RevectORIZATION-BASED ACCURATE SOFT SHADOW USING ADOPTIVE AREA LIGHT SOURCE SAMPLING

MÁRCIO C. F. MACEDO∗ ANTONÍO 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
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 with 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 \( p \) distant to the light source by \( p_z \). Also, let us assume \( t_x,y \) the shadow map texel positioned at the 2D position \( x,y \) and \( z(t_x,y) \) a function which retrieves the depth of the blocker of \( p \) as seen from a point light source. The shadow test \( s(p_z,z(t_x,y)) \) is a binary visibility function defined as [41]
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.

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 \(N\):

\[
N = \left[ s(z(t_{x-0,y})), s(z(t_{x+0,y})), s(z(t_{x,y+0})), s(z(t_{x,y-0})) \right],
\]

where \(0\) indicates that the point \(p\) 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 \(N\):

\[
s(p, z(t_{x,y})) = \begin{cases} 
0 & \text{if } p > z(t_{x,y}), \\
1 & \text{otherwise},
\end{cases}
\]

where \(0\) indicates that the point \(p\) is in umbra and \(1\) otherwise.

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 \(l\) 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
Additionally, we label a fragment as whether banding artifacts are produced by the use of those samples whose sizes are equivalent to the maximum number of samples that Algorithm 1 (Fig. 2-(e) and Line 8 of Algorithm 1). Considering (Fig. 2-(f)). This comparison is done in a two-pass strategy with the viewpoint are stored in a G-Buffer [32], which is used to guarantee susceptible to banding artifacts than $D = \frac{4}{3}$ 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 sources. $D = 2$ 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., $SS(p_i) = 4$) and fully shadowed fragments (i.e., $SS(p_i) = 0$) 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 the light source sampling, we build and update a visibility map $VM$. This map is a texture which stores the final illumination condition of the fragment previously estimated in the visibility map and the one condition of the fragment. Let us redefine the shadow sum as $SS_q(p_i)$ and the visibility map as $VM_q(p_i)$, where $q$ refers to the quad index. For the first quad of the adaptive structure, the visibility classification $VM_0(p_i)$ of the fragment $p_i$ is computed as

$$VM_0(p_i) = \begin{cases} \text{umbra} & \text{if } SS_0(p_i) = 0, \\ \text{penumbra} & \text{if } 1 \leq SS_0(p_i) \leq 3, \\ \text{lit} & \text{otherwise.} \end{cases}$$

(5)

Additionally, we define $VM_0(p_i)$ 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

$$VM_q(p_i) = \begin{cases} \text{penumbra} & \text{if } VM_{q-1}(p_i) = \text{lit} \text{ and } SS_q(p_i) = 0, \\ \text{lit} & \text{or } VM_{q-1}(p_i) = \text{umbra} \text{ and } SS_q(p_i) = 4. \end{cases}$$

(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 $l$ distributed over the surface of the area light source $L$, the final soft shadow intensity of a point $p$...
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 10242 shadow map resolution and 25 light source samples.

The revectorization-based filtering technique as visibility function in Equation 4.3 Temporally Coherent Soft Shadow Computation

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 solution of [33] and ours, we have used their solution with a reduction over the window size for occlusion query (during final rendering, Section 4.4.1) and output resolution (during final rendering, Section 4.2). All images were generated by a rectangular area light source.

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 solution of [33] and ours, we have used their solution with a reduction over the window size for occlusion query (during final rendering, Section 4.4.1) and output resolution (during final rendering, Section 4.2). All images were generated by a rectangular area light source.
Both PCF and revectorization-based PCF use the same kernel size of $2 \times 2$. 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 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.
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.

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) 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.
The performance of all the techniques evaluated in this paper can be resolution, the adaptive sampling strategy provides performance
resolutions due to the use of a screen-space criteria. For a Full HD
other hand, such a sampling strategy is sensitive to high output
becomes faster as long as the shadow map resolution increases,
for every frame. The adaptive sampling strategy proposed in [33]
worse performance, due to the large number of samples used
in Fig. 4, they also provide the worst soft shadows in terms of visual
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
time because the number of samples may vary between frames, ac-
garding 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.

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 opti-
mize 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 revectoriza-
tion 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, ac-
gording 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 adap-
tive 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 tex-
tured and non-planar area light sources. Also, trying to reduce the
computational cost of the shadow map rendering and hard shadow

<table>
<thead>
<tr>
<th>Model</th>
<th>Method</th>
<th>Shadow Map Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>512^2</td>
</tr>
<tr>
<td>Uniform S.</td>
<td>350 ms</td>
<td>360 ms</td>
</tr>
<tr>
<td>Adaptive S.</td>
<td>175 ms</td>
<td>100 ms</td>
</tr>
<tr>
<td>Armadillo</td>
<td>RB Adaptive S.</td>
<td>95 ms</td>
</tr>
<tr>
<td>PCSS</td>
<td>5.3 ms</td>
<td>5.4 ms</td>
</tr>
<tr>
<td>MSSM</td>
<td>4.0 ms</td>
<td>5.8 ms</td>
</tr>
<tr>
<td>Uniform S.</td>
<td>1.4 s</td>
<td>1.4 s</td>
</tr>
<tr>
<td>Adaptive S.</td>
<td>1.5 s</td>
<td>770 ms</td>
</tr>
<tr>
<td>YeahRight</td>
<td>RB Adaptive S.</td>
<td>340 ms</td>
</tr>
<tr>
<td>PCSS</td>
<td>11.2 ms</td>
<td>11.3 ms</td>
</tr>
<tr>
<td>MSSM</td>
<td>10.8 ms</td>
<td>11.0 ms</td>
</tr>
<tr>
<td>Uniform S.</td>
<td>800 ms</td>
<td>820 ms</td>
</tr>
<tr>
<td>Adaptive S.</td>
<td>950 ms</td>
<td>840 ms</td>
</tr>
<tr>
<td>QuadBot</td>
<td>RB Adaptive S.</td>
<td>380 ms</td>
</tr>
<tr>
<td>PCSS</td>
<td>7.4 ms</td>
<td>7.5 ms</td>
</tr>
<tr>
<td>MSSM</td>
<td>6.4 ms</td>
<td>8 ms</td>
</tr>
</tbody>
</table>

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

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.
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 de Nível Superior (CAPES).

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 de Nível Superior (CAPES).

REFERENCES


<table>
<thead>
<tr>
<th>Output Resolution</th>
<th>Model</th>
<th>Step</th>
<th>SD</th>
<th>HD</th>
<th>Full HD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>480</td>
<td>720</td>
<td>1080</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G-Buffer</td>
<td></td>
<td>2.1 ms</td>
<td>2.9 ms</td>
<td>3 ms</td>
</tr>
<tr>
<td></td>
<td>Shadow Map</td>
<td></td>
<td>41.4 ms</td>
<td>42.2 ms</td>
<td>141.1 ms</td>
</tr>
<tr>
<td></td>
<td>Disc. Map</td>
<td></td>
<td>4.7 ms</td>
<td>6.1 ms</td>
<td>34.5 ms</td>
</tr>
<tr>
<td></td>
<td>Armadillo</td>
<td></td>
<td>4.2 ms</td>
<td>7.0 ms</td>
<td>36.6 ms</td>
</tr>
<tr>
<td></td>
<td>Ref. (First P.)</td>
<td></td>
<td>1.5 ms</td>
<td>1.9 ms</td>
<td>9.6 ms</td>
</tr>
<tr>
<td></td>
<td>Final Render</td>
<td></td>
<td>16.1 ms</td>
<td>19.9 ms</td>
<td>25.2 ms</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td>27.9 ms</td>
<td>93.9 ms</td>
<td>273 ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output Resolution</th>
<th>Model</th>
<th>Step</th>
<th>SD</th>
<th>HD</th>
<th>Full HD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>480</td>
<td>720</td>
<td>1080</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G-Buffer</td>
<td></td>
<td>4.1 ms</td>
<td>4.5 ms</td>
<td>5.1 ms</td>
</tr>
<tr>
<td></td>
<td>Shadow Map</td>
<td></td>
<td>8.6 ms</td>
<td>24.3 ms</td>
<td>311.5 ms</td>
</tr>
<tr>
<td></td>
<td>Disc. Map</td>
<td></td>
<td>5.3 ms</td>
<td>18.8 ms</td>
<td>36.8 ms</td>
</tr>
<tr>
<td></td>
<td>QuadBot</td>
<td></td>
<td>4.5 ms</td>
<td>18.8 ms</td>
<td>44.2 ms</td>
</tr>
<tr>
<td></td>
<td>Ref. (First P.)</td>
<td></td>
<td>1.6 ms</td>
<td>6.0 ms</td>
<td>9.4 ms</td>
</tr>
<tr>
<td></td>
<td>Final Render</td>
<td></td>
<td>27.9 ms</td>
<td>93.9 ms</td>
<td>273 ms</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td>130 ms</td>
<td>385 ms</td>
<td>680 ms</td>
</tr>
</tbody>
</table>

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).

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.

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).

189


