

Information Visualization: Scope, Techniques and Opportunities for Geovisualization*

Daniel A. Keim, Christian Panse, Mike Sips
University of Konstanz, Germany

{keim, panse, sips}@informatik.uni-konstanz.de

June 27, 2003

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 article, we provide an overview of information visualization and visual data mining techniques, and illustrate them using a few examples. We show that an application of information visualization methods provides new ways of analyzing geography – related data.

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 2 Exabyte (= 2 Million Terabytes) of data are generated, of which a large portion is available in digital form (growth rate is about 50%). 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. Almost all transactions of every day life, such as purchases made with credit card, web pages visited or telephone calls are recorded by computers. In many application domains it might also be useful that much of the data include geospatial referencing. For example, credit card purchase transactions include both the address of the place of purchase and of the purchaser, telephone records include addresses and sometimes coordinates or at least cell phone zones, data from satellite remote sensed data contain coordinates or other geographic indexing, census data and other government statistics contain addresses and/or indexes for places, and information about property ownership contains addresses, relative location info, and sometimes coordinates etc. This data is collected because people believe that it is a potential source of valuable information, providing a competitive advantage (at some point) to its holders. Usually many parameters are recorded, resulting in data with a high dimensionality. 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

*portions of this article have been previously published in [54]

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 much less understanding of complex mathematical or statistical algorithms or parameters than other methods, visualization can provide a qualitative overview of the data, allowing data phenomena to be isolated for further quantitative analysis. Visual data exploration do include some underlying data processing, dimension reduction, re-mapping from one space to another, etc. which typically does require some basic understanding of the underlying algorithm to understand it properly. As a result, visual data exploration can lead to quicker and often superior 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 [84]). 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.

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 [18, 98, 85, 82]. The techniques can be classified based on three criteria (see Figure 1): The data to be visualized, the visualization technique, and the interaction technique used. The **data type to be visualized** [84] 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; and 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; 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; and 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 as well as 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.

3 Data Type to be Visualized

In information visualization, the data usually comprise 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-

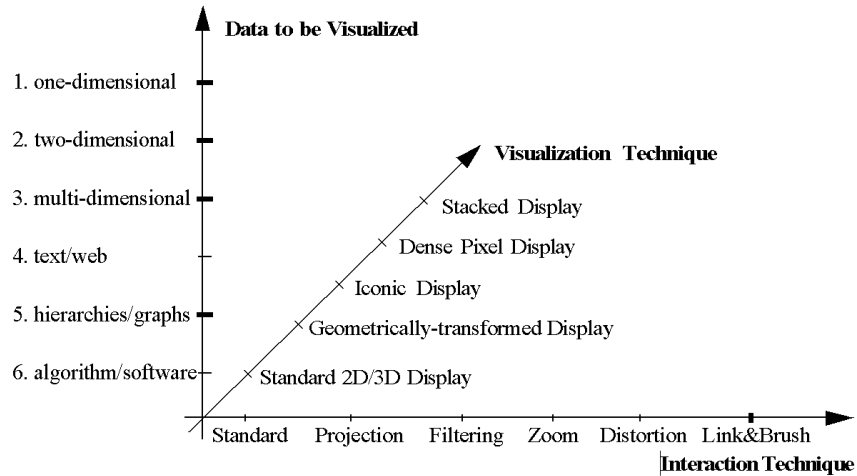


Figure 1: Classification of Information Visualization Techniques

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 grid dimensions and the dimensions that may have arbitrary values.

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 Figure 4) or the time series of news data used in the ThemeRiver [43].

Two-dimensional data

Two-dimensional data usually has two dense dimensions. This type of data can be represented by mapping the associated data values to a color and displaying the color of each data value in its (x, y) location on the screen. A typical example are geometry-related planar data in the euclidian plane. In general, planar data defines quantities of distances, or distances and angles, which define the position of a point on a reference plane to which the surface of a 3D object has been projected. A typical example is geographical data, where the two distinct dimensions are longitude and latitude. Longitude and Latitude describe locations on a 3D surface and some transformation is required to project the relationships between locations specified in this way on a plane. Besides, depending upon the cartography used, various characteristics of the relationships between locations are either preserved or lost. After the projection, the geographical data can be stored as two-dimensional data with x/y-dimensions. X-Y-plots are a typical method for showing two-dimensional data and maps are a special type of X-Y-plot for showing geographical data. Although it seems easy to deal with temporal or geographic data on 2D devices, 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. There are several approaches to draw upon with dense geographic data already in common use, such as *Gridfit* [55] (see Figure 6 in section 5.2) and *PixelMap* [61] (see Figure 7 in section 5.3).

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, such as parallel coordinates plots [48] (see Figure 12(a)). A very different method for data with geospatial attributes are Scatterplots and Scenes [37].

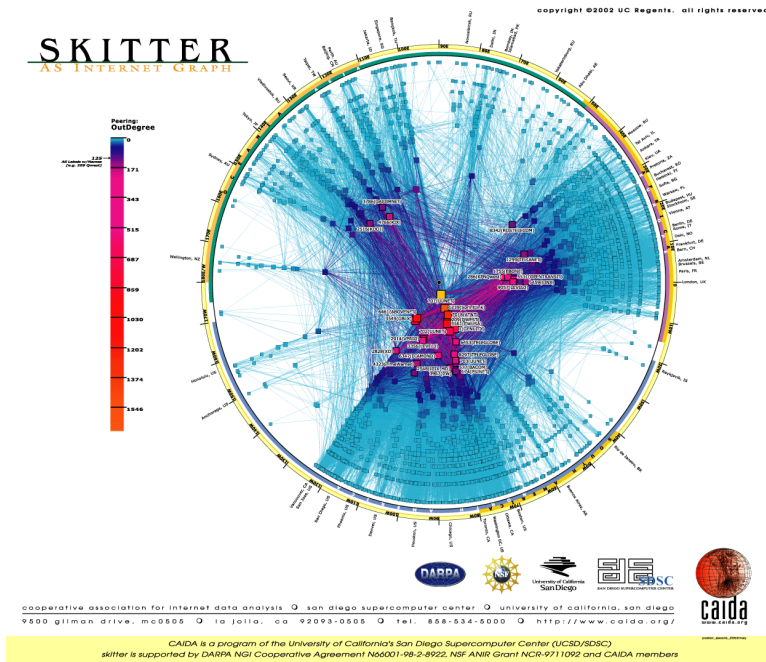


Figure 2: Skitter Graph Internet Map, CAIDA (Cooperative Association for Internet Data Analysis) ©2002 UC Regents. Courtesy University of California.

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 which is often combined with a principal component analysis or multidimensional scaling to reduce the dimensionality to two or three.

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 visualization techniques can be found in [23], an overview of web visualization techniques is presented in [29] and an overview book on all aspects related to graph drawing is [11].

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 [94, 89]. Few examples of visualization (and visually-enabled steering) of process models that depict geographic processes are the AGP-System for ocean flow model visualization [49], and visualization methods/simulation steering for the 3D turbulence model of Lake Erie [72].

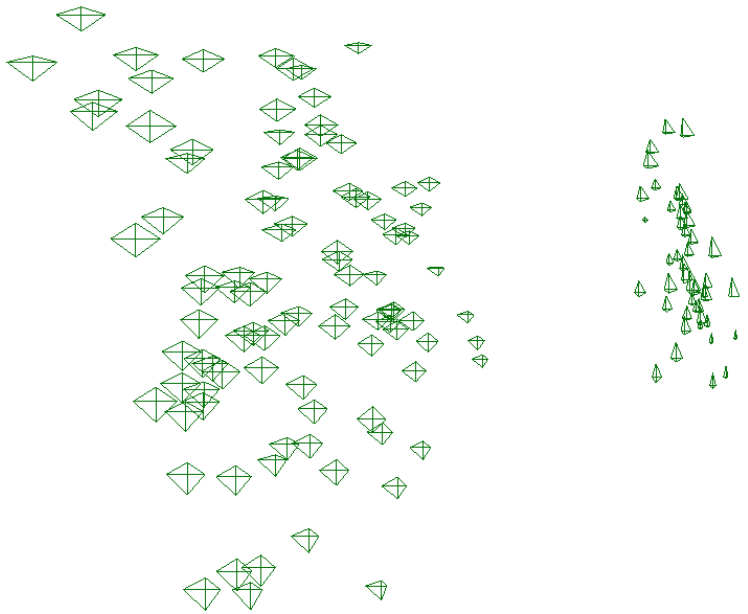


Figure 3: The iris data set, displayed using star glyphs positioned based on the first two principal components (from XmdvTool [97]);

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.

4.1 Geometrically-Transformed Displays

Geometrically transformed display techniques aim at finding “interesting” transformations of multidimensional data sets. This class of geometric display methods includes techniques from exploratory statistics such as scatterplot matrices [6, 25] and techniques that can be subsumed under the term “projection pursuit” [46]. Other geometric projection techniques include Prosection Views [36, 87], Hyperslice [95], and the well-known Parallel Coordinates visualization technique [48]. 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 12(a)). A geographic extension of the Parallel Coordinates Technique is to build them in 3D, with maps as planes orthogonal to the otherwise flat 2D Parallel Coordinates [30]

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 [24], needle icons [1, 53], star icons [97], stick figure icons [76], color icons [67, 56], or TileBars [44], 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 3 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.

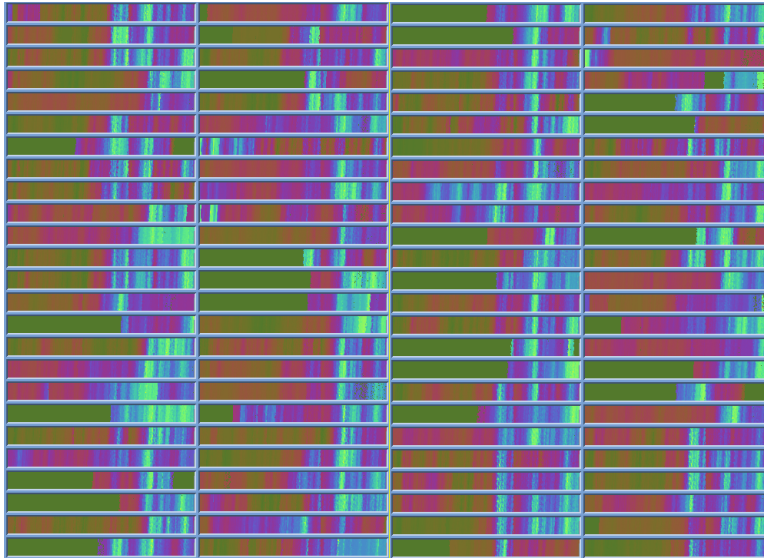


Figure 4: Dense Pixel Displays: Recursive Pattern Technique showing 50 stocks in the Frankfurt Allgemeine Zeitung (Frankfurt Stock Index Jan 1975 - April 1995). The technique map each stock value to a colored pixel; high values correspond to bright colors. ©IEEE

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. More precisely, dense pixel displays divide the screen into multiple subwindows. For data sets with m dimensions (attributes), the screen is partitioned into m subwindows, one for each of the dimensions. 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). Although, it seems to be easy to create dense pixel displays, there are some important questions to be taken into account.

Color Mapping

First, finding a path through a color space that maximizes the numbers of just noticeable difference (JND), but at the same time, is intuitive for the application domain, however, is a difficult task. The advantage of color over gray scales is that the number of JND's is much higher.

Pixel Arrangement

If each data value is represented by one pixel, the main question is how to arrange the pixels on the screen within the subwindows. Note, that only a good arrangement due to the density of the pixel display will allow a discovery of clusters, local correlations, dependencies, other interesting relationships among the dimensions, and hot spots. For the arrangement of pixels, we have to distinguish between data sets that have a natural ordering of the data objects, such as time series, and data sets without inherent ordering, such as in query result sets. Even if the data has a natural ordering to one attribute, there are many arranging possibilities. One straightforward possibility is to arrange the data items from left to right in a line-by-line fashion. Another possibility is to arrange the data items top-down in a column-by-column fashion. If these arrangements are done pixel wise, in general, the resulting visualizations do not provide useful results. More useful are techniques which provide a better clustering of closely related data items and allow the user to influence the arrangement of the data. Techniques which support the clustering properties are screen filling curves. Another technique which supports clustering is the Recursive Pattern technique [58]. 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. 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 4, an example recursive of a pattern

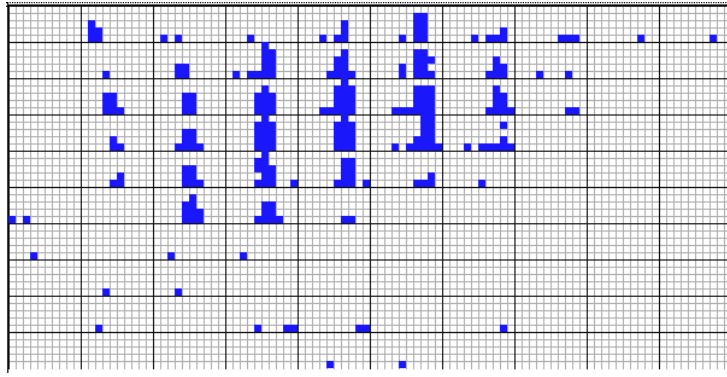


Figure 5: Dimensional Stacking visualization of drill hole mining data with longitude and latitude mapped to the outer x-,y-axes and one grade and depth mapped to the inner x-,y-axes (used by permission of M. Ward, Worcester Polytechnic Institute ©IEEE)

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).

Shape of Subwindows

The next important question is whether there exists an alternative to the regular partition of the screen into rectangular subwindows. The rectangular shape of the subwindows allows a good screen usage, but on the other hand, the rectangular shape leads to a dispersal of the pixels belonging to one data object over the whole screen. Especially for data sets with many dimensions, the subwindows of each dimension is rather far apart, which makes it very difficult to detect clusters, correlations, and interesting patterns. An alternative shape of the subwindows is the Circle Segments technique [8]. The basic idea of these technique is to display the data dimensions as segments of a circle.

Ordering of Dimensions

Finally, the last question to consider is the ordering of the dimensions. This problem is actually not just a problem of dense pixel displays, but a more general problem which arises for a number of other visualization techniques, such as the parallel coordinate plots, as well. The basic problem is that the data dimensions have to be arranged in some one- or two dimensional ordering on the screen. The ordering of dimensions, however, has a major impact on the expressiveness of the visualization. If a different ordering of the dimensions is chosen, the resulting visualization becomes completely different and leads to different interpretations. More details about designing pixel-oriented visualization techniques can be found in [53, 57] and all techniques have been implemented in the *VisDB*-System [56].

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* [65]. 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 5. Other examples of stacked display techniques include *Worlds-within-Worlds* [33], *Treemap* [50, 83], and *Cone Trees* [79].

5 Visual Data Mining on Geospatial Data

Geospatial data are relevant to a large number of applications. Examples include weather measurements such as temperature, rainfall, wind-speed, etc. measured at a large number of locations, use of connecting nodes in telephone business, load of a large number of

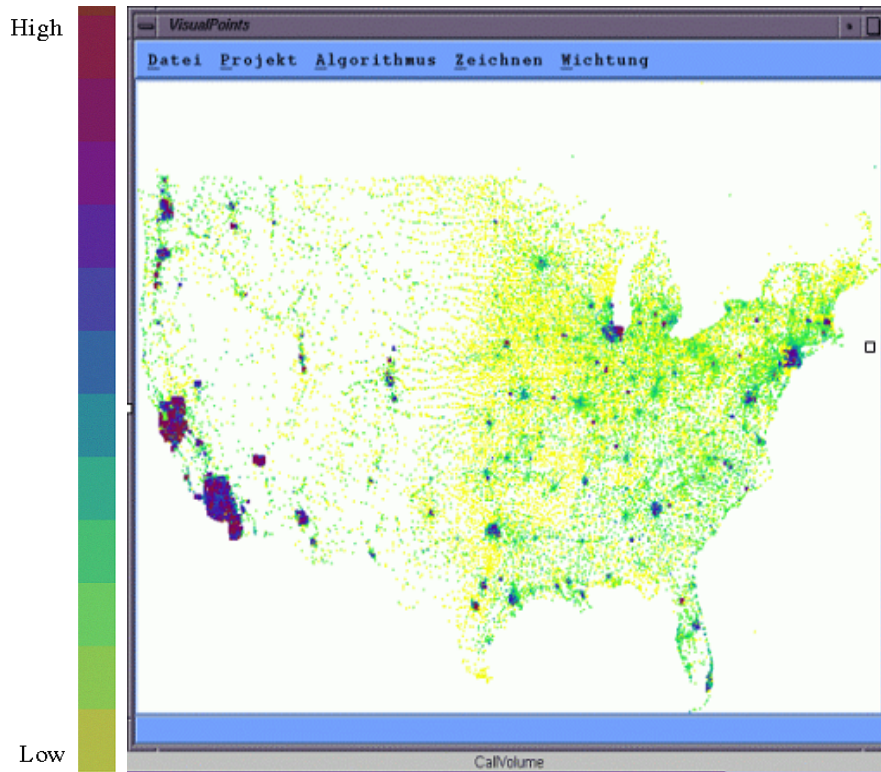


Figure 6: The picture shows the call volume of the AT&T customers at one specified time (midnight EST). Each pixel represents one of 22.000 telephone switches. A uni-color map is used to show the call volume.

internet nodes at different locations, air pollution in cities etc. Nowadays, there exist a large number of applications, in which it is important to analyze relationships that involve geographic location. Examples include global climate modeling (measurements such as temperature, rainfall, and wind-speed), environmental records, customer analysis, telephone calls, credit card payments, and crime data. Spatial data mining is the branch of data mining that deals with spatial (location) data. However, to analyze the huge amounts (usually tera-bytes) of geospatial data obtained from large databases such as credit card payments, telephone calls, environmental records, etc. it is almost impossible for users to examine the spatial data sets in detail and extract interesting knowledge or general characteristics. Automated data mining algorithms are indispensable for analyzing large geospatial data sets, but often fall short of providing completely satisfactory results. Usually, interactive data mining based on a synthesis of automatic and visual data mining techniques usually does not only yields better results, but also a higher degree of user satisfaction and confidence in the findings [42]. Although automatic approaches have been developed for mining geospatial data [42], they are often no better than simple visualizations of the geospatial data on a map.

5.1 Visualization Strategy

The visualization strategy for geospatial data is straightforward. The geospatial data points described by longitude and latitude are displayed on the 2D euclidian plain using a 2D projection. The two euclidian plain dimensions x and y are directly mapped to the two physical screen dimensions. The resulting visualization depends on the spatial **dimension** or **extent** of the described phenomena and objects. A nice overview can be found in [62]. Since the geospatial locations of the data are not uniformly distributed in a plain, however, the display will usually be sparsely populated in some regions while in other regions of the display a high degree of overplotting occurs. There are several approaches to cope with dense geospatial data [39]. One widely used method is 2.5D visualization showing data points aggregated up to map regions. This technique is commercially available in systems such as VisualInsight's In3D [2] and ESRI's ArcView [32]. An alternative that shows more detail is a visualization of individual data points as bars on a map. This technique is embodied in systems such as SGI's MineSet [45] and AT&T's Swift 3D [52]. A problem here is that a large number of data points are plotted at the same position, and therefore only a small portion of the data is actually displayed. Moreover, due to occlusion in 3D, a significant fraction of the data may not be visible unless the viewpoint is changed. Other visualization techniques that have been developed to support specific visualization and data mining tasks will be described in the following section.

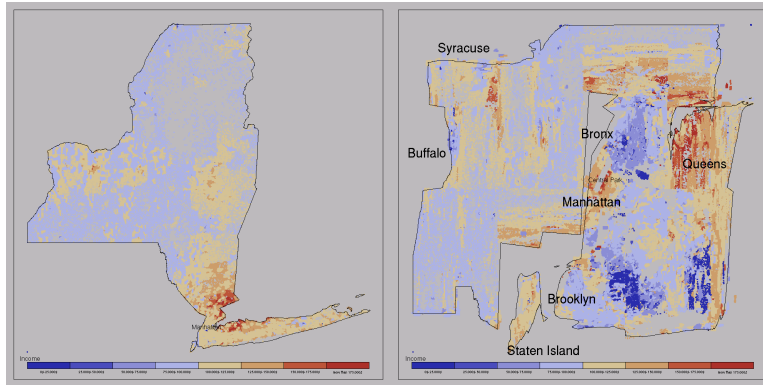


Figure 7: *Comparison Traditional Map versus PixelMap* - New York State - Year 1999 Median Household Income. This map displays cluster regions e.g. on the East side of Central Park in Manhattan, where inhabitants with high income live, or on the right side of Brooklyn, where inhabitants with low income live.

5.2 The *VisualPoints*-System

The *VisualPoints*-System solves the problem of overplotting pixel using the Gridfit algorithm [55]. The basic idea of the Gridfit algorithm is to hierarchically partition the data space. In each step, the data set is partitioned into four subsets containing the data points which belong to four equally-sized subregions. Since the data points may not fit into the four equally size subregions, we have to determine a new extent of the four subregions (without changing the four subsets of data points) such that the data points of each subset can be visualized in the corresponding subregion. For an efficient implementation of the algorithm, a quadtree-like data structure is used to manage the required information and to support the recursive partitioning process. The partitioning process works as follows. Starting with the root of the quadtree, in each step the data space is partitioned into 4 subregions. The goal of the partitioning is that the area occupied by each of the subregions (in pixels) is larger than the number of pixels to be placed within these subregions. The partitioning process works as follows. Starting with the root of the quadtree, in each step the data space is partitioned into four subregions. The partitioning is made such that the area occupied by each of the subregions (in pixels) is larger than the number of pixels belonging to the corresponding subregion. A problem of *VisualPoints* is that in areas with high overlap, the repositioning depends on the ordering of the points in the database. That is, the first data item found in the database is placed at its correct position, and subsequent overlapping data points are moved to nearby free positions, and so locally appear quasi-random in their placement.

5.3 *Fast-PixelMap*-Technique

A problem of the Gridfit-algorithm is that in areas with high overlap, the repositioning depends on the ordering of the points in the database. That is, the first data item found in the database is placed at its correct position, and subsequent overlapping data points are moved to nearby free positions, and so locally appear quasi-random in their placement. The *PixelMap*-Technique solves the problem of displaying dense point sets on maps, by combining clustering and visualization techniques [61].

First, the *Fast-PixelMap* algorithm approximates a two-dimensional kernel-density-estimation in the two geographical dimensions performing a recursive partitioning of the dataset and the 2D screen space by using split operations according to the geographical parameters of the data points and the extensions of the 2D screen space. The goal is (a) to find areas with density in the two geographical dimensions and (b) to allocate enough amount of pixels on the screen to place all data points of dense regions at unique positions close to each other. The top-down partitioning of the dataset and 2D screen space results in distortion of certain map regions. That means, however, virtually empty areas will be shrinking and dense areas will be expanding to achieve pixel coherence. For an efficient partitioning of the dataset and the 2D screen space and an efficient scaling to new boundaries, a new data structure called *Fast-PixelMap* is used. The *Fast-PixelMap* data structure is a combination of a gridfile and a quadtree which realizes the split operations in the data and the 2D screen space. *Fast-PixelMap* data structure enables an efficient determination of the old (boundaries of the gridfile partition in the dataset) and the new boundaries (boundaries of the quadtree partition in the 2D screen space) of each partition. The old and the new boundaries determine the local rescaling of certain map regions. More precisely, all data points within the old boundaries will be relocated to the new positions within the new boundaries. The rescaling reduces the size of virtually empty regions and unleashes unused pixels for dense regions.

Second, the *Fast-PixelMap* algorithm approximates a three-dimensional kernel-density-estimation-based clustering in the three dimensions performing an array based clustering for each dataset partition. After rescaling of all data points to the new boundaries, the iterative positioning of data points (pixel placement step), starting with the densest regions and within the dense regions the

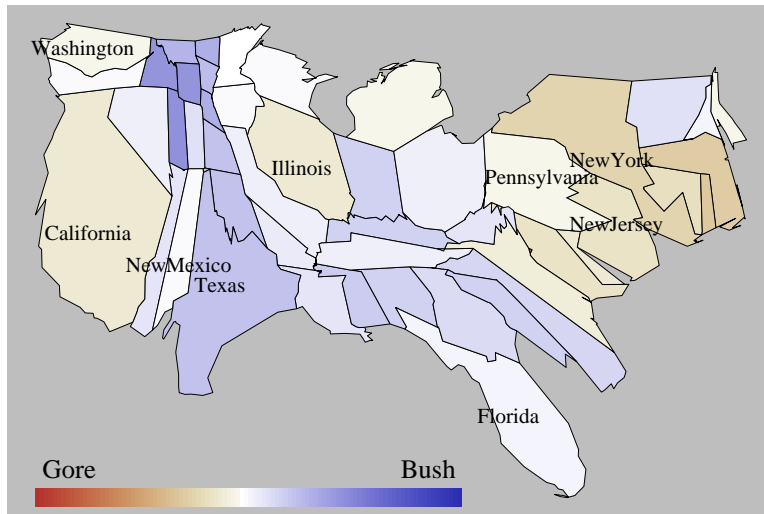


Figure 8: The Figure displays the U.S. state population cartogram with the presidential election result of 2000. The area of the states in the cartograms corresponds to the electoral voters the color corresponds to the percentage of the vote. A bipolar colormap depicts which candidate has won each state.

smallest cluster is chosen first. To determine the placement sequence, we sort all final gridfile partitions (leaves of the Fast-PixelMap data structure) according to the number of data points, they contain. The clustering is a crucial preprocessing step to making important information visible and achieving pixel coherence¹(with respect to the selected statistical parameter).

The final step of the pixel placement is a sophisticated algorithm which places all data points of a gridfile partition to pixels on the output map in order to provide visualizations which are as position, distance, and cluster-preserving as possible.

5.4 Cartogram Techniques

A cartogram can be seen as a generalization of a familiar land-covering choropleth map. In this interpretation, an arbitrary parameter vector gives the intended sizes of the cartogram’s regions, so an ordinary map is simply a cartogram with sizes proportional to land area. In addition to the classical applications mentioned above, a key motivation for cartograms as a general information visualization technique is to have a method for trading off shape and area adjustments. For example in a conventional choropleth map, high values are often concentrated in highly populated areas, and low values may be spread out across sparsely populated areas. Such maps therefore tend to highlight patterns in less dense areas where few people live. In contrast, cartograms display areas in relation to an additional parameter, such as population. Patterns may then be displayed in proportion to this parameter (e.g. the number of people involved) instead of the raw size of the area involved. Example applications in the literature include population demographics [93, 77], election results [63], and epidemiology [41]. Because cartograms are difficult to make by hand, the study of automated methods is of interest [28].

One of the latest cartogram generation algorithm is the *CartoDraw*-algorithm which was recently proposed as a practical approach to cartogram generation [60, 59]. The basic idea of *CartoDraw* is to incrementally reposition the vertices of the map’s polygons by means of scanlines. Local changes are applied if they reduce total area error without introducing excessive shape error. Scanlines may be determined automatically, or entered interactively. The main search loop runs over a given set of scanlines. For each, it computes a candidate transformation of the polygons, and checks it for topology and shape preservation. If the candidate passes the tests, it is made persistent; otherwise it is discarded. The scanline processing order depends on their potential for reducing area error. The algorithm runs until the area error improvement over all scanlines falls below a given threshold. A result of the *CartoDraw* algorithm can be seen in Figure 8.

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

¹Pixel coherence means similarity of adjacent pixels, which makes small pixel clusters perceivable.

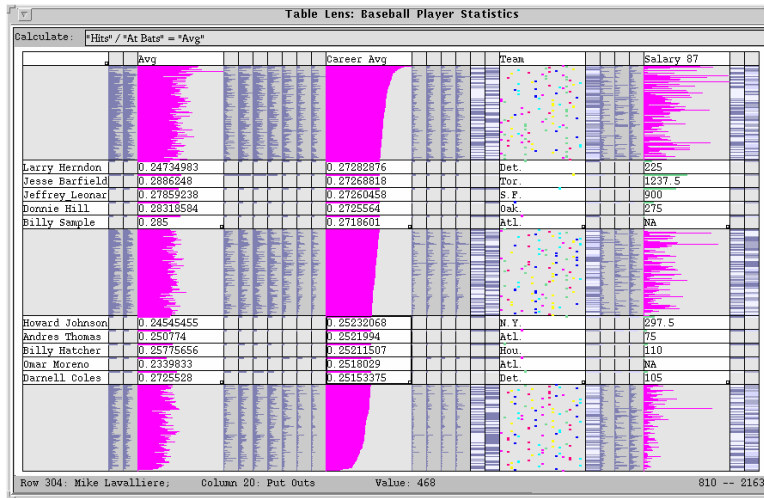


Figure 9: Table Lens showing a table of baseball players performance/classification statistics for 1986. (used by permission of R. Rao, Xerox PARC ©ACM)

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.

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 [10] 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 accordance with 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 [91, 17], XLispStat [92], and ExplorN [22].

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, but on the other hand, these tools offer many advantages over traditional controls. 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, 34]. The basic idea of Magic Lens is to use a tool similar to a magnifying glass to support the filtering of the data directly in the visualization. The data under the magnifying glass is processed by the filter, and the result is displayed differently from the remaining data set. Users may also use several lenses with different filters, e.g., magnify, then apply a blur, by carefully stacking multiple lenses, which work by applying their effects from back to front. Each lens acts as a filter that screens on some attribute of the data. The filter function only modifies the view only within the lense window. When the lenses overlap, their filters are combined. Other examples of interactive filtering techniques and tools are InfoCrystal [88], Dynamic Queries [3, 31, 40], Polaris [90], Linked Micromap Plots, and Conditioned Choropleth Maps [20, 21] (see also Figure 10). Also, a type of interactive filtering using a parallel coordinate plot linked to a map is illustrated in [70] and in [38].

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, allowing a variable display of the data at different resolutions. Zooming does not only mean displaying the data objects larger, but

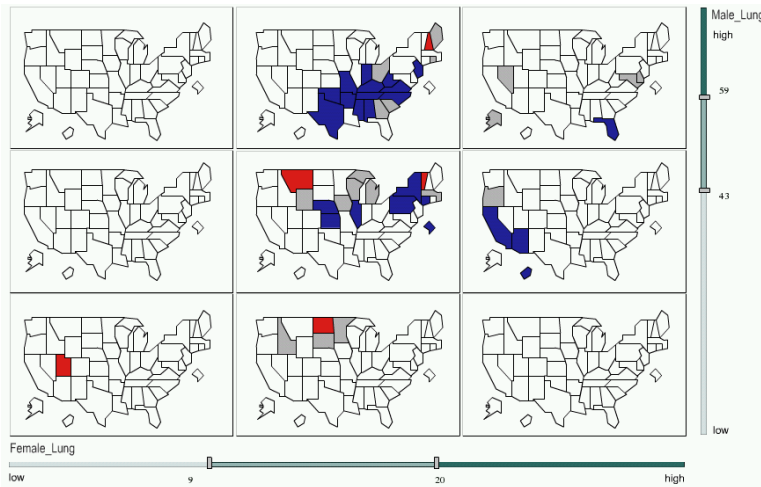


Figure 10: A conditioned Choropleth Map of Prostate Data of the USA. The mapped prostate data is conditioned by male and female lung cancer. (made by using Conditioned Choropleth Mapping System [69] developed under the project *Conditioned Choropleth Maps: Dynamic Multivariate Representations of Statistical Data* supervised by Alan MacEachren)

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 large tabular data sets is difficult if the data is displayed in textual form. The basic idea of Table Lens 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. 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 9 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++ [75, 14] [15], IVEE/Spotfire [4], and DataSpace [9]. A comparison of fisheye and zooming techniques can be found in [81].

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 [68]. An overview of distortion techniques is provided in [66, 19]. Examples of distortion techniques include Bifocal Displays [86], Perspective Wall [71], Graphical Fisheye Views [35, 80], Hyperbolic Visualization [64, 74], and Hyperbox [5]. Some of the uses of distortion with map displays that combines zooming, distortion and filtering can be found in [51]. Figure 11 shows the effect of distorting part of a familiar land-covering map to display more detail 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 to communicate 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 also 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.

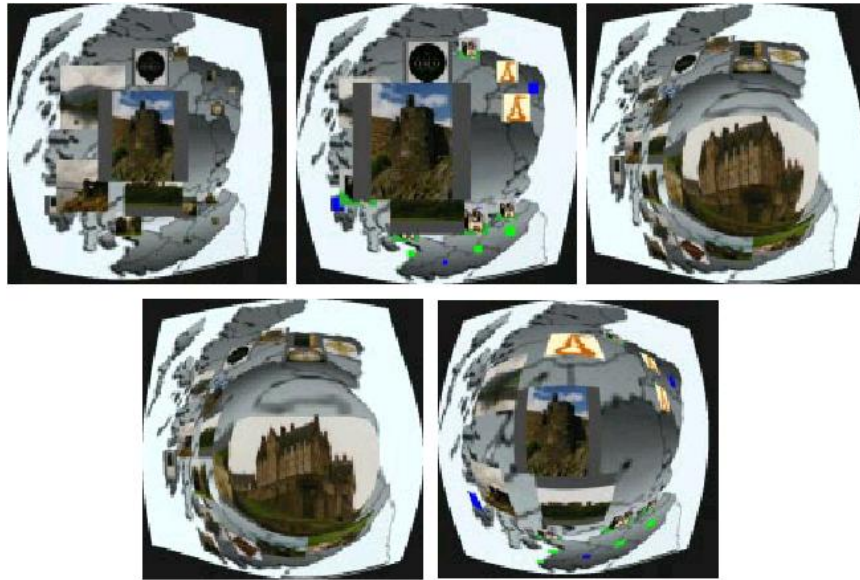


Figure 11: Combination of zooming, distortion, and filtering techniques in the area of geographic visualization (see also [51]).

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 [90] and the Scalable Framework [68]. Other tools and systems include S-Plus [12], XGobi [91, 13], XmdvTool [97] (see Figure 12), and DataDesk [96, 99]. Links between XGobi and GIS for interactive linking and brushing (as well as filtering) are discussed in [26, 7, 73].

Alternative Classification from GI-Science

Crampton [27] proposes an alternative classification of interaction techniques developed in GI-Science. The interactivity types are placed in the framework of geographic visualization (GVis) in order to extend the GVis emphasis on exploratory, interactive and private functions of spatial displays. Four categories of interactivity are proposed: (1) the Data; (2) the Data Representation; (3) the Temporal Dimension; and (4) Contextualizing Interaction. The following benefits of the classification can be observed. First, interactivity types can be combined to build an interactive environment. Second, the typology allows cartographers to compare and critique different mapping and GIS environments and gives cartography educators and students a mechanism for understanding the different types of interactivity, as well as a set of concepts for imagining and creating new interactive environments. Third, a typology of interactivity gives interface designers a mechanism with which to identify needs and measure interface effectiveness.

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 significant 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] ABELLO, J., AND KORN, J. Mgv: A system for visualizing massive multi-digraphs. *Transactions on Visualization and Computer Graphics* (2001).

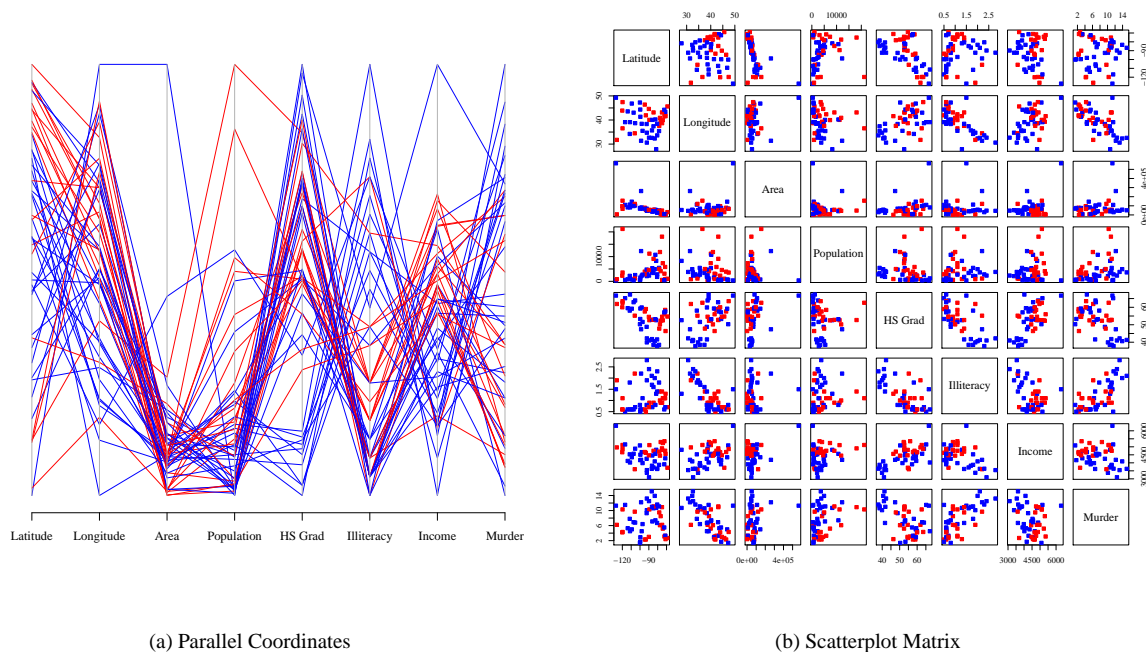


Figure 12: The Parallel Coordinates Plots 12(a) and the Scatterplot Matrix 12(b) display the US-Census data of the 50 states. A linked brushing between these multivariate visualization techniques is used. The plots are linked and brushed by color which represents the state wise winner party of 2000 US presidential election (from R [47]). The plots show for example an inverse correlation between High School Grad and Illiteracy as well as that in states with a low or a high HS Grade the republican party has won the votes.

[2] ADVIZOR SOLUTIONS, I. Visual insight in3d. <http://www.advizorsolutions.com/>, Feb. 2003.

[3] 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 (1994)*, pp. 313–317.

[4] AHLBERG, C., AND WISTRAND, E. Ivey: An information visualization and exploration environment. In *Proc. Int. Symp. on Information Visualization, Atlanta, GA (1995)*, pp. 66–73.

[5] ALPERN, B., AND CARTER, L. Hyperbox. In *Proc. Visualization '91, San Diego, CA (1991)*, pp. 133–139.

[6] ANDREWS, D. F. Plots of high-dimensional data. *Biometrics* 29 (1972), 125–136.

[7] ANDRIENKO, G. L., AND ANDRIENKO, N. V. Interactive maps for visual data exploration. *International Journal Geographic Information Science* 13, 4 (1999), 355–374.

[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] ANUPAM, V., DAR, S., LEIBFRIED, T., AND PETAJAN, E. Dataspace: 3D visualization of large databases. In *Proc. Int. Symp. on Information Visualization, Atlanta, GA (1995)*, pp. 82–88.

[10] ASIMOV, D. The grand tour: A tool for viewing multidimensional data. *SIAM Journal of Science & Stat. Comp.* 6 (1985), 128–143.

[11] BATTISTA, G. D., EADES, P., TAMASSIA, R., AND TOLLIS, I. G. *Graph Drawing*. Prentice Hall, 1999.

[12] BECKER, R., CHAMBERS, J. M., AND WILKS, A. R. *The New S Language*. Wadsworth & Brooks/Cole Advanced Books and Software, Pacific Grove, CA, 1988.

[13] 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 (1996), 123–155.

[14] BEDERSON, B. Pad++: Advances in multiscale interfaces. In *Proc. Human Factors in Computing Systems CHI '94 Conf., Boston, MA (1994)*, p. 315.

[15] BEDERSON, B. B., AND HOLLAN, J. D. Pad++: A zooming graphical interface for exploring alternate interface physics. In *Proc. UIST (1994)*, pp. 17–26.

[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 (1993)*, pp. 73–80.

- [17] BUJA, A., SWAYNE, D. F., AND COOK, D. Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics* 5, 1 (1996), 78–99.
- [18] CARD, S., MACKINLAY, J., AND SHNEIDERMAN, B. *Readings in Information Visualization*. Morgan Kaufmann, 1999.
- [19] CARPENDALE, M. S. T., COWPERTHWAIT, D. J., AND FRACCHIA, F. D. Ieee computer graphics and applications, special issue on information visualization. *IEEE Journal Press* 17, 4 (July 1997), 42–51.
- [20] CARR, D. B., OLSEN, A. R., PIERSON, S. M., AND JEAN-YVES, P. C. Using linked micromap plots to characterize omernik ecoregions. *Data Mining and Knowledge Discovery* (2000).
- [21] CARR, D. B., WALLIN, J. F., AND CARR, D. A. Two new templates for epidemiology applications: linked micromap plots and conditioned choropleth maps. *Statistics in Medicine* (2000).
- [22] CARR, D. B., WEGMAN, E. J., AND LUO, Q. Explorn: Design considerations past and present. In *Technical Report, No. 129, Center for Computational Statistics, George Mason University* (1996).
- [23] CHEN, C. *Information Visualisation and Virtual Environments*. Springer-Verlag, London, 1999.
- [24] CHERNOFF, H. The use of faces to represent points in k-dimensional space graphically. *Journal Amer. Statistical Association* 68 (1973), 361–368.
- [25] CLEVELAND, W. S. *Visualizing Data*. AT&T Bell Laboratories, Murray Hill, NJ, Hobart Press, Summit NJ, 1993.
- [26] COOK, D., SYMANZIK, J., MAJURE, J. J., AND CRESSIE, N. Dynamic graphics in a gis: more examples using linked software. *Computers and Geosciences* (1997).
- [27] CRAMPTON, J. W. Interactivity types for geographic visualization. *Cartography & Geographic Information Science* 29, 2 (2002), 85–98.
- [28] DENT, B. D. *Cartography: Thematic Map Design, 4th Ed., Chapter 10*. William C. Brown, Dubuque, IA, 1996.
- [29] DODGE, M. Web visualization. http://www.geog.ucl.ac.uk/casa/martin/geography_of_cyberspace.html, Oct 2001.
- [30] EDSALL, R. M. The dynamic parallel coordinate plot: visualizing multivariate geographic data. In *Proc. 19th International Cartographic Association Conference, Ottawa* (May 1999), pp. 89–97.
- [31] EICK, S. G. Data visualization sliders. In *Proc. ACM UIST* (1994), pp. 119–120.
- [32] ESRI. Arc view. <http://www.esri.com/software/arcgis/arcview/index.html>, Feb. 2003.
- [33] FEINER, S., AND BESHES, C. Visualizing n-dimensional virtual worlds with n-vision. *Computer Graphics* 24, 2 (1990), 37–38.
- [34] FISHKIN, K., AND STONE, M. C. Enhanced dynamic queries via movable filters. In *Proc. Human Factors in Computing Systems CHI '95 Conf., Denver, CO* (1995), pp. 415–420.
- [35] FURNAS, G. Generalized fisheye views. In *Proc. Human Factors in Computing Systems CHI 86 Conf., Boston, MA* (1986), pp. 18–23.
- [36] FURNAS, G. W., AND BUJA, A. Prosections views: Dimensional inference through sections and projections. *Journal of Computational and Graphical Statistics* 3, 4 (1994), 323–353.
- [37] GAHEGAN, M. Scatterplots and scenes: visualisation techniques for exploratory spatial analysis. *Computers, Environment and Urban Systems* 21, 1 (1998), 43–56.
- [38] GAHEGAN, M., TAKATSUKA, M., WHEELER, M., AND HARDISTY, F. Introducing geovista studio: an integrated suite of visualization and computational methods for exploration and knowledge construction in geography. *International Journal Geographic Information Science* (2002).
- [39] GEISLER, G. Making information more accessible: A survey of information, visualization applications and techniques. <http://www.ils.unc.edu/~geisg/info/infovis/paper.html>, Feb. 2003.
- [40] 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* (1994), pp. 23–29.
- [41] GUSEIN-ZADE, S., AND TIKUNOV, V. Map transformations. *Geography Review* 9, 1 (1995), 19–23.
- [42] HAN, J., AND KAMBER, M. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 1986.
- [43] HAVRE, S., HETZLER, B., NOWELL, L., AND WHITNEY, P. Themeriver: Visualizing thematic changes in large document collections. *Transactions on Visualization and Computer Graphics* (2001).
- [44] HEARST, M. Tilebars: Visualization of term distribution information in full text information access. In *Proc. of ACM Human Factors in Computing Systems Conf. (CHI'95)* (1995), pp. 59–66.
- [45] HOMEPAGE, S. M. Sgi mineset. <http://www.sgi.com/software/mineset.html>, Feb. 2002.
- [46] HUBER, P. J. The annals of statistics. *Projection Pursuit* 13, 2 (1985), 435–474.
- [47] IHAKA, R., AND GENTLEMAN, R. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics* 5, 3 (1996), 299–314.
- [48] INSELBERG, A., AND DIMSDALE, B. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proc. Visualization 90, San Francisco, CA* (1990), pp. 361–370.

- [49] JOHANNSEN, A., AND MOORHEAD, R. J. Agp: Ocean model flow visualization. *IEEE Computer Graphics and Applications* 15, 4 (1995), 28–33.
- [50] JOHNSON, B., AND SHNEIDERMAN, B. Treemaps: A space-filling approach to the visualization of hierarchical information. In *Proc. Visualization '91 Conf* (1991), pp. 284–291.
- [51] KEAHEY, T. A. The generalized detail-in-context problem. In *Proceedings IEEE Visualization* (1998).
- [52] KEIM, D., KOUTSOFIOS, E., AND NORTH, S. C. Visual exploration of large telecommunication data sets. In *Proc. Workshop on User Interfaces In Data Intensive Systems (Invited Talk), Edinburgh, UK* (1999), pp. 12–20.
- [53] KEIM, D. A. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 6, 1 (January–March 2000), 59–78.
- [54] KEIM, D. A. Visual exploration of large data sets. *Communications of the ACM (CACM)* 44, 8 (2001), 38–44.
- [55] KEIM, D. A., AND HERRMANN, A. The gridfit algorithm: An efficient and effective approach to visualizing large amounts of spatial data. *IEEE Visualization, Research Triangle Park, NC* (1998), 181–188.
- [56] KEIM, D. A., AND KRIEGEL, H.-P. VisDB: Database exploration using multidimensional visualization. *Computer Graphics & Applications* 6 (September 1994), 40–49.
- [57] KEIM, D. A., AND KRIEGEL, H.-P. Issues in visualizing large databases. In *Proc. Conf. on Visual Database Systems (VDB'95), Lausanne, Schweiz, März 1995, in: Visual Database Systems* (1995), Chapman & Hall Ltd., pp. 203–214.
- [58] KEIM, D. A., KRIEGEL, H.-P., AND ANKERST, M. Recursive pattern: A technique for visualizing very large amounts of data. In *Proc. Visualization 95, Atlanta, GA* (1995), pp. 279–286.
- [59] KEIM, D. A., NORTH, S. C., AND PANSE, C. Cartodraw: A fast algorithm for generating contiguous cartograms. *Trans. on Visualization and Computer Graphics* (Dez 2003). Information Visualization Research Group, AT&T Laboratories, Florham Park.
- [60] KEIM, D. A., NORTH, S. C., PANSE, C., AND SCHNEIDEWIND, J. Visualpoints contra cartodraw. *Palgrave Macmillan – Information Visualization* (March 2003).
- [61] KEIM, D. A., NORTH, S. C., PANSE, C., AND SIPS, M. Pixelmaps: A new visual data mining approach for analyzing large spatial data sets. Tech. rep., Information Visualization Research Group, AT&T Laboratories, Florham Park & DBVIS–Group, Uni-Konstanz, Germany, 2003. http://dbvis.fmi.uni-konstanz.de/~sips/papers/03/KNPS03_TECHREPORT.pdf.
- [62] KEIM, D. A., PANSE, C., AND SIPS, M. Visual data mining of large spatial data sets. In *3rd Workshop on Databases in Networked Information Systems, Aizu, Japan* (September 2003).
- [63] KOCMOUD, C. J., AND HOUSE, D. H. Continuous cartogram construction. *Proceedings IEEE Visualization* (1998), 197–204.
- [64] LAMPING, J., R., R., AND PIROLI, P. A focus + context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proc. Human Factors in Computing Systems CHI 95 Conf.* (1995), pp. 401–408.
- [65] LEBLANC, J., WARD, M. O., AND WITTELS, N. Exploring n-dimensional databases. In *Proc. Visualization '90, San Francisco, CA* (1990), pp. 230–239.
- [66] LEUNG, Y., AND APPERLEY, M. A review and taxonomy of distortion-oriented presentation techniques. In *Proc. Human Factors in Computing Systems CHI '94 Conf., Boston, MA* (1994), pp. 126–160.
- [67] LEVKOWITZ, H. Color icons: Merging color and texture perception for integrated visualization of multiple parameters. In *Proc. Visualization 91, San Diego, CA* (1991), pp. 22–25.
- [68] M. KREUSELER, N. L., AND SCHUMANN, H. A scalable framework for information visualization. *Transactions on Visualization and Computer Graphics* (2001).
- [69] MACEACHREN, A. M. Conditioned choropleth mapping systemproject: Conditioned choropleth maps: Dynamic multivariate representations of statistical data, June 2003. <http://www.geovista.psu.edu/grants/dg-gg/CCmaps/>.
- [70] MACEACHREN, A. M., HARDISTY, F., DAI, X., AND PICKLE, L. Geospatial statistics supporting visual analysis of federal geospatial statistics. *Digital government table of contents* (2003), 59–60. ISSN:0001-0782.
- [71] MACKINLAY, J. D., ROBERTSON, G. G., AND CARD, S. K. The perspective wall: Detail and context smoothly integrated. In *Proc. Human Factors in Computing Systems CHI '91 Conf., New Orleans, LA* (1991), pp. 173–179.
- [72] MARSHALL, R., KEMPF, J., DYER, D. S., AND YEN, C.-C. Visualization methods and simulation steering for a 3d turbulence model of lake erie. In *Symposium on Interactive 3D Graphics, Computer Graphics* (1987), pp. 89–97.
- [73] MONMONIER, M. Geographic brushing: Enhancing exploratory analysis of the scatterplot matrix. *Geographical Analysis* 21, 1 (1989), 81–84.
- [74] 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* (1995), pp. 33–38.
- [75] PERLIN, K., AND FOX, D. Pad: An alternative approach to the computer interface. In *Proc. SIGGRAPH, Anaheim, CA* (1993), pp. 57–64.
- [76] 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* (1988), pp. 514–519.

- [77] RAISZ, E. *Principles of Cartography*. McGraw-Hill, New York, 1962.
- [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* (1994), pp. 318–322.
- [79] ROBERTSON, G. G., MACKINLAY, J. D., AND CARD, S. K. Cone trees: Animated 3D visualizations of hierarchical information. In *Proc. Human Factors in Computing Systems CHI 91 Conf., New Orleans, LA* (1991), pp. 189–194.
- [80] SARKAR, M., AND BROWN, M. Graphical fisheye views. *Communications of the ACM* 37, 12 (1994), 73–84.
- [81] 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* (1993), pp. 87–96.
- [82] SCHUMANN, H., AND MÜLLER, W. *Visualisierung: Grundlagen und allgemeine Methoden*. Springer, 2000.
- [83] SHNEIDERMAN, B. Tree visualization with treemaps: A 2D space-filling approach. *ACM Transactions on Graphics* 11, 1 (1992), 92–99.
- [84] SHNEIDERMAN, B. The eye have it: A task by data type taxonomy for information visualizations. In *Visual Languages* (1996).
- [85] SPENCE, B. *Information Visualization*. Pearson Education Higher Education publishers, UK, 2000.
- [86] SPENCE, R., AND APPERLEY, M. Data base navigation: An office environment for the professional. *Behaviour and Information Technology* 1, 1 (1982), 43–54.
- [87] SPENCE, R., TWEEDIE, L., DAWKES, H., AND SU, H. Visualization for functional design. In *Proc. Int. Symp. on Information Visualization (InfoVis '95)* (1995), pp. 4–10.
- [88] SPOERRI, A. Infocrystal: A visual tool for information retrieval. In *Proc. Visualization '93, San Jose, CA* (1993), pp. 150–157.
- [89] STASKO, J., DOMINGUE, J., BROWN, M., AND PRICE, B. *Software Visualization*. MIT Press, Cambridge, MA, 1998.
- [90] 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).
- [91] SWAYNE, D. F., COOK, D., AND BUJA, A. *User's Manual for XGobi: A Dynamic Graphics Program for Data Analysis*. Bellcore Technical Memorandum, 1992.
- [92] TIERNEY, L. *LispStat: An Object-Orientated Environment for Statistical Computing and Dynamic Graphics*. Wiley, New York, NY, 1991.
- [93] TOBLER, W. Cartograms and cartosplines. *Proceedings of the 1976 Workshop on Automated Cartography and Epidemiology* (1976), 53–58.
- [94] TRILK, J. Software visualization. <http://www.broy.informatik.tu-muenchen.de/~trilk/sv.html>, Oct 2001.
- [95] VAN WIJK, J. J., AND VAN LIERE, R. D. Hyperslice. In *Proc. Visualization '93, San Jose, CA* (1993), pp. 119–125.
- [96] VELLEMAN, P. F. *Data Desk 4.2: Data Description*. Data Desk, Ithaca, NY, 1992, 1992.
- [97] WARD, M. O. Xmdvtool: Integrating multiple methods for visualizing multivariate data. In *Proc. Visualization 94, Washington, DC* (1994), pp. 326–336.
- [98] WARE, C. *Information Visualization: Perception for Design*. Morgan Kaufman, 2000.
- [99] WILHELM, A., UNWIN, A., AND THEUS, M. Software for interactive statistical graphics - a review. In *Proc. Int. Softstat 95 Conf., Heidelberg, Germany* (1995).