A Visual Analysis of Multi-Attribute Data Using Pixel Matrix Displays

Ming C. Hao, Umeshwar Dayal, Daniel Keim*, Tobias Schreck*
Hewlett Packard Laboratories, Palo Alto, CA
(ming.hao, umeshwar.dayal)@hp.com

ABSTRACT

Charts and tables are commonly used to visually analyze data. These graphics are simple and easy to understand, but charts show only highly aggregated data and present only a limited number of data values while tables often show too many data values. As a consequence, these graphics may either lose or obscure important information, so different techniques are required to monitor complex datasets. Users need more powerful visualization techniques to digest and compare detailed multi-attribute information to analyze the health of their business. This paper proposes an innovative solution based on the use of ***pixel-matrix*** to represent transaction-level information within graphics. With pixel-matrixes, users can visualize areas of importance at a glance, a capability not provided by common charting techniques. Our solutions are based on colored pixel-matrixes, which are used in (1) charts for visualizing data patterns and discovering exceptions, (2) tables for visualizing correlations and finding root-causes, and (3) time series for visualizing the evolution of long-running transactions. The solutions have been applied with success to product sales, Internet network performance analysis, and service contract applications demonstrating the benefits of our method over conventional graphics. The method is especially useful when detailed information is a key part of the analysis.

Keywords: pixel-matrix visualization, multi-attribute dataset, bar charts, tables, and time series.

1. Introduction

A common method for visualizing large volumes of data is to use charts and tables. They are widely used and are very intuitive and easy to understand. Figure 1 (A) illustrates the use of a regular bar chart to help visualize the daily product sales prices from 4/18 to 4/28. The height of the bars represents the total sales prices for eleven days. Bar charts, however, use a high degree of data aggregation and actually show only a rather small number of data values (only eleven values are shown). Figure 1 (C) illustrates the use of a spreadsheet to visualize Internet network performance. Performance metrics (e.g., *Response Time*, *Availability*, and *Throughput*) are represented by three columns in the table. Analysts need to sift through many pages of rows to get the answers. Figure 1 (E) illustrates a time series with a large number of observation values. The time series have a high degree of overlap which obscures important information.

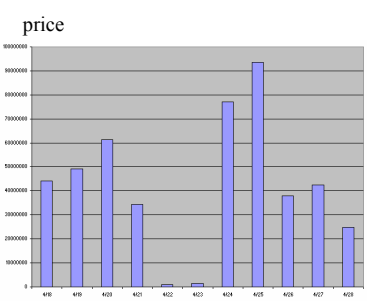
For the exploration of large volumes of multi-attribute data, the current charts and tables are not able to show important information such as:

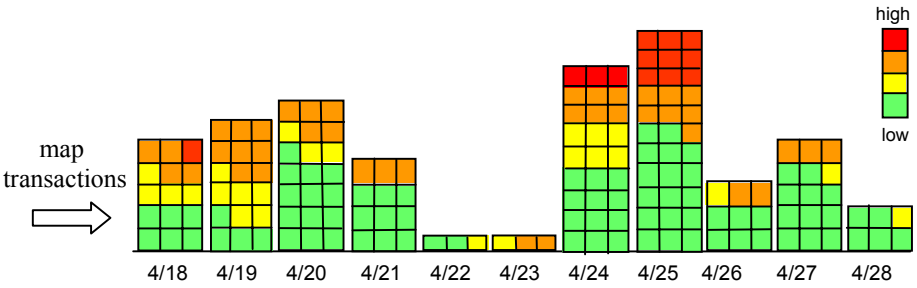
- Data distribution of multiple attributes,
- Comparison of correlations and patterns,
- Instantaneous drilldown to transaction level information (e.g., price and quantity in an invoice).

Beyond charts and tables, there are many well-known techniques developed for analyzing multi-dimensional data. For example, Tableau's visual spreadsheet [1, 2] and SpotFire's charts [3] are widely used by managers to make business decisions. The various versions of treemaps [4, 5, 6] are used to visualize hierarchical information. The VisDB system [7] uses the pixel-oriented technique to represent data patterns and trends that allow the user to explore multi-dimensional databases.

The SeeSoft line representation technique [8] is used for visualizing program changes. E_BizInsights [9] is used for web path analysis. Parallel coordinates [10] is used for visualizing correlations. To visualize time series data, there are Tominski's [11] axes-based, Wijk's [13] calendar visual presentations, as well as Shneiderman's [12] interactive pattern search. All these techniques have contributed innovative visualization techniques emphasizing the transforming of data into valuable information to show patterns and trends using either aggregated or raw data. Our approach is different from these techniques. We propose a new pixel-matrix visualization method as described in the next section to present detailed information about data distributions, correlations and patterns to the user.

* University of Konstanz, Germany



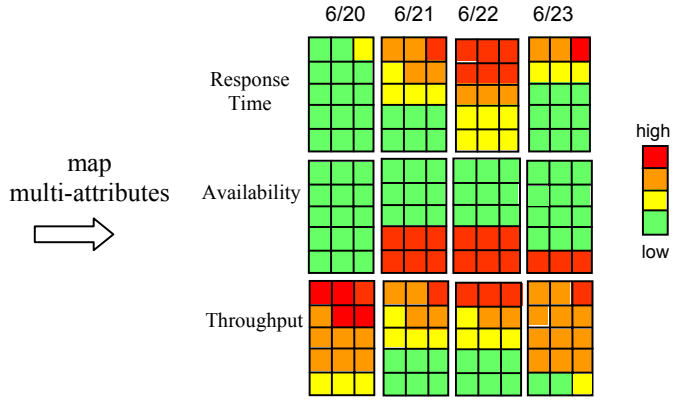


(A) A Daily Product Sales Bar Chart
Bar height showing the total price, but no other information

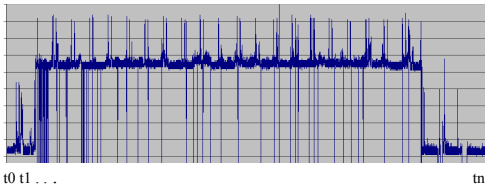
(B) A Daily Product Sales *Color Pixel-Matrix* Bar Chart
Each pixel-matrix represents a product sales transaction. Color represents the sales price. Pixel-matrixes are ordered from left to right and bottom to top. In addition to showing total price, they also show the sales distribution, patterns, and outliers (red).

	A	B	C	D	E
1	DateTime	Network	Availability	Response	Throughput
2	6/20/2005 0:00	HTTP	1	0.154	45.692
3	6/20/2005 0:05	HTTP	1	0.06	71.169805
4	6/20/2005 0:10	HTTP	1	0.15	46.484375
5	6/20/2005 0:15	HTTP	1	0.276	141.68668
6	6/20/2005 0:20	HTTP	1	0.094	51.4987
7	6/20/2005 0:25	HTTP	1	0.192	27.890625
8	6/20/2005 0:30	HTTP	1	0.023	102.61693
9	6/20/2005 0:35	HTTP	1	0.137	71.514423
10	6/20/2005 0:40	HTTP	1	0.045	101.96036
11	6/20/2005 0:45	HTTP	1	0.27	131.437
12	6/20/2005 0:50	HTTP	1	0.017	917.64323
13	6/20/2005 0:55	HTTP	1	0.408	78.594645
14	6/20/2005 1:00	HTTP	1	0.191	181.69233
15	6/20/2005 1:05	HTTP	1	0.138	51.649306

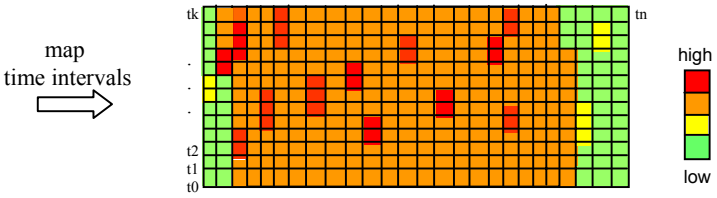
(C) An Internet Network Spreadsheet Fragment
Each column shows a network performance attribute (e.g., DateTime, Network, *Response Time*, *Availability*, *Throughput*…)



(D) An Internet Network *Pixel-Matrix* Spreadsheet
Each pixel-matrix represents a measurement for three different attributes (*Response Time, Availability,* and *Throughput*). Color represents the measurement values. *Response Time* has a close relationship with *Availability* and *Throughput*: 6/20 has a fast *Response Time* (mostly green) and high *Throughput* (red and orange) due to the network high *Availability* (all green).



(E) A Service Contract Time-Series
A fragment of time series: The height of the curve shows the service contract violation level. It is difficult to visualize each service violation value with such cluttered curves.



(F) A Service Contract *Pixel-Matrix* Time-Series
Each pixel-matrix represents a measurement interval. Color represents the service contract violation level (red: violated, orange: warning, green: not violated). Pixel-matrixes are placed from t0 to tn, from left to right, column by column.

Figure 1: Apply *Pixel-Matrixes* in Bar Charts, Spreadsheets, and Time-Series for Visual Analysis
(The pixel-matrix sizes shown above are just for illustration and do not show the actual sizes.)
Three real-world applications are shown in sections (3, 4, and 5)

2. Our Approach

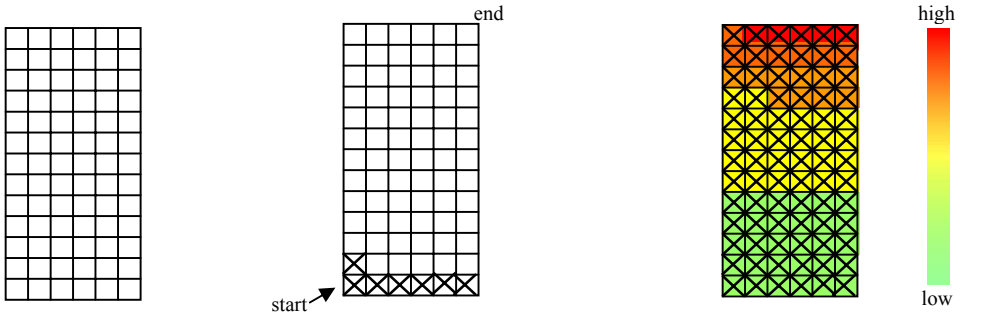
Instead of generating new graphics, we propose to fill up the regular bars, spreadsheets, and time series with colored pixel-matrix to represent the value of every transaction, (e.g., price in a product sales transaction). Previously, we introduced Pixel Bar Charts [14], a way of visualizing detailed transaction data. Now we generalize and extend this idea to ***pixel-matrix*** techniques. Our approach is to represent each data item and measure (e.g., the price of a sales transaction) with a single pixel-matrix to show data distribution, patterns, trends, and exceptions. The charts, tables, and maps generated from the pixel-matrix mechanism allow users to visualize the insight of the graphics at a glance. In addition, we provide an instantaneous display of data values as needed. Users can easily drill into the problem areas for further analysis.

To illustrate our idea, we apply pixel-matrix to bar charts, spreadsheets, and time series for visualizing multi-attribute data in Figure 1 (B, D, and F).

2.1 Layouts

A pixel-matrix is a small rectangle, which has one or more pixels. The color of pixel-matrixes varies based on the measured value (e.g., colors can vary from green to yellow to orange and to red to denote different price values). A pixel-matrix represents one data item. We use the concept of pixel-matrixes not just for bar charts in Figure 1 (A), but also for spreadsheets in Figure 1 (B), and time series in Figure 1 (C).

Pixel-matrixes are arranged from left to right and bottom to top in a graph (as shown in steps 1, 2, and 3 of Figure 2) according to the data values. Each pixel-matrix can be accessed to drill down to the information at the transaction level. In this way, we can analyze detailed information (e.g., price and quantity for each sales transaction). The color distribution of the pixel-matrixes gives us a visual indication of the statistical distribution of the measure in a way that is much more informative then what averages would give, as we are able to visually access the entire value distribution. Furthermore we can quickly spot how many transactions are above acceptable thresholds. By comparing the figures in Figure 1 with regular charts, tables, and time series, the differences in information content and usability are evident.



Step 1: Partition the display area into n pixel-matrixes based on the number of transactions in a dataset.

Step 2: Layout the pixel-matrixes in the graph in the order of the attribute value (price): from left to right and bottom to top.

Step 3: Color-code the pixel-matrixes according to the attribute value (e.g., sales price).

Figure 2: Pixel-matrix Graphical Layout

2.2 Mapping

Several new techniques are derived for mapping colored pixel-matrixes into (1) charts for visualizing data patterns and exceptions, (2) tables for visualizing correlations and root-causes, and (3) time series for visualizing the evolution of long-running transactions.

Figure 1 (B) illustrates a daily product sales pixel-matrix bar chart. It is generated by mapping the product sales transaction data to pixel-matrixes. Each pixel-matrix represents a sales transaction. The color is the value of a transaction (e.g., price) from low to high (green, yellow, orange, red). Pixel-matrix layout is based on the above

layout algorithm from left to right and bottom to top according to the price. The bar height represents the total value of a bar. The analyst can find the product sales price distribution from the color patterns, e.g., the sales on 4/25 have the most red and orange colors (on top of the bar) which indicates that there are many high priced products sold on that day. 4/28 has mostly green so the prices of the most sold products were low.

Figure 1 (D) illustrates how to map multiple attributes (*Response Time*, *Availability*, and *Throughput*) in a single spreadsheet using pixel-matrixes for visual comparison. In the mapping, each pixel-matrix represents a measurement interval. Color represents the attribute value in a measurement interval. The pixel-matrix spreadsheet is able to visualize the full data set without data aggregation, thereby avoiding information loss. For example, 6/20 shows a fast response time in all the measurements (all green except one yellow); servers are all available (green). Thus all the measurements have a high throughput (mostly orange and red). The response time is very slow on 6/22 (mostly orange and red) which is caused by server unavailability (red).

Figure 1 (F) is a pixel-matrix time series which maps the observations in a large time series. Color represents a metric value, such as the *contract violation level* from low (green) to high (red). Pixel-matrixes are placed bottom to top and left to right column by column. The pixel-matrix technique displays the full data set without cluttered curves. There are many service warnings (orange) occurring in the middle time intervals. Some violations occurred (red) before the system fulfilled the contract at the end (green).

Our three visualization techniques and their applications using pixel-matrixes in charts, spreadsheets, and time series are further described in the following Sections. The techniques are implemented in Java and can be presented via portal and web pages. They have been applied to data from sales analysis, Internet network performance analysis, and service contract violation applications.

3. Pixel-Matrix Bar Charts

We have applied the pixel-matrixes to charts to visualize product sales data. In a pixel-matrix bar chart a single pixel-matrix is used to represent a sales transaction. The bars are partitioned based on the product type. Pixel-Matrixes are ordered according to the price along the y-axis within each bar. The bar height represents the number of sales transactions in each product category. Color represents sales price. Users can select different colors to represent different metrics (e.g., *price*, *quantity*, and *discount*). The detailed information of each sales transaction is encoded into the pixel-matrix and can be accessed and displayed. As illustrated in Figure 3, users can easily find product sales distribution, patterns, correlations, and outliers (highest sales).

3.1 Patterns and Exceptions

In Figure 3, bar height represents the number of sales transactions in a bar. Pixel-Matrix colors inside a bar represent sales distribution. Users can analyze the product sales volume and price distributions from the bar height and color patterns. For example, product *BV* has the highest volume (highest bar) and sold many low price products (green, under $300). Product *A1* has the highest price products sold (mostly red and orange, above $3k). Product *A5* has many product returns (indicated by the below-the-zero line, $10k-30k). The sales price range is above $1k (yellow, orange, and red) for almost all the products. Users can move the pointer to display each sales transaction. Figure 3 shows an outlier (the highest sales transaction) appearing at the top of the bar *A1* ($2.535M).

3.2 Correlations

In many cases, the data to be analyzed consists of multiple attributes. With pixel-matrix charts we can visualize multiple attributes using multiple bar charts which use the same layouts but different color mappings. This means that the arrangement of pixel-matrixes (transactions) within the corresponding charts is the same. The attributes which are mapped to color are different. In Figures 3, 4, and 5, we show three attributes (*Price*, *Quantity*, and *Discount*) in three pixel-matrix bar charts each of which has *Product Type* as the x-axis. Pixel-matrixes are arranged according to the *Price* in all bars. The same transaction record has the same relative position within each of the corresponding bars. It is therefore possible to relate the different bar charts and observe the correlations (e.g*., Price and Quantity*, *Price and Discount*).

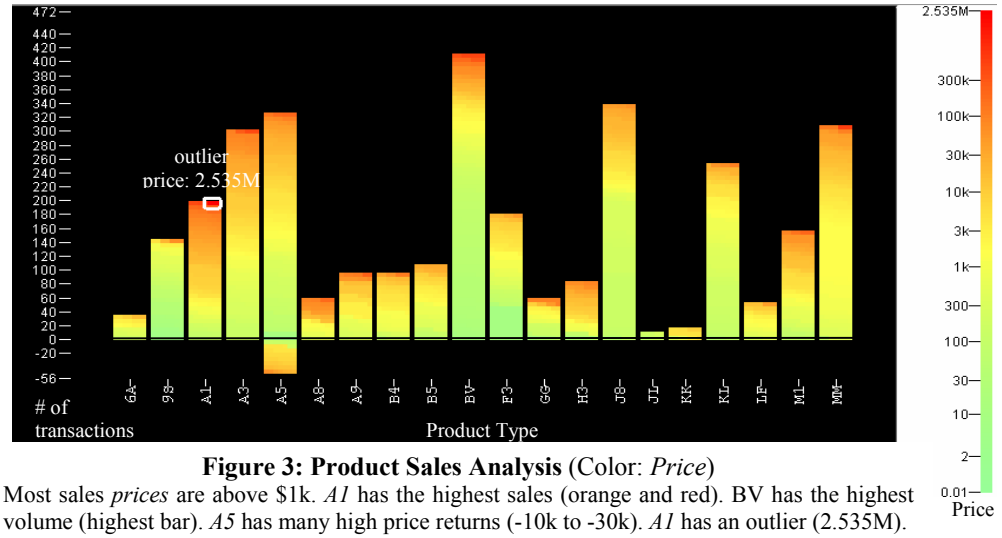


Figure 3: Product Sales Analysis (Color: *Price*)
Most sales *prices* are above $1k. *A1* has the highest sales (orange and red). BV has the highest volume (highest bar). *A5* has many high price returns (-10k to -30k). *A1* has an outlier (2.535M).

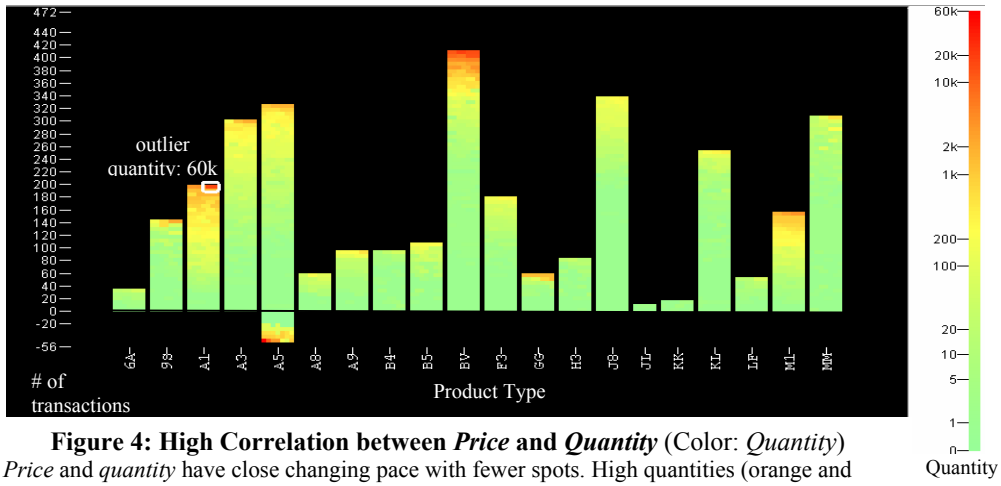


Figure 4: High Correlation between *Price* and *Quantity* (Color: *Quantity*)
Price and *quantity* have close changing pace with fewer spots. High quantities (orange and red) always reside in the high price locations (at the top of a bar) and vice versa. Outlier's quantity is 60k.

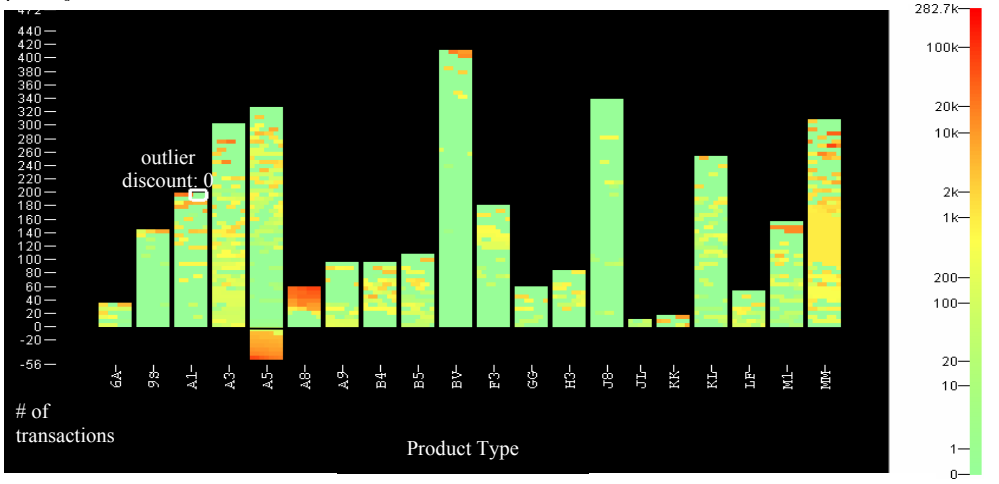


Figure 5: Low Correlation between *Price* and *Discount* (Color: Discount)
Price and *discount* do not have close changing pace with more spots except product *A8* and *BV*. Outlier has zero discounts (green).

From Figure 3, users can observe the distribution of prices for each product type. Product type A1, for example, is much more expensive than product type 9S or BV (mostly red and orange). In Figure 4, the high *Quantity* transactions (above 1k, yellow, orange, and red) always appear at the top of a bar. Users can see by looking at the pattern of colors in the bars that the higher *Quantity* transactions have the same changing pace as the higher *Price* transactions, In the bar *A1,* high *Quantity* transactions (red, above 10k) always reside at the same high *Price* transaction locations (red, 100k). Low *Quantity* transactions (green, 20 orders) always reside at the low *Price* transaction locations (green, 100) at the bottom of a bar. A1 shows the outlier's quantity that is 60k (red, highest quantity).

Spots indicate that the two attributes (e.g., *Price* and *Discount*) do not correlate. Figure 5 shows many spots indicating a low correlation between *Price* and *Discount*. There are many low *Discount* transactions (green) appearing at the top of a bar (high *price* transaction area), and there are many high *Discount* transactions (orange and red) appearing at the bottom of a bar (low *price* transaction area). Therefore, a quick way to find correlations is to compare the number of spots. From comparison of Figures 3, 4, and 5, users can conclude that *Price* is more correlated to *Quantity* than to *Discount* in most of the bars except for bars *A8* and *BV*, which each have a fewer number of spots. Note also that *Price* and *Discount* do not have the same pace of change.

4. Pixel-Matrix Tables

To find relationships among the different attributes in tables, we apply pixel-matrixes to the tables. Data metrics are aligned in a spreadsheet-like row and column format for easy comparison. For hierarchical datasets, we employ different degrees of gray (from light to dark) to show different levels of the data structure. The parent node has a lighter gray than the child nodes. The use of different shadings helps users to view the data structure.

4.1 Multi-Attribute Comparison

Figure 6 illustrates a network performance analysis using pixel-matrix spreadsheets. The dataset contains 55,769 transactions. The parent nodes are three networks (*HTTP*, *Mail Probes*, and *FTP*). Their child nodes are attributes (*Response Time, Availability*, and *Throughput*) which have a darker gray than their parent nodes. We use columns to represent the *date* attribute from 6/20 to 6/27. Each day in a row is represented by a rectangle. A rectangle contains a number of pixel-matrixes. Each pixel-matrix is associated to a web transaction and shows a color representing the metric value of that transaction.

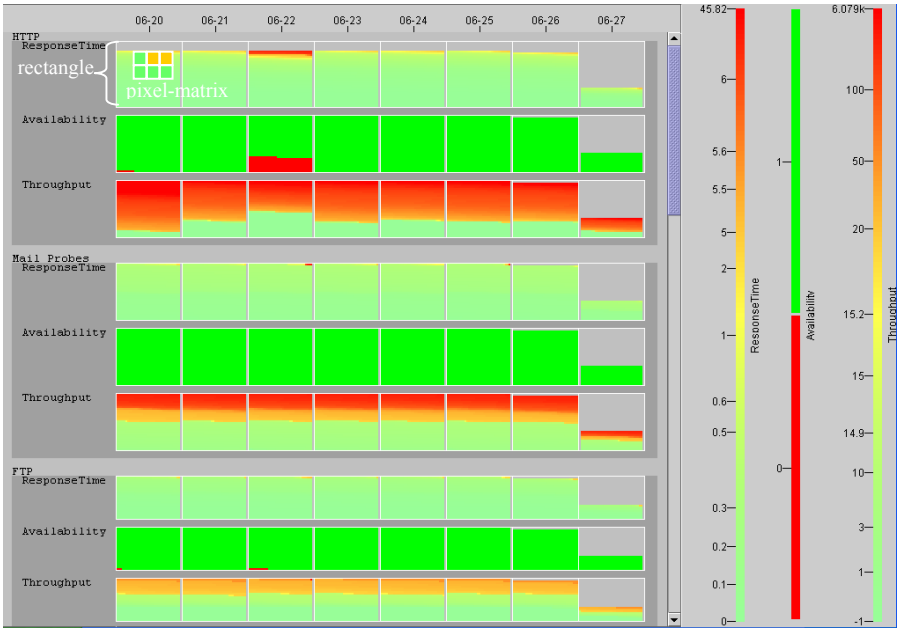


Figure 6: Internet Network Performance Analysis

(Column: date; Rows: network type; Pixel-Matrixes: Internet transactions; Color: R*esponse Time, Availability, and Throughput*)

- *Throughput* depends on *Response Time* and *Availability*, such as 6/20 has a high throughput *(*red and orange), high availability (mostly green) and fast response times (mostly green), and vice versa as shown on 6/22.
- *Mail Probes* has fastest response times (mostly green) and occasionally has some slow response times (red)
- *FTP* has fewer web transactions (shorter rectangles), less throughput (under 20), and faster *response times* (0.2 sec).

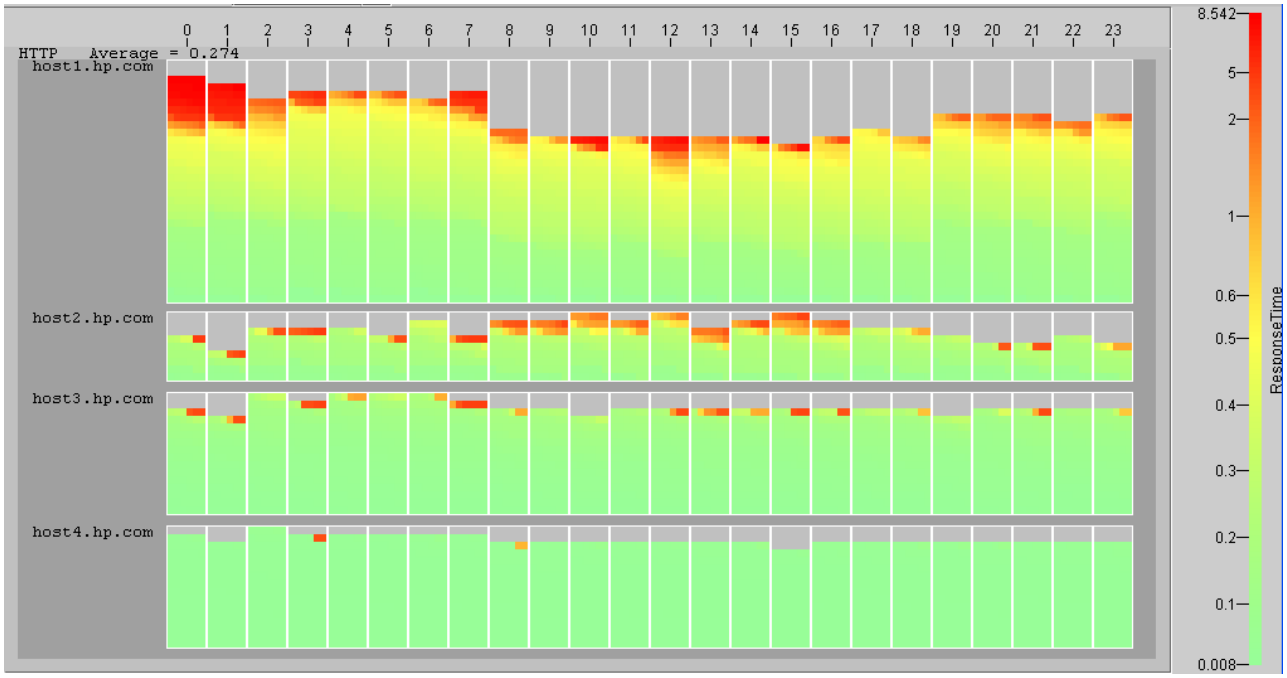


Figure 7: Drilldown from Figure 6 to show HTTP hourly Response Time
(Column: hour; Rows: network hosts; Pixel-Matrixes: Internet transactions; Color: *Response Time*.)

Find the root causes of the HTTP slow response time and low throughput from the following observations:

- Host 1 has slowest response time (many red, orange, and yellow).
- Host 1 has the most number of transactions (the highest bars across all hours).
- Host 2 has the least number of transactions (the lowest bars across all hours).
- Hosts 2 and 3 have a few long response time transactions (red).
- Host 4 has the fastest response time with one exception occurring in the morning at 3 o'clock (red).

Each pixel-matrix is ordered by its corresponding transaction's metric value from left to right and top to bottom inside the rectangle. Users can drilldown from each pixel-matrix to display detailed information on a transaction. Figure 6 shows the colors for *Response Time*, *Availability*, and *Throughput* changing at similar paces. This indicates that the high server availability (green) always generates fast response time (green) and a high throughput (red), such as on 6/20.

4.2 Problem Isolation (Root-Cause Discovery)

Network operation managers typically ask questions such as: How are my network response times today? Which host (server) has a problem? How is my workload? To answer these questions, first, we encapsulate colored pixel-matrixes to a spreadsheet-like table from the network log data shown in Figure 6 that shows an overview of daily network performance. From the overview, the service manager discovers that HTTP has many slow response time transactions (orange and red) on 6/22, and therefore, wants to drill down from Figure 6 to Figure 7 to find the causes.

Figure 7 shows that the long average turn around time in HTTP is due to too many transactions processed on *host1* (highest rectangle). Those transactions have very slow response times (yellow and red). Knowing the cause, the operational manager can take action to offload the transactions running on *host1* to other hosts (e.g., *host2*, *host3*, and *host4*) to improve the performance.

5. Pixel-Matrix Multi-Resolution Time series

In Figure 8, we display a single attribute (violation level) to a pixel-matrix service contract time series thus observing changes over time. For analyzing large complex time series, we employ a multi-resolution visualization technique. This allows the user to focus on DOI (Degree of Importance) using different resolutions, so that users are able to visually analyze the violation level of service contracts across time. Colors represent service contract violation levels from low (0, green), warning (yellow and orange), to severe (1, red). The partition is defined using relative weights for the data and display space. Three partitions (small, medium, and large) are laid out from left to right. Each partition contains increasingly more data (by a factor of 4). This technique allows the visualization to show the most recent data (right partition) at the highest resolution, while maintaining the intermediate and old data in context. This technique is capable of monitoring data streams in real time by advancing data through the display. Figure 8 shows five service contract time series (SOAP, STREAMING, SysBasicWMI, HTTP, and MailRoundTrip) for visualizing the evolution of long-running transactions (from 6/20 00:00 to 6/27 7:45). The user can zoom in on any problem area to a detailed time series chart for further analysis (shown at the left of Figure 8).

Old data:
from: 2005-06-26 14:30
to: 2005-06-20 00:00

Intermediate Data:
from: 2005-06-27 04:30
to: 2005-06-26 14.45

Most Recent Data:
from: 2005-06-27 07:45
to: 2005-06-27 04:45

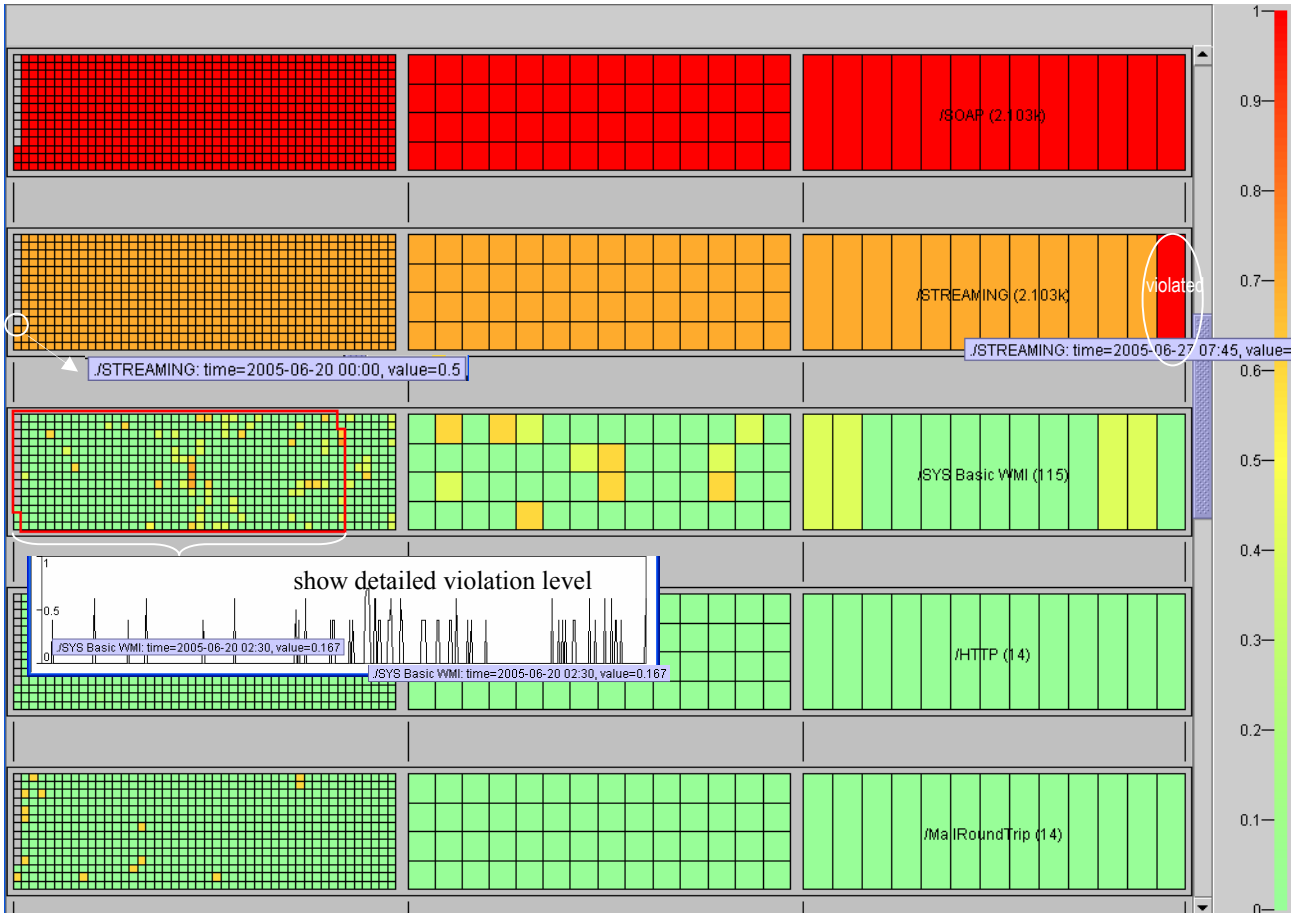


Figure 8: Visual Comparison of Five Customer Time series

SOAP has severe contract violations (all red). STREAMING has all warnings (orange) and one violation (red) at the last time interval (6/27/05, 7:45). SYS Basic WMI has many occasional warnings (yellow); Mail Roundtrip has no violations (green) except a few warnings (yellow) from 6/20 to 6/26 (old data).

6. Conclusion

In this paper, we have presented a new pixel-matrix visualization technique for visually analyzing multi-attribute data. Our approach is to encapsulate colored pixel-matrixes into charts, tables, and time series for visualizing data patterns, correlations, exceptions, and evolution in long-running time series data. This approach differs from the existing techniques by encapsulating regular graphics with color-encoded data values. Therefore, users are can analyze data and detect root-causes without clicking through many charts and listings.

From a recent customer feedback, pixel-matrix techniques were found to be intuitive and easy to use. The data analysis resulting from these techniques is about 50 times faster than the methods used previously for analyzing patterns of complex SQL queries in the customer service center. Also, sales analysts have been able to use pixel-matrix bar charts, spreadsheets, and time series to track down important issues about their data warehouse in a few minutes instead of a few hours (e.g., misplaced invoices and data quality problems). Our applications using real-world sales, network, and services show the wide applicability and usefulness of these new techniques. Further study will focus on the improvements of this technology for use in the area of automated real-time visual analysis.

Acknowledgements

Many thanks to Beth Keer of HP Laboratories for her encouragement and suggestions, to Martha Lyons for her information and comments on service contract analysis, and to Michael Haeuptle from HP OpenView for providing suggestions and Internet network data.

References

[1] C. Stolte, P. Hanrahan: "Polaris: A System for Query, Analysis and Visualization of Multi-dimensional Relational Databases", InfoVis-2000, Salt Lake City, UT, 2000.

[2] Tableau Software, http//www.tableausoftware.com, 2006.

[3] SpotFire, http//www.spotfire.com, 2006.

[4] Ben Shneiderman: Treemaps for Space-Constrained Visualization of Hierarchies, http://www.cs.umd.edu/hcil/treemap-history/, December, 2005.

[5] M. Bruls, K. Huizing, J. van Wijk: Squarified Treemaps, Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization, Salt Lake City, UT, 2000.

[6] H. van de Wetering, J. van Wijk: Cushion Treemaps: Visualization of Hierarchical Information, Proceedings of the IEEE Symposium on Information Visualization, San Francisco, 1999.

[7] D. A. Keim: Designing Pixel-oriented Visualization Techniques: Theory and Applications, Transactions on Visualization and Computer Graphics TVCG2000.

[8] S. G. Eick et al.: *Seesoft*"–*a tool for visualizing line oriented software statistic*s, IEEE Transactions on Software Engineering, November, 1992.

[9] E_Bizinsights, http://www.bizinsights.com

[10] A. Inselberg, B. Dimsdale: *Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry*, Proc.Visualization ´90, San Francisco, CA, 1990.

[11] C. Tominski, J. Abello, C. Schuman: Axes-Based Visualizations for Time Series Data, Proc. IEEE InfoVis 2002, Boston, Massachusetts, 2002.

[12] P. Buono, A. Aris, B. Shneiderman: Interactive Pattern Search in Time Series, Visual Data Analysis Conference, San Jose, CA. 2005.

[13] J. van Wijk, E. Selo: Cluster and Calendar Based Visualization of Time Series Data, Proc. IEEE InfoVis 1999, San Francisco, 1999.

[14] D. A. Keim., M. Hao, U. Dayal, M. Hsu, J. Ladisch: Pixel Bar Charts: A New Technique for Visualizing Multi-Attribute Data Sets without Aggregation, IEEE InfoVis2001, San Diego, CA. 2001.