

Hybrid DoF: Ray-Traced and Post-Processed Hybrid Depth of Field Effect for Real-Time Rendering

Tan Yu Wei

National University of Singapore
yuwei@u.nus.edu

Nathan Biette

National University of Singapore
nathan.biette@u.nus.edu

Nicholas Chua

National University of Singapore
nicholaschuayunzhi@u.nus.edu

Anand Bhojan

National University of Singapore
banand@comp.nus.edu.sg



Figure 1: From left to right: original scene, adapted post-process [Jimenez 2014] with foreground overblur, UE4 post-process [Abadie 2018], hybrid (143 fps) and ray-traced DoF on THE MODERN LIVING ROOM (CC BY) with a GeForce RTX 2080 Ti

ABSTRACT

Depth of Field (DoF) in games is usually achieved as a post-process effect by blurring pixels in the sharp rasterized image based on the defined focus plane. This paper describes a novel real-time DoF technique that uses ray tracing with image filtering to achieve more accurate partial occlusion semi-transparencies on edges of blurry foreground geometry. This hybrid rendering technique leverages ray tracing hardware acceleration as well as spatio-temporal reconstruction techniques to achieve interactive frame rates.

CCS CONCEPTS

• **Computing methodologies** → **Rendering; Ray tracing**; • **Applied computing** → **Computer games**.

KEYWORDS

real-time, depth of field, ray tracing, post-processing, hybrid rendering, games

ACM Reference Format:

Tan Yu Wei, Nicholas Chua, Nathan Biette, and Anand Bhojan. 2020. Hybrid DoF: Ray-Traced and Post-Processed Hybrid Depth of Field Effect for Real-Time Rendering. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Posters (SIGGRAPH '20 Posters)*, August 17, 2020. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3388770.3407426>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGGRAPH '20 Posters, August 17, 2020, Virtual Event, USA

© 2020 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-7973-1/20/08.

<https://doi.org/10.1145/3388770.3407426>

1 INTRODUCTION

Under partial occlusion in Depth of Field (DoF), background information is revealed through the semi-transparent silhouettes of blurry foreground geometry. Partial occlusion is inaccurate with traditional post-processing as the rasterized image does not store any information behind foreground objects. We can query the scene for background intersections with ray tracing but at poorer performance. Hence, we propose a DoF effect based on hybrid rendering, which combines rasterization and ray tracing techniques for better visual quality while maintaining interactive frame rates.

2 DESIGN

A G-Buffer is first produced in deferred shading with a sharp rasterized image that undergoes post-process filtering adapted from Jimenez [2014]. Selected areas of the scene with higher rates of post-processing inaccuracy are then chosen to undergo distributed ray tracing with spatio-temporal reconstruction. The post-process and ray trace colours are finally composited together. We build our technique on the thin lens model by Potmesil and Chakravarty [1982], and define our scene such that points in front of the focus plane are in the *near* field and points behind are in the *far* field.

2.1 Ray Trace

We shoot a variable number of rays into the scene through an adaptive ray mask. Employing a selective rendering approach like in adaptive frameless rendering, we aim to shoot more rays at edges to create clean semi-transparencies but less at regions with fewer details such as plain surfaces. Like Canny Edge Detection, we apply a Gaussian filter on the G-Buffer first to reduce noise and jaggies along diagonal edges. The filtered G-Buffer then passes through

a Sobel kernel to approximate the gradient associated with each pixel based on the differences in surface normal and depth with its surrounding pixels, detecting geometry edges for the ray mask. Accounting for temporal variation to reduce noise, we also shoot more rays at areas of high variance in luminance, including newly ray-traced regions. Hence, the ray mask is complemented with a temporally-accumulated variance estimate [Schied et al. 2017] from the ray trace output of previous frames.

We then shoot rays, taking multiple samples in a circle to produce random ray origins within the lens' area. At the edge of an out-of-focus foreground object, rays originating from the lens will either hit the object or objects behind it, yielding a semi-transparent silhouette. We sort the hits into the *near* and *far* fields. However, to have a smooth colour transition, we split the contribution of pixel colour to each layer based on its Z-distance from the separating focus plane. A hit ratio is also stored, which is the number of rays contributing to the *near* colour divided by the total rays shot.

2.2 Reconstruction

To increase our sample count, we temporally-accumulate *near* and *far* ray trace colours over time and use an exponential moving average to blend between history and current frames like in TAA. However, to account for movement, we leverage per-pixel depth and motion information for reprojection like Lehtinen et al. [2011].

As such, we use screen space motion vectors to reproject *near* pixels. As for the *far* field, we require an approximation of the *far* world position of our pixel. We first attempt to compute the average world position of our pixel from *far* ray trace hits. Under low ray counts, in the event that there is no *far* hit, we obtain the average *far* world position of neighbouring pixels instead. Then, we use this world position to calculate the previous screen space position of the pixel, approximating reprojection for occluded objects appearing within semi-transparent regions of foreground geometry. However, to prevent the ghosting (or smearing) of these regions, we use the latest hit ratio to compose the *near* and *far* colours during motion, but normalize the final colour based on the accumulated hit ratio since reprojection is different for the *near* and *far* fields. Otherwise, we risk having varying colour intensities in our merged result.

We also blend the new average *near* and *far* Circle of Confusion (CoC) sizes based on the accumulated hit ratio, so as to get an approximate CoC size for the current frame to scale our circular gathering kernel and determine the level-of-detail for sampling in spatial reconstruction. Similar to our post-processing approach, we gather the colour contributions of neighbouring pixels, and samples with CoC radius smaller than their distance to the pixel are rejected. Finally, the gathered colour is weighted based on clamped variance estimates to avoid over-blurring in converged regions.

2.3 Composite

For the final image, we apply the ray trace, post-process and sharp rasterized colours onto pixels based on their Z-values in relation to the focus zone, where the CoC size of pixels is less than $\sqrt{2}$ and hence the unblurred rasterized colour is applied. Outside the focus zone, for *near* objects in general, we apply the ray trace colour to form accurate semi-transparent silhouettes. However, for bright specular bokeh shapes, we favour the post-process over the ray

trace colour based on the intensity of the specular component of the post-process colour, so as to minimize noise from the ray trace.

For *far* geometry out of the focus zone, we adaptively blend the ray trace and post-process colours using the hit ratio. At high hit ratios, we favour the post-process colour as fewer rays hit the *far* field. On the other hand, low hit ratios indicate that the number of hits in the *far* field is comparable to that of the *near* field, so more ray trace colour is used. However, there are more *near* hits closer to the edges of the ray mask, favouring the ray trace colour. Hence, using our hit ratio as-is causes blur discontinuities and tiling artifacts from the mask. To minimize them while trying to retain the ray-traced semi-transparencies, we only blend at hit ratios below a certain value to produce a smooth transition between the ray trace and post-process colours around the edges of the ray mask.

3 DISCUSSION

Partial occlusion is more accurate for hybrid DoF as background details which would otherwise be hidden are uncovered in the silhouettes of blurry foreground objects with ray tracing. In contrast, many post-process approaches only perform a local approximation of the background with neighbouring pixels or grow blur out of the silhouette of foreground objects, reusing foreground information to avoid reconstructing the missing background [Jimenez 2014].

In future, we plan to scale the ray mask resolution based on the maximum CoC to prevent tiling artifacts and estimate per-pixel blend factors [Schied et al. 2018] to minimize ghosting. We also intend to extend our approach to support alternative bokeh shapes and not just circular ones of uniform intensities, by varying the shape of our sampling kernel and the relative weight of its samples based on the desired silhouette of the camera aperture and spherical aberration quality of its lens respectively.

ACKNOWLEDGMENTS

This work is supported by the Singapore Ministry of Education Academic Research grant T1 251RES1812, "Dynamic Hybrid Real-time Rendering with Hardware Accelerated Ray-tracing and Rasterization for Interactive Applications".

REFERENCES

- Guillaume Abadie. 2018. Advances in Real-Time Rendering in Games: A Life of a Bokeh. In *ACM SIGGRAPH 2018 Courses* (Vancouver, British Columbia, Canada) (SIGGRAPH '18). Association for Computing Machinery, New York, NY, USA, Article 20, 1 pages. <https://doi.org/10.1145/3214834.3264541>
- Jorge Jimenez. 2014. Advances in Real-Time Rendering in Games, Part I: Next Generation Post Processing in Call of Duty: Advanced Warfare. http://advances.realtimerendering.com/s2014/#_NEXT_GENERATION_POST
- Jaakko Lehtinen, Timo Aila, Jiawen Chen, Samuli Laine, and Frédo Durand. 2011. Temporal Light Field Reconstruction for Rendering Distribution Effects. *ACM Trans. Graph.* 30, 4, Article 55 (July 2011), 12 pages. <https://doi.org/10.1145/2010324.1964950>
- Michael Potmesil and Indranil Chakravarty. 1982. Synthetic Image Generation with a Lens and Aperture Camera Model. *ACM Trans. Graph.* 1, 2 (April 1982), 85–108. <https://doi.org/10.1145/357299.357300>
- Christoph Schied, Anton Kaplanyan, Chris Wyman, Anjul Patney, Chakravarty R. Alla Chaitanya, John Burgess, Shiqiu Liu, Carsten Dachsbacher, Aaron Lefohn, and Marco Salvati. 2017. Spatiotemporal Variance-guided Filtering: Real-time Reconstruction for Path-traced Global Illumination. In *Proceedings of High Performance Graphics* (Los Angeles, California) (HPG '17). ACM, New York, NY, USA, Article 2, 12 pages. <https://doi.org/10.1145/3105762.3105770>
- Christoph Schied, Christoph Peters, and Carsten Dachsbacher. 2018. Gradient Estimation for Real-Time Adaptive Temporal Filtering. *Proc. ACM Comput. Graph. Interact. Tech.* 1, 2, Article 24 (Aug. 2018), 16 pages. <https://doi.org/10.1145/3233301>