Interactivity is the Key to Expressive Visualization

I have long believed in the importance of interactivity for effective visualization. Kwan-Liu Ma and his colleagues have been doing very interesting work developing new kinds of interactive visualization techniques, so I asked them to contribute this VisFiles column about their work.

— *Bill Hibbard*

Eric B. Lum and Kwan-Liu Ma
University of California, Davis

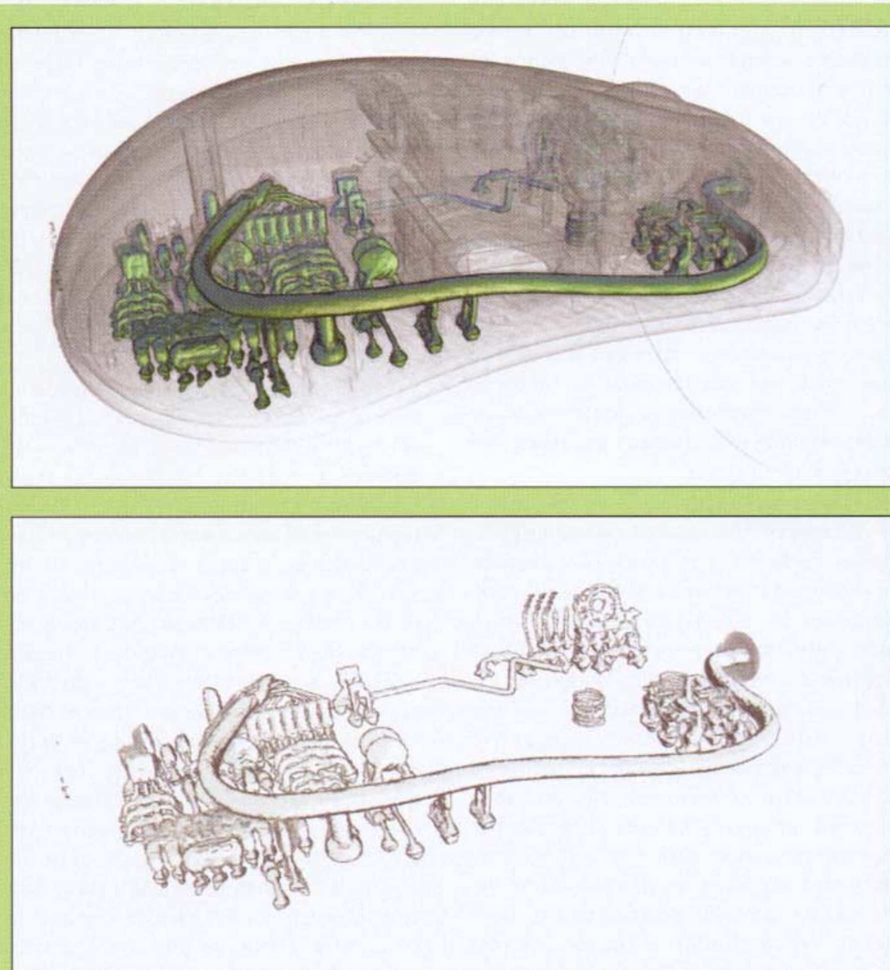

Figure 1: Two non-photorealistic volume renderings of a mouse data set with different sets of rendering parameters. Interactive rendering allows users to select rendering parameters appropriate for their application.

Scientific visualization is widely used for gaining insight into phenomena through data exploration and creating imagery that can be used to illustrate these phenomena to others. Interactive rendering has long been valued in visualization as a means of facilitating more effective exploration. More recently, non-photorealistic rendering (NPR) has been applied to scientific visualization, which consists of using artistically inspired techniques for the creation of more expressive visualizations [3, 4, 5].

In many cases, NPR has been shown to be more effective than photorealistic rendering in communicating subtle information about physical structures or phenomena. How the user chooses to portray a data set can have a significant effect on how accurately and efficiently a visualization communicates the information the user seeks to reveal. It is our belief that when NPR is made interactive, particularly with respect to the control of how NPR is applied, the user is able to more quickly derive expressive visualizations.

Interactivity is often associated with spatial exploration, where parameters such as position, zoom and light direction are varied over time. The resulting animations, controlled by the user, allow further insight to be gained in the subject being viewed. In visualization, interactive rendering includes giving the user the ability to change other rendering parameters related to the data itself. One example is the transfer function in volume rendering, which can be adjusted to change voxel classification to emphasize different aspects of a data set.

NPR techniques typically have a set of rendering parameters associated with them which dictates the style of the resulting images. The required tuning of these parameters does not make these algorithms less desirable, but rather gives the user the ability to create the types of images they seek. In scientific visualization, there is no single set of "correct" rendering parameters as seen in the two images shown in Figure 1. Rendering parameters can be selected that add emphasis and clarity to the aspects of the visualization the user is interested in.

In visualization, the user is seldom an artist, but is often a scientist who would like to generate images that illustrate a particular structure or phenomenon they are studying. Thus, the user cannot be expected to know which NPR techniques are appropriate, and might not even have a clear vision of what the resulting visualization should look like. Much like transfer function specification, the selection of NPR parameters is an iterative process, where the user changes some parameters, views the result and then changes the parameters further.

If the users can see the results of changes in sub-second times, it becomes more intuitive for them to select appropriate parameters for their task. Thus, changes in view and object position are only one aspect of interactive rendering. It is equally important to give the user the interactivity necessary to explore the rendering parameter space.

Unfortunately, many factors make interactive NPR difficult. First, the addition of NPR techniques typically adds to the amount of calculation required in the rendering process. As a simple example, silhouette edge rendering requires the view-dependent operation of determining which edges are on the silhouette. Furthermore, the standard technique of rendering data at lower resolution or rendering to a lower resolution window to achieve interactivity is often not suited to the needs of the user when selecting NPR parameters. NPR can be used effectively to clarify fine structures. In order to specify rendering parameters optimized for viewing these structures, objects must be rendered at a high enough resolution for them to be visible, resulting in increased processing and storage requirements.

For interactive rendering there has been a trend toward the use of consumer PC graphics cards, many of which have capabilities exceeding those found in high-end workstations of just a few years ago. With each product iteration these cards have improved performance, increased memory capacity and added new features supported in hardware. Many of the more recent interactive rendering algorithms load data (geometry and texture) into video memory and then perform rendering calculations on the graphics card to exploit the high performance memory and processor found on the card, and avoid the I/O costs of sending large amounts of data from main memory across the graphics bus to the graphics card for every frame. Current consumer graphics cards have up to 128 megabytes of memory and support programmable vertex operations as well as the customizable combining of textures. These new capabilities prompt us to consider non-photorealistic techniques that were too expensive to use just a few years ago.

The interactivity made possible by today's faster CPUs and graphics cards also permits the development of entirely new visualization algorithms which take full advantage of this performance. One example is a technique which we call kinetic visualization [7], which uses motion as a means for providing supplemental shape cues for the visualization of static data. The technique is inspired by ideas from kinetic art and is non-photorealistic in the sense that motion is used to view structures that are inherently static.

Non-Photorealistic Volume Rendering

Strictly speaking, traditional volume rendering is not necessarily a form of photorealistic rendering, since it often involves the visualization of values such as temperature, pressure and density that are not directly visible in reality. Nevertheless, by applying artistically inspired NPR methods to volume rendering, creating more effective visualizations is possible. Ebert and Rheingans [1] describe a software approach for non-photorealistic volume rendering using techniques that include silhouette edge rendering, cool-to-warm shading and depth-cued color variation. In our work [6] we show how texture hardware on consumer PC graphics cards can be used for interactive volume rendering with a number of non-photorealistic techniques. It is also possible to utilize a cluster of graphics card equipped PCs to achieve interactive NPR of large volumes.

Volume Rendering with Graphics Hardware

Direct volume rendering can be accomplished by drawing a set of view-aligned polygon slices that sample a 3D texture containing the volumetric data [9]. Multitexturing 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 dynamically adjusted color palettes, several different NPR techniques can be rendered interactively using graphics hardware. The Nvidia Geforce3 has four texture units, which means a total of four textures can be blended onto a single polygon. For some effects, additional rendering passes can be added, which consist of blending additional textured polygons on top of the previously rendered polygons in the frame buffer.

Paletted textures store indices into a color palette that samples the RGBA (red, green, blue, alpha) color space. Paletted textures are well suited for storing a volume's original data samples and gradient magnitudes since these are scalar values and not colors to begin with. In addition, these values are typically rendered with some type of lookup table palette (transfer function in the case of data values). Paletted textures can also be used to store normalized gradient directions by quantizing each vector direction into some finite set directions. Thus, each possible index represents a sample in a finite set of normalized directions. Through the manipulation of the palette over time, paletted textures can be varied based on transfer function or the viewing parameters without changing the data stored in the textures themselves. It is important to use a representation like this which avoids any manipulation of data that is stored for every voxel since volumetric data sets tend to be large, and traversing an entire volume in software can severely hamper interactivity.

Non-photorealistic Illumination

Artists often illustrate lighting by not only varying pigment value (intensity) but also through the variation of color temperature. Directly illuminated objects are represented with warmer colors which include yellow, orange and red. Gooch et al [3] use cool-to-warm shading for the illumination of surfaces. They present a shading model which allows extreme color values to be reserved for outlines and highlights.

The cool-to-warm shading described by Gooch et al can be implemented by modulating each voxel color with a lighting texture that manipulates color temperature. As described in the previous section, a paletted normal direction texture can be used, with each texel containing an index into a sampling of a normalized vector space. The palette for this texture is created by first calculating the dot product between each possible normal direction and the light direction. Once the dot product is calculated, the color temperature variation for that product is looked up in a shading colormap specified by the user.

The manipulation of the saturation of the colors used in the shading colormap can control how much temperature variation in shading is visible. By making the colors more saturated, the effects of this type of shading becomes more subtle permitting more of an object's original color (as specified in the transfer function) to be seen. Furthermore, by changing color value across the shading colormap more traditional lighting with variation in color intensity can be produced.

Silhouette Edges

Silhouette edges have been utilized for the illustration of surfaces since dark lines drawn around an object can be effective in showing an object's structure. Silhouette edge rendering can be particularly useful in volume rendering applications because transfer functions are often set such that objects are semi-transparent, sometimes making spatial relationship difficult to determine. Silhouette rendering can be accomplished by using the paletted gradient direction texture with the color palette adjusted such that voxels with gradient perpendicular to the view direction are modulated to black.

Position Color Cues
Color can be manipulated based on distance to improve depth perception [2]. Aerial perspective has been used by painters to convey depth through the variation of color hue and value based on depth. Often warmer hues are used for the foreground and become cooler in the background. This can be implemented in hardware using an 1D texture that modulates the color of the rendered volume along some direction.

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 right image in Figure 1. Figure 2(f), on the other hand, shows a fading of opacity based on vertical position.

Multiple Rendering Styles
Sometimes it is desirable to vary the rendering styles for different parts of a volume. This can be done by permitting the user to specify multiple transfer functions, each with its own set of NPR parameters. Each transfer function can be set to render a different type of object in the volume. By varying the NPR parameters, and rendering the volume with the multiple transfer functions simultaneously, it is possible to emphasize or de-emphasize 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.

A Complete Example
To summarize the process, we use a CT scan of feet to illustrate the effect of each NPR technique. Cool-to-warm shading is shown in Figure 2(a) and when combined with the original colors defined by the transfer function result in the image seen in Figure 2(b). 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 temperature 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 edges seen in Figure 2(c), the individual structures become much clearer as seen in Figure 2(d).

Finally in Figure 2(e) and Figure 2(f) we see the result when a second set of rendering parameters is used to render the flesh material with a more conventional rendering style. Using position based variation in opacity the lower portions of the feet are more transparent in Figure 2(f) making the bones more clearly visible.

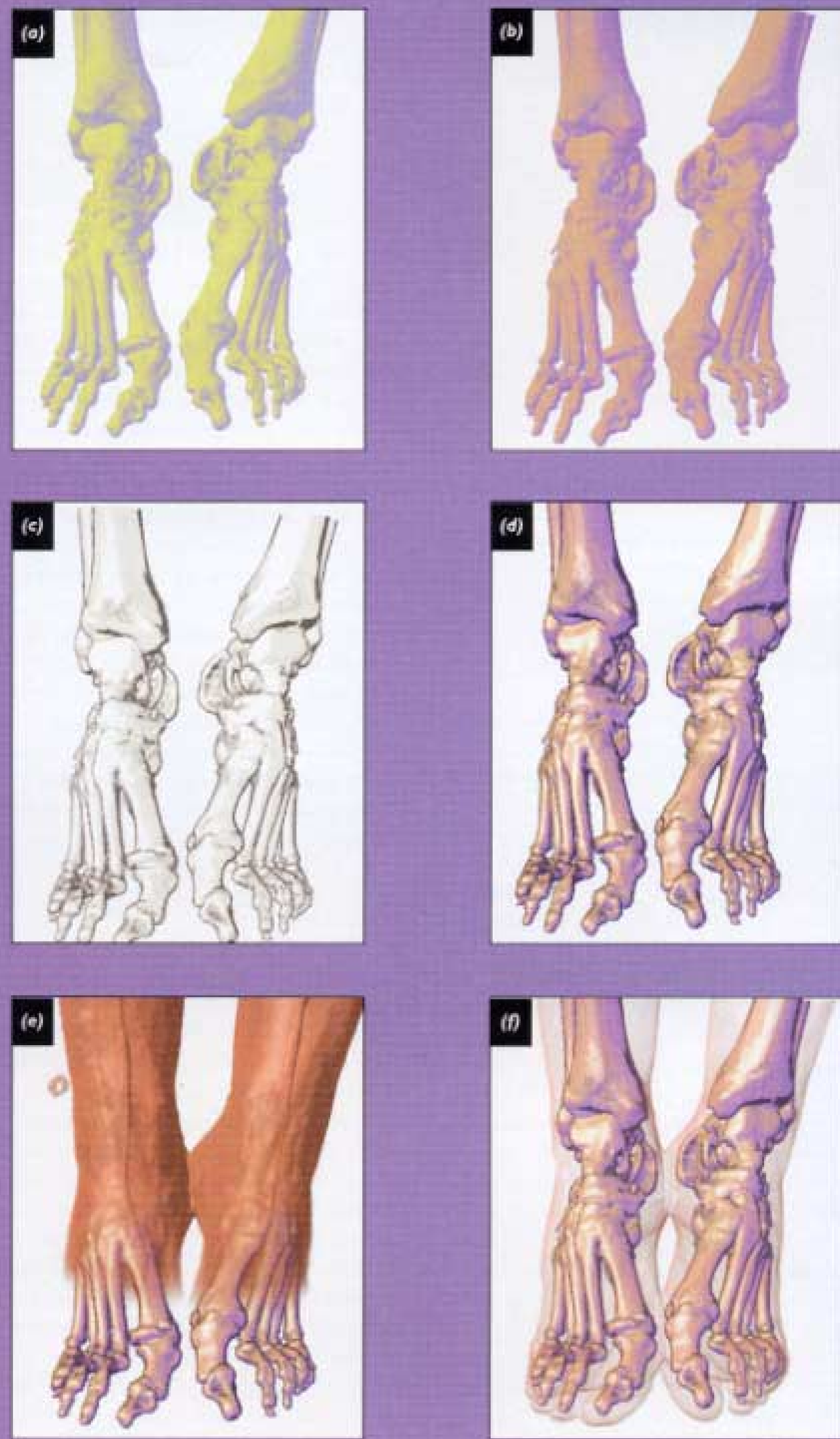

Figure 2: *A complete example of the different NPR techniques that can be combined during interactive rendering of a feet data set. (a) cool-to-warm shading contribution; (b) volume with cool-to-warm shading; (c) silhouette contribution; (d) volume with silhouette; (e) final volume visualization; (f) final volume visualization.*

Kinetic Visualization
Motion provides strong visual cues for the perception of shape and depth, as demonstrated by cognitive scientists and visual artists. The kinetic visualization technique we have developed uses particle systems to add supplemental motion cues which can aid in the perception of shape and spatial relationships of static objects. Based on a set of rules following perceptual and physical principles,

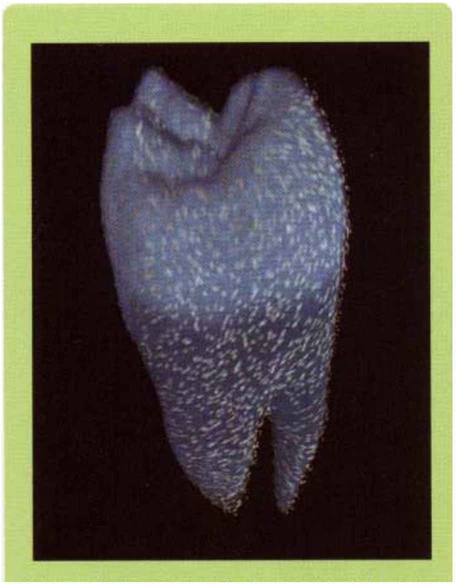

Figure 3: Kinetic visualization uses particles that flow over an object's surface to illustrate shape. In this case kinetic visualization is mixed with a traditional volume rendering of a tooth data set.

particles flowing over the surface of an object bring out shape information of the object that might not be readily visible with conventional rendering. By giving the user interactive control over how these rules are applied, animations can be tuned to attract attention to and clarify specific aspects of a visualization. The method is not meant as a replacement for traditional rendering techniques that use lighting and changes in viewpoint to show structure, but instead can be used to provide supplemental shape cues that aid in perception.

Figure 3 shows the particles used in kinetic visualization. In this case, the particles are combined with traditional volume rendering to illustrate the surface of a tooth data set. Note that since kinetic visualization relies exclusively on motion cues, a static image does not illustrate the technique itself.

The particles used in kinetic visualization are governed by a set of rules designed to result in particles that indicate shape by smoothly flowing over an object, with locally consistent directions, and a density distribution that does not let particles "clump" together in regions of little interest. The motion of particles along an object's surface helps improve shape perception over time by presenting the viewer with a set of vectors (trajectories) that run parallel to a surface. Therefore particles are given velocities that favor motion along a surface or perpendicular to gradient direction in a volumetric case.

Interrante [4] describes how line integral convolution along the principal curvature directions can generate brush-like textures that create perceptually intuitive visualizations since the resulting textures "follow the shape" of the object being rendered. Similarly we use principal curvature directions to create particles that "follow the shape" of a surface. Particle velocities are adjusted so the particles flow in a direction that favors one of the principal curvature directions chosen by the user.

Since particles moving in opposite directions can distract the user and make their motion less clear, rules can be imposed to give the particles more consistent behaviors. One way this can be achieved is to give the particles flock-like behavior as described by Reynolds [8]. Flocks exhibit motion that are fluid, with each member still exhibiting individual behavior. Thus flocking can be used to add local uniformity to particle motion while still allowing particles to have motion shaped by outside forces like principal curvature direction.

Several animation sequences generated with kinetic visualization can be downloaded from **http://www.cs.ucdavis.edu/~ma/papers/kinvis.** The animations show particles that flow over surfaces and help to clarify ambiguous structures. Particles are rendered in hardware as textured polygons and can be combined with other visualization methods like hardware accelerated volume rendering. Using graphics hardware and today's faster CPUs provides the interactivity necessary for any fine tuning of motion parameters that might be desired by the user.

Conclusion

Interactive fine turning of NPR parameters can allow for the creation of visualizations far more effective than those that would be made with slower software-based methods. Consequently, we foresee increasing the use of NPR in scientific data visualization.

There are a number of areas where future work can be devoted. Working closely with scientists in different fields, interactive NPR can also be applied to a wider range of visualization tasks. New NPR algorithms can be developed to take advantage of the new features that will be found in the next generation of graphics hardware. By developing new rendering algorithms that mimic a wider range of artistic styles, even more effective visualizations can be created.

References

1. Ebert, D. and P. Rheingans. "Volume Illustration: Non-Photorealistic Rendering of Volume Models," *Proceedings of IEEE Visualization 2000 Conference*, pp. 195-202, 2000.
2. Foley, D.J., A. van Dam, S.K. Feiner and J.F. Hughes. *Computer Graphics: Principles and Practice*, Addison Wesley, 1996.
3. Gooch, A., B. Gooch, P. Shirley and E. Cohen. "A Non-Photorealistic Lighting Model For Automatic Technical Illustration," *SIGGRAPH 98 Conference Proceedings*, pp. 447-452, 1998.
4. Interrante, V. "Illustrating Surface Shape in Volume Data via Principal Direction-Driven 3D Line Integral Convolution," *SIGGRAPH 97 Conference Proceedings*, pp. 109-116, 1997.
5. Laidlaw, D.H., R.M. Kirby and H. Marmanis. "Multivalue Data from 2D Incompressible Flows Using Concepts from Painting," *Proceedings of IEEE Visualization '99 Conference*, pp. 333-340, 1999.
6. Lum, E.B. and K.L. Ma. "Hardware-Accelerated Parallel Non-Photorealistic Volume Rendering," *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering* (NPAR 2002), June 3-5, 2002.
7. Lum, E.B., A. Stompel and K.L. Ma. "Kinetic Visualization - A Technique for Illustrating 3D Structures and Shapes," (under review, 2002) **http://cs.ucdavis.edu/~ma/papers/kinvis**.
8. Reynolds, C.W. "Flocks, Herds, and Schools: A Distributed Behavioral Model," *SIGGRAPH 87 Conference Proceedings*, pp. 25-34, 1987.
9. Van Gelder, A. and U. Hoffman. "Direct Volume Rendering with Shading via Three-Dimension Textures," *Proceedings of the ACM Symposium on Volume Visualization*, 1996.

About the Guest Columnists

Eric B. Lum and Kwan-Liu Ma
Department of Computer Science
University of California, Davis

About the Columnist

Bill Hibbard's research interests are interaction techniques, data models and distributed architecutures for numerical visualization. He leads the SSEC Visualization Project and is primary author of the Vis5D and VisAD systems. He has degrees in mathematics and computer science from the University of Wisconsin - Madison.

Bill Hibbard
Space Science and Engineering Center
1225 W. Dayton Street
Madison, WI 53706
Tel: +1-608-253-4427
Fax: +1-608-263-6738
Email: **hibbard@facstaff.wisc.edu**
Web: **http://www.ssec.wisc.edu/~billh/vis.html**