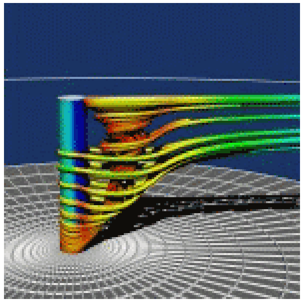
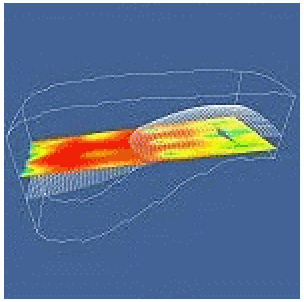
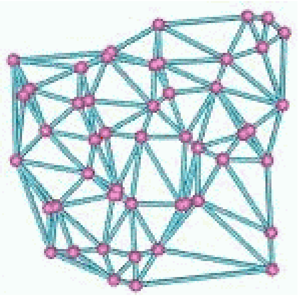
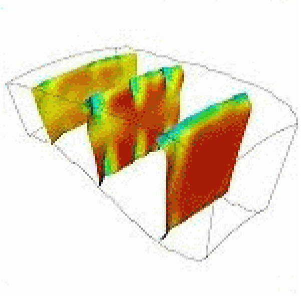
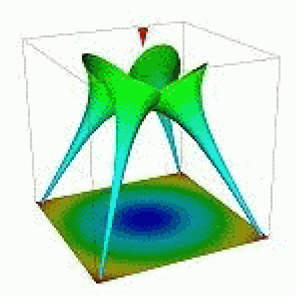
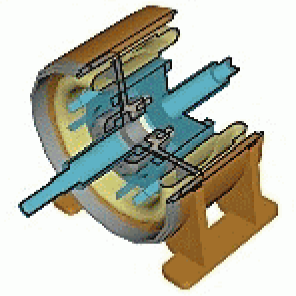
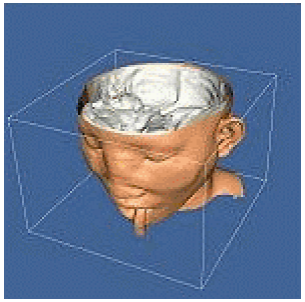
CEG790

Introduction to Scientific Visualization

Outline

- Introduction
- The visualization pipeline
- Basic data representations
- Fundamental algorithms
- Advanced computer graphics
- Advanced data representations
- Advanced algorithms

Literature

- Will Schroeder, Ken Martin, Bill Lorenson, **The Visualization Toolkit - An Object-Oriented Approach To 3D Graphics**, Kitware, 2004
- Charles D. Hansen, Chris Johnson, **Visualization Handbook**, First Edition, Academic Press, 2004
- **Scientific visualization: overviews, methodologies, and techniques**, Gregory M. Nielson, Hans Hagen, Heinrich Müller, Los Alamitos, Calif., IEEE Computer Society Press, 1997
- Woo, Neider, Davis, Shreiner, **OpenGL Programming Guide**, Addison Wesley, 2000, http://www.opengl.org/documentation/red_book_1.0
- http://www-static.cc.gatech.edu/scivis/tutorial/linked/classification.html

What is 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.

What is 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.

Motivation

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 re-compute 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.

Motivation (continued)

The main reasons for scientific visualization are the following ones: 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 in '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.

Common Questions and Concerns

The discussion is focused 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?

Common Questions and Concerns (cont.)

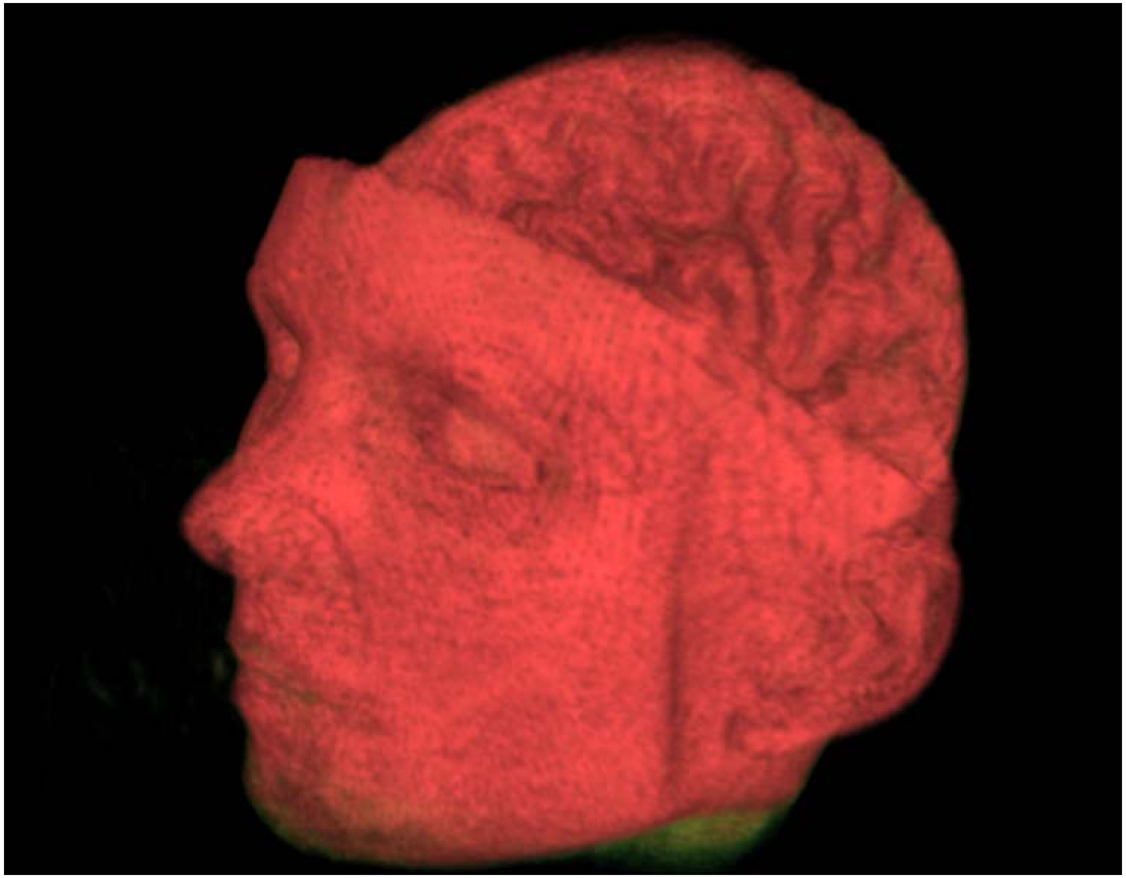
The success of visualization not only depends on the results which it produces, but also depends on the environment in which it has to be done. This environment is determined by the available hardware, like graphical workstations, disk space, color printers, video editing hardware, and network bandwidth, 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.

Application 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

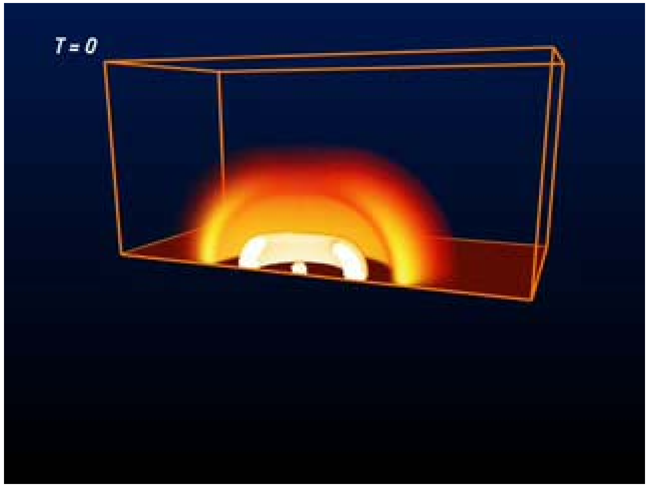
Application examples (continued)

Volume visualization of an MRI scanned brain data set



Application examples (continued)

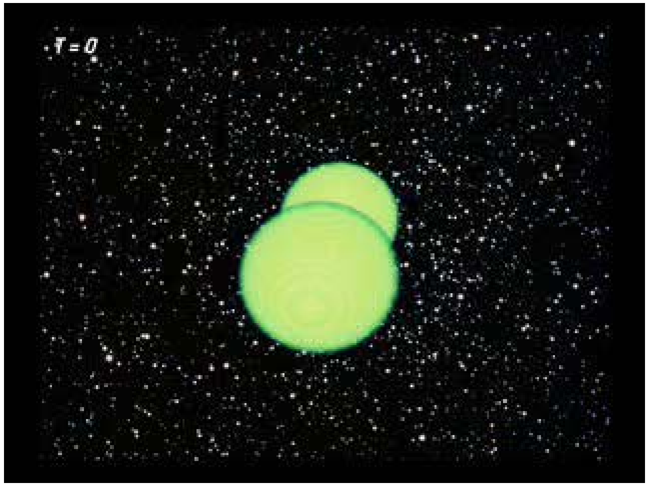
General relativistic simulation of gravitational energy



Visualization by W. Benger, Simulation by AEI Potsdam.

Application examples (continued)

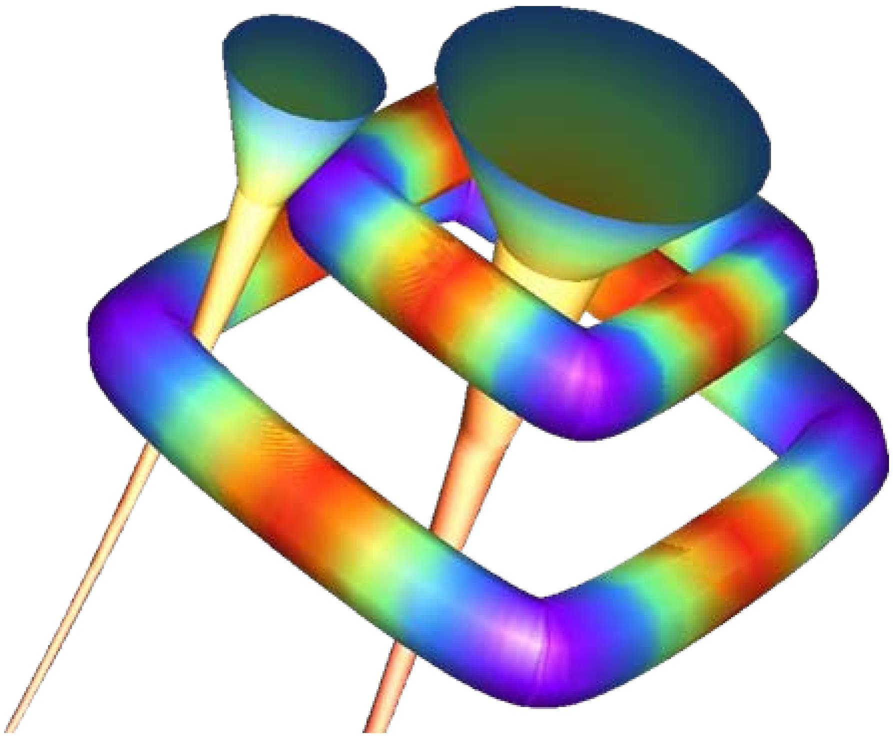
Merger of orbiting binary neutron stars. Visualization of energy density and mass density with volume rendering.



Visualization by W. Benger, Simulation by AEI Potsdam.

Application examples (continued)

Feature detection: closed hyperstreamlines

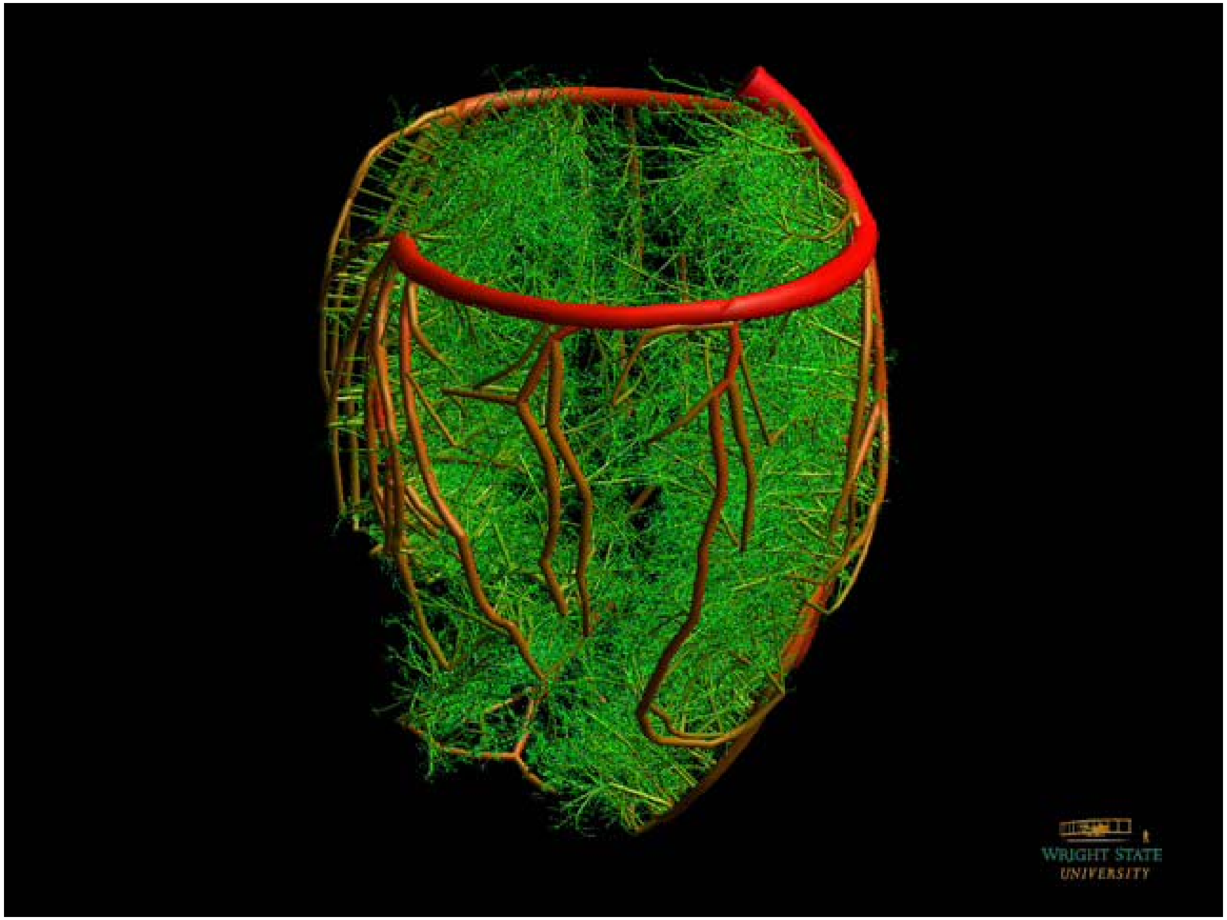


Application examples (continued)

Simulation of the air flow around a wing. Visualized using Amira's illuminated field line module.

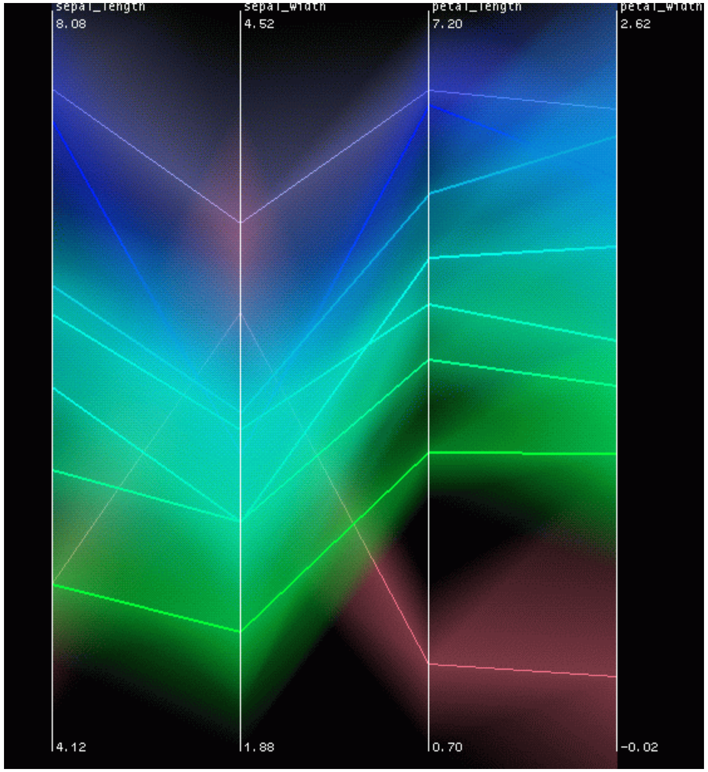
Application examples (continued)

Medical visualization: arterial tree



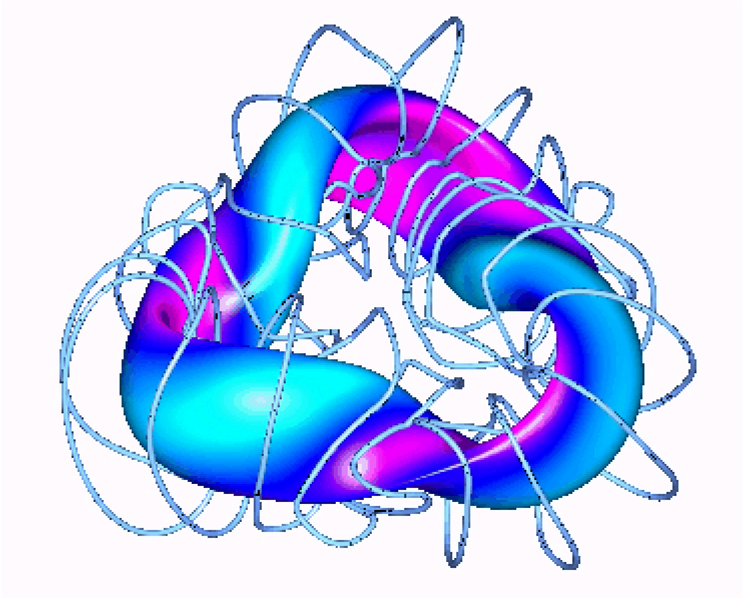
Application examples (continued)

Information visualization: parallel coordinates



Application examples (continued)

Magneto-Hydrodynamics (MHD) models are extensively used in the analysis of magnetic fusion devices, industrial processing plasmas, and ionospheric/astrophysical plasmas. MHD is the extension of fluid dynamics to ionized gases, including the effects of electric and magnetic fields. Time-independent MHD equilibria also form the basis for more advanced kinetic and particle-based models of plasma behavior.



Classification of techniques

Classification of visualization techniques 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, i.e. temperature (or pressure) and density, and two vector quantities, i.e. 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 differently 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

Different techniques are used to visualize different data sets. These techniques can be, for example:

- Surface rendering
- Volume rendering
- Glyph-based visualization
- Feature-based visualization
- Information visualization

Surface rendering

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 common method which is implemented in many software packages is the Marching Cube Algorithm.

Volume rendering

Volume rendering is used to view 3-D 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. For texture-based techniques, 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 rendering (continued)

Volumetric rendering allows the entire data set to be viewed at once, and lets the user "see inside" the data. When using ray casting, 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 real-time visualization. However, it does provide features that are obscured by surface rendering techniques.

Glyph-based visualization

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.

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.

Feature-based visualization

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 difference between streamlines and streaklines is that the streamlines technique considers the vector field to be static whereas the streaklines technique considers the vector field to be time dependent. Hence, 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.

Information visualization

The rise of the Information Age and the ascendancy of Computer Graphics come together in the area of Information Visualization, where interactive graphical interfaces are used for revealing structure, extracting meaning, and navigating large and complex information worlds.

Increasing amounts of data and information and the availability of fast digital network access (e.g., in the information highway environment) have created a demand for querying, accessing, and retrieving information and data. However, information technology will not transform business, science, medicine, engineering, and education if the users cannot use it easily and efficiently. Technology must come to the users, taking their needs into account. If we do not involve the users, we will develop useless systems. One of the concerns of this field is the human-information interface, and how advances in interactive computer graphics hardware, mass storage, and data visualization could be used to visualize information.

Visualization toolkits

Different software or toolkits are available:

- Amira (Zuse Institute Berlin (ZIB))
- Fast (NASA)
- OpenDX (formerly IBM Data Explorer)
- AVS (Advanced Visual Systems)
- Visualization toolkit (vtk) (Kitware)
- OpenGL
- …

Throughout this course, we will focus on vtk; however, other visualization tools or directly using OpenGL can be more suitable sometimes.