Fast Scene Voxelization Revisited
Elmar Eisemann, Xavier Décoret

To cite this version:

HAL Id: inria-00510222
https://inria.hal.science/inria-00510222
Submitted on 13 Oct 2010

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.
Abstract

This sketch paper presents an overview of “Fast Scene Voxelization and Applications” published at the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. It introduces slicemaps that correspond to a GPU friendly voxel representation of a scene. This voxelization is done at run-time in the order of milliseconds, even for complex and dynamic scenes containing more than 1M polygons. Creation and storage is performed on the graphics card avoiding unnecessary data transfer. Regular but also deformed grids are possible, in particular to better fit the scene geometry. Several applications are demonstrated: shadow calculation, refraction simulation and shadow volume culling/clamping.

1 Method

Voxelizing a scene means testing geometry for intersection with an englobing grid of cells. Our approach accomplishes this task efficiently with graphics hardware. Instead of interpreting RGBA as 4 unsigned byte values, we will consider it as a 32 bit vector, which itself corresponds to a 32 cell column in the voxelization. An image is thus nothing but a voxel grid, in which each cell is represented by one bit. We call this representation slicemap. The implementation is simple and allows the voxelization of millions of polygons in real-time. In the future, when 32 bit logical operations become available, 4 MRT1 × 4 channels × 32 bits = 512 slices are obtained in a single pass. Our approach shows several similarities to the work by Dong et al.[Dong et al. 2004], who also store depth in bits. Some major differences exist. First we do not fuse several views (which reduces cracks, but limits resolution). Second voxel grids usually contain axis-aligned cubes, whereas our approach offers several ways to deform the grid (e.g. improving precision by local refinement based on two depth masks containing the furthest and closest fragment for each pixel). We also present a fast technique to obtain a filled voxelization without iterative flood fill operations and present all-new applications.

2 Applications

Transmittance Shadow Maps: Illumination is attenuated based on the number of blockers traversed by each light ray before impinging on the surface. This calculation can be performed quickly due to the way we represent voxels and leads to convincing effects for complicated objects like trees or hair at low cost.

Shadow Volume Culling and Clamping: Slicemaps offer the possibility to cull and clamp shadow volumes [Lloyd et al. 2004]. The aforementioned possibility to deform the voxel grid is key here.

Refraction: Given a watertight object, we obtain a filled voxel representation avoiding flood fill. We estimate the object’s volume and use it for refraction and scattering effects.

3 Conclusion

The paper presents a GPU adapted voxel representation, applicable to any rasterizable object providing depth. The presented applications are not straightforward and present contributions on their own. Local refinement combines the advantage of exact shape information and approximated volume and results in a good overall estimation. Error analysis and conservativity remain future research.

References
