Interactive Indirect Illumination Using Voxel Cone Tracing: A Preview

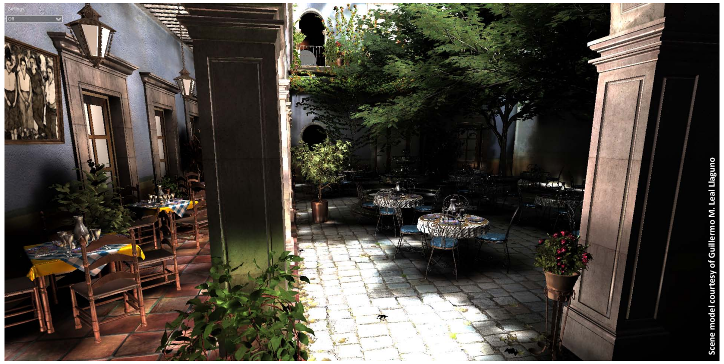
Cyril Crassin, Fabrice Neyret, Miguel Sainz, Simon Green, Elmar Eisemann

Interactive Indirect Illumination Using Voxel Cone Tracing

Cyril Crassin   Fabrice Neyret   Miguel Sainz   Simon Green   Elmar Eisemann
INRIA / LJK / Grenoble University / CNRS   NVIDIA Corporation   NVIDIA Corporation   Telecom ParisTech

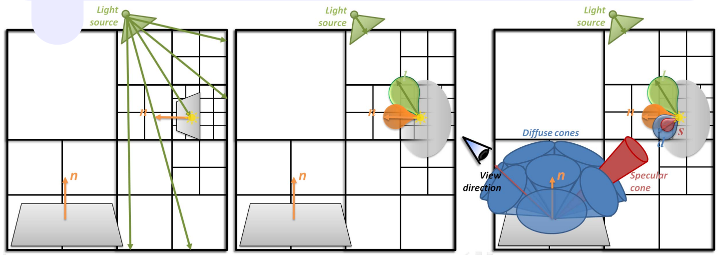
Overview

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)*.

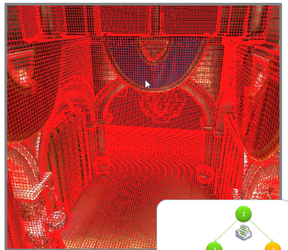


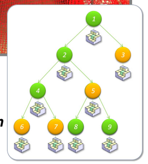
Global Algorithm

1. **Store** incoming radiance (energy and direction) from **dynamic light sources** into a **sparse voxel octree** hierarchy, by rasterizing scene meshes.
2. **Filter** the values in the higher levels of the octree (mipmap). Done efficiently in parallel by relying on **screen-space quad-tree analysis**.
3. **Render** scene from the **camera**. For each visible surface fragments, 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.

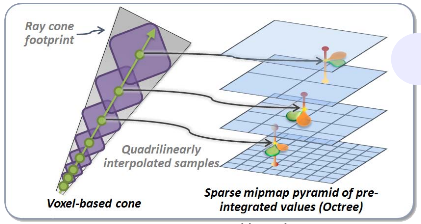


Step 1: Render scene from light sources. Bake incoming radiance and light direction into the octree
Step 2: Filter irradiance values and light directions inside the octree (mipmap)
Step 3: Render from camera. Sample diffuse + specular BRDF components using voxel based cone tracing

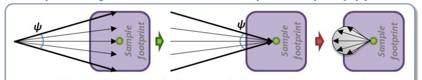




Sparse voxel octree + bricks representation storing filtered geometry, NDF and irradiance distribution



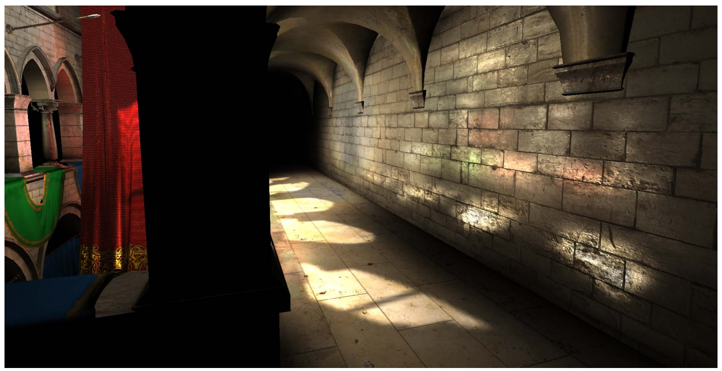
^ Approximate voxel based cone tracing using pre-integrated values inside a sparse mipmap pyramid



Gaussian lobe representation of the sustained directions distribution inside a cone over the footprint of a pre-integrated sample

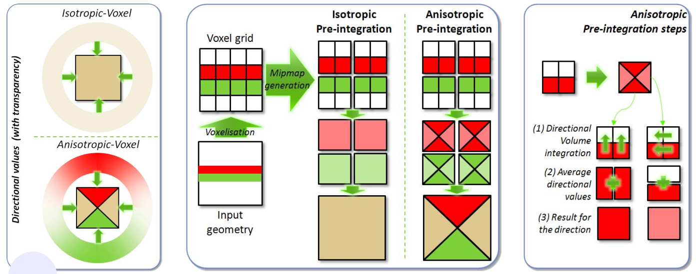
Sparse Voxel Octree Structure

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 3^3 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.



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.



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.

Reference: *GigaVoxels : Ray-guided streaming for efficient and detailed voxel rendering.* I3D 2009. CRASSIN, C., NEYRET, F., LEFEBVRE, S., AND EISEMANN, E.

We thank Guillermo M Leal Llaguno for the San Miguel Scene and Crytek for the improved Atrium Sponza palace model.