Are we done with Ray Tracing?

Alexander Keller
NVIDIA
keller.alexander@gmail.com

Timo Viitanen
NVIDIA
tviitanen@nvidia.com

Colin Barré-Brisebois
Electronic Arts SEED
cbarrebrisebois@ea.com

Christoph Schied
Facebook Reality Labs
schied@fb.com

Morgan McGuire
NVIDIA
mcguire@nvidia.com

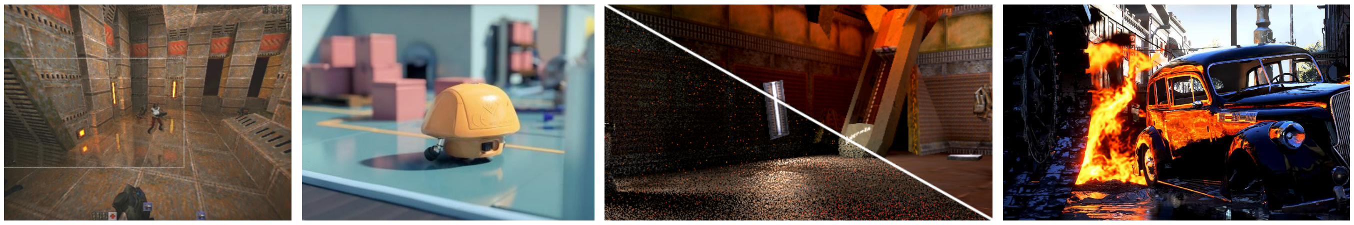

Figure 1: Software ray tracing under real-time constraints in 2005 allowed for 3 rays per pixel in a subwindow of the Quake II game on a dual core laptop (left). With the release of ray tracing APIs, games studios came up with astonishing realtime demos combining both rasterization and ray tracing (middle left). Today, with ray tracing hardware acceleration, Quake II can be played with filtered path traced global illumination at 1080p resolution and a frame rated of 60Hz (middle right) and first games adopt ray tracing (right, Battlefield V image courtesy DICE, EA, and NVIDIA).

ABSTRACT

Real-time graphics has come a long way since the "brute-force approach" of rasterization had been classified "ridiculously expensive" in 1974. Henceforth the promise "Ray tracing is the future and ever will be" drove the development of ray tracing algorithms and hardware, and resulted in a major revolution of image synthesis. This course will take a look at how far out the future is, review the state of the art, and identify the current challenges for research. Not surprisingly, it looks like we are not done with ray tracing, yet.

CCS CONCEPTS

• **Computing methodologies** → **Ray tracing**; **Graphics processors**.

KEYWORDS

Ray tracing, acceleration data structures, real-time light transport simulation.

ACM Reference Format:
Alexander Keller, Timo Viitanen, Colin Barré-Brisebois, Christoph Schied, and Morgan McGuire. 2019. Are we done with Ray Tracing?. In *Proceedings of SIGGRAPH '19 Courses.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3305366.3328096

1 INTRODUCTION

Since the path tracing revolution in the movie industry [1] offline image synthesis almost exclusively is based on ray tracing. Now that hardware acceleration is available, ray tracing, especially under real-time constraints, has become feasible more than ever before.

This course takes an in-depth look at the transition from rasterization to ray tracing and the challenges that come with this major shift of paradigms. Roughly partitioning frame generation time in updating an acceleration data structure, tracing the rays, and everything that remains, it becomes obvious that even if ray tracing was infinitely fast, dealing with dynamic geometry and shading would become a major bottleneck.

Ray tracing dynamic tessellations and displacements, as well as foliage, hair, and particle based geometry in real-time remains challenging. Yet, compared to shadow mapping, ray tracing provides precise visibility and allows for a much better scaling on massively parallel hardware, because it is sampled-based. Then again, ray tracing with adaptive level of detail is an unsolved problem.

In addition, accessing information for shading and cut-outs, especially textures, becomes more expensive when rays are incoherent. While it is simple to trace rays across multiple specular reflections and refractive transparency, this process comes at major cost, because it is inherently sequential.

These apparent contradictions in fact are research challenges and we discuss principle approaches to their solution. Future components of ray tracing hardware are discussed in Sec. 1.1 and challenges of ray tracing in games are reviewed from the point of view of a game studio in Sec. 1.2. Sec. 1.3 features a first solution to real-time path tracing in Quake II. The final Sec. 1.4 surveys the research and commercial state of the art in ray tracing for games, and charts a course for open problems to pursue in this area.

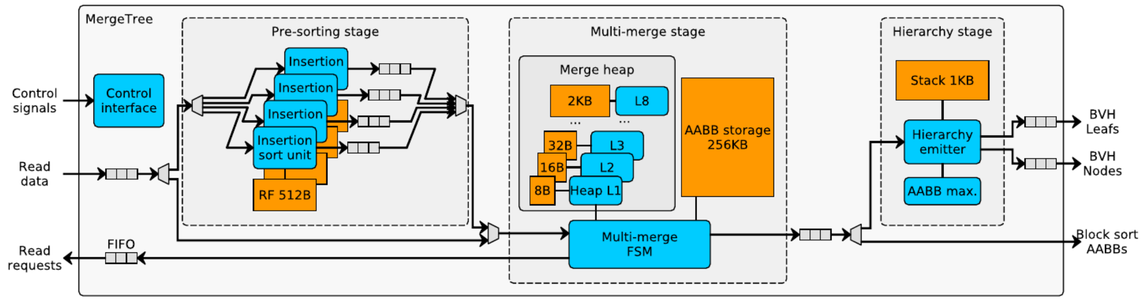

Figure 2: Schematics of a hardware architecture to construct a bounding volume hierarchy for accelerated ray tracing.

1.1 Acceleration Data Structure Hardware

Powerful hardware accelerators have been recently developed and real-time ray tracing is even in the reach of mobile devices. However, rendering animated scenes remains difficult, as updating the ray tracing acceleration data structures for each frame is a memory-intensive process. Fig. 2 shows the schematics of a hardware architecture [3]. It is the first one designed to minimize memory traffic when constructing a hierarchical linear bounding volume hierarchy (HLBVH). For evaluation, the hardware has been synthesized on a 28nm CMOS process technology. Compared to a state-of-the-art software implementation, half an order of magnitude speedup is achieved, while energy and memory traffic have been reduced by a factor of 3.

1.2 Game Ray Tracing: State of the Art

We will present the state of the art in game real-time ray tracing, discuss techniques from shipped products, and how ray tracing APIs are used. Besides the challenges in real-time game ray tracing, parallels to offline rendering are addressed. Then, real-world game production open problems that need to be researched at scale and quality are reviewed. Finally, we dare to predict industry trends for the next 2 years and where the research community should be invested.

1.3 Reconstruction for Real-Time Path Tracing

Despite recent advancements in consumer graphics hardware, the achievable sample counts for real-time path tracing remain low. To bridge this gap, real-time reconstruction filters remove the noise from extremely low sample count images, thus allowing to synthesize noise-free images at real-time refresh rates (60Hz).

Such reconstruction filters amortize cost by reusing information both spatially and temporally. To reuse information spatially, spatio-temporal variance-guided filtering relies on an estimate of the level of noise to set the filter bandwidth for a hierarchical filter. To prevent temporal filtering artifacts, intelligent temporal filters are required. Adaptive temporal filtering [2] tracks changes of the sampled signal using an estimate of the temporal gradient that is reconstructed from a sparse set of stochastic samples.

We evaluate the applicability and robustness of the aforementioned reconstruction techniques and identify shortcomings of current path tracing techniques using a prototypical implementation in the Quake 2 engine (see http://brechpunkt.de/q2vkpt).

1.4 From Rasterization to Ray Tracing

Over the span of a decade, the film industry adopted ray tracing in two stages: First came a hybrid model with micro-polygon rasterization and manually-placed bounce lights for most geometry, and ray tracing for reflections and some hard shadows (e.g., *A Bug's Life* (1998)). Second came the full switch to path tracing [1] (e.g., *Cloudy with a Chance of Meatballs* (2009)). The latter was driven largely by workflow concerns: a unified visibility and shading solution across all effects is more robust and predictable for artists as well as more practical for engineers to maintain and extend. It is reasonable to expect a similar trajectory of adoption in games.

The first stage of hybrid adoption as ray tracing layered on rasterization pipelines is occurring within games already. Ray tracing for glossy and mirror reflections and shadows is present in commercial games such as *Battlefield V* (2018, see Fig. 1) and *Shadow of the Tomb Raider* (2018). These effects in particular benefit from performance optimizations borrowed from previous, mature game techniques such as screen-space ray tracing and shadow map biasing and filtering. The emerging research on these effects addresses quality limitations from low sample counts and performance in working with large scenes.

The next stage is a true integrated ray and raster pipeline. This presents several open problems: Hybrid primary visibility for antialiasing and managing level of detail, efficient support for volumetric phenomenon and animated instances, true displacement maps and spline patch primitives, and proper filtering for texture samples (because current ray tracing APIs lack the automatic spatial derivatives of rasterization shaders). These are all related to visibility from the camera. On the shading side, the goal should be to replace the full global illumination pipeline with path tracing to achieve the same integration benefits that film enjoys today.

We end the course speculating about possible long-term strategies and research challenges of game rendering.

REFERENCES

[1] A. Keller, L. Fascione, M. Fajardo, I. Georgiev, P. Christensen, J. Hanika, C. Eisenacher, and G. Nichols. 2015. The Path Tracing Revolution in the Movie Industry. In *ACM SIGGRAPH 2015 Courses (SIGGRAPH '15)*. ACM, Article 24.
[2] C. Schied, C. Peters, and C. Dachsbacher. 2018. Gradient Estimation for Real-Time Adaptive Temporal Filtering. In *Proceedings of High Performance Graphics (HPG '18)*. ACM.
[3] T. Viitanen, M. Koskela, P. Jääskeläinen, H. Kultala, and J. Takala. 2017. MergeTree: A Fast Hardware HLBVH Constructor for Animated Ray Tracing. *ACM Trans. Graph.* 36, 5 (2017), 169:1–169:14.