Volume Terrain Modeling and Rendering for Flight Simulation

Abstract: We present a non-traditional approach for flight simulation and training, mission planning and rehearsal. It involves photo-realistic volume-based modeling and rendering of real world terrains from a single elevation map and a satellite or aerial photograph of the terrain. This has been achieved using voxelization, 3D texture splatting and ray-casting. We demonstrate the ability of our approach to create photo-realistic renderings of terrains from any viewpoint.

Previous Work: Recently, creating volumetric terrain models from elevation maps and satellite or aerial photographs has received increased attention. [1, 2, 3]. However, these systems have a number of drawbacks; in particular, they handle only 2.5D voxels which leads to 3D aliasing, when the viewpoint is changed or when viewing a region of sharp boundaries. Our research aims to make the modeling of terrain more convenient, more accurate and more photo-realistic. To do this, we have developed a new technique based on volume textures for accurate modeling of terrains, where the final images is rendered using ray-casting of the texture model.

Volume Texture Modeling: The R, G and B components of the 2D texture and the elevation for each texel is used in terrain voxelization to determine all the voxels under the terrain and their corresponding pre-texture. The final volume texture model is obtained by splatting all the voxels into a 3D array of RGB texture voxels, using a 3D Gaussian splat kernel. It achieves smooth interpolation for discrete normal estimation at corners and sharp boundaries. The advantage of this scheme is that the 3D texture is pre-computed and provides antialiased voxelization in 3D space. The voxel texture model is view-independent and hence it is computed off-line once.

Volume Terrain Rendering: In texture mapped volume ray-casting, for each pixel a ray is cast into the scene to obtain the texture of the voxel it hits first. The RGB texture is used in the shading computation to obtain the pixel color. The use of the 3D texture model eliminates the need for an interpolation computation of voxel texture from original 2D texture on the fly and therefore is a desirable feature for real-time rendering.

Embedded Objects: An attractive feature of our method is to have embedded voxelized objects in the terrain scene. This includes pre-voxelized, pre-textured and pre-antialiased geometric objects, such as houses, trees, and vehicles. It is especially suited for the incorporation of amorphous objects in the scene such as cloud, fog and fire.

Real-time Rendering: The texture modeling and ray-casting operation has been accelerated using a 16-processor SGI Challenge R10000 IR system. Preliminary performance indicates an improvement of at least a factor of 12 on 16 CPUs. An image of size 300x300 pixels can be rendered at the rate of 2 frames per second. Our goal is to achieve real-time performance for a 1000x1000 image. To achieve real-time rendering, work is in progress to perform a tiling of the terrain into rectangular regions and to incorporate level of detail models of terrain and objects.

Implementation and Results: The above scheme for volume texture modeling and rendering has been implemented and several sample terrains have been rendered to validate its utility. Fig. 1. shows the block diagram of our prototype system. The database consists of voxelized terrain and voxelized geometric objects. A scene graph represents the terrain and position and orientation of geometric objects on the terrain. The rendering pipeline performs the view frustum culling, ray-casting and display of images. The images presented here are generated using a colume visualization system, which has been expanded to support 3D texture mapped ray-casting. Fig. 2 shows the volume rendering of a terrain using a parallel ray-casting. Fig. 3-4 show the perspective ray-cast from two different view positions. Fig. 5 shows another voxelized terrain of a beach in the SIGGRAPH'97 Los Angeles site. Fig. 6 shows voxelized geometric objects (i.e., two marble benches) embedded in the beach terrain. The memory requirement for the terrain of Figs. 2-3 was 65 MB and 84 MB for terrain of Figs. 4-5.

References

[1] D. Cohen, E. Rich, U. Lerner, and V. Shenkar. A real-time photo-realistic visual flythrough. *IEEE TVCG*, 2(3):255–265, Sep 1996.

[2] A. E. Kaufman, D. Cohen, and R. Yagel. Volume graphics. *Computer*, 26(7):51–64, Jul 1993.

[3] J. R. Wright and J. C. L. Hsieh. A voxel-based, forward projection algorithm for rendering surface and volumetric data. *Visualization '92*, pages 340–348, Oct 1992.

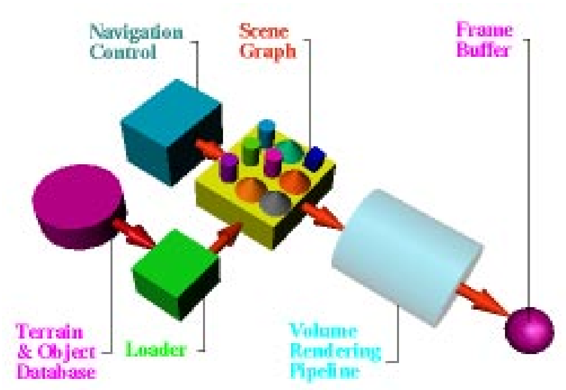

Fig 1: *Block diagram of Visual Flythrough*


Fig 2: *Parallel ray-casting of voxelized terrain*

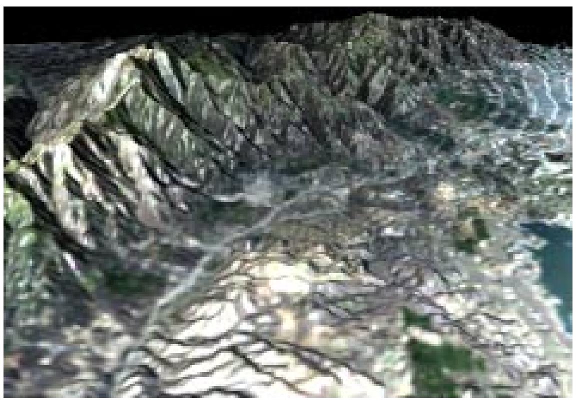
Fig 3: *Perspective ray-casting of voxelized terrain*

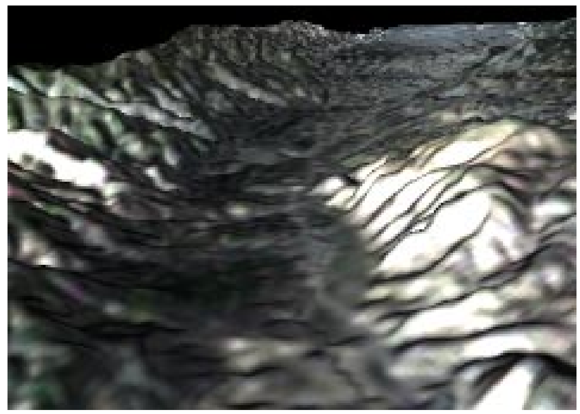
Fig 4: *A close view of the terrain*

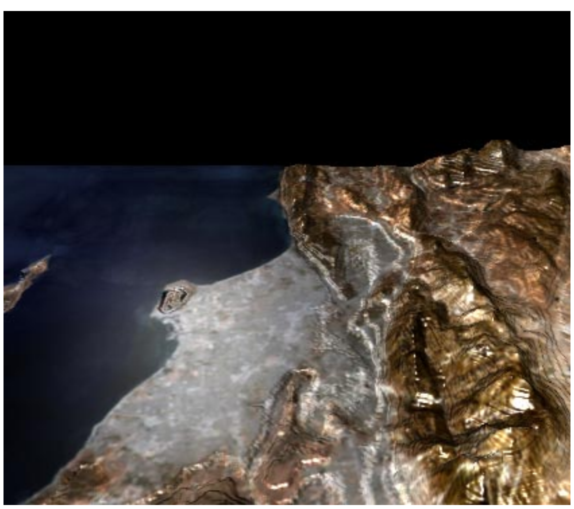
Fig 5: *Los Angeles beach - Siggraph'97 site*


Fig 6: *Los Angeles beach with voxelized benches*