

# A Fresh Look at Immersive Volume Rendering: Challenges and Capabilities

Nicholas F. Polys\*  
Virginia Polytechnic  
Institute and State  
University

Sebastian Ullrich  
SenseGraphics AB

Daniel Evestedt  
SenseGraphics AB

Andrew D. Wood  
Virginia Polytechnic  
Institute and State  
University

Michael Aratow  
Web3D Consortium



Figure 1: Examples of Immersive Volume Rendering with the open X3D and H3D platform; stereo display of cell data in both segmented and isosurface styles (left) and volume rendering combined with haptic input (right).

## ABSTRACT

In this paper, we begin from the premise that for science to progress, experiments must be repeatable. Similarly, for enterprises in a knowledge-driven economy, such as clinical medicine, the portable and repeatable presentation of visualization views is paramount. Yet both science and capitalism also rely on the innovation of researchers to explore and test new variables and models. We consider how this natural tension plays out with respect to immersive volume rendering and the open standards and open source movement. We first discuss the challenges of volume rendering in immersive environments and then describe how the ISO standard X3D meets these. Then, we provide examples with an implementation of these specifications in the open source toolkit H3D and conclude that extensibility and repeatability are not mutually exclusive. Finally, we reflect on the implications for future research programs in this area.

## 1 INTRODUCTION

The last 15+ years of computer graphics and applications have proven the value of volume rendering techniques for scientific visualization in general and interactive rendering applications across domains: from medical imaging to geophysics to transportation security. There is a healthy community of researchers working on volume rendering models and algorithms; their efforts are largely focused on different approaches to illumination, rendering techniques and the handling of large datasets [7]. However, one issue that is seldom addressed is reproducibility of volume renderings. There are a lot of reasons why reproducibility is hard—from hardware and software architectures to network protocols to the growth of data

types and sizes, it is difficult to re-implement and re-create even 'common' volume rendering algorithms on a shifting, babelized and proprietary technological landscape.

In the knowledge enterprise however, it is 'mission-critical' that data becomes verifiable information and that this information is interoperable between systems and portable across platforms. We use the term 'enterprise' broadly to describe any organization (research, clinical, industrial) where the access of diversified workers to data must be scalable and efficient. First let us consider the following examples from academia, as a scientific research enterprise: a perceptual study about the value of a new volume rendering algorithm for stereo depth discrimination, and a 3DUI study about surgical planning with volumetric data. In both of these cases, the scientific results from the study are only valid in so much as they are repeatable: that is, able to be independently implemented, run and evaluated. Several use cases are apparent in the IEEE VR 2010 Workshop on Medical Virtual Environments [9] for example. The parameters generating the stimuli (views) include camera and object transformations, lighting equations, volume ray-casting algorithms, mesh and isosurface material properties, clipping planes, etc. all must share a conceptual interoperability and perceptual equivalence regardless of the proprietary application, rendering library or display hardware used.

Now, let us discuss an example from the clinical world. The use of clinical imaging modalities, especially CT, MRI, PET and ultrasound, has been steadily increasing over the last decade [12]. The CT and MRI modalities are creating increasingly more slices of images for a given study due to a parallel increase in hardware capabilities. Consider a patient who receives a CT exam at hospital A due to a malignancy. The image is reconstructed as a 3D volume, segmented and marked up by the radiologist (and may even have a camera animation if the exam involved a virtual colonoscopy [4]). The patient pursues a second opinion and brings a copy of the studies to a specialist who practices at hospital B, which has a different vendor than hospital A for viewing medical images. Again, as in the previous example from academia, the parameters generating the

\*e-mail: npolys@vt.edu

mark ups and camera animations must demonstrate interoperability and perceptual equivalence, or those from the initial radiologist will be lost. This could lead to a delay of care as the specialist at hospital B tries to communicate with her counterpart at hospital B and/or a repeat (and redundant) imaging study, leading to increased costs and possible increased radiation exposure.

Despite the multitude of abstractions and interfaces for interactive volume rendering, we claim a premium on the requirement for a cross-platform representation of interactive rendering parameters and the accessibility of volume rendered views by a greatest common denominator—international standards and open source toolkits (see Figure 1). The ISO standard of Extensible 3D (X3D), for example, is truly a platform-independent scene graph with declarative and imperative representations of immersive virtual environments and their behaviors (i.e. [11]). This value is especially important for enterprises that require interoperability among multiple applications, the portability across World Wide Web devices and the durability of their volumetric presentations and environments (i.e. [10]).

In the case of durability, we ask "how durable is this presentation to avoid bit-rot and still be accessible in 30+ years?". Unfortunately, this is not an empty rhetorical challenge. Consider designing a contemporary human factors study of presence and simulator sickness that could be compared with SGI flight simulator results from the 1990s. Such a task would be nearly impossible; those virtual environments are frozen in human memory, never to run again. Virtual environments and virtual reality will have their greatest practical impact when they can mature to a common, durable conformance level. Innovators and entrepreneurs can extend and add value in an ecosystem on top of that base platform. Not only is this durability an imperative requirement for scientific progress, but also for the efficient realization of government, citizen's rights and the public interest.

## 2 CHALLENGES

Let us first identify and briefly describe the technological components and challenges of immersion and volume rendering; we categorize the challenges into three groups: data, software and hardware.

**Data** must support diverse sources; could be numbered stacks of image files, DICOM data, or several other formats; we use Nearly-Raw Raster Data (NRRD) as many tools and libraries support this simple format for segmentations; some data formats, like DICOM with JPG2000, require royalties from software implementers.

**Software** interoperable with diverse processing pipelines for volumetric data, depending on its sources and its data distribution. Typically, these pipelines involve: the choice of appropriate transfer functions and render styles to visualize relevant data or regions of interest, the segmentation, or grouping, of voxels to apply different rendering styles, and the surfacing of internal structures. Depending on the runtime and application, the surfacing may be an explicitly-derived mesh of vertices and faces or it may be a visual effect of rendering an isocontour along a given voxel threshold.

**Hardware** accessibility for diverse platforms from hand-helds to multi-core clusters, and visual display systems with stereoscopy and multiple screens; as well support for tracking systems (for user-centered projections) and high-DOF interaction devices (haptic or tracked devices).

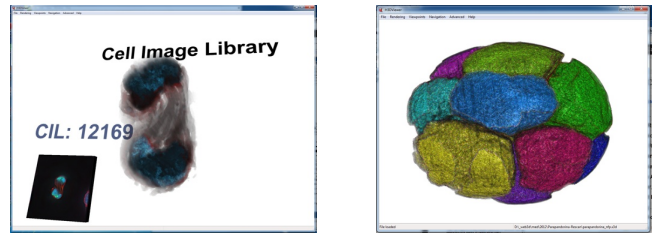


Figure 2: X3D Volume visualizations using H3D: at left, cell image visualization with segmentations from multi-channel microscopy; at right, the Parapandorina microfossil with segmentations used to create Isosurface masks.

## 3 CAPABILITIES

### 3.1 Extensible 3D (X3D)

The International Standards Organization (ISO) standard for 3D graphics over the Internet is Extensible 3D (X3D), which is maintained and developed by the Web3D Consortium (<http://www.web3d.org>). The open and royalty-free ISO standard scene graph has evolved through 15 years of hardware development and software boom-busts and still remains the greatest common denominator for communicating real-time interactive 3D scenes over the web. With dozens of implementations across industries and operating systems, X3D specifies this scene graph in layers of functionality, known as 'Profiles' and several encodings, including binary and XML. The Extensible 3D scene graph, X3D (ISO/IEC 19775), provides an expressive and durable platform to describe and deliver interactive rich-media 3D application views.

Working above any specific rendering library, X3D provides a powerful set of abstractions to compose meshes, appearances, lighting, animations, viewpoints, navigations and interactions. In addition to encoding the scene graph in XML, UTF-8 and binary format, the X3D standard also specifies the Application Programming Interface (API), known as the Scene Access Interface (SAI). The standard includes SAI language bindings for ECMAScript and Java; toolkits have demonstrated several other bindings including C++, Python and VBScript.

With a proven node set and transformation hierarchy, complex environments can be built and visualized at web enterprise scale. As a scene graph run-time, X3D also specifies a 'Behavior Graph', which describes the routing of events between nodes for animation and interaction. Several types of sensor nodes can be instantiated to enable their siblings with pointing and dragging as well as environmental (visibility, proximity, collision) events. X3D applications are being driven with a variety of user interfaces paradigms and hardware from laptops to multi-touch tables and from mobile devices to immersive VR installations today.

The newest revision of X3D: version 3.3 (2012) includes new components and functionality for reproducible, platform-independent volume rendering. X3D's Volume rendering capabilities were originally designed to improve the accessibility of 3D reconstructions of CT, MRI, PET or ultrasound presentations outside the radiology suite [5]. The expressive and compose-able rendering styles of X3D provide support for the visual differentiation of structures and systems in the volumes (segmentations), and the registration / fusion of two or more segmented volumes (blending). Figure 2 shows example screenshots from VT's X3D volume visualizations.

The new ISO specification Extensible 3D (X3D) adds significant volume rendering capabilities with support for different types of volume data and a broad selection of render styles.

### 3.1.1 Volume data

- The VolumeData node specifies a simple non-segmented volume data set that uses a single rendering style node for the complete volume.
- The IsoSurfaceVolumeData node specifies one or more surfaces extracted from a voxel data set. A surface is defined as the boundary between regions in the volume where the voxel values are larger than a given value (the iso value) on one side of the boundary and smaller on the other side and the gradient magnitude is larger than a predefined threshold.
- The SegmentedVolumeData node specifies a segmented volume data set that allows for representation of different rendering styles for each segment identifier.

### 3.1.2 Render styles

- The OpacityMapVolumeStyle specifies that the associated volumetric data is to be rendered using the opacity mapped to a transfer function texture. This is the default rendering style if no other is defined for the volume data (see Figure 3, left).
- The BoundaryEnhancementVolumeStyle node provides boundary enhancement, where faster-changing gradients (surface normals) are darker than slower-changing gradients. Thus, regions of different density are made more visible relative to parts that are of relatively constant density.
- The CartoonVolumeStyle generates a non-photorealistic rendering and uses two colours that are rendered in a series of distinct flat-shaded sections based on the local surface normal's closeness to the average normal with no gradients in between (see Figure 4).
- The EdgeEnhancementVolumeStyle node specifies edge enhancement by darkening voxels based on their orientation relative to the view direction (see Figure 5).
- The ProjectionVolumeStyle volume style node uses the voxel data directly to generate output colour based on the values of voxel data along the viewing rays from the eye point (see Figure 3, right).
- The ShadedVolumeStyle node applies the Blinn-Phong illumination model [1, 8] to volume rendering, similar to the model used for polygonal surfaces.
- The SilhouetteEnhancementVolumeStyle specifies that the associated volumetric data shall be rendered with silhouette enhancement. Enhancement of the basic volume is provided by darkening voxels based on their orientation relative to the view direction.
- The ToneMappedVolumeStyle node specifies that the associated volumetric data is to be rendered using the Gooch shading model [3] of two-toned warm/cool colouring. Two colours are defined, a warm colour and a cool colour. The renderer shades between them based on the orientation of the voxel relative to the user (see Figure 6).
- The BlendedVolumeStyle combines the rendering of two voxel data sets into one by blending the values according to a weight function.
- Finally, the ComposedVolumeStyle node is a special rendering style node that allows compositing multiple rendering styles together into a single rendering pass (see Figure 6). ProjectionVolumeStyle is the only style of the above mentioned that is not composable.

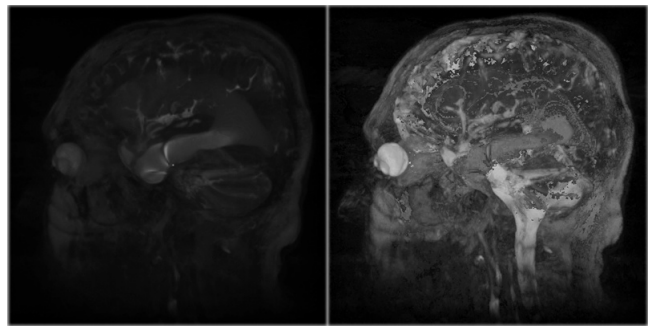


Figure 3: Brain data set visualized with different render styles: OpacityMapVolumeStyle (left) which is the default style, and ProjectionVolumeStyle with a Maximum Intensity Projection (MIP) algorithm applied (right).

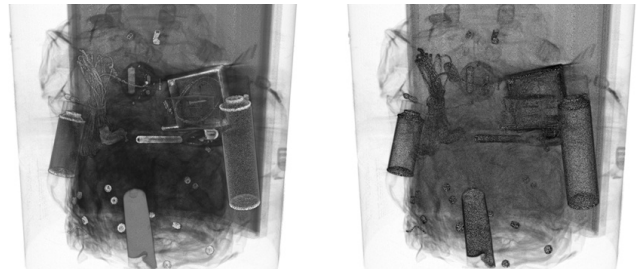


Figure 4: Default volume style on left and CartoonVolumeStyle on right.

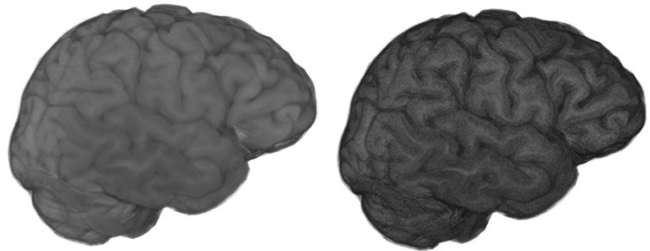


Figure 5: Default volume style on left and EdgeEnhancementVolumeStyle on right.

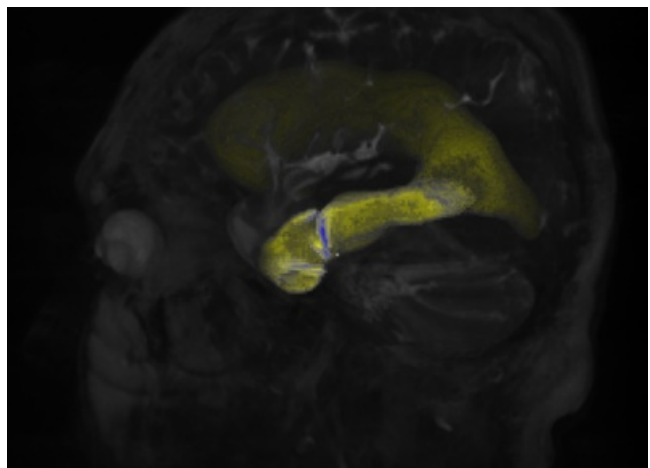


Figure 6: Segmented volume data using a combination of OpacityMapVolumeStyle and ToneMappedVolumeStyle.

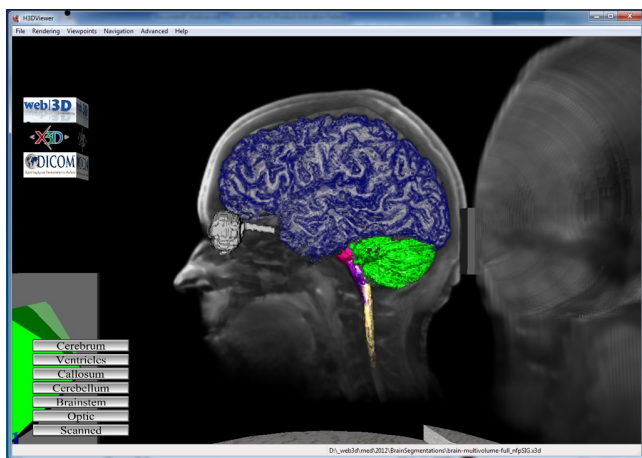


Figure 7: X3D volume visualizations using H3D in an interactive presentation, merging different image modalities in a X3D scene graph.

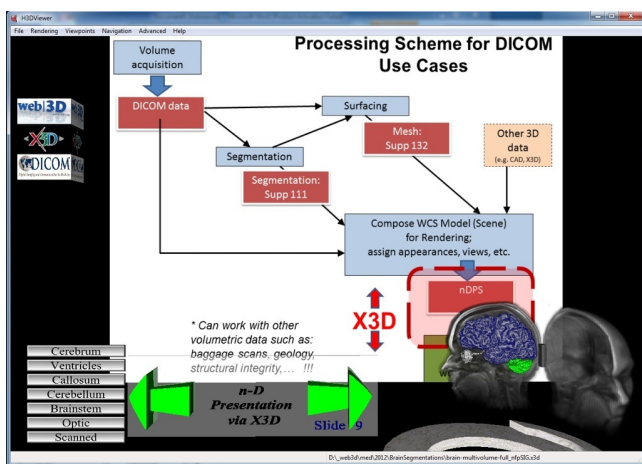


Figure 8: Alternative view of X3D volume visualizations using H3D in an interactive presentation, merging different image modalities in a X3D scene graph.

### 3.1.3 Scene graph interaction

For minimum conformance, sensor nodes that require interaction with the geometry (e.g., TouchSensor) shall provide intersection information based on the volume's bounds. An implementation may optionally provide real intersection information based on performing ray casting into the volume space and reporting the first non-transparent voxel hit.

Navigation and collision detection also require a minimal conformance requirement of using the bounds of the volume. In addition, the implementation may allow greater precision with non-opaque voxels in a similar manner to the sensor interactions.

Figures 7 and 8 show examples of compositions of various scene graph elements with polygon-based visualisations seamlessly integrated with volume rendering.

## 3.2 H3D

H3D API is an open source haptics software development platform that uses the open standards OpenGL and X3D with haptics in one unified scene graph to handle both graphics and haptics. H3D API is released under the GNU GPL license, with options for commercial licensing. It is cross platform (e.g., Windows, OSX and Linux) and haptic device independent (e.g., support for CHAI3D but also

proprietary APIs like Sensable's OpenHaptics). It enables audio integration as well as stereography on supported displays (e.g., ranging from quad-buffered stereo-modes over spanned/tiled desktops to anaglyphic rendering). Recently, HDMI 1.3 3D-Stereo frame packing was added, which allows for example stereo output on consumer display devices like the head mounted display HD1 from Sony. Unlike most other scene graph APIs, H3D API is designed chiefly to support a special rapid development process. By combining X3D, C++ and the scripting language Python, H3D API offers three ways of programming applications that offer the best of both worlds—execution speed where performance is critical, and development speed where performance is less critical. However, most importantly in the context of this paper, H3D combines those features with the volume rendering extension of the X3D standard.

## 4 EVALUATION

### 4.1 Data

Figures 1 and 2 demonstrate X3D volume visualization across a number of crucial use cases including Microscopy, Paleobiology, Custom Prostheses, Surgical Planning, Informed Consent, Anatomy Education, and Non-invasive Sensing. Volumetric data may come from a variety of modalities, but as the number and resolution of scans increases, there are two chief concerns: the utility of image archives (in electronic medical records for example) and the scalability of our current data models. In regards to the first, X3D contains a rich metadata capability, enabling the cross-referencing and fusion of image, scene graph and ontology data; as part of the original TATRC (<http://www.tatrc.org>) funded X3D project, we demonstrated the integration of the Foundational Model of Anatomy (FMA) and SNOMED ontologies with interactive X3D volume renderings, enabling knowledge-based interactive 3D applications [6]. The second issue (big data) clearly impacts the software and hardware challenges (cluster rendering), but also the data and file system format. Dougherty et al [2] propose HDF5 as a binary filesystem container to handle the current explosion of image data archives.

### 4.2 Software

The ability to process and expressively present volumetric data is a key requirement and rich in its aspects. For example, we make extensive use of ImageJ (<http://imagej.nih.gov/ij>) and Teem (<http://teem.sourceforge.net>) to load volumetric image data. For open processing pipelines, Seg3D (<http://seg3d.org>) and ITKSnap (<http://itksnap.org>) are robust tools supporting several interactive and algorithmic methods for segmentation. ParaView (<http://paraview.org>) and Visit (<https://wci.llnl.gov/codes/visit>) can leverage clusters for large-scale scientific volume rendering. Voreen (<http://voreen.org>) uses a visual programming model to build pipelines and advanced visual effects in its renderings. DICOM still does not describe 3D presentation states, but its models of segmentations and surfaces align with X3Ds' geometry types.

The X3D scene graph provides the formal structure to unify several data types from resources and services across the web and portray them as interactive scenes including lights, cameras, polygons, volumes animations and scripting. H3D (<http://h3dapi.org>) and InstantReality (<http://instantreality.org>) are two softwares that support X3D Volume rendering; at Web3D and SIGGRAPH 2012, VicomTech and Fraunhofer IGD demonstrated X3D volume rendering natively in HTML5 browsers with X3DOM (<http://x3dom.org>) and WebGL.

The concept of sensors and event routing to fields in X3D allows for a very flexible implementation of interaction with a growing range of input devices. The open source implementation H3D, with its API based on X3D, is also a prominent example with medical

simulators, featuring complex interaction schemes quickly prototyped in Python and then implemented as C++ custom nodes. Nevertheless, in future work the standard will be extended to further interaction concepts, such as advanced haptic rendering and soft tissue response. The Web3D Consortium has a User Interface Working Group that welcomes further research and development to address the challenges in interaction design and paths to support and standardize new interaction metaphors via X3D.

The combination of open standards and capable open-source tools here today is certainly a capable common denominator to deliver and interchange virtual environments. Both standards and toolkits are key ingredients for repeatability; but what about extension? Within an X3D standard runtime, there are formal methods to extend the scene graph with custom nodes, known as Prototypes. With the open toolkits, developers have the additional option of getting 'under the hood' of the scene graph and adding their own classes and modifying existing ones. Thus new information and interaction designs can be implemented and evaluated.

### 4.3 Hardware

While there are many different hardware combinations possible, we will briefly mention a few suggestions for a moderate budget. With the progress in home entertainment, 3D Display systems are getting much more affordable. Recently, we added support for HDMI 1.3 3D-Stereo frame packing in H3D. This allows, for example, immersive stereo output on consumer display devices like the head mounted display HD1 from Sony. For an even lower price the Oculus Rift HMD (<http://www.oculusvr.com>) should become available soon. As mentioned before, H3D supports a whole range of input/haptic devices. Good entry level systems are the Novint Falcon (<http://www.novint.com>), which is limited to 3DOF input, or the SenseAble Phantom Omni device (<http://www.sensable.com>), with 6DOF input. Both devices support 3DOF force feedback. Other immersive input options without haptic feedback, are the Microsoft Kinect system (<http://www.xbox.com/KINECT>) or the soon coming desktop equivalent Leap Motion Controller (<https://leapmotion.com/>).

## 5 DISCUSSION

X3D has proven to be a robust and extensible presentation standard to improve interactive visual access to volume and medical image data and clinical ontologies. Focusing on X3D Volume rendering adoption, conformance and extension can bring the full power of X3D to bear, improving efficiency and outcomes. For example, the creation of open workflows and mobile HTML5 clients for 3D display of medical imaging data will stimulate innovation and lower the barrier to entry for clinicians, students and the layman alike.

This new ISO standardization has more far reaching effects in that the imaging data is used not only for diagnosis and treatment, but as source material in physiological and surgical simulators (i.e. [13]). The surgical simulation field has traditionally been a mosaic of parallel efforts that produce cutting edge simulations of specific organ systems, specific organs or general tissue dynamics which are isolated products that cannot communicate with each other. To align these efforts so that the 'best of breed' organ or system-specific models can be brought together and we can begin to realize a truly computational representation of a human being, standards such as X3D for 3D representation of medical imaging data must be in place. Additional extensions to this standard (such as [14]) will enable simulators to conform to approved specifications and allow more valid comparisons between them when attempting to judge their fidelity and utility.

## ACKNOWLEDGEMENTS

The authors gratefully acknowledge the contributions of the international standards community, especially the Web3D Consortium in building a rich platform for volume rendering. Also, thanks to Abhijit A. Gurjarpadhye for his X3D cell image visualizations. This work was supported in part by US TATRC (W81XWH-06-1-0096) and Virginia Tech Advanced Research Computing.

## REFERENCES

- [1] J. F. Blinn. Models of light reflection for computer synthesized pictures. *ACM SIGGRAPH Computer Graphics*, 11(2):192–198, 1977.
- [2] M. T. Dougherty, M. J. Folk, E. Zadok, H. J. Bernstein, F. C. Bernstein, K. W. Eliceiri, W. Bengler, and C. Best. Unifying biological image formats with HDF5. *Communications of the ACM*, 52(10):42–47, 2009.
- [3] A. Gooch, B. Gooch, P. Shirley, and E. Cohen. A non-photorealistic lighting model for automatic technical illustration. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 447–452, New York, NY, USA, 1998. ACM.
- [4] L. Hong, A. Kaufman, Y.-C. Wei, A. Viswambharan, M. Wax, and Z. Liang. 3d virtual colonoscopy. In *Biomedical Visualization, 1995. Proceedings.*, pages 26–32, 1995.
- [5] N. W. John. Design and implementation of medical training simulators. *Virtual Reality*, 12(4):269–279, 2008.
- [6] N. W. John, M. Aratow, J. Couch, D. Evestedt, A. D. Hudson, N. Polys, R. F. Puk, A. Ray, K. Victor, and Q. Wang. MedX3D: standards enabled desktop medical 3D. *Studies in Health Technology and Informatics*, 132:189–194, 2008.
- [7] D. Jansson, E. Sundn, A. Ynnerman, and T. Ropinski. Interactive volume rendering with volumetric illumination. In *Eurographics 2012 - State of the Art Reports*, pages 53–74, 2012.
- [8] B. T. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.
- [9] N. Polys and N. John. In *IEEE VR 2010 Workshop on Medical Virtual Environments*, <http://www.hpv.cs.bangor.ac.uk/vr10-med>, 2010.
- [10] N. Polys and A. Wood. New platforms for health hypermedia. *Issues in Information Systems*, 13(1):40–50, 2012.
- [11] N. F. Polys, D. Brutzman, A. Steed, and J. Behr. Future Standards for Immersive VR: Report on the IEEE Virtual Reality 2007 Workshop. *IEEE Computer Graphics and Applications*, 28(2):94–99, 2008.
- [12] R. Smith-Bindman, D. L. Miglioretti, E. Johnson, C. Lee, H. S. Feigelson, M. Flynn, R. T. Greenlee, R. L. Kruger, M. C. Hornbrook, D. Roblin, L. I. Solberg, N. Vanneman, S. Weinmann, and A. E. Williams. Use of diagnostic imaging studies and associated radiation exposure for patients enrolled in large integrated health care systems, 1996-2010. *JAMA*, 307(22):2400–2409, Jun 2012.
- [13] S. Ullrich and T. Kuhlen. Haptic palpation for medical simulation in virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):617–625, April 2012.
- [14] S. Ullrich, T. Kuhlen, N. F. Polys, D. Evestedt, M. Aratow, and N. W. John. Quantizing the void: extending Web3D for space-filling haptic meshes. *Studies in Health Technology and Informatics*, 163:670–676, 2011.