Synergy effects of hybrid CPU-GPU architectures for Interactive Parallel Ray Tracing

Biagio Cosenza

ISISLab, Dipartimento di Informatica ed Applicazioni
Università degli Studi di Salerno, Italy
cosenza@dia.unisa.it

Rendering algorithms range from off-line, computationally costly, physically-based, realistic techniques, to fast, approximate and plausible interactive ones. Ray tracing is a popular rendering technique aiming for high realism, and is a basis for global illumination algorithms. Interactive ray tracing has seen enormous progress in recent years. Highly optimized packet-based ray tracing implementations allow the computation of millions of ray-triangle intersection per second, and fully exploit modern multi-core CPUs, or GPUs. However, complex scenes and lighting, and high-quality renderings with anti-aliasing are still not feasible at interactive speed, and only possible when using compute clusters. In these scenarios, good load balancing is crucial in order to exploit the computational power, and not to suffer from communication overhead and synchronization barriers. We introduce a Parallel Ray Tracing architecture that exploits several parallelization techniques in order to reach interactive performance. First, it exploits SIMD vector instructions and more coherent memory accesses by tracing more rays at once (packet of rays). Second, we take advantage of the multi core availability by using multiple threads and scheduling packets of rays to different threads. Finally, we afford the distributed memory parallelization by using several techniques. In particular, we introduce a method, which uses a cheap GPU rendering technique to compute a cost map: An estimation of the per-pixel cost when rendering the image using ray tracing. Using this information, we improve load balancing, task scheduling, and work stealing strategies.