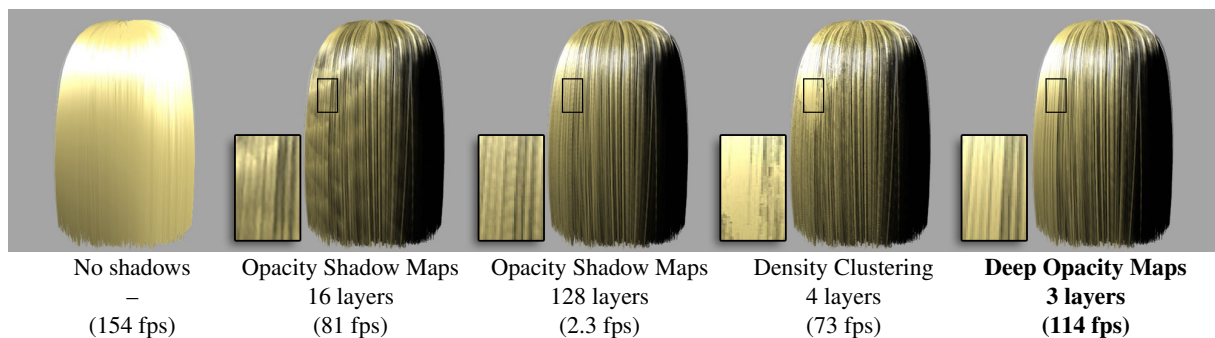# Deep Opacity Maps

Cem Yuksel[1]   and   John Keyser[2]

Department of Computer Science, Texas A&M University
[1]cem@cemyuksel.com   [2]keyser@cs.tamu.edu

| No shadows – (154 fps) | Opacity Shadow Maps 16 layers (81 fps) | Opacity Shadow Maps 128 layers (2.3 fps) | Density Clustering 4 layers (73 fps) | **Deep Opacity Maps 3 layers (114 fps)** |

Figure 1: *Layering artifacts of Opacity Shadow Maps are visible even with 128 layers, while Density Clustering has artifacts due to inaccuracies. Deep Opacity Maps with only 3 layers can generate an artifact free image with the highest frame rate.*

Abstract

We present a new method for rapidly computing shadows from semi-transparent objects like hair. Our deep opacity maps method extends the concept of opacity shadow maps by using a depth map to obtain a per pixel distribution of opacity layers. This approach eliminates the layering artifacts of opacity shadow maps and requires far fewer layers to achieve high quality shadow computation. Furthermore, it is faster than the density clustering technique, and produces less noise with comparable shadow quality. We provide qualitative comparisons to these previous methods and give performance results. Our algorithm is easy to implement, faster, and more memory efficient, enabling us to generate high quality hair shadows in real-time using graphics hardware on a standard PC.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Color, shading, shadowing, and texture

Keywords: shadow maps, semi-transparent shadows, hair shadows, real-time shadows, GPU algorithms

1. Introduction

Self-shadowing is an essential visual element for rendering semi-transparent objects like hair, fur, smoke, and clouds. However, handling the transparency component is either inefficient or not possible for simple shadowing techniques. Various algorithms have been proposed to address this issue both for offline rendering [LV00, AL04] and interactive/real-time rendering [KN01, MKBvR04]. In this paper we present the *deep opacity maps* method, which allows real-time hair rendering with dynamic lighting and semi-transparent shadows. This new method is faster than the previous ones and produces artifact free shadows (Figure 1). Even though we focus on hair shadows, our method is applicable to other semi-transparent objects.

The deep opacity maps method combines shadow mapping [Wil78] and opacity shadow maps [KN01] to give a

better distribution of opacity layers. We first render the hair geometry as opaque primitives from the light's view, recording the depth values on a shadow map. Next we render an opacity map from the light's view similar to opacity shadow maps. The novelty of our algorithm lies in the way that the opacity layers are distributed using the depth map to create opacity layers that vary in depth from the light source on a per-pixel basis. Unlike previous interactive/real-time transparent shadowing techniques [KN01, MKBvR04], this new layer distribution guarantees that the direct illumination coming from the light source without being shadowed is captured correctly. This property of deep opacity maps eliminates the layering artifacts that are apparent in opacity shadow maps. Moreover, far fewer layers are necessary to generate high quality shadows.

The layering artifacts of the previous methods are especially significant in animated sequences or straight hair models. Figure 1 shows a comparison of our deep opacity maps algorithm to the previous methods. Layering artifacts in opacity shadow maps [KN01] are significant when 16 layers are used, and even with 128 layers, diagonal dark stripes are still visible on the left side of the hair model. Density clustering [MKBvR04] produces a good approximation to the overall shadows, but still suffers from visible artifacts around the specular region. However, our deep opacity maps method can produce an artifact-free image with fewer layers and it is significantly faster.

The next section describes the previous methods. The details of our deep opacity maps algorithm are explained in Section 3. We present the results of our method in Section 4, and we discuss the advantages and limitations in Section 5 before concluding in Section 6.

2. Related Work

Most shadow computation techniques developed for hair are based on *Shadow Maps* [Wil78]. In the first pass of shadow mapping, shadow casting objects are rendered from the light's point of view and depth values are stored in a depth map. While rendering the scene from the camera view in the second pass, to check if a point is in shadow, one first finds the corresponding pixel of the shadow map, and compares the depth of the point to the value in the depth map. The result of this comparison is a binary decision, so shadow maps cannot be used for transparent shadows.

Deep Shadow Maps [LV00] is a high quality method for offline rendering. Each pixel of a deep shadow map stores a 1D approximate transmittance function along the corresponding light direction. To compute the transmittance function, semi-transparent objects are rendered from the light's point of view and a list of fragments is stored for each pixel. The transmittance function defined by these fragments is then compressed into a piecewise linear function of approximate transmittance. The value of the transmittance function

starts decreasing after the depth value of the first fragment in the corresponding light direction. The shadow value at any point is found similar to shadow maps, but this time the depth value is used to compute the transmittance function at the corresponding pixel of the deep shadow map, which is then converted to a shadow value.

The *Alias-free Shadow Maps* [AL04] method is another offline technique that can generate high quality semi-transparent shadows. In this method, rasterization of the final image takes place before the shadow map generation to find the 3D positions corresponding to every pixel in the final image. Then, shadows at these points are computed from the light's point of view, handling one occluding piece of geometry at a time.

Opacity Shadow Maps [KN01] is essentially a simpler version of deep shadow maps that is designed for interactive hair rendering. It first computes a number of planes that slice the hair volume into layers (Figure 2a). These planes are perpendicular to the light direction and are identified by their distances from the light source (i.e. depth value). The opacity map is then computed by rendering the hair structure from the light's view. A separate rendering pass is performed for each slice by clipping the hair geometry against the separating planes. The hair density for each pixel of the opacity map is computed using additional blending on graphics hardware. The slices are rendered in order starting from the slice nearest to the light source, and the value of the previous slice is accumulated to the next one. Once all the layers are rendered, this opacity map can be used to find the transmittance from the occlusion value at any point using linear interpolation of the occlusion values at the neighboring slices. Depending on the number of layers used, the quality of opacity shadow maps can be much lower than deep shadow maps, since the interpolation of the opacities between layers generates layering artifacts on the hair. These artifacts remain visible unless a large number of layers are used.

Mertens et. al. proposed the *Density Clustering* approach [MKBvR04] to adjust the sizes and the positions of opacity layers separately for each pixel of the shadow map. It uses k-means clustering to compute the centers of the opacity layers. Then, each hair fragment is assigned to the opacity layer with the nearest center, and the standard deviation of the opacity layer times $\sqrt{3}$ is used as the size of the layer. Once the opacity layers are positioned, the hair geometry is rendered once again from the light's point of view and the opacity value of each layer is recorded. In general, density clustering generates better opacity layer distributions than opacity shadow maps, but it also introduces other complications and limitations. Density clustering's major limitation is that it cannot be extended to have a high number of layers due to the way that the layering is computed, and it is only suitable for a small number of clusters (the original paper [MK-BvR04] suggests 4 clusters). Moreover, k-means clustering is an iterative method and each iteration requires a separate

(a) Opacity Shadow Maps   (b) Deep Opacity Maps
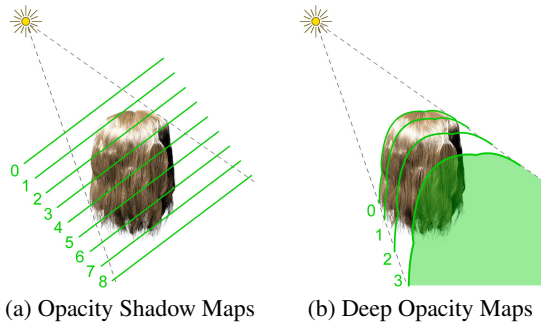
Figure 2: *Opacity shadow maps use regularly spaced planar layers. Our deep opacity maps use fewer layers, conforming to the shape of the hair model.*

pass that renders the whole hair geometry. The efficiency of the clustering depends on the initial choice of the opacity layer centers. Even if only a single pass of k-means clustering is performed, the density clustering method requires 4 passes to generate the shadow map. Finally, like opacity shadow maps, density clustering cannot guarantee that unshadowed direct illumination is captured correctly since the first opacity layer can begin before the first hair fragment.

The deep opacity maps method presented in this paper has advantages over these prior methods. It guarantees that the direct illumination of the surface hairs is calculated correctly. Unlike opacity shadow maps, opacity interpolation occurs within the hair volume, thus hiding possible layering artifacts. Unlike density clustering, deep opacity maps can easily use arbitrary numbers of layers (though usually 3 layers are sufficient). Comparing to both density clustering and opacity shadow maps, deep opacity maps achieve significantly higher frame rates for comparable quality.

Other previous methods include extensions of opacity shadow maps [KHS04], voxel based shadows [BMC05, ED06, GMT05], precomputed approaches [XLJP06], and physically based offline methods [ZSW04, MM06]. For a more complete presentation of the previous methods please refer to Ward et al. [WBK*07].

3. Deep Opacity Maps Algorithm

Our algorithm uses two passes to prepare the deep opacity map, and the final image is rendered in an additional pass, using this map to compute shadows.

The *first step* prepares the separators between the opacity layers. We render a depth map of the hair as seen from the light source. This gives us, for each pixel of the depth map, the depth z_0 at which the hair geometry begins. Starting from this depth value, we divide the hair volume within the pixel into K layers such that each layer lies from $z_0 + d_{k-1}$ to $z_0 + d_k$ where $d_0 = 0$, $d_{k-1} < d_k$ and $1 \leq k \leq K$. Note that the spacing $d_k - d_{k-1}$ (layer size) does not have to be constant. Even though we use the same d_k values for each pixel,

z_0 varies by pixel, so the separators between the layers take the shape of the hair structure (Figure 2). Note that the light source in this setup can be a point or a directional light.

The *second step* renders the opacity map using the depth map computed in the previous step. This requires rendering the hair only once and all computation occurs within the fragment shader. As each hair is rendered, we read the value of z_0 from the depth map and find the depth values of the layers on the fly. We assign the opacity contribution of the fragment to the layer that the fragment falls in and to all the other layers behind it. The total opacity of a layer at a pixel is the sum of all contributing fragments.
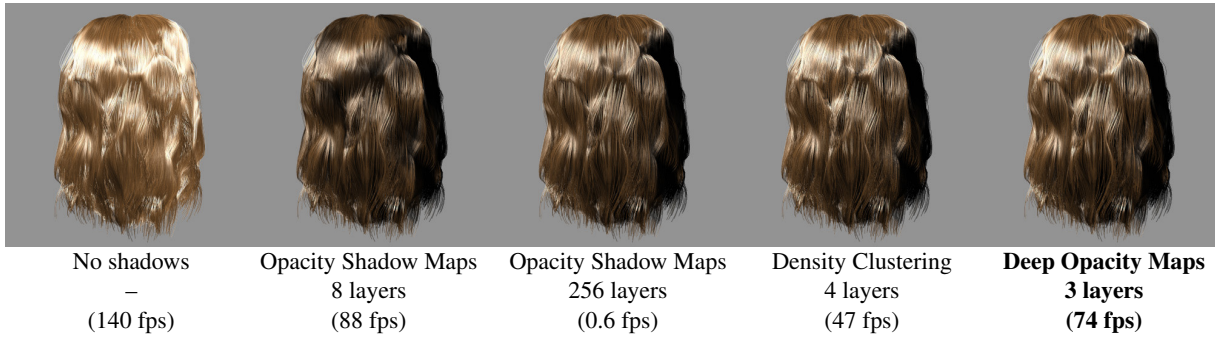
We represent the opacity map by associating each color channel with a different layer, and accumulate the opacities using additive blending on the graphics hardware. We reserve one color channel for the depth value, so that it is stored in the same texture with opacities. Therefore, using a single color value with four channels, we can represent three opacity layers. By enabling multiple draw buffers we can output multiple colors per pixel to represent more than three layers (n draw buffers allow $4n - 1$ layers). Obviously, using more than three layers will also require multiple texture lookups during final rendering.

One disadvantage of using a small number of layers with deep opacity maps is that it can be more difficult to ensure all points in the hair volume are assigned to a layer. In particular, points beyond the end of the last layer $z_0 + d_K$ do not correspond to any layer (shaded region in Figure 2b). We have a few options: ignore these points (thus, they will not cast shadows), include these points in the last layer (thus, they cast shadows on themselves), or ensure that the last layer lies beyond the hair volume by either increasing the layer sizes or the number of layers. While the last option might seem "ideal," it can lead to unnecessary extra layers that add little visual benefit at more computational cost, since the light intensity beyond a certain point in the hair volume is expected to vanish. We found that the second option, mapping these points onto the last layer, usually gave reasonable results.

Note that our algorithm uses the depth map only for computing the starting points of layers, not for a binary decision of in or out of shadow. Thus, unlike standard shadow mapping, deep opacity maps do not require high precision depth maps. For the scenes in our experiments, we found that using an 8-bit depth map visually performs the same as a 16-bit floating point depth map.

4. Results

To demonstrate the effectiveness of our approach we compare the results of our deep opacity maps algorithm to optimized implementations of opacity shadow maps and density clustering. We extended the implementation of opacity shadow maps with up to 16 layers to simultaneously generate the opacity layers in a single pass by using multiple draw

| No shadows – (140 fps) | Opacity Shadow Maps 8 layers (88 fps) | Opacity Shadow Maps 256 layers (0.6 fps) | Density Clustering 4 layers (47 fps) | **Deep Opacity Maps 3 layers (74 fps)** |

Figure 3: *Dark hair model with over one million line segments. The Opacity Shadow Maps method requires many layers to eliminate layering artifacts, Density Clustering approximates the shadows with some noise, while the Deep Opacity Maps method generates an artifact free image with higher frame rate.*

Opacity Shadow Maps - 256 layers

Density Clustering - 4 layers

Deep Opacity Maps - 3 layers
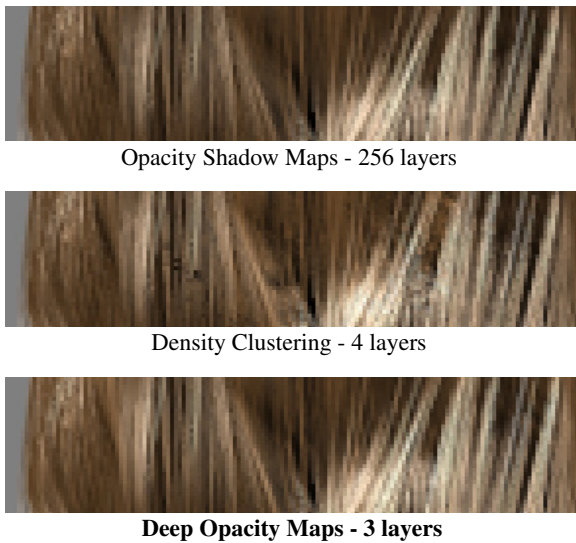
Figure 4: *Enlarged images from Figure 3 comparison.*

buffers, as opposed to the multi-pass implementation proposed in the original method [KN01]. We also introduced an additional pre-computation pass to density clustering, which computes the layer limits before the final image rendering, and achieved higher performance by optimizing the shadow lookup. All images presented in this paper were captured from our real-time hair rendering system using a standard PC with a 2.13GHz Core2 Duo processor and GeForce 8800 graphics card.

We used line drawing for rendering the hair models, and the Kajiya-Kay shading model [KK89] because of its simplicity. Antialiasing in the final image was handled on the graphics hardware using multi-sampling. We did not use antialiasing when generating opacity maps. To achieve a fake transparency effect in the final image we divided the set of hair strands into three disjoint subsets. Each one of these subsets is rendered separately with no transparency on a sep-
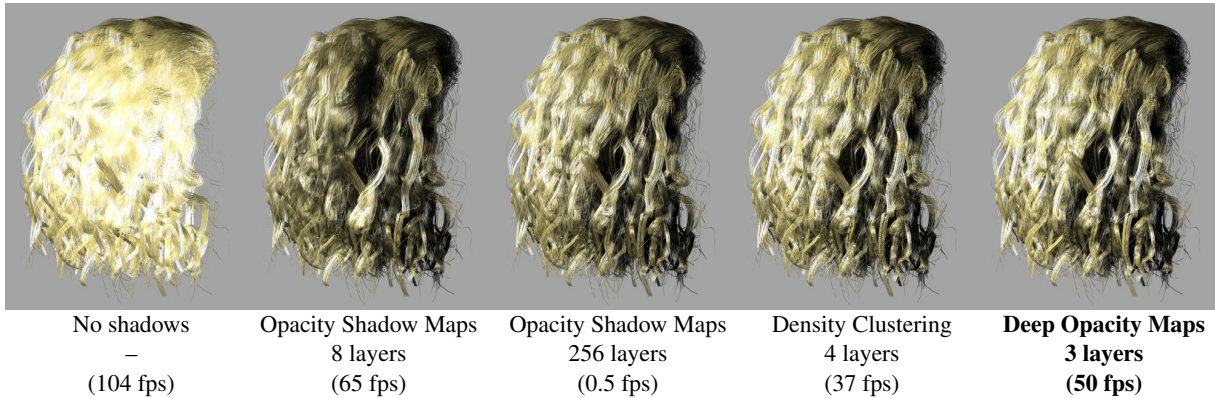
arate image. The resulting images are then blended to produce the final frame.

Figure 1 shows a straight hair model with 150 thousand line segments. The opacity shadow maps method with 16 layers produces severe artifacts of diagonal dark stripes that correspond to the beginning of each opacity layer. Though significantly reduced, these artifacts are still visible even when the number of layers is increased to 128. Density clustering, on the other hand, produces a good approximation to the overall illumination using 4 layers, however it suffers from a different kind of layering artifact visible around the specular region due to its inaccuracy. Our deep opacity maps technique produces an artifact free image with plausible shadow estimate using only 3 layers, and it is significantly faster than the other methods.

Figures 3 and 5 show two different complex hair styles with over one million and 1.5 million line segments respectively. On both of these models, the opacity shadow maps method with 8 layers produces dark stripes as interpolation artifacts between layers. When 256 layers are used with opacity shadow maps, layering artifacts visually disappear and the resulting shadows approach the correct values, but rendering one frame takes about two seconds. On the other hand, density clustering manages to produce a close approximation using only 4 layers. However, the inaccuracies of density clustering produce some noise in the illumination that are clearly visible in the enlarged images (Figures 4 and 6) and animated sequences. The deep opacity maps method manages to create an artifact free image with smooth illumination changes over the hair surface with significantly higher frame rates and less memory consumption.

Figure 7 demonstrates that deep opacity maps can be used in conjunction with traditional shadow maps. In this image of a hairy teapot, the shadow map handles the opaque shadows due to the teapot and the deep opacity map handles semi-transparent shadows due to the hair strands. Both the hair model and the teapot cast shadows onto each other as well as on the ground plane.

No shadows	Opacity Shadow Maps	Opacity Shadow Maps	Density Clustering	**Deep Opacity Maps**
–	8 layers	256 layers	4 layers	**3 layers**
(104 fps)	(65 fps)	(0.5 fps)	(37 fps)	**(50 fps)**

Figure 5: *Curly hair model with over 1.5 million line segments. Layering artifacts of Opacity Shadow Maps with 8 layers are apparent on the top of the hair model. They disappear with 256 layers with low frame rates. Density Clustering generates a noisy approximation, while the Deep Opacity Maps method generates an artifact free image with higher frame rate.*

Opacity Shadow Maps - 256 layers

Density Clustering - 4 layers

Deep Opacity Maps - 3 layers
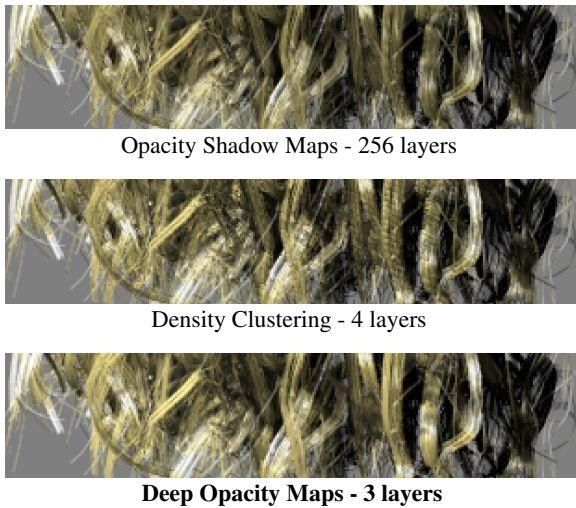
Figure 6: *Enlarged images from Figure 5 comparison.*

Figure 7: *A hairy teapot rendered using deep opacity maps with traditional shadow maps for opaque shadows.*

Figure 8 shows clustered strands rendered using deep opacity maps. In Figure 8a, 3 layers are not enough to cover the whole illuminated volume, and dark regions in appear where the strands beyond the last layer cast shadows onto themselves. By increasing the number of layers, as in Figure 8b, incorrect self-shadowing can be eliminated. This can also be achieved by increasing the layer sizes as in Figure 8c; however, this also reduces the shadow accuracy. As can be seen from these images, deep opacity maps can generate high quality shadows with non-uniform hair models.

5. Discussion

The main advantage of our method is that by shaping the opacity layers, we capture direct illumination correctly while eliminating visual layering artifacts by moving interpolation between layers to within the hair volume. This lets us hide possible inaccuracies while also allowing high quality results with fewer layers. Unlike density clustering, which tries to approximate the whole transmittance function, we concentrate the accuracy on the beginning of the transmittance decay (where the shadow begins). By doing so, we aim to be more accurate around the illuminated surface of the hair volume—the part of the hair that is most likely to appear in the final image and where inaccuracies would be most noticeable.

Since we require very few layers, all information can be stored in a small number of textures (a single texture for 3 layers). This makes our algorithm memory efficient and also reduces the load on the fragment shader.

The extreme simplicity of our approach allows us to prepare the opacity map with only 2 render passes, and only one of these passes uses blending. Opacity shadow maps of just a few layers can be generated in only a single pass, however visual plausibility requires many more layers.

(a) 3 layers   (b) 7 layers   (c) 3 (larger) layers
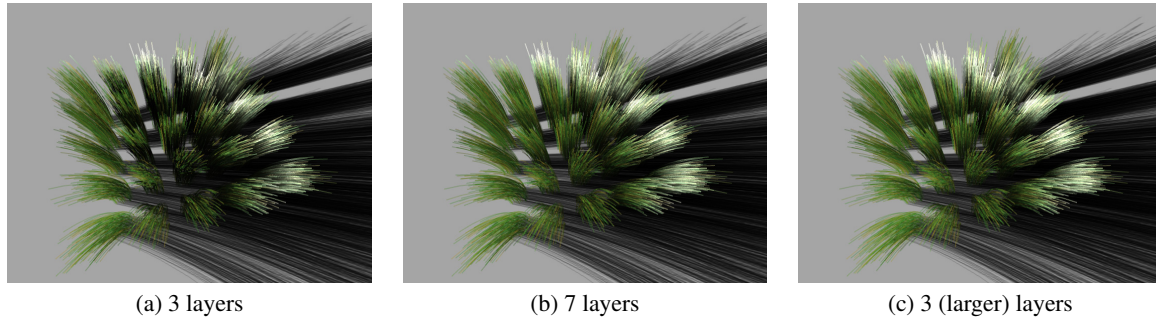
Figure 8: *Clustered strands with deep opacity maps using different number of layers and different layer sizes.*

For the examples in this paper we used linearly increasing layer sizes with deep opacity maps. This choice of layer distribution provides high accuracy around bright regions where the transmittance begins to decay, while keeping other layers large enough to cover the illuminated part of the hair volume with a few number of layers. Varying layer sizes can also be used for opacity shadow maps, but this would not provide any visual enhancement, since there is no heuristic that can reduce the layering artifacts by changing layer sizes without increasing the number of layers.

In our implementation we observed minor flickering due to aliased line drawing we used while rendering depth and opacity maps. Applying a smoothing filter to the depth and opacity maps reduced this problem, but did not completely remove it. In our experiments we found that using multi-sampling for shadow computations, a standard technique used for smoothing shadow maps, produced better results with additional computation cost.

6. Conclusion

We have introduced the deep opacity maps method, which uses a depth map to achieve per-pixel layering of the opacity map for real-time computation of semi-transparent shadows. We compared both quality and performance of our method to the previous real-time/interactive semi-transparent shadowing techniques. Our results show that deep opacity maps are fast and can generate high quality shadows with minimal memory consumption. Our algorithm does not have any restrictions on the hair model or hair data structure. Since it does not need any pre-computation, it can be used when rendering animated dynamic hair or any other semi-transparent object that can be represented by simple primitives.

References

[AL04] AILA T., LAINE S.: Alias-free shadow maps. In *Eurographics Symp. on Rendering* (2004), pp. 161–166.

[BMC05] BERTAILS F., MÉNIER C., CANI M.-P.: A practical self-shadowing algorithm for interactive hair animation. In *Proc. Graphics Interface* (2005), pp. 71–78.

[ED06] EISEMANN E., DÉCORET X.: Fast scene voxelization and applications. In *Symposium on Interactive 3D Graphics and Games* (2006), pp. 71–78.

[GMT05] GUPTA R., MAGNENAT-THALMANN N.: Scattering-based interactive hair rendering. In *Comp. Aided Design and Comp. Graphics* (2005), pp. 489–496.

[KHS04] KOSTER M., HABER J., SEIDEL H.-P.: Real-time rendering of human hair using programmable graphics hardware. In *Proceedings of the Computer Graphics International (CGI'04)* (2004), pp. 248–256.

[KK89] KAJIYA J. T., KAY T. L.: Rendering fur with three dimensional textures. In *Proceedings of SIGGRAPH 1989* (1989), pp. 271–280.

[KN01] KIM T.-Y., NEUMANN U.: Opacity shadow maps. In *12th Eurographics Workshop on Rendering Techniques* (2001), pp. 177–182.

[LV00] LOKOVIC T., VEACH E.: Deep shadow maps. In *Proceedings of SIGGRAPH 2000* (2000), pp. 385–392.

[MKBvR04] MERTENS T., KAUTZ J., BEKAERT P., VAN REETH F.: A self-shadow algorithm for dynamic hair using clustered densities. In *Proceedings of Eurographics Symposium on Rendering 2004* (2004), pp. 173–178.

[MM06] MOON J. T., MARSCHNER S. R.: Simulating multiple scattering in hair using a photon mapping approach. In *Proceedings of SIGGRAPH 2006* (2006), pp. 1067–1074.

[WBK*07] WARD K., BERTAILS F., KIM T.-Y., MARSCHNER S. R., CANI M.-P., LIN M.: A survey on hair modeling: Styling, simulation, and rendering. *IEEE TVCG 13*, 2 (Mar-Apr 2007), 213–34.

[Wil78] WILLIAMS L.: Casting curved shadows on curved surfaces. In *SIGGRAPH '78* (1978), pp. 270–274.

[XLJP06] XU S., LAU F. C., JIANG H., PAN Y.: A novel method for fast and high-quality rendering of hair. In *Proc. EGSR'06* (2006), pp. 331–341.

[ZSW04] ZINKE A., SOBOTTKA G., WEBER A.: Photo-realistic rendering of blond hair. In *Vision, Modeling, and Visualization 2004* (2004), pp. 191–198.