Beyond Triangles : Gigavoxels Effects In Video Games

Cyril Crassin, Fabrice Neyret — INRIA / Grenoble Univ.
Sylvain Lefebvre — INRIA
Miguel Sainz — NVIDIA Corporation
Elmar Eisemann — Saarland Univ. / MPI Informatik

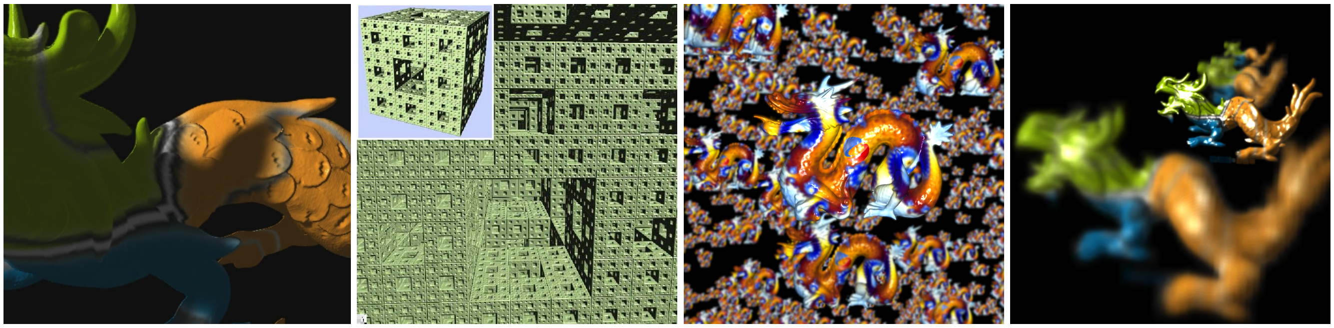

Figure 1: *Our technique renders complex scenes consisting of billions of voxels in real time and supports soft shadows, instancing, high-quality filtering, and depth-of-field effects.*

Overview

Voxel representations are commonly used for scientific data visualization, but also for many special effects involving complex or fuzzy data (e.g., clouds, smoke, foam). Since voxel rendering permits better and easier filtering than triangle-based representations it is also an efficient high-quality choice for complex meshes (with several triangles per pixel) and detailed geometric data (*e.g.,* boats in *Pirates of the Caribbean*).

We have shown in [Crassin et al. 2009] that highly detailed voxel data can be rendered in high quality at real-time rates. The work foreshadows the use of very large volumetric data sets in the context of video-games. Our system, based on ray-casting of a generalized sparse octree structure on GPU, achieves high rendering performance for billions of voxels.

To further underline the usefulness in the context of video games, this sketch introduces new features of our system, namely free object instantiation and the mixing with existing triangle scenes. We also demonstrate how to render complex visual effects like depth-of-fields or approximated soft shadows in very efficient ways, exploiting intrinsic properties of our multi-resolution scheme.

GigaVoxels rendering

In GigaVoxels, voxel data is efficiently managed by structuring small voxel volumes, so-called *bricks*, in form of an octree. In such a way, hardware-based tri-linear interpolation is combined with a fast traversal, even for semi-transparent volumes. One key feature of our method is hybrid update/ray-casting process. During the traversal of the hierarchical volume, rays record information about the traversed octree nodes. Consequently, a ray's return value is not only a color value. E.g., if visited nodes were not present at the correct resolution a subdivision requests is returned. Because the marching is terminated when the volume becomes opaque, intra-volume occlusion is taken into account and only visible data is loaded at the needed resolution and involving minimum CPU intervention. This out-of-core scheme drives a volume cache that manages the octree structure as well as the brick storage on the GPU in a unified manner.

Volume scene instancing

We extended GigaVoxels to allow scene-graph like instancing in order to create and render large scenes composed of millions of octree-based volumetric objects. To allow many semi-transparent objects, our method relies on a BVH acceleration structure. To maintain compatibility with our out-of-core update scheme, this structure is integrated in our ray-tracing process during the rendering. A new screen space structure allows us to efficiently disambiguate overlapping instances of semi-transparent volumes.

Soft shadows and depth-of-field effects

In [Crassin et al. 2009], we demonstrated that multisampling techniques can be efficiently approximated by preintegrated 3D mipmapping. Quadrilinear filtering during ray casting was enabled by maintaining a sparse octree with mipmapped bricks.

Soft shadows are usually approximated by integrating visibility over the surface of the light source. This relates to the amount of light that reaches a given point on an object. For a volumetric source, the rays towards the source form a bundle that we represent with a single cone. The blocking contribution of the object inside this cone is estimated by accumulating opacity. This is done in the same way as for eye rays, involving the mipmapping mechanism to approximate the cone integration. The resulting opacity value represents the amount of occlusion. This approach provides cheap, but plausible soft shadows while avoiding the classical multisampling.

Similarly, the mipmapping mechanism can be used to approximate a depth-of-field effect. The aperture of real cameras causes only points on a focal plane to project to a single point on the image, whereas all others produce a *circle of confusion*. During the eye-ray traversal, it is possible to compute this projection size and use it to determine a corresponding mipmap level for each sample lookup. The technique results in a cheap depth-of-field effect. As a side effect, the use of blurred data even accelerates the rendering, which is a powerful property.

In conclusion, GigaVoxels can be used to render many effects efficiently. Our ray-based update mechanism ensures to upload only the data that is actually needed for the image production which limits bandwidth consumption as well as pressure on the voxel cache.

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)*.