Table of Contents

Chapter 1
Visual Data Mining Techniques

Daniel Keim and Matthew Ward
University of Konstanz, Germany and Worcester Polytechnic Institute, USA

Abstract. Never before in history has data been generated at such high volumes as it is today. Exploring and analyzing the vast volumes of data has become increasingly difficult. Information visualization and visual data mining can help to deal with the flood of information. The advantage of visual data exploration is that the user is directly involved in the data mining process. There are a large number of information visualization techniques that have been developed over the last two decades to support the exploration of large data sets. In this paper, we propose a classification of information visualization and visual data mining techniques based on the *data type to be visualized*, the *visualization technique*, and the *interaction technique*. We illustrate the classification using a few examples, and indicate some directions for future work.

1.1. Introduction

The progress made in hardware technology allows today's computer systems to store very large amounts of data. Researchers from the University of Berkeley estimate that every year about 1 Exabyte (= 1 Million Terabytes) of data are generated, of which a large portion is available in digital form. This means that in the next three years more data will be generated than in all of human history to date. The data is often automatically recorded via sensors and monitoring systems. Even simple transactions of every day life, such as paying by credit card or using the telephone, are typically recorded by computers. Usually many parameters are recorded, resulting in data with a high dimensionality. The data is collected because people believe that it is a potential source of valuable information, providing a competitive advantage (at some point). Finding the valuable information hidden in the data, however, is a difficult task. With today's data management systems, it is only possible to view quite small portions of the data.

If the data is presented textually, the amount of data that can be displayed is in the range of some one hundred data items, but this is like a drop in the ocean when dealing with data sets containing millions of data items. Having no possibility to adequately explore the large amounts of data that have been collected because of their potential usefulness, the data becomes useless and the databases become data 'dumps'.

Benefits of Visual Data Exploration

For data mining to be effective, it is important to include the human in the data exploration process and combine the flexibility, creativity, and general knowledge of the human with the enormous storage capacity and the computational power of today's computers. Visual data exploration aims at integrating the human in the data exploration process, applying human perceptual abilities to the analysis of large data sets available in today's computer systems. The basic idea of visual data exploration is to present the data in some visual form, allowing the user to gain insight into the data, draw conclusions, and directly interact with the data. Visual data mining techniques have proven to be of high value in exploratory data analysis, and have a high potential for exploring large databases. Visual data exploration is especially useful when little is known about the data and the exploration goals are vague. Since the user is directly involved in the exploration process, shifting and adjusting the exploration goals is automatically done if necessary.

Visual data exploration can be seen as a hypothesis generation process; the visualizations of the data allow the user to gain insight into the data and come up with new hypotheses. The verification of the hypotheses can also be done via data visualization, but may also be accomplished by automatic techniques from statistics, pattern recognition, or machine learning. In addition to the direct involvement of the user, the main advantages of visual data exploration over automatic data mining techniques are:

- visual data exploration can easily deal with highly non-homogeneous and noisy data
- visual data exploration is intuitive and requires no understanding of complex mathematical or statistical algorithms or parameters.
- visualization can provide a qualitative overview of the data, allowing data phenomena to be isolated for further quantitative analysis.

As a result, visual data exploration usually allows a faster data exploration and often provides better results, especially in cases where automatic algorithms fail. In addition, visual data exploration techniques provide a much higher degree of confidence in the findings of the exploration. This fact leads to a high demand for visual exploration techniques and makes them indispensable in conjunction with automatic exploration techniques.

Visual Exploration Paradigm

Visual Data Exploration usually follows a three step process: *Overview first, zoom and filter, and then details-on-demand* (which has been called the Information Seeking Mantra [68]). First, the user needs to get an overview of the data. In the overview, the user identifies interesting patterns or groups in the data and focuses on one or more of them. For analyzing the patterns, the user needs to drill-down and access details of the data. Visualization technology may be used for all three steps of the data exploration process. Visualization techniques are useful for showing an overview of the data, allowing the user to identify interesting subsets. In this step, it is important to keep the overview visualization while focusing on the subset using another visualization technique. An alternative is to distort the overview visualization in order to focus on the interesting subsets. This can be performed by dedicating a larger percentage of the display to the interesting subsets while decreasing screen utilization for uninteresting data. To further explore the interesting subsets, the user needs a drill-down capability in order to observe the details about the data. Note that visualization technology does not only provide the base visualization techniques for all three steps but also bridges the gaps between the steps.

1.2. Classification of Visual Data Mining Techniques

Information visualization focuses on data sets lacking inherent 2D or 3D semantics and therefore also lacking a standard mapping of the abstract data onto the physical screen space. There are a number of well known techniques for visualizing such data sets, such as x-y plots, line plots, and histograms. These techniques are useful for data exploration but are limited to relatively small and low dimensional data sets. In the last decade, a large number of novel information visualization techniques have been developed, allowing visualizations of multidimensional data sets without inherent two- or three-dimensional semantics. Nice overviews of the approaches can be found in a number of recent books [21] [83] [69] [64]. The techniques can be classified based on three criteria (see figure 1.1) [45]: The data to be visualized, the visualization technique, and the interaction technique used.

The **data type to be visualized** [68] may be

- One-dimensional data, such as temporal (time-series) data
- Two-dimensional data, such as geographical maps
- Multidimensional data, such as relational tables
- Text and hypertext, such as news articles and Web documents
- Hierarchies and graphs, such as telephone calls and Web documents
- Algorithms and software, such as debugging operations

The **visualization technique** used may be classified as:

- Standard 2D/3D displays, such as bar charts and x-y plots

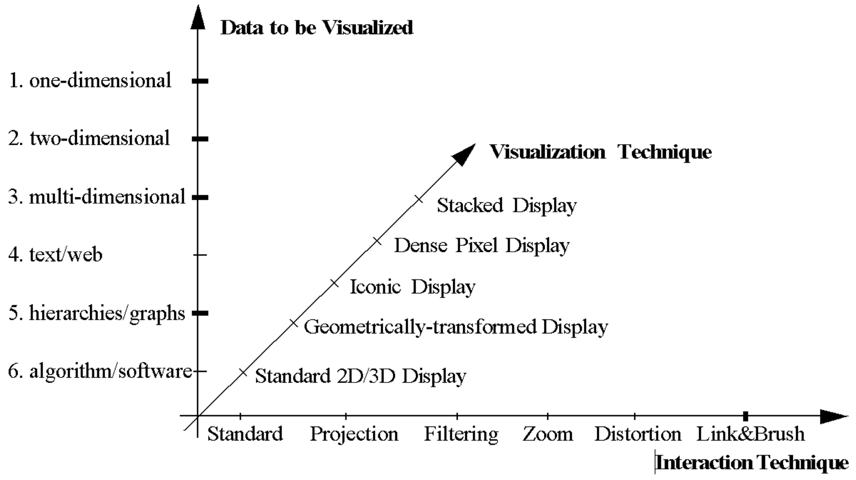


Fig. 1.1. Classification of Information Visualization Techniques

- Geometrically transformed displays, such as landscapes and parallel coordinates
- Icon-based displays, such as needle icons and star icons
- Dense pixel displays, such as the recursive pattern and circle segments
- Stacked displays, such as treemaps and dimensional stacking

The third dimension of the classification is the **interaction technique** used. Interaction techniques allow users to directly navigate and modify the visualizations, as well as select subsets of the data for further operations. Examples include:

- Dynamic Projection
- Interactive Filtering
- Interactive Zooming
- Interactive Distortion
- Interactive Linking and Brushing

Note that the three dimensions of our classification - data type to be visualized, visualization technique, and interaction technique - can be assumed to be orthogonal. Orthogonality means that any of the visualization techniques may be used in conjunction with any of the interaction techniques for any data type. Note also that a specific system may be designed to support different data types and that it may use a combination of visualization and interaction techniques.

1.3. Data Type to be Visualized

In information visualization, the data usually consists of a large number of records, each consisting of a number of variables or dimensions. Each record corresponds to an observation, measurement, or transaction. Examples are customer properties, e-commerce transactions, and sensor output from physical experiments. The number of attributes can differ from data set to data set; one particular physical experiment, for example, can be described by five variables, while another may need hundreds of variables. We call the number of variables the dimensionality of the data set. Data sets may be one-dimensional, two-dimensional, multidimensional or may have more complex data types such as text/hypertext or hierarchies/graphs. Sometimes, a distinction is made between dense (or grid) dimensions and the dimensions that may have arbitrary values. Depending on the number of dimensions with arbitrary values the data is sometimes also called univariate, bivariate, or multivariate.

One-dimensional data

One-dimensional data usually has one dense dimension. A typical example of one-dimensional data is temporal data. Note that with each point of time, one or multiple data values may be associated. An example are time series of stock prices (see figures 1.4 and 1.6 for examples) or the time series of news data used in the ThemeRiver examples (see figures 2-5 in [35]).

Two-dimensional data

Two-dimensional data has two distinct dimensions. A typical example is geographical data, where the two distinct dimensions are longitude and latitude. X-Y-plots are a typical method for showing two-dimensional data and maps are a special type of x-y-plot for showing two-dimensional geographical data. Examples are the geographical maps used in Polaris (see figure 3(c) in [74]) and in MGV (see figure 9 in [1]). Although it seems easy to deal with temporal or geographic data, caution is advised. If the number of records to be visualized is large, temporal axes and maps get quickly cluttered - and may not help to understand the data.

Multi-dimensional data

Many data sets consist of more than three attributes and therefore do not allow a simple visualization as 2-dimensional or 3-dimensional plots. Examples of multidimensional (or multivariate) data are tables from relational databases, which often have tens to hundreds of columns (or attributes). Since there is no simple mapping of the attributes to the two dimensions of the screen, more sophisticated visualization techniques are needed. An example of a technique that allows the visualization of multidimensional data is the Parallel Coordinates Technique [42] (see figure 1.3, which is also used in the Scalable Framework (see figure 12 in [52]) and XmdvTool [82]. Parallel Coordinates display each multi-dimensional data item as a set of line segments that intersect each of the parallel axes at the position corresponding to the data value for the corresponding dimension.

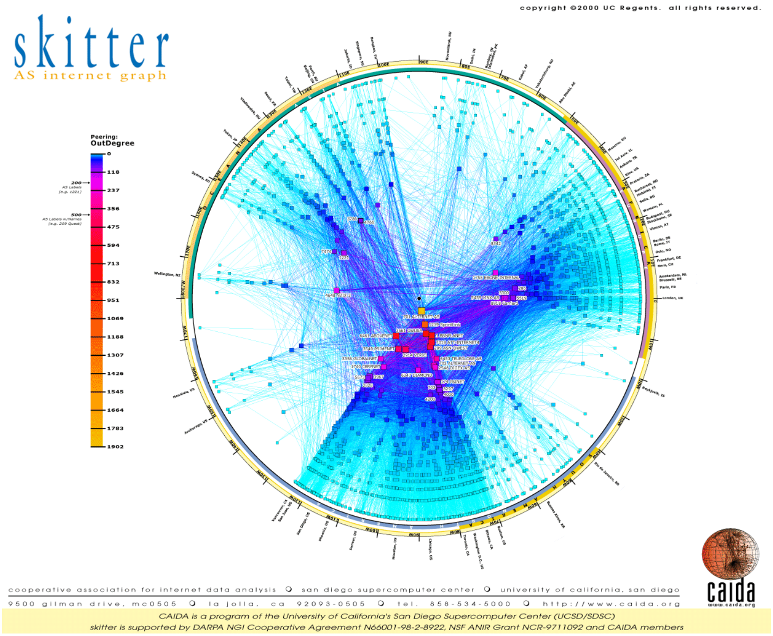

copyright ©2000 UC Regents. all rights reserved.
skitter
AS internet graph
Peering:
OutDegree
0
118
200 AS Labels (e.g. 1221)
237
356
500 AS Labels w/Names (e.g. 209 Quest)
475
594
713
832
951
1069
1188
1307
1426
1545
1664
1783
1902
Wellington, NZ
Vancouver, CA
San Jose, US
San Diego, US
Phoenix, US
Denver, US
Houston, US
Chicago, US
Toronto, CA
Washington D.C., US
Ottawa, CA
Boston, US
Buenos Aires, AR
cooperative association for internet data analysis ○ san diego supercomputer center ○ university of california, san diego
9500 gilman drive, mc0505 ○ la jolla, ca 92093-0505 ○ tel. 858-534-5000 ○ http://www.caida.org
CAIDA is a program of the University of California's San Diego Supercomputer Center (UCSD/SDSC)
skitter is supported by DARPA NGI Cooperative Agreement N66001-98-2-8922, NSF ANIR Grant NCR-9711092 and CAIDA members
caida
www.caida.org

Fig. 1.2. Skitter Graph Internet Map, CAIDA (Cooperative Association for Internet Data Analysis) ©2000 UC Regents. Courtesy University of California.

Text & Hypertext

Not all data types can be described in terms of dimensionality. In the age of the World Wide Web, one important data type is text and hypertext, as well as multimedia web page contents. These data types differ in that they cannot be easily described by numbers, and therefore most of the standard visualization techniques cannot be applied. In most cases, a transformation of the data into description vectors is necessary before visualization techniques can be used. An example for a simple transformation is word counting (see ThemeRiver [35]) which is often combined with a principal component analysis or multidimensional scaling to reduce the dimensionality to two or three (for example, see [85]).

Hierarchies & Graphs

Data records often have some relationship to other pieces of information. These relationships may be ordered, hierarchical, or arbitrary networks of relations. Graphs are widely used to represent such interdependencies. A graph consists of a set of objects, called nodes, and connections between these objects, called edges or links. Examples are the e-mail interrelationships among people, their shopping behavior, the file structure of the hard disk or the hyperlinks in the world wide web. There are a number of specific visualization techniques that deal with hierarchical and graphical data. A nice overview of hierarchical information

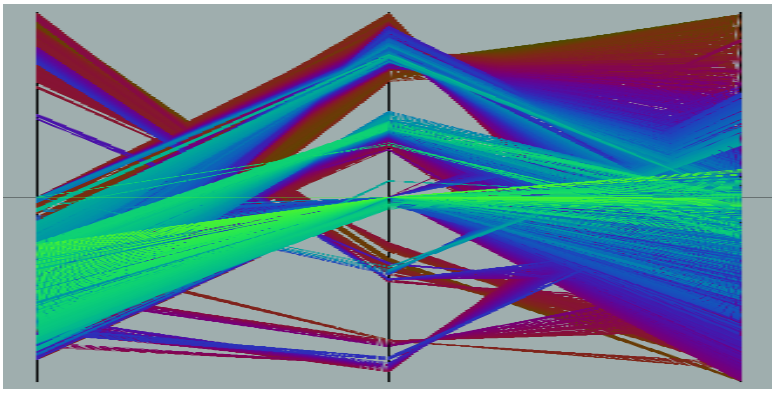


Fig. 1.3. Parallel Coordinate Visualization ©IEEE

visualization techniques can be found in [24], an overview of web visualization techniques is presented in [27] and an overview book on all aspects related to graph drawing is [12].

Algorithms & Software

Another class of data are algorithms and software. Coping with large software projects is a challenge. The goal of software visualization is to support software development by helping to understand algorithms (e.g., by showing the flow of information in a program), to enhance the understanding of written code (e.g., by representing the structure of thousands of source code lines as graphs), and to support the programmer in debugging the code (e.g., by visualizing errors). There are a large number of tools and systems that support these tasks. Nice overviews of software visualization can be found in [77] and [73].

1.4. Visualization Techniques

There are a large number of visualization techniques that can be used for visualizing data. In addition to standard 2D/3D-techniques such as x-y (x-y-z) plots, bar charts, line graphs, and maps, there are a number of more sophisticated classes of visualization techniques. The classes correspond to basic visualization principles that may be combined in order to implement a specific visualization system.

1.4.1 Geometrically-Transformed Displays

Geometrically transformed display techniques aim at finding “interesting” transformations of multidimensional data sets. The class of geometric display methods includes techniques from exploratory statistics such as scatterplot matrices [6]

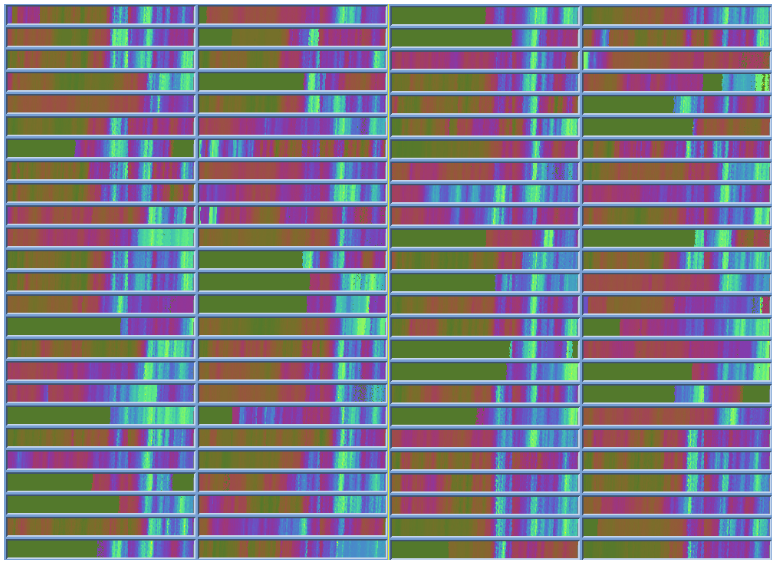


Fig. 1.4. Dense Pixel Displays: Recursive Pattern Technique ©IEEE

[26] and techniques that can be subsumed under the term "projection pursuit" [39]. Other geometric projection techniques include Prosection Views [32] [71], Hyperslice [80], and the well-known Parallel Coordinates visualization technique [42]. The *parallel coordinate technique* maps the k-dimensional space onto the two display dimensions by using k axes that are parallel to each other (either horizontally or vertically oriented), evenly spaced across the display. The axes correspond to the dimensions and are linearly scaled from the minimum to the maximum value of the corresponding dimension. Each data item is presented as a chain of connected line segments, intersecting each of the axes at a location corresponding to the value of the considered dimensions (see figure 1.3).

1.4.2 Iconic Displays

Another class of visual data exploration techniques are the iconic display methods. The idea is to map the attribute values of a multi-dimensional data item to the features of an icon. Icons can be arbitrarily defined; they may be little faces [25], needle icons as used in MGV (see figure 5 in [1]), star icons [82], stick figure icons [57], color icons [51, 46], or TileBars [36], for example. The visualization is generated by mapping the attribute values of each data record to the features of the icons. In case of the stick figure technique, for example, two dimensions are mapped to the display dimensions and the remaining dimensions are mapped to the angles and/or limb length of the stick figure icon. If the data items are relatively dense with respect to the two display dimensions, the resulting visualization presents texture patterns that vary according to the characteristics of the data and are therefore detectable by pre-attentive perception. Figure 1.5

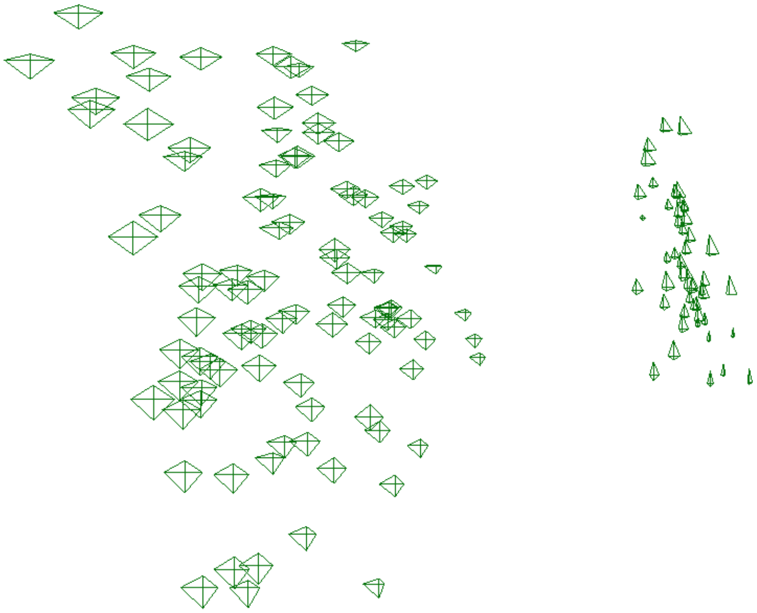


Fig. 1.5. The iris data set, displayed using star glyphs positioned based on the first two principal components (from XmdvTool [82]);

shows an example of this class of techniques. Each data point is represented by a star icon/glyph, where each data dimension controls the length of a ray emanating from the center of the icon. In this example, the positions of the icons are determined using principal component analysis (PCA) to convey more information about data relations. Other data attributes could also be mapped to icon position.

1.4.3 Dense Pixel Displays

The basic idea of dense pixel techniques is to map each dimension value to a colored pixel and group the pixels belonging to each dimension into adjacent areas [44]. Since in general dense pixel displays use one pixel per data value, the techniques allow the visualization of the largest amount of data possible on current displays (up to about 1,000,000 data values). If each data value is represented by one pixel, the main question is how to arrange the pixels on the screen. Dense pixel techniques use different arrangements for different purposes. By arranging the pixels in an appropriate way, the resulting visualization provides detailed information on local correlations, dependencies, and hot spots.

Well known examples are the recursive pattern technique [47] and the circle segments technique [9]. The *recursive pattern technique* is based on a generic recursive back-and-forth arrangement of the pixels and is particularly aimed at representing datasets with a natural order according to one attribute (e.g.

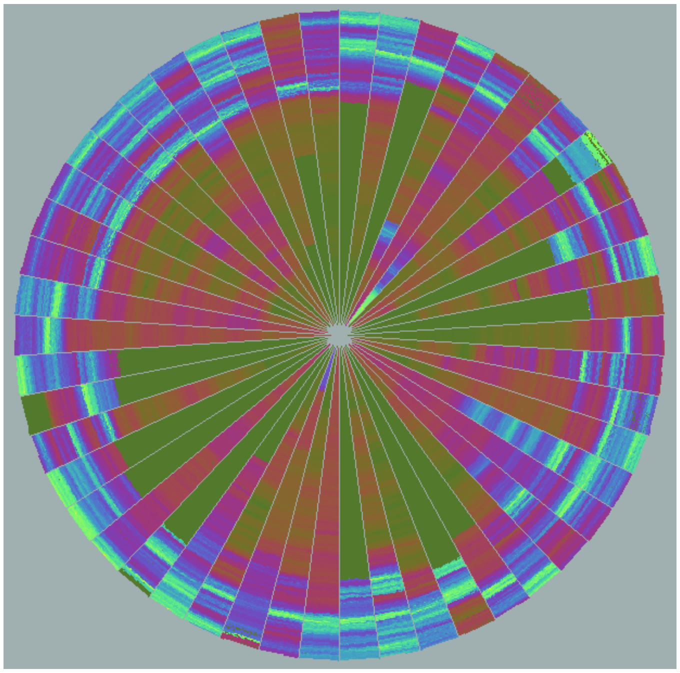


Fig. 1.6. Dense Pixel Displays: Circle Segments Technique ©IEEE

time-series data). The user may specify parameters for each recursion level, and thereby control the arrangement of the pixels to form semantically meaningful substructures. The base element on each recursion level is a pattern of height h_i and width w_i as specified by the user. First, the elements correspond to single pixels that are arranged within a rectangle of height h_1 and width w_1 from left to right, then below backwards from right to left, then again forward from left to right, and so on. The same basic arrangement is done on all recursion levels with the only difference being that the basic elements that are arranged on level i are the pattern resulting from the level $(i-1)$ arrangements. In Figure 1.4, an example recursive pattern visualization of financial data is shown. The visualization shows twenty years (January 1974 - April 1995) of daily prices of the 100 stocks contained in the Frankfurt Stock Index (FAZ). The idea of the *circle segments technique* [9] is to represent the data in a circle that is divided into segments, one for each attribute. Within the segments each attribute value is again visualized by a single colored pixel. The arrangement of the pixels starts at the center of the circle and continues to the outside by plotting on a line orthogonal to the segment halving line in a back and forth manner. The rationale of this approach is that close to the center all attributes are close to each other enhancing the visual comparison of their values. Figure 1.6 shows an example of circle segment visualization using the same data (50 stocks) as shown in figure 1.4.

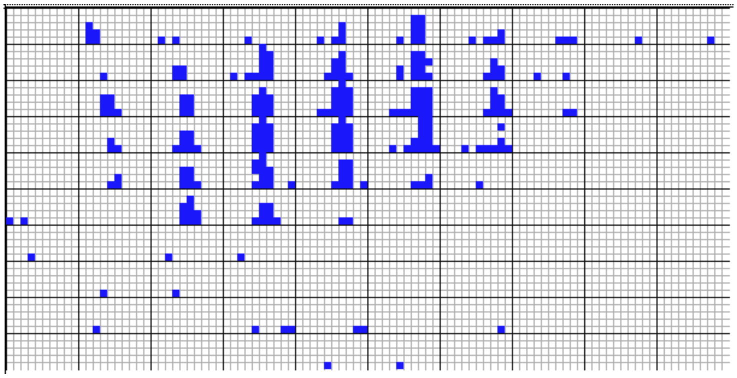


Fig. 1.7. Dimensional Stacking visualization of drill hole mining data *(used by permission of M. Ward, Worcester Polytechnic Institute ©IEEE)*

1.4.4 Stacked Displays

Stacked display techniques are tailored to present data partitioned in a hierarchical fashion. In the case of multi-dimensional data, the data dimensions to be used for partitioning the data and building the hierarchy have to be selected appropriately. An example of a stacked display technique is *Dimensional Stacking* [49]. The basic idea is to embed one coordinate system inside another coordinate system, i.e. two attributes form the outer coordinate system, two other attributes are embedded into the outer coordinate system, and so on. The display is generated by dividing the outermost level coordinate system into rectangular cells and within the cells the next two attributes are used to span the second level coordinate system. This process may be repeated multiple times. The usefulness of the resulting visualization largely depends on the data distribution of the outer coordinates and therefore the dimensions that are used for defining the outer coordinate system have to be selected carefully. A rule of thumb is to choose the most important dimensions first. A dimensional stacking visualization of mining data with longitude and latitude mapped to the outer x and y axes, as well as ore grade and depth mapped to the inner x and y axes is shown in figure 1.7. Other examples of stacked display techniques include Worlds-within-Worlds [29], Treemap [67] [43], and Cone Trees [61].

1.5. Specific Visual Data Mining Techniques

There are a number of visualization techniques that have been developed to support specific data mining tasks, such as association rule generation, classification, and clustering. In the following we describe how visualization techniques can be used to support these tasks.

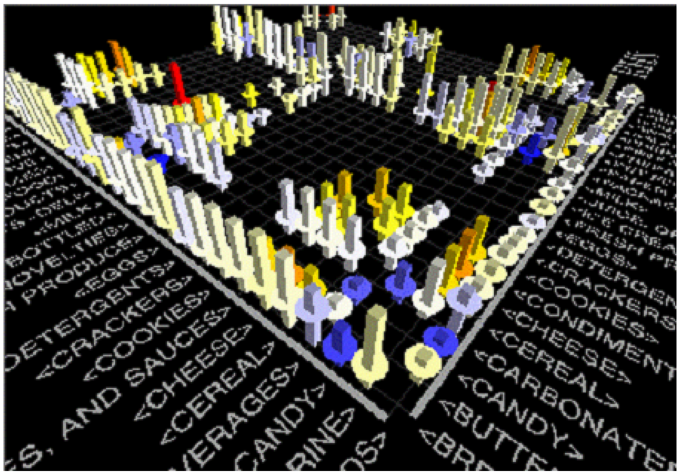


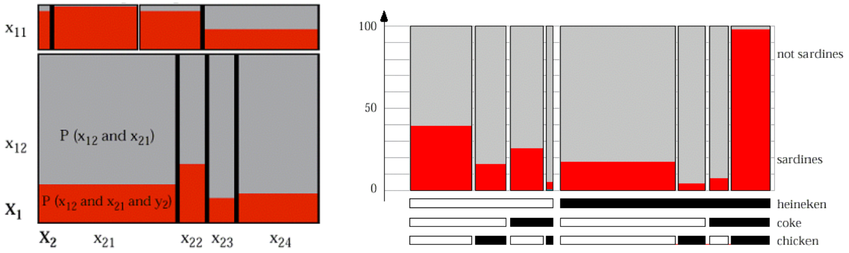
Fig. 1.8. MineSets Association Rule Visualizer [41] ©SGI

1.5.1 Association Rules

The goal of association rule generation is to find interesting patterns and trends in transaction databases. Association rules are statistical relations between two or more items in the dataset. In a supermarket basket application, associations express the relations between items that are bought together. It is for example interesting if we find out that in 70% of the cases when people buy bread, they also buy milk. Association rules tell us that the presence of some items in a transaction imply the presence of other items in the same transaction with a certain probability, called confidence. A second important parameter is the support of an association rule, which is defined as the percentage of transactions in which the items co-occur.

Let $I = \{i_1, ...i_n\}$ be a set of items and let D be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \in I$, $Y \in I$, $X, Y \neq \emptyset$. The confidence c is defined as the percentage of transactions that contain Y, given X. The support is the percentage of transactions that contain both X and Y. For a given support and confidence level, there are efficient algorithms to determine all association rules [2]. A problem however is that the resulting set of association rules is usually very large, especially for low support and confidence levels. Using higher support and confidence levels may not be effective, since useful rules may then be overlooked.

Visualization techniques have been used to overcome this problem and to allow an interactive selection of good support and confidence levels. Figure 1.8 shows SGI MineSets *Rule Visualizer* [41] which maps the left and right hand sides of the rules to the x- and y-axes of the plot and shows the confidence as the height of the bars and the support as the height of the discs. The color of the bars shows the interestingness of the rule. Using the visualization, the



(a) Mosaic Plot (b) Double Decker Plot

Fig. 1.9. Association Rule Visualization [38] ©ACM

user is able to see groups of related rules and the impact of different confidence and support levels. The number of rules that can be visualized, however, is limited and the visualization does not support combinations of items on the left or right hand side of the association rules. Figure 1.9 shows two alternative visualizations called mosaic and double decker plots [38]. The basic idea is to partition a rectangle on the y-axis according to one attribute and make the regions proportional to the sum of the corresponding data values. Compared to bar charts, mosaic plots use the height of the bars instead of the width to show the parameter value. Then each resulting area is split in the same way according to a second attribute. The coloring reflects the percentage of data items that fulfill a third attribute. The visualization shows the support and confidence values of all rules of the form $X_1, X_2 \Rightarrow Y$. Mosaic plots are restricted to two attributes on the left side of the association rule. Double decker plots can be used to show more than two attributes on the left side. The idea is to show a hierarchy of attributes on the bottom (heineken, coke, chicken in the example shown in figure 1.9) corresponding to the left hand side of the association rules and the bars on the top correspond to the number of items in the corresponding subset of the database and therefore visualize the support of the rule. The colored areas in the bars correspond to the percentage of data transactions that contain an additional item (sardines in figure 1.9) and therefore correspond to the support. Other approaches to association rule visualization include graphs with nodes corresponding to items and arrows corresponding to implications as used in DBMiner [40] and association matrix visualizations to cluster related rules [34].

1.5.2 Classification

Classification is the process of developing a classification model based on a training data set with known class labels. To construct the classification model, the attributes of the training data set are analyzed and an accurate description or model of the classes based on the attributes available in the data set is developed. The class descriptions are used then to classify data for which the class labels

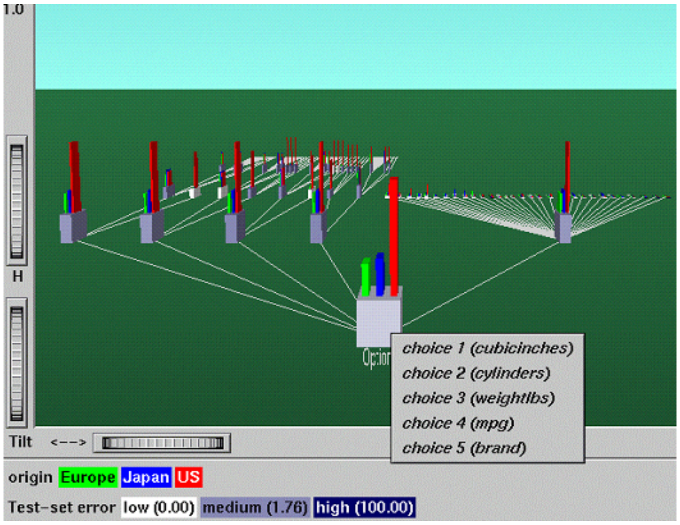


Fig. 1.10. MineSets Decision Tree Visualizer [41] ©SGI

are unknown. Classification is sometimes also called *supervised learning* because the training set is used to teach the system how to classify the data. There are a large number of algorithms for solving classification talks. The most popular approaches are algorithms that inductively construct decision trees. Examples are ID3 [58], CART [19], ID5 [78, 79], C4.5 [59], SLIQ [54], and SPRINT [66]. In addition there are approaches that use neural networks, genetic algorithms or Bayesian networks to solve the classification problem. Since most algorithms work as black box approaches it is often difficult to understand and optimize the decision model. Problems such as overfitting or tree pruning are difficult to tackle.

Visualization techniques can help to overcome these problems. The decision tree visualizer in SGIs MineSet system [41] shows an overview of the decision tree together with important parameters such as the attribute value distributions. The system allows an interactive selection of the attributes shown and helps the user understand the decision tree. A more sophisticated approach which also helps in decision tree construction is visual classification as proposed in [8]. The basic idea is to show each attribute value by a colored pixel and arrange them in bars. The pixels of each attribute bar are sorted separately and the attribute with the purest value distribution is selected as the split attribute of the decision tree. The procedure is repeated until all leaves correspond to pure classes. An example of the decision tree resulting from this process is shown in figure 1.11. Compared to a standard visualization of a decision tree, additional information is provided that is helpful for explaining and analyzing the decision tree, namely

- size of the nodes (number of training records corresponding to the node)
- quality of the split (purity of the resulting partitions)

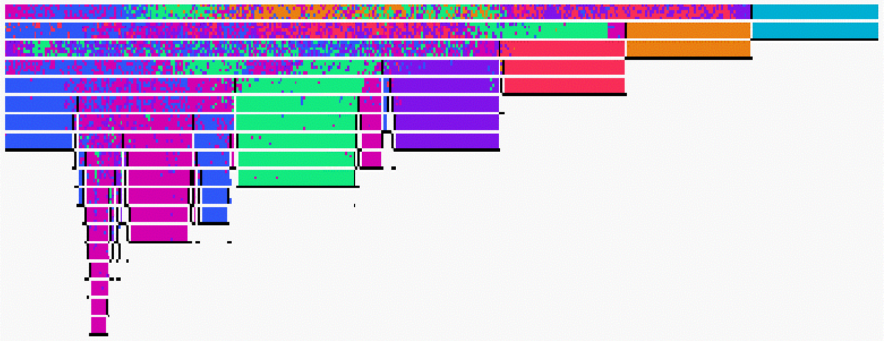


Fig. 1.11. Visualization of a decision trees [8] for the segment training data from the Statlog benchmark having 19 attributes ©ACM

- class distribution (frequency and location of the training instances of all classes).

Some of this information might also be provided by annotating the standard visualization of a decision tree (for example, annotating the nodes with the number of records or the gini-index), but this approach clearly fails for more complex information such as the class distribution. In general, visualizations can help to better understand the classification models and to easily interact with the classification algorithms in order to optimize the model generation and classification process.

1.5.3 Clustering

Clustering is the process of finding a partitioning of the data set into homogeneous subsets called clusters. Unlike classification, clustering is *unsupervised learning*. This means that the classes are unknown and no training set with class labels is available. A wide range of clustering algorithms have been proposed in the literature including density-based methods such as KDE [65] and linkage-based methods [18]. Most algorithms use assumptions about the properties of the clusters that are either used as defaults or have to be given as input parameters. Depending on the parameter values, the user gets differing clustering results. In two- or three-dimensional space, the impact of different algorithms and parameter settings can easily be explored using simple visualizations of the resulting clusters (for example, x-y plots) but in higher dimensional space the impact is much more difficult to understand. Some higher-dimensional techniques try to determine two- or three-dimensional projections of the data that retain the properties of the high-dimensional clusters as much as possible [86]. Figure 1.12 shows a three-dimensional projection of a data set consisting of five clusters.

While this approach works well with low- to medium-dimensional data sets, it is difficult to apply to large high-dimensional data sets, especially if the clusters are not clearly separated and the data set also contains noise (data that does not

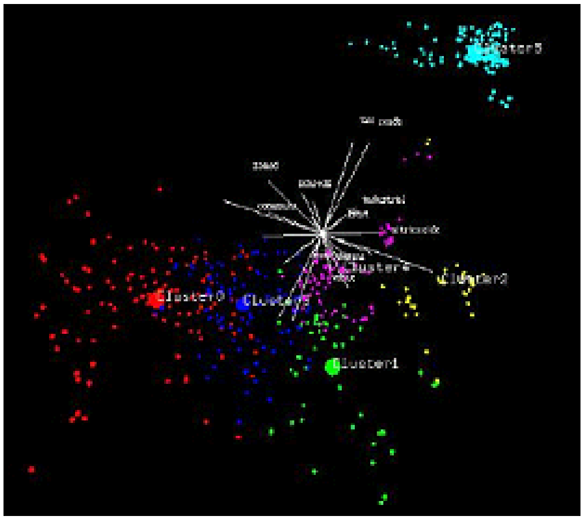
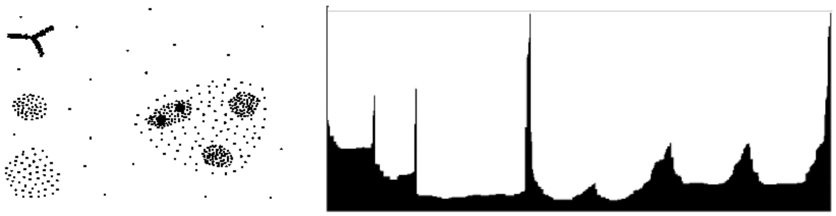


Fig. 1.12. Visualization based on a projection into 3D space [86] ©ACM

belong to any cluster). In this case, more sophisticated visualization techniques are needed to guide the clustering process, select the right clustering model, and adjust the parameter values appropriately. An example of a system that uses visualization techniques to help in high-dimensional clustering is OPTICS [7]. The idea of OPTICS (*Ordering Points To Identify the Clustering Structure*) is to create a one-dimensional ordering of the database representing its density-based clustering structure. Figure 1.13 shows a two-dimensional example data set together with its reachability distance plot. Intuitively, points within a cluster are close in the generated one-dimensional ordering and their reachability distance shown in figure 1.13 is similar. Jumping to an other cluster results in higher reachability distances. The idea works for data of arbitrary dimension. The reachability plot provides a visualization of the inherent clustering structure and is therefore valuable for understanding the clustering and guiding the clustering process.

Another interesting approach is the *HD-Eye* system [37]. The *HD-Eye* system considers the clustering problem as a partitioning problem and supports a tight integration of advanced clustering algorithms and state-of-the-art visualization techniques, allowing the user to directly interact in the crucial steps of the clustering process. The crucial steps are the selection of dimensions to be considered, the selection of the clustering paradigm, and the partitioning of the data set. Novel visualization techniques are employed to help the user identify the most interesting projections and subsets as well as the best separators for partitioning the data. Figure 1.14 shows an example screen shot of the *HD-Eye* system with its basic visual components for cluster separation. The separator tree represents the clustering model produced so far in the clustering process. The *abstract iconic displays* (top right and bottom middle in figure 1.14) visu-



(a) Data Set (b) Reachability Plot

Fig. 1.13. OPTICS Visual Clustering [7] ©ACM

alize the partitioning potential of a large number of projections. The properties are based on histogram information of the point density in the projected space. The number of data points belonging to the maxima corresponds to the color of the icon. The color follows a given color table ranging from dark colors for large maxima to bright colors for small maxima. The measure of how well a maxima is separated from the others corresponds to the shape of the icon and the degree of separation varies from sharp spikes for well-separated maxima to blunt spikes for weak-separated maxima. The *color- and curve-based point density displays* present the density of the data and allow a better understanding of the data distribution, which is crucial for an effective partitioning of the data. The visualizations are used to decide which dimensions are used for the partitioning. In addition, the partitioning can be specified interactively directly within the visualizations, allowing the user to define non-linear partitionings.

1.6. Interaction Techniques

In addition to the visualization technique, for an effective data exploration it is necessary to use one or more interaction techniques. *Interaction techniques* allow the data analyst to directly interact with the visualizations and dynamically change the visualizations according to the exploration objectives. In addition, they also make it possible to relate and combine multiple independent visualizations.

Interaction techniques can be categorized based on the effects they have on the display. *Navigation techniques* focus on modifying the projection of the data onto the screen, using either manual or automated methods. *View enhancement methods* allow users to adjust the level of detail on part or all of the visualization, or modify the mapping to emphasize some subset of the data. *Selection techniques* provide users with the ability to isolate a subset of the displayed data for operations such as highlighting, filtering, and quantitative analysis. Selection can be done directly on the visualization (direct manipulation) or via dialog boxes or other query mechanisms (indirect manipulation). Some examples of interaction techniques are described below.

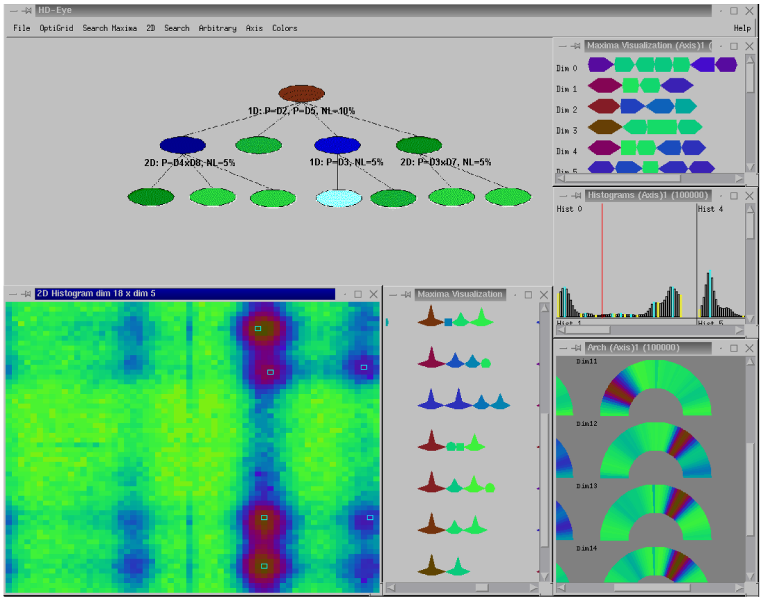


Fig. 1.14. *HD-Eye* screen-shot [37] showing different visualizations of projections and the separator tree. Clockwise from the top: separator tree, iconic representation of 1D projections, 1D projection histogram, 1D color-based density plots, iconic representation of multi dimensional projections and color-based 2D density plot. ©IEEE

Dynamic Projection

Dynamic projection is an automated navigation operation. The basic idea is to dynamically change the projections in order to explore a multi-dimensional data set. A classic example is the GrandTour system [11] which tries to show all interesting two-dimensional projections of a multi-dimensional data set as a series of scatterplots. Note that the number of possible projections is exponential in the number of dimensions, i.e. it is intractable for large dimensionality. The sequence of projections shown can be random, manual, precomputed, or data driven. Systems supporting dynamic projection techniques include XGobi [75] [20], XLispStat [76], and ExplorN [23].

Interactive Filtering

Interactive filtering is a combination of selection and view enhancement. In exploring large data sets, it is important to interactively partition the data set into segments and focus on interesting subsets. This can be done by a direct selection of the desired subset *(browsing)* or by a specification of properties of the desired subset *(querying)*. Browsing is very difficult for very large data sets and querying often does not produce the desired results. Therefore a number of interactive selection techniques have been developed to improve interactive filtering in data

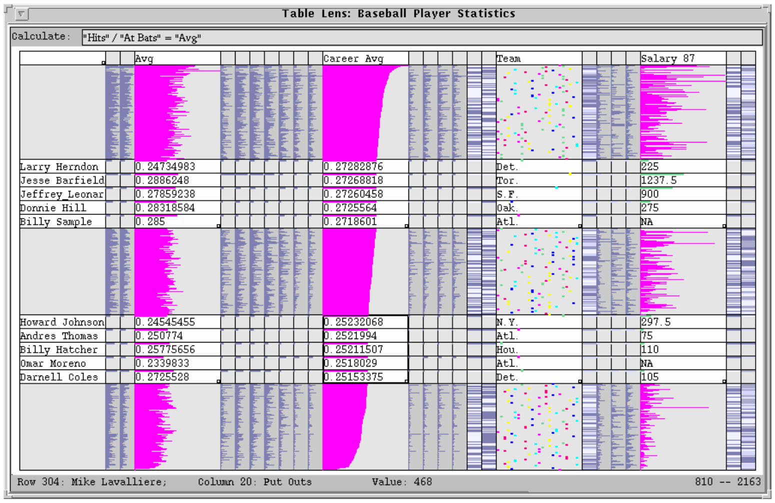


Table Lens: Baseball Player Statistics

Calculate: "Hits" / "At Bats" = "Avg"

	Avg	Career Avg	Team	Salary 87
Larry Herndon	0.24734983	0.27282876	Det.	225
Jesse Barfield	0.2886248	0.27268818	Tor.	1237.5
Jeffrey_Leonar	0.27859238	0.27260458	S.F.	900
Donnie Hill	0.28318584	0.2725564	Oak.	275
Billy Sample	0.285	0.2718601	Atl.	NA
Howard Johnson	0.24545455	0.25232068	N.Y.	297.5
Andres Thomas	0.250774	0.2521994	Atl.	75
Billy Hatcher	0.25775656	0.25211507	Hou.	110
Omar Moreno	0.2339833	0.2518029	Atl.	NA
Darnell Coles	0.2725528	0.25153375	Det.	105

Row 304: Mike Lavalliere; Column 20: Put Outs Value: 468 810 -- 2163

Fig. 1.15. Table Lens*(used by permission of R. Rao, Xerox PARC ©ACM)*

exploration. An example of a tool that can be used for interactive filtering is the Magic Lens [17] [30]. The basic idea of Magic Lens is to use a tool similar to a magnifying glass to support filtering the data directly in the visualization. The data under the magnifying glass is processed by the filter, and the result is displayed differently than the remaining data set. Magic Lens show a modified view of the selected region, while the rest of the visualization remains unaffected. Note that several lenses with different filters may be used; if the filter overlap, all filters are combined. Other examples of interactive filtering techniques and tools are InfoCrystal [72], Dynamic Queries [3] [28] [33], and Polaris [74] (see figure 6 in [74] for an example).

Zooming

Zooming is a well known view modification technique that is widely used in a number of applications. In dealing with large amounts of data, it is important to present the data in a highly compressed form to provide an overview of the data but at the same time allow a variable display of the data at different resolutions. Zooming does not only mean displaying the data objects larger, but also that the data representation may automatically change to present more details on higher zoom levels. The objects may, for example, be represented as single pixels at a low zoom level, as icons at an intermediate zoom level, and as labeled objects at a high resolution. An interesting example applying the zooming idea to large tabular data sets is the TableLens approach [60]. Getting an overview of large tabular data sets is difficult if the data is displayed in textual form. The basic idea of TableLens is to represent each numerical value by a small bar. All bars have a one-pixel height and the lengths are determined by the attribute values.

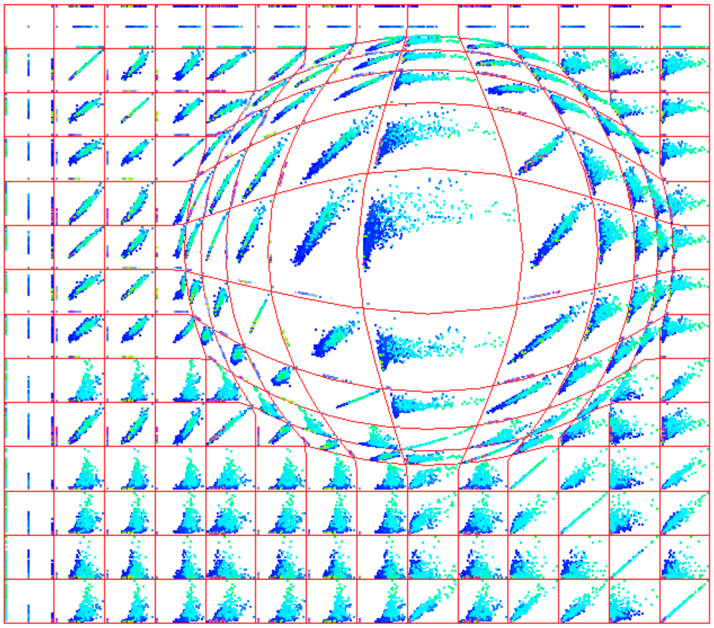
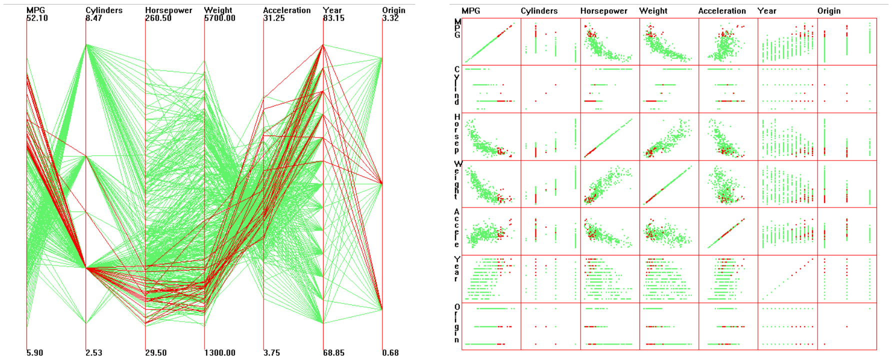


Fig. 1.16. A scatterplot matrix with part of the display distorted using a fisheye lens.

This means that the number of rows on the display can be nearly as large as the vertical resolution and the number of columns depends on the maximum width of the bars for each attribute. The initial view allows the user to detect patterns, correlations, and outliers in the data set. In order to explore a region of interest the user can zoom in, with the result that the affected rows (or columns) are displayed in more detail, possibly even in textual form. Figure 1.15 shows an example of a baseball database with a few rows being selected in full detail. Other examples of techniques and systems that use interactive zooming include PAD++ [56] [15] [16], IVEE/Spotfire [4], and DataSpace [10]. A comparison of fisheye and zooming techniques can be found in [63].

Distortion

Distortion is a view modification technique that supports the data exploration process by preserving an overview of the data during drill-down operations. The basic idea is to show portions of the data with a high level of detail while others are shown with a lower level of detail. Popular distortion techniques are hyperbolic and spherical distortions; these are often used on hierarchies or graphs but may be also applied to any other visualization technique. An example of spherical distortions is provided in the Scalable Framework paper (see figure 5 in [52]). An overview of distortion techniques is provided in [50] and [22]. Examples of distortion techniques include Bifocal Displays [70], Perspective Wall



(a) Parallel Coordinates

(b) Scatterplot Matrix

Fig. 1.17. Linked brushing between two multivariate visualization techniques, from XmdvTool [82]. Highlighted data (in red) from one display is also highlighted in other displays.

[53], Graphical Fisheye Views [31] [62], Hyperbolic Visualization [48] [55], and Hyperbox [5]. Figure 1.16 shows the effect of distorting part of a scatterplot matrix to display more detail from one of the plots while preserving context from the rest of the display.

Brushing and Linking

Brushing is an interactive selection process that is often, but not always, combined with *linking*, a process for communicating the selected data to other views of the data set. There are many possibilities to visualize multi-dimensional data, each with their own strengths and weaknesses. The idea of linking and brushing is to combine different visualization methods to overcome the shortcomings of individual techniques. Scatterplots of different projections, for example, may be combined by coloring and linking subsets of points in all projections. In a similar fashion, linking and brushing can be applied to visualizations generated by all visualization techniques described above. As a result, the brushed points are highlighted in all visualizations, making it possible to detect dependencies and correlations. Interactive changes made in one visualization are automatically reflected in the other visualizations. Note that connecting multiple visualizations through interactive linking and brushing provides more information than considering the component visualizations independently.

Typical examples of visualization techniques that have been combined by linking and brushing are multiple scatterplots, bar charts, parallel coordinates, pixel displays, and maps. Most interactive data exploration systems allow some form of linking and brushing. Examples are Polaris (see figure 7 in [74]) and the

Scalable Framework (see figures 12 and 14 in [52]). Other tools and systems include S Plus [13], XGobi [75] [14], XmdvTool [82] (see Figure 1.17, and DataDesk [81] [84].

1.7. Conclusion

The exploration of large data sets is an important but difficult problem. Information visualization techniques can be useful in solving this problem. Visual data exploration has a high potential, and many applications such as fraud detection and data mining can use information visualization technology for improved data analysis.

Avenues for future work include the tight integration of visualization techniques with traditional techniques from such disciplines as statistics, machine learning, operations research, and simulation. Integration of visualization techniques and these more established methods would combine fast automatic data mining algorithms with the intuitive power of the human mind, improving the quality and speed of the data mining process. Visual data mining techniques also need to be tightly integrated with the systems used to manage the vast amounts of relational and semistructured information, including database management and data warehouse systems. The ultimate goal is to bring the power of visualization technology to every desktop to allow a better, faster and more intuitive exploration of very large data resources. This will not only be valuable in an economic sense but will also stimulate and delight the user.

References

1. J. Abello and J. Korn. Mgv: A system for visualizing massive multi-digraphs. *Transactions on Visualization and Computer Graphics*, 2001.
2. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. Verkamo. Fast discovery of association rules. *Advances in Knowledge Discovery and Data Mining*, pages 307–328, 1996.
3. C. Ahlberg and B. Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *Proc. Human Factors in Computing Systems CHI '94 Conf., Boston, MA*, pages 313–317, 1994.
4. C. Ahlberg and E. Wistrand. Ivee: An information visualization and exploration environment. In *Proc. Int. Symp. on Information Visualization, Atlanta, GA*, pages 66–73, 1995.
5. B. Alpern and L. Carter. Hyperbox. In *Proc. Visualization '91, San Diego, CA*, pages 133–139, 1991.
6. D. F. Andrews. Plots of high-dimensional data. *Biometrics*, 29:125–136, 1972.
7. M. Ankerst, M. Breunig, H. Kriegel, and J.Sander. OPTICS: Ordering Points To Identify the Clustering Structure. *Proc. ACM SIGMOD'99, Int. Conf on Management of Data, Philadelphia, PA*, pages 49–60, 1999.
8. M. Ankerst, M. Ester, and H. Kriegel. Towards an effective cooperation of the computer and the user for classification. *SIGKDD Int. Conf. On Knowledge Discovery & Data Mining (KDD 2000), Boston, MA*, pages 179–188, 2000.
9. M. Ankerst, D. A. Keim, and H.-P. Kriegel. Circle segments: A technique for visually exploring large multidimensional data sets. In *Proc. Visualization 96, Hot Topic Session, San Francisco, CA*, 1996.
10. V. Anupam, S. Dar, T. Leibfried, and E. Petajan. Dataspace: 3D visualization of large databases. In *Proc. Int. Symp. on Information Visualization, Atlanta, GA*, pages 82–88, 1995.
11. D. Asimov. The grand tour: A tool for viewing multidimensional data. *SIAM Journal of Science & Stat. Comp.*, 6:128–143, 1985.
12. G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing.* Prentice Hall, 1999.
13. R. Becker, J. M. Chambers, and A. R. Wilks. *The New S Language.* Wadsworth & Brooks/Cole Advanced Books and Software, Pacific Grove, CA, 1988.
14. R. A. Becker, W. S. Cleveland, and M.-J. Shyu. The visual design and control of trellis display. *Journal of Computational and Graphical Statistics*, 5(2):123–155, 1996.
15. B. Bederson. Pad++: Advances in multiscale interfaces. In *Proc. Human Factors in Computing Systems CHI '94 Conf., Boston, MA*, page 315, 1994.

16. B. B. Bederson and J. D. Hollan. Pad++: A zooming graphical interface for exploring alternate interface physics. In *Proc. UIST*, pages 17–26, 1994.
17. E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. DeRose. Toolglass and magic lenses: The see-through interface. In *Proc. SIGGRAPH '93, Anaheim, CA*, pages 73–80, 1993.
18. H. H. Bock. *Automatic Classification.* Vandenhoeck and Ruprecht, Göttingen, 1974.
19. L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees.* Wadsworth and Brooks, Monterey, CA, 1984.
20. A. Buja, D. F. Swayne, and D. Cook. Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics*, 5(1):78–99, 1996.
21. S. Card, J. Mackinlay, and B. Shneiderman. *Readings in Information Visualization.* Morgan Kaufmann, 1999.
22. M. S. T. Carpendale, D. J. Cowperthwaite, and F. D. Fracchia. Ieee computer graphics and applications, special issue on information visualization. *IEEE Journal Press*, 17(4):42–51, July 1997.
23. D. B. Carr, E. J. Wegman, and Q. Luo. Explorn: Design considerations past and present. In *Technical Report, No. 129, Center for Computational Statistics, George Mason University*, 1996.
24. C. Chen. *Information Visualisation and Virtual Environments.* Springer-Verlag, London, 1999.
25. H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal Amer. Statistical Association*, 68:361–368, 1973.
26. W. S. Cleveland. *Visualizing Data.* AT&T Bell Laboratories, Murray Hill, NJ, Hobart Press, Summit NJ, 1993.
27. M. Dodge. Web visualization. http://www.geog.ucl.ac.uk/casa/martin/geography_of_cyberspace.html, Oct 2001.
28. S. G. Eick. Data visualization sliders. In *Proc. ACM UIST*, pages 119–120, 1994.
29. S. Feiner and C. Beshers. Visualizing n-dimensional virtual worlds with n-vision. *Computer Graphics*, 24(2):37–38, 1990.
30. K. Fishkin and M. C. Stone. Enhanced dynamic queries via movable filters. In *Proc. Human Factors in Computing Systems CHI '95 Conf., Denver, CO*, pages 415–420, 1995.
31. G. Furnas. Generalized fisheye views. In *Proc. Human Factors in Computing Systems CHI 86 Conf., Boston, MA*, pages 18–23, 1986.
32. G. W. Furnas and A. Buja. Prosections views: Dimensional inference through sections and projections. *Journal of Computational and Graphical Statistics*, 3(4):323–353, 1994.
33. J. Goldstein and S. F. Roth. Using aggregation and dynamic queries for exploring large data sets. In *Proc. Human Factors in Computing Systems CHI '94 Conf., Boston, MA*, pages 23–29, 1994.
34. M. Hao, M. Hsu, U. Dayal, S. F. Wei, T. Sprenger, and T. Holenstein. Market basket analysis visualization on a spherical surface. *Visual Data Exploration and Analysis Conference, San Jose, CA*, 2001.
35. S. Havre, B. Hetzler, L. Nowell, and P. Whitney. Themeriver: Visualizing thematic changes in large document collections. *Transactions on Visualization and Computer Graphics*, 2001.
36. M. Hearst. Tilebars: Visualization of term distribution information in full text information access. In *Proc. of ACM Human Factors in Computing Systems Conf. (CHI'95)*, pages 59–66, 1995.

37. A. Hinneburg, D. Keim, and M. Wawryniuk. HD-Eye: Visual Mining of High-Dimensional Data. *IEEE Computer Graphics and Applications*, 19(5), 1999.
38. H. Hofmann, A. Siebes, and A. Wilhelm. Visualizing association rules with interactive mosaic plots. *SIGKDD Int. Conf. On Knowledge Discovery & Data Mining (KDD 2000), Boston, MA*, 2000.
39. P. J. Huber. The annals of statistics. *Projection Pursuit*, 13(2):435–474, 1985.
40. D. T. Inc. Dbminer. *http://www.dbminer.com*, 2001.
41. S. G. Inc. Mineset. *http://www.sgi.com/software/mineset*, 2001.
42. A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multidimensional geometry. In *Proc. Visualization 90, San Francisco, CA*, pages 361–370, 1990.
43. B. Johnson and B. Shneiderman. Treemaps: A space-filling approach to the visualization of hierarchical information. In *Proc. Visualization '91 Conf*, pages 284–291, 1991.
44. D. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *Transactions on Visualization and Computer Graphics*, 6(1):59–78, Jan–Mar 2000.
45. D. Keim. Visual exploration of large databases. *Communications of the ACM*, 44(8):38–44, 2001.
46. D. A. Keim and H.-P. Kriegel. Visdb: Database exploration using multidimensional visualization. *Computer Graphics & Applications*, 6:40–49, Sept. 1994.
47. D. A. Keim, H.-P. Kriegel, and M. Ankerst. Recursive pattern: A technique for visualizing very large amounts of data. In *Proc. Visualization 95, Atlanta, GA*, pages 279–286, 1995.
48. J. Lamping, R. R., and P. Pirolli. A focus + context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proc. Human Factors in Computing Systems CHI 95 Conf.*, pages 401–408, 1995.
49. J. LeBlanc, M. O. Ward, and N. Wittels. Exploring n-dimensional databases. In *Proc. Visualization '90, San Francisco, CA*, pages 230–239, 1990.
50. Y. Leung and M. Apperley. A review and taxonomy of distortion-oriented presentation techniques. In *Proc. Human Factors in Computing Systems CHI '94 Conf., Boston, MA*, pages 126–160, 1994.
51. H. Levkowitz. Color icons: Merging color and texture perception for integrated visualization of multiple parameters. In *Proc. Visualization 91, San Diego, CA*, pages 22–25, 1991.
52. N. L. M. Kreuseler and H. Schumann. A scalable framework for information visualization. *Transactions on Visualization and Computer Graphics*, 2001.
53. J. D. Mackinlay, G. G. Robertson, and S. K. Card. The perspective wall: Detail and context smoothly integrated. In *Proc. Human Factors in Computing Systems CHI '91 Conf., New Orleans, LA*, pages 173–179, 1991.
54. M. Mehta, R. Agrawal, and J. Rissanen. SLIQ: A fast scalable classifier for data mining. *Conf. on Extending Database Technology (EDBT), Avignon, France*, 1996.
55. T. Munzner and P. Burchard. Visualizing the structure of the world wide web in 3D hyperbolic space. In *Proc. VRML '95 Symp, San Diego, CA*, pages 33–38, 1995.
56. K. Perlin and D. Fox. Pad: An alternative approach to the computer interface. In *Proc. SIGGRAPH, Anaheim, CA*, pages 57–64, 1993.
57. R. M. Pickett and G. G. Grinstein. Iconographic displays for visualizing multidimensional data. In *Proc. IEEE Conf. on Systems, Man and Cybernetics, IEEE Press, Piscataway, NJ*, pages 514–519, 1988.

58. J. R. Quinlan. Induction of decision trees. *Machine Learning*, pages 81–106, 1986.
59. J. R. Quinlan. *C4.5: Programs For Machine Learning.* Morgan Kaufmann, Los Altos, CA, 1993.
60. R. Rao and S. K. Card. The table lens: Merging graphical and symbolic representation in an interactive focus+context visualization for tabular information. In *Proc. Human Factors in Computing Systems CHI 94 Conf., Boston, MA*, pages 318–322, 1994.
61. G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone trees: Animated 3D visualizations of hierarchical information. In *Proc. Human Factors in Computing Systems CHI 91 Conf., New Orleans, LA*, pages 189–194, 1991.
62. M. Sarkar and M. Brown. Graphical fisheye views. *Communications of the ACM*, 37(12):73–84, 1994.
63. Schaffer, Doug, Zuo, Zhengping, Bartram, Lyn, Dill, John, Dubs, Shelli, Greenberg, Saul, and Roseman. Comparing fisheye and full-zoom techniques for navigation of hierarchically clustered networks. In *Proc. Graphics Interface (GI '93), Toronto, Ontario, 1993, in: Canadian Information Processing Soc., Toronto, Ontario, Graphics Press, Cheshire, CT*, pages 87–96, 1993.
64. H. Schumann and W. Müller. *Visualisierung: Grundlagen und allgemeine Methoden.* Springer, 2000.
65. D. W. Scott. *Multivariate Density Estimation.* Wiley and Sons, 1992.
66. J. Shafer, R. Agrawal, and M. Mehta. SPRINT: A scalable parallel classifier for data mining. *Conf. on Very Large Databases*, 1996.
67. B. Shneiderman. Tree visualization with treemaps: A 2D space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, 1992.
68. B. Shneiderman. The eye have it: A task by data type taxonomy for information visualizations. In *Visual Languages*, 1996.
69. B. Spence. *Information Visualization.* Pearson Education Higher Education publishers, UK, 2000.
70. R. Spence and M. Apperley. Data base navigation: An office environment for the professional. *Behaviour and Information Technology*, 1(1):43–54, 1982.
71. R. Spence, L. Tweedie, H. Dawkes, and H. Su. Visualization for functional design. In *Proc. Int. Symp. on Information Visualization (InfoVis '95)*, pages 4–10, 1995.
72. A. Spoerri. Infocrystal: A visual tool for information retrieval. In *Proc. Visualization '93, San Jose, CA*, pages 150–157, 1993.
73. J. Stasko, J. Domingue, M. Brown, and B. Price. *Software Visualization.* MIT Press, Cambridge, MA, 1998.
74. C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis and visualization of multi-dimensional relational databases. *Transactions on Visualization and Computer Graphics*, 2001.
75. D. F. Swayne, D. Cook, and A. Buja. *User's Manual for XGobi: A Dynamic Graphics Program for Data Analysis.* Bellcore Technical Memorandum, 1992.
76. L. Tierney. *LispStat: An Object-Orientated Environment for Statistical Computing and Dynamic Graphics.* Wiley, New York, NY, 1991.
77. J. Trilk. Software visualization. http://wwwbroy. informatik.tu-muenchen.de/˜trilk/sv.html, Oct 2001.
78. P. E. Utgoff. Incremental induction of decision trees. *Machine Learning*, 4:161–186, 1989.
79. P. E. Utgoff, N. C. Berkman, and J. A. Clouse. Decision tree induction based on efficient tree restructuring. *Machine Learning*, 29:5–44, 1997.
80. J. J. van Wijk and R. D. van Liere. Hyperslice. In *Proc. Visualization '93, San Jose, CA*, pages 119–125, 1993.

81. P. F. Velleman. *Data Desk 4.2: Data Description.* Data Desk, Ithaca, NY, 1992, 1992.
82. M. O. Ward. Xmdvtool: Integrating multiple methods for visualizing multivariate data. In *Proc. Visualization 94, Washington, DC*, pages 326–336, 1994.
83. C. Ware. *Information Visualization: Perception for Design*. Morgen Kaufman, 2000.
84. A. Wilhelm, A. Unwin, and M. Theus. Software for interactive statistical graphics - a review. In *Proc. Int. Softstat 95 Conf., Heidelberg, Germany*, 1995.
85. J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, S. A., and V. Crow. Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In *Proc. Symp. on Information Visualization, Atlanta, GA*, pages 51–58, 1995.
86. L. Yan. Interactive exploration of very large relational data sets through 3d dynamic projections. *SIGKDD Int. Conf. On Knowledge Discovery & Data Mining (KDD 2000), Boston, MA*, 2000.

Index