Efficient Shadow Anti-Aliasing Techniques using Silhouette Revectorization

Márcio C. F. Macedo, Antônio L. Apolinário Jr.
Department of Computer Science
Federal University of Bahia
Salvador, BA, Brazil
Email: marciocfmacedo@gmail.com, antonio.apolinario@ufba.br

***Abstract*—Real-time rendering of high-quality shadows is a challenging problem in computer graphics. Shadow mapping is widely adopted for real-time shadow rendering, but introduces aliasing artifacts along the shadow silhouette and is not able to simulate the penumbra effect. Techniques that simulate penumbra are computationally expensive, providing performance far from real time. In this work[1], we present the revectorization-based shadow mapping, a technique that takes advantage of the camera-view resolution and the shadow silhouette shape to suppress shadow aliasing artifacts at little additional cost. Inspired by the superior visual quality obtained with the shadow silhouette revectorization, we extend the revectorization-based visibility function to propose a set of techniques that provide high-quality anti-aliasing for both shadow rendering and penumbra simulation. We further integrate the Euclidean distance transform into the revectorization-based visibility function to provide real-time performance for the penumbra simulation. The results, evaluated in terms of visual quality and rendering time, show that the proposed techniques produce less visual quality artifacts than related work, while keeping the real-time performance, mainly for the shadow rendering without the penumbra simulation.**

I. Introduction

Shadows may be defined as a composition of points that are partially (penumbra) or not visible (umbra) by a light source. In the real world, shadows are important because they enhance our understanding of the surrounding scene, improving our visual perception with respect to the relative spatial disposition of light blocker and shadow receiver objects. In computer graphics, shadows enhance the realism of the images rendered from virtual scenes. In this sense, shadows are useful for a variety of applications, such as movies, games, simulators, augmented reality and even art.

Specifically for games and augmented reality applications, a successful shadow rendering algorithm must fulfill two essential requirements: **high visual quality**, to improve the user's perception of the virtual scene; **real-time performance**, to enable the user interactivity with the application.

Unfortunately, the methods that compute highly accurate shadows take too much processing time to be used interactively for dynamic scenes, because they need to perform an accurate visibility evaluation over a sampled version of the light source and on the basis of the geometric information available in the scene. To simplify the shadow rendering problem, some methods estimate only the umbra component of the shadow, or simulate a fixed-size penumbra effect. These techniques compute shadows from object- or image-based approaches. Object-based approaches are able to produce high-quality shadows, making use of a shadow volume [2], but they tend to be slower and less scalable than image-based approaches. Image-based approaches typically use a shadow map [3], an image-based representation of the light source view, to generate real-time shadows. Despite the advantages of such a shadow map representation, the limited resolution of the shadow map generates aliasing, light leaking and temporal incoherence artifacts into the final shadow rendering.

In this thesis' work, we present the revectorization-based shadow mapping, a new real-time shadow technique that makes use of the camera-view resolution and the shadow silhouette shape to reduce shadow aliasing artifacts generated by traditional shadow mapping. Then, we show the extensions that have been proposed for shadow revectorization to provide anti-aliasing for shadows with both umbra and penumbra effects. We also present a novel use of Euclidean distance transform for high-quality, real-time penumbra simulation with reduced light leaking artifacts. Finally, we show how the revectorization-based shadow mapping can be used for accurate soft shadow rendering.

Hence, the main contributions to the field of shadow rendering are summarized, in order of importance, as follows:

- A real-time, memory-efficient shadow anti-aliasing approach that is able to reduce the perspective aliasing artifacts for hard shadows that simulate the umbra effect [4]–[6] (Section II);
- A shadow rendering technique that makes use of Euclidean distance transform to produce real-time, high-quality soft shadows with fixed-size penumbra and less artifacts than related work [7]–[9] (Section III);
- An anti-aliasing, screen-space shadow mapping technique that generates variable-size penumbra (*i.e.,* soft shadows) as fast as related work (Section IV);
- An adaptive light source sampling approach that takes advantage of the shadow revectorization to generate accurate shadows from a few light source samples, speeding up the accurate shadow computation [10] (Section V);

[1]Ph.D. thesis [1] conducted between March 2015 and May 2018

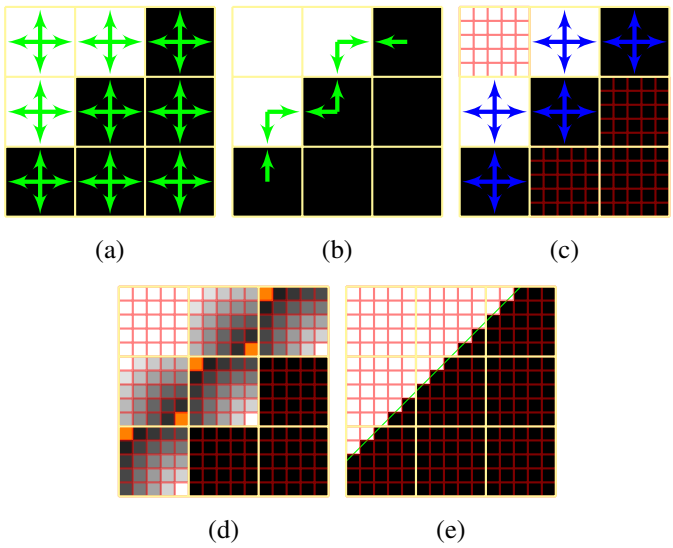
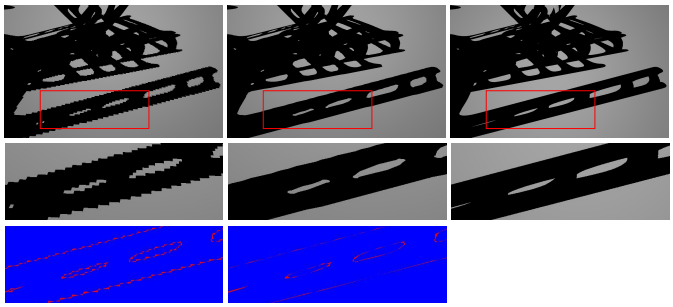


Fig. 1. An overview of revectorization-based shadow mapping. Given an aliased shadow silhouette (black squares in (a)), a neighbourhood evaluation (green arrows in (a)) is conducted to detect aliasing directions (green arrows in (b)). Then, a traversal (blue arrows in (c)) is performed to determine the normalized relative distance (gray shades in (d)) of each aliased camera-view fragment (each red square in the red grid) to the origin (orange squares in (d)) of the shadow silhouette. Finally, a visibility function (green line in (e)) works over the normalized relative distance to revectorize the shadow silhouette (e).



(a) Shadow Mapping 9.6 ms (b) RBSM 10.4 ms (c) Shadow Volume 338.9 ms

Fig. 2. A comparison between different hard shadow techniques for a 4096^2 shadow map resolution. Times were measured for an NVIDIA Titan X, 8 GB RAM, i7. False color maps highlight visual differences (a, b) to the ground-truth (c). RBSM - Revectorization-based shadow mapping.

II. Revectorization-based Shadow Mapping

Image revectorization may be defined as an anti-aliasing approach that uses the available image resolution to reduce the jagged pattern of an aliased region by the recovering of its approximate original color. The main advantage of such an approach is the ability to generate images of higher visual quality in real time.

In the field of real-time shadows, techniques based on shadow mapping generate aliasing artifacts along the shadow silhouette mainly when using low-resolution shadow maps for the shadow computation. To solve this problem, we propose the revectorization-based shadow mapping, an algorithm that aims to locate shadow silhouette patterns in the scene and to use the available screen-space resolution through the revectorization of the shadow. An overview of the proposed pipeline is shown in Figure 1 and is detailed as follows.

The first step of our algorithm consists in the generation of the aliased umbra, as proposed by the traditional shadow mapping [3]. Then, we project every camera-view fragment into the shadow map (whose resolution is depicted by the yellow grid in Figure 1) and compare the visibility condition estimated by neighbour shadow map texels (see the green arrows in Figure 1-(a)). On the basis of this comparison, we are able to detect the directions (green arrows in Figure 1-(b)) where the aliasing artifacts are located. Then, we project the fragments back to the camera space (red grid in Figure 1) in order to traverse (represented by blue arrows in Figure 1-(c)) the shadow silhouette in the camera space. This traversal is performed with the goal of computing, for each camera-view fragment located inside the shadow silhouette, the normalized relative distance (gray shades in Figure 1-(d)) of each fragment to the local origin (orange squares in Figure 1-(d)) of the shadow silhouette where the fragment is located. To detect the end of the traversal, we check if the neighbour fragment accessed during traversal has a different visibility condition that the one estimated by the initial fragment of the traversal. In this sense, for lit fragments, the shadow silhouette ends in a umbra fragment. On the counterpart, for umbra fragments, the shadow silhouette ends in a lit fragment. If the visibility condition between neighbours is the same, we check whether neighbour fragments share at least one silhouette direction in common. If that is not the case, the traversal has stepped out of the lit/shadowed side of the aliased shadow silhouette. Taking advantage of this normalized relative distance, the algorithm is able to estimate an anti-aliased shadow silhouette (green line in Figure 1-(e)) and determine whether each camera-view fragment (red square in Figure 1) lies inside or outside the anti-aliased silhouette (Figure 1-(e)).

In Figure 2, we compare the proposed approach with the traditional shadow mapping [3] and the accurate shadow volume [2] in terms of visual quality and rendering time. Shadow mapping generates aliasing artifacts along the shadow silhouette (Figure 2-(a)), even when using a high-resolution shadow map to simulate the umbra effect of the shadow. The proposed approach is able to effectively reduce the aliasing artifacts generated by shadow mapping (see the false color visualization shown in Figure 2-(b)) and keep the real-time performance of the shadow rendering. That makes the proposed approach as accurate as the shadow volume technique (Figure 2-(c)), but producing shadows much faster than such an accurate solution.

Despite the high visual quality obtained with the revectorization-based shadow mapping, the techniques shown in Figure 2 simulate only the umbra effect of the shadow. In the next section, we present our solution to simulate real-time anti-aliased fixed-size penumbra on the basis of the shadow revectorization theory.

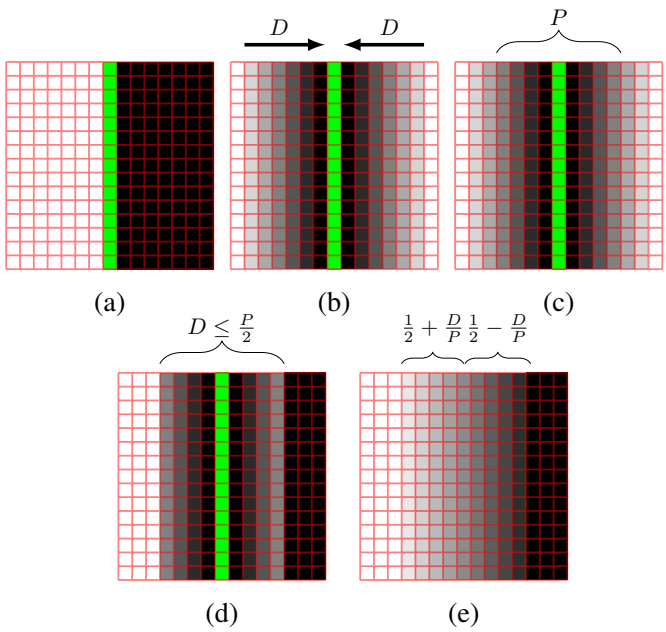


Fig. 3. An overview of Euclidean distance transform shadow mapping. First, revectorization-based shadow mapping is used (a) to generate anti-aliased shadow silhouettes (green rectangles) in the camera view. Then, for every fragment (red square) in the screen space, the world-space distance D to the closest fragment located in the shadow silhouette is computed (b). Given a user-defined penumbra size P (c), the algorithm restricts the penumbra computation for fragments located in the penumbra region (d). Finally, the Euclidean distance transform previously computed is normalized to simulate the smooth transition between lit and umbra regions that characterize the penumbra effect (e).

III. Euclidean Distance Transform Shadow Mapping

Euclidean distance transform (EDT) shadow mapping is a technique that uses EDT to simulate the fixed-size penumbra effect over anti-aliased hard shadows. The main assumption of EDT shadow mapping is that the penumbra intensity of a fragment can be approximated by the Euclidean distance of the fragment to the nearest fragment located in an anti-aliased hard shadow silhouette.

Let us call *seed* a fragment that lies in the anti-aliased hard shadow silhouette (green squares in Figure 3) generated by revectorization-based shadow mapping. A seed fragment can be easily located in the screen space of the camera view by the application of a 3×3 rectangular filter to detect whether the hard shadow intensity of the current fragment differs from one of its neighbours located in the 8-connected screen-space neighbourhood (Figure 3-(a)).

Once the seed fragments have been detected in the image, the EDT can be computed. So, for each non-seed fragment, the world-space Euclidean distance D of the fragment to the nearest seed located in the shadow silhouette is computed (Figure 3-(b)), D being a world-space distance computed on the basis of the world-space position retrieved from a G-buffer. Let us assume P as a user-defined parameter that controls the size of the penumbra that will be simulated.

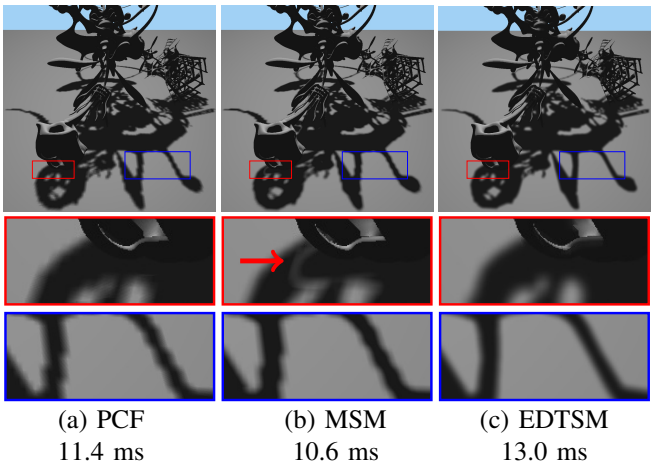


Fig. 4. A comparison between different fixed-size penumbra simulation techniques for a 1024^2 shadow map resolution. Times were measured for an NVIDIA Titan X, 8 GB RAM, i7. Each closeup shows whether the technique handles light leaking (pointed by a red arrow in the red closeup) and aliasing (blue closeup) artifacts. PCF - Percentage-closer filtering. MSM - Moment shadow mapping. EDTSM - Euclidean distance transform shadow mapping.

As shown in Figure 3-(c), each half of the penumbra size belongs to one side of the shadow silhouette. Therefore, just by checking if $D \leq \frac{P}{2}$, we can detect whether a fragment belongs to the desired penumbra region (Figure 3-(d)). For the fragments located outside of the penumbra region, the shadow intensity is given by the shadow test (umbra and lit regions in Figure 3-(d)). Meanwhile, for fragments in the penumbra region, we normalize the EDT to the closed unit interval $[0, 1]$, assuming that umbra and lit fragments have intensities 0 and 1, respectively. To make the EDT to resemble a penumbra, the final intensity $I \in [0, 1]$ of penumbra fragments is

$$I = \begin{cases} \frac{1}{2} - \frac{D}{P} & \text{if the fragment was in shadow,} \\ \frac{1}{2} + \frac{D}{P} & \text{otherwise.} \end{cases} \quad (1)$$

As shown in Figure 3-(e), the use of a normalized EDT (1) allows the simulation of the fixed-size penumbra effect.

In Figure 4, we compare our approach with the traditional percentage-closer filtering [11] and the more recent moment shadow mapping [12] in terms of visual quality and rendering time. Both related work suffer from aliasing artifacts along the fixed-size penumbra (see blue closeups of Figures 4-(a, b)). Moment shadow mapping is faster than percentage-closer filtering, at the cost of generating light leaking artifacts (see the red closeup of Figure 4-(b)). In this sense, our approach is slightly slower than related work, but does not suffer from aliasing (see blue closeup of Figure 4-(c)) and light leaking (as illustrated in the red closeup of Figure 4-(c)) artifacts as much as related work. These results show that the proposed technique is able to generate high-quality shadows in real time.

Unfortunately, fixed-size penumbra is not much realistic, because real-world penumbra has a variable size along its silhouette. In the next section, we present our solution to simulate variable-size penumbra in the screen space.

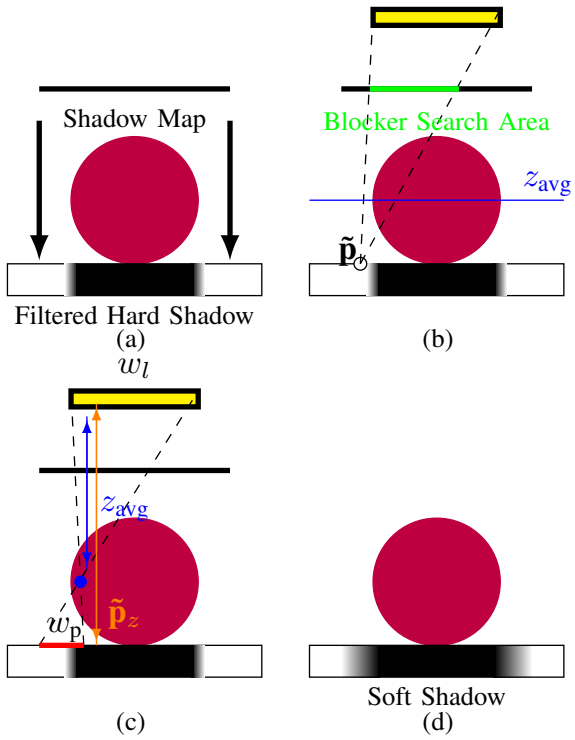


Fig. 5. An overview of the screen-space revectorization-based soft shadow mapping. After the anti-aliased filtered hard shadow rendering (a), average blocker depth (z_{avg} in (b)) and penumbra size (w_{p} in (c)) are used to guide a screen-space filtering to generate soft shadows (d).

IV. Screen-Space Revectorization-based Soft Shadow Mapping

To estimate the variable penumbra size of a given region, percentage-closer soft shadows [13] proposes that one needs to first approximate the scene (Figure 5-(a)) by objects that are planar and parallel to each other (Figure 5-(b)). Then, on the basis of such a parallel-planar assumption, the variable penumbra size w_{p} (Figure 5-(c)), can be estimated as

$$w_{\text{p}} = w_{\text{l}} \frac{\tilde{\mathbf{p}}_z - z_{\text{avg}}}{z_{\text{avg}}}, \tag{2}$$

where w_{l} is the light source size, z_{avg} is the average blocker depth, and $\tilde{\mathbf{p}}_z$ is the distance of a point $\mathbf{p}$ to the light source.

To produce anti-aliased soft shadows on the basis of the concept of shadow revectorization, we first compute a shadow map and a G-buffer (Figure 5-(a)). Then, we modify the visibility function of the revectorization-based shadow mapping to output the normalized distance of each fragment to the revectorized shadow silhouette (see the shadows in Figure 5-(a)), rather than whether the fragment is inside the revectorized hard shadow silhouette, as described in Section II. Next, both average blocker depth (z_{avg} in Figure 5-(b)) and variable penumbra size (w_{p} in Figure 5-(c)) are estimated on the basis of the parallel-planar assumption of the percentage-closer soft shadows. However, the penumbra size estimated using (2) works well for filtering in the shadow map space. Since we aim to filter the soft shadows in the screen space, we estimate a screen-space penumbra size $w_{\text{p}}^{\text{screen}}$ [14]

$$w_{\text{p}}^{\text{screen}} = \frac{w_{\text{p}} z_{\text{screen}}}{\mathbf{p}_{z_{eye}}}, \tag{3}$$

where $\mathbf{p}_{z_{eye}}$ is the distance of a point $\mathbf{p}$ to the center of the camera, and z_{screen} is the inverse of the viewport scale, in terms of field of view.

Once with $w_{\text{p}}^{\text{screen}}$, we apply a two-pass separable cross-bilateral filter over the screen-sized penumbra area to generate visually plausible soft shadows, as depicted in Figure 5-(d)).

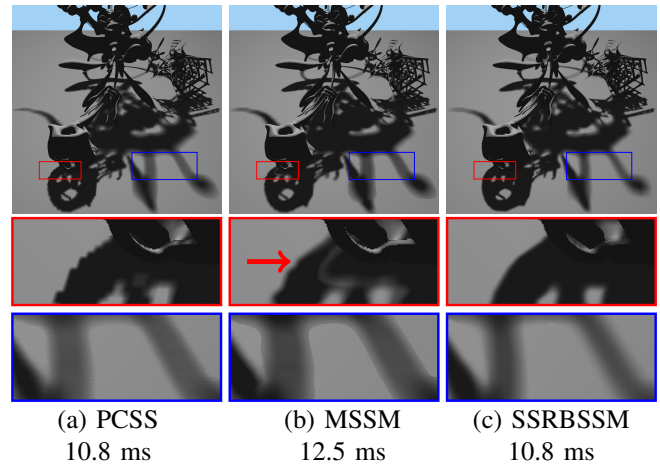


Fig. 6. A comparison between different soft shadow techniques for a 1024^2 shadow map resolution. Times were measured for an NVIDIA Titan X, 8 GB RAM, i7. PCSS - Percentage-closer soft shadows. MSSM - Moment soft shadow mapping. SSRBSSM - Screen-space revectorization-based soft shadow mapping.

In Figure 6, we compare our soft shadow technique with the traditional percentage-closer soft shadows [13] and the recent moment soft shadow mapping [15] with respect to rendering quality and performance. From the red and blue closeups of Figure 6, we can see that these three techniques are able to generate visually plausible soft shadows with variable-size penumbra. In this sense, our proposed technique (Figure 6-(c)) minimizes the presence of aliasing (visible in the red closeup of Figure 6-(a)) and light leaking (pointed by the red arrow of Figure 6-(b)) artifacts, as compared to related work. Meanwhile, we are able to provide real-time performance, being as fast as related work, but providing improved visual quality.

All the techniques compared in this section compute soft shadows by approximating the area light source as a single point/spot light, and also assuming that both light source and light blocker objects are planar and parallel to each other. This simplifying assumption is useful to produce real-time, visually plausible soft shadow algorithms. However, physical accuracy is lost because that assumption brings incorrect penumbra size and visibility estimations. In the next section, we present a new technique to generate visually accurate soft shadows at interactive frame rates on the basis of the visual quality provided by revectorization-based shadow mapping.

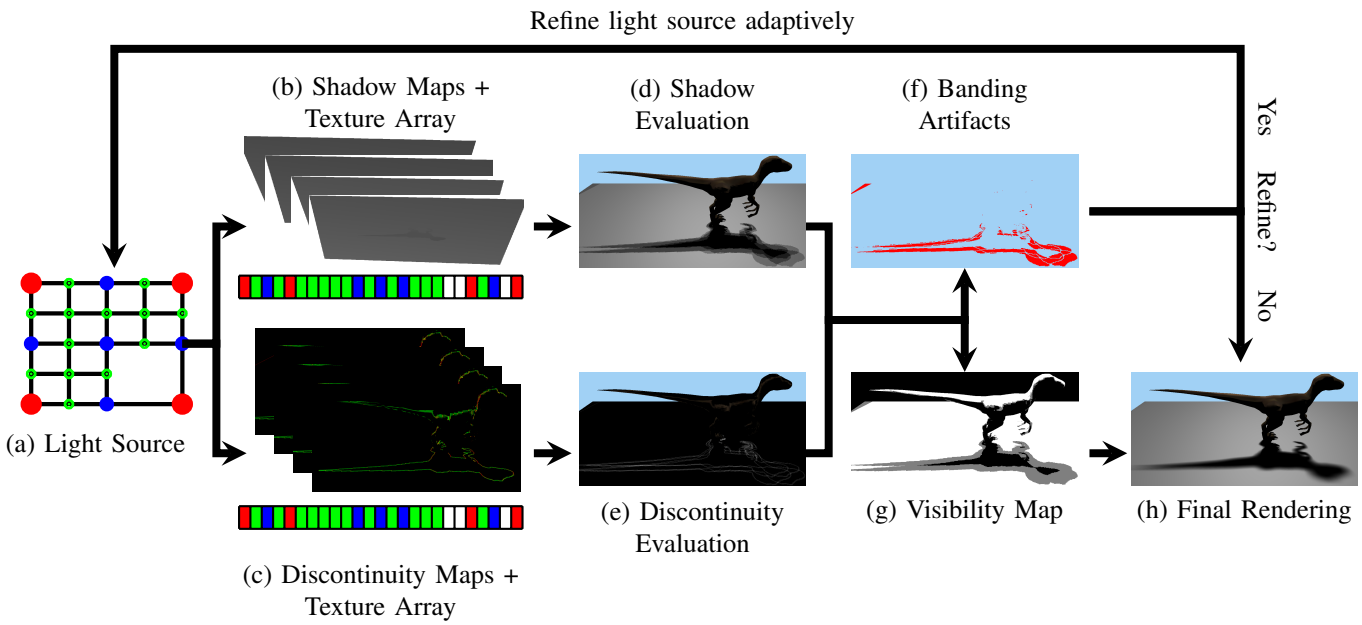


Fig. 7. An overview of our revectorization-based accurate soft shadow mapping. Given an area light source (a), we first generate four shadow (b) and discontinuity maps (c) for the neighbours point light sources located at the light source corners and store those maps into separate texture arrays. Then, 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.

V. Revectorization-based Accurate Soft Shadow Mapping

In this section, we present our adaptive approach to generate accurate soft shadows on the basis of the revectorization-based shadow mapping. An overview of the proposed algorithm is shown in Figure 7. We adaptively sample quads of four point light sources from the area light source (Figure 7-(a)) and use them to evaluate whether banding artifacts are generated by the algorithm (Figure 7-(b, c, d, e, f)). In this case, we take advantage of the improved accuracy provided by the revectorization-based shadow mapping to select a few light source samples, while providing high visual quality (Figure 7-(h)).

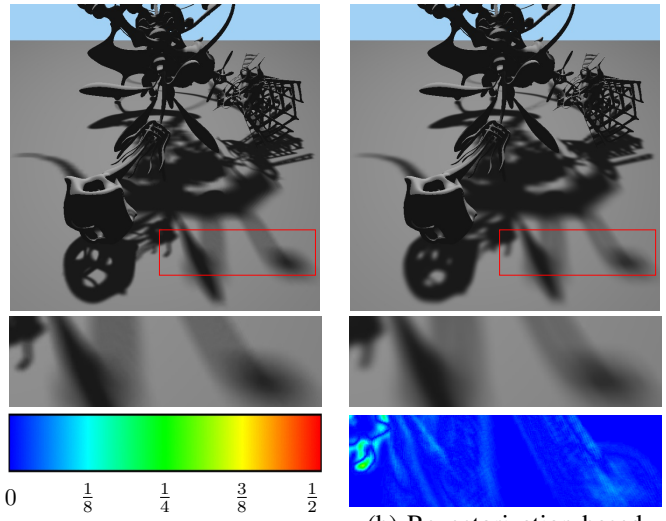
Let us define the area light source $\mathcal{L}$ as an adaptive structure where each node consists of a quad $\mathcal{Q}$ formed by four neighbour point light sources. The main goal of the adaptive sampling is to generate only the light source samples $\mathbf{l} \in \mathcal{L}$ that will contribute significantly to the final soft shadow appearance, generating visually accurate soft shadows.

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 Figure 7-(a)). Then, for each point light source, we compute a shadow map (Figure 7-(b)) and a discontinuity map (Figure 7-(c)) that stores where the shadow silhouettes are located in the camera view.

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 (Figures 7-(d, e)) to detect whether banding artifacts are produced by the use of those samples (Figure 7-(f)). Basically, for each fragment in the camera view, we sum the hard shadow intensities generated with respect to four neighbour light source samples, and check if such a sum is the same for its 8-connected neighbour fragments and the fragment is located out of the shadow silhouette. If that is the case, the fragment is located in a penumbra region that is not sufficiently smooth and produces banding artifacts. Therefore, the algorithm is iterated to produce new light quads adaptively, until no banding artifact is generated.

To produce the final accurate soft shadow rendering for each visible fragment in the camera view, we compute the average of the shadow intensities generated by each light source sample using the revectorization-based shadow mapping pipeline, as described in Section II. With this algorithm, we are able to generate visually accurate shadows, such as the one shown in Figure 7-(h).

To optimize the performance of the proposed solution, during the light source sampling, we build and update a visibility map (Figure 7-(g)) that stores the final illumination condition of a fragment (*i.e.,* whether the fragment is lit, penumbra or umbra). With such a map, we are able to restrict the final accurate soft shadow computation for penumbra fragments only.



(a) Uniform Sampling 1.4 s

(b) Revectorization-based Adaptive Sampling 0.5 s

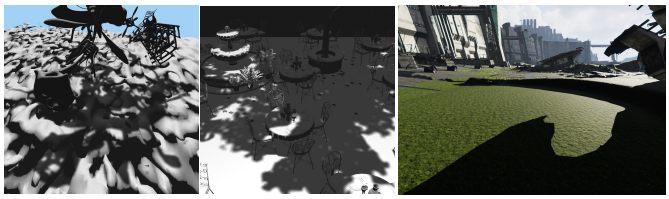
Fig. 8. Accurate soft shadows generated by the use of 289 (a) and 62 (b) light source samples generated by the uniform (a) and our adaptive (b) sampling of the area light source for a 1024^2 shadow map resolution. Times were measured for an NVIDIA Titan X, 8 GB RAM, i7.

In Figure 8, we compare, with respect to visual quality and rendering time, our revectorization-based adaptive sampling strategy with a uniform sampling approach that uses shadow mapping as a basis to generate accurate soft shadows. In this case, we used four times less light source samples than the uniform sampling approach to generate visually accurate soft shadows, whose difference is most visible due to the use of a false color map. However, even with the use of our approach, we still could not generate accurate soft shadows in real time.

VI. Conclusion

In this work, we have shown that the shadow revectorization provides anti-aliasing for four types of shadows, namely hard shadows (Section II), filtered hard shadows (or fixed-size penumbra, Section III), visually plausible soft shadows (Section IV) and accurate soft shadows (Section V).

By implementing the shadow revectorization into a popular game engine [6], we could see that our approach is highly attractive for games due to its real-time performance and high-quality rendering by the reduction of shadow aliasing artifacts. Moreover, shadow revectorization supports not only the rendering of high-quality shadows for planar receivers, as shown in Figures 2, 4, 6, and 8, but also the high-quality rendering for non-planar receivers (Figure 9-(a)), complex game-like scenarios (Figure 9-(b)), with several overlapping objects, and even outdoor scenarios (Figure 9-(c)), with streets and vegetation. Hence, shadow revectorization is able to produce real-time anti-aliasing for a variety of shadows and scenarios, outperforming state-of-the-art methods in terms of visual quality and/or processing time mainly for low-resolution shadow maps.



(a) (b) (c)

Fig. 9. Our anti-aliased shadows for scenarios with noisy shadow receivers (a), overlapping shadow casters (b), and large outdoor structures (c).

VII. Publications

During this Ph.D. work [1], we published four full papers [4], [7], [8], [10] in international conferences (**Qualis B1**) and one full paper [6] in a national conference (**Qualis B2**). Moreover, since one paper [7] was selected as **one of the four best papers** of its respective conference, we also published an extended version [9] of the paper [7] in an international journal (**Qualis A2, Impact Factor: 1.176**).

Acknowledgment

This work was financially supported by the scholarship program of CAPES. Hardware was provided by NVIDIA Corporation through the GPU Education Center program.

References

[1] M. C. F. Macedo, "Efficient Shadow Anti-Aliasing Techniques using Silhouette Revectorization," Ph.D. thesis, Federal University of Bahia, May 2018.

[2] F. C. Crow, "Shadow Algorithms for Computer Graphics," in *Proceedings of the ACM SIGGRAPH*, 1977, pp. 242–248.

[3] L. Williams, "Casting Curved Shadows on Curved Surfaces," in *Proceedings of the ACM SIGGRAPH*, 1978, pp. 270–274.

[4] M. C. F. Macedo and A. L. Apolinário, "Revectorization-Based Shadow Mapping," in *Proceedings of the Graphics Interface*, 2016, pp. 75–83.

[5] M. C. F. Macedo, A. L. Apolinário, and K. A. Agüero, "Optimized Visibility Functions for Revectorization-Based Shadow Mapping," *ArXiv e-prints*, Nov. 2017.

[6] M. C. F. Macedo, A. V. Teixeira, A. L. Apolinário, and K. A. Agüero, "Hard Shadow Anti-Aliasing for Spot Lights in a Game Engine," in *Proceedings of the SBGAMES*, Nov 2017.

[7] M. C. F. Macedo and A. L. Apolinário, "Euclidean Distance Transform Shadow Mapping," in *Proceedings of the Graphics Interface*, 2017, pp. 171–180.

[8] ——, "Euclidean Distance Transform Soft Shadow Mapping," in *Proceedings of the SIBGRAPI*, Oct 2017, pp. 238–245.

[9] ——, "Improved anti-aliasing for euclidean distance transform shadow mapping," *Computers & Graphics*, vol. 71, pp. 166 – 179, 2018.

[10] ——, "Revectorization-Based Accurate Soft Shadow Using Adaptive Area Light Source Sampling," in *Proceedings of the Graphics Interface*, 2017, pp. 181–189.

[11] W. T. Reeves, D. H. Salesin, and R. L. Cook, "Rendering Antialiased Shadows with Depth Maps," in *Proceedings of the ACM SIGGRAPH*, 1987, pp. 283–291.

[12] C. Peters and R. Klein, "Moment Shadow Mapping," in *Proceedings of the ACM I3D*, 2015, pp. 7–14.

[13] R. Fernando, "Percentage-closer Soft Shadows," in *ACM SIGGRAPH Sketches*, 2005.

[14] M. MohammadBagher, J. Kautz, N. Holzschuch, and C. Soler, "Screen-space Percentage-Closer Soft Shadows," in *ACM SIGGRAPH Posters*, 2010, pp. 133–133.

[15] 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*, 2016, pp. 159–170.