Computational Visual Media
https://doi.org/10.1007/s41095-0xx-xxxx-x

Research Article

Supplementary Note II - Temporal Vectorized Visibility for Direct Illumination of Animated Models

Zhenni Wang[1], Tze Yui Ho[1], Yi Xiao[2](✉) and Chi Sing Leung[1]

Abstract abstract

Keywords keywords

1 Convolution Soft Shadows

We faithfully implement the convolution soft shadows (CSS) algorithm by following the description in [1]. The CSS algorithm was proposed in 2008. The GPU we now used to implement CSS, i.e. NVIDIA GeForce RTX 2080 Ti, has higher processing power than the GPU used in [1], and the higher processing power influences the rendering speed of CSS. We first look into the influence of the powerful GPU on the rendering speed.

Given a more powerful GPU, the rendering speed of CSS is naturally faster than that in [1]. Table 1 shows the processing time of our CSS implementation measured with 27 area lights. With shadow maps (SM) resolution set to be 1024 × 1024, the frame rate of CSS is 70.8 fps, while the corresponding frame rate reported in [1] could be 3-4 fps approximately (see *Tabel 1* in [1], 5 fps for 20 area lights and SM resolution set to be 1K^2). We achieve a 20x times as fast rendering speed with the more powerful GPU. Note that the 20x times as fast rendering speed is not applicable to all settings of CSS. For instance, with SM resolution set to be 256 × 256, the frame rate of CSS is 89.6 fps, while the corresponding frame rate reported in [1] is 25.4 fps (see *Table 1* and *Fig. 11(c)* in [1]). We just have a 3x times as fast rendering speed. The different rendering speed improvements may be caused by the hardware architecture of GPU.

Using mipmapping as the filtering tool will introduce aliasing to the rendering images of CSS, which is also mentioned in [1]. This aliasing is mainly caused by not enough samples used in the mipmap lookup, and it is noticeable in the rendering images if we just use one mipmap sample. To obtain the similar visual quality reported in [1], we need to perform multiple sampling for mipmap lookup.

1 City University of Hong Kong, Hong Kong, China. E-mail: Z. Wang, zhenni126@126.com; T.Y. Ho, ma_hty@hotmail.com; C.S. Leung, eeleungc@cityu.edu.hk.

2 Hunan University, Changsha, China. E-mail: yixiao1984@gmail.com.

Table 1 The frame rate of CSS. The frame rate is measured with the fox model in Fig. 1 and is reported in fps. The environment map is approximated by using 27 area lights. For each area light, we prepare a depth map and four basis maps. These shadow maps are prefiltered by using mipmapping.

SM resolution	Number of mipmap samples			
	1	4	9	16
128 × 128	99.5	85.8	72.3	58.4
256 × 256	89.6	79.2	67.5	55.4
512 × 512	77.8	71.8	63.1	51.0
1024 × 1024	70.8	64.6	55.0	47.6

Fig. 1 shows a series of the rendering images of CSS, and we render these images by varying the number of mipmap samples. As shown in Fig. 1(a), just using one sample for the mipmap lookup, the shadow boundaries suffer from severe aliasing. Increasing the number of mipmap samples, the aliasing is gradually remedied (see Fig. 1(b)-(d)). By using 16 mipmap samples, the aliasing in the shadow boundaries is less noticeable, and the rendering image in Fig. 1(d) has a similar visual quality to *Fig. 11(c)* in [1]. Note that due to the nature of mipmapping, there is still some aliasing remained in the shadow boundaries even with multiple mipmap samples. Fortunately, this aliasing is visually almost indistinguishable w.r.t. the whole rendering image.

Using multiple mipmap samples improves the visual quality though, it decreases the rendering speed. The frame rate of CSS with 16 mipmap samples is 47.6 fps, which is slower than that of 1 mipmap sample by 23.2 fps. However, compared to the rendering speed in [1], using 16 mipmap samples still provides a 10x times as fast rendering speed.

We also provide the rendering images of CSS with lower resolution SM in Fig. 1(e)-(p). The behavior of mipmapping aliasing in these rendering images is similar to that in Fig. 1(a)-(d). On the other hand, even we remedy the aliasing in these rendering images using multiple mipmap samples, their shadow quality still suffers from the lower resolution SM (see Fig. 1(p)).

Given the above discussion, we choose to implement CSS for comparison by using SM with a resolution of 1024×1024 and varying the mipmap samples to adjust the tradeoff between the visual quality and the rendering speed. Specifically, we use 1 mipmap sample for the simlar rendering speed comparison and 16 mipmap samples for the similar visual quality comparison. Fig. 2 shows the corresponding rendering images and the FLIP error maps [2] of CSS and our algorithm.

With the similar frame rate around 70 fps, our algorithm provides a smooth rendering result which preserves the fidelity of shadows (see Fig. 2(a)), while CSS, as just using 1 mipmap sample, exhibits noticeable aliasing in the shadow boundaries (see Fig. 2(b)). The aliasing is significantly remedied by using 16 mipmap samples, and we can see clear subtle shadow transitions in the rendering image (see Fig. 2(c)). As mentioned before, using 16 mipmap samples also decreases the rendering speed. The frame rate of Fig. 2(c) is 47.6 fps. With the similar visual quality, our algorithm is 1.5 times as fast as CSS.

It seems that the rendering image of CSS is visually similar to the ground truth if just comparing them side by side (see Fig. 2(c) vs. (d)). However, CSS is just using a small number of area lights (i.e. 27 in our implementation) to approximate the environment map with 400k light sources. and its rendering result will not approach the actual 400k light sources integration results due to the approximation. As shown in Fig. 2(g), the shadows of CSS deviate from those in the ground truth significantly. On the other hand, our algorithm indeed considers 400k light sources in the radiance evaluation, therefore, the rendering result of our algorithm has less difference w.r.t the ground truth (see Fig. 2(e)).

The following is some additional information of our CSS implementation. The environment map is approximated by 27 area lights (see Fig. 2(h)), and the area lights are generated by using the area light generation method in [1]. For each area light, we generate a depth map and four basis maps, and these maps are pre-filtered by using mipmapping. Our CSS implementation is written in Vulkan, and the performance is evaluated by using a PC equipped with a NVIDIA GeForce RTX 2080 Ti GPU and an Intel(R) Core(TM) i5-8400 CPU.

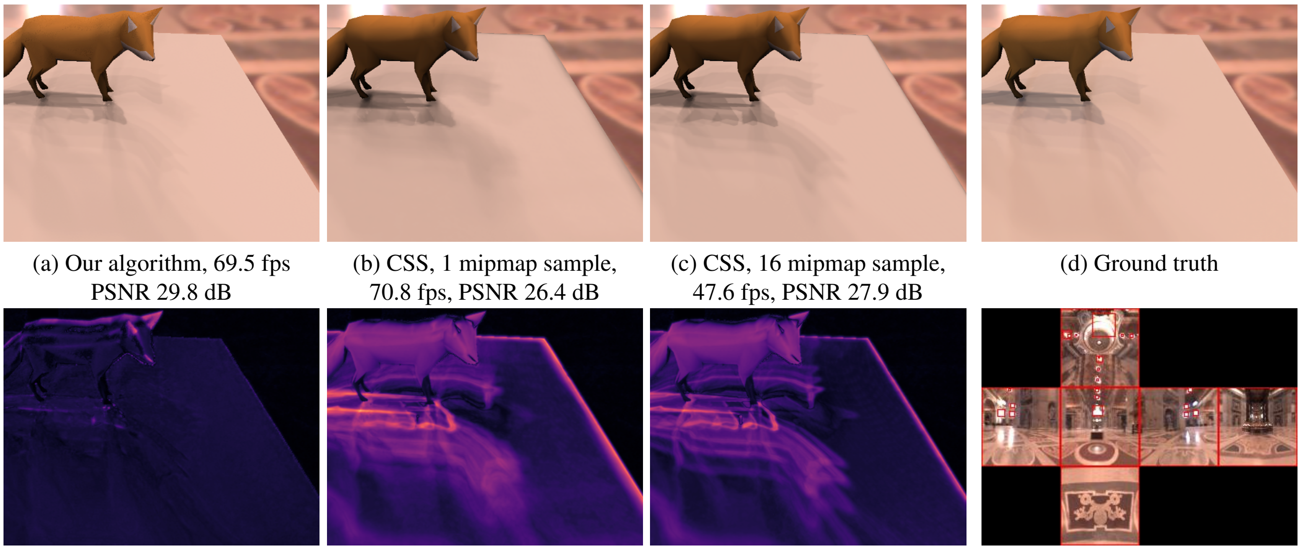
References

[1] Annen T, Dong Z, Mertens T, Bekaert P, Seidel HP, Kautz J. Real-time, all-frequency shadows in dynamic scenes. *ACM transactions on graphics (TOG)*, 2008, 27(3): 1–8.

[2] Andersson P, Nilsson J, Akenine Möller T, Oskarsson M, Åström K, Fairchild MD. ꟻLIP: A Difference Evaluator for Alternating Images. *Proc. ACM Comput. Graph. Interact. Tech.*, 2020, 3(2).

(a) SM 1024 × 1024, N=1, 70.8 fps
(b) SM 1024 × 1024, N=4, 64.6 fps
(c) SM 1024 × 1024, N=9 55.0 fps
(d) SM 1024 × 1024, N=16 47.6 fps
(e) SM 512 × 512, N=1, 77.8 fps
(f) SM 512 × 512, N=4, 71.8 fps
(g) SM 512 × 512, N=9 63.1 fps
(h) SM 512 × 512, N=16 51.0 fps
(i) SM 256 × 256, N=1, 89.6 fps
(j) SM 256 × 256, N=4, 79.2 fps
(k) SM 256 × 256, N=9 67.5 fps
(l) SM 256 × 256, N=16 55.4 fps
(m) SM 128 × 128, N=1, 99.5 fps
(n) SM 128 × 128, N=4, 85.8 fps
(o) SM 128 × 128, N=9 72.3 fps
(p) SM 128 × 128, N=16 58.4 fps

Fig. 1 The direct illumination rendering images of CSS [1] with different number of mipmap samples N. Just using 1 mipmap sample for the mipmap lookup, the aliasing is noticeable at the shadow boundaries (a). We can remedy the aliasing by increasing N (see (b)-(d)). The rendering image of 16 mipmap samples has less aliasing in the shadow boundaries, and the rendering quality is similar to that reported in [1]. Given SM with lower resolution, the aliasing is also remedied by using multiple mipmap samples, but the shadow quality still suffers from the lower resolution SM (see (m)-(p)).



(a) Our algorithm, 69.5 fps PSNR 29.8 dB

(b) CSS, 1 mipmap sample, 70.8 fps, PSNR 26.4 dB

(c) CSS, 16 mipmap sample, 47.6 fps, PSNR 27.9 dB

(d) Ground truth

(e) FLIP error map of (a), mean 0.09

(f) FLIP error map of (b), mean 0.19

(g) FLIP error map of (c), mean 0.15

(h) Env. map with fitted area lights

Fig. 2 The direct illumination rendering images of CSS [1] and our algorithm. With the similar rendering speed, the rendering image of our algorithm (a) preserves the fidelity of shadows, while the rendering image of CSS (b) has noticeable aliasing in the shadow boundaries. The aliasing is caused by mipmapping, and we can remedy it by using multiple mipmap samples. (c) is the rendering image of CSS with 16 mipmap samples. The aliasing is less noticeable, and we can see clear subtle shadow transitions. With the similar visual quality, our algorithm (a) is 1.5 times as fast as CSS (c). On the other hand, due to the environment map approximation, the rendering image of CSS (c), visually appealing though, indeed deviates from the ground truth (d), and this deviation is more noticeable in its FLIP error map (g). (h) is the environment map with the fitted area lights in red.