Real-Time Voxelization for complex polygonal models

Team 21
Lakshay Bansal

INTRODUCTION

“Voxel” is short for “volume pixle”. Just like pixels together form a 2D image, a voxel is smallest entity occuping 3d volume space. Voxels, short for "volumetric pixels," are fundamental elements in three-dimensional digital space used in various fields, notably computer graphics, medical imaging, and scientific simulations. Unlike traditional 2D pixels, voxels represent a point in a 3D grid, essentially constituting a small volume element. These building blocks are essential for creating 3D models, visualizing complex data sets, and simulating physical phenomena. Voxels are commonly employed in applications such as 3D rendering, medical CT scans, and video games, enabling the accurate representation of objects and environments in a volumetric space, contributing to immersive virtual experiences and advanced scientific research.

LITERATURE REVIEW

The voxelization algorithm consists of three stages as follows.

• **Rasterization** The triangles are rasterized to the discrete voxel space.

• **Texelization** Each voxel is encoded and accumulated in some directional sheet buffer.

• **Synthesis** Three sheet buffers are transcoded to the worksheet representing the final volume.

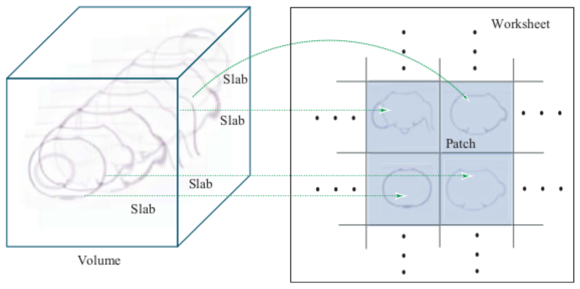


Figure 2. The worksheet versus the slabs of the volume.

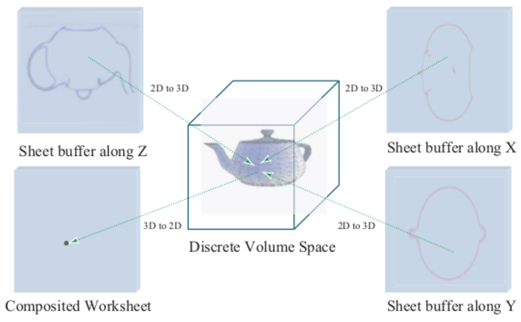
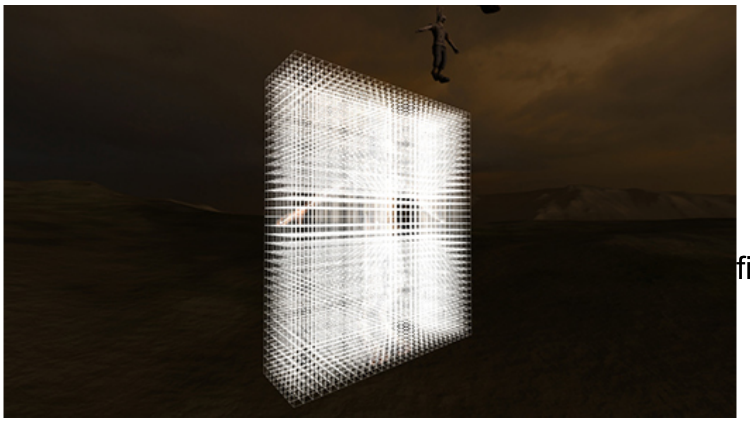


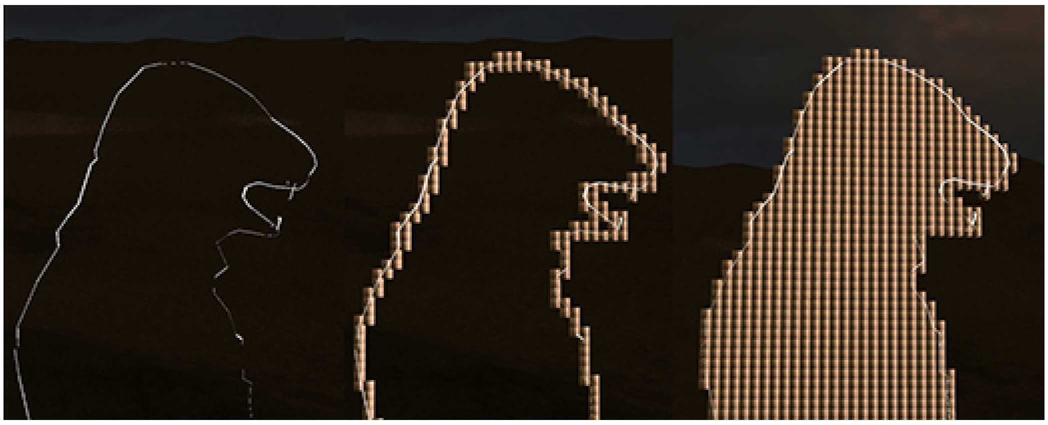
Figure 3. Synthesis of three directional sheet buffers.

Real-Time Voxelization for complex polygonal models Zhao Dong.

The dimensions of the grid have to be big enough to fit the entire model, but no bigger (to avoid inefficiency). This is fairly simple -- we can just set the dimensions equal to the size of the bounding box of the surface triangles (the smallest box that encloses every point) rounded up to the nearest centimeter. Here's a picture of the voxel grid:



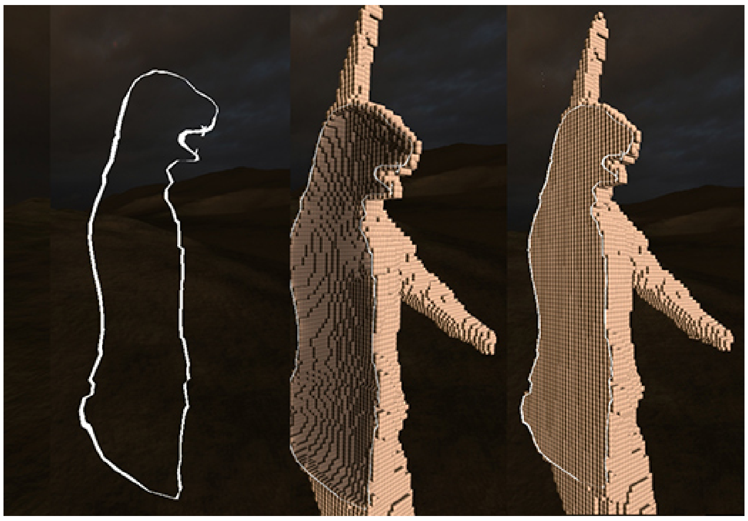
Now we have to decide which voxels are solid (intersecting the model), and which voxels are not. The method I used is a two-step process. First, we solidify a shell representing the surface, and second, we fill it in using a scanline fill algorithm. This is most clearly illustrated by looking at a single slice -- first we have just the triangle surface, next the voxel shell, and finally the filled voxel model.



ScanLine:

Calculating the shell is pretty straightforward. For every triangle, I check every voxel in the triangle's bounding box to see if it intersects. If it does, the voxel is made solid. In order to have a solid result though we will have to implement scanline (pen pick to drop) technique to obtain result as follow.

Wolffire



MILESTONES

Mid Evaluation:

1) Resteriztion
2) Texelization

End Evaluation

3) Synthesis
4) Scanline (Solid Objects)

REFERENCES

[1] Stefan Roettger. 2012. The Volume Library. http://schorsch.efi.fh-nuernberg.de/data/volume/
[2] http://blog.wolfire.com/2009/11/Triangle-mesh-voxelization