Pushing the Limit in Visual Data Exploration: Techniques and Applications*

Daniel A. Keim[1], Christian Panse[1], Jörn Schneidewind[1], Mike Sips[1], Ming C. Hao[2], and Umeshwar Dayal[2]

[1] University of Konstanz, Germany
{keim,panse,schneide,sips}@informatik.uni-konstanz.de

[2] Hewlett Packard Research Laboratories, Palo Alto, CA, USA
{ming_hao,dayal}@hp.hpl.com

Abstract. With the rapid growth in size and number of available databases, it is necessary to explore and develop new methods for analysing the huge amounts of data. Mining information and interesting knowledge from large databases has been recognized by many researchers as a key research topic in database systems and machine learning, and by many industrial companies as an important area with an opportunity of major revenues. Analyzing the huge amount (usually tera-bytes) of data obtained from large databases such as credit card payments, telephone calls, environmental records, census demographics, however, a very difficult task. *Visual Exploration* and *Visual Data Mining* techniques apply human visual perception to the exploration of large data sets and have proven to be of high value in exploratory data analysis. Presenting data in an interactive, graphical form often opens new insights, encouraging the formation and validation of new hypotheses to the end of better problem-solving and gaining deeper domain knowledge. In this paper we give a short overview of visual exploration techniques and present new results obtained from applying PixelBarCharts in sales analysis and internet usage management.

Keywords: Information Visualization, Visual Data Mining, Visual Exploration, Knowledge Discovery, Pixel Displays

1 Introduction

Progress in technology allows today's computer systems to store and exchange amounts of data that until very recently were considered extraordinarily vast. The automation of business activities produces an ever-increasing stream of data, because even simple transactions, such as the use of a credit card, shopping in e-commerce stores or telephone calls are typically recorded by a computer. The data is collected, because it is a potential source of valuable information, providing a competitive advantage to its holders. In addition, commercial databases, scientific and government databases are also rapidly growing. The data is often

* Portions of this article have previously appeared in [11,12,15].

A. Günter et al. (Eds.): KI 2003, LNAI 2821, pp. 37–51, 2003.

automatically recorded via sensors and monitoring systems. 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 this data. If the data is presented textually, the amount of data that can be displayed is in the range of some hundred 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'. The computer science community is responding to both the scientific and practical challenges. The broad topic of knowledge discovery and automatic learning is inherently cross-disciplinary in nature - it falls right into the intersection of disciplines including Statistics, Artificial Intelligence (AI), Visualization and Data Mining. Applying machine learning techniques for knowledge discovery in large databases is a major research area in the field of Artificial Intelligence. Large databases consist of millions of transactions, and it is desirable to have machine learning algorithms, that can handle and analyze such large data sets. In this paper we present visual data exploration techniques, which are indispensable to solving the problem of exploring large data sets.

2 Visual Data Mining

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 computer systems. Visual data exploration aims at integrating humans in the data exploration process, applying their perceptual abilities to the 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 human to get 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 they also have a high potential for exploring large databases.

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 [22]). 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 these 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 process, it is important to keep the overview visualization while focusing on the subset using another visualization. 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 the screen space of the 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

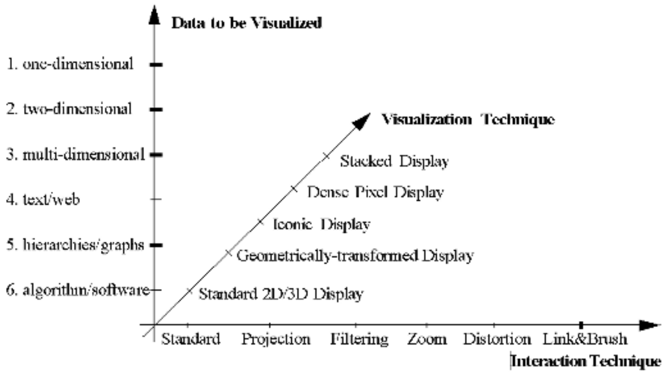


Fig. 1. Classification of visual data exploration techniques

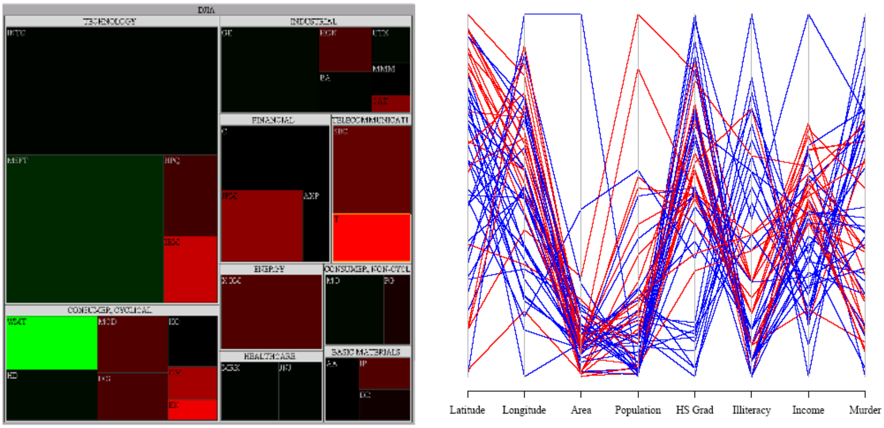
does not only provide visualization techniques for all three steps but also bridges the gaps between them.

2.1 Classification of Visual Data Mining Techniques

There are a number of well known techniques for visualizing large 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. Over the last years, 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 [3] [20] [23] [26]. The techniques can be classified based on three criteria [10] (see also Figure 1):

– the data to be visualized
– the visualization technique
– and the interaction technique used

The ***data type to be visualized*** [22] 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 *algorithms and software.*
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 hyperbolic plane [24] and parallel coordinates (Figure 2(b)) [7], *Icon-based displays*, such as chernoff faces [4] and stick figures [18] [19], *pixel displays*, such as the recursive pattern [1] and circle segments [2], and *Stacked displays*, such as treemaps (Figure 2(a)) [8] [21] and dimensional stacking [25].



(a) Treemap visualization of stock market data. The area of each box corresponds to the volume of the stock trade and the color reflects the stock price changing.

(b) The Parallel Coordinates plot displays the US-Census data of the 50 states. Color is used to point out which party has won the 2000 US presidential election

Fig. 2. Examples for a hierarchical 2(a) and geometrical 2(b) visualization techniques.

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

Note that the three dimensions of our classification - data type to be visualized, visualization technique, and interaction technique - can be assumed to be orthogonal. Orthogonality means that any of the visualization techniques may be used in conjunction with any of the interaction techniques for any data type. Note also that a specific system may be designed to support different data types and that it may use a combination of visualization and interaction techniques. More details can be found in [15].

3 Pixel Based Visualization Techniques

A special group of visual data mining techniques are the pixel-oriented techniques. The class of pixel-oriented techniques is an important class of visualization techniques for visualizing very large multidimensional data sets. The general idea of pixel oriented visualization techniques is to represent as many data objects as possible on the screen at the same time by mapping each data value to a pixel of the screen and arranging the pixels adequately. Many different pixel-oriented visualization techniques have been proposed in recent years and it has been shown that the techniques are useful for visual exploration of large databases in a number of different application contexts.

3.1 Basic Idea of Pixel-Display Techniques

The basic idea of pixel-oriented techniques is to map each data value to a colored pixel and present the data values belonging to one dimension (attribute) in a separate subwindow (see Figure 3).

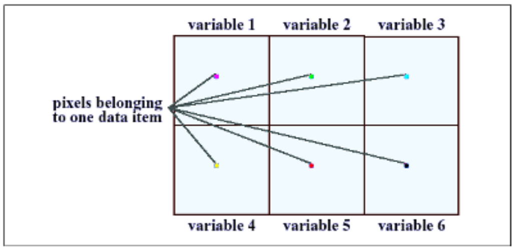


Fig. 3. Arrangement of Windows for Data with Six Variables

Since, in general, pixel display techniques use only one pixel per data value, the techniques allow us to visualize the largest amount of data which is possible on current displays (up to about $1,000,000$ data values). All pixel-display techniques partition the screen into multiple subwindows. For data sets with m dimensions (attribute), the screen is partitioned into m subwindows, one for each of the dimensions. Correlations, functional dependencies, and other interesting relationships between dimensions may be detected by relating corresponding regions in the multiple windows. To achieve that objective, a number of design problems have to be solved.

3.2 Design Issues for Pixel-Based Visualization Techniques

Color Mapping. The first problem is the mapping of data values to colors. A good color mapping is obviously very important, but has to be carefully engineered to be intuitive. The advantage of color over gray scales is that the number of Just Noticeable Differences (JND)[14] are much higher. Finding a path through a color space that maximizes the numbers of just noticeable difference, but at the same time, is intuitive for the application domain, however, is a difficult task. More information about color models and color mappings can be found in [5]

Arrangement of Pixels. As already mentioned, the basic idea of pixel-oriented visualization techniques is to present as many data values as possible at the same time with the number of data values being only limited by the number of pixels of the display. In dealing with arbitrary multivariate data without any 2D-or 3D-semantics, a major problem is to find meaningful arrangements of pixels on the screen. Even if the data has a natural ordering according to one variable, 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-dowm in a column-by-column fashion. If these arrangements are done pixelwise, 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 or the Recursive Pattern technique.

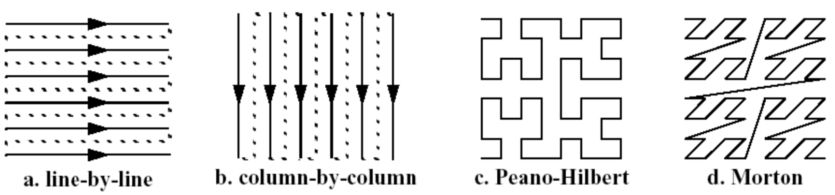


Fig. 4. Data Arrangements

The *screen-filling curves techniques* are based on the well-known space filling curve algorithms by Peano and Hilbert[6,17] and Morton[16]. The idea of these techniques is to provide a continuous curve which passes through every point of a regular spatial region. Space filling curves provide a mapping which preserves the spatial locality of the original image. In visualizing multivariate data, which is sorted according to one dimension, we can use space-filling curves to map the one dimensional distribution of data items onto the two dimensions of the screen. Space filling curves have the nice property that data items which are close together in the one dimensional distribution are likely to be close together in the two dimensional visual representation. This property of space-filling curves may help to discover patterns in the data which are difficult to discover otherwise. If each variable is visualized using the same arrangement, interesting properties of the data may be revealed including the distribution of variable values, correlations between variables, and clusters.

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 that the basic elements that are arranged on level i are the pattern resulting from the level $(i-1)$ arrangements. In Figure 5, an example recursive pattern visualization of financial data is shown. The visualization shows twenty years (January 1974 - April 1995) of daily prices of the 50 stocks contained in the Frankfurt Stock Index (FAZ).

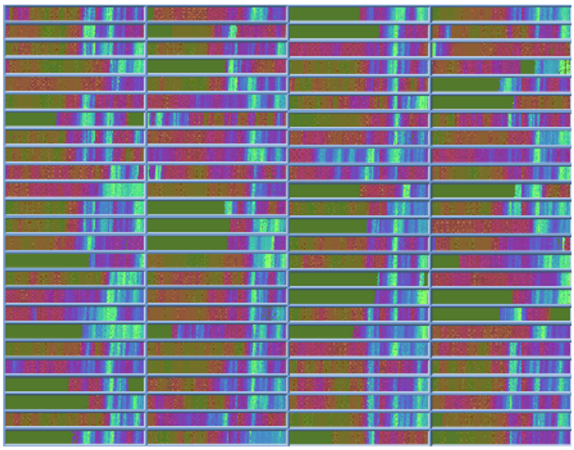
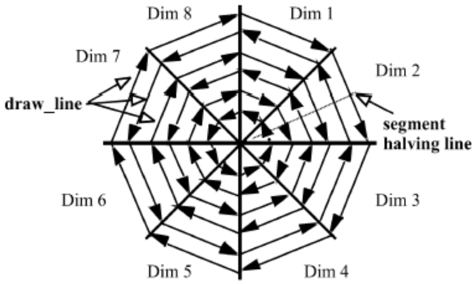


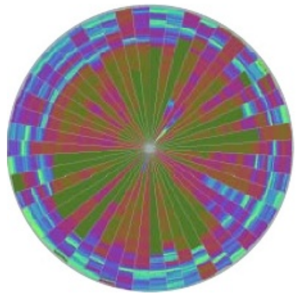
Fig. 5. Dense Pixel Displays: Recursive Pattern Technique showing 50 stocks in the Frankfurt Allgemeine Zeitung (Frankfurt Stock Index Jan 1975 - April 1995). The technique maps each stock value to a colored pixel; high values correspond to bright colors. ©IEEE

Shape of Subwindows. Another important question is whether there exists an alternative to the partitioning of the screen into rectangular subwindows. The rectangular shape of the subwindows allows a good screen usage, but at 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 are rather far apart, which prevent the user from detecting clusters, correlations, and interesting patterns. An alternative shape of the subwindows is the *Circle Segments technique* [2]. The basic idea of these technique is to display the data dimensions as segments of a circle, as shown in Figure 6.

Ordering of Dimensions. The next 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



(a)



(b)

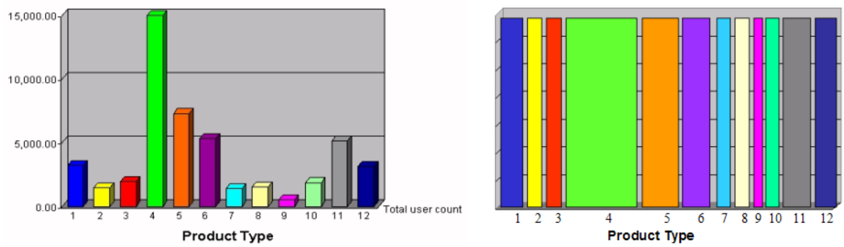
Fig. 6. Circle Segments: Figure (a) shows the circle segment technique for 8 dimensional data; Figure (b) visualizes twenty years (January 1974 - April 1995) of daily prices of the 50 stocks contained in the Frankfurt Stock Index (FAZ)

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 allows different interpretations. More details about designing pixel-oriented visualization techniques can be found in [9].

4 Pixel Bar Charts

In many cases simple presentation graphics, like bar charts or pie charts, are used to visualize large data sets, because these techniques are intuitive and easy-to-use. The disadvantage is, that they show only highly aggregated data and present only a very limited number of data values (as in the case of bar charts), and may have a high degree of overlap which may occlude a significant portion of the data values (as in the case of x-y plots). *Pixel Bar Charts*[13] solve these problems by presenting a pixel-oriented generalization of traditional bar charts and x-y-plots, which allows the visualization of large amounts of data. The basic idea is to use the pixels within the bars to present the detailed information of the data records. Pixel Bar Charts retain the intuitiveness of traditional bar charts while allowing very large data sets to be visualized in an effective way. Bar charts are widely used, very intuitive and easy to understand. Figure 7(a) illustrates the use of a regular bar chart to visualize customer distribution in an e-commerce sales transaction. The height of the bars represents the number of customers for 12 different product categories. Bar charts, however, require a high degree of data aggregation and actually show only a rather small number of data values (only 12 values are shown in Figure 7(a)). Therefore, they are of limited value for data exploration of large multidimensional data, and are not able to show important information such as:

- data distributions of multiple attributes
- local patterns, correlations, and trends
- detailed information, such as each customer profile data

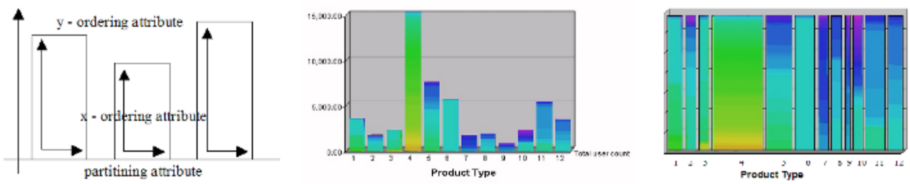


(a) Equal-Width Bar Chart (b) Equal-Height Bar Chart

Fig. 7. Traditional Bar Charts

4.1 Basic Idea of Pixel Bar Charts

Pixel bar charts are derived from regular bar charts. The basic idea of a pixel bar chart is to present the data values directly instead of aggregating them into a few data values. The approach is to represent each data item (e.g. a customer) by a single pixel in the bar chart. The detailed information of one attribute of each data item is encoded into the pixel color and can be accessed and displayed as needed. One important question is: how are the pixels arranged within each bar? Our idea is to use one or two attributes to separate the data into bars and then use two additional attributes to impose an ordering within the bars. The general idea is shown in Figure 8(a). The pixel bar chart can therefore be seen as a combination of the traditional bar charts and the x-y diagrams. Now, we have a visualization in which one pixel corresponds to one customer. If the partitioning attribute is redundantly mapped to the colors of the pixels, we obtain the regular bar chart shown in Figure 7(a) (Figure 7(b) shows the equal-height-bar-chart", which we explain in the next section). Pixel bar charts, however, can be used to present large amounts of detailed information. The one-to-one correspondence between customer data and pixels allows us to use the color of the pixels to represent an additional attribute of the customer - for example, sales amount, number of visits, or sales quantity. In Figure 8, a pixel bar chart is used to visualize thousands of e-commerce sales transactions. Each pixel in the visualization represents one customer. The number of customers can be as large as the screen size (about 1.3 million). The pixel bar chart shown in Figure 8(b) uses product type as the partitioning attribute and dollar amount as the x and y ordering attributes. The color represents the dollar amount spent by the corresponding customer. High dollar amounts correspond to bright colors, low dollar amounts to dark colors



(a) Basic Idea (b) Pixel Bar Chart (c) Space-Filling

Fig. 8. Pixel Bar Charts

4.2 Space-Filling Pixel Bar Charts

One problem of traditional bar charts is that a large portion of the screen space is not used due to the differing heights of the bars. With very large data sets, we would like to use more of the available screen space to visualize the data. One idea that increases the number of displayable data values is to use equal-height instead of equal-width bar charts. In Figure 7(b), the regular bar chart of Figure 7(a) is shown as an equal-height bar chart. The area (width) of the bars corresponds to the attribute shown, namely the number of customers.

If we now apply our pixel bar chart idea to the resulting bar charts, we obtain space-filling pixel bar charts which use virtually all pixels of the screen to display customer data items. In Figure 8(c), we show an example of a space-filling pixel bar chart which uses the same partitioning, ordering, and coloring attributes as the pixel bar chart in Figure 8(b). In this way, each customer is represented by one pixel. Note that pixel bar charts generalize the idea of regular bar charts. If the partitioning and coloring attributes are identical, both types of pixel bar charts become scaled versions of their regular bar chart counterparts. The pixel bar chart can therefore be seen as a generalization of the regular bar charts but they contain significantly more information and allow a detailed analysis of large original data sets.

4.3 Multi-pixel Bar Charts

In many cases, the data to be analyzed consists of multiple attributes. With pixel bar charts we can visualize attribute values using multi-pixel bar charts which use different color mappings but the same partitioning and ordering attributes. This means that the arrangement of data items within the corresponding bars of multi-pixel bar charts is the same, i.e., the colored pixels corresponding to the different attribute values of the same data item have a unique position within the bars. Figure 9 shows an example of three pixel bar charts with product type as the partitioning attribute and number of visits and dollar amount as the x and y ordering attributes. The attributes which are mapped to color are dollar amount spent, number of visits, and sales quantity.

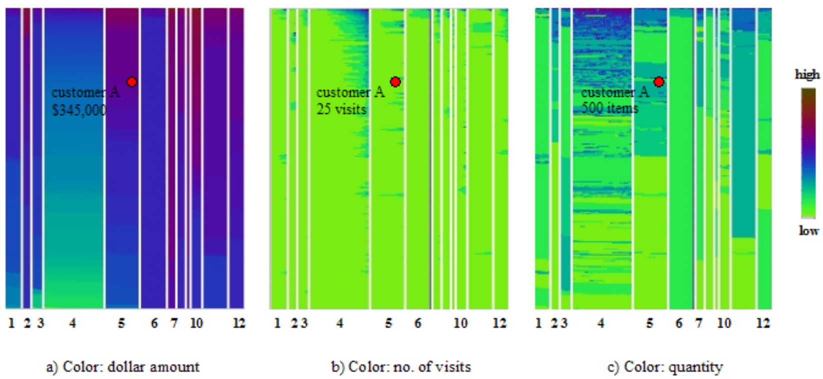


Fig. 9. Multi-pixel Bar Chart for Mining 405,000 Sales Transaction Records

4.4 Application of Pixel Bar Charts in Sales Analysis and Internet Usage

The pixel bar chart technique has been applied in two business service applications - Sales Analysis and Internet Duration Time Analysis at Hewlett Packard Laboratories. The applications show the wide applicability and usefulness of pixel bar charts.

Sales Analysis. The rapid growth of business on the Internet has led to the availability of large volumes of data. Business research efforts have been focused on how to turn raw data into actionable knowledge. In order to find and retain customers, business analysts need to improve their sales quality based on prior information. For sales analysis, sales specialists would like to discover new patterns and relationships in the invoice data. Common questions are ‘What is the sales growth rate in recent months? ’, ‘Which product has the most sales? ’, and ‘Where do the sales come from?’.

In Figure 10, each pixel represents an invoice. Color represents sales amount. The red color indicates that the sales amount exceeds $ 10.000. The width of a bar represents the sales volume. The analyst can find the following:

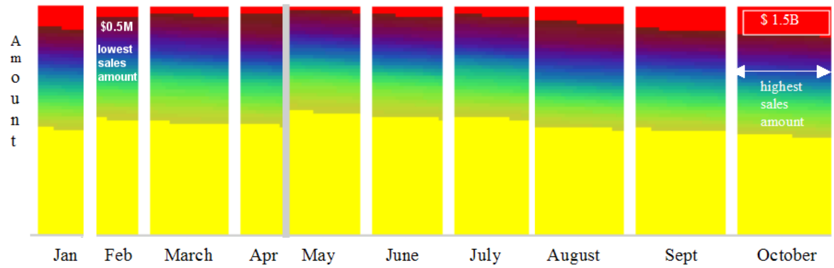
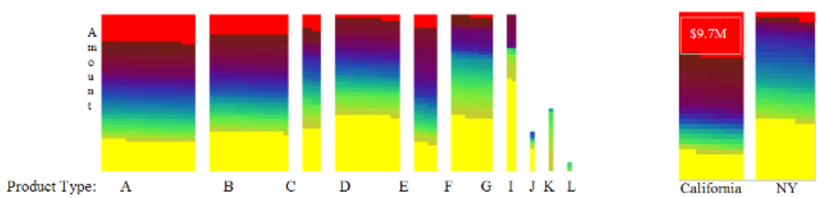


Fig. 10. Sales Growth Rate Over 10 Months

- Sales amount varies over time (shown by a growing color wave)
- Sales volume varies over time (shown by bar width)
- Sales amount distribution over time (shown by color)
- By clicking on an invoice (pixel), the user is able to find detail information about the transaction (products, location, time ...)



(a) Product Type Distribution

(b) Sales Location

Fig. 11. Drill down to show the October sales distribution 11(a) and to show that most sales come from California 11(b)

The analyst is further able to rubber-band an area inside a bar to display the total sales amount in the area. The sales amount represented by the rubber-band areas has grown from $0.5M (February) to $1.5B (October). Many important facts may be discovered in the layered drill-downs of Figure 11. The analyst may observe the following facts:

1. Sales distribution: From Figure 10, it becomes clear that October has the highest sales amount (less yellow, more burgundy and red). The analyst can drill down to the next level of hierarchy (products). A visualization of the drill-down to products is shown in Figure 11(a). Product ‘A ’has the highest sales amount (less yellow, more blue, burgundy, and red) and highest sales volumes (widest bar).

2. Sales source: The analyst may continue drill-down to examine other interesting facts, for example, where the sale comes from which source/location. From Figure 11(b), it is clear that most sales comes from 'California'(more burgundy and red). The analyst can rubber-band area to display the total sales amount in the area (i.e. $ 9.7M).

Internet Usage Management Analysis. Pixel bar charts have been also used to explore Internet usage management. The pixels of the bar chart represent Internet events. In pixel bar charts, events with similar duration time are placed close to each other. A system analyst can use the visualization to rapidly discover event patterns and use those patterns to identify the latency and manage the Internet configuration.

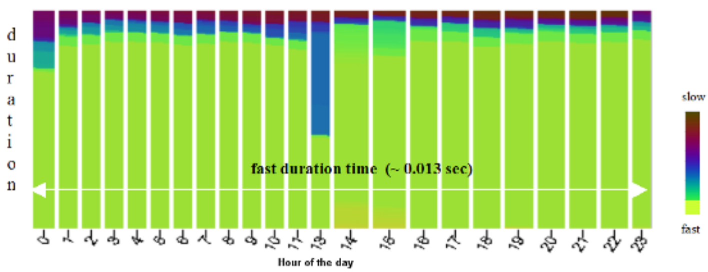


Fig. 12. Internet Event Duration Time Distribution

The pixel bar chart shown in Figure 12 illustrates the Internet event hourly duration time distribution. Each pixel represents an event record. It uses *hour* as the dividing attribute and *duration time* as the y-ordering attributes. The color represents the duration time value of an Internet event. The fast duration time corresponds to bright colors (yellow, green); the slow duration time corresponds to dark colors (blue, burgundy). From the data distribution, we can discover that a large number of fast Internet events occurred cross all 24 hours except hour 13. Hour 13 has highest average duration time (more blue, burgundy). It is crucial for analysts to realize that the number of events with high duration time in hour 13 is only half of the events processed in hour 13. The rest of events have short duration time (green). These valuable facts get lost in the aggregated data. As a result, the Internet analyst may make a wrong decision.

Figure 13 illustrates the correlations between the number of bytes and the number of pockets. The x-axis dividing attribute is hour and the y-axis ordering attribute is the number of bytes. The colors in the different bar charts represent number of pockets. The analysts can easily conclude that there is a close correlation between number of bytes and number of pockets. It shows that the high number of pockets are corresponding to the high number of bytes transferred

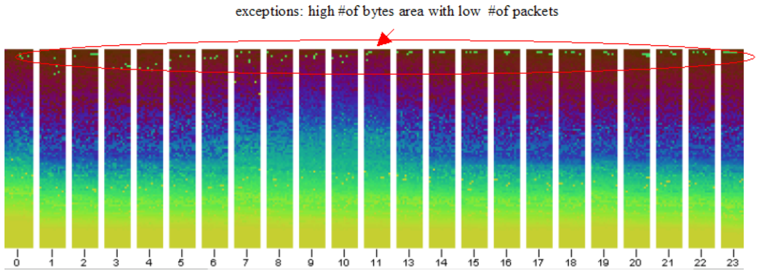


Fig. 13. Internet number of Bytes and number of Pockets Correlations

across all 24 hours (with the same patterns of yellow, green, blue, and burgundy colors) except several sprinkles (exceptional cases), such as there are several low number of pockets in the highest number of bytes areas as shown inside the red circle in Figure 13.

5 Conclusion

Visual Data Mining and Visual Data Exploration is an important research area. Many data sources provide large amounts of data with multidimensional attributes. In this article, we present an overview over methods for the Visual Exploration of large data sets. We especially focus on pixel display techniques which are able to visualize significantly more information then existing techniques. The application examples exemplify the potential of visual data exploration techniques.

References

1. M. Ankerst, D. A. Keim, and H.-P. Kriegel. Recursive pattern: A technique for visualizing very large amounts of data. In *Proc. Visualization '95, Atlanta, GA*, pages 279–286, 1995.
2. M. Ankerst, D. A. Keim, and H.-P. Kriegel. Circle segments: A technique for visually exploring large multidimensional data sets. In *Visualization '96, Hot Topic Session, San Francisco, CA*, 1996.
3. S. Card, J. Mackinlay, and B. Shneiderman. *Readings in Information Visualization.* Morgan Kaufmann, 1999.
4. H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal Amer. Statistical Association*, 68:361–368, 1973.
5. G. Herman and H. Levkowitz. Color scales for image data. *Computer Graphics and Applications*, pages 72–80, 1992.
6. D. Hilbert. Über stetige Abbildungen einer Linie auf einem Flächenstück. *Mathematische Annalen*, (78), 1891.
7. A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multidimensional geometry. In *Proc. Visualization 90, San Francisco, CA*, pages 361–370, 1990.

8. B. Johnson and B. Shneiderman. Treemaps: A space-filling approach to the visualization of hierarchical information. In *Proc. Visualization '91 Conf*, pages 284–291, 1991.
9. D. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *Transactions on Visualization and Computer Graphics*, 6(1):59–78, Jan–Mar 2000.
10. D. Keim. Visual exploration of large databases. *Communications of the ACM*, 44(8):38–44, 2001.
11. D. A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 6(1):59–78, January–March 2000.
12. D. A. Keim, M. Hao, and U. Dayal. Hierarchical pixel bar charts. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 8(3):255–269, July–September 2002.
13. D. A. Keim, M. C. Hao, J. Ladisch, M. Hsu, and U. Dayal. Pixel bar charts: A new technique for visualizing large multi-attribute data sets without aggregation. In *Visualization 2001, Los Angeles*, page 113, 2001.
14. D. A. Keim and H.-P. Kriegel. Issues in visualizing large databases. In *Proc. Conf. on Visual Database Systems (VDB'95), Lausanne, Schweiz, März 1995, in: Visual Database Systems*, pages 203–214. Chapman & Hall Ltd., 1995.
15. D. A. Keim and M. Ward. *Intelligent Data Analysis, an Introduction by D. Hand and M. Berthold*, chapter Visual Data Mining Techniques. Springer Verlag, 2 edition, 2002.
16. G. Morton. A computer oriented geodetic data base and a new technique in file sequencing. *IBM Ltd, Ottawa, Canada*, 1966.
17. G. Peano. Sur une courbe qui remplit toute une aire plane. *Mathematische Annalen*, (36), 1890.
18. R. M. Pickett. *Visual Analyses of Texture in the Detection and Recognition of Objects*. Academic Press, New York, 1970.
19. R. M. Pickett and G. G. Grinstein. Iconographic displays for visualizing multidimensional data. In *Proc. IEEE Conf. on Systems, Man and Cybernetics, IEEE Press, Piscataway, NJ*, pages 514–519, 1988.
20. H. Schumann and W. Müller. *Visualisierung: Grundlagen und allgemeine Methoden*. Springer, 2000.
21. B. Shneiderman. Tree visualization with treemaps: A 2D space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, 1992.
22. B. Shneiderman. The eye have it: A task by data type taxonomy for information visualizations. In *Visual Languages*, 1996.
23. B. Spence. *Information Visualization*. Pearson Education Higher Education publishers, UK, 2000.
24. J. Walter and H. Ritter. On interactive visualization of high-dimensional data using the hyperbolic plane. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 123–131, 2002.
25. M. O. Ward. Xmdvtool: Integrating multiple methods for visualizing multivariate data. In *Proc. Visualization 94, Washington, DC*, pages 326–336, 1994.
26. C. Ware. *Information Visualization: Perception for Design*. Morgen Kaufman, 2000.