Visual Data Mining

Daniel A. Keim [†], Wolfgang Müller [‡] and Heidrun Schumann [Δ]

[†] University of Constance, Germany, and AT&T Labs, NJ, USA
[‡] University of Applied Sciences, Darmstadt, and Ekkono GmbH, Germany
[Δ] Department of Computer Science, University of Rostock, Germany

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 star report, we provide an overview of information visualization and visual data mining techniques, and illustrate them using a few examples.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Visual Data Mining, Information Visualization

1. Introduction

The amount of data collected in corporate and public databases increases every day. Databases containing several terabyte of data are no longer uncommon. 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 before. These data are 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 them is a difficult task. *Data mining* denotes the approach to analyze this data and to extract information from this data. However, with today's data management systems, it is only possible to view quite small portions of the data. Having no possibility of adequately exploring the large amounts of data, which have been collected because of their potential usefulness, the data becomes useless and the databases become data "dumps".

1.1. Visual Data Mining

Visual data mining is a novel approach to data mining. It denotes the combination of traditional data mining techniques and information visualization methods. The utilization of both the automatic analysis methods and human perception and understanding promises more effective data mining techniques. From the very beginning, information visualization techniques have been considered to be a promising alternative to analysis methods based e.g. on statistic and AI techniques. Information visualization exploits the phenomenal abilities of human perception to identify structures by presenting abstract data visually, allowing the user to explore the complex information space to get insight, to draw conclusions and directly interact with the data.

Visual data mining techniques have proven to be of high value in exploratory data analysis. They have high potential especially

- For exploring large databases,
- When little is known about the data and the exploration goals are vague, and
- When highly inhomogeneous and noisy data is given.

Visual data exploration is intuitive and requires no understanding of complex mathematical or statistical algorithms or parameters. Since the user is directly involved in the exploration process, shifting and adjusting the exploration goals is automatically done if necessary. 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.

1.2. Information Model

The design of visual data mining tools requires a formal and easily adaptable information model for the description of information units and the general characteristics of the information space. It is our goal to define a general model, which is suitable for different domains and a variety of visual data mining applications.

We follow Bertin [14] to distinguish the *information content* and the *structure* in which the information is organized. Furthermore, we follow Wendt [107] who introduces the concept of *information objects* as a necessary abstraction of *data*, which represents information in an information processing system.

Let **IM** be a discrete set of information objects:

$$\mathbf{IM} = \{IO_1, ..., IO_n]$$
$$\forall i \forall j: \quad IO_i = IO_j \quad \Leftrightarrow \quad i = j \quad i,j,n \in \mathbf{N}$$

Information objects are characterized by a set of attributes. These attributes can have arbitrary continuous or categorical ranges of values in order to describe object properties and the characteristics of the information. The function *attr* provides all attributes of a set of information objects.

$$attr(\{IO_1,IO_2,...,IO_n\}) = \{A_1,A_2,...,A_k\}$$
$$\forall i \forall j: \quad A_i = A_j \quad \Leftrightarrow \quad i = j \quad i,j,k,n \in \mathbf{N}$$

The attribute set **AM** is the set of all attributes A_i of the set of information objects.

$$\mathbf{AM} = attr(\{IO_1,IO_2,...,IO_n\}) \quad n \in \mathbf{N}$$

The attributes define dimensions and span the *information space* **IR**, and the ranges of attribute values define the scaling of the related axes of the information space.

The dimensionality of the information space **IR** is defined as the cardinality of the attribute set **AM**.

$$dim(\mathbf{IR}) = |\mathbf{AM}|$$

In other words, the attributes and their ranges of values represent the dimensions of the information space in our model. Thus, information objects IO_i can be understood as points within the multi-dimensional information space.

In order to model arbitrary relations between the information objects IO_i, which might either be given explicitly or obtained implicitly, we introduce the information structure **IS**. The information structure **IS** is defined as a relation on the information set **IM**

$$\mathbf{IS} \subseteq \mathbf{IM} \times \mathbf{IM}$$

The cardinality of **IS** may be 0, i.e. in some cases there may be no description of the relation between information objects.

Summarizing our model, the information space **IR** is defined by means of the attribute set **AM**, which describes the information properties and represents the dimensions of **IR**, the information set IM, where the elements are given as points in **IR** and the information structure **IS**. The information definition given above allows modeling of complex information spaces. Arbitrary visual data mining can be handled due to the use of attributes for characterizing information objects and the use of relations for describing connections between pieces of information.

2. Principles of Visual Data Mining

The goal of the visual data mining approach is the development of a new generation of data mining tools. These tools will provide solutions to the problem of analyzing and understanding the large data sets which we find in research and business.

The integration of information visualization techniques with analytical analysis methods from the fields of data mining and statistics plays a major role in this context. As mentioned above, this integration will allow the exploitation of the phenomenal abilities of the human mind while using the enormous computing power of today's information technology at the same time.

Visual data mining systems may be characterized by a number of aspects [110]:

- Simplicity – Visual data mining systems should be intuitive and easy to use while providing effective and powerful techniques at the same time.
- User autonomy – Visual data mining systems should assist the user but also keep him always in control.
- Reliable – Visual data mining systems should provide estimated error or accuracy of the presented information for each step of the mining process.
- Reusable – A visual data mining system should be adaptable to the various environments and application scenarios.
- Availability – Since the quest for new knowledge cannot be planned, the access to visual data mining systems should be ubiquitous. This calls for portable and distributed solutions, making use of sophisticated user interface technology.
- Security – Knowledge is a valuable good in today's society. Data, background knowledge, and hypotheses as well as gained knowledge should therefore be protected against unauthorized access.

2.1. General Rules for Information Visualization

To end up with an expressive and effective visualization, appropriate transformation operators have to be applied for a given visualization problem. The type of data, the visualization goal, the targeted media, and other contextual information may influence the selection. Models of human vision may play an important role as a guideline, however it is difficult to use such implicit knowledge. Several collections of visualization rules have been published, e.g. [97]. However, there are only few formal approaches laying the ground for explicit rule systems [66][85].

A graphical encoding of an information object has to be expressive, effective, and appropriate. Furthermore, they have to be appropriate in the specific context [84].

The *expressiveness* of an encoding [66] relates to the requirement that all relevant attributes and only those must be expressed by the visualization. This criterion is strongly connected to Tufte's *lie factor*, which is defined as size of effect shown in graphic divided by the size of effect shown in the data [97]: expressive visualizations are characterized by a minimal lie factor.

The *effectiveness* of an encoding depends on the ability of the observer to interpret the presented facts correctly and efficiently. Here, the visual abilities of the observer play a major role. Mackinlay [66] provided a ranking of basic visual encodings with respect to accuracy of interpretation in the cases of nominal, ordinal, and quantitative information.

For information graphics Tufte defined the term *graphical excellence* for well-designed presentation [97]. His rules to achieve graphical excellence are also valid in visual data mining and information visualization. Besides clarity, precision, and efficiency the avoidance of *chart junk* is an important principle.

2.2. Presentation Techniques

Visualizing huge amounts of data requires customized methods and techniques which take into account the limited display capacity of the output devices. To exploit the screen space efficiently and to increase the amount of visualized information, specialized presentation techniques have been proposed. The so-called *Focus & Context* techniques are a popular example of this approach. Focus and Context techniques combine a focus display, which shows a part of the layout at a high degree of detail, and a context display, which presents the whole picture in lower detail to provide an overview. The current point of interest of the user determines the position of the focus.

There are several possibilities for *Focus & Context* techniques. They have been first used in computer science by Furnas [36] for the presentation of structured data. This technique is called *Fisheye View* and achieves context reduction by hiding nodes in the structure based on the distance from the focus and an interest measure. In doing so, focus and context are specified in the information domain rather than in the layout.

In contrast to this approach, *distortion oriented techniques* define focus and context in the presentation domain. Thereby, focus and context both are integrated into one single presentation [64]. Depending on a measure of distance from the focus, the context is distorted such that the space requirements decrease. This method offers a detailed view near the point of interest and maintains large-scale features of the whole layout at the cost of introducing distortion far away from the point of interest. Typical examples of this technique are *graphical Fisheye Views* [80], *Perspective Wall* [67] and *Hyperbolic Viewer* [62][69] (see Figure 1).

Since the focus shows only a part of the whole picture in high detail, the user often wishes to move it to another position to reveal detail there as his point of interest changes. The interaction exploration requires re-computation of the visualization, which needs a powerful processor for fast response using the technique described above. Sakar et. al. [82]

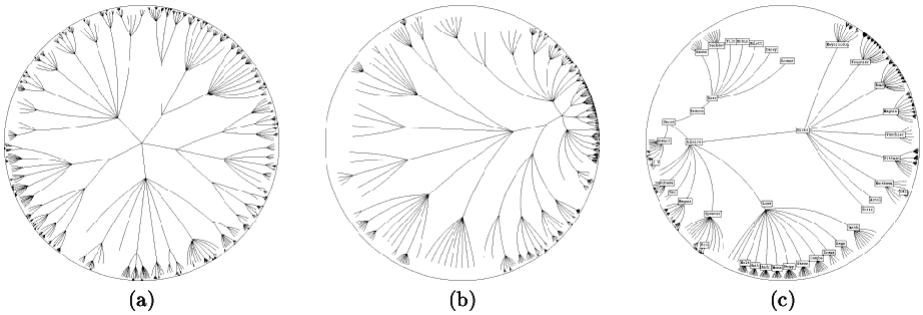


Figure 1*: Hyperbolic Viewer [60]. (a) Original layout (b) Moving the root (c) New focus with annotation*

proposed a method, called *rubber sheets* for graphical layouts, which divides the image into stripes and stretches all pixels of a stripe by the same factor. Thus, computing requirements are decreased. An excellent comprehensive bibliography about non-linear magnification techniques and fisheye views can be found on the non-linear magnification home page [53].

Another approach is the display of focus and context separate over time (this corresponds to zooming) or in two separate presentations side by side. The latter method is called overview & detail [17]. This approach avoids introducing distortion, but the user is required to make the link between focus and context mentally.

2.3. Visual Data Mining Reference Model

In visual data mining we can differentiate between 3 different levels of integration based on the degree of integration of information visualization and automated data mining techniques:

- No or very limited integration. This corresponds to the application of either traditional information visualization or automated data mining techniques on the raw data.
- Loose integration of information visualization and traditional data mining. Visualization and automated mining techniques are applied sequentially. The results of each mining operation can be used as an input to the following analysis step.
- Full integration of information visualization and automated techniques. This corresponds to a full integration of both types of techniques, allowing for a parallel combination and integration of the results of both techniques.

In the visualization process we can distinguish different phases which can be described in terms of data stages (value, analytical abstraction, visualization abstraction, view) and transformation operators (data transformation, visualization transformation, visual mapping transformation) [25].

3. Information Visualization Techniques

We can differentiate between 3 different types of information visualization techniques, differing in the data aspects being visualized. These techniques are targeted to

- The visualization of information structure,
- The visualization of multivariate data, and
- The visualization of entire information objects.

Note that we restrict the discussion to those classes of techniques clearly dealing with abstract information. Techniques targeted to visualize large amounts of time-dependent or spatial data fall into the area of *Scientific Visualization*. See [84] for an overview on these types of techniques.

3.1. Visualization of Information Structure

An important task of visual data mining is the exploration of relationships between information objects. Visualization techniques supporting this goal put the focus on presenting the relations between information objects rather than on visualizing the properties of these elements. The relations can be given explicitly, specified by the information structure **IS** (e.g. **IS** represents the structure of a file system) or they can be given implicitly. Implicit means that the elements of the information structure **IS** have to be computed by automatic mining algorithms (e.g. obtaining relationships based on the similarity of information objects by hierarchical clustering).

A number of customized methods for visualizing the information structure have been developed. We want to distinguish between methods presenting hierarchical structures and methods for more general classes of networks.

Hierarchy Visualization

There are two principal alternatives for visualizing hierarchies:

- Explicit vs. implicit:
  Explicit methods represent the edges between the elements of the hierarchy. Implicit techniques are space-filling methods. They show relations by special arrangements of elements.
- Horizontal vs. radial:
  Horizontal layouts arrange elements line by line. One line depicts the nodes of one level of the hierarchy. Radial layouts present circles with the root of the hierarchy in the midpoint.

In the following, we want to discuss some examples for each strategy. Traditional techniques are explicit methods with horizontal layout. Usually they can display about 100 nodes [62]. If the number of levels and nodes increases, visualizing hierarchies becomes more complicated. In order to solve this problem *Focus & Context* techniques may be applied.

A well-known example for this approach is the *Hyperbolic Viewer* [62]. The Hyperbolic Viewer is an explicit technique with radial layout. The main idea is to use the hyperbolic plane for arranging the nodes of the hierarchy. The node of interest is located at the midpoint of the circle. Due to the fact that the circumference of a circle grows exponentially with the distance to the center in the hyperbolic space the transformation in Euclidian space provides more space for nodes in the neighborhood of the center point. Figure 1 shows an example of the Hyperbolic Viewer.

Another example for an explicit radial layout is the *Magic Eye View* [60]. In this approach a layout of the hierarchy is generated with a simple 2D radial algorithm and mapped onto a hemisphere. A projection is introduced in order to achieve a focus & context display and to enable a smooth transition between these regions. The center of projection is the midpoint of the hemisphere. We construct rays from the center point to each node on the hemisphere. By retaining the direction of these rays and changing the center of projection we obtain new

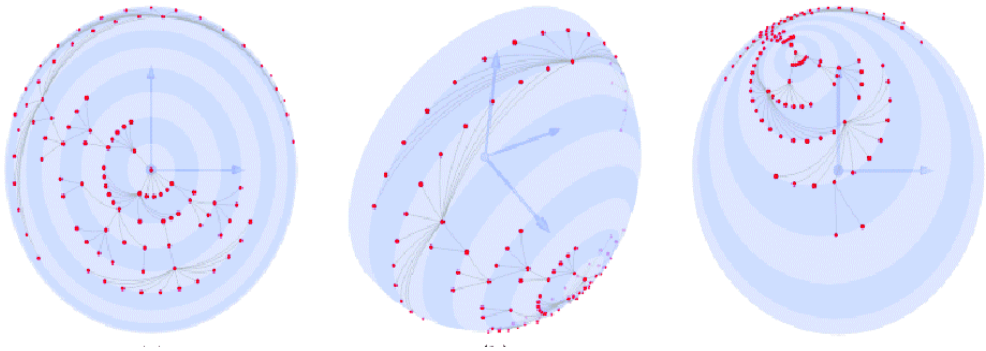


Figure 2: *Magic Eye View with different focus areas [60]*

positions of the nodes as the intersection points of the rays with the hemisphere. Thus, the distances between nodes are increased or decreased depending on the position of the center of projection. The increasing region provides more space to view details and can be considered as the focus region. Figure 2 demonstrates this technique.

Popular examples for space filling techniques are *Treemap* [87] and *Sunburst* [92].

Treemaps use a rectangular layout with a recursive subdivision according to the number and size of child nodes. The root is presented by the whole rectangle. There exist some extensions to this basic principle, e.g. the *Cushion Treemap* [100]. Figure 3 demonstrates both techniques.

In contrast, *Sunburst* uses a radial layout (see Figure 4). A small circle presents the root and for each level a new ring is added to the graph. The rings are subdivided in sectors according to the number and size of nodes in the respective levels. No child node is placed outside of the sector of its father node. The sunburst-technique was extended to allow a more detailed view on interesting regions. Three focus techniques (Angular detail, detail outside, and detail inside) show details in different regions of the image.

Visualization of more General Networks

When visualizing general networks, an explicit representation of relationships is required. Several methods have been proposed in the past, especially for specific application domains. A popular system in this context is *SDM – Selective Dynamic Manipulation* [27]. Furthermore, special systems for the visualization of electric power system information [71] or for the visualization of large-scale telecommunication networks [59] have been presented.

Adaptive Techniques and Information Hiding

Exploring very large graphs requires special treatment. Large structures can be effectively handled by using adaptive techniques and information hiding. We may start with an overview image, integrate adaptive features into the image and refine it for regions of interest. This can be done for example by applying the Fish Eye View of Furnas [36] or by an

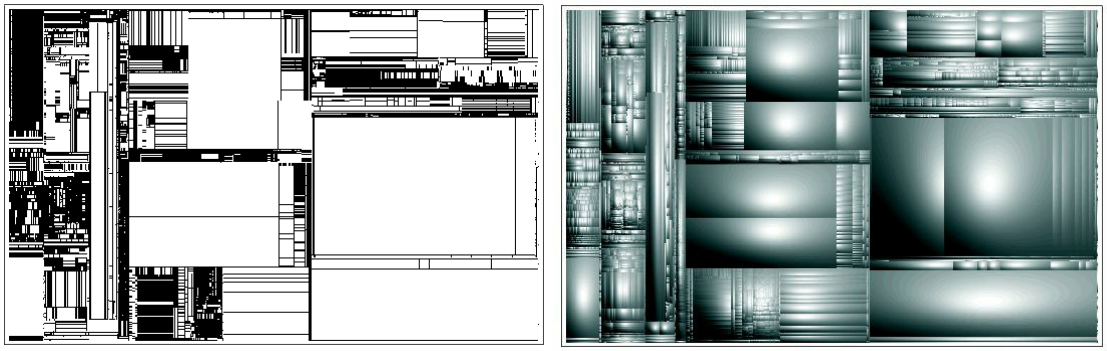


Figure 3: *Treemap (left) and Cushion Treemap (right) (http://www.win.tue.nl/sequoiaview/)*

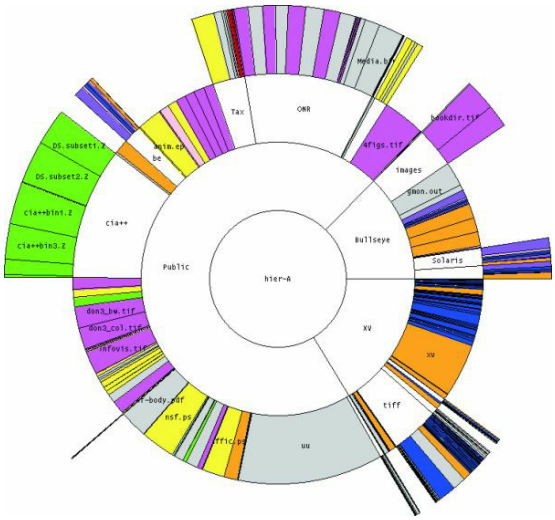


Figure 4: *Visualization of a UNIX file system directory with the SUNBURST technique (http://www.cc.gatech.edu/gvu/ii/sunburst/)*

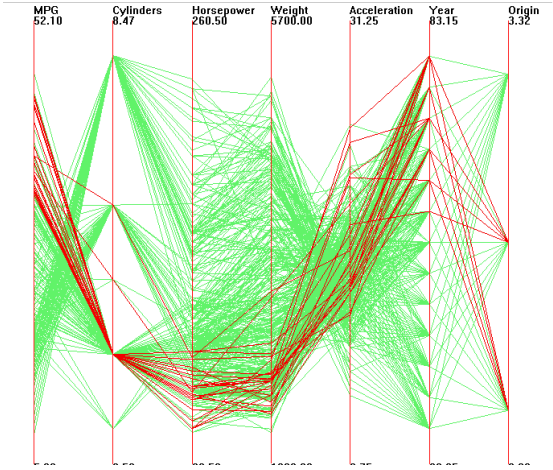


Figure 6*: Parallel coordinates for the visualization of car data: each multi-dimensional data item is displayed as a polygonal line intersecting the dimension axes at the position corresponding to the data value for the dimension [53] (©IEEE)*

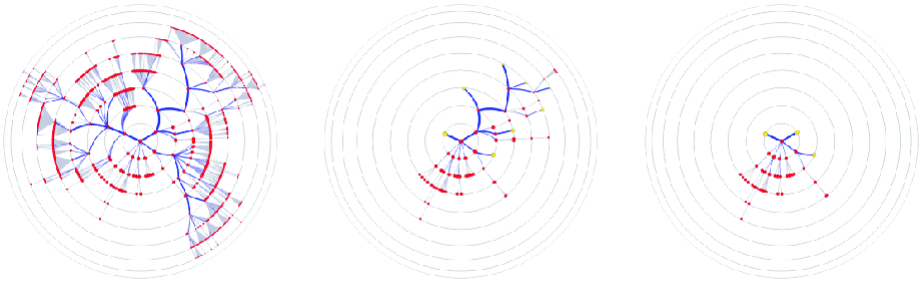
automatic folding and unfolding of subtrees [47]. The technique can be combined with the use of so-called Strahler numbers. Strahler numbers denote the complexity of a node's subtree. They are computed for all non-leaf nodes and may be mapped onto colors and widths of the incoming edges, indicating the direction in which a tree actually grows [47]. Figure 5 demonstrates an application of adaptive techniques for the Magic Eye View.

3.2. Visualization of Multivariate Data

In addition to exploring the information structure, the identification of *structures in the data values* is an important task in visual data mining. The identification of structures in the data requires techniques for visualizing the data values of the attribute set **AM**, i.e. techniques to visualize qualitative and quantitative properties of the information objects. Important techniques in this context are: panel matrices, parallel or star coordinates, icon-based techniques and pixel-based techniques.

Panel Matrices

The basic idea of *Panel matrices* is to arrange bivariate displays of adjacent attributes in matrix form. A popular visualization technique in this category is the *Scatterplot Matrix*, where multiple adjacent scatterplots are displayed in one image [28]. Other examples are *Hyperslices* [101] and *Prosection Views* [38].



(a) $\vartheta_{\mathrm{FOLD}} = \infty$ (b) $\vartheta_{\mathrm{FOLD}} = 15$ (c) $\vartheta_{\mathrm{FOLD}} = 35$

Figure 5: *Magic Eye View with Strahler numbers and Automatic Folding for different threshold[103]s.*

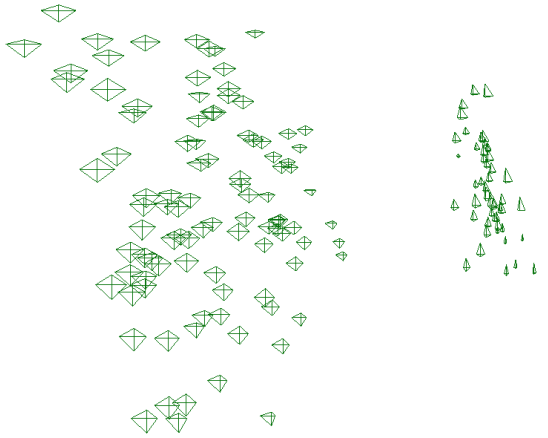


Figure 7: *The iris data set - displayed using star glyphs positioned based on the first two principal components (generated using XmdvTool [105])*

Parallel and Star Coordinates

Parallel and Star Coordinates map the n-dimensional space onto a two-dimensional plane. In our context, the information space **IR** is mapped onto the two display dimensions. A coordinate axis is constructed for each dimension of **IR** and scaled from the minimum to the maximum value of the corresponding attribute. In case of *Parallel Coordinates* the axes are parallel and equidistant. *Star Coordinates* form star-shaped axes. Each information object is presented as a polygonal line intersecting each of the axes at the point corresponding to the value of the considered attribute. Figure 6 presents an example for a *Parallel Coordinates* visualization.

Icon-based Techniques

Another class of visual data exploration techniques are the icon-based techniques. The idea is to map the attribute values of an information object to the features of an icon. Icons may be defined arbitrarily - for example as little faces in the example of *Chernoff Faces* [24], *Needle Icons* [1], *Star Icons* [105], *Stick Figure Icons* [75], *Color Icons* [55] [65], *TileBars* [44][45]. The visualization is generated by mapping the attribute values of each information object to the features of the icons. In case of the *Stick Figure* technique, for example, two dimensions of the information space 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. Transitions between two regions of different texture are readily detectable, as this is an innate pre-attentive perceptual ability of the human visual system. Figure 7 shows an example of this class of techniques. Each information object is represented by a star icon/glyph, where each attribute value 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 attributes could also be mapped to the icon position.

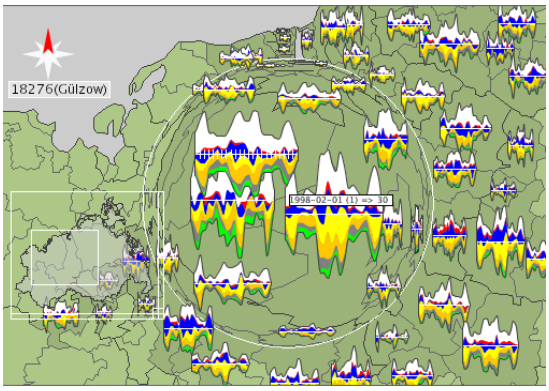


Figure 9: *Focus & Context-Display of health data over an area of Germany with ThemeRiver-icons [96].*

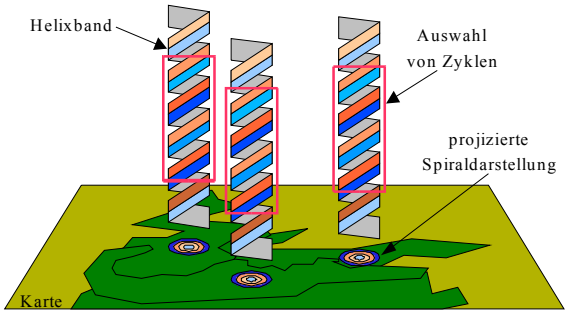


Figure 8: *Illustration of the Helices over a landscape [96].*

An advantage of icon-based techniques is the possibility to arrange icons in a 2-dimensional or 3-dimensional space. Thus, information objects can be related to a spatial context. Furthermore, asides the known icon-types several visualization techniques can be used to produce special types of icons [96]. Figure 8 demonstrates this principle for helices [106] and Figure 9 for *ThemeRiver* [43].

Pixel-based techniques

The basic idea of pixel-based techniques is to map each attribute value to a colored pixel and group the pixels belonging to each dimension of the information space into adjacent areas [54]. Since in general dense pixel displays use one pixel per data value, the techniques allow the visualization

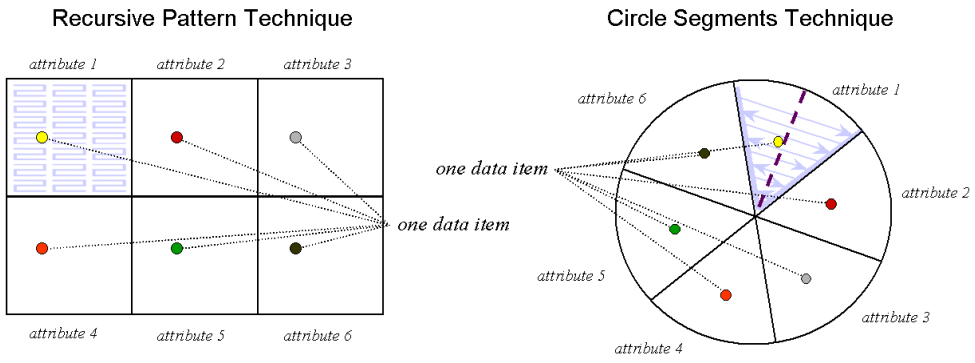


Figure 10: *Basic Idea of Recursive Pattern and Circle Segments Techniques*

of the largest amount of data possible on current displays (up to about 1,000,000 data values). If each attribute value is represented by one pixel, the main question is how the pixels are arranged on the screen. Dense pixel displays use different arrangements to provide detailed information on local correlations, dependencies, and hot spots.

Well-known examples are the *Recursive Pattern* technique [56] and the *Circle Segments* technique [8]. The recursive pattern technique is based on a generic recursive back-and-forth arrangement of the pixels (see Figure 10) and is particularly aimed at representing data sets with a natural order according to one attribute (e.g. 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 basic 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_i and width w_i 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 that the basic elements that are arranged on level i are the patterns resulting from level i - 1. Figure 11 provides an example recursive pattern visualization of financial data. 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 [8] is to represent the data in a circle that is divided into segments, one for each attribute (see Figure 10). 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 11 shows an example of Circle Segment visualization using the same data (50 stocks) as used before with the Recursive Patterns technique.

3.3. Visualization of Entire Information Elements

Another set of visualization techniques focuses on the presentation of all aspects of information objects, mostly to support their identification, their analysis, or to find relations to other elements. Typical examples are the visualization of DNA sequences, documents, or search results. The transition to information graphics - specifically designed for a given application - is sometimes fluid. In the following we restrict the discussion to techniques that present a somehow general approach to the visualization of information objects.

In this context, the information objects are often characterized by a large number of attributes in a multidimensional information space. Visualization of such multidimensional objects requires a mapping onto the restricted dimensions of the presentation space. To achieve a useful mapping, a feature representation abstracting the object and its attributes is necessary. The feature representation needs to represent the diverse aspects of the information objects.

A number of techniques can be applied to reduce the dimensionality of the feature space, while capturing the important interrelations between single attributes. Usually, these techniques also provide a spatial arrangement of the information objects in the resulting feature space, which corresponds to the semantic distances. The following techniques are often used for dimensionality reduction:

- Factor Analysis,
- Multidimensional Scaling, and
- Kohonen networks.

Factor analysis describes a general class of techniques targeted at reducing a larger number of variables (attributes) to a smaller number of factors. The factors are calculated as a weighted linear combination of the original variables. The

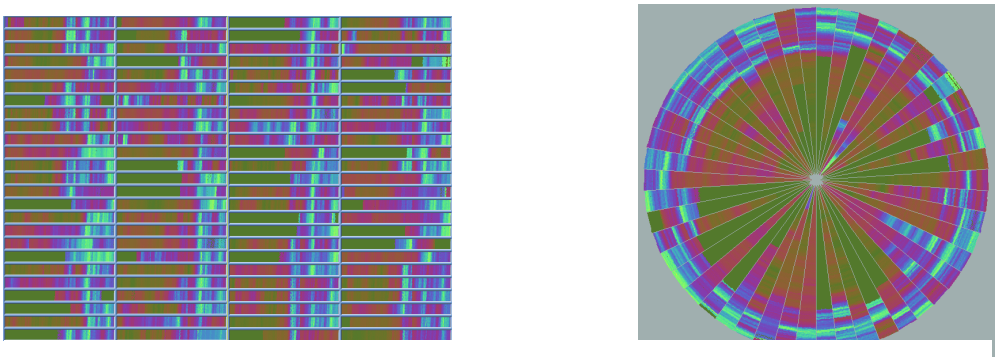


Figure 11:*Visualization of stock data with the Recursive Pattern Technique [56](left) and the Circle Segments Technique [8] (right). Both graphs present daily data for 20 years (Jan. 74 - Apr. 95) of 100 stocks of the Frankfurt stock index. Data values are mapped to colored pixels, where high values correspond to bright colors. The position of each pixel is based on a recursive generalization of a line- and column-based arrangement in case of the recursive pattern technique. For the Circle Segments Technique, different dimensions are mapped to the segments of the circle and pixels are arranged in a back-and-forth fashion adjacent to the segment-halving line.,(©IEEE)*

principal component analysis [29] is a specific instance of a factor analysis model where the weights are based on a least square fit on the feature matrix. Eigenvectors and eigenvalues are used to determine the principal components that are used to summarize the feature matrix using a reduced number of dimensions.

Multidimensional Scaling [30][61] represents a class of optimization techniques where lower dimensional representations approximating the object's distances in information space are generated. The specific algorithms differ in the type of stress function used, which is based on the dissimilarities of the information objects and on the type of optimization strategy. *Simulated annealing* achieves good results in this context [11].

Spring Models [95] make use of a similar optimization approach. In this case, however, predefined feature points (one for each attribute) are fixed in the target space. An information object is mapped into the target space by calculating the effect of virtual springs attached to the information object and each feature point. The effect of the springs depends on the distance of the feature point from the information object in information space. The resulting representation resembles the semantic distances of information objects but highly depends on the selection of the feature points. A general problem of these techniques is their exponential complexity. We can reduce this complexity by clustering the information objects in high dimensional information space and using the *centroids* of the resulting clusters in the following dimension reduction steps instead [109].

Kohonen Networks (self-organizing maps, SOM [58]) provide another approach for generating a spatial layout of information objects. Kohonen networks lend from biological models and provide both, a dimension reduction of the attributes to two dimensions and a classification of the data objects based on their features in information space.

Feature vectors with a reduced dimensionality can be visualized using standard visualization techniques which map the features onto combinations of

- Position (x,y),
- Color,
- Form, and
- Connection

Visualization techniques usually extend standard 2D and 3D scatter plots and graphs. Color is used to encode single aspects of the feature vector. Certain techniques also map information onto the form of the presentation objects, which is similar to the icon-based visualization techniques [49][95].

A combination of position and form is used in the information *Landscape Visualization* paradigm [111]. Information objects are positioned in the 2D plane based on specific features. Additional features are mapped onto the third dimension and characteristics of 3D icons are used to represent the information objects in 3D space. A 3D rendering leads to a landscape of visualization objects.

In the following, we discuss the different approaches and corresponding techniques with examples in different application domains.

Document Visualization

ValueBars [26] represent a general approach to add a visualization of quantitative attributes to a line-oriented listing. Attributes are visualized in a different value bar and each

attribute is mapped onto the size of a field in this value bar. On a selection of an information object in the list all corresponding fields are highlighted in the value bars, thus allowing for a direct identification of the information object's attributes.

TileBars ([44][45], see icon-based techniques) are frequently used to visualize the relevance of documents with respect to a search query. Each document is represented by a number of tiled bars. The length of these bars corresponds to the length of the document. Each bar again corresponds to a component of the search query, usually a requested category. The correspondence of a document's segment to a requested category is represented by coloring the corresponding tile in the category bar using an appropriate gray scale value. The tiles may also be used as an interactive short cut to access the corresponding paragraph of the document. Figure 12 displays an example for the application of TileBars to visualize search results.

Galaxies [46][109] are an example of a direct visualization of the dimension reduction results. Information objects are represented by marks or glyphs at the positions in the visualization of the information space, determined by the previous dimension reduction step. Usually, the visualization space is three-dimensional and appropriate navigation techniques allow sifting through the information available. In similar approaches 2D and 3D graphs are used to visualize document relationships [46]. Here again the object's positions are determined in a preceding dimension reduction step.

ThemeScapes [109] are an example for the application of the information landscapes paradigm to document visualization. In this technique, the first two spatial dimensions are used to represent thematic similarity of documents while the third dimension is used to represent the strength of a theme in a given region. Figure 13 shows an example for the results on the visualization of document relations.

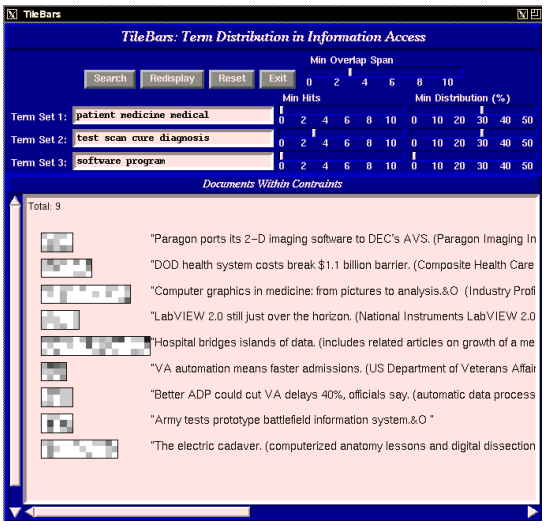


Figure 12: *Example of the use of TileBars for the visualization of search results in a medical database [44]*

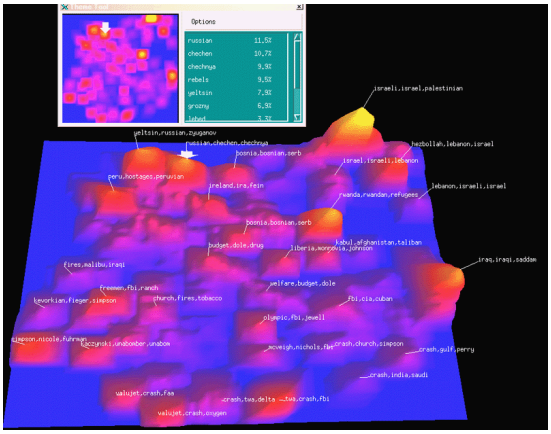


Figure 13: *ThemeView visualization of document relations [72]*

Software Visualization

In software visualization, a number of document visualization are applicable. However, in addition to text information a large number of other qualitative and quantitative data is available, such as execution statistics, error characteristics, and version information. The most prominent example in this context is *SeeSoft* [33]. SeeSoft uses an information mural approach [42] for the representation of the software code: lines of code are represented in a zoomed manner to give context information on the software code. Additional line color-coding may provide supplementary information such as code author, frequency of code change in a sector, last modification, etc.

Visualization of Genome Data

In the analysis of genome data researchers are interested in

- Detailed information about the order of elements in a genome or protein sequence,
- The structural domains in a sequence,
- Detailed information about the physical and chemical characteristics of the sequences, and
- The function of a new sequence, which often can be deduced by comparing the sequence to homologues sequences in a database.

For the comparison with sequences in a database a number of customized algorithms have been developed, among them *FASTA* [73] and *BLAST* [6]. These tools provide detailed information on all alignments, the matching sequences in the database. Usually, genome and protein sequences are

visualized as text strings where a color-coding is used to enhance the identification of nucleotides, proteins, or functional elements. Form is also used as an abstraction of specific protein codes in this context to extend the available graphical codes.

H-curves [63] are a 3D visualization technique targeted to present DNA and RNA strings of A, G, C, and T nucleotides. It is based on a G-curve representation [41] of the sequence, which is a five-dimensional interpretation of the sequence data where the first four dimensions are used to present the occurrence of nucleotides of a specific type while the fifth dimension is used to present the position in the sequence. The H-curve starts at (0,0,0) and is defined by adding (1,1,1) for each occurrence of an A nucleotide, (-1,0,1) for a G nucleotide, (-1,1,-1) for a C nucleotide, and (1, 1, -1) for a T nucleotide. Interactive rotation of the complete scene allows an analysis of the H-curve from different angles, which is especially useful since in the side and front view purin and pyrimidin bases respectively are directly visible. Figure 14 gives an example.

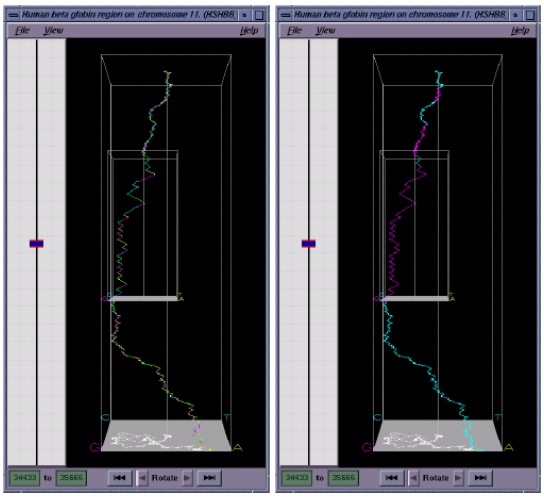


Figure 14: *H-curve representation of human beta globin on chromosome 11 [63]*

4. Interaction Techniques for Visual Data Mining

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 presented data and dynamically change the visualizations according to the exploration objectives. In addition, they enable the user to relate and to combine multiple independent views of the data.

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 the visualization or parts of it, furthermore, they allow for the modification of 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 and other query mechanisms (*indirect manipulation*). Some examples of interaction techniques are described below.

4.1. 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 the properties of the desired subset (*querying*). Browsing is 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 exploration. An example of a tool that can be used for interactive filtering is the *Magic Lens* [16][35]. Magic Lenses apply an interactive tool similar to a magnifying glass to filter the data directly in the visualization. The data under the magnifying glass is processed by the filter and displayed different from the remaining data set. Usually, the selected region is presented in more detail, while the rest of the visualization remains unaffected. However, other representations of the selected area providing a different view on the data are also possible. Note that several lenses with different filters may be used; if the filters overlap, they are combined. Other examples of interactive filtering techniques and tools are *InfoCrystal* [90], *Dynamic Queries* [4][32][39], and *Polaris* [93].

4.2. 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 *Table Lens* approach [78]. Getting an overview of such data sets is difficult if the data are displayed in textual form. The basic idea of *Table Lens* is to represent each numerical value by a small bar. All bars are one-pixel high while the lengths are determined by the attribute values.

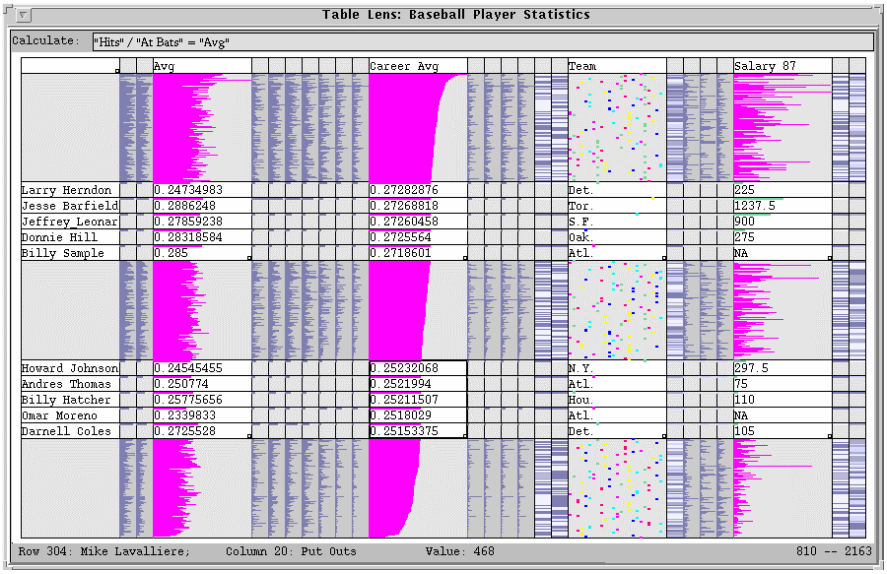


Figure 15: *Table Lens visualization of baseball player statistics (used by permission of R. Rao, Xerox PARC©ACM)*

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 15 shows an example of a baseball database with a few rows and columns being selected in full detail. Other examples of techniques and systems that use interactive zooming include *PAD++* [12][13][74], *IVEE/Spotfire* [3], and *DataSpace* [10]. A comparison of fisheye and zooming techniques can be found in [83].

4.3. Modification of the focus area

As mentioned before Focus & Context techniques show portions of the data with a high level of detail (Focus) while others are shown with a lower level of detail (context). Since the focus shows only a part of the whole picture we need interaction techniques to modify the focus area. Useful manipulation functions are: *Resize* (changing the size of the focus area., *Pointing & Selecting* (clicking a point by the user and calculating a new focus area automatically) and *Dragging* (moving the focus by the user). Furthermore, combinations of these techniques are possible.

4.4. 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. For example, scatterplots of different projections may be linked by consistently coloring subsets of the presented 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*. Example tools and systems include *SPlus* [17], *XGobi* [18][94], *XmdvTool* [105] and *DataDesk* [102][112].

5. Automated Mining and Visualization

[1]There are a number of visualization techniques that have been developed to support automated data mining, such as

[1] Portions of the text in this section have first been published in [57].

association rule generation, classification, and clustering. In the following, we describe how visualization techniques can be used to support these tasks.

5.1. Association Rule Generation

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 data set. 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 implies 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 $X \subseteq I$, $Y \in I$, $X,Y \neq \varnothing$. 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 be overlooked.

Visualization techniques have been used to overcome this problem and to allow an interactive selection of good support and confidence levels. Figure 16 shows the SGI MineSet's *Rule Visualizer* [21] which maps the left and right hand sides of the rules to the x- and y-axes of the plot, respectively, and shows the confidence as the height of the bars and the support

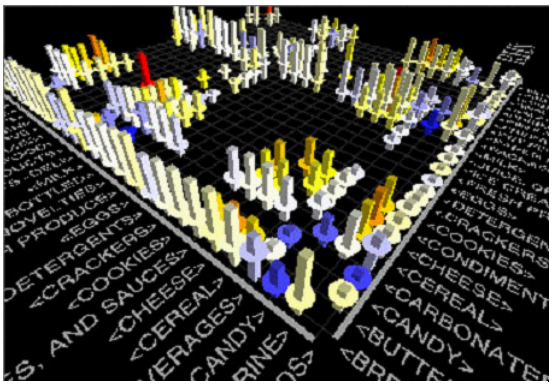


Figure 16*: Visualization of association rules with SGI MineSet's* Rule Visualizer *[21]*

as the height of the discs. The color of the bars shows the interestingness of the rule. Using the visualization, the 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 17 shows two alternative visualizations called *Mosaic* and *Double Decker Plots* [50]. The basic idea is to partition a rectangle on the y-axis according to one attribute and make the size of 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 their 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 $X_1 \wedge X_2 \rightarrow Y$. Mosaic plots are restricted to two attributes on the left side of the

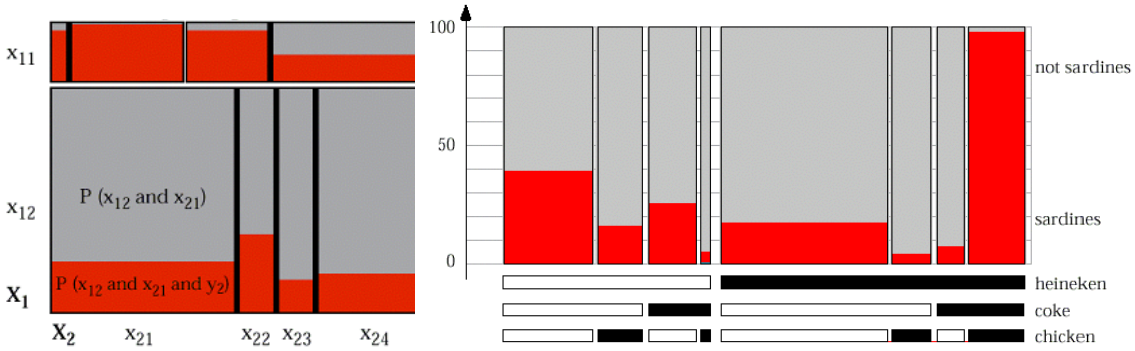


Figure 17*: Mosaic Plot (left) and Double Decker Plot (right) Association Rule Visualizations (from [50] © ACM)*

association rule. *Double Decker Plots* can be used to show more than two attributes on the left side. The idea is to display a hierarchy of attributes on the bottom (heineken, coke, chicken in the example shown in Figure 17 corresponding to the left hand side of the association rules. The bars on the top correspond to the number of items in the considered 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 17) and therefore represent the support. Other approaches to association rule visualization include graphs with nodes corresponding to items and arrows corresponding to implications (as used in DBMiner [31]) and association matrix visualizations to cluster related rules [42].

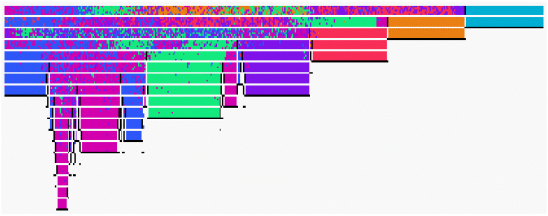


Figure 19: *Visualization of a decision tree [9] for the segment training data from the Statlog benchmark having 19 attributes (used by permission of M. Ankerst [9] © ACM)*

5.2. Classification

Classification is the process of developing a classification model based on a training data set with known class labels. 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 to classify data for which the class labels 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 tasks. Popular are algorithms that inductively construct decision trees. Examples are *ID3* [76], *CART* [20], *ID5* [98][99] *C4.5* [77], *SLIQ* [68], and *SPRINT* [86]. 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.

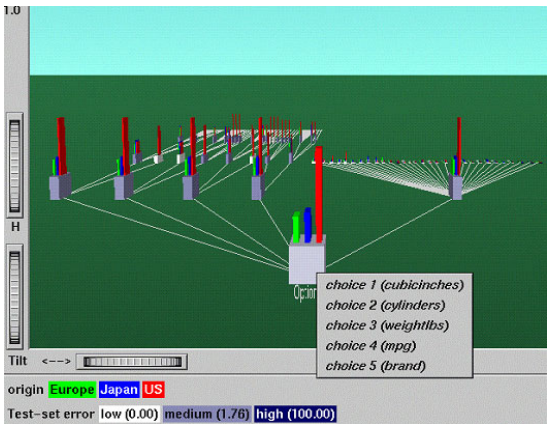


Figure 18: *MineSet's Decision Tree Visualizer [21] (© SGI)*

Visualization techniques can help to overcome these problems. The decision tree visualizer in SGIs MineSet system [21] shows an overview of the decision tree together with important parameters such as the attribute value distributions (see Figure 18). The system allows an interactive selection of the attributes shown and helps the user to understand the decision tree. A more sophisticated approach, which also helps in decision tree construction, is visual classification as proposed in [9]. The basic idea is to show each attribute value by a colored pixel and arrange them in bars - similar to the *Dense Pixel Displays* presented in subsection 3.2. 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 decision tree resulting from this process is shown in Figure 19. 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)
- Distribution (frequency and location of the training instances of all classes).

Some of this information might also be provided by annotating the standard representation 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 provide a better understanding of the classification models and they can help to interact more easily with the classification algorithms in order to optimize the model generation and classification process.

5.3. Clustering

Clustering is the process of finding a partitioning of the data set into homogeneous subsets called clusters. Unlike classification, clustering is often implemented as a form of *unsupervised learning*. This means that the classes are

unknown and no training set with class labels is available. A wide range of clustering algorithms has been proposed in the literature including density-based methods such as *KDE* [94] and *linkage-based methods* [19]. Most algorithms use assumptions about the properties of the clusters which either can either applied as defaults or have to be specified by the user as input parameters. Depending on the parameter values, the user obtains different clustering results. In two- or three-dimensional space, the impact of different algorithms and parameter settings can be explored easily 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 [113]. Figure 20 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 it 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 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 *(Ordering Points To Identify the Clustering Structure)* [7]. The idea of OPTICS is to create a one-dimensional ordering of the database representing its density-based clustering structure. Figure 21 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 21) is similar. Jumping to another 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.

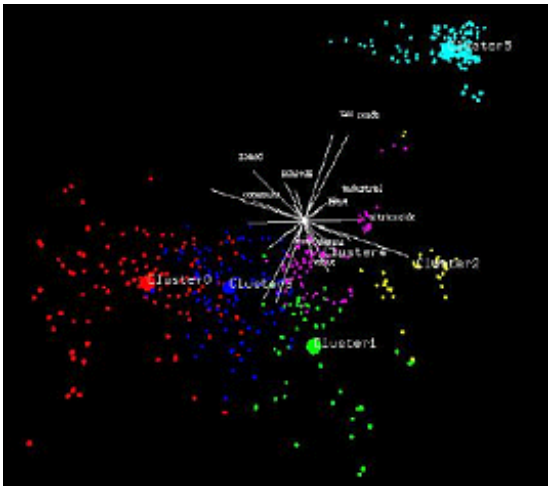


Figure 20*: Visualization based on a projection into 3D space (from [113] © ACM)*

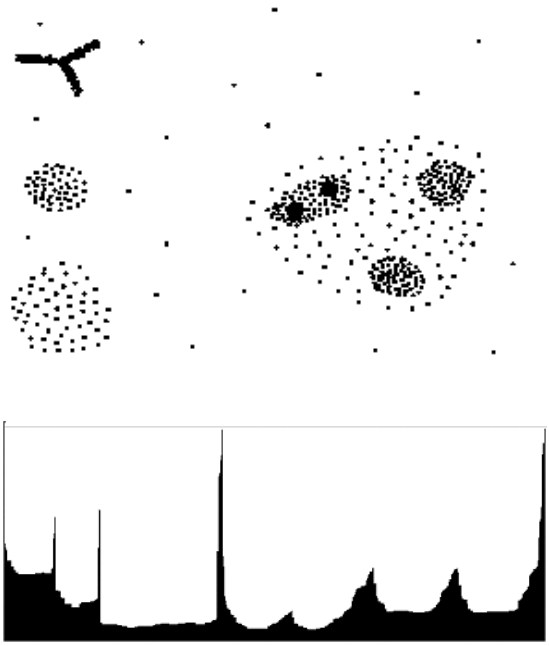


Figure 21*: Data Set (top) and Reachability Plot (bottom) from OPTICS Visual Clustering (used by permission of M. Ankerst [7] © ACM)*

Another interesting approach is the *HD-Eye* system [48]. 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 22 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 22) visualize 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 icons

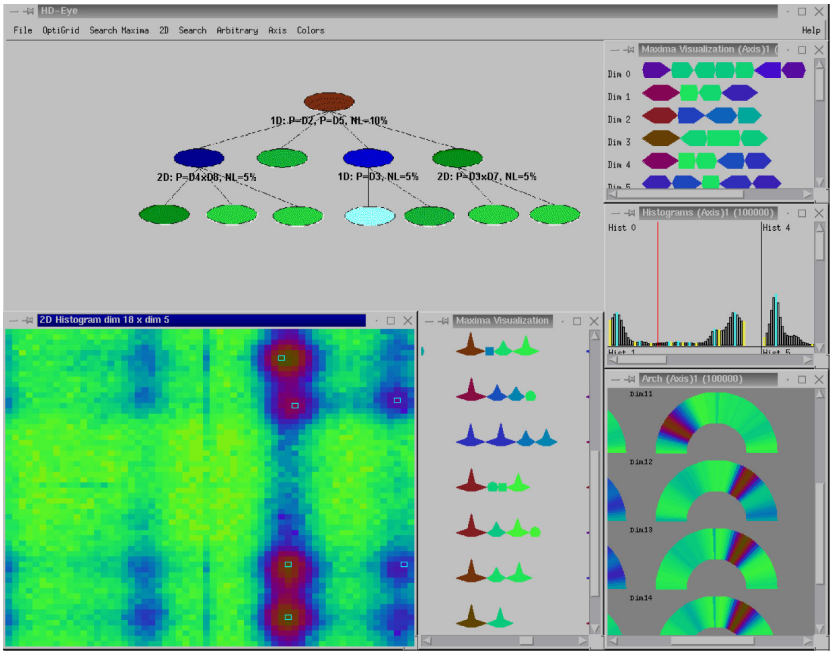


Figure 22: *HD-Eye screen-shot [48] showing different visualizations of projections and the separator tree. Clockwise from the top: separator tree, iconic representation of 1D projections, 1D projection histogram, 1D color-based density plots, iconic representation of multi-dimensional projections and color-based 2D density plot. (from [48] © IEEE)*

corresponds to the number of peaks in the projection and their color to the number of data points belonging to the maximum. 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 maximum is separated from the others is reflected by 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 taken for the partitioning. In addition, the partitioning can interactively and directly be specified within the visualizations, allowing the user to define non-linear partitionings (e.g., in the 2D density plots).

6. Conclusions

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 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 semi-structured information, including database management and data warehouse systems.

References

1. Abello J.; and Korn, J.: MGV: A system for visualizing massive multi-digraphs, Transactions on Visualization and Computer Graphics, 2001.

2. Agrawal, R.; Mannila, H.; Srikant, R.; Toivonen, H.; and Verkamo, A.: Fast discovery of association rules.,Advances in Knowledge Discovery and Data Mining, pages 307–328, 1996.

3. Ahlberg C.; and Wistrand, E.: IVEE: An information visualization and exploration environment. In Proc. Int. Symp. on Information Visualization, Atlanta, GA, pages 66–73, 1995.

4. Ahlberg, C. and Shneiderman, B.: 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.

5. Alpern, B.; and Carter, L.: Hyperbox, In Proc. Visualization '91, San Diego, CA, pages 133–139, 1991.

6. Altschul, S.F., et.al.: Basic local alignment search tool, Journal Molecular Biology, 215, 1990, pp. 403-410.

7. Ankerst, M.; Breunig, M.; Kriegel, H.P.; and Sander. J.: 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. Ankerst, M.; Keim,, D.A.; and Kriegel, H.P.: Circle segments: A technique for visually exploring large multidimensional data sets, In Proc. Visualization 96, Hot Topic Session, San Francisco, CA, 1996.

9. Ankerst, M.; Ester, M.; and Kriegel, H.P.: Towards an e_ective 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.

10. Anupam, V.; Dar, S.; Leibfried, T.; and Petajan, E.: Dataspace: 3D visualization of large databases, In Proc. Int. Symp. on Information Visualization, Atlanta, GA, pages 82–88, 1995.

11. Basalaj, W.: Proximity Visualization of Abstract Data, http://www.pavis.org/essay/index.html, January 2001.

12. Bederson, B.B.: Pad++: Advances in multiscale interfaces. In Proc. Human Factors in Computing Systems CHI '94 Conf., Boston, MA, page 315, 1994.

13. Bederson, B.B.; and Hollan, J.D.: Pad++: A zooming graphical interface for exploring alternate interface physics. In Proc. UIST, pages 17–26, 1994.

14. Bertin, J.: Graphics and Graphic Information Processing, deGruyter Press, Berlin 1977.

15. Bertin, J.: Semiology of Graphics, The University of Wisconsin Press, 1983.

16. Bier, E.A.; Stone, M.C.; Pier, K.; Buxton, W.; and DeRose, T.: Toolglass and magic lenses: The see-through interface. In Proc. SIGGRAPH '93, Anaheim, CA, pages 73–80, 1993.

17. Becker, R.A.; Chambers, J.M; and Wilks, A.R.: The New S Language, Wadsworth & Brooks/Cole Advanced Books and Software, Pacific Grove, CA, 1988.

18. Becker, R.A.; Cleveland, W.S.; and Shyu, M.J.: The visual design and control of trellis display, Journal of Computational and Graphical Statistics, 5(2):123–155, 1996.

19. Bock, H.H.: Automatic Classification. Vandenhoeck and Ruprecht, G¨ottingen, 1974.

20. Breiman, L.; Friedman, J.H.; Olshen, R.A.; and Stone, C.J.: Classification and Regression trees, Wadsworth Inc., 1984.

21. Brunk, C.; Kelly, J.; and Kohavi, R.: MineSet: An Integrated System for Data Mining. KDD 1997, pp. 135-138.

22. Card, S. K.; Mackinlay, J. D.; Shneiderman, B. (eds.): Readings in Information Visualization – Using Vision to Think. Morgan Kaufman, San Francisco, CA, 1999.

23. Carpendale, M.S.T.; Cowperthwaite, D.J.; and Fracchia., F.D.: Extending distortion viewing techniques from 2D to 3D data, IEEE Computer Graphics and Applications, Special Issue on Information Visualization. IEEE Journal, Press, 17(4), pp. 42–51, July 1997.

24. Chernoff, H.: The use of faces to represent points in k-dimensional space graphically, Journal Amer. Statistical Association, 68:361–368, 1973.

25. Chi, E. H.: A Taxonomy of Visualization Techniques using the Data State Reference Model. Proc. IEEE Information Visualization 2000, IEEE Press, 2000, pp. 69-75.

26. Chimera, R.: ValueBars: An Information Visualization Navigation Tool for Multi-attribute Listing, Proc. ACM CHI '92, pp. 293-294, 1992.

27. Chuah, M.; Roth, S.F.; Mattis, J.; Kolojejchick, J.: SDM: malleable information graphics. Proc. IEEE Information Visualization, October 1995, pp. 36 - 42.

28. Cleveland, W. S.: Visualizing Data, Hobart Press, New Jersey, 1993

29. Cliff, N.: Analyzing Multivariate Data, Harcourt, Brace Jovanovich Pub., 1987.

30. Cox, T.F.; and Cox, M.A.A.: Multidimensional Scaling. Chapman & Hall, London, 1994.

31. DBminer. http://www.dbminer.com, 2001.

32. Eick, S.G.: Data visualization sliders. In Proc. ACM UIST, pages 119–120, 1994.

33. Eick, S.G.; Steffen, J.L; Sumner, E.E.: SeeSoft – A Tool For Visualizing Line Oriented Software Statistics, IEEE Transactions on Software Engineering, 18, Nov. 1992, pp. 957-968

34. Fayyad, U.; Grinstein, G.; Wierse, A.: Information Visualization in Data Mining and Knowledge Discovery. Morgan Kaufmann Publishers, San Francisco, 2001.

35. Fishkin K.; and Stone, M.C: Enhanced dynamic queries via movable filters. In Proc. Human Factors in Computing Systems CHI '95 Conf., Denver, CO, pages 415–420, 1995.

36. Furnas, G.W.: The FishEye View: a new look at structure files, Technical Report, Bell Laboratories, 1981.

37. Furnas, G.W.: Generalized fisheye views. In Proc. Human Factors in Computing Systems CHI 86 Conf., Boston, MA, pages 18–23, 1986.

38. Furnas, G.W.; Buja, A.: Prosection Views: Dimensional Inference through sections and Projections, J.Computational and Graphical Statistics, vol. 3, no.4, 1994, pp..323-353.

39. Goldstein, J.; and Roth, S.F: Using aggregation and dynamic queries for exploring large data sets, In Proc. Human Factors in Computing Systems CHI '94 Conf., Boston, MA, pp. 23–29, 1994.

40. Gross, M. H.; Sprenger, T.C.; Finger, J.: Visualizing Information on a Sphere. Proc. IEEE Information Visualization, October, 1997, pp. 11-16.

41. Hamori, E.: Visualization of biological information encoded in DNA, in: Pickover, C.A.; and Tewksbury, S.K. (eds.): Frontiers of Scientific Visualization, Vol. 3, chapter 4, Wiley, pp. 90-121, 1994.

42. Hao, M.; Hsu, M.; Dayal, U.; Wei, S.F.; Sprenger, T.;and Holenstein, T.: Market basket analysis visualization on a spherical surface, Visual Data Exploration and Analysis Conference, San Jose, CA, 2001.

43. Havre, S.; Hetzler,B.; Nowell, L.: ThemeRiver: Visualizing Theme Changes over Time, Proceedings IEEE InfoVis'2000, Salt Lake City, Oct., 2000, pp. 115-124.

44. Hearst, M.A.: TileBars: Visualization of Term Distribution Information in Full Text Information Access, Proc. ACM CHI 1995, pp. 59-66.

45. Hearst, M.A.: User Interfaces and Visualization, in: Baeza-Yates, R.; and Ribeiro-Net, B. (eds.): Modern Information Retrieval, Addison-Wesley, Harlow, 1999.

46. Hendley, R.J.; Drew, N.S.; Wood, A.M.; and Beale, R.: Narcissus: Visualizing Information, Proc. IEEE InfoVis '95, New York, 1995, pp. 90-96.

47. Herman, I.; Melançon, G.; Delest, M.: Tree Visualization and Navigation Clues for Information Visualization. Computer Graphics Forum, Vol. 17, 1998, pp. 153-165.

48. Hinneburg, A.; D. Keim, and M. Wawryniuk. HD-Eye: Visual Mining of High-Dimensional Data. IEEE Computer Graphics and Applications, 19(5), 1999.

49. Hoffmann, P.; Grinstein, G.; Marx, K.; Grosse, I.; and Stanley, E.. DANN Visual And Analytic Data Mining, Proc. Vis '97, 1997, IEEE Visualization, Phoenix, pp. 437-441.

50. Hofmann, H.; Siebes, A.; and Wilhelm, A.: Visualizing association rules with interactive mosaic plots, SIGKDD Int. Conf. On Knowledge Discovery & Data Mining (KDD 2000), Boston, MA, 2000.

51. Jerding, D.F.; Stasko, J.T.: The Information Mural: A Technique for Displaying and Navigating Large Information Spaces, TR GIT-GVU-97-24, Georgia Institute of Technology, 1997

52. Keahey, T.A.: The Generalized Detail-In-Context Problem. Proc. Vis'98, IEEE Visualization, Research Triangle Park NC, 1998.

53. Keahey, T.A.: The Nonlinear Magnification Home Page, http://www.cs.indiana.edu/hyplan/tkeahey/research/nlm/nlm.html, October 2000.

54. Keim, D.A.: Designing pixel-oriented visualization techniques: Theory and applications. Transactions on Visualization and Computer Graphics, 6(1):59–78, Jan–Mar 2000.

55. Keim, D.A.; and Kriegel, H.P.: VisDB: Database exploration using multidimensional visualization, Computer Graphics & Applications, 6:40–49, Sept. 1994.

56. Keim, D.A.; Kriegel, H.P.; and M. Ankerst, M.: Recursive Pattern: A technique for visualizing very large amounts of data, In Proc. Visualization 95, Atlanta, GA, pages 279–286, 1995.

57. Keim, D.A.; and Ward, M.: Visual Data Mining, in: Berthold, M.; and Hand, D.J. (eds): Intelligent Data Analysis - An Introduction, Second Edition, Springer-Verlag, 2002.

58. Kohonen, T.: Self-organizing maps, 2nd edition, Springer, Berlin, 1997.

59. Koutsofios, E.E.: Visualizing Large-Scale Telecommunication Networks and Services, Proceedings Vis'1999, IEEE Visualization, San Francisco, 1999.

60. Kreuseler, M.; Lopez, N.; Schumann, H.: A Scalable Framework for information Visualization, Proceedings InfoVis'2000, Salt Lake City, 2000, pp.27-36.

61. Kruskal, J.N.; and Wish, M.: Multidimensional Scaling, Sage, 1978.

62. Lamping, J. et al.: A focus+context technique based on hyperbolic geometry for viewing large hierarchies. ACM Proceedings CHI'95, Denver, May, 1995, pp. 401-408.

63. Lantin, M.L.; Carpendale, M.S.T.: Supporting Detail-in-Context for the DNA Representation, H-Curves. IEEE Computer Graphics and Applications, 443-446, 1998.

64. Leung, Y.K.; Apperley, M.D.. A Review and Taxonomy of Distortion – Oriented Presentation Techniques. ACM Transactions on Computer-Human-Interaction, Vol. 1, 2, 1994, pp. 126-160.

65. Levkowitz, H.: Color icons: Merging color and texture perception for integrated visualization of multiple parameters. In Proc. Visualization 91, San Diego, CA, pages 22–25, 1991.

66. Mackinlay, J.: Automating the Design of Graphical Presentations of Relational Information. ACM Transactions on Graphics, Vol. 5, No. 2, April 1986.

67. Mackinlay, J.D.; Robertson, G.G.; Card, S.K. The Perspective Wall: Detail and Context Smoothly Integrated. Proc. ACM Conf. On Human Factors in Computing Systems (CHI' 91), New York, NY, 1991, pp. 173-180.

68. Mehta, M.; Agrawal, R.; and J. Rissanen. SLIQ: A fast scalable classifier for data mining. Conf. on Extending Database Technology (EDBT), Avignon, France, 1996.

69. Munzner, T.: Exploring Large Graphs in 3D Hyperbolic Space. IEEE Computer Graphics Vol. 18, 1998.

70. Munzner, T.; and Burchard, P.: Visualizing the structure of the world wide web in 3D hyperbolic space, In Proc. VRML '95 Symp, San Diego, CA, pages 33–38, 1995.

71. Overbye, T.J.; Weber, J.D.: New Methods for the Visualization of Electric Power System Information, Proceedings IEEE InfoVis'2000, Salt Lake City, Oct.2000, pp.131-136.

72. Pacific Northwest Laboratory: ThemeView Showcase, http://showcase.pnl.gov/show?MODULE&2, 2001

73. Pearson, W.R; and Lipman, D.J.: Improved tools for biological sequence comparison, Proc. Natl. Acad. Sci. USA, 85, 1988, pp. 2444-2448.

74. Perlin, K.; and Fox, D.: Pad: An alternative approach to the computer interface. In Proc. SIGGRAPH, Anaheim, CA, pages 57–64, 1993.

75. Pickett, R.M.; and Grinstein, G.G.: Iconographic displays for visualizing multidimensional data, In Proc. IEEE Conf. on Systems, Man and Cybernetics, IEEE Press, Piscataway, NJ, pages 514–519, 1988.

76. Quinlan, J.R.: Induction of decision trees, Machine Learning, pages 81–106, 1986.

77. Quinlan, J.R.: C4.5: Programs For Machine Learning. Morgan Kaufmann, Los Altos, CA, 1993.

78. Rao, R.; and Card, S.K.: 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.

79. Robertson, G.G; Mackinlay, J.D.; Card, S.K.: Cone Trees: Animated 3D Visualizations of Hierarchical Information. Proceedings ACM CHI International Conference on Human Factors in Computing (CHI'91), 1991, pp. 189-194.

80. Sarkar, M; Brown, M.H.: Graphical FishEye Views of Graphs. ACM CHI'92, 1992, pp. 83-91.

81. Sarkar M.; and Brown, M.: Graphical fisheye views, Communications of the ACM, 37(12):73–84, 1994.

82. Sakar, M.; Snibbe, S.S.; Tversky, O.; Reiss, S.P.: Stretching the rubber sheet, Proceedings ACM Symposium on User Interface Software and Technology, 1993.

83. Schaffer, D.; Zuo, Z.; Bartram, L.; Dill, J.; Dubs, S.; Greenberg, S., and Roseman, M.: 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.

84. Schumann, H.; Müller, W.: Visualisierung – Grundlagen und allgemeine Methoden. Springer Verlag, ISBN 3-540-64944-1, Heidelberg, 2000.

85. Senay, H.; and Ignatius, E.: A Knowledge-Based System for Visualization Design, IEEE Computer Graphics and Applications 14, 6 (1994), 36--47.

86. Shafer, J.; Agrawal, R.; and Mehta, M.: SPRINT: A scalable parallel classifier for data mining, Conf. on Very Large Databases, 1996.

87. Shneiderman, B.: Tree Visualization with Treemaps: A 2D Space Filling Approach. ACM Transactions on Graphics, Vol.11, No. 1, 1992, pp. 92-99.

88. Spence, R.: Information Visualization. Addison-Wesley, Harlow, 2001.

89. Spence, R.; and Apperley, M.: Data base navigation: An office environment for the professional, Behaviour and Information Technology, 1(1):43–54, 1982.

90. Spoerri., A.: Infocrystal: A visual tool for information retrieval. In Proc. Visualization '93, San Jose, CA, pages 150–157, 1993.

91. Stasko, J.; Domingul, J.; Brown, M.H.; Price, B.A. (eds.): Software Visualization. MIT Press, Cambridge, 1997.

92. Stasko, J.; Zhang, E.: "Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations, Proc. IEEE Information Visualization 2000, Salt Lake City, UT, Oct. 2000, pp. 57-65.

93. Stolte, C.; Tang, D.; and Hanrahan, P.: Polaris: A system for query, analysis and visualization of multi-dimensional relational databases. Transactions on Visualization and Computer Graphics, 2001.

94. Swayne, D.F.; Cook, D.; and Buja, A.: User's Manual for XGobi: A Dynamic Graphics Program for Data Analysis. Bellcore Technical Memorandum, 1992.

95. Theisel, H.; and Kreuseler, M.: An Enhanced Spring Model for Information Visualization, Computer Graphics Forum, 17(3), Sep.1998, pp. 335-343.

96. Tominski, C.: Visualisierungstechniken zur Analyse zeitlicher Verläufe auf Karten, Master Thesis, University of Rostock, Department of Computer Science, 2002

97. Tufte, E.R.: The Visual Display of Quantitative Information, Graphics Press, Cheshire, Connecticut, 1983.

98. Utgoff, P.E.: Incremental induction of decision trees. Machine Learning, 4:161–186, 1989.

99. Utgoff, P.E.; Berkman, N.C.; and Clouse, J.A.: Decision tree induction based on e_cient tree restructuring. Machine Learning, 29:5–44, 1997.

100. van Wijk, J.J.; van de Wetering, H.: Cushion Treemaps: Visualization of Hierarchical Information, Proceedings InfoVis'1999, San Francisco, Oct. 2000, pp.73-78.

101. van Wijk, J.J.; van Liere, R.D.: Hyperslices, Proceedings: IEEE InfoVis'93, Oct. 1993, pp.119-125.

102. Velleman, P.F.: Data Desk 4.2: Data Description. Data Desk, Ithaca, NY, 1992, 1992.

103. Voigt, D.: WWW - basierte Darstellung komplexer Informationsstrukturen, Master Thesis, University of Rostock, Department of Computer Science, 2001

104. Ware, C.: Information Visualization. Morgan Kaufmann Publishers, San Francisco, 2000.

105. Ward, M.O.: Xmdvtool: Integrating multiple methods for visualizing multivariate data. In Proc. Visualization 94, Washington, DC, pages 326–336, 1994.

106. Weber, M; Alexa, M; Müller, W.: Visualizing Time-Series on Spirals, Proc: IEEE Information Visualization 2001, San Diego, USA, October 2001, pp. 7-14.

107. Wendt, S.: Nichtphysikalische Grundlagen der Informationstechnik, Springer-Verlag, Berlin, 1991 (in German).

108. Westphal, C.; Blaxton, T.: Data Mining Solutions, John Wiley & Sons, New York, 1998.

109. Wise, J.A. et. al.: Visualizing the Non-Visual: Spatial Analysis and Interaction with Information from Text Documents, IEEE Proc. of InfoVis '95, pp. 51-58.

110. Wong, P.C.: Visual Data Mining, IEEE CG&A, September/October 1999, pp. 20-21.

111. Wright, W.: Information Animation Applications in the Capital Markets, Proc IEEE InfoVis '95, New York, pp. 19-25.

112. Wilhelm, A.; Unwin, A.; and Theus, M.: Software for interactive statistical graphics - a review, In Proc. Int. Softstat 95 Conf., Heidelberg, Germany, 1995.

113. Yan. L.: 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.

114. Young, F.W.; Bann, C.M.: ViSta: A Visual Statistics System. In: Stine, R.A. & Fox, J. (Eds.), Statistical Computing Environments for Social Research. Sage Publications.1996, pp. 207-236.