Interactive Indirect Illumination Using Voxel Cone Tracing: A Preview
Cyril Crassin, Fabrice Neyret, Miguel Sainz, Simon Green, Elmar Eisemann

To cite this version:
Cyril Crassin, Fabrice Neyret, Miguel Sainz, Simon Green, Elmar Eisemann. Interactive Indirect Illumination Using Voxel Cone Tracing: A Preview. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D), Feb 2011, San Francisco, California, United States. 2011. hal-00650196

HAL Id: hal-00650196
https://inria.hal.science/hal-00650196
Submitted on 9 Dec 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Interactive Indirect Illumination Using Voxel Cone Tracing

Cyril Crassin  Fabrice Neyret
INRIA / LJK / Grenoble University / CNRS

Miguel Sainz  Simon Green  Elmar Eisemann
NVIDIA Corporation  NVIDIA Corporation  Telecom ParisTech

Overview
Indirect illumination is an important element for realistic image synthesis, but its computation is expensive and highly dependent on the complexity of the scene and of the BRDF of the surfaces involved. We present a novel algorithm to compute global illumination in real-time that avoids costly precomputation steps and is not restricted to low frequency illumination. It is based on a voxel octree representation and an approximate voxel cone tracing that allows a fast estimation of the visibility and incoming energy. Our approach can manage both Lambertian and Glossy materials at interactive framerates (5-30FPS).

Global Algorithm
1. Store incoming radiance (energy and direction) from dynamic light sources into a sparse voxel octree hierarchy, by rasterizing scene meshes. Done efficiently in parallel by relying on screen-space quad-tree analysis.
2. Filter the values in the higher levels of the octree (mipmap). Done efficiently by calculating the Gaussian lobe representation of the irradiance distribution.
3. Render scene from the camera. For each visible surface fragment, we combine the direct and indirect illumination. We employ an approximate cone tracing to perform a final gathering, sending out a few cones over the hemisphere to collect illumination distributed in the octree.

Pre-integrated Voxel Cone Tracing
Our approach approximates the result of the visibility, energy and NDF estimation for a bundle of rays in a cone using only a single ray and our filtered (mipmapped) voxel structure. The idea is to perform volume integration steps along the cone axis with lookups in our hierarchical representation at the LOD corresponding to the local cone radius. During this step, we use quadrilinear interpolation to ensure a smooth LOD variation. Our voxel shading convolves the BRDF, the NDF, the distribution of light directions and the span of the view cone, all considered as Gaussian lobes. These lobes are reconstructed from direction distributions stored in a compact way as non-normalized vectors in the structure.

Sparse Voxel Octree Structure
The core of our approach is built upon a pre-filtered hierarchical voxel version of the scene geometry. For efficiency, this representation is stored in the form of a compact pointer-based sparse voxel octree in the spirit of [Crassin et al. 2009]. We use small 3² bricks with values located in octree-node corners. This structure exhibits an almost scene-independent performance and is suitable to be extended to out-of-core rendering, hereby allowing for arbitrarily complex scenes.

Anisotropic Pre-Integration
In order to get higher quality visibility estimation and to limit leaking of light when low resolution mipmap levels are used, we propose to rely on an anisotropic pre-integration of voxel values stored in a direction-dependant way in the structure. We use the 6 main directions and values are reconstructed by linear interpolation of the 3 closest directions. It provides a better approximation of the volume rendering integral, at the cost of 1.75x the storage requirement and a slightly slower sampling.