

3D Volume Rendering Engine for Interactive Applications

Report Name	Outline Project Specification
Author (User Id)	Hristoz Stefnov Stefanov (hzs9)
Supervisor	Bernie Tiddeman (bpt)
Degree Scheme	G451 Computer Graphics, Vision And Games (Inc Integrated Industrial And Professional Training)
Module	CS39440
Date	October 22, 2012
Revision	0.1
Status	Release

1 Project description

Volume rendering in computer graphics refers to a number of techniques for rendering 3D objects described in terms of their substance rather than their outer surfaces, which is how most modern polygon based technology works. Volume graphics, analogous to 2D raster graphics, use primitive color elements called “voxels” (derived from “volume”+“pixel”), which are arranged in a 3D Cartesian grid to compose the 3D images.

Volume graphics are a more natural way to describe real objects as they do not suffer from penalties when rendering complex shapes over flat surfaces and also make it easier to generate and modify 3D data. Voxels display transparent substance much more accurately as they can accumulate the color in depth. They find their application in medical visualization, the filming industry and 3D graphics packages. However voxels also have several big disadvantages over polygons - the amount of memory they require to store scenes at native resolutions on modern machines could easily reach gigabytes; the popularity of their polygon counterpart has pushed the hardware development in a different direction, so there is little innate support for voxels and animating volume data is not as easy as animating polygons.

There has been a lot of research on different aspects of volumetric graphics tackling these issues and there is a number of open source projects that use the technology, so voxels are not considered to be a new thing, but recent innovations in hardware has made it possible to program the GPU directly, as a result voxel engines starting to gain more popularity and will soon be the next big thing in the gaming industry.

For my project I have as a goal to implement an open source SDK for storing, manipulating and rendering 3D voxel data at interactive frame rate (30fps) that is suitable for use in games software. It will initially be based on OpenGL and will be for the Windows OS platform (but not directly reliant on either).

2 Work to be tackled

I will design and implement an object-oriented C++ SDK for use in games software. This includes tackling the problem of efficient voxel data representation and researching the various techniques that are currently used — octrees and other sparse structures; instancing; data compression; batching and fetching used data only; different file formats support. Rendering volumes brings another broad category of issues — deciding on a rendering technique: ray-tracing vs volume slicing vs volume splatting or if even better allow for multiple supported techniques; hybrid voxel and polygon graphics rendering; lighting, shading, reflection and other light effects; anti-aliasing and voxel interpolation.

Most of my work will probably be based on existing projects and ideas. I have not found a project which I am completely happy to use as a base for my project so I will most probably start from scratch and build up implementing different features based on the time left. I do not (yet) plan to employ hardware acceleration techniques for the scope of this project but my ultimate goal is to design a system that is reliable and works, has the minimal set of features required and is readable and flexible enough to be used as a playground by other developers and researchers.

3 Project deliverables

SDK — sources and binaries (DLLs, LIBs) of the rendering engine

Test Suite — Unit tests covering most of the code.

Demos — At least one executable which runs using the SDK and demonstrates its capabilities.

Documentation — API reference, installation and usage manual, final year report.

4 Initial bibliography

A short non-extensive bibliography on the resources I have taken a closer look at and find most relevant.

- [1] Cyril Crassin. *GigaVoxels: A Voxel-Based Rendering Pipeline For Efficient Exploration Of Large And Detailed Scenes*. PhD thesis, UNIVERSITE DE GRENOBLE, July 2011.
- [2] Randima Fernando. *GPU Gems*, chapter 39 "Volume Rendering Techniques". Addison-Wesley Professional, March 2004.
- [3] Sony Pictures Imageworks. Field3d project. <http://opensource.imageworks.com/?p=field3d>, 2009. Open-source voxel data library. Last accessed October 2012.
- [4] J. Krüger and R. Westermann. Acceleration techniques for GPU-based volume rendering. In *Visualization, 2003. VIS 2003. IEEE*, pages 287 – 292, October 2003.
- [5] David Williams Mat Williams et al. Polyvox project. <http://www.volumesoffun.com/polyvox-about/>, 2006. Open-source polygon based voxel engine, online publications. Last accessed October 2012.
- [6] Ken Silverman. Voxlap project. <http://advsys.net/ken/voxlap.htm>, May 2000. Open-source voxel ray-tracing rendering engine. Last accessed October 2012.
- [7] Codermind Team. Codermind Technical Articles. <http://www.codermind.com/articles/Technical-articles.html>. Articles on ray tracing and volumetric terrain rendering. Last accessed October 2012.
- [8] Various. Volreen project. <http://www.voreen.org/>, 2005. Open-source volume rendering engine, online publications. Last accessed October 2012.