Stereo Pseudo 3D Rendering for Web-based Display of Scientific Volumetric Data

Daniel Chern-Yeow Eng and Yoonsuck Choe

Department of Computer Science, Texas A&M University, College Station, TX 77843-3112
Email: {edaniel,choe}@tamu.edu

Abstract

Advancement in high-throughput microscopy technology such as the Knife-Edge Scanning Microscopy (KESM) is enabling the production of massive amounts of high-resolution volumetric data of biological microstructures. To fully utilize these data, they should be efficiently distributed to the scientific research community (e.g., through the Internet) and should be easily annotated and analyzed. Given the volumetric nature of the data, visualizing them in 3D is important. Since we cannot assume that every end user has high-end hardware, an approach that has minimal hardware requirement will be necessary. There are several prominent applications that facilitate the viewing of large collections of images over the web. Google Maps and Google Maps-like interfaces such as Brainmaps.org allow users to pan and zoom 2D images efficiently. However, they do not yet support the rendering of volumetric data in their standard web interface. Thus, we propose a new method of rendering volumetric data over the web that directly uses the raw image stack, without any computation on the data at all. The human visual system has the capability of viewing stereo images in 2D and turn that into a 3D perception. To generate stereo images, we will need to create the effects of depth and binocular disparity using 2D images. By using simple HTML and JavaScript that are computationally cheap, we can accomplish both tasks dynamically in a standard web browser, by overlaying the images with intervening semi-opaque layers. We expect the approach presented in this paper to be applicable to a broader domain, including geology and meteorology.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Perception and Cognition in Volume Visualization, Multi-dimensional Data

1. Introduction

1.1. Overview

Advancement in high-throughput microscopy technology such as Knife-Edge Scanning Microscopy (KESM) is enabling the production of massive amounts of high-resolution volumetric data of biological microstructures [MAM08] (figure 1, section 2.1). KESM has been used for serial sectioning and imaging whole mouse brains. In order to tap into the full potential of such a dataset, data sharing and model sharing becomes important tasks. For example, the massive amounts of data (typically 2 to 20 TB per mouse brain, depending on the microscope objective used) need to be made accessible to researchers around the globe. These researcher in turn should be able to annotate and analyze, and ship their models back to the main server, integrating those into the original data source.

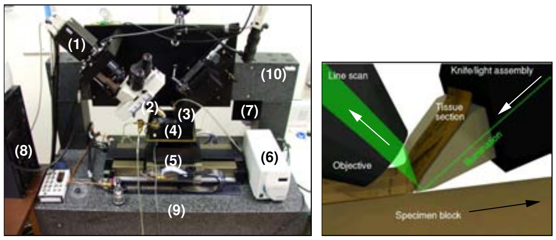
Given the volumetric nature of the data, visualizing them in 3D is mandatory. There are many volume visualization algorithms that can be used for this purpose (see [Yag00] for an extensive review). However, since we cannot assume that every end user has a high-end hardware, an approach that has minimal hardware and software requirement will be necessary, such as a standard web browser running on a typical personal computer. There are several prominent applications that facilitate the viewing of large collections of images over the web. Google Maps [Goo] and Google Maps-like interface such as Brainmaps.org [SIJJ07] allow users to pan and zoom 2D images efficiently. However, they do not yet support the rendering of volumetric data in their standard web interface. The standard web-based data delivery method does not seem to be suitable for 3D visualization. Our task here is to use this existing infrastructure plus properties of the human visual system to enable efficient and effective 3D visualization of scientific data.

Here, we propose a new method of rendering volumetric data over the web with a minimal set of required computing resources by tapping into the functionality of our perceptual system. In order to do so, stereo images will be used to represent the volumetric data. As shown in [LC03] and [BB99], stereo image pairs can serve as a vivid 3D representation.

To render image based stereo images, we will need to create the effects of depth and binocular disparity using a stack of 2D images (cf. [LC03]). To create depth, we interleaved semi-opaque layers between scanned data images to create a depth effect for the data. Images closer to the user will look clearer, while images further away from the user will appear hazy (distance attenuation). This approach is analogous to aerial perspectives described in [KSTE06] where "aerial perspective cue was modeled by decreasing contrast as distance from the viewer increased ... is present in the atmosphere." The resulting image stacks will provide the depth cue that is similar to the linear intensity ramp method and accumulation of buffers method mentioned in [HMS95]. However, since our method is image based, no extra computation is needed to re-render the images unlike the above approaches.

To create binocular disparity, we stacked the same set of data next to each other, and offsetted the position of each image layer with the layer on top having the most disparity. There are several fast algorithms for generating stereo images, most prominently [HK96] and [AH94]. In both algorithms, they followed the general algorithm of selecting a correct view point for the left-eye image, and then reprojecting it onto the right-eye image. The main difference between the two algorithm is that [HK96] uses a shear warp algorithm to generate the initial view, while [AH94] uses a ray casting strategy. The approach that we are taking here is similar to the slice-by-slice rendering method (shear warp volume rendering) mentioned in [HK96], but since our approach is image based, we are not re-projecting any data to create a new image, we are just simply assembling preexisting image slices produced by the KESM.

The operations of generating depth and binocular disparity are not computationally intensive, and can be done dynamically on a standard web browser. This is in the same spirit as the Image-based rendering approach where the object is rendered from existing images rather than geometrical data and is similar to the approach found in [CBY06]. Since we are dealing with vascular objects that have been scanned into sliced images, we simply reassembled it by overlaying one slice on top of another slice. This operation can be simply done by using straight forward Hyper Text Markup Language (HTML) and Cascading Style Sheets (CSS). JavaScript is also used to facilitate the transmission of the data over the web and for the dynamic generation of HTML tags. Thus, for the web-based visualization, we do not require specialized software or plug-ins such as Virtual Reality Mark-up Language (VRML), Java Applets, or QuickTime Virtual Reality (QTVR) as used in [PHKH04], [EWE99], [CBY06], and [CYB08]. Our approach is general enough to be applicable to other types of volume data, e.g., from geology and meteorology.



(*a*) KESM (*b*) Sectioning and imaging

Figure 1: ***The Knife-Edge Scanning Microscope (KESM).*** *Adapted from [MAM08]. See section 2.1 for details.*

The main contribution of this paper is to have shown how raw image stacks of large volumes of scientific data can be used directly to visualize the data in a web environment, without any computation at all. The use of standard HTML and JavaScript with an image overlaying technique makes the approach very efficient, with low hardware and network bandwidth requirements.

The rest of the paper is organized as follows. In section 2, we will discuss background work motivating our research. Next, in section 3, we will describe our approach. Section 4 presents the main results, followed by section 5, discussions, and section 6 conclusion.

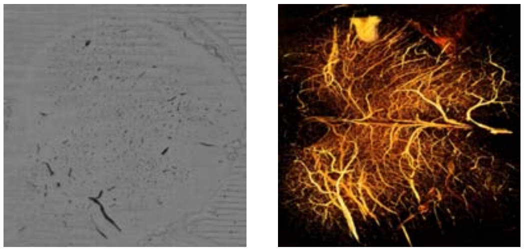
2. Background

In this section, we will briefly review the type of volumetric data we deal with, and discuss related works that motivated our approach.

2.1. Knife-Edge Scanning Microscopy

The Knife-Edge Scanning Microscope enables sectioning and imaging of whole small animal organs such as the mouse brain. The data resolution of the images can be up to 300 nm × 300 nm × 500 nm, and when a whole mouse brain is scanned, the resulting data can reach 20 TB. Transmitting such a data set across the network in its entirety is not a viable option, so an on-demand, multi-resolution approach is necessary, similar to Google Maps.

Figure 1 presents an overall view of the Knife-Edge Scanning Microscope (KESM) and its principle of operation. Figure 1*a* shows a photo of the KESM with its major components marked: (1) high-speed line-scan camera, (2) microscope objective, (3) diamond knife assembly and light collimator, (4) specimen tank (for water immersion imaging), (5) three-axis precision air-bearing stage, (6) white-light microscope illuminator, (7) water pump (in the back) for the



(*a*) Single raw image slice (*b*) Volume rendering

Figure 2: ***KESM data.*** *See section 2.1 for details.*

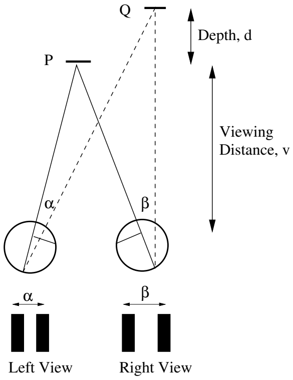


Figure 3: ***Horizontal Disparity.*** *Redrawn from [Mal01]. See section 2.2 for details.*

removal of sectioned tissue, (8) PC server for stage control and image acquisition, (9) granite base, and (10) granite bridge. Figure 1*b* illustrates the principle of operation of KESM. The objective and the knife are held in place, while the specimen affixed on the positioning stage moves (black arrow) and gets scraped against the diamond knife, generating a thin section flowing over the knife. Line-scan imaging is done at the very tip of the knife where the distortion is minimal. Illumination is provided through the diamond knife (white arrows indicate the light path).

The images obtained from KESM are high contrast and high quality. Figure 2 shows (*a*) an example image slice and (*b*) a volume rendering of 200 slices (200 μm thick section, 1 μm per slice) from a mouse spinal cord specimen. The vascular network in the specimen was stained through perfusion using India ink. Black dots of various diameters appearing in figure 2*a* are the blood vessels. A complex web of blood vessels can be seen in figure 2*b*. (Note that the image was rotated and scaled to match the orientation and aspect ratio of *a*.)

2.2. Binocular Disparity

One possible approach to distribute the volumetric data generated by instruments such as the KESM is to use the 2D images directly, instead of reconstructing the geometry of the objects of interest prior to transmitting them over the Internet. Using these original images, we can dynamically generate stereo images using the concept of binocular disparities.

Due to the disparity in the two images perceived by our two eyes caused by their displacement (about 6 cm apart), stereoscopic depth perception is made possible. When such horizontal disparities happen, points in the three dimensional world is slightly different from each of the half images perceived by the eyes. Figure 3 [Mal01] shows a typical setup of our eyes and horizontal disparities. As one fixes both eyes at image P, image cast by P will fall on both left and right eye, and if image Q is located β degrees from the left eye and α from the right eye, then image Q is said to have disparities of β-α degrees. Thus, the depth (*d*) is proportional to the distance or degree of separation between P and Q. The resulting images on the left and the right eye shown below in the same figure (figure 3, bottom) gives rise to depth when merged (parallel viewing).

2.3. Stereo Viewing

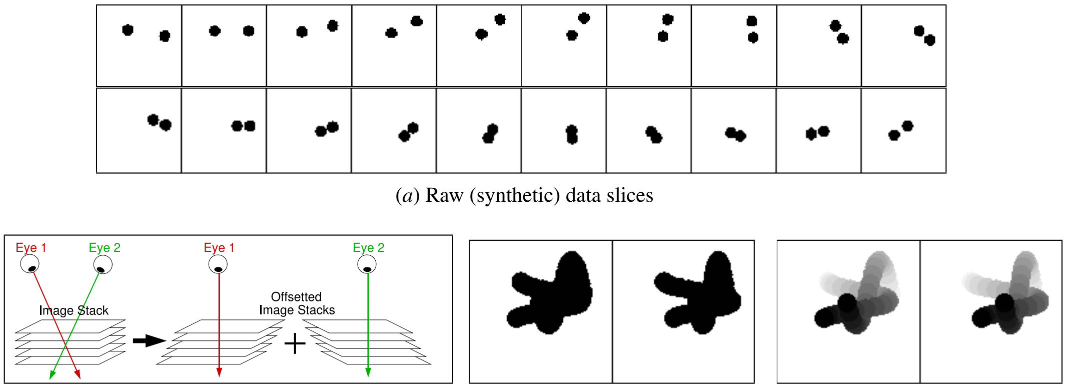
According to [Rho97] there are two methods to view a stereo image without any visual aids: convergent (or crossed) or divergent (or parallel) viewing. In convergent viewing, users will fix their right eye on the left image, and left eye on the right image while keeping their head straight. The image should converge and will lie in between the two original images. In divergent viewing, users will fixed their right eye on the right image, and their left eye on the left image. Initially the images will look superimposed and out of focus (when the images are really close to the eyes), as the images are moved further out, a third image should emerge in between the two images, and it should have convincing depth.

2.4. Realism of Stereoscopic Display

There has been much discussion about how realistic stereo images are and whether they can be a good representation of 3D or volumetric objects. Several studies including [LC03] and [BB99] showed that stereo images are perceptually effective for visualizing 3D or volumetric objects. On the other hand, [KSTE06] pointed out that the advantages of stereo display exhibited in the above studies might be task dependent, thus in some tasks stereo viewing might have no perceptual impact at all. Our interpretation is that stereo display is *as good as* other 3D visualization techniques, and thus suitable for our purposes. (Although motion parallax can provide even stronger depth cues, the effect cannot be achieved on a static image stack.)

3. Methods

In this section, we will describe how realistic 3D effect can be achieved using standard web protocols such as HTML



(*a*) Raw (synthetic) data slices

(*b*) Generating 3D views

(*c*) Offset image stacks

(*d*) Dist.-atten./offs. image stacks

Figure 4: ***Pseudo Stereo Pair Generation and Distance Attenuation.*** *Three-dimensional effects can be generated using a simple overlaying method. (a) A series of 20 image stacks taken from a synthetic volume data set is shown (left to right, top to bottom). (b) An illustration of how two stereo pairs can be generated using simple offsetting (shearing) and overlaying of the image stack. Such overlays can be easily produced in a web browser using Cascading Styling Sheets (CSS), from base images with transparencies (alpha channel). (c) Stereo-pair (for cross viewing) of offset image stacks are shown. The three dimensional effect is weak due to ambiguities in stereo registration. (d) Stereo-pair (for cross viewing) of offset image stacks with distance attenuation is shown. This can be achieved easily by inserting white semi-opaque layers interleaved between the data images. The three-dimensional structure of the embedded object is clearly visible.*

and JavaScript. Both binocular disparity and distance attenuation can be produced by using image overlays. Figure 4 shows an overview of our method.

3.1. Image Data Preparation

We obtained an image stack from KESM, consisting of 50 slices of mouse spinal cord, with its blood vessels stained in India ink. The images were 512 × 512 pixels at full-resolution, but for faster transmission, we prepared a subsampled version where we chopped up the full-resolution version into 180 × 180 pixel units, in the style of Google Maps.

Prior to further processing, the image stack needs to be preprocessed so that overlaying is possible. Alpha channels need to be added to the images, and background pixels turned transparent. We used command-line tools from ImageMagick (`http://www.imagemagick.org`) for this step.

3.2. Distance Attenuation

Distance attenuation can be achieved by interleaving white semi-opaque images (typically 10% opacity) in between the data images. This approach has a similar effects as aerial perspectives where objects closer to the image surface appear clearer while objects farther away from the image surface appear hazy. This approach can also be thought of as a variation of the painter's algorithm and image-based rendering where semi-opaque images are added in between images to render depth. One desirable thing here is that we only need a single image for the semi-opaque layer, and reuse that by inserting the same image between the data images (i.e., we only need to download the semi-opaque image once). Figure 4*c*–*d* shows the visual effect achieved by distance attenuation. The depth is much more apparent with distance attenuation (*d*) than without (*c*), even without stereo merging.

3.3. Horizontal Disparities

Horizontal disparities can be achieved using an overlaying technique similar to distance attenuation. Instead of stacking each image directly on top of each other, the horizontal position of each image can be offsetted slightly. Figure 4 shows a simple example with a synthetic volume image stack. Figure 4*b* shows how shearing the image stack can result in a stereo pair that gives a pseudo 3D effect. When offsetting the images, the depth of focus can be adjusted. For example, in figure 4*d*, the depth of focus is at the far background, thus images at the bottom of the stack will not be shifted while the images on the top of the stack will be shifted the most. Note that the horizontal disparity effect is much more vivid when combined with distance attenuation, as can be seen from figure 4*c* and *d*.

3.4. Implementation Techniques

In this section, we will explain in detail how we implemented the horizontal disparity and depth attenuation effect using standard web protocols.

3.4.1. HTML Implementation

To create two image stacks in HTML, we simply use the division `<div>` tags to represent two image stacks. Through manipulation of Document Object Model (DOM) using JavaScript we will be able to dynamically add images into these divisions. Since the position of these divisions are absolute, we'll be able to specify the location of each image, thus allowing us to overlay them or offset them before overlaying an image on top of another image [Cro07]. The following is the listing of the actual code used in the HTML file.

```
<div id="imgcontent" position ="absolute">
<div id="imgstacks" position ="absolute">
</div>
<div id="imgoffsets" position ="absolute">
</div>
</div>
```

3.4.2. JavaScript Implementation

We implemented several operations in JavaScript. The first thing we implemented is the fetching routine where we load the images transmitted from the web server to the browser's cache. Then we implemented the overlaying routine where new images are stacked on top of each other using the absolute position property found in Cascading Styling Sheets (CSS). To generate offsetted image stacks, we simply offsetted the position of the image by *N* pixels either to the right or to the left depending on the choice of the user (for parallel or crossed viewing). To stack an image on top of another image, we dynamically generated an image element and then set its position to an absolute position using the CSS properties, before appending the element back to the HTML document. It should be noted that even though we are generating two stacks of images, the images are only loaded once, but used twice, thus reducing the memory required to store the images.

The following is the actual JavaScript code we used to implement this functionality. We first create an image element in the HTML DOM tree and then populate its attributes through JavaScript. Since the position of the image is absolute, we are able to stack multiple images on top of each other.

```
var newimg=document.createElement('img');
newimg.style.position='absolute';
newimg.src=currImageArray[counter].src;
newimg.style.top = toppos;
newimg.style.left = leftpos;
newimg.style.width = width;
newimg.style.height = height;
document.getElementById("imgstacks")
                         .appendChild(newimg);
var opagueimg = document.createElement('img');
opagueimg.style.position='absolute';
opagueimg.src = OpaqueImage.src;
```

To offset the position of each image slides, we just update the position of the image element by adding an offset value to its position.

```
var newimg=document.createElement('img');
newimg.style.position='absolute';
newimg.src=currImageArray[counter].src;
newimg.style.top = toppos;
newimg.style.left = leftpos + offsetValue;
document.getElementById("imgoffsets")
                         .appendChild(newimg);
```

4. Results

We applied the technique detailed above to the image stack produced by KESM. The images used in this section are from the mouse spinal cord scans. 50 image slices were used and in between each image slice a semi-opaque image with 10% opacity was inserted. For horizontal disparity, every image slice on the offsetted stack is offsetted by 1 pixel from the previous location.

4.1. Visualization Results

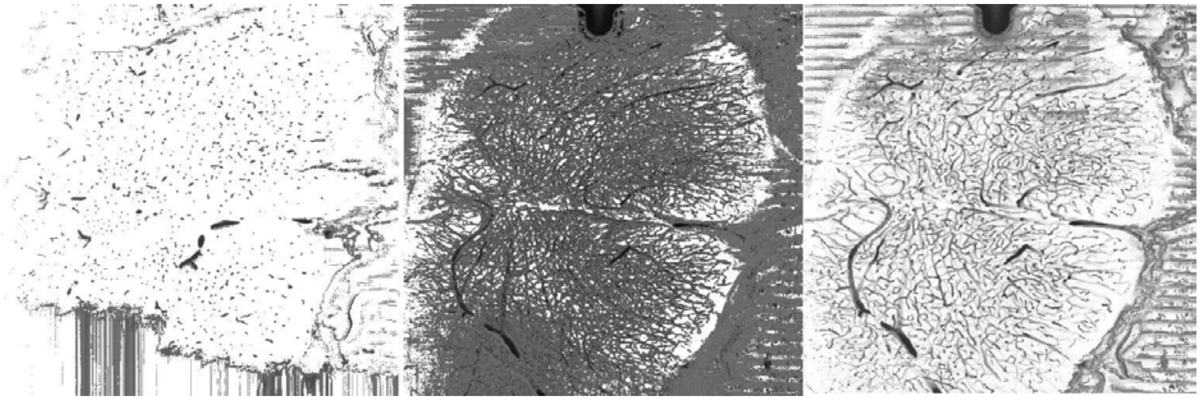
Figure 5 shows the effect of distance attenuation. Compared to minimum intensity projection, the data visualized with distance attenuation is much more effective in conveying the 3D properties of the data.

Figure 6 shows actual screenshots from a web browser displaying the spinal cord data. A full view and a zoomed-in view of the data set are shown, with horizontal disparity (offset) and distance attenuation turned on. When the stereo pairs are merged (through cross viewing), the full 3D effect can be perceived.

4.2. Performance

We tested our application on both Mozilla Firefox 2.0 and Microsoft Internet Explorer 7.0. We examined the image load time as well as memory usage for 15,30 and 50 images produced by KESM. The images used in this implementation were in Portable Network Graphics (PNG) format. The image size ranged from 95kB to 126 kB per image. A linear growth in image load time and memory usage is expected since the images were no further compressed.

Figure 7 shows the memory usage and web page loading times. First, we tested the memory usage when displaying 15, 30, and 50 image slices. Two typical web browsers, Firefox 2.0 (red circle) and Internet Explorer 7.0 (blue square) were tested. As expected, the memory usage scales linearly. Next, we measured the download time for 15, 30, and 50 image slices. For this test, we used the *Web Page Speed Report* at `http://www.websiteoptimization.com`. Two types of networks were tested, ISDN 128Kbps (red circle) and T1 1.44Mbps (blue square), and both show linear scaling property. For T1 connections, downloading even 500 images (10 times the amount reported in our results) could be done in a reasonable amount of time. Once the images are downloaded (and put in the browser cache), displaying the overlayed images takes only several seconds (3 seconds on a 1.7 GHz PC running Linux, with 1GB RAM), so that different display parameter values can be tested.



(*a*) Single image slice (*b*) Minimum intensity projection (*c*) Overlay with distance attenuation

Figure 5: ***Visualizing Image Stacks.*** *Two methods are compared, for visualizing volume data contained in mouse spinal cord image stacks. (a) A single image slice is shown. (b) Minimum intensity projection of 50 images is shown. Due to the density of objects, it is hard to understand the geometric properties due to ambiguities. (c) The same data set in (b) is shown with distance attenuation by interleaving white semi-opaque layers between individual images. As in figure 4d, the 3D properties of the objects are much more clear.*

5. Discussion

In this paper, we have shown that using a combination of HTML, JavaScript and CSS, were are able to dynamically generate pseudo 3D stereo images directly from 2D image stacks. We were able to render complex microstructures with the only bottleneck being the number of images and their sizes (cf. [CBY06]). Our main contribution is to have shown an effective method for 3D visualization of large volume data through standard web protocols, with minimum hardware and software requirements. Even though most computers today are equipped with modest graphics capabilities for fast volume rendering, the massive volumes of images and data generated by high-throughput 3D microscopy will continue to exceed these capabilities, thus our method can be a good alternative for viewing of such data.

Our approach is ideal for delivering large biological volume data sets such as those from the mouse brain. There are notable efforts in building web-based brain atlases [Lei07, All, MGJS*03, SIJJ07]. Among these, BrainMaps.org is the closest to our approach [SIJJ07]. The main difference between our approach and BrainMaps.org is that we allow the viewing of multiple stacked images at once. We allow the user to view the volumetric structures in a pseudo 3D way while BrainMaps.org only allows the user to view one image slide at a time.

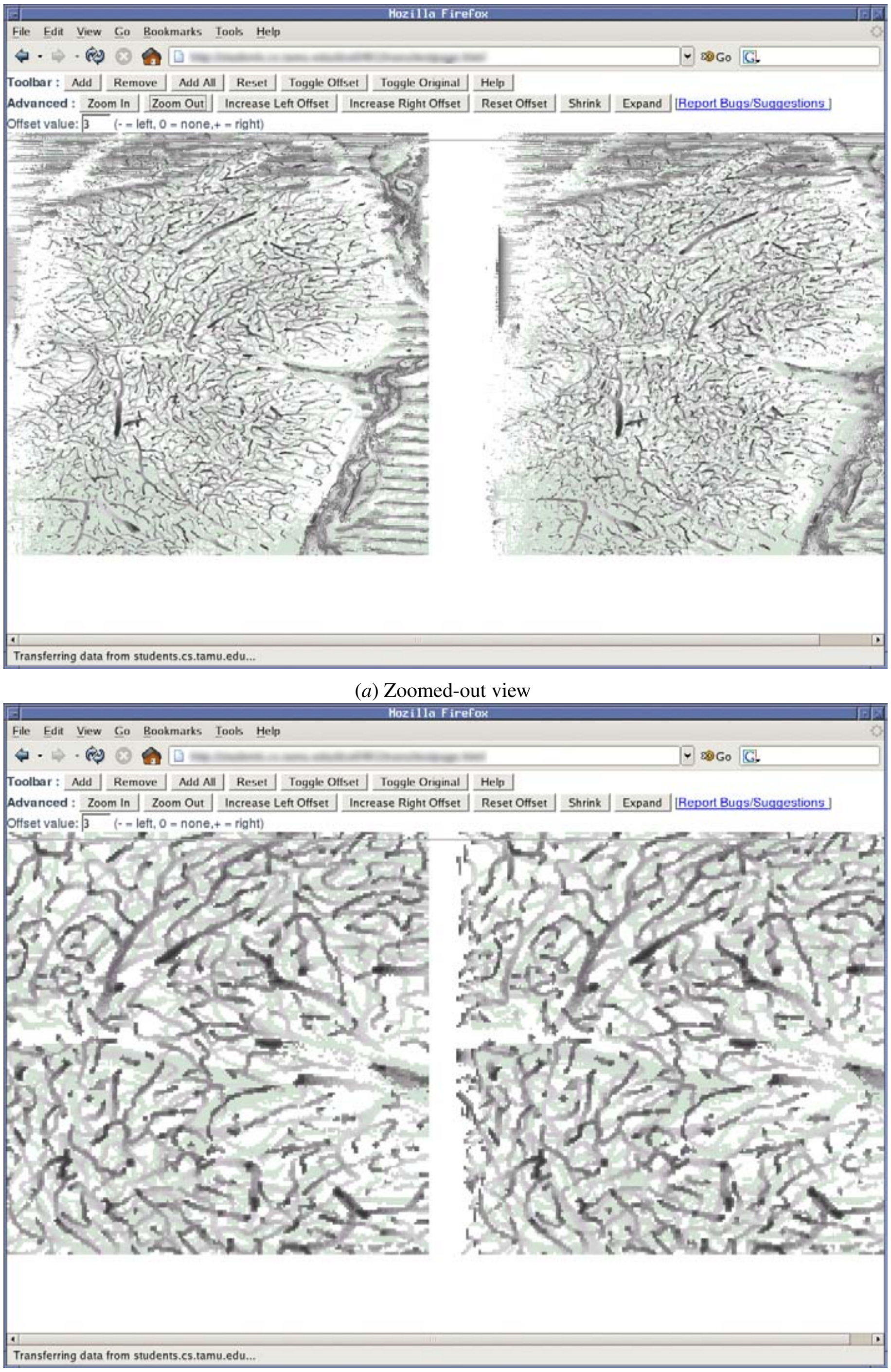
There are other approaches for web-based 3D visualization, such as [PHKH04], [EWE99] and [CYB08], but unlike these, our approach requires no specialized software or hardware, or any computation on the images; just a standard web browser that supports CSS and JavaScript is all that is necessary.

There are several limitations of our current approach. The obvious one is the requirement that viewers should be able to do stereo merging, and extended viewing (especially in the crossed viewing case) can put a lot of stress on the eyes. Simple and cheap viewing aids such as mirrors or prisms, or similar optical components can be used to overcome this issue. Another issue is that the stereo pairs generated by offsetting is only "pseudo" accurate, and not fully accurate. This issue could be resolved by introducing nonlinear offsetting. Instead of offsetting each image by a fixed amount, we can offset the images by an amount determined by a nonlinear function of depth. A major issue that cannot be easily addressed is that of interactivity. Unlike 3D visualization using full hardware/software support, our approach cannot generate different view points on the fly: the view point is predetermined by how the image stack is organized and cannot be changed. Finally, one might argue that 50 images is not enough to represent a large volume of data.

As shown in the results section, our approach scales linearly, so adding more images is not a problem. Depending on the number of images, the opacity of the semi-opaque layers need to be reduced. Another approach is to generate premerged images (5, 10, or more images stacked and merged into a single image) when larger, lower-resolution images are being viewed. This way, deeper volumes can be viewed with fewer images.

6. Conclusion

By exploiting how the human visual system works regarding 3D perception, we were able to develop an efficient visualization method for the distribution and visualization of


Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Go
Toolbar : Add Remove Add All Reset Toggle Offset Toggle Original Help
Advanced : Zoom In Zoom Out Increase Left Offset Increase Right Offset Reset Offset Shrink Expand [Report Bugs/Suggestions]

Offset value: 3 (- = left, 0 = none,+ = right)

Transferring data from students.cs.tamu.edu...

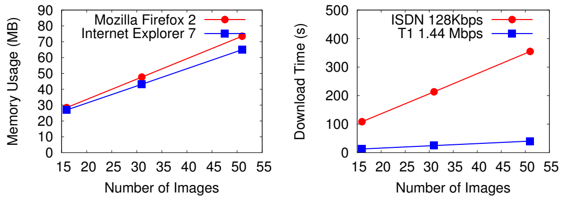
(*a*) Zoomed-out view

Offset value: 3 (- = left, 0 = none,+ = right)

Transferring data from students.cs.tamu.edu...

(*b*) Zoomed-in view

Figure 6: ***Screenshots of the Web-based Display of Mouse Spinal Cord Vascular Network.*** *Screenshots of web browser windows displaying stereo pairs (for crossed viewing) of a 30-image stack from the mouse spinal cord data set is shown. Once the eyes are crossed and the stereo pairs registered (right eye viewing left image, and left eye viewing the right image), a full 3D perception is achieved. The skew of the image stacks can be controlled by adjusting the "offset" in the interface, so stereo pairs can be generated on-the-fly for parallel viewing as well. (a) Zoomed-out view of the full data set is shown (compare to figure 2 b). (b) Zoomed-in view of the central region in (a) is shown.*



(*a*) Memory usage (*b*) Download time

Figure 7: ***Scalability Analysis: Memory and Download Time.*** *See text for details.*

volumetric data over the Internet, without any added computation. We showed that depth and binocular disparity effects can be dynamically generated over the Internet using a combination of HTML, CSS, and JavaScript, directly using the original image stack data. We also showed that these operations are computationally efficient, with a linear scaling property with regard to the number and size of the image slides. We expect our approach to provide an efficient alternative to specialized 3D visualization techniques, and allow dissemination of data to a larger population of researchers.

Acknowledgments

This research was supported in part by NIH/NINDS grant #R01-NS54252. We would like to thank L. C. Abbott, D. Mayerich, and J. Kwon for KESM data acquisition, and J. Keyser for valuable feedback.

References

[AH94] ADELSON S. J., HANSEN C. D.: Fast stereoscopic images with ray-traced volume rendering. In *VVS '94: Proceedings of the 1994 symposium on Volume visualization* (New York, NY, USA, 1994), ACM, pp. 3–9. 2

[All] ALLEN INSTITUTE FOR BRAIN SCIENCE: Allen brain atlas. http://www.brain-map.org/. 6

[BB99] BETHEL W., BASTACKY S. J.: Measurement of perceived objects. In *IEEE Visualization 99, Late Breaking Hot Topics* (1999). 2, 3

[CBY06] CHEN J., BETHEL E. W., YOON I.: Interactive internet delivery of scientific visualization via structured prerendered imagery. *Internet Imaging VII 6061* (2006), 60610A. 2, 6

[Cro07] CROFT J.: Creative use of png transparency in web design. http://www.digital-web.com/articles/web_standards_creativity _png/, May 2007. 5

[CYB08] CHEN J., YOON I., BETHEL W.: Interactive, internet delivery of visualization via structured prerendered multiresolution imagery. *IEEE Transactions on Visualization and Computer Graphics 14*, 2 (2008), 302–312. 2, 6

[EWE99] ENGEL K., WESTERMANN R., ERTL T.: Isosurface extraction techniques for web-based volume visualization. In *VIS '99: Proceedings of the conference on Visualization '99* (Los Alamitos, CA, USA, 1999), IEEE Computer Society Press, pp. 139–146. 2, 6

[Goo] GOOGLE: Google maps. http://maps.google.com. 1

[HK96] HE T., KAUFMAN A.: Fast stereo volume rendering. In *VIS '96: Proceedings of the 7th conference on Visualization '96* (Los Alamitos, CA, USA, 1996), IEEE Computer Society Press, pp. 49–ff. 2

[HMS95] HEIDRICH W., MCCOOL M., STEVENS J.: Interactive maximum projection volume rendering. In *VIS '95: Proceedings of the 6th conference on Visualization '95* (Washington, DC, USA, 1995), IEEE Computer Society, p. 11. 2

[KSTE06] KERSTEN M., STEWART J., TROJE N., ELLIS R.: Enhancing depth perception in translucent volumes. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 1117–1124. 2, 3

[LC03] LO C. H., CHALMERS A.: Stereo vision for computer graphics: the effect that stereo vision has on human judgments of visual realism. In *SCCG '03: Proceedings of the 19th spring conference on Computer graphics* (New York, NY, USA, 2003), ACM, pp. 109–117. 2, 3

[Lei07] LEIN ET AL. E. S.: Genome-wide atlas of gene expression in the adult mouse brain. *Nature 445* (2007), 168–176. 6

[Mal01] MALLOT H. A.: *Computation Vision : Information Processing in Perception and Visual Behavior*. Bradford Book, MIT Press, Cambridge, MA, 2001. Translated from German by J. S. Allen. 3

[MAM08] MAYERICH D., ABBOTT L. C., MCCORMICK B. H.: Knife-edge scanning microscopy for imaging and reconstruction of three-dimensional anatomical structures of the mouse brain. *Journal of Microscopy 231* (2008), xxxx–xxxx (in press). 1, 2

[MGJS*03] MACKENZIE-GRAHAM A., JONES E. S., SHATTUCK D. W., DINOV I. D., BOTA M., TOGA A. W.: The informatics of a C57BL/6J mouse brain atlas. *Neuroinformatics 1* (2003), 397–410. 6

[PHKH04] PROHASKA S., HUTANU A., KAHLER R., HEGE H.-C.: Interactive exploration of large remote micro-ct scans. *Visualization, 2004. IEEE* (10-15 Oct. 2004), 345–352. 2, 6

[Rho97] RHODES G.: Stereo viewing. http://www.usm.maine.edu/ rhodes/0Help/StereoView.html, 1997. 3

[SIJJ07] S.MIKULA, I.TROTTS, J.STONE, JONES E.: Internet enabled high-resolution brain mapping and virtual microscopy. *NeuroImage 35*, 1 (2007), 9–15. 1, 6

[Yag00] YAGEL R.: Volume viewing algorithms: Survey. In *International Spring School on Visualization* (2000). 1