Scientific Visualization: An Introduction

Gitta O. Domik

November 28, 1995

High Performance Scientific Computing
University of Colorado at Boulder
Copyright ©1995 by the HPSC Group of the University of Colorado

The following are members of
the HPSC Group of the Department of Computer Science
at the University of Colorado at Boulder:

Lloyd D. Fosdick
Elizabeth R. Jessup
Carolyn J. C. Schauble
Gitta O. Domik
### Contents

1 Definitions and goals of scientific visualization ........................................ 1

2 History of scientific visualization ............................................................. 3

3 Example of scientific visualization ............................................................. 4

4 Concepts of scientific visualization ........................................................... 11
   4.1 Mapping *numbers to pictures* .......................................................... 11
   4.2 Expressiveness .................................................................................. 11
   4.3 Effectiveness ...................................................................................... 12
   4.4 Subjectivity of interpreter .................................................................. 12
   4.5 A picture is the summary of visual cues ............................................ 12

5 Visual cues ................................................................................................. 13
   5.1 Innate reactions to visual cues ......................................................... 14
   5.2 Acquired reactions to visual cues ...................................................... 14
   5.3 Illusory visual cues .......................................................................... 14

6 Characterization of scientific data .............................................................. 15
   6.1 Points ............................................................................................... 16
   6.2 Scalars ............................................................................................. 17
   6.3 Vectors ............................................................................................. 18
   6.4 Tensor fields ..................................................................................... 19

7 Visualization techniques ........................................................................... 19
   7.1 Scatter plots ...................................................................................... 19
   7.2 Glyphs .............................................................................................. 21
   7.3 Line graphs ....................................................................................... 23
   7.4 Histograms, pie charts ..................................................................... 23
   7.5 Contour plot (isolines) ...................................................................... 25
   7.6 Image display .................................................................................... 25
   7.7 Surface view ..................................................................................... 26
   7.8 Color transformations ....................................................................... 26
   7.9 Isosurfaces ....................................................................................... 29
   7.10 Ray tracing of volumes ..................................................................... 29
   7.11 Data slicers ...................................................................................... 34
Scientific Visualization

7.12 Arrows ................................................................. 34
7.13 Streamlines and particle tracks .............................. 35
7.14 Animation ............................................................. 35

8 Annotations .............................................................. 36

9 Interactivity .............................................................. 36

10 Interpretation goals to pursue with visualization ........ 37

11 Quantitative versus qualitative data interpretation ....... 37

References ............................................................... 39
Trademark Notice

- AVS is a trademark of Advanced Visual Systems, Inc.
- IDL is a registered trademark of Research Systems, Inc.
- MATLAB is a trademark of The MathWorks, Inc.
- X-Window System is a trademark of The Massachusetts Institute of Technology.
Scientific Visualization: 
An Introduction*

Gitta O. Domik†

November 28, 1995

1 Definitions and goals of scientific visualization

Visualization of Scientific Data describes the application of graphical methods to enhance interpretation and meaning of scientific data. Visualization of Scientific Data is abbreviated to Scientific Visualization or Visualization throughout this tutorial. Scientific data can be derived from various sources, including measuring instruments, or may be obtained as a result of scientific computations performed on supercomputers. However, data do not become useful until some (or all) of the information they carry is extracted. The goal of scientific visualization is to provide concepts, methods and tools to create expressive and effective visual representations from scientific data. Such visual representations convey new insights and an improved understanding of physical processes, mathematical concepts and other quantifiable phenomena expressed in the data [Pang 93]. Together with quantitative analysis of data,

*This work has been supported by the National Science Foundation under an Educational Infrastructure grant, CDA-9017953. It has been produced by the HPSC Group, Department of Computer Science, University of Colorado, Boulder, CO 80309. Please direct comments or queries to Elizabeth Jessup at this address or e-mail jessup@cs.colorado.edu.

†The author can be reached at University of Paderborn, Informatik, FB 17, Warburgerstrasse 100, D-33102 Paderborn, Germany, domik@uni-paderborn.de.
such as offered by statistical analysis, image and signal processing, visualization attempts to explore all information inherent in scientific data in the most effective way. Therefore, scientific visualization is expected to enhance and increase scientific productivity.

Concepts and tools of scientific visualization are based on other disciplines: psychology/perception and human factors offer a scientific basis to understand human visual performance, its abilities and limitations. Experts in computer graphics provide algorithms and tools to transfer numerical data values into pictures. Artists and graphic designers offer their knowledge of aesthetics and other design issues to increase interpretability of visual representations. Scientists define their needs to explore scientific data and thus drive the quest for visual exploration. Scientific visualization provides concepts, methods and tools from existing disciplines to best use human abilities and computer algorithms for the display of scientific data.

On the other hand, there is a fine difference between the goals of visualization and goals of other disciplines or subdisciplines. While psychology and perception are important for understanding abilities and limitations of a scientist viewing a picture, basic principles of perception theories and the awareness of visual illusions, such as the Hermann grid or the Müller-Lyer illusion [Sekuler & Blake 85], do not fully explain the complex visual information present in a three-dimensional vector field visualization.

While image processing exclusively deals with images, it uses a limited number of visual representations (gray values or color pixel displays; shaded surfaces) to visually express the result of numerical algorithms. The chosen visual representations are not of essence to image processing, but rather the development of the underlying techniques, such as filters, geometric corrections, or image compression.

Computer vision is concerned with the computerized extraction of information from images [Boyle & Thomas 88]. It is therefore not of concern to Computer Vision how the human viewer extracts information from a picture, but rather how the computer may accomplish a similar task and initiate a certain action dependent on the result.

The field of computer graphics provides tools to design pictures from symbolic or numeric descriptions and to interact with these pictures [Hill 90]. Computer graphics is concerned with the development of algorithms (and their efficiency) to create pictures on a computer display. While computer graphics works hand in hand with visualization, it is not concerned with
pictures on displays once their appearance is satisfactory. The extraction of meaning from the picture in the human mind is not of concern to this field.

User interface issues have developed in parallel, but separately, from computer graphics [Olsen 92]. The wide availability of bitmapped graphics gave access to new visual appearances of user interfaces that have cumulated in the widespread use of windows and widgets. However, user interface design is not applying its methods to the understanding of processes and data, but rather to the ease-of-use of programs. In a similar way, human factors take into account the problems humans encounter when working with machines, not the output from these machines.

In having reviewed a series of areas that add to the understanding of visualization, we can conclude that scientific visualization cannot be replaced by existing disciplines, but it offers more than the sum of knowledge derived from these separate disciplines. It has therefore become a discipline of its own.

The aim of this tutorial is to discuss meaningful visual representations of scientific data, by providing a framework of underlying concepts in visualization. Even though the examples are focused on scientific data, concepts and techniques discussed here apply to other types of data as well, such as engineering data, financial data, computer software and hardware performance data.

2 History of scientific visualization

In the late 1980’s, data rates increased sharply both from measuring devices (such as data collecting missions in space or medical instruments) and as a result of computations on fast computers (such as supercomputers). In addition to the increase in the number of measuring devices, their resolution was multiplied. National supercomputers allowed access to virtually any scientist in the United States for large calculations. Computers and measuring devices add gigabytes (10^9) of data to the already existing amounts of data on a daily basis. Knowing that the trend to increase available data volumes will only continue, there is clearly a need to search for more efficient ways of dealing with large amounts of numbers.

Computer graphics received a boost in the mid 80’s through the development of improved and faster graphics hardware. New raster graphics tech-
Figure 1: Wind velocity vectors over the surface of a fluid body cause particles in the fluid to move.

Techniques replaced the previous technology of limited, slower vector graphics. Combined with powerful and affordable processors, personal graphics workstations emerged. The saying *A picture is worth a thousand words*, surpassed its promise: one picture might express several megabytes of data values.

In 1986, the National Science Foundation sponsored an advisory panel on *Graphics, Image Processing and Workstations* to make recommendations in response to the needs developed by high data rates and the opportunity of using the new generation of graphics workstations. The widely published report produced by the panel [McCormick et al 87] called for new tools in a new field termed *Visualization in Scientific Computing*, or short *Scientific Visualization*. Since 1987, a multitude of new applications have confirmed the necessity and power of this new methodology.

### 3 Example of scientific visualization

The following example shows various graphical ways to effectively visualize vector fields and the movement of particles.

Wind velocity vectors over the surface of a fluid body cause movement inside the fluid. We can derive the velocity of the fluid at control points. In order to observe the movement of the fluid better, we cut one two-dimensional (2D) slice through the three-dimensional (3D) body of fluid (figure 1).
Examples for effective graphical displays of the movement in the fluid along the 2D cut include:

- plotting vectors (using length, color, shape of arrows to distinguish magnitudes) to show the velocity of the fluid.
- animate the actual motion of particles suspended in the fluid.

An observer should be able to visually derive answers from questions such as:

- Is the movement symmetric?
- Where are the strongest/ weakest movements for time step  \( t = t_k \)?
- Where are the strongest/ weakest movements over time?
- How do the particles distribute over time?

Different graphical representations answer different questions that an observer might have.

The following figures were created to display the movement of the fluid using features of three different visualization tools: IDL (Interactive Data Language; AVS (Application Visualization System); and MATLAB. Figure 2 shows lines depicting information about strong and weak movements for one time step. Long rods indicate strong movements in the area surrounding the center, but not in the center or at the border. Figure 3 provides arrows signifying the direction of the velocities; the lengths of the arrows denote the magnitude of the velocities. Figure 4 is a combination of the two previous figures: the arrows denote the direction of the velocity; and both the size of the arrowheads and the length of the arrows indicate the magnitude of the velocity. Figure 5 shows only arrowheads: while the direction of velocity vectors is indicated by the arrowhead, color depicts their magnitude (short vectors received blue colors, medium vectors increased from green to red, and long vectors received orange and yellow colors). This visual representation clearly enhances the symmetry of the data set. Figure 6 is one frame of an animation of the data; it shows the positions of the particles after a certain time.
Figure 2: Lines depict information about strong and weak movements for one time step. Long rods indicate strong movements in the area surrounding the center, but not in the center. (This figure was created by Salim Alam using IDL.)
Figure 3: Arrows indicate the direction of the velocity and the length of the arrows denote the magnitude of the velocity. (This figure was created by Paul Pinkney using AVS.)
Figure 4: Arrows indicate the direction of the velocity while the size of the arrowheads and the length of the arrows indicate the magnitude of the velocity. (This figure was created by Anna Szczyrba using the `quiver` function in MATLAB.)
Figure 5: Velocity is shown by arrowheads: direction of velocity vectors is indicated by arrowhead; color depicts magnitude of vector (blue, green, red, yellow indicate growing vectors). This visual representation clearly enhances the symmetry of the data set. (This figure was created by Wolfgang Schildbach using IDL.)
Figure 6: Distribution of particles after a certain time. To observe the changes in their positions, an animation is most effective. (This figure was created by Salim Alam using AVS.)
4 Concepts of scientific visualization

4.1 Mapping numbers to pictures

Scientific Visualization is essentially a mapping process from one domain (a real phenomena) into another domain (numbers) into yet another domain (pictures) and further into a fourth domain (the subjective interpretation of the viewer) as shown in figure 7 [Domik & Gutkauf 94].

A scientist looking at a picture uses the picture as a vehicle of thinking [McKim 80], but intents to interpret the meaning of the numbers (or the real phenomena itself) expressed in the picture. The picture activates mental processes such as the perception of spatial relationships, the discovery of patterns or anomalies in large data sets, or the intuitive comprehension of complex processes. These mental processes are obviously different from the ones activated when interpreting numbers without the help of pictures.

In the following sections, we suggest a strategy to create expressive and effective pictures to ensure correct and meaningful interpretation by the viewer.

4.2 Expressiveness

The process of data interpretation becomes one step removed from the actual data themselves. If the mapping of numbers to pictures is not performed carefully, pictures might not express the true meaning of the underlying numbers, and therefore lead to misinterpretation of scientific facts. Examples of a non-intentional artifact might be a color coding of data values that produce abruptly changing hues from continuously increasing numbers. A picture is called expressive [Mackinlay 86] if it expresses the characteristics of the underlying data values, nothing more or less.

Expressiveness is strongly influenced by the structure of data (section 6) and their type of data values.
4.3 Effectiveness

If the mapping process is not performed purposefully, the resulting images might not be effective for the interpretation aims the scientist has in mind and therefore they are not useful. The visual representation of a two-dimensional data set in form of isolines (contour lines) is very effective when identifying local maxima, but very ineffective when trying to locate south-facing slopes. Effectiveness is strongly influenced by the interpretation goals of the scientist (section 10).

4.4 Subjectivity of interpreter

Visual cues are graphical elements of a picture that we visually separate out as a single entity, e.g., shape/form, a line and its orientation, or color [Keller & Keller 92]. Interpretation of visual cues can be subjective. The meaning of visual cues depends on culture, education, experience, and individual abilities and disabilities of the viewer. For instance, various interpretations of an underlying order of hue (hue does NOT have an inherent order in human perception) are strongly influenced by education. The sequence of hues

```
green - yellow - red - white
```

is interpreted as a sequence of increasing numerical values for most geographers and geologists. In order to reach the same conclusion, an astronomer would expect a color coding of

```
red - yellow - white.
```

The first sequence of hues relates to the color coding in a map (green meadows in flat areas, less growth at higher elevations as yellow, red rocks, white snow) to which geographers and geologists have adapted; the second sequence relates to the brightness of stars in a telescope (red for faint stars, yellow for brighter stars, white for strongest brightness), something astronomers can better relate to when interpreting visual cues.

4.5 A picture is the summary of visual cues

If the numerical data to be visualized are very simple, e.g., a list of numbers, mapping the numbers to one type of visual cue might suffice. For instance,
the numbers 13, 2, 15, 17, 29, 10 can be expressed as a histogram (using bars – positions, length – as visual cues) as demonstrated in figure 8.

In most cases, numerical data to be visualized might consist of complex data structures and many parameters. For example, data for aerodynamic research might consist of pressure (scalar values at 3D location), deformation (vectors distributed in 3D space), and shape (e.g., airplane). To visualize all available data, various visual cues need to be used. Each individual parameter needs to be mapped onto one or more such cues (e.g., shapes of arrows in 3D space; color; shaded rendering of airplane): the resulting picture is thus a summary of visual cues. In order to further the understanding of the relationship between parameters, the mapping should result in a coherent picture, where the picture could also be described as one entity.

5 Visual cues

Visual cues are elements of a picture. We can produce visual cues with the aid of our computer graphics tools. Perceptual counterparts of visual cues are called perceptual elements, and relate to our perception of visual cues. Examples of visual cues/perceptual elements are below:

1. spatial: position, motion
2. shape: length, depth, area, volume, thickness
3. orientation: angle, slope
4. density

5. color: color (hue, brightness, saturation), contrast

Complex visual cues are combinations of simple cues, e.g., a realistic picture of a natural scene is composed of objects of various shapes, sizes and colors at various positions.

5.1 Innate reactions to visual cues

Some visual cues are natural for us to interpret: if we increase the brightness on a series of objects, we interpret a natural ordering of the information from low to high, or less to more, or similar. The hue blue makes objects/information appear farther away, cooler, and lower than the hue red, that seems to relate to objects/information that are nearer, warmer, and higher. In part, the difference of our perception of blue vs. red is influenced by biological facts, as you could read in chapter 2 of the Lab Manual (Computer Graphics and Visualization).\(^1\)

5.2 Acquired reactions to visual cues

Interpretation of other visual representations are acquired through education, e.g., the interpretation of street signs, international travel signs, isolines or isosurfaces. Once we learn the meaning of these representations, this knowledge usually stays with us. The use of natural visual cues is advantageous, because it reduces the danger of misinterpretation. However, natural visual cues are usually too simple to use for the representation of complex information contents. Acquired visual cues are often powerful, but yet simple to interpret, once the knowledge to do so has been acquired.

5.3 Illusory visual cues

Some visual representations are known to fool the viewer: e.g., the illusory triangle, the Hermann grid, equal brightness steps, or simultaneous contrast. Look into chapter 2 of the Lab Manual to review some of these illusions. In order to avoid the danger of illusions, one must beware of their occurrence.

\(^1\)The Lab Manual chapters for the HPSC course are available via anonymous ftp at the cs.colorado.edu site in the /pub/HPSC directory.
6 Characterization of scientific data

Scientific data can be classified according to their semantics, e.g., data representing:

1. temperature dependent on location and time;
2. the DNA structure of a bacterium;
3. a digital elevation map;
4. surface of a space shuttle

usually carry different visual appearances. In order to effectively represent scientific data, knowledge about their semantics is important.

Data values can also be classified by data type as nominal, ordinal and quantitative [Mackinlay 86]. Nominal values describe members of a certain class, e.g., Iron, Magnesium, Calcium, Copper, Zinc]; no ordering can be imposed on this class, e.g., Iron is not larger or higher or earlier than Calcium. Visual cues that naturally relate to nominal data values are hue and position.

Ordinal values are related to each other by a sense of order: low density in growth; medium density in growth; high density in growth. Visual cues used to express ordinal values should depict this order, such as density, brightness, position, or size. If color is used, a color bar must be present.

Quantitative values carry a precise numerical value; scalar fields are often expressions of quantitative values, such as a three-dimensional MRI (Magnetic Resonance Imaging) data set. Even though color is often used to visualize quantitative data sets, this is in general a very imprecise visualization of the underlying values. The value in displaying quantitative information via color is that this transforms it to ordinal information (color bar or equivalent explanation must be present) and lets us quickly pick out low, medium or high values.

Visual representation of data is also strongly influenced by the syntax of the underlying data. The scientific data above may be syntactically represented by different dimensions and data structures, such as:

1. three-dimensional grid containing a floating-point number at each grid point for a series of time intervals: $y_t = f_t(x_1, x_2, x_3)$;
2. positions in 3D space, each position carrying information of its coordinate values and a category descriptor (type of molecule):
\[ y_t = (x_1^t, x_2^t, x_3^t, x_4^t); \]

3. a two-dimensional, even-gridded, array of integer values: \( y = f(x_1, x_2). \)

We need to classify the underlying structure of a data set in an easily comparable format. The following denotations of data sets and basic visualization techniques of section 7 are based on [Brodie et al 92].

Data sets \( D \) are described as \( D_{n}^{mC}(d) \), where

- \( n \) describes the dimensions of the data set(s)
- \( m \) describes the number of data sets defined over the same dimensions or the length of a tuple in a point data set
- \( C \) describes the category of data (\( P \) for single points, \( S \) for scalar data, \( V \) for vectors, \( T \) for tensors)
- \( d \) provides, if necessary, a more detailed description of \( C \). In the case of \( V \) or \( T \), \( d \) defines the length of vector or size of matrix, respectively; in the case of \( P \) or \( S \), \( q \) (quantitative), \( o \) (ordinal), or \( n \) (nominal), or a combination of these data types combined by '+', can be specified.

Valid data sets are, for example, \( D_2^S \) (two-dimensional scalar data set), or \( D_2^{3S}(2q + n) \) (three data sets defined over the same two dimensions; two data sets contain quantitative values, one contains nominal values); \( D_3^V \) (one vector field in three-dimensional space with each vector of length three); \( D_3^T(5q + o + n) \) (point data set with each point described by seven values, five of which are of quantitative type, one ordinal and one nominal).

### 6.1 Points

A set of points \([P_1, P_2, P_3, \ldots]\) is denoted as \( D^{mP} \). An example is \( D^P = [1, 5, 8, 33, 4, 15] \). The sets \( D^{2P} \) (e.g., \([(1, 2), (3, 4), (5, 5), (4, 4), (8)]\)) and \( D^{3P} \) (e.g., \([(1, 2, 3), (4, 5, 6), (4, 5, 6, 8, 0), (6, 0, 4.5)]\)) are pairs and triplets of numbers, respectively. Expressive visualizations for point data sets are scatter plots \((m \leq 3)\) and glyphs (see section 7.2). In the case of position information, as in molecular dynamics, an obvious visual presentation of the information is a map of suitable objects (e.g., spheres) to the indicated positions.
6.2 Scalars

Scalars are usually samples of a continuous function. In this tutorial, we are assuming that we sample a function in equidistant steps in each dimension in order to receive a discrete data set. In reality, sampling is often done in random order. In such a case, we assume that data can be resampled in equidistant steps. Data sets $D$ based on continuous functions can be characterized by functions $f$ of independent ($x$) and dependent ($y$) variables:

$$y_i = f_i(X), \text{ where } X = (x_1, x_2, ..., x_n); i = [1, ..., m]$$

The so represented functions can take on various shapes: $y = f(x_1, x_2)$ is a two-dimensional, scalar function, and the data set it produces is denoted by $D_2^S$; in this tutorial we usually work with $D_1^S, D_2^S$ and $D_3^S$.

In the one-dimensional (1D) case, $D_1^S$, data are sampled from a one-dimensional function, $y = f(x)$. Positions of the data values are therefore determined from the variable $x$. Typical representations are line drawings, scatter plots, or histograms.

$D_2^S$ is data sampled from a two-dimensional function, $y = f(x_1, x_2)$. A typical example is a digital topographic map (= digital elevation model), where $x_1, x_2$ denote the sampling locations (usually an even two-dimensional grid) and $y$ represents the elevation at this location. Examples of visual representations in this category are contour lines, wire frame models, shaded surfaces, and images. The latter category, images, denotes the visual representation of each $y$ value as one pixel on the screen, either in gray shades or color.

Several scalars can be available at the same location, e.g., $y_1 = f_1(x_1, x_2)$ and $y_2 = f_2(x_1, x_2)$, where $y_1$ denotes height and $y_2$ denotes density of growth. The resulting data set may be defined as $D_2^{2S} (D_2^{2S}(q + o))$ in our notation. A valid representation of this entity is a two-dimensional wire frame plot to indicate $y_1$ by surface height and the use of color (presence of color bar necessary to display ordinal data!) to indicate $y_2$ within the same picture. This not only allows the representation of both variables at the same time, it also enhances the understanding of the relationship between $y_1$ and $y_2$: if lowest elevations have the densest growth and highest elevations have the least dense growth, this becomes obvious in the picture.

Often a series of data measurements are conducted over one area. This leads to data sets defined as $D_2^{mS}$. Visual cues that can be superimposed in
a meaningful way (such as hue and brightness; color and shaded view; image and contour lines) are limited. With larger \( m \), glyphs become very useful to indicate each individual scalar but also relate all scalars belonging to the same location in the two-dimensional space to each other.

Data sets of type \( D^{(m+2)P} \) can be converted to data sets of type \( D^{mS} \) (and vice versa). \( D^{mS}_2 \) can be treated as a special case of \( D^{(m+2)P} \) in that two values of the \((m + 2)\) tuple are treated as spatial positions.

\( D^{mS}_2 \) can also be seen as a three-dimensional scalar field \( (D^S_3) \), with \( m \) overlapping 2D slices. Possible visualizations for this entity therefore also include overlapping (with or without regard to hidden lines/surfaces) of visualizations fitting for \( D^S_2 \).

Similar to \( D^S_2 \), data that are sampled from a three-dimensional function, \( y = f(x_1, x_2, x_3) \), is denoted as \( D^S_3 \). Tools to visualize a regular three-dimensional lattice of points have recently been developed in computer graphics. There are basically two approaches to finding a visual representation of three-dimensional points:

1. if the importance is on clusters of data points, the view of the surface of these clusters might suffice (e.g., isosurfaces); or

2. if the importance is on the visual representation of every single data point, the volume directly needs to be displayed (e.g., ray tracing of volumes). Furthermore, subsets in the form of two-dimensional slices of the volume can be displayed as suggested for \( D^S_2 \), or as suggested for \( D^{mS}_2 \), if more than one slice is to be displayed simultaneously.

Note the similarity of \( D^{mS}_3 \) and \( D^{mP} \) as noted for the two-dimensional case above.

### 6.3 Vectors

The expression \( y = f(x_1, x_2, x_3) \) might describe a function of vectors, with a resulting data set of \( D^V_2 \), if each independent variable is of length two. Vectors need to be defined by the dimension of each vector as well as by the dimensionality of the data set containing vectors: \( D^V_2 \) defines a set of three-dimensional vectors in a two-dimensional plane.

Possible displays for \( D^V_2 \) are arrows, indicating length and direction of each two-dimensional vector.
Similar to the scalar cases, $D_2^{mV_2}$ indicates the availability of $m$ vectors, each of dimension 2, at any 2D position of interest. Animations are expressive visualizations for this category, usually expressing a time series of vector data information.

Three-dimensional vector visualization, $D_3^{V_3}$ or $D_3^{mV_3}$, uses 3D shapes of arrows, particle traces and streamlines to display available data. Again, animation can be used to express time or other information.

6.4 Tensor fields

An n-dimensional tensor field is denoted as $D_n^T$, with $k$ describing the order of the tensor. One example of a symmetric, second order tensor display is given in [Haber & McNabb 90]: first, the tensor’s principal directions and magnitudes are calculated. A cylindrical shaft is oriented along the major principal direction; the color and length of the shaft indicate the sign and magnitude of the stress in this direction. An ellipse wraps around the central portion of the shaft and its axes correspond to the middle and minor principal directions of the stress tensor. The color distribution of the disk indicates the stress magnitude in each direction.

7 Visualization techniques

7.1 Scatter plots

Scatter plots use position as their primary visual cue. A one-dimensional scatter plot uses positions along one axis to indicate values of $D^P$, e.g.,

$$D^P = [P_1 = 1; P_2 = 2; P_3 = 3; P_4 = 4; P_5 = 0; P_6 = 6; P_7 = 7; P_8 = 8; P_9 = 8; P_{10} = 7; P_{11} = 8; P_{12} = 7]$$

might be measurements taken independently.

The scatter plot in figure 9 uses a linear mapping between data values and y-position; often a logarithmic scale is being used. For any non-linear mapping (and, if appropriate, also for linear mappings), the mapping function [data value → position on plot] should be annotated.
A two-dimensional scatter plot relates pairs of points in a two-dimensional coordinate grid. A data set \((D^{2P}(2q))\) might be given as

\[
[(3, 3), (3, 4), (4, 4), (5, 4), (5, 5), (6, 4), (6, 5), (7, 5), (7, 8), (8, 8), (9, 9), (12, 8)]
\]

and represent in a linear manner the relationship of measurements of 12 randomly picked leaves (figure 11).

A similar data set, \(D^{2P}(q + n)\), is expressed in figure 11. It depicts leaf width \((q)\) and leaf type \((n)\). For nominal information, symbols or color can be very effective to indicate an additional dimension. For ordinal or quantitative information, size or orientation might be the more appropriate indicator. The influence of data type on effective visual cue is discussed in more detail in section 6.

If measurements of leaf length, width, and thickness of leaves are available, we receive triples of numbers instead of pairs: \(D^{3P}(3q)\). Three-dimensional scatter plots are ambiguous on two-dimensional screens. However, if the view point is mobile (e.g., the view is constantly rotating around the scattered points), a good three-dimensional impression of the point cloud can be obtained. If points on the screen are represented by larger objects, depth-cueing (loss of reflection with distance from viewer), perspective viewing (decrease of object size with distance from viewer) is recommended. Other possibilities are the use of stereo equipment or even virtual reality environments to observe the location of each point in 3D space.

---

\(\textit{CUBoulder: HPSC Course Notes}\)
7.2 Glyphs

Sometimes several variables or constants are used to describe each dimension in a data set. Such data sets are of form $D^{mP}$ (e.g., auto parts; computer hardware or software performance measurements; census data), or $D^{mS}_2$ (e.g., velocities of particles in 2D space at $m$ time steps; absorption of light on 2D surface for $m$ different wavelengths) or $D^{mS}_3$.

Glyphs (called textons in perception research) have been invented to specifically express such complex data sets. One glyph is composed of individual parts or segments. Each part or segment is seen as one visual cue and relates to one data value. Data values that should appear in the same spatial position on the screen are mapped into individual parts of the glyph. One glyph is usually identified by the viewer as one figure (relating to the sum of all parts of the glyph). In our example in figure 12 we look at 9 data items, each consisting of a quadruple of data values. The data set is therefore of form $D^{4P}$, and appropriate visualizations might include color and symbols combined with a two-dimensional scatter plot. The glyph we are using for a visual representation of the data sets is composed of four individual parts:
Numerical data

\[ D_{m}^{4P} (3 o + n) = \]

P_1 = [2, 1, 0, 1]
P_2 = [1, 1, 0, 1]
P_3 = [0, 0, 1, 2]
P_4 = [2, 0, 2, 2]
P_5 = [1, 1, 0, 0]
P_6 = [2, 1, 0, 2]
P_7 = [0, 0, 1, 2]
P_8 = [2, 0, 0, 2]
P_9 = [2, 1, 1, 1]

Visual entities

<table>
<thead>
<tr>
<th>m = 1</th>
<th>m = 2</th>
<th>m = 3</th>
<th>m = 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Visualization of data

Figure 12: The use of glyphs to depict a data set of type $D_{m}^{P}$.

Figure 13: Three line graphs distinguished by line style.
a pair of circles (large, medium or small, depending on the first data value in the quadruple); one vertical line (long, medium or short, depending on the second data value); brightness (white, medium gray or black, depending on the third data value) and curve (concave, straight or convex, depending on the fourth data value). These four visual entities form a face that can be recognized as a new single entity as a smiling, sad, mean or unintelligent face. The search for similar quadruples is now reduced to matching similar faces.

Faces have shown to be powerful glyphs for small data sets [Chernoff 73]. Glyphs for larger data sets are suggested by [Picket & Grinstein 88] and by [Beddow 90].

Looking beyond visualization, one can also use senses other than our visual sense to express data values. Smith et al [Smith et al 90] and Scaletti and Craig [Scaletti & Craig 91] use sound as an additional visual cue.

7.3 Line graphs

Line graphs are used to display continuous information and are therefore an effective visual representation of scalar data sets of form $D^S_1$ and of form $D^{mS}_1$. For $D^{mS}_1$, $m$ lines—differing by color, style or thickness—can be displayed on the same output media (see figure 13).

7.4 Histograms, pie charts

A histogram looks like a discrete line graph (see figure 8). The area of each rectangle (length of each histogram bar) is of special meaning to the observer, usually relating to the number of occurrences of an event. The location of each histogram bar usually relates to a certain category or class of information (data). The total area of all rectangles also carries meaning in a histogram, namely to describe the number of all occurrences in all categories.

If information is compared to some total number, then a constant comparison between each data value and the whole data set should be possible, such as in figure 14.

For this type of visual representation, pie charts as depicted in figure 15 (circles to depict the whole) are typically in use.
October 1992 (in hours)

2 hours

Figure 14: Relating information to the total sum.

Figure 15: Relating information to the total sum: pie chart.
7.5 Contour plot (isolines)

For functions $y = f(x_1, x_2)$ that produce data sets of type $D_2^S$, contour plots can be used to show lines of constant values (= threshold values) inside the two-dimensional data set: $y = c$. The full data set can be visually expressed by contour lines $y = [c_1, c_2, c_3, ..., c_n]$, where distances between threshold values are usually equidistant (figure 16). Contour lines are also called isolines because they connect points of equal values.

7.6 Image display

Image displays are frequently chosen as visual representations of quantitative data of type $D_2^S$. For raster devices, image displays are a straightforward mapping of each data point along a two-dimensional grid into a pixel (= picture element) on the screen. To indicate the data value at the pixel location, gray levels or color can be chosen. If color is used, simultaneous use of a color bar (see section 8 on Annotations) is mandatory. Figure 17 displays data of a digital elevation model of the area surrounding Boulder, Colorado.
as a color image.

A combination of contour lines and images are often used to enhance interpretability of data. Redundancy of visual techniques produce, in general, desirable results. Figure 18 shows a digital elevation model of the same area around Boulder overlaid with isolines.

7.7 Surface view

Wire frame drawing is a fast drawing method to depict surfaces of data sets of type $D_2^S$. The numeric values of the data set are treated as elevations, and visually depicted as a terrain, showing peaks for local maxima and valleys for local minima in the data set (figure 19). The surface of the terrain is drawn as if it were made of wire; hidden surfaces are often not eliminated to add efficiency to the drawing algorithm. Projection of the three-dimensional surface onto the two-dimensional screen allows a natural interpretation of the data values, similar to standing on a mountain and observing the surrounding terrain.

In a similar but more time-consuming algorithm, the surface can be shaded by an artificial light source to lend more realism to the display (figure 20). In this case hidden surfaces are always removed, and only front views are visible. Shading does not necessarily involve color: the amount of light reflected is a function of light source, viewpoint and surface characteristics, and may be expressed by the brightness of a pixel on the screen.

In all surface views, but specifically if data are noisy or characterized by high-frequency data values, animation of the scene supports the interpretation. Moving the view point around the three-dimensional scene gives visibility to otherwise hidden parts of the surface view. Surface views are recommended for smooth data sets. In noisy data sets, visibility can be strongly diminished.

7.8 Color transformations

Data sets of type $D_2^S$ can be displayed as one single color image, if each single data set $D_2^S$ is mapped into one color dimension. Color dimensions are independent (orthogonal) characteristics of color models, e.g., red, green, blue in the RGB color cube, or hue, saturation, value in the HSV model (e.g., [Foley et al 90] or chapter 2 of the Lab Manual). For an example, look at the
Figure 17: This image display showing the contour of the terrain around Boulder, Colorado was created with IDL, using data derived from the `denver-w` and `greeley-w` files available at the USGS/EROS Data Center anonymous FTP server for the United States Geological Survey.

Figure 18: IDL image display and contour lines showing the area around Boulder, using the same data as in figure 17.
Figure 19: Wire frame surface of the area surrounding Boulder, Colorado. This was created with IDL with the same data set used in figures 17 and 18.

Figure 20: Shaded surface view of area around Boulder, Colorado created with IDL from the same data used in figures 17, 18, and 19.
RGB color transformation in figures 21 and 22 below. Figure 21 shows four separate images (astrophysical data collected at different wavelengths), from which three were chosen to be superimposed on the screen: the lower right image was displayed on the red phosphors; the lower left image on the green phosphors; and the upper left image on the blue phosphors of a workstation monitor. The resulting image in figure 22 displays values from the color gamut on the available output device.

An interpretation of color transformations is fairly simple, if the underlying color model is well understood. In our case, yellow objects indicate high data values in the red image and in the green image at the corresponding spatial location; dark blue objects indicate low values in the red image and green image but high data values in the blue image.

7.9 Isosurfaces

In data sets of type $D_3^S$ or similar, we assume that data are available in a regular lattice in 3D. When using isosurfaces to visually represent volumetric data, the assumption is given that traceable objects are hidden inside the 3D volume. Similar to isolines, surfaces of constant values are identified and illuminated, shaded and projected onto the two-dimensional screen. Figure 23 shows one isosurface of an MRI (Magnetic Resonance Imaging) data set.

7.10 Ray tracing of volumes

If data values inside a volumetric data set, $D_3^S$, are not expected to form solid objects, a view of the whole volume is usually preferred. This is done by assigning opacity and color to each volume element (voxel) and displaying all of the elements inside a volume. The projection from three dimensions to two dimensions is done via ray casting (ray tracing) from a point-of-view through the volume. Each voxel touched by the same ray adds to the resulting color of its representative image point. Each ray results in one image pixel to be displayed on a two-dimensional screen. Ray-traced images of volumes (see figure 24) can assume a transparent, cloud-like appearance (translucency).
Figure 21: Astrophysical data from NASA’s Infrared Astronomy Satellite (IRAS) mission of 1983 varying only in their spectral response.
Figure 22: RGB color transformation of the three IRAS data sets of figure 21.
Figure 23: Rendered isosurface of MRI data created by IDL. Data for this figure may be obtained via anonymous ftp at omicron.cs.unc.edu, provided through the courtesy of Siemens Medical Systems, Inc., Iselin, NJ.
Figure 24: Volume visualization (translucent display) from the same data sets as depicted in figure 23. (This figure was created by Michael Kreutner using IDL.)
Figure 25: Use of IDL’s data slicer. Three slices show subsets of a $D^S_3$ data set.

### 7.11 Data slicers

Any data sets of higher order can be explored by observing subsets of the data. This is specifically meaningful with volumetric data, where a view of a two-dimensional slice through the three-dimensional volume provides a view of the inside of the data set. Figure 25 shows three slices through a data set of type $D^S_3$.

### 7.12 Arrows

Data sets of type $D^Y_2$ can be displayed as arrows in a two-dimensional plane. An arrow is a symbol that can, at the least, display dual information: one type of information (e.g., magnitude) is expressed by the length of the arrow; and one type of information (e.g., direction) is expressed by the slope of the arrow. Other visual cues can express additional information, e.g.,
• starting location of arrow
• thickness of shaft
• color of arrow.

The danger in this type of graphical display is that some of the vectors might get enormously long and interfere with vectors in the surroundings. (For example, see figure 2.) If scaling of the vectors is used, so that there is sufficient distance between neighboring arrows, some of the arrows might become too small for visual interpretation. Alternative displays of vectors, such as shown in figure 5, may be very useful.

It might be of advantage to combine a vector display with a scalar visualization, by extracting one scalar information from the vector field, e.g., magnitude, and superimposing the arrows with an appropriate visual representation of the scalar field, e.g., image display.

Visual representation of data sets of type $D_2^V$ are an extension of the above explained technique. Arrows are rendered as three-dimensional shapes that point into or out of the display surface. Similarly, $D_3^V$ is visualized by displaying shapes of three-dimensional arrows inside a three-dimensional volume. For a more realistic display, depth-cueing (reflection loss with distance) should be used.

### 7.13 Streamlines and particle tracks

Streamlines or particle tracks trace a direction, e.g., the direction of a flow computed for an application in computational fluid dynamics (CFD). This method can be seen as an extension of arrows, in that arrows are added onto each other and create polylines (= a set of contiguous straight line segments approximating a curve). This method is useful to depict vector data over a certain time. This method can be used in two as well as three dimensions.

### 7.14 Animation

Animation can become a visual cue as well. Because of its analogy to nature (things change over time), time is often expressed in form of an animation. An example might be the distribution of carbon monoxide in Denver: the
data set of type $D^3$, showing a snapshot of the distribution at 5 am, is depicted as a translucent volume. The distribution of each following hour is now displayed in consecutive frames, using animation to move from one picture to the next. Animation shows a continuous sequence of visual representations, allowing the viewer to observe changes between pictures. Changes between pictures might contain as much information for the scientist as the pictures themselves.

8 Annotations

A visualization might be unreadable because certain information is omitted. If color is used, a color scale (usually a bar relating color and corresponding data values) must be present. Furthermore, the scaling of world coordinates (the coordinate system used to describe the problem) to screen coordinates must be documented by a scale bar if there is any relevance in this information. If spatial dimensions are expressed on the screen, orientation must be presented to the viewer; e.g., a North arrow, or an indication of the vertical and horizontal directions. The use of animation to express a time series (or spectral series) of data must be annotated by a time indicator (or spectral indicator) or any other appropriate indicator specifying the relationship of picture and data variable. Too much clutter on the screen distracts from essential information, but omitting explanatory notes and cues might make visual representation useless.

9 Interactivity

Interactivity describes actions the user is able to perform on graphical representations. Interactivity enhances the value of visualizations by letting the user explore data and steer visualization processes. Examples of interactions are

- using the mouse to click on objects on the screen to request (often numerical) information about these objects;
- changing parameters of data sets to control physical phenomena;
• moving/rotating objects in three-dimensional space (e.g., three-dimensional scatter plot) to lend a 3D effect to two-dimensional screens.

10 Interpretation goals to pursue with visualization

Besides syntax and semantics of data sets to be visualized, the interpretation goal of the scientist should influence the choice of visual representation. The interpretation goal defines the task a scientist has set out to do with the help of visualization: e.g., identify local maxima in the data set; observe the behavior of one variable \( y_1 \) in relationship to another one \( y_2 \); observe symmetry in the data set; etc. [Wehrend & Lewis 90].

While a contour plot is an effective visual tool to identify areas of a certain threshold value, an image display of the same data set is ineffective for the same purpose. In general, a surface plot is more effective in classifying slopes in a data set than an image display. However, an image display has no hidden surfaces and can thus depict even noisy data. Clever use of color tables can enhance image displays for various tasks, e.g., quickly locating values of specific characteristics.

It is impossible to list all interpretation aims and give hints about effective visual representations, because options of interpretation aims are too numerous. However, the visualizer must carefully consider the use of the visual representation BEFORE time is spent in encoding the data.

11 Quantitative versus qualitative data interpretation

Clearly, the picture does not always tell us the whole story about underlying information. Quantitative data interpretation gives us precision that is often necessary to prove/disprove a theory. A mean or median value, or maximum/minimum numbers in a large array may characterize the data set sufficiently and more accurately than any pictorial information. On the other hand, pictures are supplementary to numerical information in that they convey information contained in the data, but not obvious when interpreting
numbers. Approximate symmetry in a large data set might be hard to derive from numbers alone, but can be obvious when looking at a visual representation. Likewise, similarities, patterns, or sudden deviations might be effectively represented in the picture, but hardly noticeable in the numbers.

For most scientific purposes, a combination of pictures and numbers is needed to take advantage of all information that can be extracted from a data set. This fact bears a strong message on writing software for visualization purposes: do not lose the numbers while creating pictures! In other words, if a scientist clicks on various molecules on his/her screen in order to receive quantitative information about the molecules, the information sought for is not the shading value or reflection constant of the spheres representing the molecules, but rather information about the chemical character of the molecule.
References


