Euclidean Distance Transform Soft Shadow Mapping

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

***Abstract*—The real-time computation of high-quality soft shadows is a challenging task in computer graphics. The existing approximate solutions for soft shadow computation are prone to aliasing artifacts for small penumbra sizes and banding or light leaking artifacts for large penumbra sizes. In this paper, we aim to minimize these problems with the use of a normalized Euclidean distance transform for soft shadow mapping. We compute anti-aliased hard shadows in the camera view, and, on the basis of the percentage-closer soft shadows framework, we estimate the penumbra size of the shadow silhouette and compute a filtered normalized Euclidean distance transform to generate the soft shadows at the penumbra location. Our approach yields real-time performance and generates soft shadows with less artifacts than related work.**

I. Introduction

In computer graphics, shadows are fundamental because they provide important visual cues about the virtual scene, improving the realism of the rendered images. The real-time, photorealistic shadow computation is especially useful for games and augmented reality, which strive to keep the user interactivity with the application. Unfortunately, the exact rendering of accurate shadows requires the determination of the fraction of the area occupied by the light source which is visible for each fragment of the scene. This visibility evaluation is computationally expensive and still cannot be computed in real-time.

To generate real-time soft shadows, many techniques approximate the area light source by a single point light source and try to simulate the penumbra effect by blurring the shadows generated from shadow mapping [1] with a variable-size filter. Despite the plausible visual quality obtained for the soft shadows, the use of shadow mapping as a basis for shadow rendering introduces aliasing artifacts, which are typically visible along the shadow silhouette of small penumbra sizes. Moreover, banding artifacts can be seen in large penumbra sizes if an insufficient filter size is used. Finally, light leaking artifacts may be generated due to the shadow map filtering.

In this paper, we aim to reduce the aforementioned problems with the Euclidean Distance Transform Soft Shadow Mapping (EDTSSM), our approach to compute soft shadows in real-time. Taking advantage of the planar-parallel assumption of the Percentage-Closer Soft Shadows (PCSS) technique [2] and the anti-aliasing provided by the shadow revectorization [3], we determine the penumbra region in the scene and compute a filtered normalized Euclidean Distance Transform (EDT) on the Graphics Processing Unit (GPU) to generate the smooth, visually plausible soft shadows.

II. Related Work

One of the most popular shadow generation techniques, the shadow mapping [1], lacks realism because the hard shadows generated by the use of the shadow map do not simulate the penumbra effect commonly seen in real scenarios.

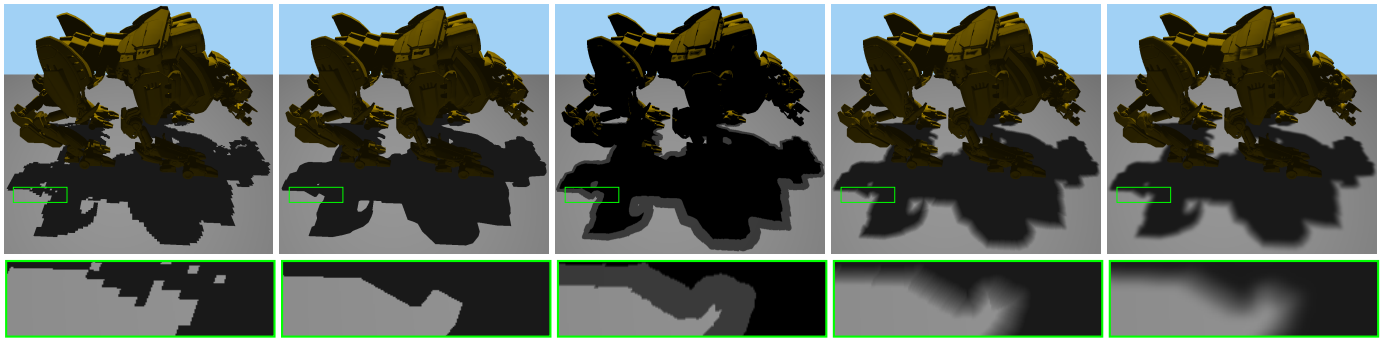
In this section, we briefly review the techniques which compute more realistic soft shadows on the basis of a single shadow map. An in-depth review of the existing soft shadow techniques can be found in [4], [5].

To fake the penumbra effect, some techniques [6]–[8] perform a search in the shadow map to locate the closest shadow map texel whose visibility condition is different from the one estimated by the current texel. In this case, the penumbra size is determined by the search radius, which is used to filter the hard shadows and generate the soft ones. These techniques require costly search steps and have problems to simulate inner and outer penumbra.

Chan and Durand [9] compute soft shadows on the basis of the shadow map and the smoothies constructed from the silhouette of the polygon mesh. However, the smoothies are only able to simulate the outer penumbra, producing unrealistic soft shadows for large penumbra sizes.

Backprojection techniques assume that the shadow map is a discrete representation of the shadow blocker geometry and project each shadow map texel into the area light source to determine the amount of the light source visible from a given surface point [10]–[13]. Despite the advances [14]–[16] to reduce the light leaking and shadow overestimation caused by the coarse approximation of the blocker geometry, the back-projection techniques achieve only interactive performance.

To compute more realistic soft shadows, PCSS assumes that both shadow blocker and shadow receiver surfaces are planar and parallel to the light source. Then, the average blocker depth is computed in a given search area and the penumbra size can be estimated according to the aforementioned planar-parallel assumption. At the penumbra size location, the Percentage-Closer Filtering (PCF) [17] is applied to compute the soft shadows. Since the PCF is not scalable with respect to the filter size, the PCSS algorithm may suffer from banding artifacts for large penumbra sizes, because real-time performance can be obtained at the cost of the usage of a low-



(a) Aliased Shadow (b) Anti-aliased Shadow (c) Penumbra Size (d) EDT Shadowing (e) EDT Filtering

Fig. 1. An overview of the EDTSSM approach. We estimate hard shadows in the camera view with the traditional shadow mapping (a) and minimize the aliasing artifacts with the shadow revectorization (b). On the basis of the percentage-closer soft shadows, we estimate the penumbra size for the fragments located at the hard shadow silhouette (c), and use it to normalize a Euclidean distance transform (EDT) which is computed to measure the distance of each fragment to the closest shadow silhouette, simulating the penumbra effect (d). To suppress the skeleton artifacts generated by the EDT, an edge-aware viewpoint-invariant filtering algorithm is applied over the shadow (e). Images were generated for the QuadBot model using a 512^2 shadow map resolution.

order filter size. Also, because the PCSS algorithm uses the shadow test of the shadow mapping as visibility function, aliasing artifacts may arise along the shadow silhouette for small penumbra sizes.

Since the proposition of PCSS, many techniques have replaced the PCF step by more scalable shadow map pre-filtering techniques to reduce the processing time of the soft shadow mapping. Summed-Area Variance Shadow Mapping (SAVSM) [18] and Variance Soft Shadow Mapping (VSSM) [19], [20] uses the Chebyshev's Inequality solution proposed in [21] to optimize the shadow filtering. Convolution Soft Shadow Mapping (CSSM) [22] uses the Fourier basis to pre-filter the shadow map [23] and compute the soft shadows. Exponential Soft Shadow Mapping (ESSM) [24] uses an exponential function [25], [26] to reduce the memory footprint of the CSSM. Moment Soft Shadow Mapping (MSSM) [27], [28] uses a solution of the Hamburger moment problem [29] to improve the accuracy of the VSSM.

Differently from the PCSS framework, Selgrad et al. [30] assume that, for a scene with an area light source and a planar blocker, visually plausible soft shadows may be computed by approximating the scene configuration with a point light source and a semitransparent blocker. In this case, a multi-layer shadow map [31] must be captured and pre-filtered to generate soft shadows.

The main problem with the soft shadow techniques which use shadow map pre-filtering is that they are more prone to light leaking artifacts than related work. Also, they suffer from performance issues for high shadow map resolutions.

Temporal coherence has already been exploited to generate visually plausible soft shadows on the basis of the PCSS framework [32]–[34]. While these techniques are able to provide realistic soft shadows in real-time, they require several frames to converge to a correct solution. Also, the frame rate may change considerably between frames for dynamic scenes with moving objects, cameras or light sources, being inadequate for applications which demand constant frame rates, such as games. Despite these shortcomings, temporal coherence is a complementary approach which could be integrated into our approach as well.

To improve the performance of PCSS, Klein et al. [35] compute the average blocker depth and estimate the penumbra size only for the fragments located at the hard shadow silhouette. Then, for each fragment outside the shadow silhouette, a gathering approach and an erosion operation are used to locate the shadow silhouette and estimate the penumbra intensity of the soft shadows. This approach is slightly faster than PCSS for large kernel sizes, but is still prone to aliasing artifacts at the penumbra location.

In this paper, we aim to solve some of the problems found in PCSS and the works which use its basis framework to compute soft shadows. To do so, we estimate the anti-aliased hard shadows in the camera view and, similarly to the work of Klein et al., we compute the penumbra size in the location of the shadow silhouette only. Then, we use a filtered normalized Euclidean distance transform to compute the penumbra intensities and generate the soft shadows in real-time.

It is worthy to mention that, since all these soft shadow techniques (including ours) work on the basis of a single shadow map and a single light source sample, they tend to produce only visually plausible soft shadows, but are capable to achieve real-time performance. An accurate soft shadow computation would require a more complete sampling of the area light source, at the cost of achieving near interactive performance [36], [37].

III. Euclidean Distance Transform Soft Shadow Mapping

2D Euclidean distance transform may be defined as an operation which assigns to the pixel intensity, the Euclidean distance of the pixel to the nearest pixel located in a region of interest [38]. If we assume that the intensity of a pixel, located in the penumbra of a soft shadow, can be computed by the

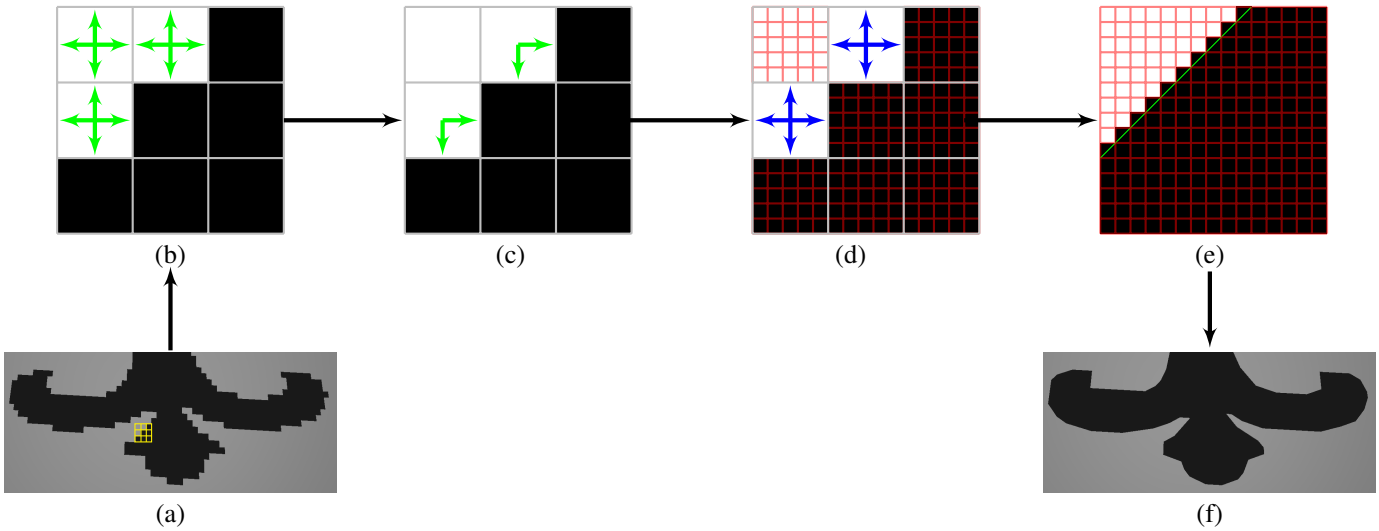


Fig. 2. An overview of the hard shadow revectorization algorithm. Given an aliased shadow silhouette (yellow grid in (a)), shadow revectorization evaluates the spatial coherency between shadow tests (b) in the projected light space (gray grid), detects the directions of where the shadow silhouette is located (c), estimates its size in the camera view (red grid) (d), and traces a revectorization line (green line in (e)) to determine whether the fragment should be included in the revectorized shadow. Shadow revectorization is able to minimize shadow aliasing artifacts at little additional cost (f).

Algorithm 1 Euclidean distance transform soft shadow mapping

1: **for each frame do**
2: $I_{SM} \leftarrow$ RENDERSHADOWMAP;
3: $I_{GB} \leftarrow$ RENDERGBUFFER;
4: $I_{RS} \leftarrow$ COMPUTEREVECTORIZATION(I_{SM}, I_{GB});
5: $I_{PS} \leftarrow$ ESTIMATEPENUMBRASIZE(I_{RS}, I_{SM}, I_{GB});
6: $I_{EDT} \leftarrow$ COMPUTEEDTSHADOWS(I_{RS}, I_{PS}, I_{GB});
7: $I_{MF} \leftarrow$ FILTEREDTSHADOWS(I_{EDT});
8: RENDERSHADOWEDSCENE(I_{MF}, I_{GB});
9: **end for**

normalized Euclidean distance of the pixel to the nearest pixel located out of the penumbra (in an umbra region, for instance), we can use the Euclidean distance transform as a basis to estimate the soft shadow intensity of a penumbra fragment. To provide a well-defined penumbra simulation, the Euclidean distance transform must operate over an anti-aliased shadow region, which can be generated by recent shadow mapping techniques, such as the revectorization-based shadow mapping [3]. This would prevent the soft shadow technique to suffer from banding, light leaking or aliasing artifacts typically seen in the majority of the existing real-time soft shadow techniques and caused by the filtering of the shadow map, which stores an undersampled representation of the light blocker object.

In Algorithm 1, we present a high-level overview of the proposed approach to compute soft shadows on the basis of a normalized EDT. This algorithm is depicted in Figure 1. The first two steps in Algorithm 1 render the scene from the light source and camera viewpoint to produce a hard shadow, as we can see in Figure 1-(a). To reduce the aliasing artifacts, our method applies a revectorization step (Line 4 of Algorithm 1) to improve the quality of the hard shadow, as shown in Figure 1-(b). Then, we estimate the penumbra size of the fragments located in the shadow silhouette (Figure 1-(c), Line 5 of Algorithm 1), such that we can apply the normalized EDT to compute the soft shadow (Figure 1-(d), Line 6 of Algorithm 1) and filter it to suppress the skeleton artifacts generated by the application of the EDT (Figure 1-(e), Line 7 of Algorithm 1). Further details of each step of the proposed technique are presented in the next subsections.

A. Shadow Map Rendering

Initially, we render the basic structures to generate soft shadows in real-time: a shadow map (I_{SM}) from the light source viewpoint to capture the closest fragments seen from the point light source [1], and a G-buffer (I_{GB}) [39] from the camera viewpoint to remove hidden fragments from the soft shadow calculation. Each pixel of I_{SM} stores the distance of the light source to the closest fragments visible to the light source. As for I_{GB}, we store the world-space position and normals of each visible fragment.

Unfortunately, for shadow maps of low-resolution, shadow mapping produces hard shadows with undesirable aliasing artifacts, which introduce a jagged appearance in the hard shadow silhouette (Figure 1-(a)). To compute high-quality hard shadows efficiently, regardless of the shadow map resolution, we apply the hard shadow revectorization algorithm [3] over the aliased hard shadows to recover an approximate accurate representation of the shadow silhouette (Figure 1-(b)).

B. Hard Shadow Revectorization

Hard shadow revectorization [3], [40] uses the available screen-space resolution of the camera view to improve the accuracy of the shadow estimation. Given the jagged shadows generated by shadow mapping (Figure 2-(a)), the first step

of the shadow revectorization consists in the detection of the jagged shadow boundaries, where the aliasing artifacts are located. To do so, each fragment in the camera view is projected into the shadow map, and the result of the shadow test for that fragment is compared against the shadow test results of the 1-ring neighbourhood of the fragment projected in the shadow map (Figure 2-(b)). Since the hard shadow revectorization is performed for the external (lit) side of the shadow silhouette, only fragments estimated to be lit by the shadow test are compared to their neighbours in the shadow map.

After the neighbourhood evaluation, whenever a difference between shadow test results is detected, the direction which determines where the closest shadow boundaries are located is stored by the algorithm in a discontinuity map (Figure 2-(c)). Then, this direction is used to guide a traversal over the shadow silhouette, which is performed to estimate the size of the shadow silhouette, as well as the relative position of the camera view fragment inside the shadow silhouette (Figure 2-(d)). By using the relative position of the fragment with respect to the shadow silhouette, the algorithm is able to determine whether the fragment is located inside or outside the revectorized shadow (Figure 2-(e)) and shade the fragment accordingly (Figure 2-(f)). As shown in Figure 1-(b), shadow revectorization minimizes the aliasing artifacts generated by the finite shadow map resolution, and generates high-quality hard shadows even for a low-resolution shadow map.

This shadow revectorization step is implemented in a single pass on the shader, following [3]. The result of this step is the generation of an image I_{RS} which stores, for each pixel, whether the corresponding fragment is in umbra or lit (Line 4 of Algorithm 1).

C. Penumbra Size Estimation

For each fragment in the camera view, we must determine whether the fragment is located in a penumbra region, and what is the entire penumbra region on the world-space, such that we can compute the soft shadows on the basis of an EDT. The PCSS framework proposes that the penumbra size must be computed for every fragment in the camera view, since this size determines the area of the shadow map which must be filtered by the algorithm [2]. Similarly to [35], we optimize such approach by computing the penumbra size only for the fragments located at the hard shadow silhouette and propagating such a data for the remaining fragments outside the shadow silhouette. Differently from [35], we compute the penumbra size for the fragments located at the revectorized, anti-aliased hard shadow silhouette. Also, we propagate the penumbra size in real-time using the nearest neighbour search provided by the EDT algorithm.

To determine whether a fragment is located in the shadow silhouette in the image with the revectorized hard shadows (I_{RS}, Figure 1-(b)), we check whether the visibility condition of the current fragment is different from at least one of the neighbours located inside the filter region. Then, for each fragment in the shadow silhouette, we take advantage of the planar-parallel assumption of the PCSS to estimate the penumbra size in real-time (Figure 1-(c)).

Following the PCSS framework, we need to compute the average blocker depth of the fragment to be able to compute the penumbra size. This is done by applying a filter which locates the blockers of the fragment according to the shadow test result [1] and averages their depths, which are stored in the shadow map. In this step, the blocker search area of the filter is defined by the intersection of the near plane of the shadow map and the frustum formed by the fragment and the light source.

After the average blocker depth computation, the penumbra size w_p can be estimated by using similarity of triangles, taking into consideration the light source size w_l, the average blocker depth z_{avg} and the distance of the fragment to the light source $\mathbf{p}_z$ [2]

$$w_p = w_l \frac{\mathbf{p}_z - z_{avg}}{z_{avg}}. \tag{1}$$

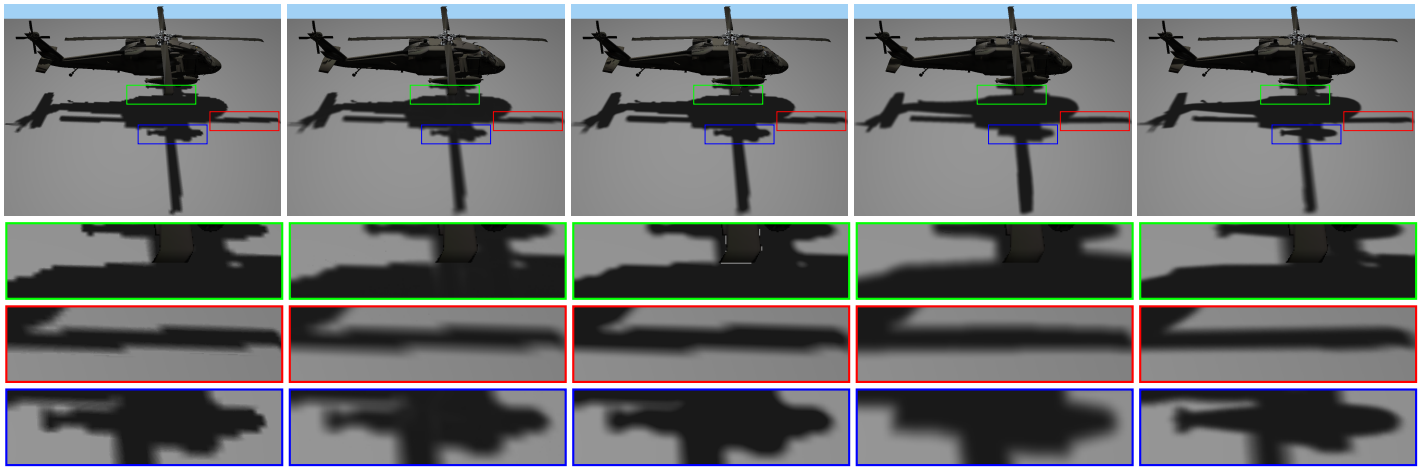
The result of the penumbra size estimation step is the generation of the image I_{PS}, which stores the value of w_p for the pixels whose corresponding fragments are located in the hard shadow silhouette computed from I_{RS} (Line 5 of Algorithm 1).

D. Euclidean Distance Transform Shadowing

Once we have the penumbra size computed for every fragment located in the revectorized hard shadow silhouette, we need to propagate this data for the fragments located outside the shadow silhouette. As we show in this subsection, according to the distance of the fragments to the shadow silhouette and the penumbra size data, we are able to determine whether each fragment is in the penumbra, and what is the soft shadow intensity of the fragment.

To locate and compute the exact distance of each fragment to the closest shadow silhouette at the same time, in real-time, we make use of the EDT algorithm proposed by Cao and colleagues, the Parallel Banding Algorithm (PBA) [41]. PBA divides the input image into vertical and horizontal bands and computes an 1D Voronoi diagram for each band in parallel. Then, the diagram data is propagated between different bands and is finally used to compute the EDT. The PBA is highly parallelizable, and, to the best of our knowledge, it is the fastest and most accurate algorithm proposed so far to compute exact EDT in real-time.

By locating the closest shadow silhouette, we are able to retrieve the penumbra size w_p stored at the corresponding position in I_{PS}. Since w_p was estimated with respect to the light space, it cannot be directly used to define the size of the penumbra in the world-space, since this task is non-trivial [42]. Hence, we multiply w_p by a user-defined parameter β, which helps with the definition of the penumbra size in the world-space. In our experimental tests, we have verified that $\beta = 750$ keeps the penumbra size similar to the one found in the most common soft shadow mapping algorithm, such as PCSS.



(a) PCSS (b) VSSM (c) MSSM (d) EDTSSM (e) Ground-truth

Fig. 3. Soft shadows produced by different techniques. The closeups show whether the techniques suffer from light leaking artifacts caused by pre-filtering (green closeup), aliasing artifacts (red closeup) or shadow overestimation (blue closeup). Images were generated for the Helicopter model using a 512^2 shadow map resolution. Ground-truth image was produced using an average of 1024 area light source samples.

Now, let us define the world-space Euclidean distance d as the distance of each fragment to the closest fragment located in the revectorized shadow silhouette. The world-space value d is computed by retrieving the world-space position of the fragments stored in I_{GB}. Given that the inner (shadowed) and outer (lit) sides of the penumbra must have half of the user-defined penumbra size ($\frac{\beta w_p}{2}$), the visibility condition of the fragment $\mathbf{p}$ is estimated as

$$\mathbf{p} \text{ is } = \begin{cases} \text{in penumbra} & \text{if } d \leq \frac{\beta w_p}{2}, \\ \text{outside penumbra} & \text{otherwise.} \end{cases} \quad (2)$$

As shown in (2), a fragment is in penumbra if its world-space Euclidean distance to the closest shadow silhouette is less than a half of the penumbra size.

For the fragments located in the penumbra, the next step consists in determining the soft shadow intensity of the fragment, which we assume that is proportional to the distance of the fragment to the closest fragment located in the shadow silhouette. However, we need to normalize such a distance since the penumbra intensities must lie in the closed interval $[0, 1]$, characterizing the smooth intensity variation of the penumbra region, which goes from the absence of direct light (in the umbra region) to the full direct illumination (in the lit region).

For a fragment $\mathbf{p}$ in penumbra, we can compute its penumbra intensity α as

$$\alpha = \begin{cases} \frac{1}{2} - \frac{d}{\beta w_p} & \text{if } \mathrm{I}_{RS}(\mathbf{p}) = \text{umbra}, \\ \frac{1}{2} + \frac{d}{\beta w_p} & \text{otherwise}, \end{cases} \quad (3)$$

which states that, fragments located in the edge of the penumbra size ($d = \frac{\beta w_p}{2}$) have full umbra ($\alpha = 0$) or lit ($\alpha = 1$) intensities, meanwhile fragments inside the penumbra have the intensities lying in the closed interval $[0, 1]$.

As a result of this step, we generate an image I_{EDT}, which stores the shadow, penumbra or lit intensity of each fragment in the camera view (Line 6 of Algorithm 1).

E. Euclidean Distance Transform Filtering

As shown in Figure 1-(d), the use of EDT to compute the soft shadows may generate skeleton artifacts at locations with gradient discontinuities [43]. To suppress both artifacts, we apply a separable mean filter over the soft shadows in the screen-space. To make this screen-space filter edge-aware, we take into account the depth difference between fragments when applying the mean filter. Indeed, we only include a neighbour fragment in the filtering if the difference between the depth of the current fragment and its neighbour is below a user-defined depth threshold (in all our experiments, we have empirically set the depth threshold as 0.0025). Also, since we are applying this filter in the screen-space, we need to make it viewpoint-invariant, such that the size of the filter varies according to the distance of the scene and the camera. We estimate this variable mean filter size w_f^s for each fragment [44] as

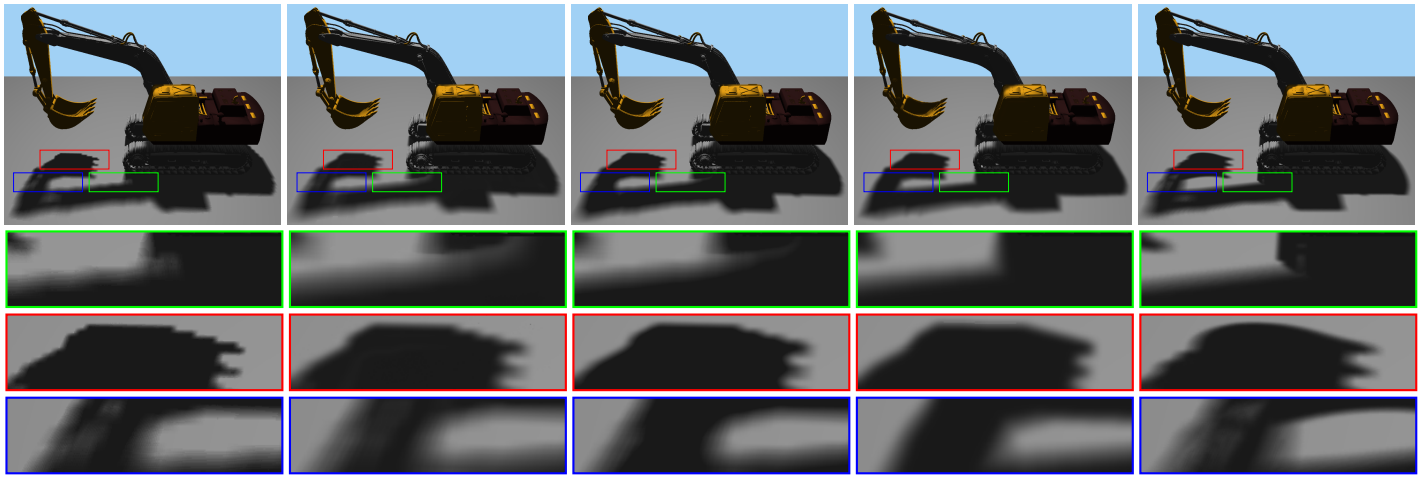
$$\begin{aligned} w_f^s &= \frac{w_f z_s}{\mathbf{p}_{z_{eye}}}, \\ z_s &= \frac{1}{2\tan\frac{fov_y}{2}}, \end{aligned} \quad (4)$$

where fov_y specifies the vertical field of view, $\mathbf{p}_{z_{eye}}$ is the depth of the fragment $\mathbf{p}$ with respect to the camera. w_f is the mean filter size.

IV. Results and Discussion

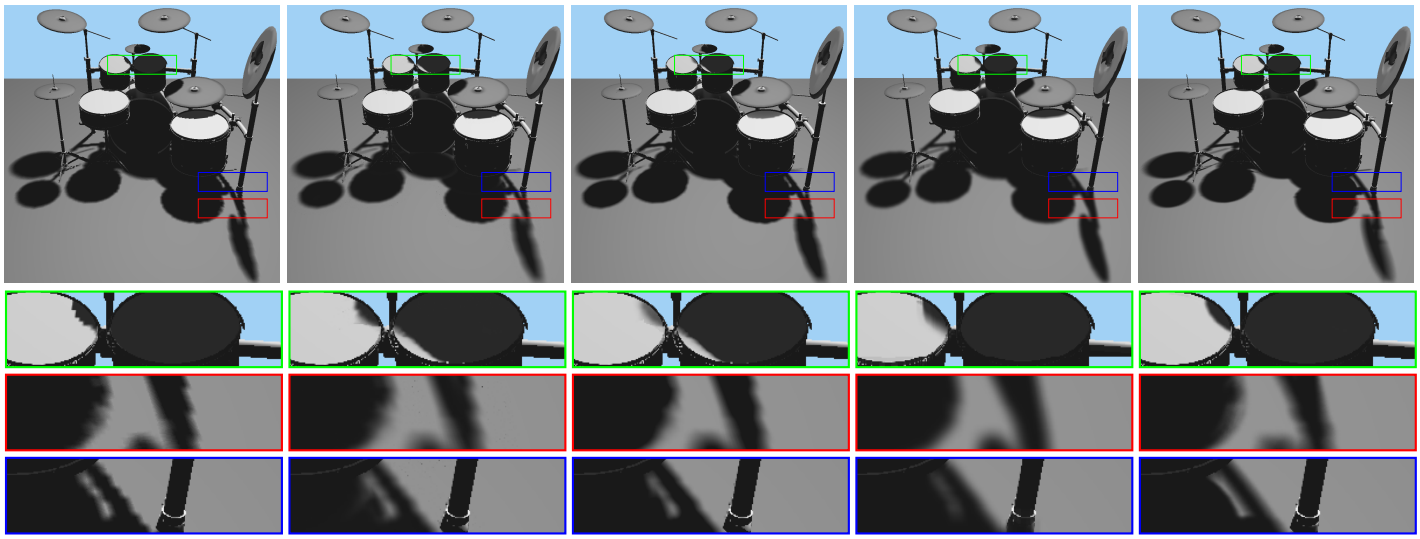
A. Experimental Setup

In this section, we compare our approach with the related work in terms of quality and performance. We used an Intel Core™ i7-3770K CPU (3.50 GHz), 8GB RAM, and an



(a) PCSS (b) VSSM (c) MSSM (d) EDTSSM (e) Ground-truth

Fig. 4. Soft shadows produced by different techniques. The closeups show whether the techniques suffer from light leaking artifacts caused by pre-filtering (green closeup), aliasing artifacts (red closeup) or shadow overestimation (blue closeup). Images were generated for the Excavator model using a 512^2 shadow map resolution. Ground-truth image was produced using an average of 1024 area light source samples.



(a) PCSS (b) VSSM (c) MSSM (d) EDTSSM (e) Ground-truth

Fig. 5. Soft shadows produced by different techniques. The closeups show whether the techniques suffer from light leaking artifacts caused by pre-filtering (green closeup), aliasing artifacts (red closeup) or shadow overestimation (blue closeup). Images were generated for the Drum model using a 1024^2 shadow map resolution. Ground-truth image was produced using an average of 1024 area light source samples.

NVIDIA GeForce GTX Titan X graphics card to run the tests. Additional results of this work, such as temporal coherency, can be seen in the supplementary video.

Since many soft shadow techniques have been proposed in the literature, we have focused on the comparison of our approach with the other ones which generate soft shadows in real-time: the traditional PCSS algorithm [2], the VSSM [19], [20] and MSSM [27], [28] pre-filtering soft shadow techniques.

The filter size greatly influences in the quality of the soft shadow filtering, either by affecting the penumbra visual quality, reducing potential banding artifacts, or by affecting the performance, increasing the rendering time. Similarly to [27], we used a filter size of 9×9 for the blocker search step and a filter size of 15×15 for the soft shadow filtering step for all the techniques evaluated in this section. For EDTSSM, the filter size of 15×15 refers to the mean filter size (w_f in (4)), used for EDT filtering.

B. Rendering Quality

In Figures 3, 4 and 5, we show the output produced by different soft shadow mapping techniques for small, large penumbra sizes and shadows cast on curved surfaces, respectively.

TABLE I
RENDERING PERFORMANCE FOR DIFFERENT SOFT SHADOW MAPPING TECHNIQUES FOR THREE SCENES SHOWN IN THIS PAPER. MEASUREMENTS INCLUDE VARYING SHADOW MAP RESOLUTION RENDERED AT A VIEWPORT $720p$ RESOLUTION.

		Shadow Map Resolution			
Scene	**Method**	512^2	1024^2	2048^2	4096^2
Fig. 3	PCSS	2.8 ms	2.9 ms	3.0 ms	3.1 ms
	VSSM	2.0 ms	4.0 ms	6.5 ms	8.2 ms
	MSSM	1.8 ms	3.6 ms	5.1 ms	6.5 ms
	EDTSSM	4.3 ms	4.5 ms	4.6 ms	5.0 ms
Fig. 4	PCSS	3.4 ms	3.5 ms	3.8 ms	4.0 ms
	VSSM	2.7 ms	4.6 ms	6.4 ms	8.2 ms
	MSSM	2.4 ms	4.1 ms	5.5 ms	7.0 ms
	EDTSSM	5.5 ms	5.6 ms	5.7 ms	6.1 ms
Fig. 5	PCSS	4.8 ms	4.9 ms	5.0 ms	5.6 ms
	VSSM	4.7 ms	6.2 ms	7.0 ms	9.1 ms
	MSSM	4.2 ms	5.7 ms	6.5 ms	8.2 ms
	EDTSSM	6.7 ms	6.8 ms	7.0 ms	7.2 ms

TABLE II
RENDERING TIME FOR DIFFERENT SOFT SHADOW MAPPING TECHNIQUES FOR THREE SCENES SHOWN IN THIS PAPER. MEASUREMENTS INCLUDE THE MOST COMMON VIEWPORT RESOLUTIONS RENDERED AT A 1024^2 SHADOW MAP RESOLUTION.

		Viewport Resolution		
Scene	**Method**	$480p$	$720p$	$1080p$
Fig. 3	PCSS	1.4 ms	2.9 ms	3.1 ms
	VSSM	3.8 ms	4.0 ms	4.3 ms
	MSSM	3.2 ms	3.6 ms	4.0 ms
	EDTSSM	2.3 ms	4.5 ms	5.9 ms
Fig. 4	PCSS	2.1 ms	3.5 ms	4.2 ms
	VSSM	4.3 ms	4.6 ms	5.0 ms
	MSSM	3.7 ms	4.1 ms	4.4 ms
	EDTSSM	3.0 ms	5.6 ms	6.5 ms
Fig. 5	PCSS	3.8 ms	4.9 ms	5.3 ms
	VSSM	5.8 ms	6.2 ms	6.6 ms
	MSSM	5.3 ms	5.7 ms	6.0 ms
	EDTSSM	4.5 ms	6.8 ms	8.4 ms

TABLE III
RENDERING PERFORMANCE FOR DIFFERENT SOFT SHADOW MAPPING TECHNIQUES FOR VARYING KERNEL SIZE. TIMES WERE MEASURED FOR A VIEWPORT $720p$ RESOLUTION AND A 1024^2 SHADOW MAP RESOLUTION.

		Kernel Size		
Scene	**Method**	7×7	15×15	23×23
Fig. 3	PCSS	1.7 ms	2.9 ms	4.0 ms
	VSSM	3.9 ms	4.0 ms	4.1 ms
	MSSM	3.5 ms	3.6 ms	3.6 ms
	EDTSSM	4.4 ms	4.5 ms	4.6 ms
Fig. 4	PCSS	2.4 ms	3.5 ms	4.5 ms
	VSSM	4.6 ms	4.6 ms	4.7 ms
	MSSM	4.1 ms	4.2 ms	4.2 ms
	EDTSSM	5.5 ms	5.6 ms	5.8 ms
Fig. 5	PCSS	4.2 ms	4.9 ms	5.5 ms
	VSSM	6.2 ms	6.2 ms	6.2 ms
	MSSM	5.7 ms	5.7 ms	5.8 ms
	EDTSSM	6.7 ms	6.8 ms	6.9 ms

In the green closeups of Figures 3, 4 and 5, we show that the VSSM technique suffers from different light leaking artifacts caused by the pre-filtering of the shadow map. Hence, umbra regions are incorrectly rendered as penumbra (Figures 3-(b) and 4-(b)) or lit regions (Figure 5-(b)). MSSM generates less light leaking artifacts than VSSM with similar performance [27], [28]. PCSS and EDTSSM are less prone to these kind of artifacts, but they generate different shadow intensities from the ground-truth ones due to the single light source approximation of the area light source.

From the red closeups of Figures 3, 4 and 5, we see that both PCSS, VSSM and MSSM techniques may generate aliasing artifacts along the penumbra regions of the shadow. By the use of the shadow revectorization, EDTSSM is able to reduce those artifacts efficiently.

One of the limitations of EDTSSM is shown in the blue closeups of Figures 3, 4 and 5. Our approach causes shadow overestimation mainly for umbra regions (Figure 4-(d)), potentially merging disconnected parts of the shadow silhouette if they are close to each other (Figures 3-(d) and 5-(d)). In our approach, the shadow overestimation is caused mainly by the shadow revectorization (Figure 1-(b)), which is not able to separate the different parts of the shadow silhouette.

C. Performance

In terms of performance, pre-filtering techniques are not scalable with respect to the shadow map resolution, as shown in Table I, because of the summed-area table building, which becomes the performance bottleneck for high-resolution shadow maps, as already discussed in [20], [24], [45]. On the other hand, these techniques are more scalable to the viewport resolution (Table II) and filter kernel sizes (Table III) than the PCSS approach, although they may be slower than PCSS in some scenarios, such as the one obtained for Figure 5.

As shown in Table I, the EDTSSM is scalable with respect to the shadow map resolution. For low-resolution shadow maps (i.e., 512^2 and 1024^2), our technique is slower than related work, but, as shown in Figures 3, 4 and 5, our approach generates soft shadows with less aliasing artifacts than related work, providing high accuracy at the cost of slight loss in performance. On the other hand, EDTSSM is faster than the pre-filtering soft shadow techniques for high-resolution shadow maps (i.e., 2048^2 and 4096^2), configuration in which the pre-filtering techniques still suffer from light leaking artifacts and the PCSS technique is prone to small aliasing artifacts. Hence, the visual quality and performance results show that the EDTSSM is suitable to be used for scenarios with low- and high-resolution shadow maps.

EDTSSM is not as scalable as related work with respect to the viewport resolution (Table II). Even in this case, EDTSSM is faster than the pre-filtering techniques for low-viewport resolutions ($480p$), and provides competitive performance for mid- ($720p$) and high-viewport resolutions ($1080p$).

Similarly to the pre-filtering techniques, EDTSSM is scalable with respect to the kernel size (Table III), because of the mean filter algorithm used to suppress skeleton artifacts.

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a new approach which uses the concept of Euclidean distance transform to compute high-quality soft shadows in real-time. Compared to related work, the EDTSSM technique generates anti-aliased soft shadows

with reduced light leaking artifacts. However, shadow overestimation makes the approach less accurate in the places where different penumbra regions are fused into a single one.

Future work may solve the shadow overestimation problem by the reformulating the visibility function of the shadow revectorization. Also, one could adapt the use of Euclidean distance transform to compute highly accurate soft shadows on the basis of an area light source sampling.

ACKNOWLEDGMENT

The authors would like to thank Cao et al. [41] for sharing the source code of PBA, the NVIDIA Corporation for providing the NVIDIA GeForce GTX Titan X through the GPU Education Center program, and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) for financial support. Helicopter, Excavator and Drum models are courtesy of Free3D (3dregenerator and drumssultan users).

REFERENCES

[1] L. Williams, "Casting curved shadows on curved surfaces," in *Proceedings of the ACM SIGGRAPH*, 1978, pp. 270–274.

[2] R. Fernando, "Percentage-closer soft shadows," in *ACM SIGGRAPH Sketches*, 2005.

[3] M. Macedo and A. Apolinario, "Revectorization-based shadow mapping," in *Proceedings of the GI*, 2016, pp. 75–83.

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

[5] A. Woo and P. Poulin, *Shadow Algorithms Data Miner*. CRC Press, 2012.

[6] S. Brabec and H.-P. Seidel, "Single sample soft shadows using depth maps," in *Proceedings of the GI*, 2002, pp. 219–228.

[7] F. Kirsch and J. Dollner, "Real-time Soft Shadows using a Single Light Sample," in *Proceedings of the WSCG*, 2003, pp. 255–262.

[8] W. H. de Boer, "Smooth penumbra transitions with shadow maps," *J. Graphics Tools*, vol. 11, no. 2, pp. 59–71, 2006.

[9] E. Chan and F. Durand, "Rendering fake soft shadows with smoothies," in *Proceedings of the EGSR*, 2003, pp. 208–218.

[10] B. Aszodi and L. Szirmay-Kalos, "Real-time Soft Shadows with Shadow Accumulation," in *Eurographics Short Papers*, 2006, pp. 53–56.

[11] L. Atty, N. Holzschuch, M. Lapierre, J.-M. Hasenfratz, C. Hansen, and F. Sillion, "Soft shadow maps: Efficient sampling of light source visibility," *Computer Graphics Forum*, vol. 25, no. 4, pp. 725–741, dec 2006.

[12] L. Bavoil, S. P. Callahan, and C. T. Silva, "Robust soft shadow mapping with backprojection and depth peeling." *J. Graphics Tools*, vol. 13, no. 1, pp. 19–30, 2008.

[13] G. Guennebaud, L. Barthe, and M. Paulin, "Real-time soft shadow mapping by backprojection," in *Proceedings of the EGSR*, 2006, pp. 227–234.

[14] ——, "High-quality adaptive soft shadow mapping," *Computer Graphics Forum*, vol. 26, no. 3, pp. 525–533, 2007.

[15] M. Schwarz and M. Stamminger, "Bitmask soft shadows," *Computer Graphics Forum*, vol. 26, no. 3, pp. 515–524, 2007.

[16] B. Yang, J. Feng, G. Guennebaud, and X. Liu, "Packet-based hierarchal soft shadow mapping," in *Proceedings of the EGSR*, 2009, pp. 1121–1130.

[17] 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.

[18] A. Lauritzen, "Summed-area variance shadow maps," in *GPU Gems 3*, H. Nguyen, Ed. Addison-Wesley, 2008, pp. 157–182.

[19] Z. Dong and B. Yang, "Variance soft shadow mapping," in *Proceedings of the ACM I3D*, 2010, pp. 1–1.

[20] B. Yang, Z. Dong, J. Feng, H.-P. Seidel, and J. Kautz, "Variance soft shadow mapping," *Computer Graphics Forum*, vol. 29, no. 7, pp. 2127–2134, 2010.

[21] W. Donnelly and A. Lauritzen, "Variance shadow maps," in *Proceedings of the ACM I3D*, 2006, pp. 161–165.

[22] T. Annen, Z. Dong, T. Mertens, P. Bekaert, H.-P. Seidel, and J. Kautz, "Real-time, all-frequency shadows in dynamic scenes," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–8, Aug. 2008.

[23] T. Annen, T. Mertens, P. Bekaert, H.-P. Seidel, and J. Kautz, "Convolution Shadow Maps," in *Proceedings of the EGSR*, 2007, pp. 51–60.

[24] L. Shen, J. Feng, and B. Yang, "Exponential soft shadow mapping," *Computer Graphics Forum*, vol. 32, no. 4, pp. 107–116, 2013.

[25] T. Annen, T. Mertens, H.-P. Seidel, E. Flerackers, and J. Kautz, "Exponential shadow maps," in *Proceedings of the GI*, 2008, pp. 155–161.

[26] M. Salvi, "Rendering filtered shadows with exponential shadow maps," in *ShaderX 6.0 Advanced Rendering Techniques*. Charles River Media, 2008, pp. 257–274.

[27] 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.

[28] ——, "Improved moment shadow maps for translucent occluders, soft shadows and single scattering," *Journal of Computer Graphics Techniques (JCGT)*, vol. 6, no. 1, pp. 17–67, March 2017.

[29] C. Peters and R. Klein, "Moment shadow mapping," in *Proceedings of the ACM I3D*, 2015, pp. 7–14.

[30] K. Selgrad, C. Dachsbacher, Q. Meyer, and M. Stamminger, "Filtering multi-layer shadow maps for accurate soft shadows," *Computer Graphics Forum*, vol. 34, no. 1, pp. 205–215, 2015.

[31] F. Xie, E. Tabellion, and A. Pearce, "Soft shadows by ray tracing multilayer transparent shadow maps," in *Proceedings of the EGSR*, 2007, pp. 265–276.

[32] D. Scherzer, M. Schwärzler, O. Mattausch, and M. Wimmer, "Real-time soft shadows using temporal coherence," in *Proceedings of the ISVC*, ser. Lecture Notes in Computer Science, 2009, pp. 13–24.

[33] D. Scherzer, M. Schwärzler, and O. Mattausch, "Fast soft shadows with temporal coherence," in *GPU Pro 2*, W. Engel, Ed. A.K. Peters, Feb. 2011.

[34] M. Schwärzler, C. Luksch, D. Scherzer, and M. Wimmer, "Fast percentage closer soft shadows using temporal coherence," in *Proceedings of the ACM I3D*, 2013, pp. 79–86.

[35] A. Klein, A. Nischwitz, and P. Obermeier, "Contact Hardening Soft Shadows using Erosion," in *Proceedings of the WSCG*, 2012.

[36] E. Sintorn, E. Eisemann, and U. Assarsson, "Sample Based Visibility for Soft Shadows Using Alias-free Shadow Maps," in *Proceedings of the EGSR*, 2008, pp. 1285–1292.

[37] M. Schwarzler, O. Mattausch, D. Scherzer, and M. Wimmer, "Fast Accurate Soft Shadows with Adaptive Light Source Sampling," in *Proceedings of the VMV*, Nov. 2012, pp. 39–46.

[38] A. Rosenfeld and J. L. Pfaltz, "Sequential operations in digital picture processing," *J. ACM*, vol. 13, no. 4, pp. 471–494, Oct. 1966.

[39] T. Saito and T. Takahashi, "Comprehensible rendering of 3-d shapes," in *Proceedings of the ACM SIGGRAPH*, 1990, pp. 197–206.

[40] V. Bondarev, "Shadow map silhouette revectorization," in *Proceedings of the ACM I3D*, 2014, pp. 162–162.

[41] T.-T. Cao, K. Tang, A. Mohamed, and T.-S. Tan, "Parallel Banding Algorithm to Compute Exact Distance Transform with the GPU," in *Proceedings of the ACM I3D*, 2010, pp. 83–90.

[42] R. Fernando, S. Fernandez, K. Bala, and D. P. Greenberg, "Adaptive shadow maps," in *Proceedings of the ACM SIGGRAPH*, 2001, pp. 387–390.

[43] M. W. Wright, R. Cipolla, and P. J. Giblin, "Skeletonization using an extended Euclidean distance transform," *Image and Vision Computing*, vol. 13, no. 5, pp. 367 – 375, 1995.

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

[45] J. Hensley, T. Scheuermann, G. Coombe, M. Singh, and A. Lastra, "Fast summed-area table generation and its applications," *Computer Graphics Forum*, vol. 24, no. 3, pp. 547–555, 2005.