

Interactive Indirect Illumination Using Voxel-Based Cone Tracing : An Insight

Cyril Crassin
Grenoble Univ, INRIA

Fabrice Neyret
CNRS, Grenoble Univ, INRIA

Miguel Sainz
NVIDIA

Simon Green
NVIDIA

Elmar Eisemann
Telecom ParisTech



Scene courtesy of Guillermo M. Leal Llaguno

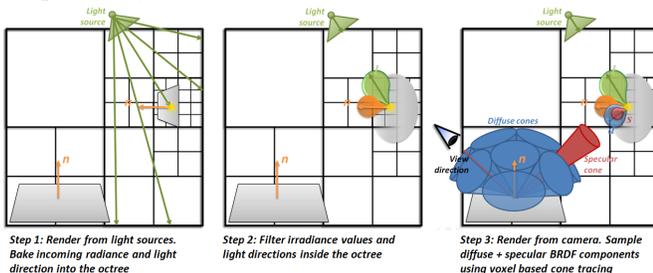
Figure 1: Real-time indirect illumination (25-70 fps on a GTX480): Our approach supports diffuse and glossy light bounces on complex scenes. We rely on a voxel-based hierarchical structure to ensure efficient integration of 2-bounce illumination. (Right scene courtesy of G. M. Leal Llaguno)

Abstract

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. While off-line computation and pre-baking can be acceptable for some cases, many applications (games, simulators, etc.) require real-time or interactive approaches to evaluate indirect illumination. We present a novel algorithm to compute indirect lighting in real-time that avoids costly precomputation steps and is not restricted to low frequency illumination. It is based on a hierarchical voxel octree representation generated and updated on-the-fly from a regular scene mesh coupled with an approximate voxel cone tracing that allows a fast estimation of the visibility and incoming energy. Our approach can manage two light bounces for both Lambertian and Glossy materials at interactive framerates (25-70FPS). It exhibits an almost scene-independent performance and allows for fully dynamic content thanks to an interactive octree voxelization scheme. In addition, we demonstrate that our *voxel cone tracing* can be used to efficiently estimate Ambient Occlusion.

1 Algorithm overview

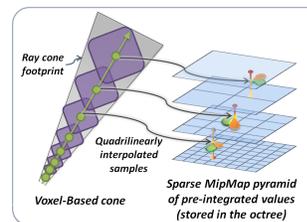
Our approach is based on a three-steps algorithm. We first inject incoming radiance (energy and direction) from dynamic light sources into the sparse voxel octree hierarchy, by rasterizing scene meshes and splatting a photon for each visible surface fragment. In a second step, we filter the values in the higher levels of the octree (mipmap). This is done efficiently in parallel by relying on screen-space quadtree analysis. Finally, we render the 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. Typically for Phong-like BRDF, a few large cones (~5) estimate the diffuse energy coming from the scene, while a tight cone in the reflected direction with respect to the viewpoint captures the specular component. The aperture of this cone is derived from the specular exponent of the material, allowing us to compute glossy reflections.



2 Dynamic 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]. To adapt to the specificities of the scenes, we use small 3^3 bricks with values located in octree-node corners. This allows us to employ hardware interpolation, while using less than half the amount of memory compared to previous solutions. This structure exhibits an almost scene-independent performance. It is updated dynamically thanks to a fast GPU based mesh voxelization and octree building system. This system handle dynamic updates of the structure, allowing for animated objects and dynamic modifications of the environment.

3 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, similarly to anti-aliasing filtering [Crassin et al. 2009]. 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

4 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-dependent 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.

References

CRASSIN, C., NEYRET, F., LEFEBVRE, S., AND EISEMANN, E. 2009. Gigavoxels : Ray-guided streaming for efficient and detailed voxel rendering. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D)*.