

Christian Noon
Ph.D.
Human Computer Interaction and Computer Engineering
Major Professor: Eliot Winer

A Volume Rendering Engine for Desktops, Laptops, Mobile Devices and Immersive Virtual Reality Systems using GPU-Based Volume Raycasting

Abstract:

Volume rendering is the process of visualizing characteristics and properties of three-dimensional (3D) volume data as a 3D object. The most extensive use of volume rendering takes place within the medical field. Physicians are using a combination of medical imaging technologies and volume rendering techniques to non-invasively examine patients to make critical medical decisions and diagnoses such as finding tumors, searching for blood clots and monitoring unborn fetuses. As the technological computing power continues to increase at a rapid rate, so do the opportunities to provide volume rendering solutions on new and innovative platforms such as mobile devices and immersive clustered environments. This dissertation presents a new volume rendering engine for visualizing volumetric data on multiple platforms. Three different sandbox applications were developed to investigate the challenges and architectural requirements in encapsulating the platform specific volume rendering logic inside the engine to abstract the complexity from the application level. The development of the sandbox applications resulted in the completion of the Volume Image Processing and Rendering Engine, also known as VIPRE.

To encapsulate the platform specific implementation inside the engine, several open source application programming interfaces (APIs) were identified as worthy candidates to support the engine's volume rendering core. OpenSceneGraph (OSG) is an open source, cross-platform graphics toolkit that supports high performance rendering through components critical to the volume rendering pipeline. The DICOM Toolkit (DCMTK) is a collection of libraries and applications implementing a large majority of the DICOM standard capable of examining, constructing and converting DICOM image files. Finally, VR Juggler is a cross-platform, open source virtual reality software development environment designed specifically for creating and executing immersive applications. With native OSG support, application data serialization, display and device abstraction and cluster node swap barriers, VR Juggler was an ideal API for ensuring adequate performance in cluster configurations.

With the architectural design in place, three sandbox applications were developed to investigate platform specific challenges and opportunities. The desktop application was developed to create the core volume rendering algorithms for the engine such as resampling, coloring, shading and compositing. The development also produced several unique contributions including real-time windowing, a GPU compositing algorithm supported by all generic graphics cards and a convex clipping plane algorithm that supports an unlimited number of clipping planes. The immersive sandbox application was built on top of the same volume rendering core designed in the desktop application. With no modifications, the volume rendering core was successfully implemented into the immersive application resulting in the first GPU-based volume raycasting solution for

immersive clustered environments. The mobile sandbox application investigation proved that despite the improved computational power of mobile devices, they are still not powerful enough to support raycasting due to the lack of 3D texture support. However, mobile devices are now fully capable of supporting orthogonal texture slicing. The development of orthogonal texture slicing required the invention of several performance enhancing features including dynamic modification of the render resolutions, an incremental render loop, a shader-based clipping algorithm to support OpenGL ES 2.0, and an internal backface culling algorithm for properly sorting rendered geometry with alpha blending.

The development of the sandbox applications proved that the encapsulation of platform specific volume rendering logic was possible with the designed architecture. This resulted in the development of VIPRE, a unified solution for performing volume rendering on multiple platforms. VIPRE contains many common volume rendering features such as multiple render modes, color and opacity transfer functions and trilinear interpolation. It also contains many more advanced features including real-time windowing, custom CPU and GPU clipping algorithms, accurate depth sorting, dynamic render quality modification, early ray termination and empty space skipping, Phong illumination and multi-pass rendering for backface depth rasterization. VIPRE is going to be released with many example applications and robust documentation to help lower the barrier to entry for novice developers. It is also going to be released under licensing terms to allow it to be used in both academic and commercial communities.

Future work of VIPRE includes extending the compositing algorithm to support the insertion of surgical instruments into the volume for surgical planning. Additionally, the integration of segmentation routines would allow new methods of interaction for segmentation routine training to be studied for different platforms. VIPRE will also be extended to support multiple volumes and independent clipping for visualizing segmented data. A final area of optimization would include reusing previous rendered textures to lazily render the volume while interacting with the user interface in immersive environments.