Visualization in Science and Medicine

Bill Ribarsky
Scientific Visualization Group
and
Graphics, Visualization, and Usability Center

This Paper to be Published in the

Georgia Institute of Technology
Atlanta, Georgia 30332

ccsupwr@hydra.gatech.edu
VISUALIZATION IN SCIENCE AND MEDICINE

Martin W. Ribarsky
Georgia Institute of Technology, Atlanta, GA 30332-0710

Key Words: Data, scientific, visualization, multivariate, complexity, interactivity, representation

After a very gradual 20 year development, the discipline of scientific visualization was given form and focus (and its name) by the seminal 1987 report, "Visualization in Scientific Computing". Since then the methods, means, and results of scientific visualization have grown and diversified so that now one can travel from SIGGRAPH to Visualization '91 to this conference and see different (and also lots of similar) effective graphical approaches for revealing the meaning in data. This growth, of course, has paralleled and been fed by the growth in computing technology, software development, and data accumulation. And because long-term development in computing technology and data generation should continue at the current pace, scientific visualization will also continue to diversify and increase in importance.

In spite of its title, this paper does not purport to cover, even in a general sense, the whole of visualization in science and medicine (a rather large undertaking). Instead, it will present a few examples of the use of visualization in these research areas at Georgia Tech that show the breadth of representation necessary in this discipline. We will use these examples to suggest some key elements that must be present if scientific visualization is to succeed in overcoming the computing and analysis challenges of the next several years. And we will then outline the research we are undertaking to provide those key elements.

Examples of Visualization in Science

The examples we will present are from the area of molecular dynamics (MD) simulations. This is an area in which little development of visual representations for analysis has been done. Although the methods of molecular modeling, which are quite rich, are applicable to the visualization of MD data, these methods tend to deal mostly with structural details of the system, and the whole area having to do with dynamics of the atoms or molecules is left out. Thus one might see a sequence of images (or even an animation) showing the time development of a system to which external heat or pressure is applied, but one rarely sees such a sequence where the localized forces or energetics are represented. Yet, knowledge of such forces or energetics is often essential to understanding how such systems evolve or why certain experimental probes behave as they do.

Since MD calculations follow the dynamics of every atom (or molecule) in a system, atomic stresses can be calculated for each of these constituents. Atomic stresses are the forces that, when summed over all atoms, give rise to shearing, compressing, and stretching of the system. An appealing method for depicting these forces is to use shape distortions that mimic the expected result of applying stress to the atom. Thus, as shown in Fig. 1, a shear stress in a given direction produces a shear distortion in that direction, an on-axis negative force produces a stretching along that axis, and so on. Using this method one could, in fact, display all six independent atomic stresses (3 on-axis forces and 3 shears) at once.

Unfortunately using shape distortion by itself, it is not easy to distinguish between a shear force alone and a shear force in combination with an on-axis force (see Fig. 1). Because of this and similar problems, we decided to represent the on-axis forces by spots of color centered on the surface at the points where imaginary body axes would emerge from each atom. The viewer can easily make out the direction of each localized patch and can also tell the magnitude of the force in that direction from the color. Now the viewer can see the directions and
Fig. 1 Shape distortions due to atomic forces. Counter-clockwise from lower left around undistorted sphere: shear to left, shear to right, pushing force along vertical, combination of pulling force and shear right, combination of pulling force and shear left, and pulling force along vertical.

Fig. 2 Atomic Force Microscope with nickel tip (above) pulling neck of atoms from gold surface (below). Shape distortions on individual atoms due to atomic forces.
magnitudes of shear and on-axis forces distinctly through the orthogonal representations of shape distortion and color.\(^5\)

We have used our methods to visualize several MD simulations including: an atomic force microscope (AFM) system composed of a nickel tip approaching and then drawing back from a gold surface, a very thin polymer droplet squeezed between metallic walls, and an AFM system with a film of polymer molecules between the tip and substrate. We present here results from two of these studies. Both of these simulations are large (involving several thousand atoms) and must be run on a supercomputer (in this case a Cray Y-MP). In addition both simulations generate thousands of files each containing atomic positions at a given time and associated velocities, atomic stresses, energies, and other properties. These simulations are good candidates for our multivariate analysis because they are complex, generate massive amounts of data, and involve highly localized interactions that promote important dynamical processes. They also offer a range of visual structures since the atomic force images provide a view of crystal arrangements and disordered atomic structures while the polymer images show the conformations and interactions of chain molecules. Thus we were able to test the effectiveness of our multivariate representations in two distinct situations.\(^5\)

Fig. 2 is a snapshot of the AFM system after the weakly bound gold surface atoms have jumped up to coat the nickel tip and form a neck connecting the tip to the surface. One can clearly see the shape distortions caused by shear stresses in the nickel tip and in the gold atoms in the neck. By leafing through the sequence of these images (showing the tip moving toward and away from the surface), one sees that the tip distortions, most prominent at the edges, are mostly frozen in. They indicate that when the hard nickel tip was formed, a large amount of unrelieved stress was locked in. It is clear from the sequence of images, however, that additional shearing forces along crystal axes in the tip occur as the neck forms.

In contrast to the tip, the soft gold substrate quickly relaxes to relieve shearing forces. The only strong shearing forces remaining are in the neck and in isolated gold atoms on the surface. Fig. 2 cannot show the color variations due to on-axis forces, but these indicate a strong stretching of the top substrate layer and compression of the second layer. Some of these forces are due to the presence of the interface and would remain even if the tip were removed. However, the presence of the tip causes a strong additional pulling on the top substrate layer that is also shown in the color variations. This example shows how the depictions of build-up and relief of localized forces as the system is brought to new separations gives insight into the complex dynamics that occur.

Fig. 3 presents a snapshot of the same AFM system except that approximately 2 layers of hexadecane molecules have been placed on the gold substrate. Hexadecane is a chain molecule composed of CH\(_2\) molecules with CH\(_3\) molecules on each end. In our case we use pseudoatoms to represent the CH\(_2\) or CH\(_3\) molecules and do not distinguish between them. Through the use of the appropriate interparticle and dihedral potentials, it has been shown that this model for hexadecane gives reasonable results for the bulk structure.\(^6\) We have also shown that the model gives results consistent with experiment and theory for the behavior of thin interfacial films (2 or 3 layers) squeezed between crystalline surfaces.\(^3\)

The hexadecane molecules act as a lubricant in the AFM system and profoundly affect the nature of the tip-substrate interactions on the atomic scale. When, as in the previous simulations, the nickel tip was brought down toward the substrate, several chain molecules migrated to the tip, coating its underside and even climbing its sides.\(^7\) However, the lubricant film prevented the gold substrate atoms from jumping up to the tip as in the previous simulations. In fact, as the tip was lowered further, a few hexadecane molecules became briefly trapped between the tip and the gold substrate.
Fig. 3  Atomic Force Microscope with polymer film between nickel tip and gold substrate. Shape distortions due to atomic forces applied to all atoms and molecules in the system.

Data Consumer

Fig. 4 The Data Consumer
The depiction in Fig. 3 must convey connectivity information for the chain molecules in addition to force and structural information. We found that the best way to show connectivity was to connect the molecular segments by small rods. The rod diameters were chosen so that they did not obscure the spots of color representing on-axis forces. As can be seen in Fig. 3, this method of representation successfully shows connectivity of individual molecules while also showing distortions due to atomic stresses. The Figure shows the mechanism by which molecules reach out, attach to the tip, and sometimes even align themselves in the furrows between layers (see the right-hand side of the tip for one molecule reaching out and another aligning in a furrow). Further, a close examination of the force distortions on the molecules underneath the tip shows the build-up and then diminishing of stretching and shearing forces as molecules coat the underside of the tip.

One last image representation tool that significantly increases the information that any of these images convey is stereoscopy. With the first AFM system, for example, the structure and depth of the substrate hole created when the neck of gold atoms formed was much clearer in stereo (see Fig. 2). Stereo views of systems without regular arrangement, such as the polymer systems, show the structure more readily because the viewer can more easily make out which molecules are outside and which are inside for a particular view. Finally, stereoscopy becomes more effective as the amount of detail in the image increases. In our examples, it brings out the distortions of individual atomic spheres more clearly and, in particular, sharply distinguishes the color spots on the curved sphere surface.

Examples of Visualization in Medicine

We will discuss here the work of the Medical Informatics Laboratory at Georgia Tech. The Medical Informatics Laboratory is part of the Graphics, Visualization, and Usability Center and is concerned with exploring theoretical and practical issues dealing with cognitive, reasoning, information processing, and information management tasks in medicine, health care, and biomedical research. Activities involving visualization include visualization of biomedical structures and functions, computer graphics and display of multiparametric data, and medical image processing, analysis, and understanding.

Most of the medical visualization and image analysis work, especially if it is to be used as a diagnostic aid, must be done under certain strictures. In particular, there is resistance among many medical doctors to visualization formats that differ from the formats they were trained to analyze. There is also a fundamental need not to introduce artifacts into images of biomedical data and, in fact, to leave obscuration and fuzziness if they exist in the original sample. Thus aggressive image filtering and enhancement techniques might not be appropriate for this data. The goal here is not to develop general relationships based on an overview of the data; it is to make accurate interpretations and diagnoses based on detail.

In spite of these strictures a major task facing medical visualization is the same as that facing scientific visualization—the depiction for analysis of complex, multidimensional data. In addition, medical visualization deals extensively with integrating multimodality data—information gathered from a variety of observations and observational methods. This is an area that will grow in prominence in scientific visualization as systems such as observational satellites and synchrotron radiation facilities come online.

In the Medical Informatics Lab the different modalities analyzed include x-ray imaging, MRI (magnetic resonance imaging), computer-aided tomography, SPECT (single photon emission tomography), and others. The visualization techniques used include various 3D surface and volume rendering methods. The samples from these non-invasive, in-vivo probes are collected, organized, and processed by computer, and the processed data is then displayed and manipulated on the graphics computer.

An illustration of this process is 3D reconstruction by back projection of MRI
A Paradigm for Interactive Visualization

Most of the above examples were created by researchers who new a great deal about graphics programming or who worked with a graphics programmer. If scientific visualization is to take the necessary step from producing striking images for show-and-tell to being one of the essential analysis tools that scientists use, the researcher must be able to produce these visualizations herself using visual representations of her own choosing. She will also require interactivity—devices and controls that allow full manipulation of data and hardware that responds quickly enough to not impede the flow of analysis. Ultimately, based on the integration of stereoscopy and these tools, scientific visualization will give way to "scientific realization" where the researcher climbs inside the data through a virtual environment and uses interactive devices to navigate and control the realization.

What will the researcher gain in this visualization environment? She will gain the ability and freedom to explore. More often than not researchers do not know fully what to expect from simulations and must change and refine their analyses in reaction to the data. More often than not the visualization that best reveals the data is not a standard one, although it may be a variation on a well-known theme. (Our examples show this point clearly.) Customization and interactivity allow the researcher to iterate her analysis to explore fully an aspect of the data or change her focus to reveal other aspects. Finally, in a fully integrated simulation and analysis environment, the customized representation becomes effective feedback that the researcher acts upon to guide the simulation.

Progress is already being made toward this "user-centered" interactive environment. Systems such as AVS,9 apE,10 the SGI Iris Explorer (soon to be released), and Wavefront's Data Visualizer, among many others, provide environments for data-driven, interactive visualization. The first three of these provide highly flexible and powerful "object-oriented" environments that can be distributed across a net-
work of machines and that are composed of modules that the user can link to construct personalized analysis and visualization applications. The key to all these systems for data visualization is, as Foley points out, a clean separation between the data producer (simulation or data collection system) and the data consumer (manipulation/visualization system). This separation allows the formulation of general visualization and analysis schemes that are not embedded in the data production phase and thus need not be known in detail (e.g., at the level of graphics calls) by the user.

The Interactive Glyphmaker

In Fig. 4 we present our view of the data consumer. All parts are controlled through an interactive interface manager, and to the left their are incoming (data) and outgoing (control) paths through the user interface that indicate the ability to steer the data producer based on visual feedback. In addition the user can direct the data stream through other processes (e.g., AVS modules). An important new feature in our approach is the interactive glyphmaker which permits the user to bind the attributes of visual objects to individual streams of data. To allow maximum flexibility in the depiction of the bound data, we show a preprocessor stage in Fig. 4 through which the user, employing her own filtering routines or using supplied routines, can convert data to whatever form she needs. This feature is important since data, especially data taken from more than one source (e.g., two different simulations or from observations and simulations), must almost always be converted to allow comparison. In the future this feature will grow in importance as simulational data must be transformed for comparison with data previously deposited in vast archives.

We envision the glyphmaker as having both a modeler for creating glyphs and predefined glyphs. Glyphs from either source could be grouped into compound icons. In our above AFM example, the glyph is a sphere and atomic force data is bound to distortion points on the sphere’s surface. In addition the overall color of the sphere is bound by classification (i.e., the color is red for nickel atoms and gold for gold atoms). The color spectrum for each of the on-axis color spots is bound to the appropriate on-axis forces. Finally, the glyph is instanced thousands of times for each atom in the system and the position for each instance is bound to the coordinates of corresponding atoms. For the case of the chain molecules we have composite icons composed of sphere and cylinder glyphs. Thus we see that even in the case of simple, 3D glyphs the binding process can be fairly complicated.

Another point revealed by this example is the need for creating, in some cases, many thousands of instances of glyphs. The code generated in response to the instancing operation must be efficient to quickly generate images containing thousands of objects. The most likely choice will be a do-loop structure with the loop index running over the number of instances and within the loop calls to subroutines creating the glyphs. Data flow will have to be carefully considered; perhaps the module containing bindings to the glyphs will collect data for all instances into arrays before firing and creating the scene. Of course, fast creation of each instance will be quite important, too. Options for creating crude objects or for only constructing the glyph surfaces that face the viewer, which we have used successfully in other visualization software, could be used here.

The set of pre-formed glyphs would be simple and the active binding points on each glyph limited. A simple set might include a point, a line, a cube, a cylinder, and a sphere. It might also be straightforward to bind to more general shapes like surfaces or voxels. Of course other objects could be formed from combinations or deformations of these basic glyphs. For example, a set of lines could form a vector, a cube could be deformed into a rectangular solid or a wedge, a cylinder could form a cone by collapsing the radius of one of its (circular) surfaces, and so on. One can imagine how many of the standard visual representations could be constructed from combinations of these glyphs. In addition, of course, with this approach there is the possibility of combining methods of representation or of creating entirely new representations.
Let us re-emphasize the importance of tailoring visual representations to the data. As has been pointed out by Becker and Cleveland, the relations in multivariate data, such as scattered data taken from series of observations, can often be obscured rather than enhanced by standard visualization techniques. They show that these data are "non-scenic" and often have pairs of variables that may have no apparent functional relationship with each other. In this case the usual techniques involving, say, the fitting of 3D surfaces to a set of variables is not as useful as merely arranging a matrix of scatter plots. Our glyph-making technique could easily be applied to this type of data. Of course that is not the whole story; Becker's carefully thought-out arrangements of scatter plots bring out the multivariate relationships in the data.

One last point should be made about the approach of Becker and Cleveland. They show how useful it is to provide visual links between different views of multivariate data. One dynamical, interactive method they use to achieve this is called brushing. With this method the mouse, or other interactive tool, is used to highlight data in a certain region of a view. At the same time the variables represented by this data are also highlighted in other views. These other views may represent different sets of relationships in the multivariate data and thus the brushing highlights connections and trends among variables. An obvious extension for our environment would be that variables, objects, or other elements so highlighted could then be followed in subsequent animations of the data. This tool would be very useful in following specific elements in dense, complex datasets. Also, as Becker and Cleveland point out, another use of brushing is conditioning whereby the brush is used to highlight one or more variables in a given range of values and thus bring out the relationship with other variables under that condition. The brush is a straightforward extension of our binding approach, and we plan to implement it in our application.

The method for binding data to active elements on the glyphs will employ binding tables similar to those described by Foley and McMath in their implementation of dynamic process visualization. The approach will also use a design similar to that of the Connections Editor, which is part of the newly developed Iris Explorer. The binding tables will contain lists of elements for the glyphs (defined at the glyph-making stage) and lists of user-defined data variables. Both the glyph and data lists will be organized into hierarchical structures to show various groupings and connections. Such items as the number of instances for a certain glyph, type of highlighting for the brush, etc., will also be bound at this stage. We will describe the organization and operation of binding more fully in a future publication.

We are now working to develop the interactive glyphmaker and binding modules based on the design considerations discussed here. Our first implementation is in terms of the set of simple, pre-defined glyphs outlined above and their groupings. In a future version, we will consider the development of a general glyph modeler. We also plan implementations in terms of object-oriented modules that would accept data streams from any source (including remote machines or databases) and standalone applications that would read data from files. We will report on these developments in a future publication.

Acknowledgement: Some of the calculations presented in this paper were performed at the Pittsburgh Supercomputing Center via a grant of computer time.

References
4. W. D. Luedtke, Uzi Landman, and M. W. Ribarsky, "Structural and Dynamical Consequences of Interactions in..."


