Sparse Voxel Octrees, SVOs, are hierarchical representations that can be used instead of 3D grids and also triangle meshes. Unlike 3D grids which quickly use up a lot of memory, SVOs can take advantage of the topology of the model. This way, only the voxels contributing to the geometry is actually stored. To achieve this, SVOs create a hierarchical representation of the space using octree space subdivisions. Other than volume rendering considerations, SVOs can also be used as building blocks for a generic renderer. Of course triangles are still behind most computer graphics. But as hardware gets faster and faster, efficiency perks of the triangle meshes gets shadowed. Even today, the most demanding part is the storing of contour information like normal, displacement maps etc. SVOs give a uniform representation for both modeling and shading.

In the SVO representation, each voxel stores necessary geometry, contour, shading, texture etc. info in its own cell, or rather together with its parent and sibling cells. This makes is possible to use one data structure for both geometry and shading, no separate structures for storing textures, displacement maps etc. Uniformity is the key behind SVOs’ simplicity. There are couple of techniques developed for efficient ray casting, mega textures, compression etc. Also this hierarchical representation makes LOD calculations a lot easier, you cast go up on the tree. So, the key idea behind SVOs is to use it as a uniform way to specify geometry, stream and traverse this data structure only when needed, i.e. when visible or near enough etc.

Although we used trees mainly for range searching problems in the lectures, we believe exploring geometric modeling features of SVOs is beneficial for use. We are both already using octrees in our projects for things like grass simulation, terrain rendering etc. This efficient sparse model will hopefully help on these projects.

References:

Samuli Laine and Tero Karras, Efficient Sparse Voxel Octrees -- Analysis, Extensions, and Implementation, Feb 2010, NVIDIA Corporation