Abstract

GPU-Accelerated Point-Based Color Bleeding

Ryan Schmitt

Traditional global illumination lighting techniques like Radiosity and Monte Carlo sampling are computationally expensive. This has prompted the development of the Point-Based Color Bleeding (PBCB) algorithm by Pixar in order to approximate complex indirect illumination while meeting the demands of movie production; namely, reduced memory usage, surface shading independent run time, and faster renders than the aforementioned lighting techniques [8].

The PBCB algorithm works by discretizing a scene's directly illuminated geometry into a point cloud (surfel) representation. When computing the indirect illumination at a point, the surfels are rasterized onto cube faces surrounding that point, and the constituent pixels are combined into the final, approximate, indirect lighting value.

In this thesis we present a performance enhancement to the Point-Based Color Bleeding algorithm through hardware acceleration; our contribution incorporates GPU-accelerated rasterization into the cube-face raster phase. The goal is to leverage the powerful rasterization capabilities of modern graphics processors in order to speed up the PBCB algorithm over standard software rasterization. Additionally, we contribute a preprocess that generates triangular surfels that are suited for fast rasterization by the GPU, and show that new heterogenous architecture chips (e.g. Sandy Bridge from Intel) simplify the code required to leverage the power of the GPU. Our algorithm reproduces the output of the traditional Monte Carlo technique with a speedup of 41.65x, and additionally achieves a 3.12x speedup over software-rasterized PBCB.