Bridging Ray and Raster Processing on GPUs

Kenny Mitchell
Black Rock Studio

Christian Oberholzer
ETH Zurich

Jan Kautz
University College London

Peter-Pike Sloan
Disney Interactive Studios

ABSTRACT

Real-time graphics is dominated by hardware accelerated rendering pipelines based on rasterization. Such pipelines are the result of many years of development exploiting the locality and coherence of compute kernels. Ray tracing, on the other hand, is often considered to be a tedious, difficult, and inefficient pattern of code. This work develops a solution leveraging the strengths of both techniques.

SCALABILITY

Hybrid Parallelization

We perform parallel processing in the selection of only visible ray paths and sampling from the 3D cache illumination into 2D image space for the deferred rasterization step. Upsampling the 2D irradiance texture with an edge aware cross bilinear filter [3] retains geometric detail in the process of achieving a smooth filtering of noisy irradiance cache entries. Instead of two additional raster passes, we apply a single horizontal filter pass and vertical sampling is incorporated into the deferred shading pass.

Parallel Raster Processing Steps

1. rasterize scene into G-buffer
2. downsample G-buffer to lower resolution
3. select only visible cells for upsample
4. for all cell in visible set of 3D cells do
5. ray trace secondary irradiance sampling from cell
6. collect 3D visible cell results into 2D image space
7. rasterize image space global illumination
8. upsample combined blend of 6 and 7 results
9. rasterize deferred local shading with G-buffer and

DESIGN

System

A deferred shading process [2], which is often used in video game rendering engines to reduce lighting and shading calculations to only visible pixels, generates G-Buffer albedo, positions and normals for use in both visible ray spawn cell location and the raster shader. The deferred shading process is applied to an image space global illumination pass.

ANIMATION

As global illumination is continuously updated, the system is applicable to scenes with animated light sources and deformable geometry. By artificially limiting the range of indirect illumination to a quadratic fall-off, a reduced volume of animated cache updates may be applied as in the squashing ball and bouncing bunny in Sponza animations above.

REFERENCES


