# Hardware-Accelerated Parallel Non-Photorealistic Volume Rendering

Eric B. Lum *  Kwan-Liu Ma *

University of California at Davis

## Abstract

Non-photorealistic rendering can be used to illustrate subtle spatial relationships that might not be visible with more realistic rendering techniques. We present a parallel hardware-accelerated rendering technique, making extensive use of multi-texturing and paletted textures, for the interactive non-photorealistic visualization of scalar volume data. With this technique, we can render a $512 \times 512 \times 512$ volume using non-photorealistic techniques that include tone-shading, silhouettes, gradient-based enhancement, and color depth cueing, as shown in the images on the color plate, at multiple frames second. The interactivity we achieve with our method allows for the exploration of a large visualization parameter space for the creation of effective illustrations.

**Keywords:** interactive visualization, non-photorealistic rendering, scientific visualization, silhouette, texture graphics hardware, visual perception, volume rendering, parallel rendering

## 1 Introduction

Using a brush or pen, skilled artists are able to apply pigments in a manner that create meaningful abstractions of reality. Using a wide range of techniques, an artist can emphasize features that might not otherwise be visible, while de-emphasizing those features of less importance. In the field of non-photorealistic rendering, methods have been developed that attempt to mimic some of the techniques an artist might utilize. Surface rendering techniques include methods for creating illustrations that mimic the use of pen-and-ink [7, 23, 26], including work that uses multi-texturing capability of PC graphics cards for rendering illustrations interactively [22]. Additional research has been performed on paint and brush inspired rendering [1, 20, 15] and visualization [12]. A technique has also been developed for the use of non-photorealistic tone shaded lighting [5]. In addition there has been research on the interactive rendering of surfaces using an efficient set of lines [18].

Non-photorealistic rendering for volumetric data visualization has recently become an area of active research. Treavett and Chen show how pen-and-ink rendering can be applied to volume visualization [24]. Ebert and Rheigans describe how a number of non-photorealistic rendering techniques can be applied to volume rendering [2]. They demonstrate that non-photorealistic methods can enhance features and improve depth perception.

Interactive volume rendering methods give scientists the ability to quickly explore a data set. These methods are often associated with the real time exploration of a volume in the spatial domain, where parameters such as camera position and zoom are varied in the visualization process. However, for volume rendering applications, equally important is the ability to interactively change the transfer function which maps the scalar data in the volume to color and opacity values. The ad hoc nature of transfer function specification makes it an iterative process, where a scientist changes the opacity and color map, renders and examines the volume with these changes, and then repeats the process as necessary. Transfer function specification requires a great deal of fine tuning, making interactivity extremely important.

When non-photorealistic rendering methods are utilized interactivity becomes even more critical. Each non-photorealistic rendering technique adds its own set of addition parameters that must be specified. For example, if tone shading is utilized, for best results the variation in color should be carefully specified with respect to both hue, saturation and value until the result desired by the user is met. Much like transfer function specification there is no "correct" set of parameters that can be utilized for all data sets since these parameters vary widely depending on what type of features the user would like to accentuate or deemphasize. In fact, often the user does not know what type of rendering style is desired, only through experience and experimentation can parameters be found suited for their particular application. In addition, it might be desired to mix both photorealistic as well as non-photorealistic rendering styles when rendering a single volume. This requires multiple sets of transfer functions and lighting parameters, multiplying the number of parameters that must be set and as well as the need for interactivity. Thus, when non-photorealistic rendering is used in a volume rendering context, interactively with respect to viewpoint, transfer function and non-photorealistic rendering parameter space is essential.

Unfortunately, many factors make interactive non-photorealistic rendering extremely difficult. First, the addition of non-photorealistic rendering techniques only adds to the number of calculations required in the rendering process. As a simple example, silhouette rendering requires the additional calculations associated with silhouette detection. Furthermore, the standard technique of rendering lower resolution data or rendering to a lower resolution window to achieve interactivity is often not suited to the needs of the user when selecting non-photorealistic rendering parameters. Non-photorealistic rendering can be used effectively to clarify fine structures in a volume. In order to specify rendering parameters optimized for viewing these structures, the volume must be rendered at high resolutions. For example if a user is trying to accentuate blood vessels in a data set by manipulating the parameters associated with silhouette and gradient, it is necessary that the volume be rendered at its full resolution to a screen resolution high enough to view the vessels clearly.

In this paper we present a method for interactive non-photorealistic volume rendering using hardware accelerated rendering techniques with a PC cluster. We demonostate that using a number of newer features found in modern consumer graphics cards, including 3-D textures, multi-texturing, and paletted textures, it is possible to implement several NPR techniques in hardware such as tone shading, silhouette illustration and depth based color cues. By using multiple graphics cards spread across a PC cluster, we are able to render high resolution volumes at frame-rates interactive enough for the tuning of view, transfer function, and non-photorealistic rendering parameters. The interactivity we achieve makes possible the creation of highly effective non-photorealistic visualizations which would not be possible with less interactive methods.

*Department of Computer Science, University of California, One Shields Avenue, Davis 95616, {lume,ma}@cs.ucdavis.edu

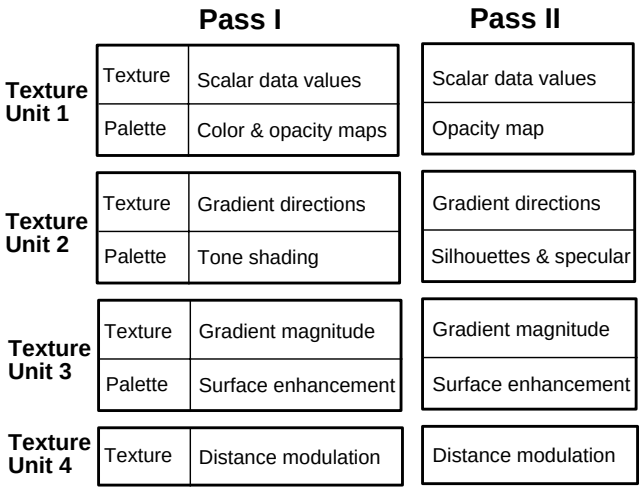



Figure 1: Rendering requires two passes each with 4 texture units.

## 2 Non-Photorealistic Volume Rendering in Hardware

Direct volume rendering can be accomplished by drawing a set of view-aligned polygon slices that sample a 3-D texture containing the volumetric data [25]. Using pixel textures allows a texel value to store coordinates into a second texture. Because scalar and lighting information can be encoded into a texel, transfer function and shading can be changed through the variation of a single texture [21]. Kniss et al. [10], describe how multi-dimensional transfer functions can be interactively specified and rendered with traditional lighting by using multi-textured/multi-pass rendering. Engel et al. [3] use a technique they call pre-integated volume rendering, which also uses multi-textured/multi-pass techniques for improved rendering accuracy with fewer polygon slices.

Our hardware-based volume renderer methods also makes extensive use of the multi-texturing capabilities of modern graphics cards but with the goal of producing non-photorealistic volume renderings. Multi-texturing allows several textures to be combined on a single polygon during the rendering process. By utilizing several separate volumetric textures that store scalar data value, gradient magnitude, and gradient direction, and combining them with properly adjusted color palettes, several different non-photorealistic rendering techniques can be rendered in hardware.

Paletted textures store indices into a color palette that samples the RGBA color space. Through the manipulation of the palette over time, the textures can be varied based of the viewing parameters without changing the data stored in the textures themselves. It is important to use a representation that avoids any manipulation of data that is stored for every voxel since volumetric data sets tend to be large, and traversing the entire volume in software can severely hamper interactivity. Unlike pixel textures that are filtered prior to their texel lookup, paletted texture are filtered after color lookup in RGBA space, permitting non-linear palette mappings to be used with trilinear interpolation in hardware.

We utilize four texture units for each rendering pass, with two passes for every view aligned polygon, as shown in Figure 1. The first pass renders the tone shaded volume, while the second pass contains silhouette and specular contributions. For both rendering passes the four texture units are assigned to the same paletted texture data. The difference between the two passes occurs in the adjustment of the palettes used for each texture.

Each of the four texture units are assigned a different texture. The first texture consists of the original scalar values stored in a paletted texture with 8-bit precision. If the original data set is of higher precision, it is quantized to minimize mean square error using Lloyd-Max quantization [14, 19]. The second stores the normalized gradient direction of each voxel. These directions are encoded in an 8-bit paletted texture, with each direction quantized to one of 240 vectors obtained from the faces of a subdivided version of a combined dodecahedron and icosahedron [25]. The gradient direction information is used for lighting and silhouette operations. A third texture contains gradient magnitudes and will be used for enhancing surfaces. The fourth is a 1-D texture for manipulating color and opacity based on the spatial properties of voxels.

Aside from our emphasis on non-photorealistic rendering, our work differs from previous hardware accelerated volume rendering techniques in our extensive use of paletted textures. Paletted textures permit an arbitrary mapping of gradient direction to color while using only a single texture unit (unlike pixel textures which require two). Arbitrary palette lookups gives flexibility in how gradient direction vectors are used, permitting a wide variety of NPR techniques to be applied, while the use of a single texture unit allows more of these techniques to be combines simultaneously in the limited number of available texture units.

### 2.1 Tone Shading

Lighting can be extremely effective in conveying the shape and structure of an object. Tone shading involves the variation of color temperature or tone to indicate an objects illumination. It is based on the manner that artists often convey lighting through not only the varying of pigment value (intensity) but also through the variation of color temperature or tone [17]. Directly illuminated objects are represented with warmer colors which include yellow, orange and red. This type of lighting is associated with light that comes directly from natural light sources which tend to emit warm light, like candles or the sun. Ambient lighting is typically shown using cooler colors like blue or purple. This is motivated by the fact that reflected light tends to be cooler in nature, like the light received from a blue sky rather than directly from the sun. When considering color temperature it is important to consider relative temperature rather than absolute. For example there can be warm blues and cool reds. Tone shading is accomplished through the variation of relative color tone over an object. Thus ambient lighting is not required to be blue or purple, but instead should simply be cooler than the diffuse lighting.

Gooch et al. [5] use tone shading for the illumination of surfaces. They present a shading model for tone based illumination which allows extreme color values to be reserved for outlines and highlights.

We compute tone based lighting using the gradient direction texture in the second texture unit. As described in the previous section, each index in this texture contains an index into a sampling of a normalized vector space. Each time a spatial viewing parameter is changed, the gradient direction palette is modified. For each index the dot product is calculated between the transformed vector represented by each index, and the light direction. Once the dot product is calculated, the color tone for that product is looked up in a tone shading colormap. The entry found in the colormap is finally stored in that palette for that texture index. The second texture unit is setup to modulate the color from the scalar data in the first texture unit. The second texture unit does not affect the alpha channel, since lighting does not influence opacity.

The user is able to interactively specify the colors used in tone shading by selecting a set of key color entries that are linearly interpolated across the colormap. By manipulating and shifting these key colors across the colormap, lighting parameters can be selected suited for the data set being visualized.

For example, artists typically do not paint smooth Phong shad-

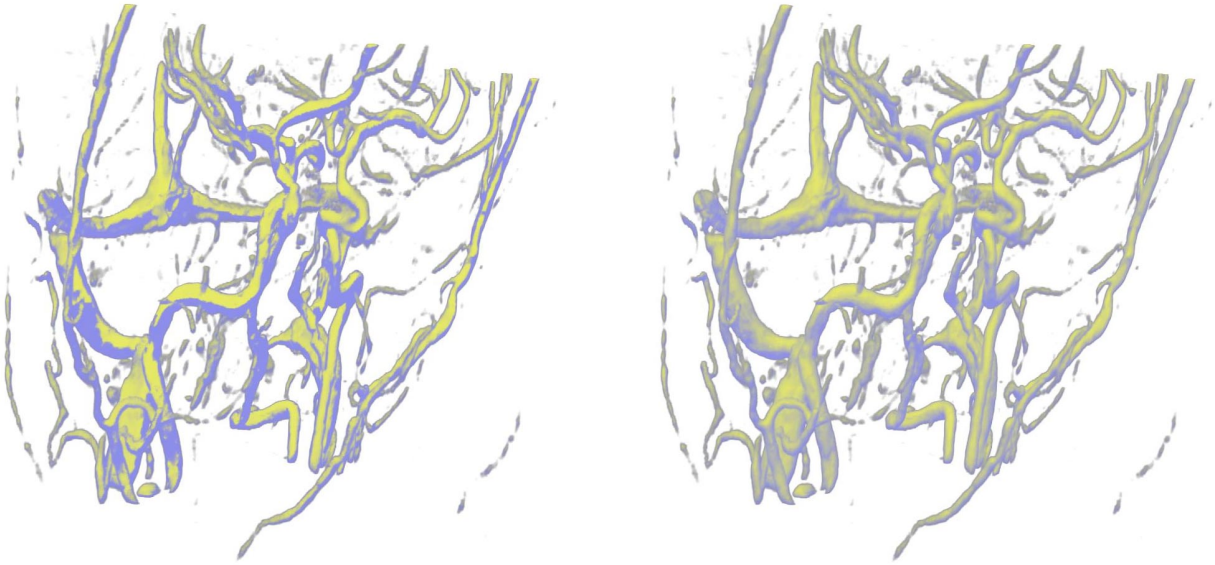


Figure 2: Left: Vessel with abrupt tone shading. Right: smooth tone shading.

ing, but rather use more abrupt lighting transitions. An extreme example of this type of lighting is found in animation cell painting, where painted cells often have two discrete lighting levels, one for ambient lighting and one for diffuse. By making the transition between cool and warm colors relatively short, it is possible to produce this kind of abrupt lighting transition as seen in the image of blood vessels shown in the left image of Figure 2. This can be contrasted with the relatively smooth lighting found in the right image of Figure 2.

The manipulation of the saturation of the colors used in the tone colormap controls the degree tone shading is visible. By making the colors more saturated, the effects of tone shading becomes more subtle permitting more of an objects original color (as specified in the transfer function) to be seen. Manipulation of the saturation and value of the tone colormap, can also be used to control the degree color intensity varies with lighting. For example by using a highly saturated tone colormap with a transition between dark and bright intensities, more traditional lighting can be produced.

### 2.2 Silhouette Illustration

Silhouettes have been utilized for the illustration of surfaces [23, 8, 9] since dark lines drawn around an object can be effective in showing an objects structure. These lines can also be used to help indicate an objects spatial relationship with other objects that consist of similar material. For example, dark silhouettes drawn around overlapping objects can provide depth cues with respect to those objects. This is particularly valuable in volume rendering applications where transfer functions are often set such that objects are semi-transparent sometimes making inner-spatial relationship difficult to determine.

We implement silhouette rendering using a second rendering pass that applies the silhouette and specular contribution. For this pass we once again use the scalar value texture in the first texture unit, and the gradient direction texture in the second. The desired combined texture has an opacity that depends on both the opacity of the voxel from the transfer function as well as degree to which that voxel's gradient is perpendicular to the viewing direction. For this pass, we therefore assign each palette entry in the gradient direction texture with an opacity that is the highest when the transformed gradient vector represented by that index is perpendicular to the viewing direction. We then assign the color for each entry to be the desired color of the silhouette, typically black. The image on the left of Figure 3 shows fine blood vessels that are enhanced using silhouettes. Notice that the spatial relationship between the overlapping vessels near the top of the image are clearer than those found in the image without tone shading at the bottom of Figure 3.

The user is able to vary to what extent the dot product between the gradient and viewing direction influences opacity. For narrow silhouettes the user can specify a rather narrow range of near zero dot products that result in an opaque silhouette. For thicker silhouettes a fairly wide range of dot products can be specified. By varying the opacity with dot product value the user can make the silhouette fade as the objects gradient becomes aligned with the viewing direction.

The same texture is also used simultaneously for specular lighting. For each encoded gradient direction the specular lighting contribution is calculated and added to the red green and blue palette entry. The alpha channel is not adjusted since specular lighting should not affect the opacity of a rendered voxel.

### 2.3 Color Based on Position

Color can be manipulated based on distance to improve depth perception [4]. Aerial perspective has been used by painters to convey depth through the variation of color hue and value based on depth. Typically warmer hues are used for the foreground and become cooler in the background. In addition, color values tend to become lighter and less intense with distance [11].

We implement this technique in hardware by assigning the textured polygons per vertex colors that are modulated with the textures. Each vertex is assigned a color that is varied based on the distance between that vertex and the viewpoint. Colors are interpolated along each polygon using linear interpolation and is then blended with the texturing stage. The depth clues provided by the variation in tone are evident in the left image of Figure 4 where the warmer colored foreground vessels correctly appear to be in front. This can be contrasted with the right image of Figure 4 where the spatial relationship between vessels is less clear. If mipmapped textures are used, the level-of-detail bias of each slice can be varied to

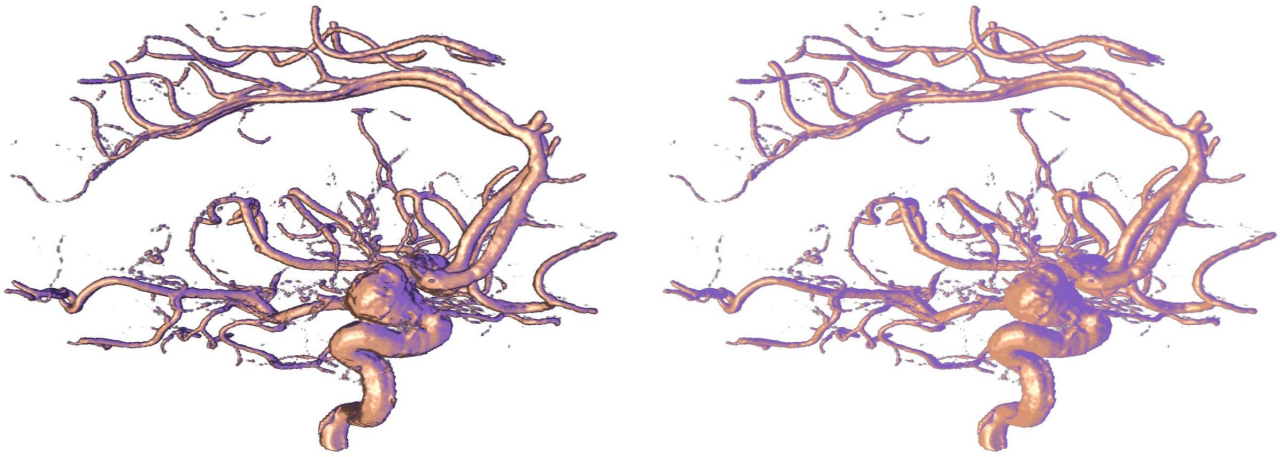


Figure 3: The silhouettes shown on the left image help clarify the spatial relationship between vessels.

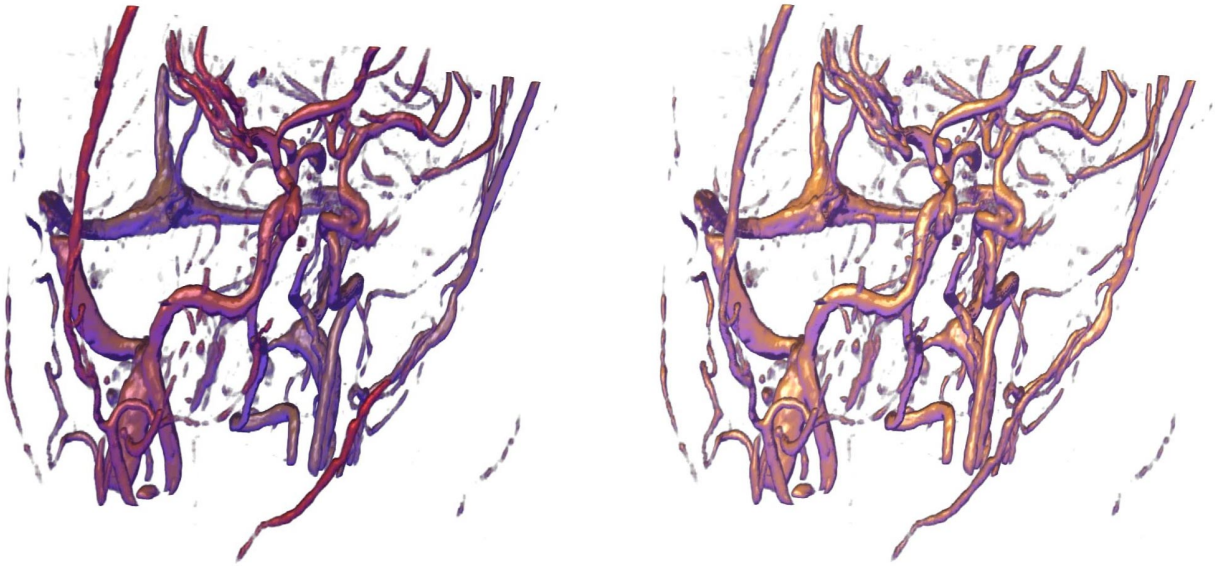


Figure 4: Left: Depth based tone manipulation Right: Without depth based tone manipulation .

manipulate focus for a depth of field effect.

As an extension to depth based color manipulation, we allow the user to perform non-linear color variations along arbitrary directions in the volume. This is accomplished by assigning a 1-D texture to the fourth texture unit that controls how color is varied based on position. The user can specify how color is modulated across this direction in an arbitrary manner, with axis rotation implemented using matrix transformations of the texture coordinates. This technique is particularly effective in manipulating the alpha channel, permitting opacity to be reduced in certain regions that might obscure or detract from the feature the user would like to visualize. Through non-linear fading of the alpha channel along the view direction, closer material can be made more transparent making underlying features more visible, with foreground material still slightly visible to provide context for the features of interest as shown in the bottom right image on the color plate.

### 2.4 Gradient Based Enhancement

Gradients have been used for the enhancement of surfaces in volume rendering applications [13]. Since the transition between features in a volume tend to have the highest gradient magnitude, the enhancement of the opacity in these regions can help to clarify surfaces. The skin surface shown on the left side of Figure 5 is made visible through gradient enhancement, allowing the underlying material to be made visible using a technique described in the next section.

We implement surface enhancement with the third texture unit that is assigned to a gradient magnitude texture. The user has the ability of specifying an arbitrary gradient opacity map that modulates the rendered texel. In cases where one wants to visualize structures that exist between two materials of similar scalar value, enhancing regions with the highest gradient would emphasize the wrong features. By specifying an arbitrary gradient map, one can reduce opacity in these high gradient regions.

The user is able to specify separate gradient enhancing functions for the specular and silhouette rendering pass. Specular lighting and silhouettes are usually associated with surfaces and are less meaningful when applied to the solid semi-transparent regions often associated with direct volume rendering. Thus a map can be set such that specular and silhouette rendering only occurs on the surfaces.

### 2.5 Multiple Transfer Functions

Hauser et al. describe how multiple rendering techniques can be combined when visualizing a single volume to better illustrate different types of objects in a volumes [6]. We provide a similar capability by permitting the user to specify multiple transfer functions, each with its own set of non-photorealistic rendering parameters. Each transfer function can be set to render a different type of object in the volume. By varying the non-photorealistic rendering parameters, and rendering the volume with the multiple transfer functions simultaneously, it is possible to emphasize or deemphasize the different types of objects in a volume.

One result of using multiple sets of rendering parameters is that the parameter space is multiplied in complexity, making interactivity all the more important. Rendering is accomplished by simply applying an extra set of rendering passes for each additional transfer function.

The bottom right image on the color plate shows the use of this technique, where the internals of the Microsoft mouse have been rendered in a non-photorealistic style, while the external plastic parts are rendered more photorealistically. The right image in Figure 5 shows bones rendered non-photorealistically using tone shading and silhouettes. The skin on the other hand is rendered in a more photorealistic style, with a vertical fade that allows the bones in the feet to be completely unobstructed by the flesh and skin. In Figure 6 a piece of coral is shown using non-photorealistic and photorealistic rendering parameters. The smooth transition between parameters sets is accomplished using position-based opacity modulation.

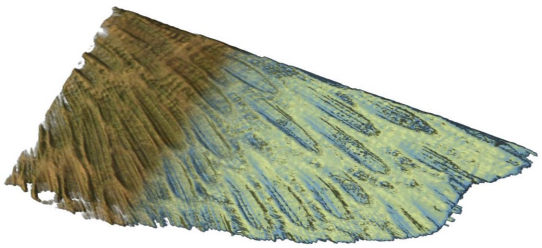


Figure 6: Non-photorealistic and photorealistic rendering styles are mixed in this image of coral.

### 2.6 Combining Techniques

The techniques described in the previous sections can all be combined for hardware assisted non-photorealistic volume rendering. The rendering process requires two passes for every view aligned polygon used to render the volume. For each polygon, the first rendering pass deposits the voxel color looked up from the transfer function, modulated by the tone shading color in the second texture unit, followed by opacity modulation by the gradient enhancing texture in the third unit, with distance based color modulation finally occurring in the fourth unit. Followed by this pass, palettes are reset for specular and silhouette contributions. With this second pass, opacity from the transfer function is utilized in the first texture unit, silhouette and specular contributions are modulate in the second texture unit with gradient based manipulation occurring in the third unit. Finally, distance based opacity modulation occurs in the fourth texture unit.

## 3 A Complete Example

To summarize the process, we use a CT scan of a Microsoft mouse to illustrate the effect of each non-photorealistic rendering technique. As shown in Figure 7(a), when the data set is rendered using only the transfer function without lighting or NPR enhancements it is very difficult to acquire intuition about the spatial structure of each object, particularly with respect to depth.

With addition of tone shading, shown in Figure 7(b), it becomes easier to determine the surface orientations as seen in Figure 7(c). The addition of warmth and coolness to the volumes does not yield distinctly warm or cool colors, but rather results in a variation in relative tone to indicate shape information. Notice however, that there is little variation in color intensity across each volume making it still difficult to gain depth clues as to the spatial interactions of the rendered structures. With the addition of the silhouettes seen in Figure 7(d) the individual structures become much clearer as seen in Figure 7(e). For example, the separation between the two capacitors near the center of the mouse is much more distinct that when only tone shading is used. Next, Figure 7(f) displays depth color cues for the volume, and Figure 7(g) shows the result after the volume is modulated by these cues. This has the subtle effect of adding warmth to the nearer portion of the mouse. Finally in Figure 7(h) we see the result when a second set of rendering parameters is used to render the outer shell of the mouse with a more photorealistic rendering style. Using depth based variation in opacity the closer

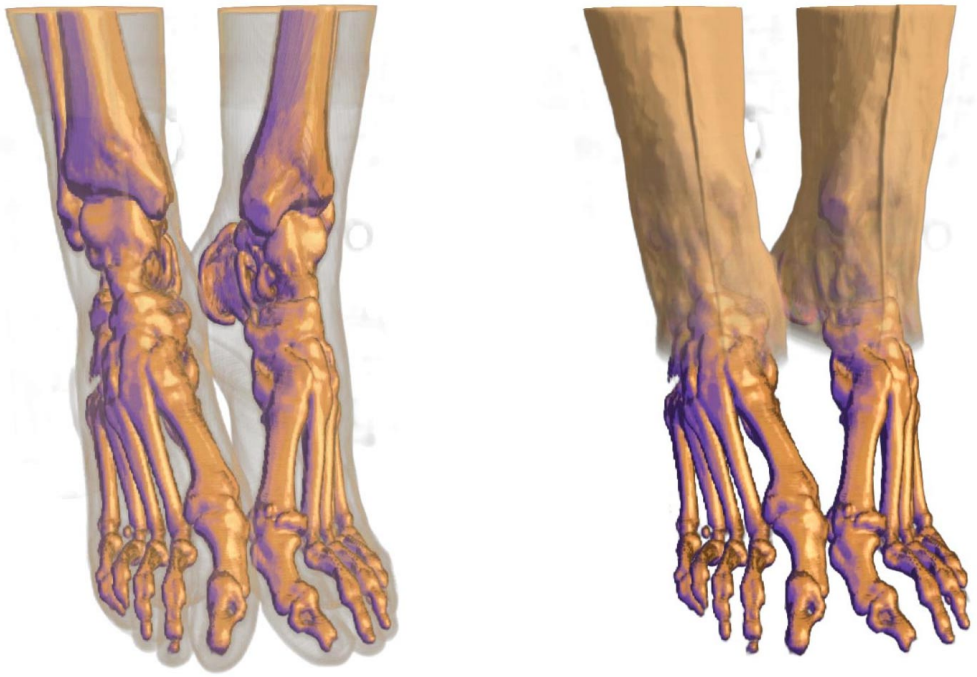


Figure 5: Left: Using gradient based feature enhancement the skin surface is made visible. Right: The skin and flesh are rendered in a more photorealistic style, while the bones are rendered with tone shading and silhouettes.

portions of the mouse are more transparent allowing the inside to be seen.

## 4 Parallel Rendering

One significant limitation of volume rendering using consumer PC graphics hardware is the limited amount of video memory. For example, the Nvidia Geforce 3 has 64 megabytes of video memory that is shared between the frame buffer and texture memory. It is very desirable to fit the volume being rendered entirely in texture memory to avoid having to swap data into the graphics card from main memory over the relatively slow graphics bus. By subdividing the volume spatially and distributing it across a cluster of PCs equipped with graphics cards it is possible to fit significantly larger volumes into the aggregated video memory of the entire cluster. In addition to the larger amounts of texture memory provided by a PC cluster, performance improvements also result from the combined fill-rate of multiple graphics cards.

In our implementation we subdivide and distribute the volume using k-d tree subdivision. Each node resamples it's subvolume to a size of $256\times256\times256$ regardless of the dimensions of the actual volume. For smaller volumes this permits higher order resampling to be done in software prior to the tri-linear interpolation done in hardware. During rendering, for each frame, every node on the cluster renders its subvolume and composites the resulting subimage using binary-swap [16] with the final image being sent to the host for display.

We implemented our technique using a PC cluster with nine computers, each with an AMD Athlon 1.3 Ghz processor, one gigabyte of PC133 SDRAM and a Geforce 3 with 64 megs of video memory. Eight of the computers use 100-Base-T fast Ethernet. The ninth host computer, used for final display and user interface control, has a gigabit Ethernet connection to the cluster's switch.

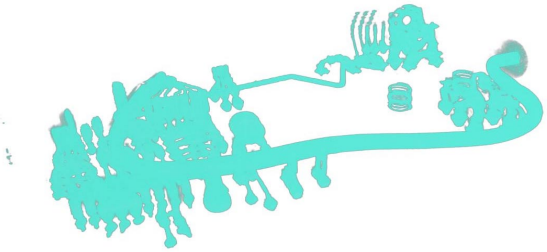
With this low cost cluster we are able to render to a $512\times512$ window, a $512\times512\times512$ volume at about 2.0 frame per second using two rendering passes. If four rendering passes are used to render the volume using two sets of rendering parameters, the frame rates drops to about 1.2 frame per second. Our experimentation showed this framerate is sufficiently high to make possible interactive exploration of rendering parameter space, in particular the variation of transfer function, viewing direction and non-photorealistic rendering parameters. This permits the tuning of parameters for the creation of meaningful images that illustrates specific structures in a volumetric data set. If higher frame rates are desired higher performance can be achieved by rendering to either a lower resolution window and by rendering fewer axis aligned polygons.

## 5 Future Work and Conclusions

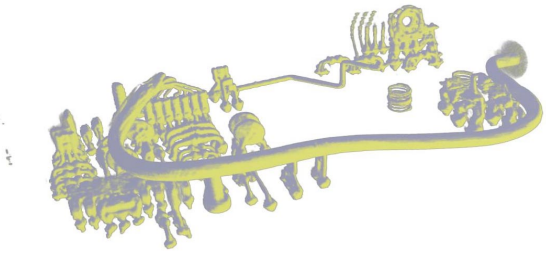
In this paper, we show how a suite of perceptually effective rendering techniques can be efficiently implemented in hardware. We also show how a PC cluster can be used with these techniques for the interactive visualization of large data sets. The interactivity we achieve makes possible fine turning of rendering parameters for the creation of visualizations far more effective than those that would be made with slower software-based methods. Consequently, we foresee increasing use of non-photorealistic rendering in scientific data visualization.

A skilled illustrator uses artistic techniques to produce images with a level of abstraction from the complexities of photorealism. These abstractions tend to be much simpler than reality. Non-photorealistic rendering techniques, however, are often more complex and less efficient than their photorealistic counter-parts despite the level of abstraction in the resulting image. A future area of research is therefore to add the level of abstraction inherent in non-photorealistic rendering into the rendering pipeline for more efficient rendering.

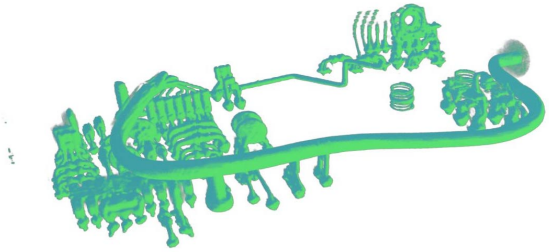
Other future work includes trying to extract and convey even more information from a data set, perhaps using newer hardware features found in the next generation of graphics cards. In addition,



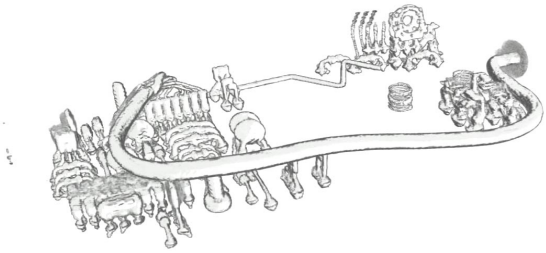
(a) Volume without lighting



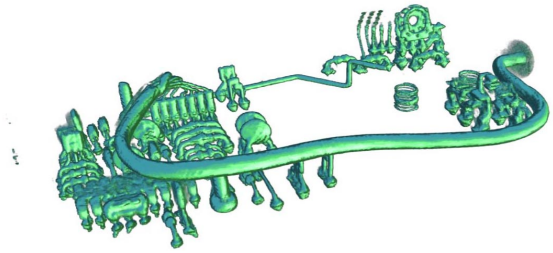
(b) Tone shading contribution



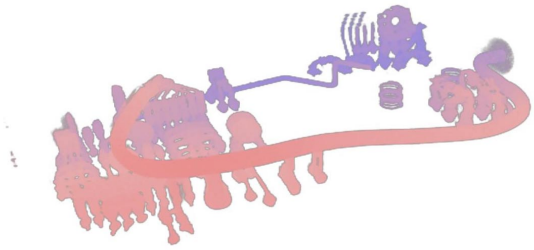
(c) Volume with tone shading



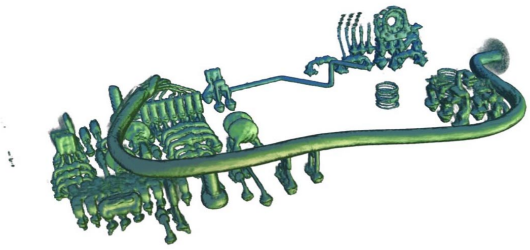
(d) Silhouette contribution



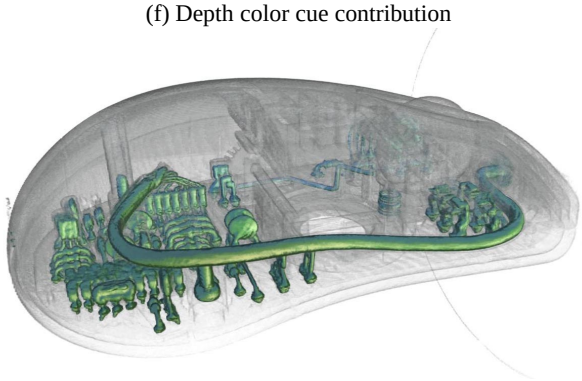
(e) Volume with silhouette



(f) Depth color cue contribution



(g) Volume with depth color cue



(h) Final volume visualization

Figure 7: A complete example of combined techniques using the mouse data set.

we would like to use a wider variety of rendering styles for volume visualization.

## Acknowledgments

This work has been sponsored by the National Science Foundation under contract ACI 9983641 (PECASE Award) and through the Large Scientific and Software Data Set Visualization (LSSDSV) program under contract ACI 9982251. The Los Alamos National Laboratory released the CT mouse data set, which was generated with FlashCT, a software product of Hytec. The authors are also grateful to the University of Utah Medical School, Philips Research at Hamburg, Germany, and the Visible Human Project for providing other test data sets.

## References

[1] CURTIS, C. J., ANDERSON, S. E., SEIMS, J. E., FLEISCHER, K. W., AND SALESIN, D. H. Computer-generated watercolor. In *SIGGRAPH '97 Conference Proceedings* (August 1997), pp. 421–430.

[2] EBERT, D., AND RHEINGANS, P. Volume illustration: Nonphotorealistic rendering of volume models. In *Proceedings of IEEE Visualization 2000 Conference* (October 2000), pp. 195–202.

[3] ENGEL, K., KRAUS, M., AND ERTL, T. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In *Eurographics / SIGGRAPH Workshop on Graphics Hardware '01* (2001), pp. 9–16.

[4] FOLEY, D. J., VAN DAM, A., FEINER, S. K., AND HUGHES, J. F. *Computer Graphics: Principles and Practice*. Addison Wesley, 1996.

[5] GOOCH, A., GOOCH, B., SHIRLEY, P., AND COHEN, E. A non-photorealistic lighting model for automatic technical illustration. In *SIGGRAPH '98 Conference Proceedings* (July 1998), pp. 447–452.

[6] HAUSER, H., MROZ, L., BISCHI, G.-I., AND GROLLER, M. Two-level volume rendering- fusing mip and dvr. In *IEEE Visualization 2000 Conference Proceedings* (2000), pp. 211–218.

[7] HERTZMANN, A., AND ZORIN, D. Illustrating smooth surfaces. In *SIGGRAPH 2000 Conference Proceedings* (August 2000), pp. 517–526.

[8] INTERRANTE, V., FUCHS, H., AND PIZER, S. Enhancing transparent skin surfaces with ridge and valley lines. In *Proceedings of IEEE Visualization '95 Conference* (October 1995), pp. 52–59.

[9] INTERRANTE, V., AND GROSCH, C. Visualizing 3d flow. *IEEE Computer Graphics and Applications 18*, 4 (July/August 1998), 49–53.

[10] KNISS, J., KINDLMANN, G., AND HANSEN, C. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings of IEEE Visualization 2001 Conference* (October 2001), pp. 255–262.

[11] KUNZ, J. *Watercolor Basics: Color*. North Light Books, 1999.

[12] LAIDLAW, D. H., KIRBY, R. M., AND MARMANIS, H. Multivalue data from 2d incompressible flows using concepts from painting. In *Proceedings of IEEE Visualization '99 Conference* (October 1999), pp. 333–340.

[13] LEVOY, M. Efficient ray tracing of volume data. *ACM Transactions on Graphics 9*, 3 (July 1990), 245–261.

[14] LLOYD, S. P. Least squares quantization in PCM. *IEEE Transactions on Information Theory IT-28*, 2 (March 1982), 129–137.

[15] LUM, E. B., AND MA, K.-L. Non-photorealistic rendering using watercolor inspired textures and illumination. In *Pacific Graphics '01 Conference Proceedings* (October 2001).

[16] MA, K.-L., PAINTER, J., KROGH, M., AND HANSEN, C. Parallel volume rendering using binary-swap compositing. *IEEE Computer Graphics & Applications 14*, 4 (July 1994), 59–68.

[17] MACKENZIE, G. *The Watercolorist's Essential Notebook*. North Light Books, 1999.

[18] MARKOSIAN, L., KOWALSKI, M. A., TRYCHIN, S. J., BOURDEV, L. D., GOLDSTEIN, D., AND HUGHES, J. F. Real-time nonphotorealistic rendering. In *SIGGRAPH '97 Conference Proceedings* (August 1997), pp. 415–420.

[19] MAX, J. Quantizing for minimum distortion. *IEEE Transactions on Information Theory IT-6*, 1 (March 1960), 7–12.

[20] MEIER, B. J. Painterly rendering for animation. In *SIGGRAPH '96 Conference Proceedings* (August 1996), pp. 477–484.

[21] MEISSNER, M., HOFFMANN, U., AND STRASSER, W. Enabling classication and shading for 3d texture mapping based volume rending using opengl and extensions. In *IEEE Visualization '99 Conference Proceedings* (1999).

[22] PRAUN, E., HOPPE, H., WEBB, M., AND FINKELSTEIN, A. Real-time hatching. In *SIGGRAPH '01 Conference Proceedings* (2001).

[23] SALISBURY, M. P., ANDERSON, S. E., BARZEL, R., AND SALESIN, D. H. Interactive pen-and-ink illustration. In *SIGGRAPH '94 Conference Proceedings* (July 1994), pp. 101–108.

[24] TREAVETT, S., AND CHEN, M. Pen-and-ink rendering in volume visualisation. In *Proceedings of IEEE Visualization 2000 Conference* (October 2000), pp. 203–209.

[25] VAN GELDER, A., AND HOFFMAN, U. Direct volume rendering with shading via three-dimension textures. In *ACM Symposium on Volume Visualizatrion '96 Conference Proceedings* (1996).

[26] WINKENBACH, G., AND SALESIN, D. H. Computer-generated pen-and-ink illustration. In *SIGGRAPH '94 Conference Proceedings* (July 1994), pp. 91–100.