UNIT-V

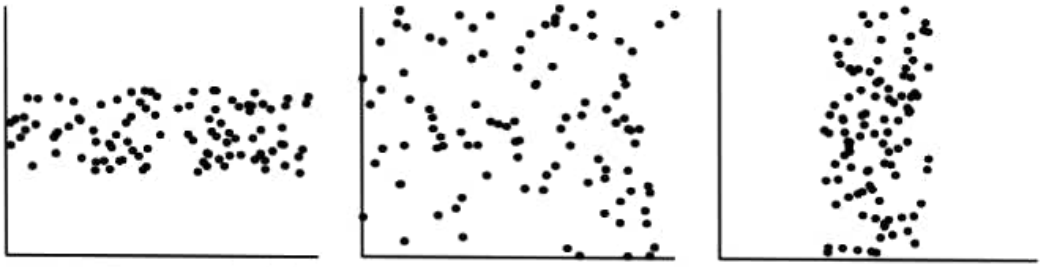


Figure 2.9 Three cases where there is no observed correlation between the two plotted attributes in each of the data sets.

from lower left to upper right, this means that the values of X increase as the values of Y increase, suggesting a *positive correlation* (Figure 2.8a). If the pattern of plotted points slopes from upper left to lower right, the values of X increase as the values of Y decrease, suggesting a *negative correlation* (Figure 2.8b). A line of best fit can be drawn to study the correlation between the variables. Statistical tests for correlation are given in Chapter 3 on data integration (Eq. (3.3)). Figure 2.9 shows three cases for which there is no correlation relationship between the two attributes in each of the given data sets. Section 2.3.2 shows how scatter plots can be extended to n attributes, resulting in a *scatter-plot matrix*.

In conclusion, basic data descriptions (e.g., measures of central tendency and measures of dispersion) and graphic statistical displays (e.g., quantile plots, histograms, and scatter plots) provide valuable insight into the overall behavior of your data. By helping to identify noise and outliers, they are especially useful for data cleaning.

2.3 Data Visualization

How can we convey data to users effectively? **Data visualization** aims to communicate data clearly and effectively through graphical representation. Data visualization has been used extensively in many applications—for example, at work for reporting, managing business operations, and tracking progress of tasks. More popularly, we can take advantage of visualization techniques to discover data relationships that are otherwise not easily observable by looking at the raw data. Nowadays, people also use data visualization to create fun and interesting graphics.

In this section, we briefly introduce the basic concepts of data visualization. We start with multidimensional data such as those stored in relational databases. We discuss several representative approaches, including pixel-oriented techniques, geometric projection techniques, icon-based techniques, and hierarchical and graph-based techniques. We then discuss the visualization of complex data and relations.

2.3.1 Pixel-Oriented Visualization Techniques

A simple way to visualize the value of a dimension is to use a pixel where the color of the pixel reflects the dimension's value. For a data set of m dimensions, **pixel-oriented techniques** create m windows on the screen, one for each dimension. The m dimension values of a record are mapped to m pixels at the corresponding positions in the windows. The colors of the pixels reflect the corresponding values.

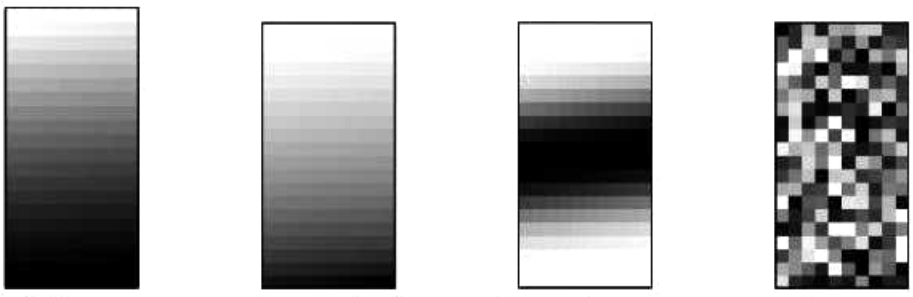
Inside a window, the data values are arranged in some global order shared by all windows. The global order may be obtained by sorting all data records in a way that's meaningful for the task at hand.

Example 2.16 Pixel-oriented visualization. *AllElectronics* maintains a customer information table, which consists of four dimensions: *income*, *credit_limit*, *transaction_volume*, and *age*. Can we analyze the correlation between *income* and the other attributes by visualization?

We can sort all customers in income-ascending order, and use this order to lay out the customer data in the four visualization windows, as shown in Figure 2.10. The pixel colors are chosen so that the smaller the value, the lighter the shading. Using pixel-based visualization, we can easily observe the following: *credit_limit* increases as *income* increases; customers whose income is in the middle range are more likely to purchase more from *AllElectronics*; there is no clear correlation between *income* and *age*. ■

In pixel-oriented techniques, data records can also be ordered in a query-dependent way. For example, given a point query, we can sort all records in descending order of similarity to the point query.

Filling a window by laying out the data records in a linear way may not work well for a wide window. The first pixel in a row is far away from the last pixel in the previous row, though they are next to each other in the global order. Moreover, a pixel is next to the one above it in the window, even though the two are not next to each other in the global order. To solve this problem, we can lay out the data records in a space-filling curve



(a) *income* **(b)** *credit_limit* **(c)** *transaction_volume* **(d)** *age*

Figure 2.10 Pixel-oriented visualization of four attributes by sorting all customers in *income* ascending order.

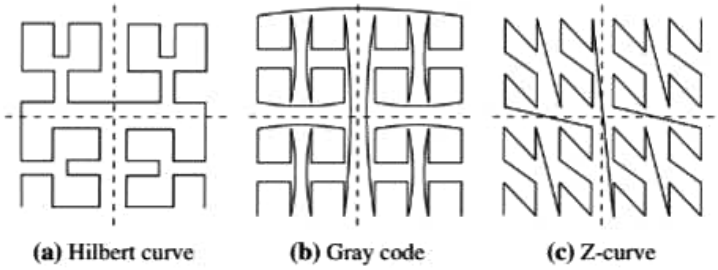


Figure 2.11 Some frequently used 2-D space-filling curves.

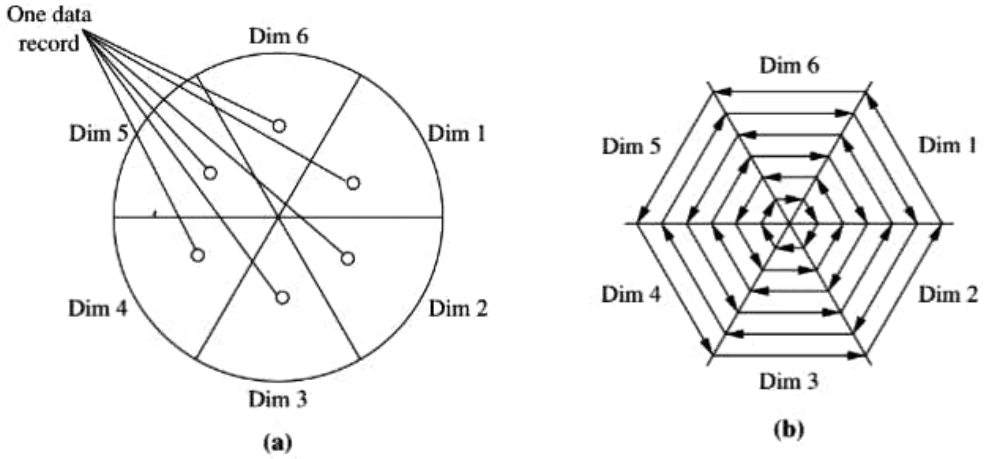


Figure 2.12 The circle segment technique. (a) Representing a data record in circle segments. (b) Laying out pixels in circle segments.

to fill the windows. A *space-filling curve* is a curve with a range that covers the entire n-dimensional unit hypercube. Since the visualization windows are 2-D, we can use any 2-D space-filling curve. Figure 2.11 shows some frequently used 2-D space-filling curves.

Note that the windows do not have to be rectangular. For example, the *circle segment technique* uses windows in the shape of segments of a circle, as illustrated in Figure 2.12. This technique can ease the comparison of dimensions because the dimension windows are located side by side and form a circle.

2.3.2 Geometric Projection Visualization Techniques

A drawback of pixel-oriented visualization techniques is that they cannot help us much in understanding the distribution of data in a multidimensional space. For example, they do not show whether there is a dense area in a multidimensional subspace. **Geometric**

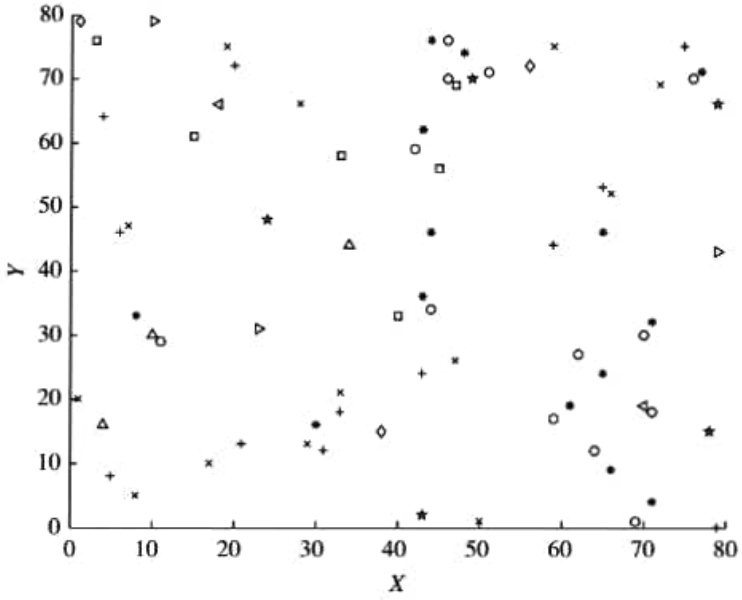


Figure 2.13 Visualization of a 2-D data set using a scatter plot. *Source: www.cs.sfu.ca/jpei/publications/rareevent-geoinformatica06.pdf*.

projection techniques help users find interesting projections of multidimensional data sets. The central challenge the geometric projection techniques try to address is how to visualize a high-dimensional space on a 2-D display.

A **scatter plot** displays 2-D data points using Cartesian coordinates. A third dimension can be added using different colors or shapes to represent different data points. Figure 2.13 shows an example, where X and Y are two spatial attributes and the third dimension is represented by different shapes. Through this visualization, we can see that points of types "+" and "×" tend to be colocated.

A 3-D scatter plot uses three axes in a Cartesian coordinate system. If it also uses color, it can display up to 4-D data points (Figure 2.14).

For data sets with more than four dimensions, scatter plots are usually ineffective. The **scatter-plot matrix** technique is a useful extension to the scatter plot. For an n-dimensional data set, a scatter-plot matrix is an $n \times n$ grid of 2-D scatter plots that provides a visualization of each dimension with every other dimension. Figure 2.15 shows an example, which visualizes the Iris data set. The data set consists of 450 samples from each of three species of Iris flowers. There are five dimensions in the data set: length and width of sepal and petal, and species.

The scatter-plot matrix becomes less effective as the dimensionality increases. Another popular technique, called parallel coordinates, can handle higher dimensionality. To visualize n-dimensional data points, the **parallel coordinates** technique draws n equally spaced axes, one for each dimension, parallel to one of the display axes.

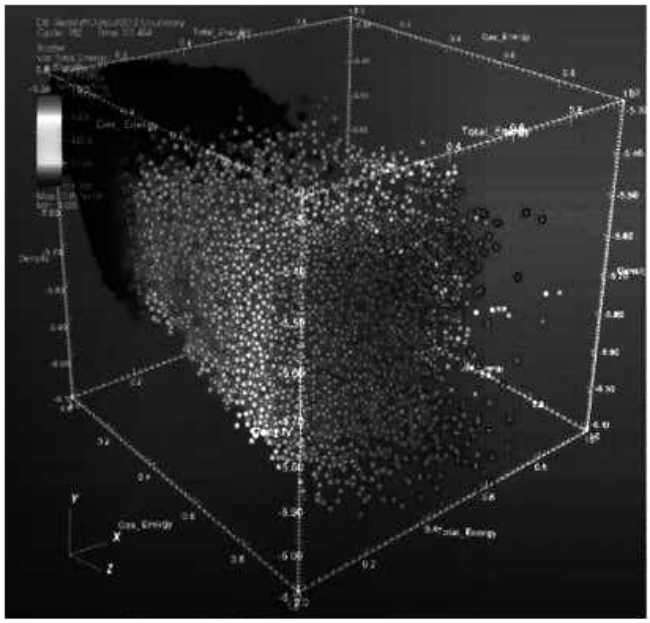


Figure 2.14 Visualization of a 3-D data set using a scatter plot. *Source: http://upload.wikimedia.org/wikipedia/commons/c/c4/Scatter_plot.jpg.*

A data record is represented by a polygonal line that intersects each axis at the point corresponding to the associated dimension value (Figure 2.16).

A major limitation of the parallel coordinates technique is that it cannot effectively show a data set of many records. Even for a data set of several thousand records, visual clutter and overlap often reduce the readability of the visualization and make the patterns hard to find.

2.3.3 Icon-Based Visualization Techniques

Icon-based visualization techniques use small icons to represent multidimensional data values. We look at two popular icon-based techniques: *Chernoff faces* and *stick figures.*

Chernoff faces were introduced in 1973 by statistician Herman Chernoff. They display multidimensional data of up to 18 variables (or dimensions) as a cartoon human face (Figure 2.17). Chernoff faces help reveal trends in the data. Components of the face, such as the eyes, ears, mouth, and nose, represent values of the dimensions by their shape, size, placement, and orientation. For example, dimensions can be mapped to the following facial characteristics: eye size, eye spacing, nose length, nose width, mouth curvature, mouth width, mouth openness, pupil size, eyebrow slant, eye eccentricity, and head eccentricity.

Chernoff faces make use of the ability of the human mind to recognize small differences in facial characteristics and to assimilate many facial characteristics at once.

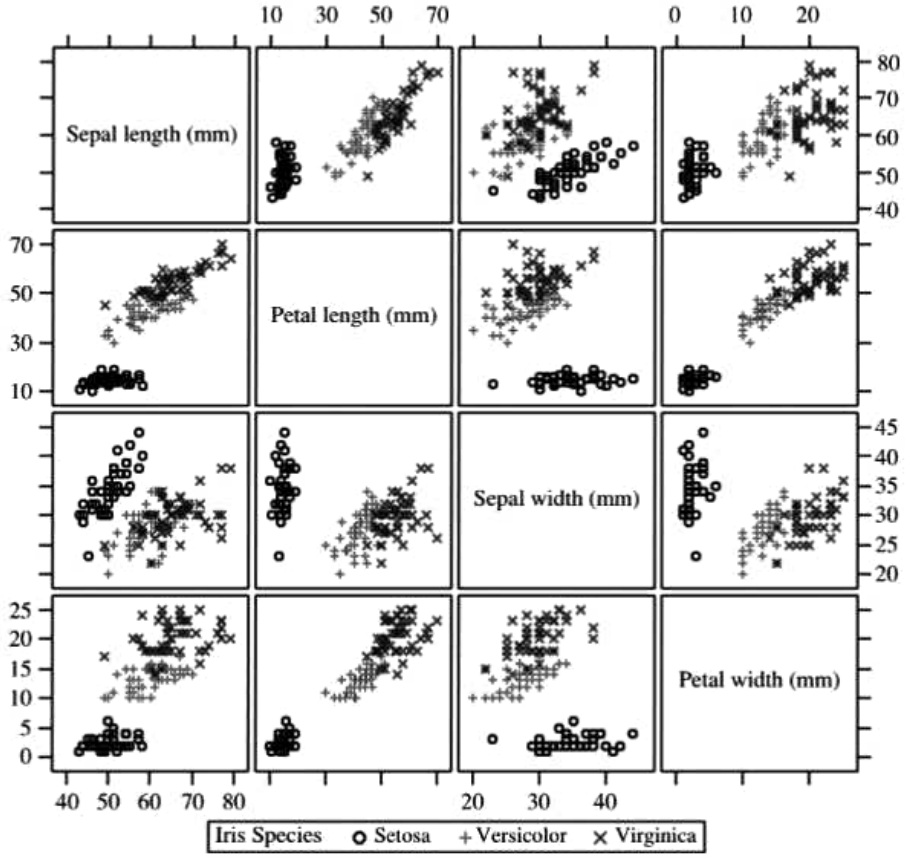


Figure 2.15 Visualization of the Iris data set using a scatter-plot matrix. *Source: http://support.sas.com/documentation/cdl/en/grstatproc/61948/HTML/default/images/gsgscmat.gif*.

Viewing large tables of data can be tedious. By condensing the data, Chernoff faces make the data easier for users to digest. In this way, they facilitate visualization of regularities and irregularities present in the data, although their power in relating multiple relationships is limited. Another limitation is that specific data values are not shown. Furthermore, facial features vary in perceived importance. This means that the similarity of two faces (representing two multidimensional data points) can vary depending on the order in which dimensions are assigned to facial characteristics. Therefore, this mapping should be carefully chosen. Eye size and eyebrow slant have been found to be important.

Asymmetrical Chernoff faces were proposed as an extension to the original technique. Since a face has vertical symmetry (along the y-axis), the left and right side of a face are identical, which wastes space. Asymmetrical Chernoff faces double the number of facial characteristics, thus allowing up to 36 dimensions to be displayed.

The **stick figure** visualization technique maps multidimensional data to five-piece stick figures, where each figure has four limbs and a body. Two dimensions are mapped to the display (x and y) axes and the remaining dimensions are mapped to the angle

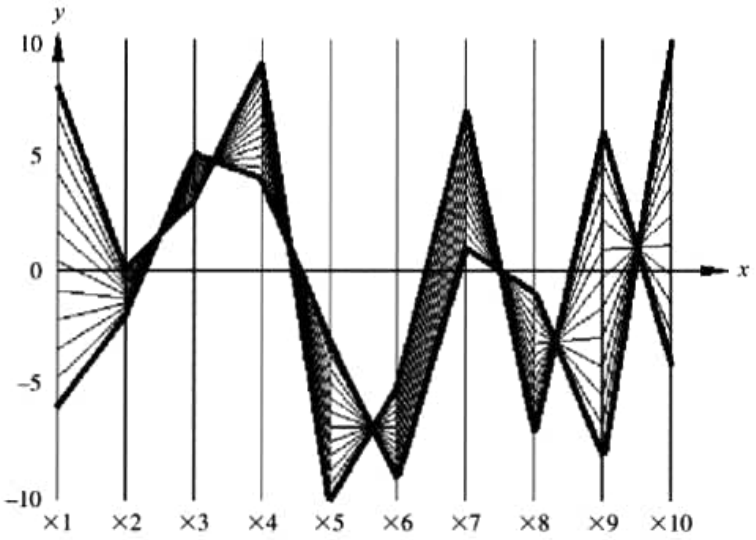


Figure 2.16 Here is a visualization that uses parallel coordinates. *Source: www.stat.columbia.edu/~cook/movabletype/archives/2007/10/parallel_coordi.thml.*

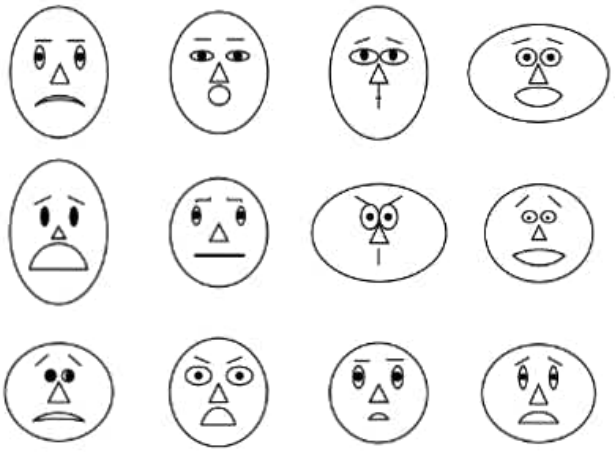


Figure 2.17 Chernoff faces. Each face represents an n-dimensional data point ($n \leq 18$).

and/or length of the limbs. Figure 2.18 shows census data, where *age* and *income* are mapped to the display axes, and the remaining dimensions (*gender, education*, and so on) are mapped to stick figures. If the data items are relatively dense with respect to the two display dimensions, the resulting visualization shows texture patterns, reflecting data trends.

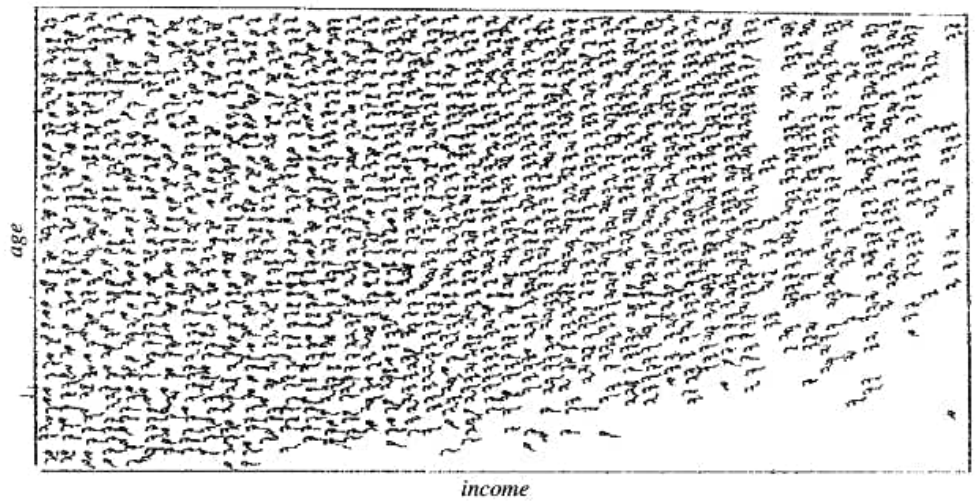


Figure 2.18 Census data represented using stick figures. *Source:* Professor G. Grinstein, Department of Computer Science, University of Massachusetts at Lowell.

2.3.4 Hierarchical Visualization Techniques

The visualization techniques discussed so far focus on visualizing multiple dimensions simultaneously. However, for a large data set of high dimensionality, it would be difficult to visualize all dimensions at the same time. **Hierarchical visualization techniques** partition all dimensions into subsets (i.e., subspaces). The subspaces are visualized in a hierarchical manner.

"Worlds-within-Worlds," also known as *n*-Vision, is a representative hierarchical visualization method. Suppose we want to visualize a 6-D data set, where the dimensions are $F, X_1, \ldots, X_5$. We want to observe how dimension F changes with respect to the other dimensions. We can first fix the values of dimensions X_3, X_4, X_5 to some selected values, say, c_3, c_4, c_5. We can then visualize F, X_1, X_2 using a 3-D plot, called a *world*, as shown in Figure 2.19. The position of the origin of the inner world is located at the point (c_3, c_4, c_5) in the outer world, which is another 3-D plot using dimensions X_3, X_4, X_5. A user can interactively change, in the outer world, the location of the origin of the inner world. The user then views the resulting changes of the inner world. Moreover, a user can vary the dimensions used in the inner world and the outer world. Given more dimensions, more levels of worlds can be used, which is why the method is called "worlds-within-worlds."

As another example of hierarchical visualization methods, **tree-maps** display hierarchical data as a set of nested rectangles. For example, Figure 2.20 shows a tree-map visualizing Google news stories. All news stories are organized into seven categories, each shown in a large rectangle of a unique color. Within each category (i.e., each rectangle at the top level), the news stories are further partitioned into smaller subcategories.

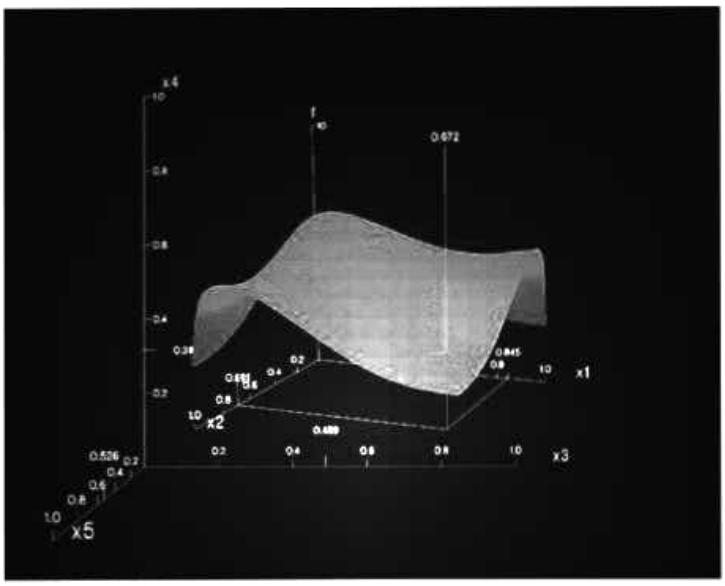


Figure 2.19 "Worlds-within-Worlds" (also known as *n*-Vision). *Source: http://graphics.cs.columbia.edu/projects/AutoVisual/images/1.dipstick.5.gif.*

2.3.5 Visualizing Complex Data and Relations

In early days, visualization techniques were mainly for numeric data. Recently, more and more non-numeric data, such as text and social networks, have become available. Visualizing and analyzing such data attracts a lot of interest.

There are many new visualization techniques dedicated to these kinds of data. For example, many people on the Web tag various objects such as pictures, blog entries, and product reviews. A **tag cloud** is a visualization of statistics of user-generated tags. Often, in a tag cloud, tags are listed alphabetically or in a user-preferred order. The importance of a tag is indicated by font size or color. Figure 2.21 shows a tag cloud for visualizing the popular tags used in a Web site.

Tag clouds are often used in two ways. First, in a tag cloud for a single item, we can use the size of a tag to represent the number of times that the tag is applied to this item by different users. Second, when visualizing the tag statistics on multiple items, we can use the size of a tag to represent the number of items that the tag has been applied to, that is, the popularity of the tag.

In addition to complex data, complex relations among data entries also raise challenges for visualization. For example, Figure 2.22 uses a disease influence graph to visualize the correlations between diseases. The nodes in the graph are diseases, and the size of each node is proportional to the prevalence of the corresponding disease. Two nodes are linked by an edge if the corresponding diseases have a strong correlation. The width of an edge is proportional to the strength of the correlation pattern of the two corresponding diseases.

Figure 2.20 Newsmap: Use of tree-maps to visualize Google news headline stories. *Source: www.cs.umd.edu/class/spring2005/cmsc838s/viz4all/ss/newsmap.png.*

In summary, visualization provides effective tools to explore data. We have introduced several popular methods and the essential ideas behind them. There are many existing tools and methods. Moreover, visualization can be used in data mining in various aspects. In addition to visualizing data, visualization can be used to represent the data mining process, the patterns obtained from a mining method, and user interaction with the data. Visual data mining is an important research and development direction.

2.4 Measuring Data Similarity and Dissimilarity

In data mining applications, such as clustering, outlier analysis, and nearest-neighbor classification, we need ways to assess how alike or unalike objects are in comparison to one another. For example, a store may want to search for clusters of *customer* objects, resulting in groups of customers with similar characteristics (e.g., similar income, area of residence, and age). Such information can then be used for marketing. A **cluster** is