

# A Handheld Flexible Display System

Jonathan Konieczny\*

Clement Shimizu†

Gary Meyer‡

D'nardo Colucci§

University of Minnesota  
Digital Technology Center

## ABSTRACT

A new close range virtual reality system is introduced that allows intuitive and immersive user interaction with computer generated objects. A projector with a special spherical lens is combined with a flexible, tracked rear projection screen that users hold in their hands. Unlike normal projectors, the spherical lens allows for a 180 degree field of view and nearly infinite depth of focus. This allows the user to move the screen around the environment and use it as a virtual "slice" to examine the interior of 3D volumes. This provides a concrete correspondence between the virtual representation of the 3D volume and how that volume would actually appear if its real counterpart was sliced open. The screen can also be used as a "magic window" to view the mesh of the volume from different angles prior to taking cross sections of it. Real time rendering of the desired 3D volume or mesh is accomplished using current graphics hardware. Additional applications of the system are also discussed.

**CR Categories:** I.3.3 [Computer Graphics]: Picture/Image Generation—Display Algorithms; I.3.7 [Computer Graphics]: Three Dimensional Graphics and Realism—Virtual Reality; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques

**Keywords:** visualization, virtual reality, user interfaces, projectors, volume rendering, curved sections

## 1 INTRODUCTION

Viewing slices of volume data is still one of the most prevalent methods to analyze medical data such as MRI, CT, and PET. While many techniques for visualizing such medical data (as well as other volume data) have been created, in nearly all of them the user views slices of the data on a flat screen. In order to observe these slices from different positions or angles, some form of mouse and keyboard interface is used. In terms of software design and implementation this may be the most straightforward way to create visualization systems, but it may not be the best way to actually view the data. Rather than looking at such three dimensional cross sections as projections onto a fixed two dimensional screen, it would be better to observe the slices in their actual positions within the volume dataset. Projectors provide this possibility, as they are capable of casting light into a large 3D volume. However, most projectors can only focus on a small portion of this possible volume at any given time.

This paper demonstrates how a video projector and a movable projection screen can be used to create a novel system for taking slices through volume data. A projector system is introduced that

has an extremely wide field of view and nearly infinite depth of focus. These characteristics allow the screen to remain in focus at any part of the projection volume. The ability to maintain focus is combined with a tracking system to create a simple and intuitive user interface. The tracker is placed on a flexible projector screen, which allows the program to know the exact position and orientation of that screen within the volume data set. This allows a correct image from the volume data set to be projected onto the screen from the spherical projector.

The end goal of this system is to allow a user to view virtual objects, such as medical volume data, closely and naturally without any special knowledge about the equipment. An important aspect of most real volume data is that information of interest rarely follows a perfect plane. For instance, a doctor may be interested in viewing a section of the spine, or a layer of the kidney. Therefore, it is important that the viewer be able to see curved slices of the data. By using a lightweight flexible display, an easy way to manipulate the shape of a desired volume slice is created. This part of the system can even be used with a normal display screen for 3D visualization and modeling tasks. With the addition of the special projection system, not only can the user easily manipulate viewing of virtual objects, but they can also view them in a real environment as if actually holding a slice of the data.

In addition to the visualization of volume data, this system can also be used to aid in the viewing and design of other kinds of data. This paper also presents the use of this system for both examining polygonal meshes and for design and viewing of object material properties. Also, methods for combining these different viewing modes are described in order to further increase interactivity and realism.

### 1.1 Relevant Work

The idea of using projectors on non-uniform surfaces has been utilized for some time [6, 7]. Many of these systems concentrate on using multi-projector display systems onto arbitrary surfaces and ad hoc clusters of projectors [5, 11]. Most of these methods employ some form of environment sensing to compute the transformation between the projectors and the displayed objects.

This system adapts methods for display on arbitrary surfaces, and also extends them for use with the spherical projection system. In addition, this paper concentrates on the use of the system for user interface and visualization purposes. In particular, much of the user tests and volume data concentrate on medical visualization.

Specific applications that utilize these types of display methods are scarce, especially systems designed for virtual reality at close range. A similar application to our magic window was done by Tsang et al. but their approach employed a flat panel display on a mechanical arm [10]. The system presented in this paper allows for a flexible display surface that is light weight and easy to manipulate.

There are tabletop virtual reality systems that have already used the idea of a transparent prop. Schmalstieg et al. [9] and Coquillart et al. [2] both use transparent props as a user interface tool in their systems. Schmalstieg in particular introduces the idea of using a transparent rectangle to sweep through a virtual scene. However, neither of these systems specifically intersect screens with 3D vol-

\*e-mail: jkonie@cs.umn.edu

†e-mail: clement@cs.umn.edu

‡e-mail: meyer@cs.umn.edu

§e-mail: colucci@dtc.umn.edu

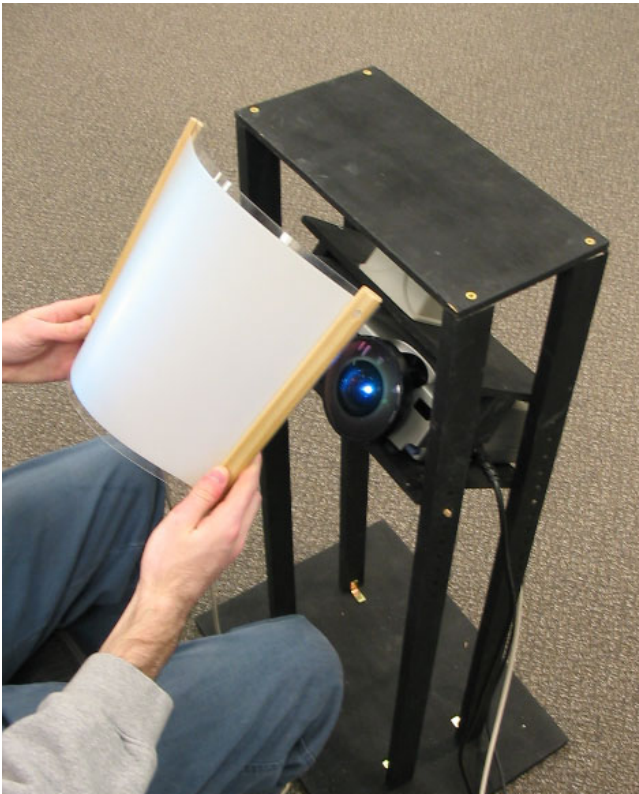


Figure 1: The system setup.

ume data sets. Both of these systems also use large scale custom hardware. All of the components in our system are small, portable, and can be easily integrated into an existing computer system. Also, this system introduces the ability to flex the prop, an important advancement when viewing certain types of real data.

Viewing medical data as a set of volume slices has also been widely used for some time. Likewise, methods for specifying and extracting curved slices from volumes have been proposed before [3, 8]. However, all of these methods have a keyboard and mouse interface for specifying this extraction, and they use a normal display medium for viewing it, such as a monitor or flat projection screen. The capability of our system to view curved slices is similar to that of Figueiredo and Hersch [3], but the curved slices are viewed directly on the screen that creates the slices.

## 1.2 Overview

This paper first covers the setup of the projector and tracking system. Then, it describes how to correctly calibrate the projected image for the rear-projection screen. Next, specific applications are presented. Finally, conclusions and future work are discussed.

## 2 SYSTEM SETUP

The following describes the hardware and software used in this system.

### 2.1 Spherical Projection

The spherical projector used in this implementation is an Epson 715c, a three LCD color projector with 1024 x 768 resolution.



Figure 2: A close up of the spherical projector lens.

The stock lens has been removed and a custom fisheye lens (Elumens Corporation) installed. The fisheye lens projects  $180^\circ$  along the horizontal axis and  $135^\circ$  along the vertical axis. This truncated fisheye projection makes more efficient use of the 4:3 aspect ratio of the LCD panel.

A special property of this fisheye lens, shown in Figure 2, is that it projects with an exact  $f-\theta$  pixel distribution. This means that the angle to which a specific pixel is projected is linearly proportional to its distance from the optical axis. The result is an equiangular pixel distribution across the entire projected field. Typical rectilinear projection lenses project with an  $f-\tan(\theta)$  distribution which results in smaller angular pixel distribution at the center than at the edge of the projected field. Use of this  $f-\theta$  pixel distribution in image correction is described in Section 2.3.

While the extremely wide projected field of view creates a large viewing volume even at close range, it is the nearly infinite depth of field that makes this system successful. The infinite depth of field is a by-product of the short focal length fisheye lens.

In optics a common rule of thumb is that for a lens of focal length “ $f$ ” an object that is  $\geq 20f$  away is essentially infinitely far away. Reversing this rule for projection, if the screen is  $\geq 20f$  away from the lens the image will always be in focus. The fisheye used in this system has a focal length of 6mm. Therefore, the screen can be placed anywhere from infinity to 120mm from the lens while maintaining a focused image. In practice, the lens is focused to allow the screen to be as close as 75mm from the lens. This sacrifices the focus at infinity which is not relevant to the application, as the furthest away from the projector the screen may be tracked is 3 feet (91.44cm).

### 2.2 Flexible Display Design and Tracking

The screen’s size and ability to flex into different shapes are both critical to the success of the system. It is important to have a screen that is large enough to view a reasonable amount of the volume data at once, while not being too bulky to manipulate easily. The ability of the screen to curve into different shapes is equally important. If it is too rigid, it will take too much effort to bend, and will not be able

to be manipulated into a reasonable set of slice shapes. On the other hand, if it is too flexible, it will be difficult for the user to maintain it in a desired shape. Another consideration when designing the flexibility of the screen is the number of dimensions along which the screen can be bent. More dimensions require more trackers in order to accurately approximate its actual shape, and it also takes more user effort to get into a desired shape. However, being able to bend it along more dimensions will allow the screen to more accurately follow the curvature of the volume data.

These considerations led to the choice of the screen size to be 9x11 inches, or just slightly larger than a normal piece of paper. Since people handle paper on a regular basis, this is a natural shape for a user to manipulate. Also, this size is large enough to view an entire cross section of many important data sets, such as a human head and internal organs.

For the flexibility of the screen, it was decided that two trackers would be used, one on each bottom corner of the screen. Since each tracker gives position and orientation information, not only is the exact location of the screen known, but also the curvature of screen. In order to calculate the location of the screen, the origin of the generated scene is set to the position of the magnetic field generator, and all other objects are translated to their correct locations in relation to it. The orientation of the trackers is separated into the surface normal, tangent, and bitangent vectors. Let  $P_0, N_0, T_0,$  and  $B_0$  be the position, surface normal, tangent, and bitangent of the tracker on the bottom left of the screen. Let  $P_1, N_1, T_1,$  and  $B_1$  be the position, surface normal, tangent, and bitangent of the tracker on the bottom right of the screen.

Since each screen support is rigid, the positions of the top two control points can be deduced from the bottom control points and the length of the screen,  $l$ . This information is used to construct the geometry matrix as follows:

$$G_H = \begin{bmatrix} P_0 & P_0 + T_0 * l & B_0 & B_0 \\ P_1 & P_1 + T_1 * l & B_1 & B_1 \\ T_0 & T_0 & N_0 & N_0 \\ T_1 & T_1 & N_1 & N_1 \end{bmatrix} \quad (1)$$

The geometry of the flexible display can now be approximated by evaluating the Hermite patch Equation 2 over the domain  $(0, 1)^2$ .

$$P(x, y) = Y \cdot M_H \cdot G_H \cdot M_H^T \cdot X^T \quad (2)$$

$$X = [x^3 \quad x^2 \quad x \quad 1] \quad (3)$$

$$Y = [y^3 \quad y^2 \quad y \quad 1] \quad (4)$$

$$M_H = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (5)$$

When the screen is flexed, its stiffness could affect the resulting shape. Screen stiffness is not explicitly addressed in Equation 2. However, by scaling the length of the bitangent vectors in Equation 2, stiffness can be expressed implicitly.

Using two trackers in this way allows for bends along the vertical axis to be performed, but not bends along the horizontal axis. Although more trackers would allow for more degrees of freedom, it would make the screen more bulky and harder to manipulate, because holding the screen in a position with more than a single curve is difficult. In addition, it would increase the cost of the system.

The final screen is composed of Stewart Filmscreen rear projection material. This is placed on top of a clear plastic backing to provide stiffness. Along each side of the screen is a rigid support bar with a mounting for a tracker (Figure 1). The tracking system used is an Ascension Tech PCIBird<sup>TM</sup> magnetic tracking system.

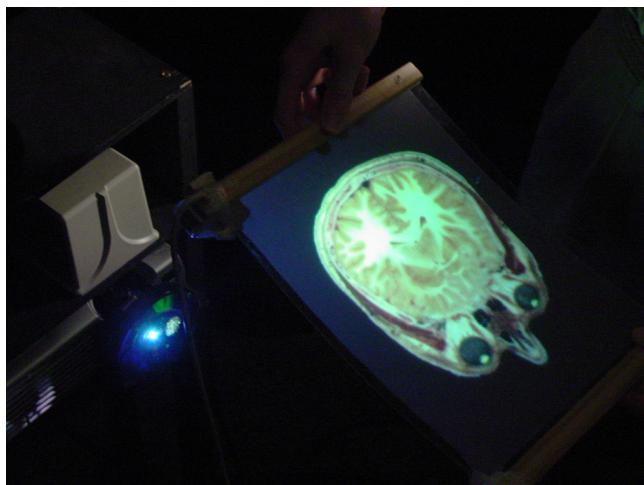


Figure 3: A cross section of a 3D volume is projected onto the flexible screen.

### 2.3 Image Correction

The spherical lens that is used causes the image to be projected as a sphere into the environment rather than the typical rectangular display. While this gives a much larger field of view, and nearly unlimited depth of focus, it means that the projected image will be distorted. In order to create a correct perspective image, the hardware vertex shader is used to pre-warp the object(s) in the scene.

This adjustment is done by creating a mapping between the spherical projection space and the rectangular screen space. The equiangular projected pixel distribution allows the creation of a single transform that will move every vertex in spherical space to its proper location in rectangular screen space. First, the world location of each vertex is converted into spherical space:

$$r = \frac{2}{\pi} \arccos\left(\frac{z}{d}\right) \quad (6)$$

$$\vec{\phi} = \left( \frac{x}{\sqrt{x^2 + y^2}}, \frac{y}{\sqrt{x^2 + y^2}} \right) \quad (7)$$

$$(x', y') = r * \vec{\phi} \quad (8)$$

where  $z$  is defined as the axis parallel to the center of the projector's field and  $d$  is the distance of the vertex to the center of the projector.

Now  $f$  corresponds to an  $x, y$  screen space vector and  $r$  is the length of that vector. This moves every vertex into a two dimensional screen space between  $-\pi$  and  $\pi$ . Dividing  $r$  by  $\pi$  as in the above equation normalizes this result to be between  $-1$  and  $1$ . In order to preserve depth, each vertex is then given a depth value equal to its distance from the projector.

An important property of this image correction is that it is done per vertex, not per pixel. Therefore, object tessellation can become an issue. If an object has an insufficient number of vertices, or they are poorly spaced, the normal linear interpolation between them done by the standard graphics pipeline will have poor results. Therefore objects with low vertex counts will need to be re-tessellated to take full advantage of this method.

This method bears some similarity to the quadric transfer functions described by Raskar et al. [5], but is specialized for spherical correction. Coombe et al. [1] also use this technique to project points onto a hemisphere in graphics hardware for radiosity calculation.



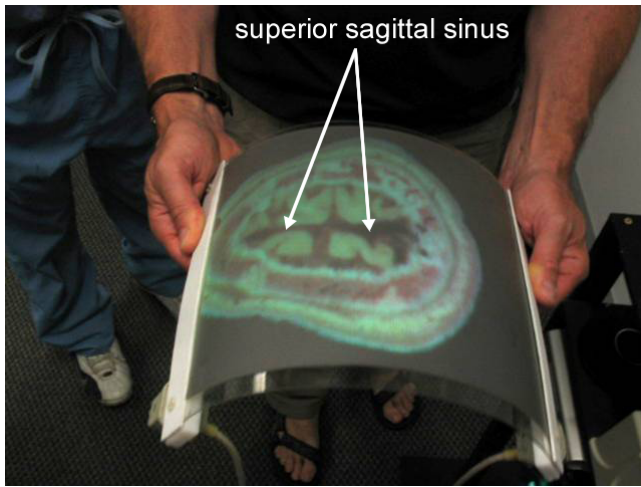


Figure 4: The curved section showing the area just below the top of the skull

### 3 APPLICATIONS

The result of combining a spherical projection lens with a flexible display screen is an excellent interface for the exploration of volume data sets and virtual objects. In this section, the paper first describes the use of this setup as a volume visualization system. It then explores the system's use as a magic window application and investigate how to combine the magic window with the volume visualization system. The section ends by examining a shader lamp application.

#### 3.1 3D Volume Visualization

The primary application for this novel display device is a volume visualization system. This application takes advantage of current graphics hardware in order to render cross sections of a 3D volume in real time. Rendering the rear-projection screen as a slice into a 3D volume creates a method of viewing cross sections of a 3D volume faster and more intuitively than using a mouse or keyboard interface.

In order to use the screen as a virtual slice into the volume, the data set is first loaded as a 3D texture into the graphics hardware. The volume is then given a center in virtual space that matches with real world coordinates. All data is stored in terms of inches. This keeps the measurement data exact, and allows the user to know precisely where in real space the virtual data set is supposed to be. The volume is also scaled to match the size of its real-world counterpart. Now there is a virtual data set "floating" in space in front of the projector.

Once the volume data is set up, the screen can be used to slice into it. In order to do this, the coordinates of the screen, as given by the tracker, are sent into the pixel shader and compared with the coordinates of the 3D volume. When they intersect, the correct voxel of data at that location is found and output from the pixel shader. In this way, the complete intersection between the screen and the 3D volume is output, as seen in Figure 3. Since the intersection between the volume and screen is computed on a per-pixel basis, this solution is not only accurate, but is easily extendable to arbitrarily sized and shaped screens. This allows the intersection between the curved screen, represented as a Hermite surface patch, and the volume to be computed.

Since real volume data rarely follows a perfect plane, this ability to intersect curved sections of the data allows the user to view

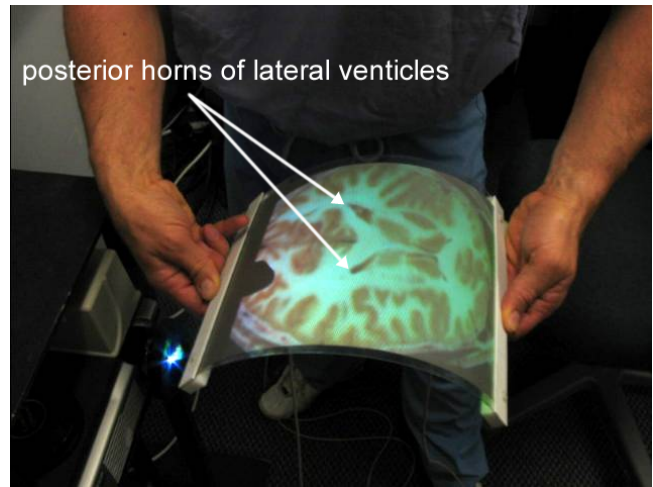


Figure 5: Moving the flexible screen downward from that shown in Figure 4 reveals a section of the top of the brain

larger portions of relevant data at once. This is illustrated in Figure 6 where the user is slightly curving the screen to view a larger portion of the alveolar nerve than a simple plane would allow. Another example of this is shown in Figures 4 and 5 where the user is viewing different sections along the top of the brain, which is highly curved.

Additionally, the current screen could be replaced with a screen of different size or shape simply by replacing the screen object used by the program. This could allow the user to swap between desired screens interactively, for instance to use a larger screen to view the overall volume, then a smaller screen to more easily inspect specific parts of the data set.

#### 3.2 Magic Window

While it is valuable to view cross sections as if you are actually holding the slicer, it can be somewhat confusing if the user is unable to see the volume prior to slicing into it. In order to help alleviate this issue, the concept of using the screen as a "magic window" is introduced. This approach involves the use of a projection screen as a window into the virtual world, allowing the user to see any virtual objects that are behind it.

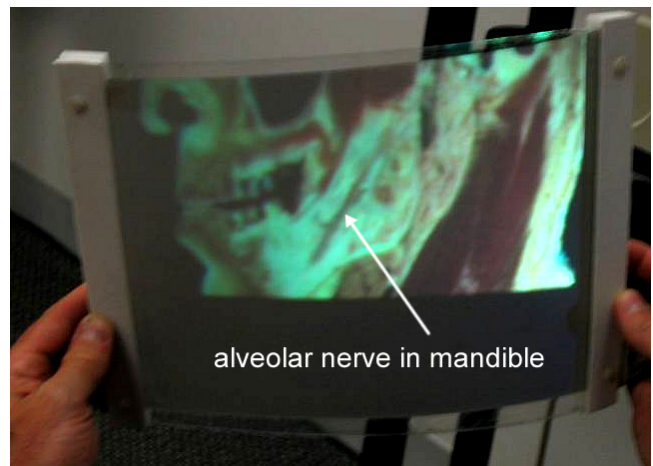


Figure 6: A subtle screen curve allows viewing of non-planar anatomy, such as the alveolar nerve.

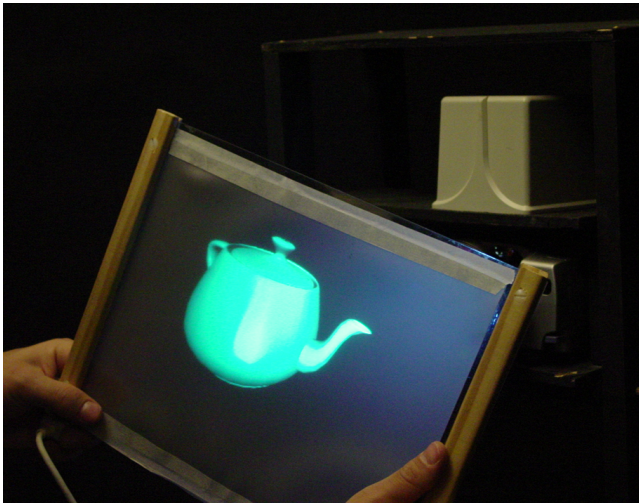


Figure 7: A magic window teapot.

In addition to aiding the visualization of 3D volumes, the use of the screen as a magic window can be useful as a standalone application. Any object with a geometric mesh can be viewed through the screen as if it was a real object in the environment. This could potentially be very useful in design. For example, determining how an object would look in a room before it was purchased or perhaps even created.

In order to do this, the geometry of the desired object is first loaded as in a normal graphics application. Then a viewport is created that matches with the projector screen dimensions and viewer location. For the current setup, viewer location is assumed to be a certain distance directly behind the screen. However, an additional tracker can be used to allow an arbitrary viewer location. An easy way to think about this is that in a normal graphics program, the screen stays in place and all the virtual objects move. In this setup, the virtual objects all stay static and the screen itself moves.

### 3.3 Combining the Magic Window with 3D Volumes

The magic window can be combined with the 3D volume visualization system to allow for a more immersive user experience. Once a mesh has been generated for a given 3D volume (using marching cubes, for instance), both algorithms can be run at the same time to allow the volume to be viewed through the screen as in the magic window or on the screen as in the volume rendering. Integrating the two algorithms is relatively simple. The solution can take advantage of the fact that if the screen is currently intersecting the volume, the cross section will always be shown on that portion of the screen. Therefore, an effective way to combine the two is to render the magic window portion first, turn off depth testing, and then render the volume slice. On portions of the volume that have no data (are black) the slice is turned completely transparent so that the object mesh can be seen behind it.

### 3.4 Large Data Sets

With small data sets, the entire volume can be loaded into memory at the same time, and the system can be run in real time. However, for larger volume data sets such as the visible human, not all of the data may fit in the memory of the graphics card at the same time. This can lead to waiting time while the data loads new textures for the rest of the data set when the screen is moved.

Fortunately, the system lends itself well to predictive loading. As users move the screen through the data set they tend to continue in

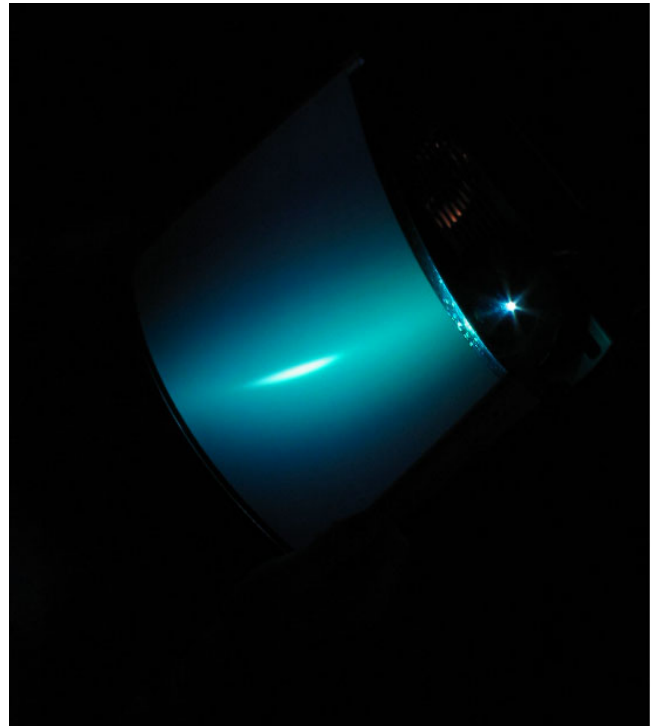


Figure 8: The system being used as a shader lamp.

the same direction they are already heading. Therefore, by sampling the tracker data the program can predict ahead of time which 3D textures will be needed and load them up before an interruption is caused.

### 3.5 User Interface Additions

After initial reactions from volunteers who used the system, a couple of potential problems were found. The first is that the users sometimes had trouble knowing the exact location of the slice within the volume. This happened because once the slice was inside the volume, there was no output cue to the location or size of the volume. The second problem was difficulty with holding the screen steady for extended periods of time. If a user wished to carefully examine a particular slice, they had to hold the screen still in space for an uncomfortably long period of time.

As a result of this user input, we created a couple of user interface additions. The first is a display of the overall scene the user is currently viewing. Both the 3D volume and screen are displayed on either a separate display or in the corner of the rear-projection screen itself. This allows the user to see exactly how large the volume is in relation to the screen, as well as where the screen is located inside the volume. The view of the 3D volume can be a complete volume rendering or, for the sake of preserving real time speed, just the hull of the volume.

The second addition is a simple “freezing” command. When the user has placed the screen at a position he/she wished to view for a while, a button can be pressed that locks the screen at that slice of the volume. The user can then inspect the slice on a separate monitor. Note that the current implementation does not allow these frozen slices to be viewed on the flexible screen itself.

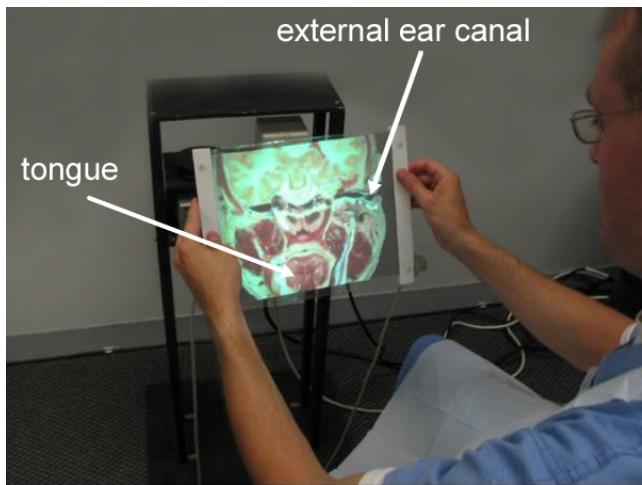


Figure 9: A medical student views a section of the visible human head

### 3.6 Shader Lamp

Here an alternative application is presented. Rather than visualizing volumes, this system can be used as a shader lamp [7]. A shader lamp is the use of a projector to make an arbitrary real world object take on a different or modified appearance. For the purposes of this system, the object to be modified is the projection screen itself. Since the screen can be moved around and flexed by the user, the use of a shader lamp in this system lends itself well to material design and visualization. For instance, the user could change the properties of the material they are currently viewing, and see how these changes affect the appearance of the screen in real time.

In order to implement the shader lamp approach, the object to be viewed is placed in the environment. This is similar to both the magic window and 3D volume applications, except in this case the object that is placed in the scene is the screen itself. Instead of the object remaining static in the environment as with the other two applications, it moves according to the tracked location of the screen. This causes the screen to always be entirely lit by the projector. The user can then give the screen any desired material properties and view how these properties appear in a desired environment by moving the screen around and flexing it. Figure 8 shows the system being used as a shader lamp. In this case, the screen is being made to look as if it was painted with a blue metallic paint.

One thing to note is that the location and size of the specular highlight in this application is dependent upon viewer location. In order to handle this, the viewer location can either be tracked or assumed to be static at a reasonable location. The screen is then rendered from the projector position, with the viewer location used for the lighting calculations. This is similar to the method presented by Raskar et al. [7].

As with the 3D volume visualization system, the shader lamp can be combined with the magic window application. The user may press a button that “freezes” the current shape of the screen. This is similar to the user interface addition described above. The current shape of the object is then loaded into the magic window viewer and may be viewed through the screen.

## 4 PERFORMANCE

The current implementation of the system runs on a Pentium 4 3.2Ghz processor with 2 gigabytes of ram. An NVIDIA Quadro FX 1000 graphics card is used. For the volume slicer, the pixel shader dominates the rendering time, except when using

very high numbers of polygons. As a result, the number of frames per second that can be achieved is based largely on how far away from the projector the screen is located. Table 1 shows the relationship between number of polygons, distance from the projector, and frames per second achieved. All numbers are for the volume slicer. Since neither the magic window nor the shader lamp require any pixel calculations, the performance for those applications is as good or better than the volume slicer. As Table 1 shows, real time interactivity is achieved except at very high numbers of polygons. The current implementation uses only 400 polygons, which is enough to properly tessellate the screen for the spherical distortion correction (see Section 2.3 for details). Therefore, the system normally runs at very good frame rates.

For texture loading, the current implementation uses blocks of  $256^3$  RGB textures for volume data. Loading each block into the card from main memory requires 170 milliseconds. Therefore, if a slice not currently loaded is accessed, a 170 millisecond wait time is incurred. Otherwise, this portion of the program does not affect the listed frame rates.

## 5 CONCLUSIONS AND FUTURE WORK

This new system is ideal for the design and visualization of objects in a real environment. The ability to bend the viewing surface and still see a correct image gives it a significant user interface advantage over a normal flat screen. At the same time, the spherical lens gives a large viewable area and solves the depth of field problem associated with normal projection systems.

Most of the volunteers who tried this system were interested in its use a medical tool. Volunteers who had an actual medical background were especially excited by its ability to curve and follow actual anatomical structures. Figures 4, 5, 6, and 9 show the volunteers exploring the visible human head with the system. The most common reaction was that it has great potential as a learning tool.

There are some modifications to the system that could improve it. First, tracking latency can be an issue with some applications, especially the shader lamp. When an object being viewed has parts that are close to the edge of the screen (which is always the case with the shader lamp), tracker latency becomes very noticeable and it can detract from the user’s experience. A faster, more accurate tracker could help with this problem.

Second, as the object moves further from the projector, the pixel size becomes larger much faster than with a normal projector, a result of the hemispherical field of view. Having a projector with higher resolution could help alleviate this problem to some extent, as could the addition of more projectors.

Third, although the field of view is quite large, it is not arbitrary. As the handheld display becomes parallel to the axis of the projector lens, the pixels enlarge, to the point that the image becomes unrecognizable. Once again, having more than one projector could solve this problem.

This system also has some possibilities for extension that we wish to pursue. One possible extension is the use of a tracker combined with a trigger along the side of the frame (or a foot pedal) for use as an input device. Currently, the only input the system allows

Distance	Number of Polygons					
	100	400	1,600	6,400	25,600	102,400
4	47	45	43	35	23	9.7
12	70	65	63	55	28	10
36	90	85	82	65	30	10.25

Table 1: Frame rate (FPS) based on number of polygons and distance from projector



is rescaling of the volume (via the keyboard) and freezing of the image (with the keyboard or a foot pedal). We believe that making the tracker into an input device will open up interesting user interface possibilities.

Another possible extension would be to increase the scale of the setup. Currently, there is only one projector, and a single hand held screen. More projectors and a large screen attached to a pivot could allow for much larger slices of data sets to be explored. This could even allow for slices through the entire human data set to be viewed. Such a display system could be very useful for teaching purposes.

In the near future, flexible OLEDs are likely to become widely available. By using this new technology, all of the advantages of the present system could be preserved while eliminating many of the disadvantages. With a flexible OLED screen, the warping step for the projector would not be required. Also, the current implementation has varying pixel size based on the distance and orientation between the projector and screen. An OLED would have a constant pixel resolution across the screen regardless of orientation.

## ACKNOWLEDGEMENTS

The authors of this paper would like to thank the people at Vital Images for their valuable input about the system, and for providing the data sets used in initial testing of the system. We would also like to thank Dr. Ken Roberts, Dr. Mark Cook, and the members of the UM gross anatomy class for their input about the system. Finally, we would like to thank Stephen Shimizu for constructing the screen frame for us.

## REFERENCES

- [1] Greg Coombe, Mark J. Harris, and Anselmo Lastra. Radiosity on graphics hardware. In *Proceedings of Graphics Interface 2004*, 2004.
- [2] Sabine Coquillart and Gerold Wesche. The virtual palette and the virtual remote control panel: A device and an interaction paradigm for the responsive workbench((tm)). In *VR '99: Proceedings of the IEEE Virtual Reality*, page 213, Washington, DC, USA, 1999. IEEE Computer Society.
- [3] O. Figueiredo and R.D. Hersch. Parallel unfolding and visualization of curved surfaces extracted from large 3d volumes. In *Journal of Electronic Imaging*, volume 11, pages 423–433, 2002.
- [4] J. Foley, A. van Dam, S. Feiner, and J. Hughes. Computer graphics: Principles and practice. Addison-Wesley Publishing Co., ISBN 0-201-12110-7, 1990.
- [5] Ramesh Raskar, Jeroen van Baar, Paul Beardsley, Thomas Willwacher, Srinivas Rao, and Clifton Forlines. ilamps: geometrically aware and self-configuring projectors. *ACM Trans. Graph.*, 22(3):809–818, 2003.
- [6] Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs. The office of the future: a unified approach to image-based modeling and spatially immersive displays. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 179–188, New York, NY, USA, 1998. ACM Press.
- [7] Ramesh Raskar, Greg Welch, Kok-Lim Low, and Deepak Bandyopadhyay. Shader lamps: Animating real objects with image-based illumination. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 89–102, London, UK, 2001. Springer-Verlag.
- [8] L. Saroul, S. Gerlach, and R. D. Hersch. Exploring curved anatomic structures with surface sections. In Greg Turk, Jarke J. van Wijk, and Robert J. Moorhead, editors, *Proceedings of IEEE Visualization 2003*, pages 27–34. IEEE Computer Society, IEEE Computer Society Press, October 2003.
- [9] Dieter Schmalstieg, L. Miguel Encarna, and Zsolt Szalav. Using transparent props for interaction with the virtual table. In *SIG3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 147–153, New York, NY, USA, 1999. ACM Press.
- [10] Michael Tsang, George W. Fitzmaurice, Gordon Kurtenbach, Azam Khan, and Bill Buxton. Boom chameleon: simultaneous capture of 3d viewpoint, voice and gesture annotations on a spatially-aware display. In *UIST '02: Proceedings of the 15th annual ACM symposium on User interface software and technology*, pages 111–120, New York, NY, USA, 2002. ACM Press.
- [11] Ruigang Yang, David Gotz, Justin Hensley, Herman Towles, and Michael S. Brown. Pixelflex: A reconfigurable multi-projector display system. In Thomas Ertl, Ken Joy, and Amitabh Varshney, editors, *Proceedings Visualization 2001*, pages 167–174. IEEE Computer Society Technical Committee on Visualization and Graphics Executive Committee, 2001.