

# Visualizing Bioelectric Fields

Robert S. MacLeod, Christopher R. Johnson, and Mike A. Matheson  
*University of Utah*

When you think of visualization in medicine, usually X rays, magnetic resonance images, and computed tomography scans come to mind. The information from these techniques can be displayed as 2D images or combined to create a 3D database for viewing with volumetric rendering schemes. Whatever the method, the goal is almost always the same: to visualize the anatomical structure of some part of the body. However, just as medicine is more than anatomy, the field of medical visualization encompasses much more than displaying anatomical structure.

Bioelectricity refers to electricity generated by the cells of a living organism. For example, quiescent tissues of the brain and muscle maintain potential differences of some 50 to 100 mV across the membrane enclosing each cell. When these cells are electrically or chemically stimulated beyond a threshold, charged ions move through pores in the membrane. As a result, the potential difference breaks down momentarily, only to be restored by energy-consuming regenerative processes. In fields such as cardiology and the neurosciences, registration and visualization of the resulting voltages and currents are of special interest because this bioelectricity is the source of subsequent movement, action, or thought.

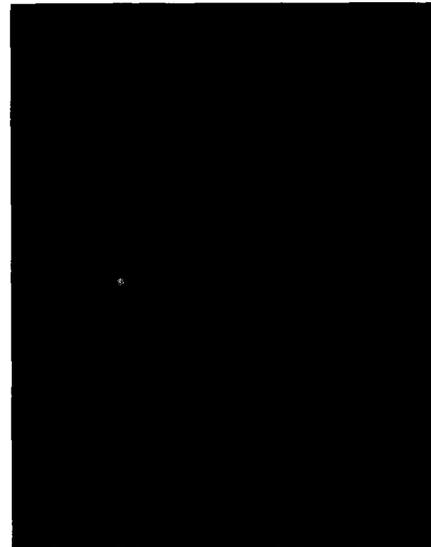
Here we describe some applications of visualization to cardiac electrophysiology—the study of the heart’s bioelectric activity.

We wanted to equip ourselves with the tools needed to display the results of our experimental and modeling research. The paucity of appropriate software led us to develop our own visualization toolkit.

## Making models

Geometric models play a crucial role in the analysis, manipulation, and visualization of cardiac bioelectric data. Experimenters record electric potentials (voltages) from arrays of 32 to 1,300 electrodes distributed on the heart or chest surface or within the

**Figure 1. Voltage distribution over the heart for a single instant in time, as recorded with a 64-electrode “sock” array. Rendering was done in real time using hardware Gouraud shading. Red shades indicate positive potential; green, negative potential; and gray, the range of values near zero. Interpolation was bilinear over the relatively coarse recording mesh, which provides smooth shading while retaining important information about electrode location. The red tubes represent the coronary arteries digitized with the heart geometry, and the spheres mark locations of stimulus electrodes and occlusions of the arteries.**



volume of the heart or thorax. To display the resulting voltage and current distributions, we must determine the electrode locations and from them construct polygonal models. In computer modeling studies, we need not only the electrode locations, but also accurate representations of the surfaces and volumes of the heart and the surrounding thorax. Of particular interest are models that describe the relationship between voltage on the surface of the heart and voltage and electric current throughout the thorax (the “forward and inverse problems in electrocardiology”). Solutions to these problems would provide cardiologists with valuable information about the state of a patient’s heart from noninvasively obtained electrocardiograms and would also assist in the development of better cardiac defibrillators.

For visualization, we first needed tools to create, display, and interactively edit geometric models of the heart and torso. Data for such models come from sources as diverse as histological slices, mechanical measurements, laser surface-scans, and nuclear magnetic resonance images. The resulting sets of 3D points must then be connected to form polygonal

elements. While this is a mesh-generation problem common to many areas of engineering, our grids are very irregular and contain nodes that cannot be shifted to accommodate algorithms based on a divide-and-conquer strategy. We therefore developed semiautomatic algorithms for constructing the triangular surface and tetrahedral volume elements, together with interactive graphics programs that allowed us to display and manually correct the results. The size of the finished geometric models ranged from 32 to 80,000 nodes and from 100 to 80,000 surface elements with more than 500,000 volume elements.

## Displaying the data

The needs of a scientist trying to scan a large volume of data as efficiently as possible differ radically from those of that same scientist presenting a selected sample of the data to others. The two separate data visualization systems we developed reflect these sometimes contradictory goals.

For quick, flexible viewing of spatially distributed data, we wrote an interactive program—Map3d—based on Silicon Graphics’ GL graphics library and designed to run on SGI and



**Figure 2.** Photograph of an electrolytic tank shaped like a human thorax. Electrodes fixed to the surface of the tank record electric potentials from cardiac sources suspended in the tank.

**Figure 3.** An example from Map3d of the potential distribution for a single instant in time on the surface of the torso tank shown in Figure 2. Rendering was accomplished using hardware Gouraud shading over the 1,250 triangles that connect the 650 nodes of the tank model. Potentials were recorded from 192 electrodes, shown as yellow spheres in the figure, and interpolated using a Laplacian smoothing algorithm.



IBM workstations.<sup>2,3</sup> Map3d employs multiple display windows in which we can display one or more "metasurfaces." The minimal content of a metasurface is a set of points that we can connect to form polygons (line segments, triangles, tetrahedra, or cubes). We can display information about any point or polygonal element and interactively edit the connectivities of the polygons.

The geometry of the metasurface then forms the spatial structure on which we can display the measured or computed voltage and current data. The data most often take the form of multivariate time series, with a scalar or vector value associated with each node or polygon of the geometry. It is in the rendering and scaling of these data that Map3d is most specific to the field of cardiac bioelectricity and also most flexible. The display design started with some quasi-standards that have evolved for this type of data: color, shading model, scaling scope and model, and contour construction. To these standards we added many options and adjustable parameters, selectable from menus. The power of this approach lies

in presenting a familiar, standard view that we can then modify through a restricted set of options derived from experience with similar data. Paradoxically, the infinite variety of often irrelevant options offered by many general-purpose programs intimidates and even irritates scientists and restricts the software's utility.

At any time during the viewing session, Map3d saves the current display settings in a Unix resource file (.map3drc) for later reuse. Figures 1 and 3 contain examples of Map3d's screen display. Figure 2 shows the electrolytic tank used to obtain the data for Figures 3 and 4.

Rendering distributed time-series data on a geometrical model provides valuable spatial information but hides the temporal relationships from direct view. In Map3d we can quickly step through the frames of the sequence or select any point in the geometry with the mouse and obtain a window containing the annotated time signal for that point. When we view multiple data sets simultaneously, Map3d permits synchronization of separate instances of the program through Unix message

passing. Thus all rotations and translations of the geometry, or changes in the data frame, are applied to all instances simultaneously.

Although Map3d provides a powerful means of scanning large amounts of data, its rendering models are primitive, and the level of detail in the image is traded for interactive performance. To display selected frames of data, or prepare presentation-quality images, we developed a second set of programs, based on ray-traced rendering and distributed computing. Since ray tracing is so computation intensive, individual frames are generated in a distributed batch mode and viewed as a sequence afterwards. The complete visualization system consists of a database manager that provides consistent handling of a variety of input data types and computers; an interface manager that controls feedback to the user; a visualization manager responsible for the creation of all images; and an animation manager that automates the creation of animated sequences. To efficiently use decentralized computational resources, we designed the system to function using shared memory and distributed sys-

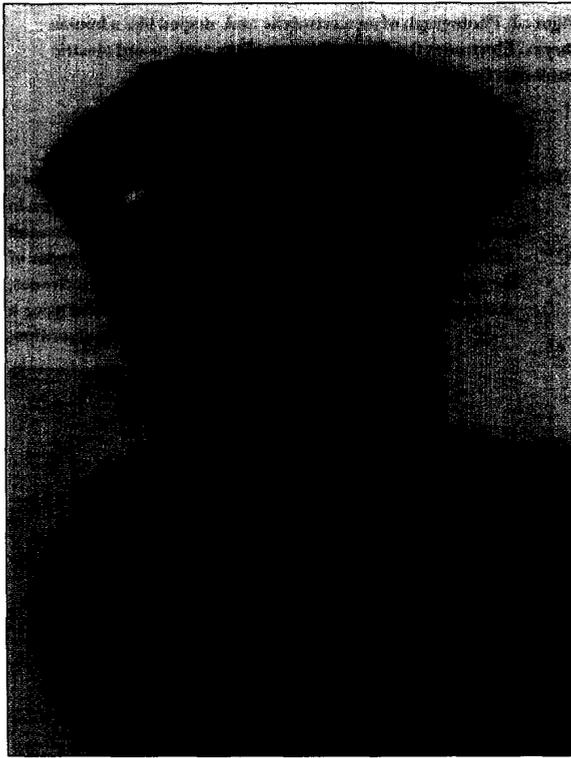


Figure 4. Another sample of potential distributions recorded from the surface of the torso tank. Rendering here was done using ray tracing over the same geometry as in Figure 3. A stylized base was added to mimic the physical structure of the tank.

Figure 5. A ray-traced composite image of the geometry and current flow in a multichambered model of the human thorax. The model was created from MR images and contained more than 22,000 nodes and 126,000 tetrahedra. Shown are segments of the body surface, the boundary between subcutaneous fat and muscle, the heart, and the lungs. The colored loops represent the flow of current through the thorax for a single instant in time, computed from potentials recorded from the surface of the heart during open chest surgery.



tems. We used this visualization system to produce striking images like those shown in Figures 4 and 5.

### The programming dilemma

We were driven to produce our own visualization software because, as the project began, the programs available to us were neither sufficiently powerful nor flexible enough for adaptation to the field of cardiac electrophysiology—especially when compared to the tremendous capabilities of existing hardware. It is worth noting that some of the ideas for the visualization tools we developed came not from medical imaging but from other engineering disciplines, such as computational fluid dynamics and geographical information systems. However, most of the ideas came from our own experience and that of our colleagues, often in the form of examples of what *not* to do, based on previous encounters with existing general-purpose scientific visualization systems.

With Map3d, our major goal was to maximize interactive performance so that we could manipulate large finite element meshes and multivariate data sets. To do this, we sacrificed elaborate shading and lighting features. The batch

mode graphics, in contrast, we developed to generate high-quality rendered images of selected data. This combination of visualization programs has allowed us to generate and manage some of the most complex and elaborate models of the human thorax in this field and to produce uniquely revealing images of our results. Although the costs of developing the code ourselves have been significant, the level of control and flexibility we achieved has proved crucial. □

### Acknowledgments

This research was supported in part by awards from the Nora Eccles Treadwell Foundation and the Richard A. and Nora Eccles Harrison Fund for Cardiovascular Research, by the Heart and Stroke Foundation of Canada, by the Whitaker Foundation, by the National Institutes of Health, and by a grant for computer time from the Utah Supercomputing Institute, which is funded by the State of Utah and the

FSC (Federal Systems Company) Houston of the IBM Corporation.

### References

1. C.R. Johnson, R.S. MacLeod, and M.A. Matheson, "Computer Simulations Reveal Complexity of Electrical Activity in the Human Thorax," *Computers in Physics*, Vol. 6, No. 3, May/June 1992, pp. 230-237.
2. R.S. MacLeod, C.R. Johnson, and M.A. Matheson, "Visualization Tools for Computational Electrocardiography," in *Proc. Visualization in Biomedical Computing 1992*, R.A. Robb, ed., Vol. 1808, SPIE, Bellingham, Wash., 1992, pp. 433-444.
3. R.S. MacLeod, C.R. Johnson, and M.A. Matheson, "Visualization of Cardiac Bioelectricity—A Case Study," in *Proc. IEEE Visualization 92*, IEEE CS Press, Los Alamitos, Calif., 1992, pp. 411-418.