

Scientific Data Visualization

3.05. 2018

Scientific Visualization

- Scientific visualization, sometimes referred to in shorthand as SciVis, is the representation of data graphically as a means of gaining understanding and insight into the data. It is sometimes referred to as visual data analysis. This allows the researcher to gain insight into the system that is studied in ways previously impossible. *What it is not*- It is important to differentiate between scientific visualization and presentation graphics. Presentation graphics is primarily concerned with the communication of information and results in ways that are easily understood. In scientific visualization, we seek to understand the data. However, often the two methods are intertwined.

... *Scientific Visualization*

- From a computing perspective, SciVis is part of a greater field called visualization. This involves research in computer graphics, image processing, high performance computing, and other areas. The same tools that are used for SciVis may be applied to animation, or multimedia presentation, for example.
- As a science, scientific visualization is the study concerned with the interactive display and analysis of data. Often one would like the ability to do real-time visualization of data from any source. Thus our purview is information, scientific, or engineering visualization and closely related problems such as computational steering or multivariate analysis. The approaches developed are general, and the goal is to make them applicable to datasets of any size whatever while still retaining high interactivity. As an emerging science, its strategy is to develop fundamental ideas leading to general tools for real applications. This pursuit is multidisciplinary in that it uses the same techniques across many areas of study.

... *Scientific Visualization*

- Through the availability of increasingly powerful computers with increasing amounts of internal and external memory, it is possible to investigate incredibly complex dynamics by means of ever more realistic simulations. However, this brings with it vast amounts of data . To analyze these data it is imperative to have software tools which can visualize these multi-dimensional data sets. Comparing this with experiment and theory it becomes clear that visualization of scientific data is useful yet difficult. For complicated, time-dependent simulations, the running of the simulation may involve the calculation of many time steps, which requires a substantial amount of CPU time , and memory resources are still limited, one cannot save the results of every time step. Hence, it will be necessary to visualize and store the results selectively in `real time' so that we do not have to recompute the dynamics if we want to see the same scene again. `Real time' means that the selected time step will be visualized as soon as it has been calculated.

... Scientific Visualization

- The main reasons for scientific visualization are: it will compress a lot of data into one picture (data browsing), it can reveal correlations between different quantities both in space and time, it can furnish new space-like structures beside the ones which are already known from previous calculations, and it opens up the possibility to view the data selectively and interactively in *real time*. By following the formation and the deformation as well as the motions of these structures in time, one will gain insight into the complicated dynamics. As was mentioned before, we also want to integrate our simulation codes into a visualization environment in order to analyze the data 'real time' and to by-pass the need to store every intermediate result for later analysis. This is possible by means of processing in which the simulation is distributed over a set of high-performance computers and the actual visualization is done on a graphical distributive workstation. It is also very useful to have the possibility to interactively change the simulation parameters and immediately see the effect of this change through the new data. This process is called computational steering and it will increase the effective use of CPU time.

- The discussion is focussed on the following questions:
 - What is the improvement in the understanding of the data as compared to the situation without visualization?
 - Which visualization techniques are suitable for one's data? Are direct volume rendering techniques to be preferred over surface rendering techniques?
 - Can current techniques, like streamline and particle advection methods, be used to appropriately outline the known visual phenomena in the system?

- The success of visualization not only depends on the results which it produces, but also depends on the environment in which it has to be done. This environment is determined by the available hardware, like graphical workstations, disk space, color printers, video editing hardware, and network bandwidth, and by the visualization software. For example, the graphical hardware imposes constraints on interactive speed of visualization and on the size of the data sets which can be handled. Many different problems encountered with visualization software must be taken into account. The user interface, programming model, data input, data output, data manipulation facilities, and other related items are all important. The way in which these items are implemented determines the convenience and effectiveness of the use of the software package as seen by the scientist. Furthermore, whether software supports distributive processing and computational steering must be taken into account.

Examples

- Engineering
- Computational Fluid Dynamics
- Finite Element Analysis
- Electronic Design Automation
- Simulation
- Medical Imaging
- Geospatial
- RF Propagation
- Meteorology
- Hydrology
- Data Fusion
- Ground Water Modeling
- Oil and Gas Exploration and Production
- Finance
- Data Mining/OLAP

Visualization techniques

Classification

- is often based on the dimension of the domain of the quantity that is visualized, i.e. the number of independent variables of the domain on which the quantity acts, and on the type of the quantity, i.e. scalar, vector, or tensor .
- In MHD, two scalar quantities occur, viz. temperature (or pressure) and density, and two vector quantities viz. magnetic field and velocity field. These quantities are defined on a four-dimensional domain which is spanned up by the space and time coordinates. The time dependence is treated different than other dependencies. In particular, animation is used to visualize this dependency (see below).
- Visualization techniques can also be divided into surface rendering techniques, and (direct) volume rendering techniques . Surface rendering is an indirect geometry based technique which is used to visualize structures in 3D scalar or vector fields by converting these structures into surface representations first and then using conventional computer graphics techniques to render these surfaces. Direct volume rendering is a technique for the visualization of 3D scalar data sets without a conversion to surface representations.

Visualization techniques and methods

We will apply on MHD. The focus is on 3D scalar and vector techniques, because often data consists of 3D scalar and vector fields. Since color is involved in most techniques we start with a description of three frequently used color coding systems.

- Color coding
- Classification of techniques
- Surface rendering techniques
- Volume rendering techniques
- Animation techniques

1. *Isosurfaces*

The first visualization technique discussed is the generation of isosurfaces. Since the attributes consist of scalar data defined on a three dimensional grid, isosurfaces served as a natural way to extract surface geometry for this data set. By appropriate thresholding the colormap so that positive potential is represented by red, and negative potential by blue, a visualization was generated that indicated both the shape of the molecular orbits as well as their potential. An example using this isosurface visualization technique is provided below.

Another isosurface technique used was to render semi-transparent isosurfaces. This allowed the user to observe orbitals that were enveloped by other orbitals. An example of this is shown below.

2. *Volumetric rendering*

As mentioned in class, volumetric rendering allows the entire data set to be viewed at once, and lets the user to "see inside" the data. For each pixel in an image created using volumetric rendering, a ray is cast through the semi-transparent volume. The resulting color at the pixel is a composite of all the voxels the ray has intersected. As a consequence, such images tend to be blurry. Another characteristic volumetric rendering is that it is typically slower than surface rendering techniques.

Volumetric renderings of this data set took over ten times as long to generate; therefore, volumetric rendering of this data set was not well suited for realtime visualization. However, it does provide features that are obscured by surface rendering techniques. An example using volumetric rendering is provided below.

3. *Slicing*

Slicing was another technique applied to this data set. Slicing through the grid with a plane provides the user with detailed information of scalar values within the grid volume. To implement the slicing, I first generated a slice plane that was positioned at the bottom of the grid volume and parallel to the xy axis. Then, I moved the slice plane up the z axis incrementally until it reached the top of the grid. Such a slice appears in the image below. By taking slices, an animation was produced which shows how the value of the electron potential through the volume.

4. *Contours*

Taking 2D contours through this data set was another visualization technique explored in this project. These were produced by slicing the data using a plane with a normal oriented up the x-axis, and then applying isosurfaces on the 2D domain. These contours were found to offer detailed information about the shape of the atomic orbitals, and were computed in a computationally efficient manner. An example of this 2D contour technique is provided below.

5. *Animation*

Since the simulation consists of ten time steps of the system as the oxygen atom approaches the carbon atom, the data set naturally lends itself to an animation. The isosurface and volumetric rendering animations demonstrate the motion and formation of molecular bonds. The slicing animation offers a closer inspection of each frame. Animations are shown at the theatre.

Color coding

Colors and light are essential to visualization. Most visualization techniques contain a step in which data values are mapped to colors to make the range of the data visible (see the next sections). The interpretation of results produced by these visualization techniques depends crucially on the mapping of data to colors because the human eye is more sensitive to some parts of the visible spectrum of light than to other parts and the brain may interpret different color patterns differently. Since the mapping of data values to colors involves color coding, we will describe this in some detail. There exist quite a few color coding systems. We discuss here the RGB, CMY, and HSV systems.

... Color coding

The *RGB* code system is mainly used in television sets and computer screens. Color is determined by the three independent components Red (R), Green (G), and Blue (B). These components correspond with the red, green, and blue cathode ray tubes. Each component has values in the interval $[0,1]$. A value of 0 corresponds to black and a value of 1 corresponds with full color. For example $\{R=0, G=0, B=0\}$ is black, $\{R=1, G=0, B=0\}$ is full red, $\{R=0, G=1, B=1\}$ is full cyan, and $\{R=1, G=1, B=1\}$ is white. Every color can be constructed by taking the right combination of the three components. However, the number of values that can be represented by cathode ray tubes for each component in the interval $[0, 1]$ is finite. Most graphical display systems currently use 256 different values, which means that $256^{*}3 = 16,777,216$ different colors are supported: more than enough for most applications.

... *Color coding*

Another color coding system is the **CMY** system. Color is determined by the three independent components Cyan ($C=1-R$), Magenta ($M=1-G$), and Yellow ($Y=1-B$). It is widely used in the publishing world. The CMY system is related to the RGB system as follows: $C=1-R$, $M=1-G$, and $Y=1-B$. It is clear that these systems are complementary to each other.

... Color coding

The color coding which is used most frequently in visual environments is the *HSV system*. The three components Hue (H), Saturation (S), and Value (V) have values between 0 and 1. Whereas one can view the volume spanned up by the three independent components of the RGB and CMY color systems as a unit cube, the HSV volume is an inverted cone. The Hue is the angular coordinate in a plane perpendicular to the symmetry axis, where $H=0$ corresponds to red and an angle of 0 degrees, $H=0.33$ corresponds to green and an angle of 120 degrees, $H=0.66$ corresponds to blue and an angle of 240 degrees, and $H=1.0$ corresponds again to red. The Saturation determines the saturation of the color. Saturation is the radial coordinate in the plane perpendicular to the symmetry axis, so that $S=1$ corresponds to full saturation and $S=0$ corresponds to no color (i.e. white), $S=0$ is situated on the symmetry axis and $S=1$ corresponds to the maximum radius in the plane, i.e. the boundary of the cone. The Value determines the intensity of the color. Value is the coordinate along the symmetry axis of the cone, where $V=0$ corresponds to no intensity (i.e. black), and $V=1$ corresponds to maximum intensity. $V=0$ is at the top of the cone and $V=1$ is at the base of the cone. We will not discuss the relation between the RGB and HSV system as it is a complicated one. Interested people are referred to refs. [2] and [3]. HSV is frequently used in visual environments, because the effects of manipulating the components are much more predictable than in RGB and CMY. HSV corresponds to a more natural experience of colors than the other systems provide.

... *Color coding*

In most visualization techniques, colors from red to blue are used to reveal transitions in some quantity. If one wants to use the complete set of available colors one needs to map the range of values to this complete set. Storing this mapping consumes too much memory.

Therefore, one normally chooses N colors of the available set (typically $N=256$) and maps the range of values to these N colors. This mapping of the values to colors will be denoted as the colormap in the rest of this report.

It is important to notice that using colors to reveal transitions in quantities can be misleading if there are no transitions at all, e.g. for a monotonically increasing quantity. For the kind of phenomena in which the quantity changes smoothly it is better to use continuously changing schemes like gray scales, saturation scales, and intensity scales. This immediately shows another advantage of the HSV code system: saturation and intensity are natural variables in the HSV system.

Surface rendering techniques

A general set of 3D scalar and vector surface rendering techniques.

The first four descriptions deal with scalar field techniques and the other two with vector field techniques.

Scalar glyphs

Scalar glyphs is a technique which puts a sphere or a diamond on every data point. The scale of the sphere or diamond is determined by the data value. The scalar glyphs may be colored according to the same scalar field or according to another scalar field. In this way correlations can be found. As no interpolations are needed for this technique it consumes few CPU seconds.

... Surface rendering techniques

Isosurfaces

This technique produces surfaces in the domain of the scalar quantity on which the scalar quantity has the same value, the so-called isosurface value. The surfaces can be colored according to the isosurface value or they can be colored according to another scalar field using the texture technique. The latter case allows for the search for correlation between different scalar quantities.

There are different methods to generate the surfaces from a discrete set of data points. All methods use interpolation to construct a continuous function. The correctness of the generated surfaces depends on how well the constructed continuous function matches the underlying continuous function representing the discrete data set. The method which is implemented in the software packages described in chapter 3, is the Marching Cube Algorithm.

... Surface rendering techniques

Cutting planes

This technique makes it possible to view scalar data on a cross-section of the data volume with a cutting plane. One defines a regular, Cartesian grid on the plane and the data values on this grid are found by interpolation of the original data. A convenient colormap is used to make the data visible.

Orthogonal slicers

It often occurs that one wants to focus on the influence of only two independent variables (i.e. coordinates). Thus, the other independent variables are kept constant. This is what the orthogonal slicer method does. For example, if the data is defined in spherical coordinates and one wants to focus on the angular dependences for a specific radius, the orthogonal slicer method constructs the corresponding sphere. No interpolation is used since the original grid with the corresponding data is inherited. A convenient colormap is used to make the data visible.

... Surface rendering techniques

Vector glyphs

This technique uses needle or arrow glyphs to represent vectors at each data point. The direction of the glyph corresponds to the direction of the vector and its magnitude corresponds to the magnitude of the vector. The glyphs can be colored according to a scalar field..

Textures

This is a technique to color arbitrary surfaces, e.g. those generated by the isosurface techniques, according to a 3D scalar field. An interpolation scheme is used to determine the values of the scalar field on the surface. A colormap is used to assign the color.

Streamlines, streaklines, and particle advection

This is a set of methods for outlining the topology, i.e. the field lines, of a vector field. Generally, one takes a set of starting points, finds the vectors at these points by interpolation, if necessary, and integrates the points along the direction of the vector. At the new positions the vector values are found by interpolation and one integrates again. This process stops if a predetermined number of integration steps has been reached or if the points end up outside the data volume. The calculated points are connected by lines.

The streamlines technique considers the vector field to be static whereas the streaklines technique considers the vector field to be time dependent. The streakline technique interpolates not only in the spatial direction, but also in the time direction. The particle advection method places little spheres at the starting points representing massless particles. The particles are also integrated along the field lines. After every integration step each particle is drawn together with a line or ribbon tail indicating the direction in which the particle is moving.

Volume Visualization

Volume rendering is used to view 3d data without the usual intermediate step of deriving a geometric representation which is then rendered. The volume representation uses voxels, or volume elements to determine visual properties, such as opacity, color, shading at each point in the computational domain. Several images are created by slicing the volume perpendicular to the viewing axis at a regular interval and compositing together the contributing images from back to front, thus summing voxel opacities and colors at each pixel. By rapidly changing the color and opacity transfer functions, various structures are interactively revealed in the spatial domain.

... Volume Visualization

Volumetric rendering

Volumetric rendering allows the entire data set to be viewed at once, and lets the user "see inside" the data. For each pixel in an image created using volumetric rendering, a ray is cast through the semi-transparent volume. The resulting color at the pixel is a composite of all the voxels the ray has intersected. As a consequence, such images tend to be blurry. Another characteristic volumetric rendering is that it is typically slower than surface rendering techniques. Therefore, volumetric rendering of a data set is often not well suited for realtime visualization. However, it does provide features that are obscured by surface rendering techniques.

Volume rendering techniques

Volume rendering techniques have been developed to overcome problems of the accurate representation of surfaces in the isosurface techniques. In short, these problems are related to making a decision for every volume element whether or not the surface passes through it and this can produce false positives (spurious surfaces) or false negatives (erroneous holes in surfaces), particularly in the presence of small or poorly defined features. Volume rendering does not use intermediate geometrical representations, in contrast to surface rendering techniques. It offers the possibility for displaying weak or fuzzy surfaces. This frees one from the requirement to make a decision whether a surface is present or not.

... Volume rendering techniques

Volumetric rendering

Volume rendering involves the following steps:

1. the forming of an RGBA volume from the data,
2. reconstruction of a continuous function from this discrete data set, and
3. projecting it onto the 2D viewing plane (the output image) from the desired point of view.

An RGBA volume is a 3D four-vector data set, where the first three components are the familiar R, G, and B color components and the last component, A, represents opacity. An opacity value of 0 means totally transparent and a value of 1 means totally opaque. Behind the RGBA volume an opaque background is placed. The mapping of the data to opacity values acts as a classification of the data one is interested in.

... *Volume rendering techniques*

... *Volumetric rendering*

Isosurfaces can be shown by mapping the corresponding data values to almost opaque values and the rest to transparent values. The appearance of surfaces can be improved by using shading techniques to form the RGB mapping. However, opacity can be used to see the interior of the data volume too. These interiors appear as clouds with varying density and color. A big advantage of volume rendering is that this interior information is not thrown away, so that it enables one to look at the 3D data set as a whole. Disadvantages are the difficult interpretation of the cloudy interiors and the long time, compared to surface rendering, needed to perform volume rendering.

... *Volume rendering techniques*

We will describe two implementations of volume rendering:

1. *ray casting* and
2. *splatting*.

These implementations are used in the four visualization packages we have compared. The two methods differ in the way the RGBA volume is projected onto the 2D viewing plane.

... Volume rendering techniques

Ray casting

Several implementations exist for ray casting. We describe the implementation used in Visualization Data Explorer. For every pixel in the output image a ray is shot into the data volume. At a predetermined number of evenly spaced locations along the ray the color and opacity values are obtained by interpolation. The interpolated colors and opacities are merged with each other and with the background by compositing in back-to-front order to yield the color of the pixel. These compositing calculations are simply linear transformations. Specifically, the color of the ray C_{out} as it leaves each sample location, is related to the color C_{in} of the ray, as it enters, and to the color $c(x_j)$ and the opacity $\alpha(x)$ at that sample location by the transparency formula :

$$C_{out} = C_{in} (1 - \alpha(x_j)) + c(x_j) * \alpha(x_j)$$

... *Volume rendering techniques*

... *Ray casting*

Performing this formula in a back-to-front order, i.e. starting at the background and moving towards the image plane, will produce the pixel color. It is clear from the above formula that the opacity acts as a data selector.

For example, sample points with opacity values close to 1 hide almost all the information along the ray between the background and the sample point and opacity values close to zero transfer the information almost unaltered.

This way of compositing is equal to the dense-emitter model, where the color indicates the instantaneous emission rate and the opacity indicates the instantaneous absorption rate.

... *Volume rendering techniques*

Splatting

This technique was developed to improve the speed of calculation of volume rendering techniques like ray casting, at the price of less accurate rendering. We will not go into detail here as this technique is rather complicated. It differs from ray casting in the projection method. Splatting projects voxels, i.e. volume elements, on the 2D viewing plane. It approximates this projection by a so-called Gaussian splat, which depends on the opacity and on the color of the voxel (other splat types, like linear splats can be used also). A projection is made for every voxel and the resulting splats are composited on top of each other in back-to-front order to produce the final image.

Data Types

- ❑ Hierarchical Data Formats (HDF)
- ❑ Network Common Data Format (netCDF)
- ❑ Databases- The currently accepted storage method for most scientific data is the Relational Database Management System. This is the format used by many commercial databases, such as Oracle. Data can be extracted using the Standard Query Language (SQL) commands.

Animation techniques

These techniques simulate continuous motion by rapidly displaying images. The viewer is given the impression that he is watching a continuous motion. To achieve this impression the graphical hardware needs image display rates of at least 25 images per second, since otherwise motion will look shaky. As most graphical hardware can not reach that display rate for moderate sized images (i.e. 256x256 pixels), one uses video hardware. One either sends every image to a framebuffer to write one videoframe at a time to videotape or one stores the images on a fast accessible device, CMY a laserdisk, and, after all images have been stored, displays them on a television screen from where they can be put on a videotape. There are two kinds of animation which we will describe below.

... Animation techniques

Flipbook animation

This is a well known technique. The generated images are displayed one after the other. Its name is attached to the thumbing or flipping through a series of images.

Keyframe animation

For this technique one only has to generate so-called keyframes. Keyframes mark changes in the characteristics of the motion, for example the sudden change in the direction of motion of an electron due to a collision with an ion. Interpolation techniques are used to generate a set of images between two keyframes. The larger the interpolated set of images the smoother the conversion from one keyframe to the other will appear to the viewer.

References

Research at the Scientific Visualization Lab

<http://www.cc.gatech.edu/scivis/research/research.html>

<http://www.cc.gatech.edu/scivis/tutorial/tutorial.html>

- Facilities
 - [Parallel Rendering](#)
 - Visualization Software
 - [Graphics Libraries and Presentation Package](#)
 - [Turnkey Visualization Packages](#)
 - [Application Builders](#)
 - [Graphics Libraries and Presentation Packages](#)
- Choosing Software
 - [Choosing a visualization application package](#)
 - [Comparison of Visualization Packages](#)
 - [Further Analysis of Visualization Packages](#)
- [In the Future](#)