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

Research Article

Supplementary Note I - 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 More direct illumination rendering images

This supplementary note documents some additional direct illumination rendering images of our algorithm and ReSTIR [1] we prepared for the paper. These rendering images, though did not get the chance to be presented in the paper, might still be interesting for the readers.

The rendering images in this supplementary note are obtained with similar per frame processing time. The rendering images of our algorithm are generated by updating temporal vectorized visibility on a per group basis, where the plane updates every 2 frames, and the animated model updates every 8 frames. The rendering images of ReSTIR are generated by using the suggested unbiased setting. We vary the number of reservoirs N of each pixel to adjust the tradeoff between the rendering speed and the rendering quality. The ground truth images are generated by using multiple importance sampling (MIS) [2], and they are generated by accumulating the samples until the images converge. This supplementary note contains a sequence of 3D models with increasing number of triangles. The number of triangles increases deliberately until our algorithm is no longer efficient. Table 1 shows the general information of these 3D models.

As mentioned in the paper, the complexity of our algorithm is scene complexity dependent. Inevitably, it will be less efficient to ReSTIR given complex enough scenes. In these cases, ReSTIR can use a larger N to maintain the similar per frame processing time comparison, which also improves its visual quality.

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 general information of 3D models.

Model	Number of triangle	Number of vertex	Number of edge
Flamingo	742	394	1.1k
Parrot	826	436	1.2k
Robot	3.4k	1.9k	5.3k
Succulent	5.2k	2.6k	7.8k
Smooth Lou	11k	5.6k	16k
Junkyard robot	47k	25k	73k
Toy car	109k	57k	166k

Fig. 1 shows the direct illumination rendering images of our algorithm and ReSTIR given complex models. Given the model smooth lou with 11k triangles, the per frame processing time of our algorithm is 88.2 ms (i.e. 11.3 fps). With the similar per frame processing time, ReSTIR uses 8 reservoirs for each pixel, and there is less pepper noise in its rendering image (see *row 1*, *column 2* in Fig. 1). In terms of PSNR, the PSNR of our algorithm is 31.6 dB, and the PSNR of ReSTIR is 25.5 dB. The PSNR difference between our algorithm and ReSTIR decreases to 6 dB.

As a ray tracing algorithm, ReSTIR is flexible with complex models. It can vary the number of reservoirs of each pixel to adjust the tradeoff between the rendering speed and the rendering quality. Fig. 2 shows a series of rendering images of ReSTIR given complex models. Given the model junkyard robot with 47k triangles and the model toy car with 109k triangles, ReSTIR with the AI denoiser [3] can just use $N = 1$ to render the visually appealing images in real time. Given such complex models, our algorithm is no longer efficient and has to render images in seconds. As shown in Fig. 1, the per frame processing time of our algorithm is 3.3s for the model junkyard robot (*row 2*, *column 1*), and it even increases to 8.3s for the model toy car (*row 3*, *column 1*). With the similar processing time, the rendering images of ReSTIR are already close to converge and have similar visual quality to ours.

Shown in Fig. 3 are the direct illumination rendering images of our algorithm and ReSTIR given some simple 3D

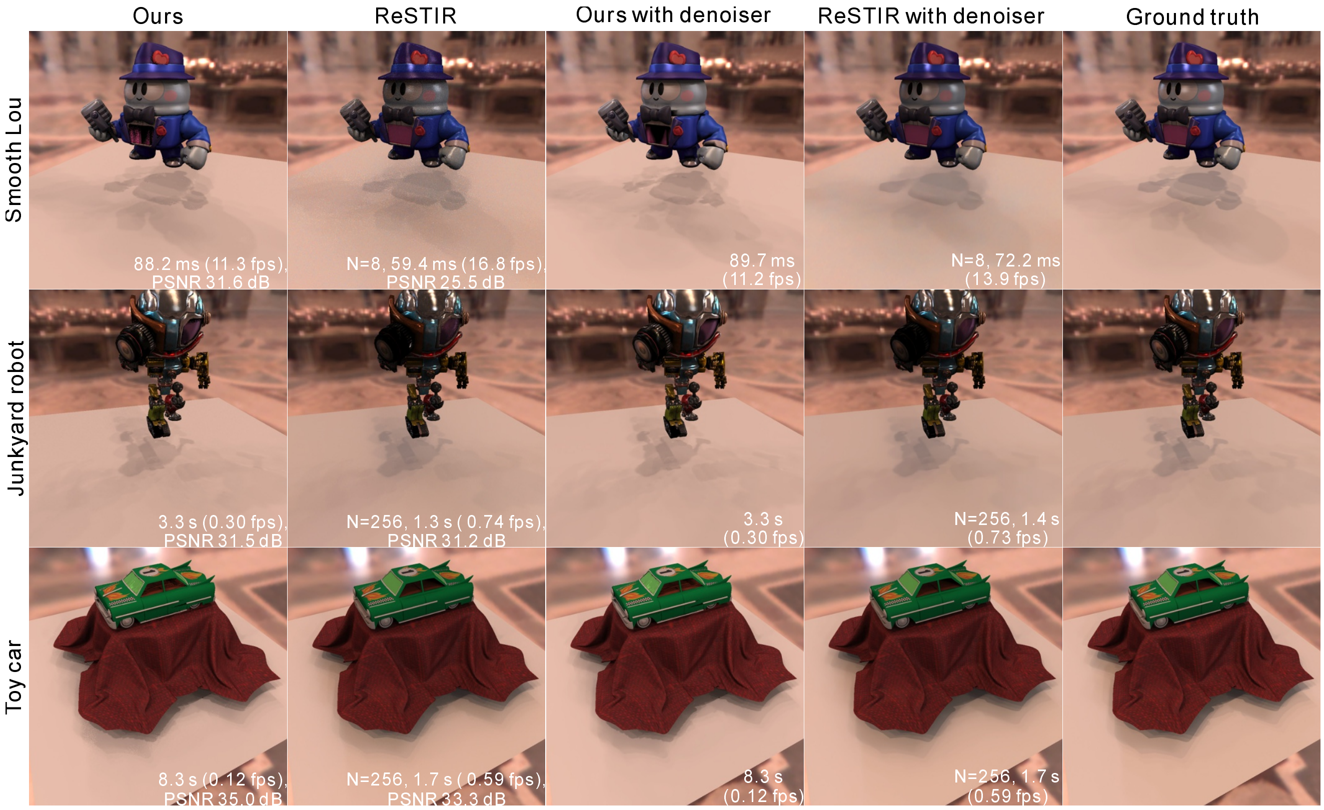

Fig. 1 The direct illumination rendering images of our algorithm and ReSTIR [1] given complex 3D models. For the model junkyard robot and the model toy car, our algorithm needs seconds (3.3 s and 8.3 s, respectively) to render the images. With the similar per frame processing time, the rendering images of ReSTIR are already close to converge and have the similar visual quality to ours.

models. Similar to the observation in the paper, with the similar per frame processing time, the rendering images of ReSTIR shown in the second column have noticeable noise, while the rendering images of our algorithm shown in the first column practically have no noise and capture the subtle shadow transitions to an extend similar to the ground truth in the last column. In terms of PSNR, with the similar processing time, our algorithm provides a visual quality improvement around 10 dB w.r.t. ReSTIR. Shown in the fourth column is the rendering images of ReSTIR with the AI denoiser [3]. By applying the denoiser, the noise is removed, while the subtle shadow transitions are also blurred, e.g. the shadows of the parrot wings (*row 2*, *column 4*).

2 Biased ReSTIR

The rendering images of ReSTIR in Fig. 1, Fig. 2 and Fig. 3 are rendered with its unbiased solution. We also prepare the rendering images of the biased ReSTIR (see Fig. 4), and these images are rendered by using the suggested biased setting.

Compared to the unbiased ReSTIR (see Fig. 4(b), rendered at 16.3 ms), the biased ReSTIR has less noise in its rendering image (see Fig. 4(c), rendered with 19.3 ms). However, its rendering image is unnecessarily darker and deviates from the ground truth (see Fig. 4(l)) significantly.

As a biased solution, the biased ReSTIR will never converge to the ground truth. With the increase of N, the PSNR of the biased ReSTIR increases just because of having less noise (see Fig. 5(a) and Fig. 4(k)). Further increasing N, its PSNR remains to be similar and is the smallest among the comparison algorithms. The FLIP [4] curve of the biased ReSTIR in Fig. 5(b) demonstrates the same behavior.

References

[1] Bitterli B, Wyman C, Pharr M, Shirley P, Lefohn A, Jarosz W. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 2020, 39(4).

[2] Veach E, Guibas LJ. Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, 419–428.

[3] Chaitanya CRA, Kaplanyan AS, Schied C, Salvi M, Lefohn A, Nowrouzezahrai D, Aila T. Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. *ACM Transactions on Graphics (TOG)*, 2017, 36(4): 1–12.

[4] Andersson P, Nilsson J, Akenine Möller T, Oskarsson M, Åström

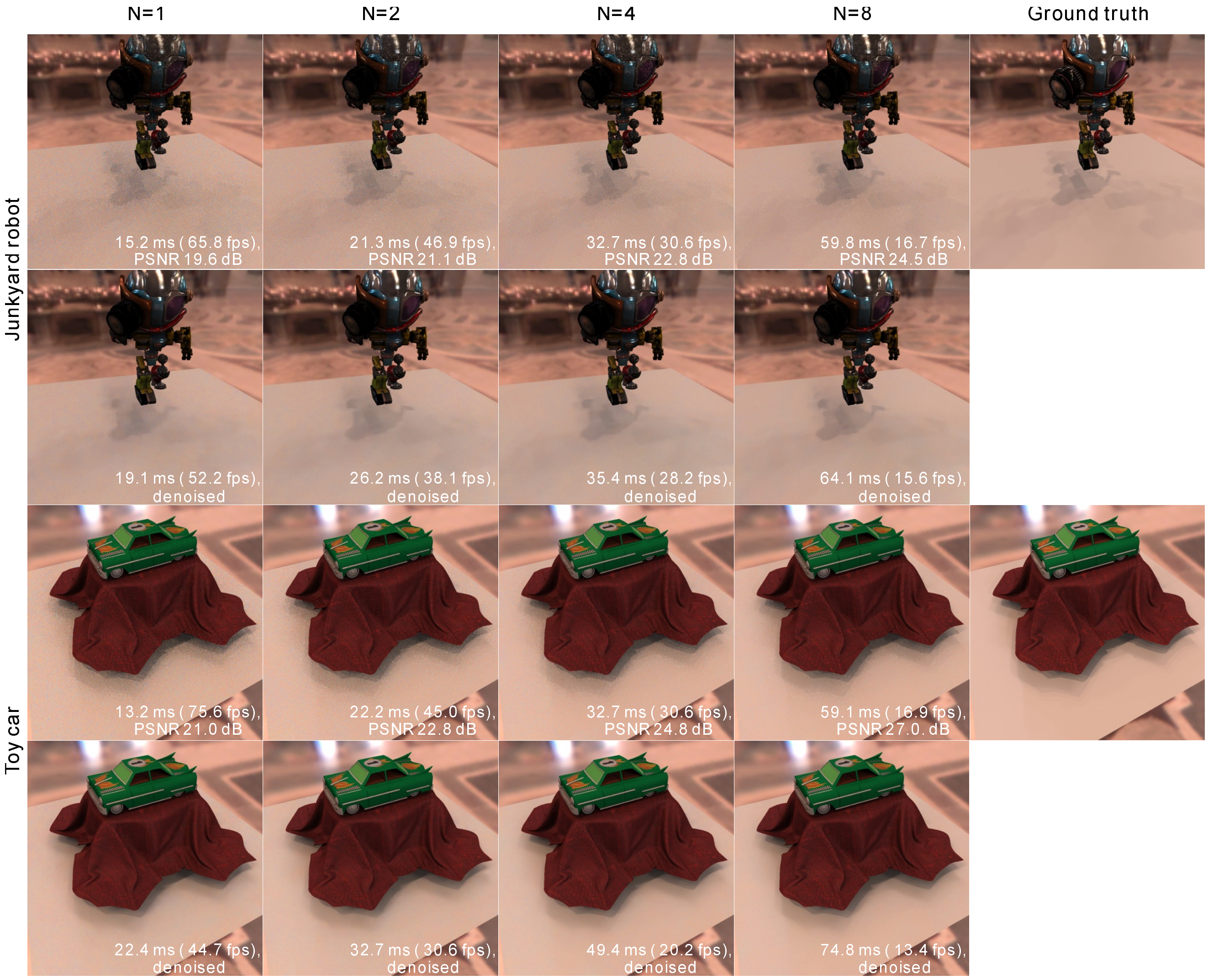

Fig. 2 A series of direct illumination rendering images of ReSTIR [1] given complex 3D models. As a ray tracing algorithm, ReSTIR is flexible with the complex 3D models. It can modify N to adjust the tradeoff between the rendering speed and the rendering quality. For the model junkyard robot and the model toy car, ReSTIR with the AI denoiser [3] can just use $N = 1$ to render the visually appealing images in real time.

K, Fairchild MD. ꟻLIP: A Difference Evaluator for Alternating Images. *Proc. ACM Comput. Graph. Interact. Tech.*, 2020, 3(2).

Fig. 3 The direct illumination rendering images of our algorithm and ReSTIR [1] given simple 3D models. For each model, with the similar per frame processing time, the rendering images of ReSTIR in the second column have noticeable pepper noise, while the rendering images of our algorithm in the first column practically have no noise and capture the subtle shadow transitions to an extent equivalent to the ground truth in the last column. In terms of PSNR, our algorithm provides a visual quality improvement around 10 dB w.r.t. ReSTIR. Applying the AI denoiser [3] removes the noise, but the subtle shadow transitions are also blurred in the denoised images of ReSTIR in the fourth column.

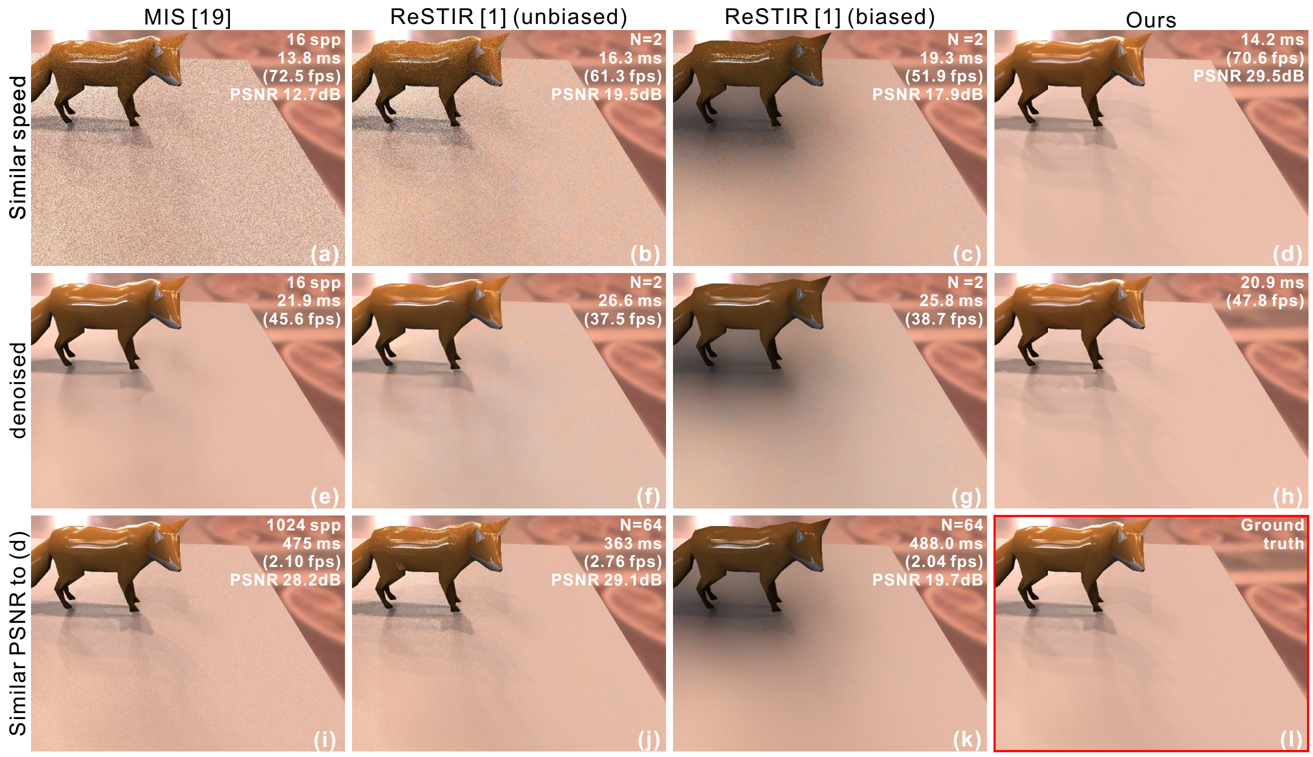

Fig. 4 The direct illumination rendering images of MIS [2], ReSTIR [1] and our algorithm.

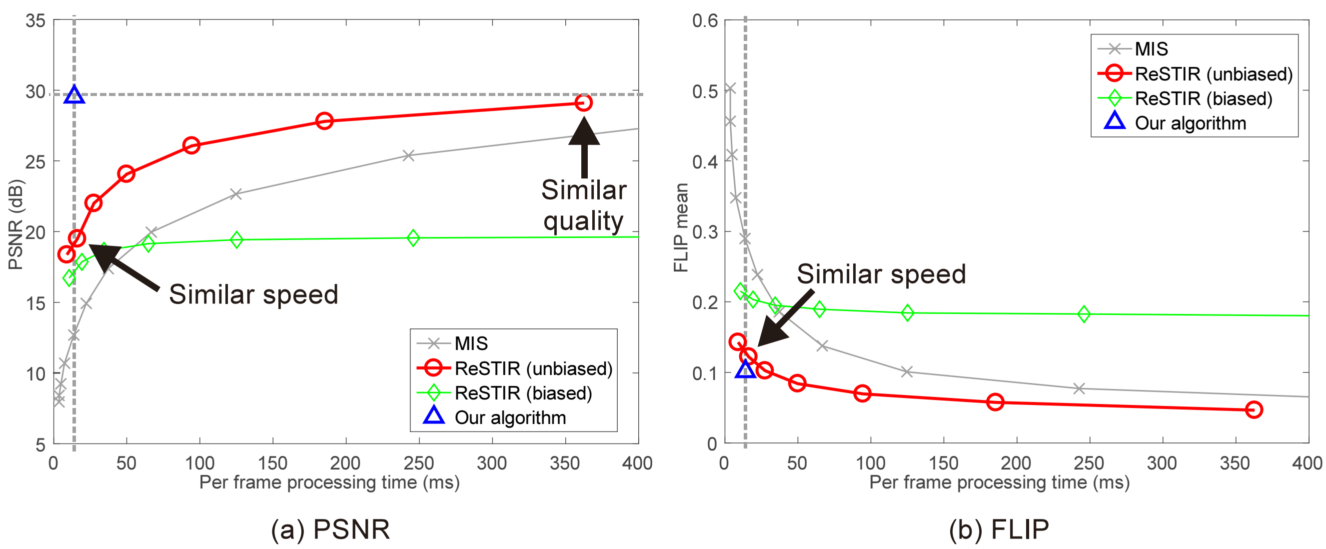

Fig. 5 The PSNR and FLIP curves of MIS [2], ReSTIR [1] and our algorithm.