22].

We could compute (7) in n shader passes, evaluating $v(\mathbf{p}, \mathbf{l}_i)$ per sample in each pass, and using the accumulation buffer [14] to store the accumulated soft shadow intensity. Since we store the n shadow and discontinuity maps into two texture arrays (Fig. 2-(b, c)), we are able to compute (7) and evaluate the RBSM visibility function for all the light source samples in a single pass on the shader, further saving many read/write operations that would be needed by the accumulation buffer.

4.3 Temporally Coherent Soft Shadow Computation

One alternative to further reduce processing time and the number of light source samples selected relies on the reduction of the viewport size used for occlusion query during the adaptive refinement (Section 4.1). Unfortunately, the algorithm becomes prone to banding artifacts due to the insufficient number of samples. Rather than using the revectorization-based filtering technique as visibility function in (7), the revectorization-based PCF can be used to solve this problem (Section 3), but the performance drops considerably when using this technique. Another problem with the viewport size reduction is that a fixed reduction factor produces incoherent soft shadows as the camera moves in the scene. In this way, we can use an adaptive approach to estimate this reduction factor to produce temporally coherent soft shadows. Also, we need to determine whether the revectorization-based PCF is useful to solve the banding artifacts generated from the viewport size reduction.

As can be seen in Fig. 1-(d), the revectorization-based filtering technique is well suited for scenarios with small penumbra sizes because it adds filtering for a limited extension of the anti-aliased shadow. However, for large penumbra sizes, several light source samples are still required to generate artifact-free soft shadows, because the small filter size of the filtering technique does not solve the banding artifacts in the penumbra (Fig. 3-(a)). A more appropriate alternative for large penumbra sizes is the revectorization-based PCF technique, which requires a few light source samples to provide high-quality soft shadows (Fig. 3-(b)). In this sense, according to the area light source and the penumbra size, each one of the revectorization-based techniques is more adequate for accurate soft shadow rendering.

To compute the appropriate window size for occlusion query automatically, we estimate such value according to the RBSM technique used and the current level of the adaptive structure. We draw this approach from the observation that the window size reduction may change from adaptive structure level because, as long as the structure is refined, we can relax the criteria to guarantee that a small number of samples will be used for rendering. Let us assume ws the original window size, ws_{RSMSS} and ws_{RPCF} the window sizes used for the revectorization-based filtering and revectorization-based PCF techniques, respectively. In this paper, we have used the following window sizes

$$ws_{\text{RSMSS}} = \begin{cases} ws & \text{if } 0 \le \alpha \le 1, \\ \frac{ws}{2} & \text{else if } 2 \le \alpha \le 3, \\ \frac{ws}{4} & \text{otherwise.} \end{cases} \tag{9}$$

$$ws_{\text{RPCF}} = \begin{cases} \frac{ws}{4} & \text{if } 0 \le \alpha \le 1, \\ \frac{ws}{6} & \text{else if } 2 \le \alpha \le 3, \\ \frac{ws}{8} & \text{otherwise.} \end{cases} \tag{10}$$

As we aim to generate a few light source samples, the idea of keeping the adaptive structure built from the previous frame and refining or condensing it in the next frame did not improve the performance of the algorithm.

5 Results and Discussion

In this section, we evaluate the soft shadow techniques in terms of visual quality and performance. In our experimental setup, time usage was evaluated in an Intel Core™ i7-3770K CPU (3.50 GHz), 8GB RAM, and an NVIDIA GeForce GTX Titan X graphics card. We compare our revectorization-based (RB) adaptive sampling with other sampling strategies, namely the uniform sampling of the area light source (using 289 samples, as suggested in [33]) and the adaptive sampling solution proposed in [33]. Also, we compare our approach with two techniques from the field of real-time soft shadow mapping: the Percentage-Closer Soft Shadows (PCSS) [11], which is one of the most traditional real-time soft shadow techniques, and Moment Soft Shadow Mapping (MSSM) [28], one of the most recent soft shadow mapping techniques. To provide a fair comparison between the adaptive sampling of [33] and ours, we have used their solution with a reduction over the window size for occlusion query by a factor of 4 and PCF [31] to compensate the banding artifacts. Their solution is always slower than ours when using the same window size for both occlusion query (during adaptive sampling, Section 4.1) and output resolution (during final rendering, Section 4.2). All images were generated by a rectangular area light source.

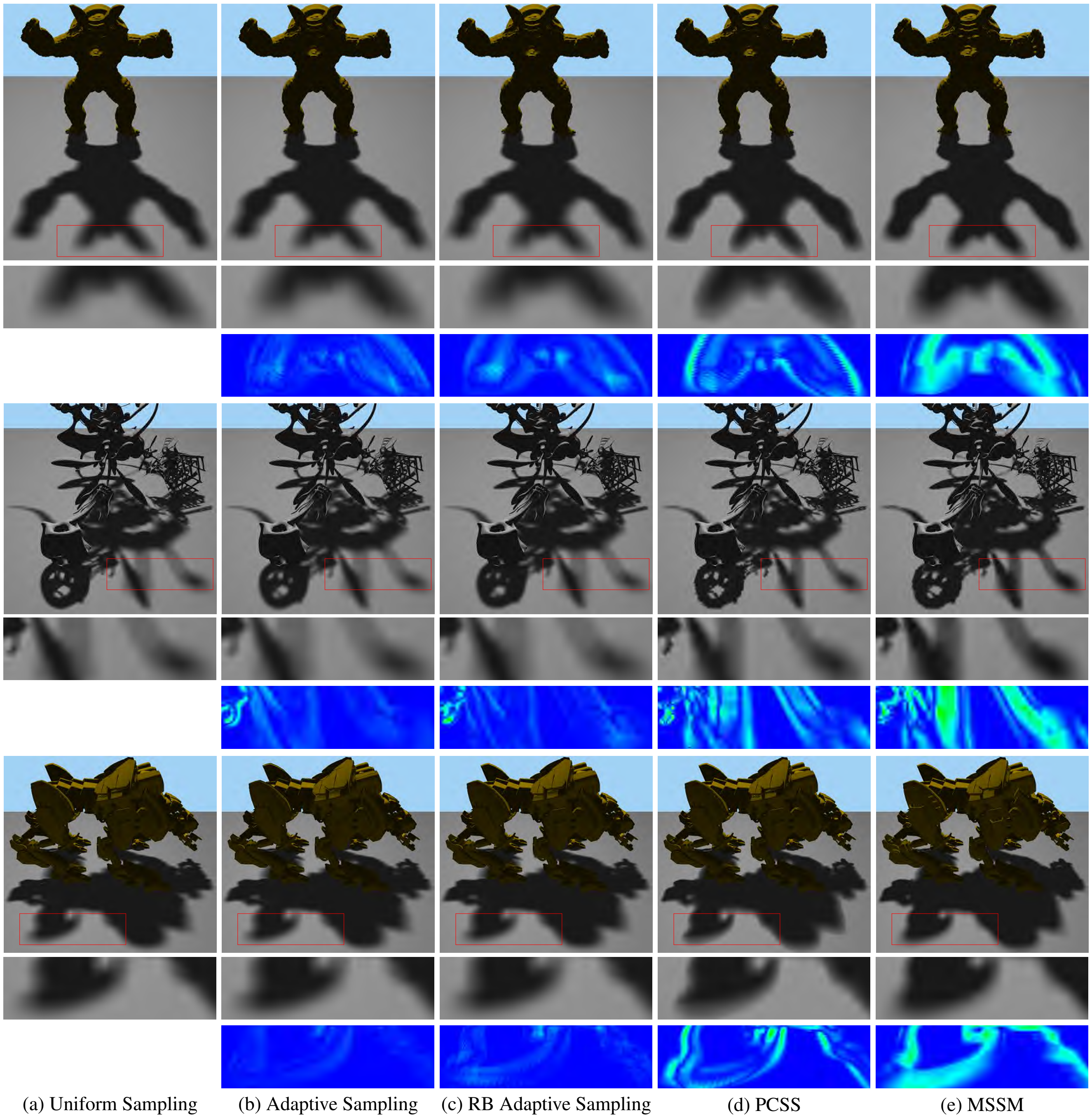


Figure 4: Accurate soft shadows produced by different techniques. For uniform sampling (a), we have used 289 light source samples for all the scenarios. Adaptive sampling (b) has selected 47, 134 and 246 light source samples for Armadillo (top), YeahRight (middle) and QuadBot (bottom) models. Our RB adaptive sampling (c) has used only 25, 62 and 63 light source samples for the same scenarios, respectively. The real-time soft shadow techniques (d, e) use a single point light source sample. The false color visualizations show the difference between the shadows obtained with uniform sampling (which uses the largest number of samples) and the other techniques. Images were generated using a 1024^2 shadow map resolution.

Both PCF and revectorization-based PCF use the same kernel size of 2×2. We refer the reader to our accompanying video to see the temporal stability of our approach.

5.1 Rendering Quality

As shown in Fig. 4, our revectorization-based adaptive sampling provides high-quality, accurate soft shadows (Fig. 4-(c)), needing a few light source samples to achieve such visual quality. We require about 4-11 times less samples than the uniform sampling approach (Fig. 4-(a)) and 2-4 times less samples than the adaptive sampling

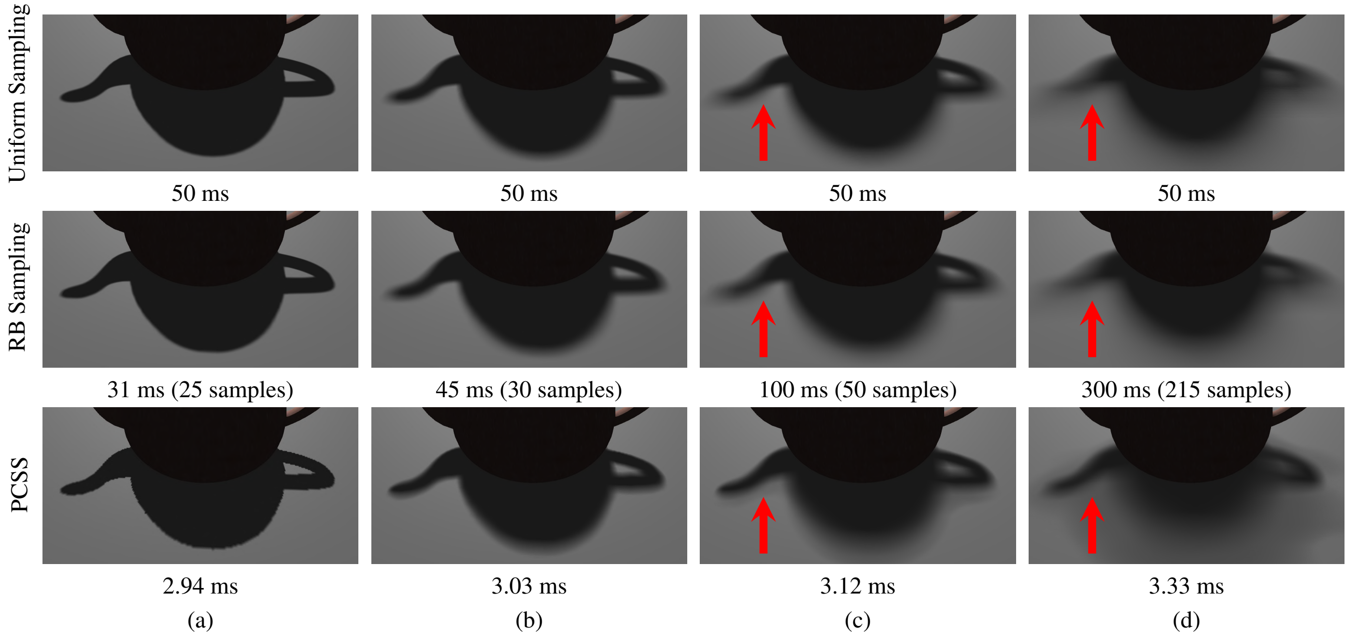
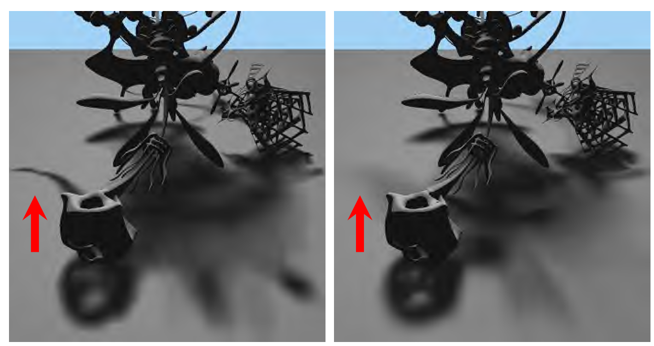


Figure 5: A performance/visual quality comparison between different soft shadow techniques under distinct penumbra sizes. For small penumbra sizes (a, b), our approach is generally faster than the uniform sampling approach. The opposite occurs for large penumbra sizes (c, d), which demands an increased number of samples to minimize the banding artifacts. A real-time soft shadow approach is able to render visually plausible soft shadows for small penumbra sizes (a, b), but deviates from the accurate soft shadow under large penumbra sizes (see the region pointed by the red arrows in c, d). Images were generated for the Teapot model using a 1024^2 shadow map resolution.

approach proposed in [33] (Fig. 4-(b)) to achieve high visual quality. Although the real-time soft shadow techniques (Fig. 4-(d, e)) generate visually plausible soft shadows, the penumbra size is estimated incorrectly and some details of the shadow are lost due to the approximation of the area light source by a single point light source.

An analysis of the influence of the penumbra size over the proposed adaptive sampling algorithm can be seen in Fig. 5. For small penumbra sizes (Fig. 5-(a, b)), our approach is faster than the uniform sampling approach, while generating accurate soft shadows from a few light source samples. Real-time soft shadow algorithms, in this case, tend to produce visually plausible soft shadows, but aliasing artifacts can be seen in the shadow silhouette due to the use of an insufficient shadow map resolution (Fig. 5-(a)). As long as the penumbra size increases (Fig. 5-(c, d)), more light source samples are needed to effectively suppress banding artifacts. In this scenario, while being able to generate accurate soft shadows, our approach may be slower than the uniform sampling approach mainly because of three factors: 1. the revectorization is slower than the traditional shadow mapping, although it provides improved visual quality, 2. the adaptive refinement provides an additional cost to the final rendering time, while the uniform approach does not have such a step, 3. in the case where the penumbra fills much of the screen-space available, the use of the visibility map does not discard a high number of fragments from the final rendering evaluation. As shown in Fig. 5-(c, d), for large penumbra sizes, real-time soft shadow algorithms are able to generate soft shadows with low processing time, but cannot generate accurate soft shadows (as pointed by the red arrows in Fig. 5-(c, d)). Indeed, such a difference is mainly visible for situations such as the one shown in Fig. 6, where the umbra region disappears entirely in penumbra. In these cases, the approximation of the area light source by a single point light source does not provide enough information for the sampling and rendering of those fine details of the penumbra.

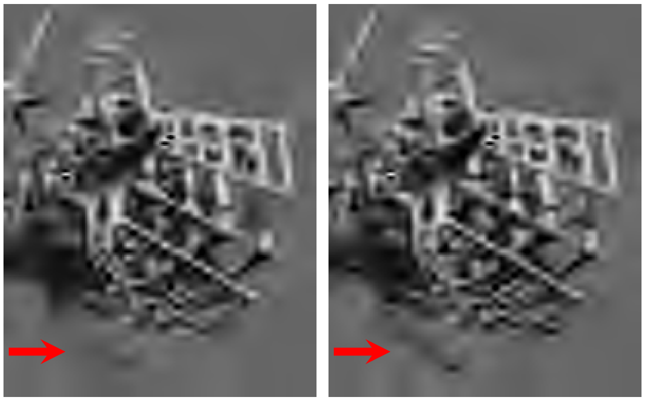


(a) PCSS (b) RB Adaptive Sampling

Figure 6: For complex, large penumbra sizes, common real-time soft shadow techniques (a) fail to produce near accurate soft shadows (b) (see the region pointed by the red arrows). Images were generated for the YeahRight model using 1024^2 shadow map resolution.

Since we compute accurate soft shadows on the basis of shadow maps, we may suffer from subsampling artifacts if a low-resolution shadow map is used to generate the soft shadows. An example of those artifacts can be seen in Fig. 7-(a), in the region pointed by the red arrows. As shown in Fig. 7-(b), these artifacts can be minimized by increasing the shadow map resolution.

Subsampling artifacts may be caused not only because of the shadow map resolution, but also because of the light source sampling itself. If a few samples have inadequately been selected from the light source, fine details of the shadow silhouette may be lost because of the shadow overestimation caused by the blurring of the shadow silhouette. This kind of blurring happens when the revectorization-



(a) 512^2 (b) 1024^2

Figure 7: For a low-resolution shadow map (a), fine details (pointed by red arrows) of the shadow silhouette (b) may not be captured by our algorithm. Images were generated for the YeahRight model using 512^2 (a) and 1024^2 (b) shadow map resolutions.

		Shadow Map Resolution		
Model	**Method**	512^2	1024^2	2048^2
Armadillo	Uniform S.	350 ms	360 ms	380 ms
	Adaptive S.	175 ms	100 ms	95 ms
	RB Adaptive S.	95 ms	80 ms	80 ms
	PCSS	5.3 ms	5.4 ms	5.5 ms
	MSSM	4.0 ms	5.8 ms	7.1 ms
YeahRight	Uniform S.	1.4s	1.4s	1.4 s
	Adaptive S.	1.5 s	770 ms	950 ms
	RB Adaptive S.	340 ms	495 ms	620 ms
	PCSS	11.2 ms	11.3 ms	11.7 ms
	MSSM	10.8 ms	11.0 ms	11.1 ms
QuadBot	Uniform S.	800 ms	820 ms	830 ms
	Adaptive S.	950 ms	840 ms	610 ms
	RB Adaptive S.	380 ms	385 ms	400 ms
	PCSS	7.4 ms	7.5 ms	7.6 ms
	MSSM	6.4 ms	8 ms	9.2 ms

Table 1: Rendering times for different sampling strategies measured for the different scenes shown in Fig. 4. Measurements include varying shadow map resolution.

based PCF (Fig. 3-(b)) is used as a visibility function to compute the soft shadows. As we discuss in Section 4.1, we reduce this problem by defining a refinement criteria which generates the appropriate number of samples according to the presence of banding artifacts in the final rendering.

The proposed adaptive approach can be extended for colored textured area light sources as well. Rather than using the samples located at the corners of the area light source, one must rearrange the samples to the center of the sub-quads, and use those samples to access the colored texture. Also, since an adaptive, sparse representation of the area light source may be sampled by the algorithm, one must take this fact into consideration when sampling the colored information of the light source. Indeed, instead of retrieving the actual color of the texture for the sample position, the level of the sample in the adaptive structure can be used as an index to access the appropriate level of a mip-mapped version of the texture. Unfortunately, since the light source refinement criteria do not take into account the color information of the light source to generate new samples, one can lose the details of the texture if a few samples are selected from the light source.

In this work, we have proposed an adaptive sampling approach assuming that the area light source consists of a rectangular, planar shape. Therefore, the use of an adaptive structure where the light source is subdivided into quads is well suited for our purposes. To use our approach for more complex, non-rectangular, planar area light source shapes, one would need to fit a bounding box over the area of the light source, and then proceed with the light source refinement, testing whether the select samples are in the area light source surface.

5.2 Performance

The performance of all the techniques evaluated in this paper can be seen in Tables 1 and 2. The uniform sampling of the area light source provides stable frame rates under different parameters, but provides the worst performance, due to the large number of samples used for every frame. The adaptive sampling strategy proposed in [33] becomes faster as long as the shadow map resolution increases, because less samples are required to generate high-quality accurate soft shadows when high resolution shadow maps are used. On the other hand, such a sampling strategy is sensitive to high output resolutions due to the use of a screen-space criteria. For a Full HD resolution, the adaptive sampling strategy provides performance similar to uniform sampling. Our revectorization-based sampling strategy provides the best performance among the accurate soft shadow techniques evaluated in this paper, regardless of the shadow map and output resolutions used. Obviously, PCSS and MSSM techniques obtain better performance since they use only one sample of the light source to compute the soft shadows. However, as shown in Fig. 4, they also provide the worst soft shadows in terms of visual quality.

An in-depth evaluation of the rendering times obtained for each step of our algorithm is shown in Tables 3 and 4. It is visible that the bottlenecks of our approach are the shadow map rendering and the accurate soft shadow rendering. The shadow map rendering is costly because, different from the discontinuity map rendering and other steps, this one cannot take advantage of a G-buffer rendering to optimize the performance of the scene rendering. So, the entire scene must be rendered several times, according to the number of samples selected from the area light source. On the other hand, the accurate soft shadow rendering is costly because of the shadow revectorization visibility function, which must be computed for every light source sample. The other steps of our approach (e.g., discontinuity map rendering, light source refinement) are more sensitive to output resolution changes, since the calculations are done for even more fragments in the camera view.

Although we have proposed a temporally coherent solution for adaptive sampling, we still cannot guarantee constant, stable frame rate because the number of samples may vary between frames, according to camera and light source movements. Such a limitation is common for adaptive sampling strategies [33]. Even in this case, we show in the supplementary video that our approach provides stable results under different light source and camera movements.

6 Conclusion and Future Work

We have presented a revectorization-based algorithm to compute accurate soft shadows on the basis of a temporally coherent adaptive light source sampling solution. We use the notions of shadow revectorization and discontinuity space to efficiently sample the area light source, generating high-quality soft shadows at interactive speed. The use of a visibility map allows us to further improve the performance of our proposal by restricting the costly hard shadow revectorization for fragments located in penumbra.

In future work, we would like to investigate more efficient ways to solve the problem of accurate soft shadow computation for textured and non-planar area light sources. Also, trying to reduce the computational cost of the shadow map rendering and hard shadow

		Output Resolution		
Model	**Method**	SD	HD	Full HD
Armadillo	Uniform S.	360 ms	360 ms	360 ms
	Adaptive S.	50 ms	100 ms	270 ms
	RB Adaptive S.	70 ms	80 ms	250 ms
	PCSS	3.7 ms	5.4 ms	8.1 ms
	MSSM	5.4 ms	5.8 ms	6.6 ms
YeahRight	Uniform S.	1.4 s	1.4 s	1.4 s
	Adaptive S.	280 ms	770 ms	1.6 s
	RB Adaptive S.	180 ms	495 ms	850 ms
	PCSS	10 ms	11.3 ms	14.4 ms
	MSSM	10.0 ms	11.0 ms	12.6 ms
QuadBot	Uniform S.	800 ms	820 ms	830 ms
	Adaptive S.	220 ms	840 ms	1 s
	RB Adaptive S.	130 ms	385 ms	680 ms
	PCSS	6 ms	7.5 ms	10.2 ms
	MSSM	7.5 ms	8 ms	8.6 ms

Table 2: Rendering times for different sampling strategies measured for the different scenes shown in Fig. 4. Measurements include varying output resolution. SD - Standard Definition ($480p$). HD - High Definition ($720p$). Full HD - Full High Definition ($1080p$).

		Shadow Map Resolution		
Model	**Step**	512^2	1024^2	2048^2
Armadillo	G-Buffer	2.9 ms	2.9 ms	2.9 ms
	Shadow Map	49.6 ms	42.2 ms	42.5 ms
	Disc. Map	8.3 ms	6.1 ms	6.1 ms
	Ref. (First P.)	7.7 ms	7.0 ms	7.0 ms
	Ref. (Sec. P.)	2.5 ms	1.9 ms	1.6 ms
	Final Render	24.0 ms	19.9 ms	19.9 ms
	Total	**95 ms**	**80 ms**	**80 ms**
YeahRight	G-Buffer	5.9 ms	5.9 ms	5.9 ms
	Shadow Map	243 ms	351 ms	443 ms
	Disc. Map	12.3 ms	17.9 ms	21.0 ms
	Ref. (First P.)	12.3 ms	18.9 ms	21.0 ms
	Ref. (Sec. P.)	3.4 ms	5.7 ms	6.5 ms
	Final Render	63.1 ms	95.6 ms	122.6 ms
	Total	**340 ms**	**495 ms**	**620 ms**
QuadBot	G-Buffer	4.5 ms	4.5 ms	4.5 ms
	Shadow Map	234 ms	243 ms	256 ms
	Disc. Map	17.8 ms	18.8 ms	18.2 ms
	Ref. (First P.)	18.4 ms	18.8 ms	19.9 ms
	Ref. (Sec. P.)	5.0 ms	6.0 ms	5.1 ms
	Final Render	100.3 ms	93.9 ms	96.3 ms
	Total	**380 ms**	**385 ms**	**400 ms**

Table 3: Rendering times for each step of the proposed approach, namely G-buffer, shadow map and discontinuity map rendering, first and second passes of the light source refinement, and the final accurate soft shadow rendering. Times were measured for the scenes shown in Fig. 4, including varying shadow map resolution.

accumulation. Furthermore, we intend to revisit the hard shadow revectorization theory to propose improvements in terms of accuracy and performance for the revectorization-based shadow mapping.

Acknowledgments

We wish to thank Keenan Crane for the YeahRight model, and the GPU Education Center program of the NVIDIA Corporation for providing the graphics hardware for the experimental tests. This research is financially supported by Coordenação de Aperfeiçoamento de Pessoal do Nível Superior (CAPES).

		Output Resolution		
Model	**Step**	SD	HD	Full HD
Armadillo	G-Buffer	2.1 ms	2.9 ms	3 ms
	Shadow Map	41.4 ms	42.2 ms	141.1 ms
	Disc. Map	4.7 ms	6.1 ms	34.5 ms
	Ref. (First P.)	4.2 ms	7.0 ms	36.6 ms
	Ref. (Sec. P.)	1.5 ms	1.9 ms	9.6 ms
	Final Render	16.1 ms	19.9 ms	25.2 ms
	Total	**70 ms**	**80 ms**	**250 ms**
YeahRight	G-Buffer	5.9 ms	5.9 ms	5.9 ms
	Shadow Map	136.5 ms	351 ms	460 ms
	Disc. Map	5.6 ms	17.9 ms	35.6 ms
	Ref. (First P.)	4.6 ms	18.9 ms	46.2 ms
	Ref. (Sec. P.)	1.8 ms	5.7 ms	9.1 ms
	Final Render	25.6 ms	95.6 ms	293.2 ms
	Total	**180 ms**	**495 ms**	**850 ms**
QuadBot	G-Buffer	4.1 ms	4.5 ms	5.1 ms
	Shadow Map	86.6 ms	243 ms	311.5 ms
	Disc. Map	5.3 ms	18.8 ms	36.8 ms
	Ref. (First P.)	4.5 ms	18.8 ms	44.2 ms
	Ref. (Sec. P.)	1.6 ms	6.0 ms	9.4 ms
	Final Render	27.9 ms	93.9 ms	273 ms
	Total	**130 ms**	**385 ms**	**680 ms**

Table 4: Rendering times for each step of the proposed approach, namely G-buffer, shadow map and discontinuity map rendering, first and second passes of the light source refinement, and the final accurate soft shadow rendering. Times were measured for the scenes shown in Fig. 4, including varying output resolution. SD - Standard Definition ($480p$). HD - High Definition ($720p$). Full HD - Full High Definition ($1080p$).

References

[1] T. Akenine-Möller and U. Assarsson. Approximate Soft Shadows on Arbitrary Surfaces Using Penumbra Wedges. In *Proceedings of the EGRW*, pp. 297–306. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2002.

[2] U. Assarsson and T. Akenine-Möller. A Geometry-based Soft Shadow Volume Algorithm Using Graphics Hardware. *ACM Trans. Graph.*, 22(3):511–520, July 2003.

[3] D. Bartz, M. Meiner, and T. Httner. Extending Graphics Hardware For Occlusion Queries In OpenGL. In *Proceedings of the EGGH*. The Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 1998.

[4] N. Billen and P. Dutré. Line Sampling for Direct Illumination . *Computer Graphics Forum*, 35(4), July 2016.

[5] V. Bondarev. Shadow Map Silhouette Revectorization. In *Proceedings of the ACM I3D*, pp. 162–162. ACM, New York, NY, USA, 2014.

[6] L. S. Brotman and N. I. Badler. Generating soft shadows with a depth buffer algorithm. *IEEE Computer Graphics and Applications*, 4(10):5–14, Oct 1984.

[7] R. L. Cook. Stochastic Sampling in Computer Graphics. *ACM Trans. Graph.*, 5(1):51–72, Jan. 1986.

[8] R. L. Cook, T. Porter, and L. Carpenter. Distributed Ray Tracing. In *Proceedings of the ACM SIGGRAPH*, pp. 137–145. ACM, New York, NY, USA, 1984.

[9] F. C. Crow. Shadow Algorithms for Computer Graphics. In *Proceedings of the ACM SIGGRAPH*, pp. 242–248. ACM, New York, NY, USA, 1977.

[10] E. Eisemann, M. Schwarz, U. Assarsson, and M. Wimmer. *Real-Time Shadows*. A.K. Peters, 2011.

[11] R. Fernando. Percentage-closer Soft Shadows. In *ACM SIGGRAPH 2005 Sketches*. ACM, New York, NY, USA, 2005.

[12] V. Forest, L. Barthe, and M. Paulin. Realistic Soft Shadows by Penumbra-wedges Blending. In *Proceedings of the ACM Symposium on Graphics Hardware*, pp. 39–46. ACM, New York, NY, USA, 2006.

[13] T. Hachisuka, W. Jarosz, R. P. Weistroffer, K. Dale, G. Humphreys, M. Zwicker, and H. W. Jensen. Multidimensional Adaptive Sampling

and Reconstruction for Ray Tracing. *ACM Trans. Graph.*, 27(3):33:1–33:10, Aug. 2008.
[14] P. Haeberli and K. Akeley. The Accumulation Buffer: Hardware Support for High-quality Rendering. In *Proceedings of the ACM SIGGRAPH*, pp. 309–318. ACM, New York, NY, USA, 1990.
[15] W. Heidrich, S. Brabec, and H.-P. Seidel. Soft Shadow Maps for Linear Lights. In *Proceedings of the Workshop on Rendering Techniques*, pp. 269–280. Springer Vienna, Vienna, 2000.
[16] D. S. Immel, M. F. Cohen, and D. P. Greenberg. A Radiosity Method for Non-diffuse Environments. In *Proceedings of the ACM SIGGRAPH*, pp. 133–142. ACM, New York, NY, USA, 1986.
[17] J. Jimenez, B. Masia, J. I. Echevarria, F. Navarro, and D. Gutierrez. Practical Morphological Anti-Aliasing. In W. Engel, ed., *GPU Pro 2*, pp. 95–113. AK Peters Ltd., Natick, MA, USA, 2011.
[18] J. T. Kajiya. The Rendering Equation. In *Proceedings of the ACM SIGGRAPH*, pp. 143–150. ACM, New York, NY, USA, 1986.
[19] S. Laine, T. Aila, U. Assarsson, J. Lehtinen, and T. Akenine-Möller. Soft Shadow Volumes for Ray Tracing. *ACM Transactions on Graphics*, 24(3), 2005.
[20] P. Lecocq, J.-E. Marvie, G. Sourimant, and P. Gautron. Sub-pixel Shadow Mapping. In *Proceedings of the ACM I3D*, pp. 103–110. ACM, New York, NY, USA, 2014.
[21] J. Lehtinen, S. Laine, and T. Aila. An Improved Physically-Based Soft Shadow Volume Algorithm. *Computer Graphics Forum*, 25(3):303–312, 2006.
[22] M. Macedo and A. Apolinario. Revectorization-Based Shadow Mapping. In *Proceedings of the GI*, pp. 75–83. Canadian Information Processing Society, Toronto, Ont., Canada, 2016.
[23] S. U. Mehta, B. Wang, and R. Ramamoorthi. Axis-aligned Filtering for Interactive Sampled Soft Shadows. *ACM Trans. Graph.*, 31(6):163:1–163:10, Nov. 2012.
[24] D. P. Mitchell. Consequences of Stratified Sampling in Graphics. In *Proceedings of the ACM SIGGRAPH*, pp. 277–280. ACM, New York, NY, USA, 1996.
[25] F. Mora, L. Aveneau, O. Apostu, and D. Ghazanfarpour. Lazy Visibility Evaluation for Exact Soft Shadows. *Computer Graphics Forum*, 31(1):132–145, 2012.
[26] M. J. Ouellette and E. Fiume. On Numerical Solutions to One-dimensional Integration Problems with Applications to Linear Light Sources. *ACM Trans. Graph.*, 20(4):232–279, Oct. 2001.
[27] M. Pan, R. Wang, W. Chen, K. Zhou, and H. Bao. Fast, Sub-pixel Antialiased Shadow Maps. *Computer Graphics Forum*, 28(7):1927–1934, 2009.
[28] C. Peters, C. Münstermann, N. Wetzstein, and R. Klein. Beyond Hard Shadows: Moment Shadow Maps for Single Scattering, Soft Shadows and Translucent Occluders. In *Proceedings of the ACM I3D*, pp. 159–170. ACM, New York, NY, USA, 2016.
[29] A. Pilleboue, G. Singh, D. Coeurjolly, M. Kazhdan, and V. Ostromoukhov. Variance Analysis for Monte Carlo Integration. *ACM Trans. Graph.*, 34(4):124:1–124:14, 2015.
[30] R. Ramamoorthi, J. Anderson, M. Meyer, and D. Nowrouzezahrai. A Theory of Monte Carlo Visibility Sampling. *ACM Trans. Graph.*, 31(5):121:1–121:16, Sept. 2012.
[31] W. T. Reeves, D. H. Salesin, and R. L. Cook. Rendering Antialiased Shadows with Depth Maps. In *Proceedings of the ACM SIGGRAPH*, pp. 283–291. ACM, New York, NY, USA, 1987.
[32] T. Saito and T. Takahashi. Comprehensible Rendering of 3-D Shapes. In *Proceedings of the ACM SIGGRAPH*, pp. 197–206. ACM, New York, NY, USA, 1990.
[33] M. Schwarzler, O. Mattausch, D. Scherzer, and M. Wimmer. Fast Accurate Soft Shadows with Adaptive Light Source Sampling. In *Proceedings of the VMV*, pp. 39–46. Eurographics Association, Aire-la-Ville, Switzerland, Nov. 2012.
[34] P. Sen, M. Cammarano, and P. Hanrahan. Shadow Silhouette Maps. *ACM Trans. Graph.*, 22(3):521–526, July 2003.
[35] L. Shen, J. Feng, and B. Yang. Exponential Soft Shadow Mapping. *Computer Graphics Forum*, 32(4):107–116, 2013.
[36] E. Sintorn, E. Eisemann, and U. Assarsson. Sample Based Visibility for Soft Shadows Using Alias-free Shadow Maps. In *Proceedings of the EGSR*, pp. 1285–1292. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2008.
[37] J.-F. St-Amour, E. Paquette, and P. Poulin. Soft Shadows from Extended Light Sources with Penumbra Deep Shadow Maps. In *Proceedings of the GI*, pp. 105–112. Canadian Information Processing Society, Toronto, Ont., Canada, May 2005.
[38] L. Wang, S. Zhou, W. Ke, and V. Popescu. GEARS: A General and Efficient Algorithm for Rendering Shadows. *Computer Graphics Forum*, 33(6):264–275, 2014.
[39] L.-Y. Wei. Parallel Poisson Disk Sampling. *ACM Trans. Graph.*, 27(3):1–9, Aug. 2008.
[40] T. Whitted. An Improved Illumination Model for Shaded Display. *Commun. ACM*, 23(6):343–349, June 1980.
[41] L. Williams. Casting Curved Shadows on Curved Surfaces. In *Proceedings of the ACM SIGGRAPH*, pp. 270–274. ACM, New York, NY, USA, 1978.
[42] C. Wyman, R. Hoetzlein, and A. Lefohn. Frustum-traced Raster Shadows: Revisiting Irregular Z-buffers. In *Proceedings of the ACM I3D*, pp. 15–23. ACM, New York, NY, USA, 2015.
[43] B. Yang, Z. Dong, J. Feng, H.-P. Seidel, and J. Kautz. Variance Soft Shadow Mapping. *Computer Graphics Forum*, 29(7):2127–2134, 2010.