STAG: Smart Tools and Applications in Graphics (2023)
G. Manfredi, G. Caggianese, N. Capece, and U. Erra (Editors)

Poster

A Foveated Framework to Accelerate Real-Time Path Tracing

Bipul Mohanto[†1], Sven Kluge [1], and Oliver Staadt [1]

[1]University of Rostock, Institute of Visual and Analytic Computing, Germany

Abstract

We developed a framework to accelerate real-time path tracing through foveated rendering, a robust technique that leverages human vision. Our dynamic foveated path-tracing framework integrates fixations and selectively lowers the rendering resolution towards the periphery. Through comprehensive experimentation, we demonstrated the effectiveness of our framework in this paper. Our solution can significantly enhance rendering performance, up to 25× without any notable visual differences. We further evaluated the framework using a structured error map algorithm with variable sample numbers.

CCS Concepts

• ***Computing methodologies*** → ***Ray tracing; Perception;***

1. Introduction

We present a foveated framework that allows real-time path tracing on geometry-intensive complex scenes at a *4K resolution*. Using foveated path tracing, we can significantly improve the rendering performance. Unlike rendering the entire scene uniformly, our framework only renders the foveated area homogeneously and further divides the screen space into two subregions, rendering in a $(n \times n)$ and $(m \times m)$ pixel-block approach.

2. Related work

The foveated path tracing remains a challenging task, with limited available research. Recently, Koskela et al. [KIV*18, KLM*19] have made significant strides in the field, pioneering foveated path tracing on VR. Other notable studies include the first foveated path tracing framework for smartphone-based VR [KS21], emulated foveation for eccentricity angle threshold [PKM21], and a framework for tiled display walls [RCMPS23].

3. The rendering framework

3.1. Area of interest

At the beginning of the pipeline (Figure 1.a), we extract the fixations in Cartesian space to determine the three consecutive regions and then calculate the Euclidean distance for each pixel. To match the highest visual acuity, we exploited the *tangent* formula [Mat97], $a = 2 \cdot d \cdot tan\frac{e}{2}$, to convert the cycles per degree into pixels per degree. Here, e is the eccentricity angle (degree) and d is the distance from the eye (cm). Further, we leveraged this formula to convert eccentricity into pixel number $r = \frac{a \cdot R}{A}$. Here, A is the display dimension in pixels, and R is the display size (cm). We used a 60 cm distance in our framework. Our display has a $4k$ resolution with an area of $(70.848 \times 39.852)\ cm^2$. The approximate values for the foveated and intermediate radii are 147 and 482 pixels.

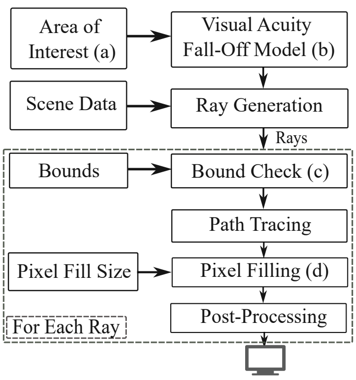

Figure 1: *Foveated path tracing pipeline.*

3.2. Piecewise visual acuity fall-off model

We have employed the piecewise visual acuity fall-off model (Figure 1.b) that follows the widely accepted sample distribution ratio, $1, \frac{1}{2}$, and $\frac{1}{4}$ [LW90, GFD*12]. To prevent visual artifacts in real-time ray tracing, at least 32 samples per pixel have been suggested [FH14] which we applied in piecewise sampling ratio:

$$f(e) = \begin{cases} \tau_1/px, & if,\ e \le e_f, & \text{(1a)} \\ \tau_2/(n \times n)px, & elif,\ e_f < e \le e_i, & \text{(1b)} \\ \tau_3/(m \times m)px, & otherwise. & \text{(1c)} \end{cases}$$

† bipul.mohanto@uni-rostock.de

DOI: 10.2312/stag.20231305

The samples ($\tau_1 = 32, \tau_2 = 16, \tau_3 = 8$) propagate according to the eccentricity angle (e), e_f is the foveated (5.2°) and e_i is the intermediate (17°) angle [MIGS22]. Here, $n = 2$ and $m = 4$ are the pixel numbers that create a block. Finally, we blend the three regions under one single rendering pass.

3.3. Bound check and path tracing

Before casting any primary ray, we check its initial position in screen space against the boundaries of the radii (Figure 1.c). The bound check is simply a binary operation. If the ray does not meet the bounds check, we discard it immediately. The path tracing [Kaj86] is the backbone of our framework. In addition, we employed the multiple importance sampling [Vea98] and Disney BRDF [BS12]. We limited the recursion depth to 3 in the scene to avoid further computation load but allow global illumination.

3.4. Pixel filling and post-processing

The final aspect of our framework is the pixel-filling step (Figure 1.d). As we cast fewer rays than the actual pixel number, the missing pixels are filled with the respective pixel block of the scene using standard linear interpolation. Additionally, we applied subpixel jitter, exposure compensation, and Reinhard tone mapping [SbMEMS12] during post-processing for a mellow output.

3.5. Framework setup

The framework is developed with OptiX 7.5 API [PBD*10] with CUDA 12.1. We utilized an Intel Core i9 12900Ks processor with 128 GB of DDR5 RAM and an NVIDIA GeForce RTX 3090 GPU with 24 GB VRAM.

4. Results and discussion

Table 1 presents the performance comparison with the previously mentioned setup (Section 3.5). The test results show that without foveation, Lost Empire, Crytek Sponza, and San Miguel 2.0 have an average fps, 0.42, 0.24, and 0.23, respectively, while with foveation, the average fps, 10.20, 7.48, and 5.70, respectively. This is on average more than 25× performance boost. The outputs are presented in Figure 2. Without a denoiser, the outputs with lower sampling numbers are naturally noisier.

Table 1: *The $2^{nd}, 3^{rd}, 4^{th}$, and 5^{th} column exhibits the triangle count, average fps (32 samples), average fps (32, 16, 8 samples), and the speed up, respectively.*

Scene	Triangles	fps-uni	fps-fov	Speed up
Lost Empire	224,998	0.42	10.20	23.29
Crytek Sponza	262,267	0.24	7.48	30.17
San Miguel 2.0	9,980,699	0.23	5.71	23.83

Applying a more aggressive foveation area and/or sample number can further improve the rendering performance. For instance, reducing the foveated and intermediate areas in half on the Crytek Sponza while maintaining the same foveated sample number

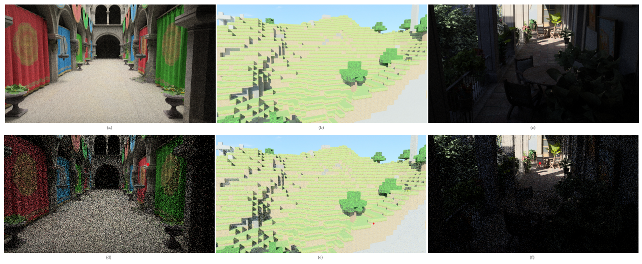

Figure 2: *Visual comparison between uniform (top row) and foveated path tracing (bottom row). We selected Crytek Sponza (a, d), Lost Empire (b, e), and San Miguel 2.0 (c, f) for evaluation.*

(Table 1) and pixel-block approach can speed up the rendering process by 44.67 times. Besides, we tested different sample configurations on the Crytek Sponza to compare performance differences (Table 2).

Table 2: *The speed up of Crytek Sponza using different sampling configurations.*

Scene	Fovea	Intermediate	Periphery	fps	Speed up
Crytek Sponza	32	–	–	0.30	–
	32	16	8	7.48	23.93
	32	8	4	10.90	35.33
	32	2	1	17.14	56.13

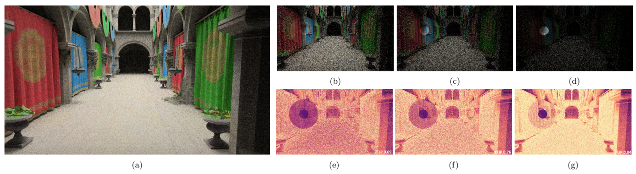

Figure 3: *Visual outputs with (a) uniform path tracing, (b)–(d) the foveated sample number (32,16,8), (32,8,4), and (32,2,1), (e)–(g) the FLIP error map wrt. the reference image (a).*

Besides, we further evaluated the distinct sampling combinations with the latest structured error map, FLIP [ANAM*20] that focuses on differences between rendered and reference images. The algorithm generates a map that approximates the perception difference between two images. Figure 3 illustrates the perceptual difference between different sampling combinations employed in Table 2. The areas highlighted in black and yellow represent alikeness and perceptual errors, respectively.

5. Conclusion

We presented a novel foveated framework that employs the variable sampling number and pixel-block rendering approach. This framework can meet the high computational demands of path tracer on a pixel-intensive display and provides an efficient, flexible, and practical solution.

References

[ANAM*20] ANDERSSON P., NILSSON J., AKENINE-MÖLLER T., OSKARSSON M., ÅSTRÖM K., FAIRCHILD M. D.: Flip: A difference evaluator for alternating images. *Proc. ACM Comput. Graph. Interact. Tech. 3*, 2 (aug 2020). URL: https://doi.org/10.1145/3406183, doi:10.1145/3406183. 2

[BS12] BURLEY B., STUDIOS W. D. A.: Physically-based shading at disney. In *Acm Siggraph* (2012), vol. 2012, vol. 2012, pp. 1–7. 2

[FH14] FUJITA M., HARADA T.: Foveated real-time ray tracing for virtual reality headset. *Light Transport Entertainment Research* (2014). 1

[GFD*12] GUENTER B., FINCH M., DRUCKER S., TAN D., SNYDER J.: Foveated 3d graphics. *ACM Trans. Graph. 31*, 6 (nov 2012). URL: https://doi.org/10.1145/2366145.2366183, doi:10.1145/2366145.2366183. 1

[Kaj86] KAJIYA J. T.: The rendering equation. *SIGGRAPH Comput. Graph. 20*, 4 (aug 1986), 143–150. URL: https://doi.org/10.1145/15886.15902, doi:10.1145/15886.15902. 2

[KIV*18] KOSKELA M. K., IMMONEN K. V., VIITANEN T. T., JÄÄSKELÄINEN P. O., MULTANEN J. I., TAKALA J. H.: Instantaneous foveated preview for progressive monte carlo rendering. *Computational Visual Media 4*, 3 (2018), 267–276. URL: https://www.sciopen.com/article/10.1007/s41095-018-0113-0, doi:10.1007/s41095-018-0113-0. 1

[KLM*19] KOSKELA M., LOTVONEN A., MÄKITALO M., KIVI P., VIITANEN T., JÄÄSKELÄINEN P.: Foveated Real-Time Path Tracing in Visual-Polar Space. In *Eurographics Symposium on Rendering - DL-only and Industry Track* (2019), Boubekeur T., Sen P., (Eds.), The Eurographics Association. doi:10.2312/sr.20191219. 1

[KS21] KIM S. K., SUNG M. K.: Efficient path tracer for the presence of mobile virtual reality. *Hum.-Cent. Comput. Inf. Sci 11* (2021), 1–14. 1

[LW90] LEVOY M., WHITAKER R.: Gaze-directed volume rendering. *SIGGRAPH Comput. Graph. 24*, 2 (feb 1990), 217–223. URL: https://doi.org/10.1145/91394.91449, doi:10.1145/91394.91449. 1

[Mat97] MATKOVIC K.: *Tone mapping techniques and color image difference in global illumination*. PhD thesis, Matkovic, 1997. 1

[MIGS22] MOHANTO B., ISLAM A. T., GOBBETTI E., STAADT O.: An integrative view of foveated rendering. *Computers & Graphics 102* (2022), 474–501. URL: https://www.sciencedirect.com/science/article/pii/S0097849321002211, doi:https://doi.org/10.1016/j.cag.2021.10.010. 2

[PBD*10] PARKER S. G., BIGLER J., DIETRICH A., FRIEDRICH H., HOBEROCK J., LUEBKE D., MCALLISTER D., MCGUIRE M., MORLEY K., ROBISON A., STICH M.: Optix: A general purpose ray tracing engine. *ACM Trans. Graph. 29*, 4 (jul 2010). URL: https://doi.org/10.1145/1778765.1778803, doi:10.1145/1778765.1778803. 2

[PKM21] POLYCHRONAKIS A., KOULIERIS G. A., MANIA K.: Emulating foveated path tracing. In *Proceedings of the 14th ACM SIGGRAPH Conference on Motion, Interaction and Games* (New York, NY, USA, 2021), MIG '21, Association for Computing Machinery. URL: https://doi.org/10.1145/3487983.3488295, doi:10.1145/3487983.3488295. 1

[RCMPS23] ROMERO CALLA L. A., MOHANTO B., PAJAROLA R., STAADT O.: Multi-Display Ray Tracing Framework. In *Eurographics 2023 - Posters* (2023), Singh G., Chu M. R., (Eds.), The Eurographics Association. doi:10.2312/egp.20231025. 1

[SbMEMS12] SALIH Y., BT. MD-ESA W., MALIK A. S., SAAD N.: Tone mapping of hdr images: A review. In *2012 4th International Conference on Intelligent and Advanced Systems (ICIAS2012)* (2012), vol. 1, pp. 368–373. doi:10.1109/ICIAS.2012.6306220. 2

[Vea98] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford, CA, USA, 1998. AAI9837162. 2