Embree Ray Tracing Kernels: Overview and New Features

Attila T. Áfra* Ingo Wald† Carsten Benthin‡ Sven Woop§
Intel Corporation

Figure 1: *Images rendered with Embree. Mazda model by Evermotion, Imperial Crown of Austria model by Martin Lubich (http://www.loramel.net), Toad King model by Craig Barr.*

Abstract

Embree is an open source ray tracing library consisting of high-performance kernels optimized for modern CPUs with increasingly wide SIMD units. Since its original release, it has become the state-of-the-art for professional CPU-based rendering applications. In the first half of this talk, we will give a brief overview of the Embree framework and how to use it. In the second half, we will present recent improvements and features introduced since the initial publication of the system. These additions include new geometry types commonly used in production renderers (quads, subdivision surfaces, and hair), improved motion blur support, and ray streams that can be traversed more efficiently than single rays and ray packets.

Keywords: ray tracing, SIMD, CPU

Concepts: •**Computing methodologies** → **Ray tracing;**

1 Introduction

Embree [Wald et al. 2014] is an open source ray tracing framework designed to achieve high performance in professional rendering environments. It features a set of state-of-the-art spatial acceleration structures and ray traversal kernels optimized for the latest x86 processor architectures, leveraging the SSE, AVX, AVX2, AVX-512, and IMCI instruction sets. The Embree API enables these kernels to be used effectively in existing and new renderers with minimal programming effort.

Embree has received increasing attention since its original publication and has become the method of choice for a wide range of ray tracing based applications, including production renderers and visualization systems. Also, major improvements and new features have been introduced, such as faster ray traversal and acceleration structure building, quad geometry, subdivision surfaces, hair, linear motion blur of user defined geometries, and ray streams.

2 New Features

Quads. Until recently, Embree supported only triangle meshes, but most 3D modeling packages produce quad meshes which first had to be converted to triangles. Embree now supports quads as well, and using them results in half the memory usage in the best case, faster BVH building, and higher ray intersection throughput.

Subdivision surfaces. We have implemented Catmull-Clark subdivision surfaces for triangle and quad meshes, including support for edge creases, vertex creases, holes, non-manifold geometry, and displacement mapping. We employ a fixed-size lazy-build tessellation cache [Benthin et al. 2015], which allows real-time rendering performance for complex scenes and low memory usage.

Hair. We have added support for hair geometries consisting of cubic Bézier curves [Woop et al. 2014] and, more recently, line segments as well, with varying radius per control point. For hair we build BVHs that use both axis-aligned and oriented bounding boxes to exploit similarity in the orientation of neighboring hairs, increasing ray culling efficiency.

Motion blur of user geometries. The user defined geometry feature enables renderer developers to implement custom geometric primitives and intersection algorithms that are not available in Embree by default (e.g., points and disks). Rendering these user geometries with linear motion blur has been recently implemented, which is a common requirement of production renderers.

Ray streams. In addition to single rays and SIMD-sized ray packets, it is now possible to intersect arbitrarily large streams of rays as well. Ray streams enable better coherence extraction, which results in higher rendering performance. We provide novel stream traversal kernels that group rays with similar directions together and process them in a round-robin fashion to better hide memory access latency, which is the main bottleneck of ray traversal on CPUs.

*attila.t.afra@intel.com
†ingo.wald@intel.com
‡carsten.benthin@intel.com
§sven.woop@intel.com

SIGGRAPH '16, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4282-7/16/07
DOI: http://dx.doi.org/10.1145/2897839.2927450

References

BENTHIN, C., WOOP, S., NIESSNER, M., SELGRAD, K., AND WALD, I. 2015. Efficient ray tracing of subdivision surfaces using tessellation caching. In *Proceedings of the 7th High-Performance Graphics Conference*, ACM.

WALD, I., WOOP, S., BENTHIN, C., JOHNSON, G. S., AND ERNST, M. 2014. Embree: A kernel framework for efficient CPU ray tracing. *ACM Trans. Graph. 33*, 4 (July), 143:1–143:8.

WOOP, S., BENTHIN, C., WALD, I., JOHNSON, G. S., AND TABELLION, E. 2014. Exploiting local orientation similarity for efficient ray traversal of hair and fur. In *High-Performance Graphics 2014, Lyon, France, 2014. Proceedings*, 41–49.