Cross-Platform Presentation of Interactive Volumetric Imagery

Nicholas Polys*
Andrew D. Wood†
Patrick Shinpaugh‡
Virginia Tech

Abstract

Volume data is useful across many disciplines, not just medicine. Thus, it is very important that researchers have a simple and lightweight method of sharing and reproducing such volumetric data. In this paper, we explore some of the challenges associated with volume rendering, both from a classical sense and from the context of Web3D technologies. We describe and evaluate the proposed X3D Volume Rendering Component and its associated styles for their suitability in the visualization of several types of image data. Additionally, we examine the ability for a minimal X3D node set to capture provenance and semantic information from outside ontologies in metadata and integrate it with the scene graph.

Keywords: volume rendering, standards, Extensible 3D (X3D), DICOM

1 Introduction

Volume rendering is a well-researched and powerful tool for visualizing information that would be difficult to present using only conventional 3D techniques, such as polygonal meshes and point sets. Volume rendering allows the presentation of multiple overlapping, interdependent structures within a dataset simultaneously. There are many different techniques with which to render a volumetric data set, each able to tease out and highlight different information depending on the areas of interest and the distribution of values in the volume. For enterprise and regulatory use however, volume rendering suffers from crucial limitations, specifically the reproduction of volume rendering visualizations across platforms and vendor tools.

While individuals can create impressive and enlightening visualizations of volumetric data, the process for colleagues and collaborators to recreate these presentations can be complicated and depends on many different factors, such as work domain, platform, and specific software. As reproducibility is one of the central tenants of respectable science, researchers need a way to share their process which is simple and exact. So far, there has been work towards standardizing volume data formats, but this effort has been almost exclusively in realms of medical data and only on the interchange level. But volumetric information is useful for many different domains from confocal microscopy in Cellular Biology, to Paleontologists' micro-fossil scans, to non-invasive scans of structures such as bridges or carry-on and checked baggage at airports. While the fundamental data type is the same, there remains an conspicuous gap in reproducible volume rendering.

*e-mail: npolys@vt.edu
†e-mail: andywood@vt.edu
‡e-mail: shpatric@vt.edu

2 Background

2.1 Volume Rendering

There has been much research into volume rendering techniques since the emergence of the field, most of which is outside of the scope of this paper. For a general survey of volume rendering techniques, see Kaufman and Mueller [2004] and for a perceptual evaluation approach, see Boucheny et al. [2009].

There has been less work in the past decade concerning volume rendering within the context of Web3D technologies. Behr and Alexa [2001] proposed a volume rendering component for VRML based around 2D and 3D textures, but this work is one of the few general-purpose efforts.

Other research into the intersection of Web3D and volume rendering lies neatly in the realms of healthcare. Web3D technologies, particularly X3D along with a volume rendering component, have been used mostly in anatomical training such as [Brenton et al. 2007; John 2007]. One specific training system for endoscopic procedures [Jung et al. 2008] uses extended X3D and haptic feedback afforded by custom equipment and the H3D renderer (www.h3dapi.org) to create a high fidelity of interaction and visualization.

2.2 X3D

Extensible 3D Graphics (X3D), with its modular Profiles, simple XML-based syntax, and cross-platform tool support is already positioned to fill the niche of cross-discipline portability for static and interactive 3D graphics data. Recently, a new volume rendering component for X3D has been published which could offer the same value for volumetric data. The component includes a number of different rendering styles that provide different transfer fucntions and parameters for rendering volumetric data; these styles may also be combined in interesting ways to visualize information across domains.

In addition to describing and displaying volumetric data, X3D also provides meta-data support at a fundamental level: every node in the scene can have metadata of any type associated with it. This metadata can be referenced to specific knowledge bases, allowing the X3D graphics runtime to include or link semantic information to the scene. Specific target knowledge bases for the medical domain are the FMA (http://sig.biostr.washington.edu/projects/fm/) and SNOMED (http://www.snomed.org), but the concept is certainly not limited to medical ontologies. This integration of semantics with the X3D scene graph, along with the proposed volume rendering styles and existing event/trigger framework, allows X3D to serve as a simple and consistent medium for effectively communicating and reproducing meaningful 3D data, both for simple interchange or for a fully interactive, multidimensional presentation.

2.3 Med3D Working Group

The X3D Medical Working Group (MWG) of the X3D Consortium has been developing a medical imaging profile (MedX3D) and a

Volume Rendering Component (VRC) for X3D. Originally funded by the National Library of Medicine and the Army's Telemedicine and Advanced Technology Research Center (TATRC.org), this specification address the needs of the medical community for reproducible 3D visualization of medical images over the network. The VRC to X3D was created after a thorough search of volume rendering styles in the literature was undertaken. It was decided that the focus of the specification was to add the most common and currently used rendering styles, including: boundary enhancement, cartoon, composite, edge enhancement, isosurface, opacity map, silhouette enhancement and the Gooch shading model of two-toned warm/cool coloring. See the TATRC grant report for more details [John et al. 2008; Aratow et al. 2007].

There was a priority placed on simplicity, both to benefit people making an implementation of the specification and for the user making content using X3D. Although a limited number of rendering nodes have been initially declared, the VRC can evolve to include more advanced/general rendering methods that can be added later at a higher support level. The project included the specification and implementation of the necessary components to support rich X3D volume rendering, which are now evolved into the Texture3D and Volume Rendering Components forthcoming in X3D 3.3. As vertical markets mature, Volume Interchange or Volume Interactive Profiles (suites of nodes supporting a functionality) may be added to the ISO specification.

2.4 n-Dimensional Presentation

Since 2008, the Web3D Consortium and DICOM standards bodies have been working together to align stakeholders and requirements for a Work Item scoped toward the specification of DICOM n-Dimensional Presentation. The work item for n-Dimensional Presentation States is born out of the need for consistent presentation of volumes, surfaces, animations and annotations across the healthcare enterprise. From the most advanced medical centers to combat doctors in the field, to small clinics, caregivers and patients, many individuals need to be able to re-create and view medical image data. With a significant amount of 3D and 4D information residing in DICOM files, the need for interoperable access to these as interactive renderings is crucial.

For example, as WG-17 (3D) was specifying the DICOM 3D data storage model, Web3D participated and insured compatibility. For example, WG 17: SUP 132 Surface Segmentation (geometry SOP) — geometry data structures map directly to X3D vertex and index data structures (e.g. triangle strips, indexed face sets). Also relevant is WG 11: SUP 120 (extended presentation states), which includes 2D graphical objects such as styled text, rulers, axes, arrows, crosshairs etc., which can be implemented within the current X3D nodeset and easily with the PROTO mechanism. Meetings at RSNA, Web3D and SPIE have continued to advance this standardization effort.

3 Challenges

There are many challenges to cross-platform, reproducible volume rendering. In this section we explore these challenges, namely: Representation, Implementation, Scalability, Interaction and Integration.

3.1 Representation

In the world of volume rendering, there is a plethora of tools and techniques to choose from to produce a visualization. This variety is, however, a two-edged sword. Where options flourish, there is a lack of standardization. Each of the many different modalities of volumetric information often has its own file format, courtesy of the proprietary scanning device. For example, Magnetic Resonance (MR) data alone may exist in any number of proprietary formats. To make matters worse, there is no standard set of supported formats across renderers. The DICOM format is a notable exception and has made strides to standardize an interchange format for health-related data, but is largely unhelpful to those outside of medicine. If a user wishes to share their data with a colleague from a different discipline (or a visualization center), it falls to one of the parties to figure out how to coordinate file formats and renderers—often a challenging task.

3.2 Implementation

Key challenges to reproducible volume presentation over the web include both perceptual and technical considerations. The rendered product of an engine can be confounded by different application requirements, programmer conventions, and graphics libraries. Depending on the degree and nature of these differences, the rendered images may present subtly different perceptual cues, jeopardizing the reproducibility of an interpretation or conclusion. Thus, the role of conformance tools for implementation efforts cannot be underestimated.

In order to reproduce a volume rendering consistently across several platforms, we seek to identify a 'greatest common denominator' set of functionalities. The ISO scene graph specified in VRML and X3D provides well-defined data structures and semantics for polygonal rendering of complex, realtime scenes including lights, materials, animations and behaviors. For the new X3D Volume rendering styles, common requirements were derived from a broad survey of techniques in use by industry and validated by experts.

3.3 Scalability

Efficiency of rendering algorithm implementations also varies widely and volume rendering tends to be computationally heavy. Even with the state-of-the-art algorithms, we are faced with the classic tradeoff between high-fidelity models and interactive models. Thus, we must consider the issue of scalability. In increasingly large data sets, it becomes difficult to use commodity or thin clients to render big data, especially at interactive rates. Therefore, there is a trend to server-side (in situ) rendering of large datasets at the high-performance computing data center where there is fast I/O, powerful CPUs and GPUs, and large quantities of fast RAM. As the rendering of large data moves to the server, image buffers are served or streamed to clients and may be compressed using a number of video compression codecs; in addition, user interactions on the client machine must have real time effect on the rendering (such as navigation, selection and manipulation).

3.4 Interaction

Yet another challenge for volume data presentation concerns interactivity and usability from a 3D User Interface (3DUI) perspective. There are several important aspects to this challenge. Generally these aspects can be categorized as: navigation, selection, manipulation, and system control [Bowman et al. 2004]. In basic volume rendering, navigation is mostly limited to simple orbital rotation and possibly zooming; pre-defined camera trajectories for animation are common. In many cases, users need to be able to select sub-components or segmentations of a volume and dynamically alter the render properties of that segment, perhaps through menus and buttons. Finally, many programs offer various tools to examine the interior of a volume by manipulating sliders or cutting planes/shapes to clip the volume.

From clinical and other use cases, common volume presentations include meshes, appearances, textures, text, animation and lights. We seek to organize and render objects and data in the scene with groupings such as Billboard and Switch. To reproduce an interactive scenario, such as an eBook of anatomy, we would also need to instantiate widgets such as buttons and sliders and have access to environmental, point and drag sensor events. In addition, there are exciting applications pushing interaction into new modalities such as the area of haptic rendering for telemedicine and surgical training [Vidal et al. 2009; Kurmos et al. 2010].

Looming large are further challenges in 3DUI for volume rendering applications. For example, Curved Planar Reformation [Kanitsar et al. 2002] provides a technique to project or flatten winding 3D structures such as vasculature in order to extract a linear measurement from it (e.g. the length of a stent). Such an application provides the user a way to define a 3D curve along which the volume is projected. In this way, curved structures like vasculature and the spine can be measured accurately. Working with practitioners and technicians, the Web3D Medical Working Group is gathering the required parameters designing an MPR node.

3.5 Informatics Integration

A key challenge to the efficiency of both the clinical and research enterprise is the integration of multiple data sources and records. For example, there is no consistent practice for cross-referencing 3D spatial structures, features or segments such as anatomy with patient databases or clinical procedure codes. In addition, new forms of knowledge representation such as ontologies and the semantic web can provide richer machine reasoning, but are not widely explored or adopted in interactive 3D graphics.

Broad integration of this kind is still rare principally because of the challenge in harmonizing the data structures of knowledge schemas and ontologies with the data structures of interactive 3D worlds, a.k.a. scene graphs. We have demonstrated X3D worlds that could be enhanced with semantic information include: integrating a medical volume with the FMA (http://sig.biostr.washington.edu/projects/fm/) and SNOMED (http://www.snomed.org) vocabularies and integrating a geologic or seismic profilereading with the SEDRIS Environmental Data Coding Specification (EDCS) vocabulary (http://www.sedris.org/edcs.htm).

4 Solutions

At the time of this writing, the new volume rendering component of X3D is under review by the ISO as part of X3D 3.3. This component adds support for a variety of techniques and rendering styles, and seeks to solve many of the challenges described in the preceding section. Additionally, X3D already offers infrastructure to address some of the interactivity concerns, and can be further extended to incorporate other information models and services.

The X3D 3.3 Volume Component specification is already supported by popular X3D engines. InstantReality (instantreality.org) is known for its industrial-strength rendering system, powerful extensions and cross-platform support. The open-source Haptics 3D (H3D) from SenseGraphics (www.h3dapi.org) offers extensive volume rendering features in addition to support for haptic devices. Because of the additional render style support, we chose to use H3D to provide all of the following figures.

In addition to exploring these new features of X3D for delivery, we also explore cluster rendering as a solution to the scalability problem of volume rendering.

4.1 Data Representations

While MedX3D offers many options and styles for rendering volumes, we first have to get our target volume data into a format that is readable by H3D. Our datasets are originally available in zipped RAW, NRRD, and PNG stack formats, courtesy of Volvis.org and the Web3D working group. Since H3D has strong built-in support for NRRD (nearly-raw raster data), we first convert each of the datasets to this format through the simple command-line utility ‘unu make,’ provided by the UNRRDU utilities of Teem (http://teem.sourceforge.net). Although UNRRDU does support image stacks, we instead chose to use the free image editor ImageJ (http://rsbweb.nih.gov/ij/) to convert the stack to a RAW file before running unu.

4.2 Rendering

In this section we take a brief look at some of the new volume rendering styles available from X3D, along with the tags that formalize such parameters. For all of the following examples, we are using a simple scene graph with a minimal number of nodes defined, with one scene and group tag. The following VolumeData snippet is used in all of the examples, where we merely change out the ‘VolumeRenderStyle’ with a specific tag and use the appropriate dataset. To achieve our results, we also defined a scaling transform and specific viewpoint.

```
<VolumeData dimensions=’1.28 1.28 1.28’
  useStochasticJittering=’true’ >

  <!-- VolumeRenderStyle -->

<ImageTexture3D containerField="voxels"
  url=’"dataset.nrrd"’ >

  <TextureProperties boundaryModeR=
  ’CLAMP_TO_EDGE’ boundaryModeS=
  ’CLAMP_TO_EDGE’ boundaryModeT=
  ’CLAMP_TO_EDGE’ magnificationFilter=
  ’AVG_PIXEL’ minificationFilter=
  ’AVG_PIXEL’  />

  </ImageTexture3D>

</VolumeData>
```

4.2.1 Projection Styles

The first of several categories of rendering style we examine are the projection styles, which are variants of the <ProjectionVolumeStyle/> tag, shown in Figure 4 compared to the default OpacityMap style. Essentially this method works by casting rays from the camera into the volume. If the type attribute is set to MAX, only largest intensity values for each ray are returned (Maximum Intensity Projection—MIP) and rendered (Figure 1b). The algorithm performs similarly for type="MIN" and type="AVERAGE". An additional parameter, intensityThreshold, can be used with MIP to return the first voxel the ray intersects which crosses the given threshold. To obtain Figure 1c, we used the tag:

```
<ProjectionVolumeStyle type="MAX"
intensityThreshold="0.25" />
```

(a) Opacity Mapped (Default)

(b) Max Intensity Projection

(c) Thresholded Projection (LMIP)

Figure 1: CT scan of a backpack using different projection rendering styles

4.2.2 Enhancement Styles

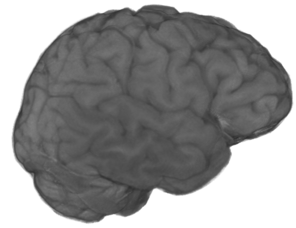
A second category of rendering styles work by computing the surface normals of the volume's voxels, and applying different lighting techniques to emphasize features of the data. Edge (Figure 2b), boundary, and silhouette styles all color voxels based on how close to orthogonal their normal vector comes to the viewpoints'. They are all similar in syntax, for example: `<EdgeEnhancementVolumeStyle />`. The cartoon style (Figure 2c) behaves similarly, but with a user specified number of color steps. The tone mapped (Figure 2d) style works similarly to the default opacity map, but maps colors to intensities instead of opacity values. Finally, the shaded style is a little more complicated, and allows the user to supply custom materials, lighting, and shadows. The code to produce Figure 2e is as follows:

```
<ShadedVolumeStyle lighting="TRUE"
   shadows="TRUE">
  <Material diffuseColor='0 .5 1'
   specularColor='1 1 1'
   ambientIntensity='.4'/>
</ShadedVolumeStyle>
```

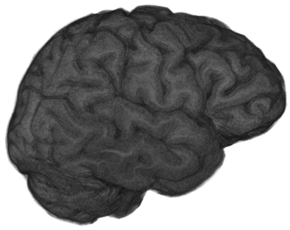
4.2.3 IsoSurface Volumes

A common technique for displaying volumes is to extract isosurfaces. X3D provides this capability through the IsoSurfaceVolumeData node (as opposed to the VolumeData node). The volume can then be rendered with any of the normal enhancement styles discussed above. The code to render Figure 4.2.3 was rendered with the following code replacing the VolumeDataNode.

```
<ISOSurfaceVolumeData surfaceValues=".15"
   dimensions="1.28 1.28 1.28">
  <CartoonVolumeStyle colorSteps="32"/>
  <ImageTexture3D containerField="voxels"
   url=""./Datasets/skull.nrrd""/>
```

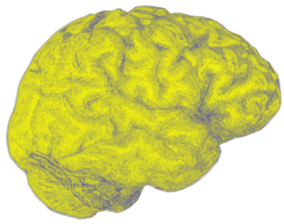


(a) Opacity Mapped (Default)

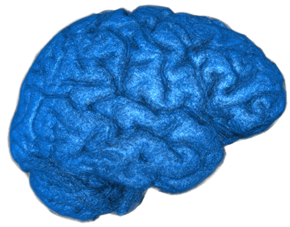


(b) Edge Enhanced

(c) Cartoon Style



(d) Tone Mapped



(e) Shaded Style

Figure 2: MRI Brain Data presented using several different enhancement rendering styles

```
</ISOSurfaceVolumeData>
```

4.2.4 Combining Styles and Volumes

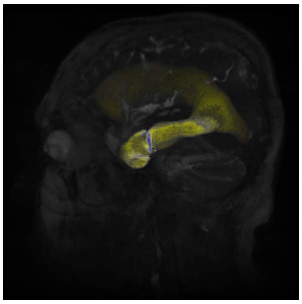
In addition to the individual rendering styles and volumes, X3D also offers a variety of ways to combine styles and volumes to provide a more customizable and distinctive rendering. These techniques include composing styles on an individual volume, selectively choosing styles to render segments of a volume, and combining separate volumes (blending).

Composing Styles Certain enhancement styles are 'composable' and can be combined together with the ComposedVolumeStyle node to gain the benefits of each. This style can be applied anywhere any of the other enhancement styles can, including within blended and isosurface volumes. Figure 4c is the result of combining and silhouette enhancement styles:

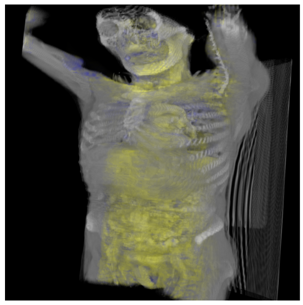
```
<ComposedVolumeStyle>
  <SilhouetteEnhancementVolumeStyle
   silhouetteBoundaryOpacity="1"
   silhouetteRetainedOpacity=".1"
   silhouetteSharpness="10"/>
  <EdgeEnhancementVolumeStyle
   gradientThreshold=".8" edgeColor=".5 0 0"/>
</ComposedVolumeStyle>
```

Segmentation In order to make sub-components of a volume stand out, a segmentation file can be supplied and a different style applied to each segment. Figure 4a was rendered with two ImageTexture3D

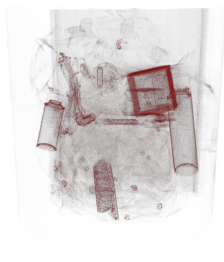
Figure 3: Isosurface rendering, using CartoonVolumeStyle



(a) Segmented Volumes



(b) Blended Volumes



(c) Composed Styles

Figure 4: Several methods of combining styles and volumes

nodes within a SegmentedData node, one for the segmentation data, one for the volume. Different styles are then simply placed sequentially, at the same level as the texture nodes. The following node replaces the VolumeData node:

```
<SegmentedVolumeData
   dimensions="2.304 2.304 1.116">
  <ImageTexture3D
   containerField="segmentIdentifiers"
   url=""mri_ventricles_segment.nrrd""/>
  <ImageTexture3D containerField="voxels"
   url=""mri_ventricles.nrrd""/>
  <OpacityMapVolumeStyle/>
  <ToneMappedVolumeStyle/>
</SegmentedVolumeData>
```

Blending In addition to selectively rendering components of a single volume, two separate volumes may be blended together (with separate styles) using the BlendedVolumeData node. Figure 4b was rendered with the following code inside the VolumeData node:

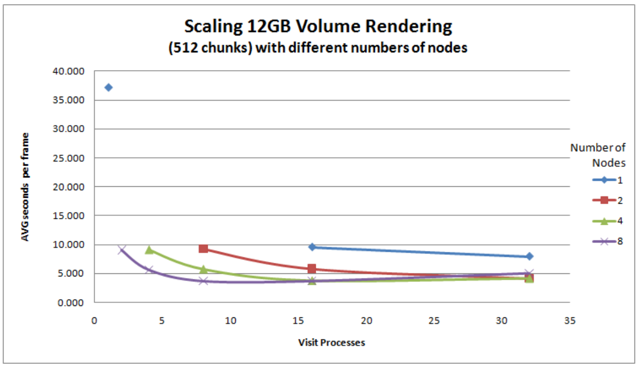
```
<BlendedVolumeStyle weightConstant1="0.51">
  <ToneMappedVolumeStyle/>
  <ImageTexture3D containerField="voxels"
   url=""internals.nrrd""/>
  </BlendedVolumeStyle>
  <ImageTexture3D containerField="voxels"
   url=""body.nrrd""/>
</BlendedVolumeStyle>
```

4.3 Scalability and Cluster Rendering

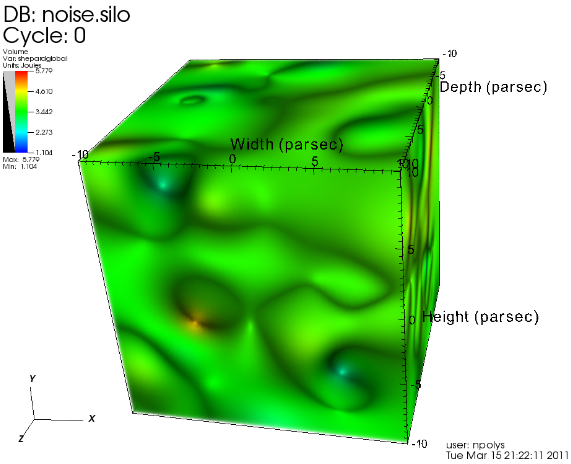
To address the scalability challenges of volumes We explore cluster rendering in the server-side using an open source visualization package called Visit (vers. 1.11.1). We ran a set of rendering tests on Athena, our data analytic cluster. The Athena cluster consists of 42 nodes, each with 32 cores. 64 GB RAM, and Infiniband network connections. Athena also includes 8 nVidia Telsa S2050 servers with Fermi based GPUs with 2 GPUs attached to each of 16 of the cluster nodes. Examining the strong scaling of the Visit volume rendering algorithms, we divided the work among 1, 2, 4 and 8 nodes with 1, 2, 3, 4, 8, 16, and 32 processes per node. While we tested several render sizes in pixels (800x600), (1024x768) and (1280x1024), here we report only the 1280x1024 image size.

We considered a 12 GigaByte volumetric data set created using the visitconvert utility based on the noise.silo data included with the VisIt package. For this testing, a zone size of 1,073,741,824 and 512 chunks. This resulted in a dataset of 12,000,000,000 bytes divided between 512 files. The test harness was implemented in Python and consisted of loading the data set, applying the default transfer function (rainbow color mapped from scalar variable; e.g. figure visit0000.eps) and rendering 100 frames where the data was rotated through the same heading and pitch for between each frame (through 360 degrees). VisIt provides a "-timings" parameter which sends timing information to various files. When a parallel engine is used, there will be an engine_par timing file for each process; for RayCasting and RayCastingIntegration, the server timings were collected and recorded from the "Timing for NM::Render". Timings (to draw) were averaged across all 100 frames; so this is the average time to draw a frame.

Overall, Raycasting with Integration was slightly faster than Raycasting, but both showed consistent strong scaling trends (see figure scaling.eps for Raycasting Integration performance profile). Rendering on 1 node with 1 process took on average 37.24 seconds per frame while the fastest time was on 8 nodes with 8 or 16 processes each (3.68 seconds) and comparable times for 2 nodes with 32 processes each (4.15 seconds) and 4 nodes with 16 processes each (3.75 seconds) or 32 processes each (4.12 seconds). These results show that the rendering problem of large volume data over the web is reasonably scalable across increased threads or processes ¡and¿ increased nodes available at an HPC system. We hope clustered remote rendering tools such as Visit and Paraview will eventually map the transfer functions of X3DVolumeRenderStyles, extending the broad impact of reproducible science across platforms and applications.



(a)



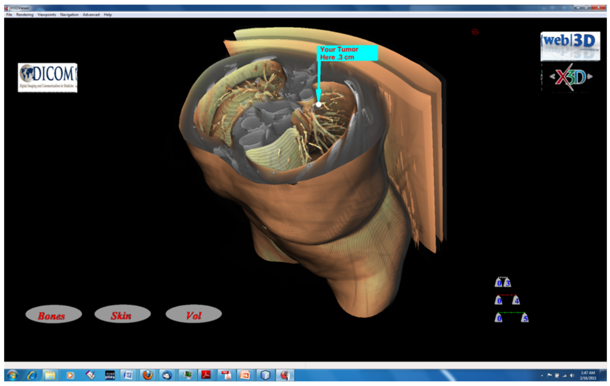
(b)

Figure 5: Performance graph of test rendering runs on the Athena analytic cluster

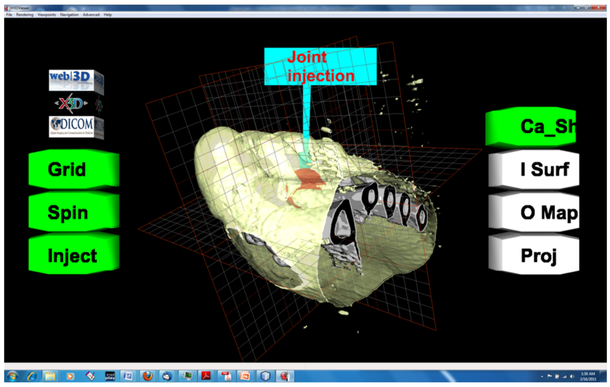
4.4 Interaction

X3D manages complex multi-dimensional information for real-time rendering through two well-known representations, collectively known as a ‘Scene graph’. The X3D specification describes the scene graph as a single-parent, n-child hierarchy of nodes (a directed acyclic graph); this tree structure serves as the ‘transformation graph’, which embodies both spatial and logical relationships among the nodes. Because X3D also has well-defined event types and semantics for event evaluation, we can consider the second representation of a scene a ‘Behavior graph’, which describes the circuitry (or routes) of information among the nodes. While the Volume Rendering Component provides a standard baseline for the interchange of reproducible volume presentations, X3D as a language provides much more.

We implemented two X3D applications that demonstrate interactive level functionality with volume rendering. The first uses a DICOM data set of a torso where bones and skin are surfaced as a polygonal meshes and reside under switches that can be toggled on or off by user selection. This example included perspective and orthogonal Viewpoints, HUD for buttons and logos, Text, Billboard, 2D circles and arrows to mark features as well an animated endoscopy trajectory for a viewpoint (Figure 6a). The second example included the foot dataset with several rendering styles driven by buttons, which implemented mouseOver behavior (Figure 6b, right). In addition, this example included the ability to toggle on or off inline movable



(a)



(b)

Figure 6: Two examples of interactive volumetric applications

grids, a rotation animation, or a procedure annotation (Figure 6b, left). Script nodes were not needed for either of these example applications- Event Utility ROUTES were sufficient.

4.5 Integration

X3D is designed for the networked information ecology and is designed to play well with others. One powerful example is the integration of semantic web technologies with the scene graph. In X3D, each node can carry a metadata set. This metadata could include any number of things including the provenance of a particular segmented surface, the authorship of 2D markup or annotations, and well as semantic information referencing some external knowledge base. Through the US Army TATRC grant, we developed and demonstrated the lossless integration of FMA and SNOMED vocabularies, references, and relationships with the X3D scene graph. Ontology types such as Integers and Booleans map directly to X3D 3.3 Metadata types. However, we had to devise simple rules for attaching metadata terms to Shapes and Groups:

1. MetadataSet nodes refer to their sibling Transform node, where the object’s shape geometry may be specified. A sibling Group node may be instantiated for parts or subdivisions of the referent object. This allows larger containing structures or anatomical systems to be easily accessible programmatically and additional detail accessible when needed

2. The MetadataSet node is instantiated with its source specified as the reference field (e.g. FMA,. SNOMED); its children are typically MetadataString nodes specifying its attributes and its relationships to other entities in the source ontology

3. Unique identifier names of source entities (integers) are prepended with an ‘m’. This allows result data to conform to the Web3D scene graph identifier convention (DEF); to cross-reference corresponding entities in the scene graph or to programmatically access named nodes, one must remove this first character (‘m’) and compare it with a MetadataSet node’s name=”” attribute.

5 Reflection

While X3D 3.3 and its Volume Component may not cover 100 percent of requirements for effective cross-disciplinary communication of volumetric data, it does address many of the problems we introduced in section 3.

5.1 Representation

Data format standardization remains a challenge, and likely will for the foreseeable future. The only suggestion we can make is that the X3D specification enforce compatibility with certain, general datatypes, specifically DICOM and NRRD and RAW. Volume data ultimately boils down to stacks of images and metadata, which can then be converted via free and open-source tools into such standards as DICOM and NRRD.

5.2 Implementation

X3D offers a wide range of rendering styles, each with their strengths and their weaknesses. For instance, the different projection styles in Figure 4 each show different items, which may be harder to see with other rendering methods. The brain enhancements (Figure 2), really bring out the intricate surface detail, but say little of the structure underneath. The segmented volume in Figure 4a partially solves this problem.

While these tools are useful as is, we would suggest a few minor tweaks to the spec. First of all, we appreciate the power of the projection styles, but they could gain a further degree of control if the intensityThreshold attribute were replaced with minIntensityThreshold and maxIntensityThreshold. Thus, they would form a window which could slide around the intensity spectrum. The different projection types would behave as normal, but `MAX` would take the largest values within the range, `MIN` the minimum, etc. Combined with interactive widgets, this would allow us to further explore the internals of a volume on the fly, which is probably the single weakest point of the current spec.

The second change is also minor, but can be a source of major frustration. The default opacity map is not very useful for getting at the interiors of volumes, but with a customized transfer function it could be. The node does accept a texture as an input for the transfer function, but there is not a clearly documented way to generate one within the node short of creating your own image externally.

5.3 Interaction

Bundling Interactive Profile Nodes and Components with volume rendering has demonstrated rich applications across use cases from anatomical education, and informed consent. Event Utilities are remarkably expressive for user interface widgets, animation behaviors and basic logic. One issue that must be addressed is that an X3D ClipPlane (ISO-IEC -19775-1.2 clause 11.4.1) is defined as an inputOutput SFVec4f field with the exposed field of plane, which specifies a four-component plane equation that describes the inside and outside half space. The first three components are a normalized vector describing the direction of the plane’s normal direction. Unfortunately, there are no specified Interpolators or EventUtilities to animate this type, thus no easy way to ROUTE data to this node without a Script.

5.4 Integration

It is clear that the X3D scene graph has rich metadata capabilities. Being able to associate semantic classes and relationships with the graphical objects in a scene graph provides at least two primary integration options: terminology can be embedded in the X3D scene file itself OR use it can use the ontologies ID conventions to reference to an external (URI/URL) store. By adopting simple conventions, several ontologies can be referenced from within an X3D file or live scene graph.

6 Conclusions and Future Work

In this paper, we have shown that X3D volume rendering offers cross platform reproducibility across domains at both the interchange and interactive functional levels. From medical imaging to Paleo-Biology to non-invasive sensing, we have noted how such an ISO specification for interactive presentations meets the key ‘repeatability’ and ‘durability’ requirement of repeatable n-D volume image presentation over the web. The Web3D Consortium works with many other industry groups to insure the harmonization of international standards that are open and royalty-free. X3D’s content model is extensible enough to accommodate the semantics of these domains and we have demonstrated the principles by which multiple ontologies could be integrated with the X3D scene graph.

To address the problem of large datasets, we have shown how the specification can provide value in either client or a server-side/cluster rendering architectures. Data size will continue to be a challenge even if advanced file structures such as HDF5 are widely adopted [Dougherty et al. 2009]. In working with datasets from so many sources and domains, we have found some cases that warrant specific attention. For example in a Projection rendering style, while a threshold may be set from an upstream application or interactively by the user, this alone may not provide enough sensitivity to show features of interest. We suggest the addition of fields to this node to specify essentially an attenuation of the range of intensity mapped (e.g. minThreshold and maxThreshold).

We will continue to test the perceptual and technical scalability of interactive volume rendering applications across multiple platforms and displays- most immediately, across immersive environments with stereo rendering and 6DOF head and wand tracking. Our group will focus on developing and evaluating 3DUI techniques for information and interaction design using experimental displays and devices. For example, we are interested in user perception and performance across tasks where we manipulate variables such as: graphics resolution, screen form factor (size, surround), information layouts, and explore new interaction designs including a range of input and haptic devices and their corresponding techniques.

Even with this greatest (declarative) common denominator for volume rendering styles, there is room for multiple implementations — and innovations on top of — the specification. We believe there is tremendous value in an open source implementation to insure common reference and consistency among engines. Future work should focus on adoption and conformance efforts for the X3D ISO Volume Component and its RenderingStyles. For example adding X3D Volume component support to open source tools such as Paraview and Visit and adding NRRD and RAW to DICOM as the list of supported volume data file formats. Furthering the goals of wide dissemination and broad impact, we speculate there will be opportunity to improve the Volume rendering capabilities of WebGL (OpenGL ES) as exposed through high-level declarative representations such as X3DOM.

7 Acknowledgements

We would like to thank the Web3D Consortium membership and Medical Working Group for their efforts, as well as DICOM for evolving and aligning our standards.

References

ARATOW, M., JOHN, N. A., COUCH, J., EVESTEDT, D., HUDSON, A., POLYS, N., PUK, R., RAY, A., VICTOR, K., WANG, Q., AND GODIL, A. 2007. Tatrc grant report: <www.web3d.org/documents/medical/2008/Web3D_TATRC_FinalReportPublic.pdf>. Web3d Consortium.

BEHR, J., AND ALEXA, M. 2001. Volume visualization in vrml. In *Proceedings of the sixth international conference on 3D Web technology*, ACM, New York, NY, USA, Web3D '01, 23–27.

BOUCHENY, C., BONNEAU, G.-P., DROULEZ, J., THIBAULT, G., AND PLOIX, S. 2009. A perceptive evaluation of volume rendering techniques. *ACM Trans. Appl. Percept. 5* (February), 23:1–23:24.

BOWMAN, D. A., KRUIJFF, E., LAVIOLA, J. J., AND POUPYREV, I. 2004. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.

BRENTON, H., HERNANDEZ, J., BELLO, F., STRUTTON, P., PURKAYASTHA, S., FIRTH, T., AND DARZI, A. 2007. Using multimedia and web3d to enhance anatomy teaching. *Comput. Educ. 49* (August), 32–53.

DOUGHERTY, M. T., FOLK, M. J., ZADOK, E., BERNSTEIN, H. J., BERNSTEIN, F. C., ELICEIRI, K. W., BENGER, W., AND BEST, C. 2009. Unifying biological image formats with hdf5. *Queue 7* (October), 20:20–20:25.

JOHN, N. W., ARATOW, M., COUCH, J., EVESTEDT, D., HUDSON, A. D., POLYS, N., PUK, R. F., RAY, A., VICTOR, K., AND WANG, Q. 2008. Medx3d: standards enabled desktop medical 3d. *Studies In Health Technology And Informatics 132*, 189–194.

JOHN, N. W. 2007. The impact of web3d technologies on medical education and training. *Comput. Educ. 49* (August), 19–31.

JUNG, Y., RECKER, R., OLBRICH, M., AND BOCKHOLT, U. 2008. Using x3d for medical training simulations. In *Proceedings of the 13th international symposium on 3D web technology*, ACM, New York, NY, USA, Web3D '08, 43–51.

KANITSAR, A., FLEISCHMANN, D., WEGENKITTL, R., FELKEL, P., AND GRÖLLER, M. E. 2002. Cpr: curved planar reformation. In *Proceedings of the conference on Visualization '02*, IEEE Computer Society, Washington, DC, USA, VIS '02, 37–44.

KAUFMAN, A., AND MUELLER, K. 2004. *The Visualization Handbook*. Academic Press, ch. 7, 127–174.

KURMOS, L., JOHN, N. W., AND ROBERTS, J. C. 2010. Integration of haptics with web3d using the sai. *Web3D 09 Proceedings of the 14th International Conference on 3D Web Technology 1*, 212, 25–32.

ULLRICH, S., KUHLEN, T., POLYS, F., EVESTEDT, D., ARATOW, M., AND JOHN, N. W. 2011. Quantizing the void : Extending web3d for space-filling haptic meshes. In *Medicine Meets Virtual Reality*, 670–676.

VIDAL, F. P., JOHN, N. W., HEALEY, A. E., AND GOULD, D. A. 2009. Simulation of ultrasound guided needle puncture using patient specific data with 3d textures and volume haptics. *Computer Animation And Virtual Worlds 19*, 2, 111–127.