Visual Analytics Techniques for Large Multi-Attribute Time Series Data

Ming C. Hao, Umeshwar Dayal, Daniel A. Keim*
Hewlett Packard Laboratories, Palo Alto, CA
(ming.hao, umeshwar.dayal)@hp.com
keim@informatik.uni-konstanz.de

ABSTRACT

Time series data commonly occur when variables are monitored over time. Many real-world applications involve the comparison of long time series across multiple variables (multi-attributes). Often business people want to compare this year's monthly sales with last year's sales to make decisions. Data warehouse administrators (DBAs) want to know their daily data loading job performance. DBAs need to detect the outliers early enough to act upon them. In this paper, two new visual analytic techniques are introduced: The cell-based Visual Time Series highlight significant changes over time within complex data sets and the new Visual Content Query facilitates finding the contents and histories of exceptions, which leads to root cause identification. We show examples of using these techniques to mine customer credit card fraud data to illustrate the wide applicability and usefulness of these techniques.

Keywords: visual analytics, multi-attribute data, time series, visual content query, contents and relationships

1. Introduction

Because of accelerating technological progress, the amount of data stored in computers has increased rapidly. Today, computers typically record transactions of everyday life, such as paying by credit card, using the telephone, and shopping in e-commerce stores. This data is collected and stored in data warehouses because business people believe that time series data is a potential source of valuable information and provides a competitive advantage. Most of the data typically contains a time attribute, such as the date and time of a telephone call or a credit card purchase. Usually, time series data are inherently large and have many different attributes [KK94], such as a stream of sales data with attributes like product type, location, sales amount, quantity, price, etc. The current analyses of time series only focus on observing changes in a single value (e.g., sales amount). This paper shows how to successfully arrange multiple time series with different attributes for visual comparison and problem discovery.

A common technique for users to get detailed information about time series data is through drilling down to the next category. But, users are sometimes also interested in the content (value) of related attributes. In a sales application, for example, the sales managers want to know purchasing behavior of the top customer and related information. In a fraud application, the financial analysts want to examine the historical evolution of top fraud transactions and determine relevant attributes such as related locations and credit card types. This paper addresses how to query interesting attribute contents and relations for further identification and analysis of problems.

2. Related Work

Research in the area of time series visualization has been published under a variety of subjects in the recent past. There are many well-known techniques developed for various applications. The Polaris [STH02] system, for example, allows the analyst to pivot and refine visual specifications of table-based graphical displays. Schuman [TAS02] employs a TimeWheel that presents the time axis in the center of the display and circularly arranges the variables around the time axis. Van Wijk [WS99] introduced a clustering-based visualization to condense multiple time series data into a calendar-based view. Shneiderman's [BAS05] interactive pattern search provides fast visualization methods to retrieve information from large time series. Ward [CW06] uses histogram and nearest neighbor methods to measure data abstraction quality in multi-resolution visualization. Users can use both methods to evaluate how well the abstracted dataset represents the original data set. Gautam Kumar [KG06] structures dense graphs via stratification using a force-directed layout algorithm to visualize U.S. Stock (1990-2005) financial correlations.

* Presently with University of Konstanz, keim@informatik.uni-konstanz.de

Different from the above methods, we apply familiar visual metaphors to layout multiple time series data in a spreadsheet-like layout for visual comparison. We adopt color cells to represent the data values in a time interval to allow users to visualize information at the detailed transaction record level. To help the user find information related to some data items of interest (e.g., outliers or exceptions), we introduce the concept of Visual Content Query. Our techniques employ automated data mining methods and adapt them to the human ability to quickly analyze visual patterns, find relationships, identify exceptions, and retrieve related information.

3. Basic Ideas

Appropriate visualization of time-series data is a valuable tool for exploring previously unknown information and searching for interesting patterns. The common approach using bar charts and line charts is ineffective for visual analysis of time-series data: Given the limits of current display devices, we either have to accept overplotting (occlusion) effects in the display or we heavily depend on scrolling interaction. Both effects are not optimal regarding usability effectiveness. A common technique to solve the problem is to reduce the data size by sampling or aggregation. These techniques may introduce a loss of information. Therefore, for the exploration of large multi-attributes data, these techniques are of limited value and do not show important information such as

- data distributions of multiple attributes;
- patterns, correlations, trends, and exceptions; and
- detailed information, e.g., the purchase history and behavior of a top customer

To achieve the above goals, we have created two innovative visual analytics techniques and developed Visual Time Series (VisTS) and Visual Content Query (VisConQ). Both VisTS and VisConQ visualization techniques are implemented in Java and can be presented via portal and web pages. They have been successfully applied to data warehouse, customer services, and credit card fraud analysis.

3.1 Use of Visual Time Series to See Data Changes Over Time (Patterns, Trends, and Exceptions)

To show both aggregate and detailed information in the time series, we fill up each time interval with colored cells to represent metric values for individual transactions. Previously, we had introduced pixel bar charts [KHD00] as a way of visualizing detailed transaction data. In this paper, we generalize this idea to a color cell-based techniques. In the pixel bar chart technique, each data item (transactions/events) is represent by a single colored pixel. The color cell-based technique generalizes to allow one color cell either to represent mutiple data items or a single data items. Each cell can be accessed and drilled down to the information at the record level. Cells are arranged from bottom to top and left to right according to the metric values. Colors vary based on the metric value (e.g., from green, yellow to red to denote different metric values). The color cell distribution is much more informative than averages or variances, as we are able to assess the entire value distribution at a glance. By comparing the VisTS with a regular time-series, the difference in information content and usability becomes evident.

3.2 Use of Visual Content Query to Discover Related Information

Most time series charts show only aggregated values. The usefulness of the charts is especially limited if the user is interested in finding exceptions, such as an extremely slow data warehouse loading job. The Data Warehouse System Administrator (DBA) wants to know how slowly this job executed previously.

A Visual Content Query in this paper is different from an ordinary database query. Most visualization tools, such as the well-known Tableau's Polaris system [SH00], for example, allow users to generate visual representations from relational data interactively. The user can select single dimensions or measures from the underlying relational data via drag and drop. From the user's selection, the Polaris system constructs different visual representations, such as line charts, scatter plots, and bar charts.

The basic idea of VisConQ is to allow the DBA to select a data item on the graph and query its contents. Using visual time series, the exceptions are highlighted. The DBA can click on an outlier and select an attribute value on which VisConQ performs attribute content association, retrieves related information, and lays out visual presentations in a content query result window. The DBA can easily view and interact with the information regarding the selected content. This functionality is not available with the other currently available visualization tools.

4. Visual Time Series Layout

Laying out a Visual Time Series means to transform the spreadsheet-like data to present multiple time series in an aligned hierarchical graph. The user specifies a hierarchical structure of a time series using time (column), levels (rows), and metrics (colors). VisTS constructs a tree to represent the hierarchy as illustrated in Figure 1.

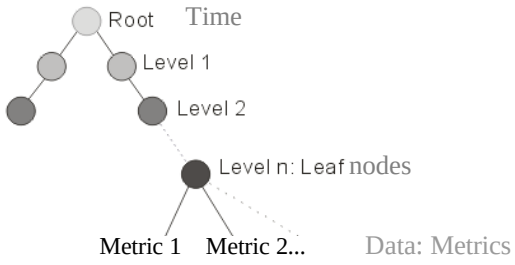
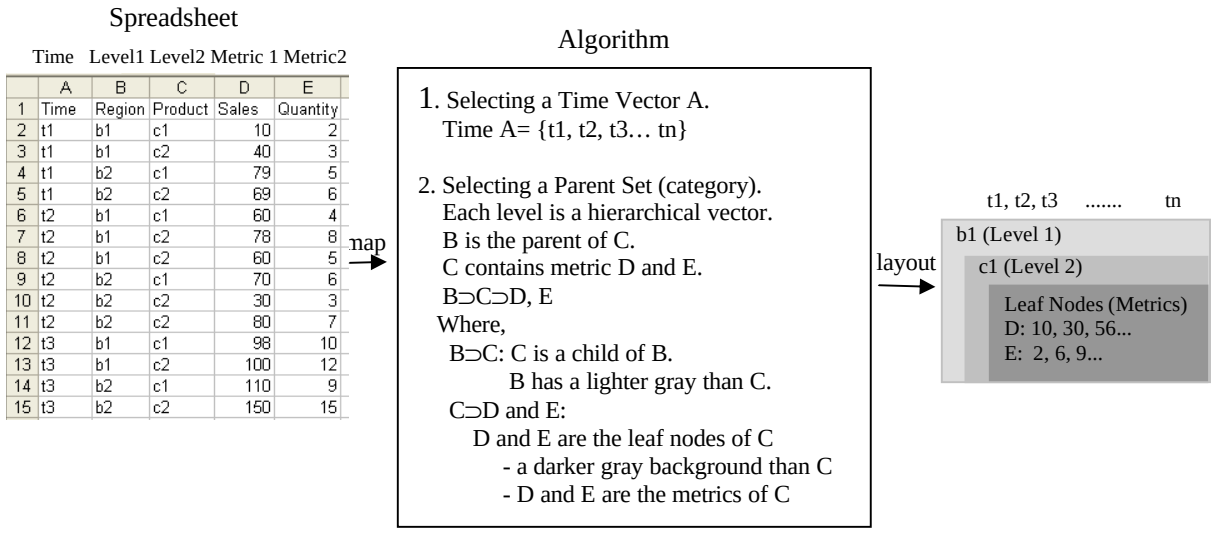


Figure 1: A Tree Layout

For a general layout of a visual time series, we need to specify:

1. Time Vector
   The user selects a time attribute from a spreadsheet column (see Figure 2, Column A) as a time vector.
2. Level Vectors
   The user selects level vectors (categorical attributes) from the spreadsheet category columns to define a hierarchical structure, such as level 1 (Column B) is the parent of level 2 (Column C).
3. Data Vectors (Metrics)
   The user selects data attributes (metrics) from the spreadsheet metric columns, such as Columns D and E. The graphs for the leaf nodes are filled with color cells.
4. Color
   The color of a cell is the value of a metric, such as column D: Sales value (i.e., 10, 40...).



	A	B	C	D	E
1	Time	Region	Product	Sales	Quantity
2	t1	b1	c1	10	2
3	t1	b1	c2	40	3
4	t1	b2	c1	79	5
5	t1	b2	c2	69	6
6	t2	b1	c1	60	4
7	t2	b1	c2	78	8
8	t2	b1	c2	60	5
9	t2	b2	c1	70	6
10	t2	b2	c2	30	3
11	t2	b2	c2	80	7
12	t3	b1	c1	98	10
13	t3	b1	c2	100	12
14	t3	b2	c1	110	9
15	t3	b2	c2	150	15

Figure 2: Visual Time Series Construction

Time series data are usually visualized by line charts (Figure 3A). In Figure 3B, we show cell-based visual time series line charts, a simple form of VisTS. Figure 4B shows a cell-based visual time series maps, a generalization of cell-based line charts. The basic idea of VisTS is to use a cell to represent each data item and a time interval area to represent a number of data items. The size of a time interval area grows with the number of data items in the interval. Cells are ordered by an attribute to show data distributions. VisTS retains the intuitiveness of a regular time series while allowing very large datasets to be visualized at a glance.

4.1 Visual Time Series Line Charts

In a data warehouse loading operation, the DBA employs regular time series line charts to provide accurate data to users. To manage data quality, an important question is how the error rate develops over time. Is it persistent or occasional? But a regular time series line charts often results in occlusion as shown in Figure 3A, where only 8 of 38 time series are displayed.

The visual time series line chart shown in Figure 3B is constructed by an x-ordering attribute (month-day), a y-ordering attribute (average job error rate), and a color attribute (job error rate) for each time interval (one day). Figure 3B shows the error rates of the top 18 jobs. In the cell-based visual line charts, each time series is represented by a row of cells. Each cell represents one measurement and is colored by the error rate. This allows the changes over time of all jobs to be easily captured and immediately understood. The first job has a persistent high error rate after 8/30 (all red). The error rate of the third job is occasional, first increasing to red on 8/30, and then decreasing to green after 9/10.

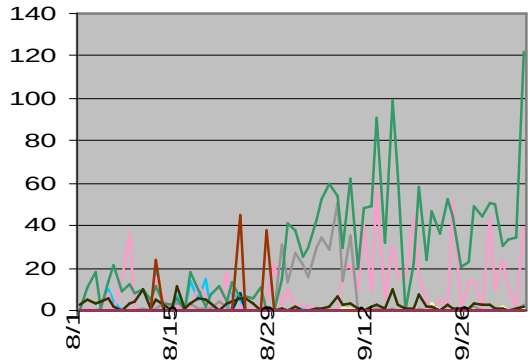


Figure 3A: Time Series Line Chart (38 jobs; only 8 jobs are displayed, x-axis: month-day; y-axis: #error/sec)

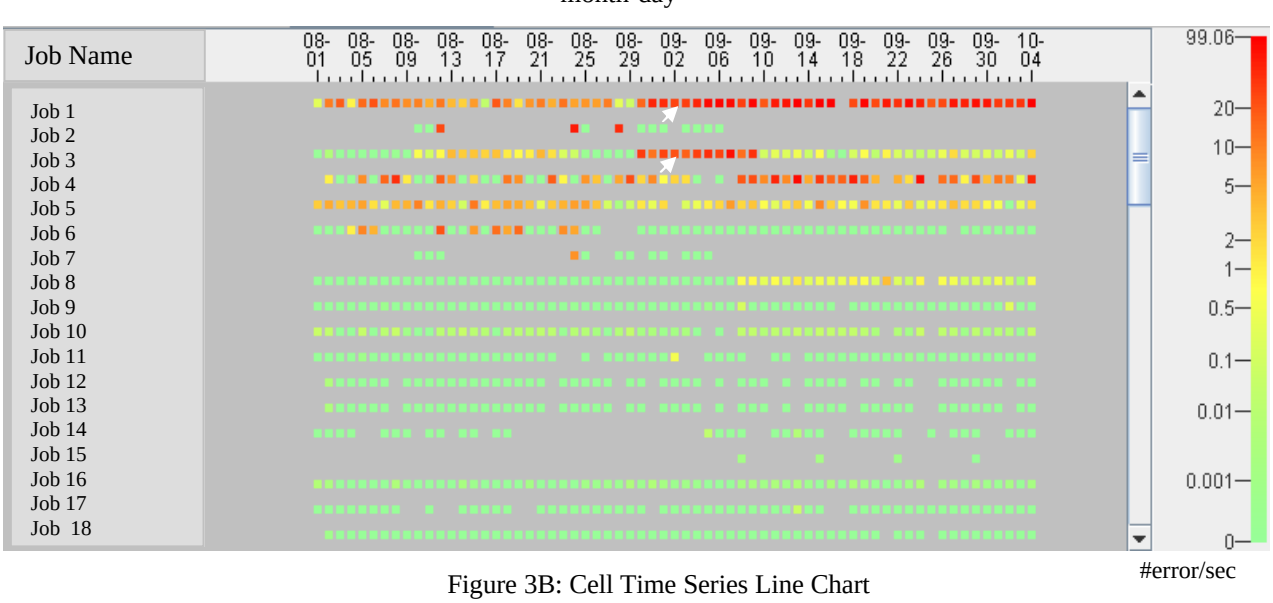


Figure 3B: Cell Time Series Line Chart

4.2 Visual Time Series Maps

To find patterns and exceptions of time series data, we employ cell-based visual time series maps. In Figure 4A, each cell represents a job. The color represents the job-loading rate in seconds at each time interval. The height of a bar represents the number of jobs executed on that day. 8/2 has the highest number of jobs (highest bar). In Figure 4B, the DBA can easily see the job execution patterns and notice an outlier with slowest loading rate on 7/21. Figure 4B also shows that most jobs executed in a day have a short loading rate (green), but a few long loading rates (yellow and red) appear for all days.

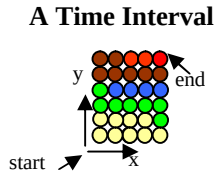


Figure 4A: Color Cell-Based Visual Time Series Map Construction

- Each cell represents a job.
- Cells (jobs) within a day are ordered from left to right then bottom to top according to the attribute value (job loading rate).
- Color is the value of an attribute (job loading rate).

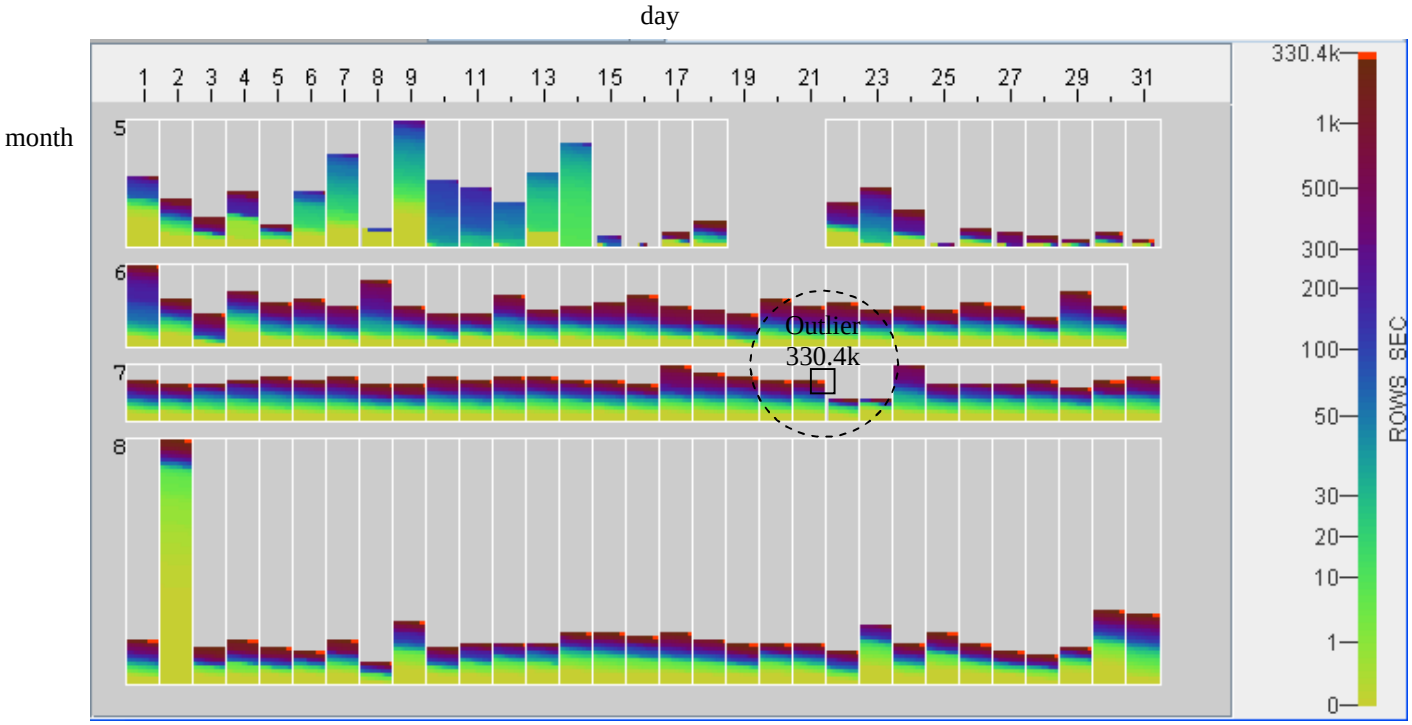


Figure 4B: Daily Loading Job Visual Time Series Map (24,953 observations)

5. Visual Content Query

Visualization of large multi-attribute time series data sets often reveals interesting information, such as dependencies and exceptions. Visual Content Query is designed for analyzing the contents and relationships of the most interesting data shown in a visualization. For example, to analyze the long-loading outlier data highlighted in Figure 4B.

5.1 The VisConQ Process

VisConQ consist of a five-step process as shown in Table 1 with pseudo code. Queries can be performed recursively to narrow down or widen up the searching scope. The result of a query is a set of interactive visual representations generated from the content associated information for users to view and refine the results.

```
Input: Column: attribute, Row: content
       Level:  the current time series drilldown level

// for each column(Attribute) a1, ..., aN, create ActionListener
      for (final Column c : vmdata.column) {
         ActionListener listener = new ActionListener() {
            public void actionPerformed(ActionEvent actionEvent) {
               contentQuery.findValue(c, row);
            }
         };
      }
//Drill down to all values where column has the value of
// column[row].   column:   column to search for the value
//                row:      column[row] is the search value
    public void findValue(Column column, int row) {
       // build content association matrix
       level.tuple = new int [count];
       // put matching row number into the array
       count = 0;
           for (int i=0; i<column.size(); i++) {
               if (column.compare(i, row) == 0)
                   level.tuple[count++] = i;
               }
           }
// display new content query result window
           display(level)
```

Five-Step Process

1. Click on an interesting cell (i.e., outlier).
2. Select a value (content) from the pulldown.
3. Based on the characteristics of the selection, VisConQ retrieves associated data items and relationships that match the selected attribute content.
4. Result is stored in a Content Association Matrix.
5. Layout various interactive visual representations from the content association matrix as needed.

Table 1: Visual Content Query Pseudo code and the Five-Step Process

5.2 A VisConQ Example

To manage data warehouses, DBAs are interested in knowing “Which job has the longest loading rate? How many times does this job need to run? How do these jobs perform between 5/31 and 8/31?

To answer these questions, the DBA first identifies that L4_DAILY is the slowest job; then the DBA needs to perform VisConQ on job L4_ DAILY. In Figure 5A, the DBA selects the content (L4_ DAILY) from the attribute (SUBJECT) and performs a VisConQ to find the L4_ DAILY jobs from 5/31 to 8/31 (Figure 5B). VisConQ has constructed the query result in a Content Association Matrix as shown in Figure 5B and displayed the content query result in a graphical representation as shown in Figure 5C. Figure 5C shows that most L4_ DAILY jobs have very slow loading times (most red and orange) for a period of over three months. 7/21 has the slowest loading rate. L4_ DAILY ran many times on most days. On 6/9 it ran only once (one yellow cell). 6/22 and 8/19 have the highest number of L4_DAILY jobs running (highest bar). With the above information, the DBA may take some action to improve the L4_DAILY loading rate. Empty spaces in Figure 5C mean that no L4_DAILY ran on those days, e.g., 6/3.

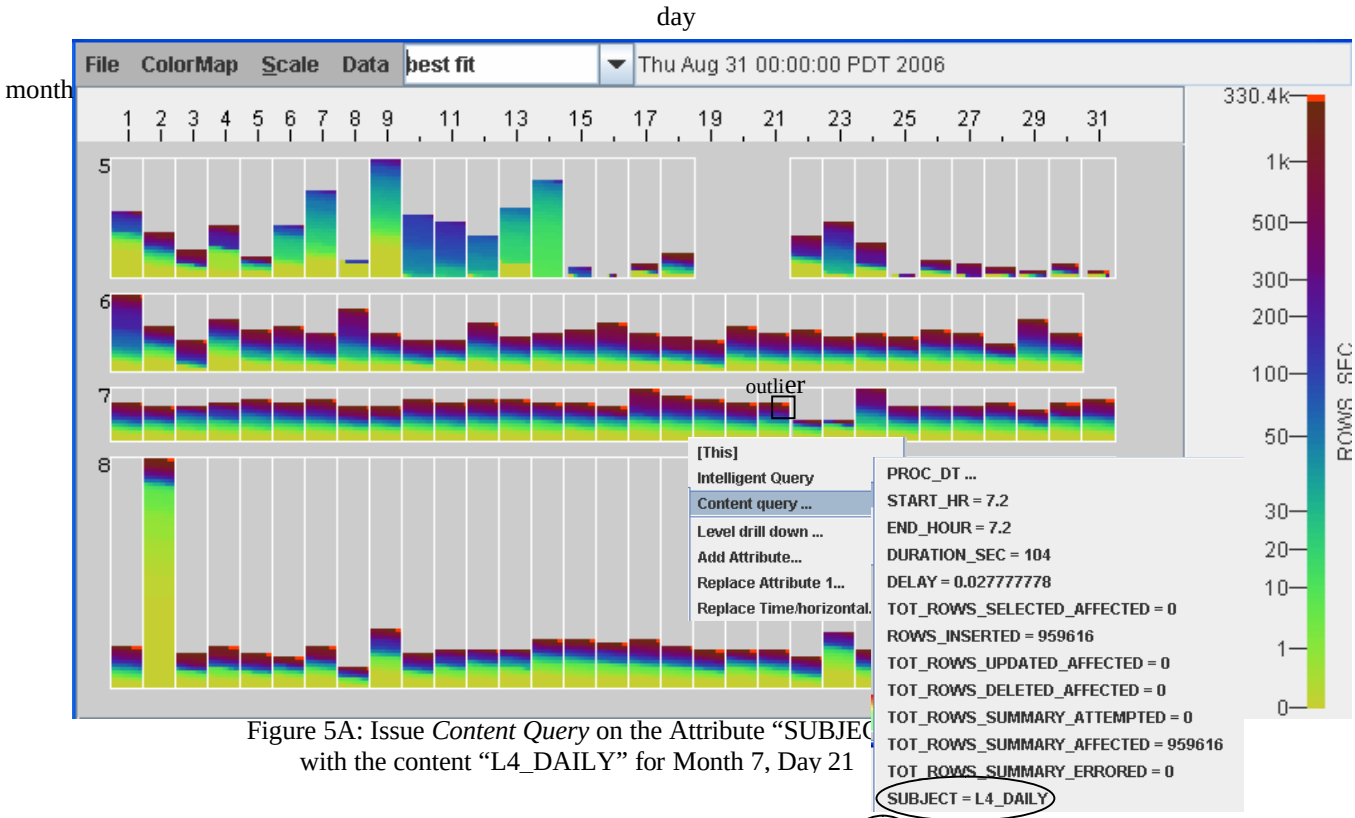


Figure 5A: Issue *Content Query* on the Attribute "SUBJEC
with the content "L4_DAILY" for Month 7, Day 21

PROC_DT	STA...	END...	DURA...	DELAY	ROWS_INS...	TOT_RO...	TOT_ROW...	SUBJECT	TOT_ROW...	TOT_RO...	ROWS_INS...
Fri Jul 21 00:00:00 PDT 2006	6.6	7.1	1736	0.481	142064352	0	142064352	L4_DAILY	142064352	0	142064352
Fri Jul 21 00:00:00 PDT 2006	6.6	10.9	15242	4.162	0	0	0	L4_DAILY	0	0	0
Fri Jul 21 00:00:00 PDT 2006	7.1	7.2	430	0.119	142064352	0	142064352	L4_DAILY	142064352	0	142064352
Fri Jul 21 00:00:00 PDT 2006	7.1	7.2	271	0.073611111	959616	0	959616	L4_DAILY	959616	0	959616
Fri Jul 21 00:00:00 PDT 2006	7.2	7.2	104	0.027777778	959616	0	959616	L4_DAILY	959616	0	959616
Fri Jul 21 00:00:00 PDT 2006	10.9	10.9	86	0	52824	0	52824	L4_DAILY	52824	0	52824
Fri Jul 21 00:00:00 PDT 2006	10.9	10.9	206	0	0	5756168	5756168	L4_DAILY	5756168	5756168	0
Fri Jul 21 00:00:00 PDT 2006	10.9	10.9	117	0.001111111	52824	0	52824	L4_DAILY	52824	0	52824
Fri Jul 21 00:00:00 PDT 2006	10.9	10.9	113	0.030277778	1889	0	1889	L4_DAILY	1889	0	1889
Fri Jul 21 00:00:00 PDT 2006	10.9	10.9	54	0	1889	0	1889	L4_DAILY	1889	0	1889
Fri Jul 21 00:00:00 PDT 2006	10.9	10.9	17	0.001388889	0	1	1	L4_DAILY	1	1	0

Figure 5B: Construct a new Content Association Matrix to associate all transaction records that have the attribute values matched to the query content (L4_DAILY) shown for partial section of data

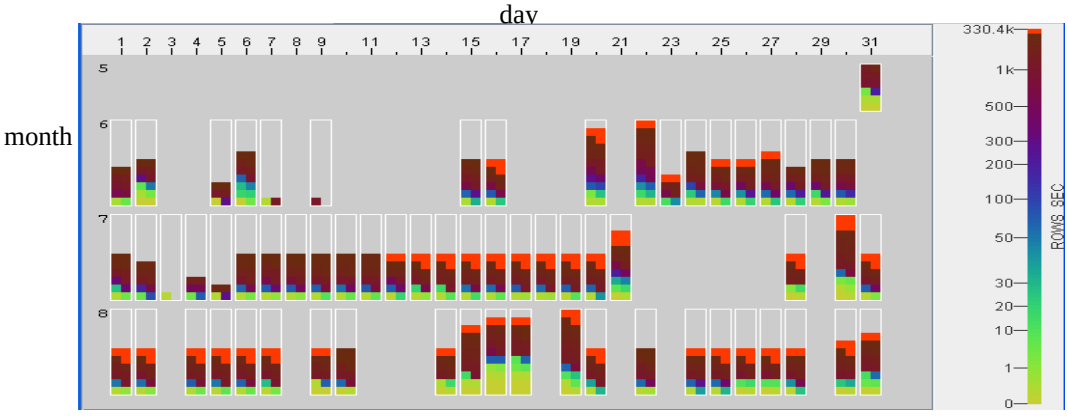


Figure 5C: Layout the query results in SUBJECT L4-DAILY Loading Rate Graph
L4-DAILY has a history of long loading rate

6. An Application Example

We have experimented with Visual Time Series Maps and Visual Queries for sales analysis, service contract analysis, and credit card fraud analysis using real business data. These applications show the wide applicability and usefulness of VisTS and VisConQ. Credit card fraud analysis is reviewed in detail below.
We have conducted experiments in fraud analysis to discover patterns, relationships, and history over two years of time series data. Examples of our analysis include the following:

1) What is the growing rate of credit card fraud over the last two years?
2) Which country has the top rate of fraud over time?
3) What is the history of a top fraud country?
4) Which credit cards do people in this country use? Which credit card has the highest fraud in this country?

6.1. Fraud Visual Time Series (to find patterns, trends, and exceptions)

To observe the historical evolution of fraud transactions, Figure 6 compares quarterly distributions of fraud. In the map, Purchase Quarter is the Time attribute, Region is the Hierarchy Level attribute, and Amount and Count are the leaf nodes (metrics). The graph in the leaf nodes uses a color cell-based representation. Each cell represents a fraud transaction. The colors represent the Amount and Count of each fraud transaction from low (green) to high (red) in a time interval. Each interval contains the number of fraud transactions occurred in the quarter.

From Figure 6, the analyst can quickly determine that the year 2001 has a higher Amount and Count (red and orange) than the year 2000. The fraud rate has grown slightly from 2000 to 2001. There is a high correlation between Amount and Count. Regions are ordered by fraud amount from top to bottom in the Visual Time Series Map. Region 6 resides on the top of the map and has the highest Amount and Count (top red and orange values). Region 2 resides at the bottom of the map and has the lowest Amount (most green).

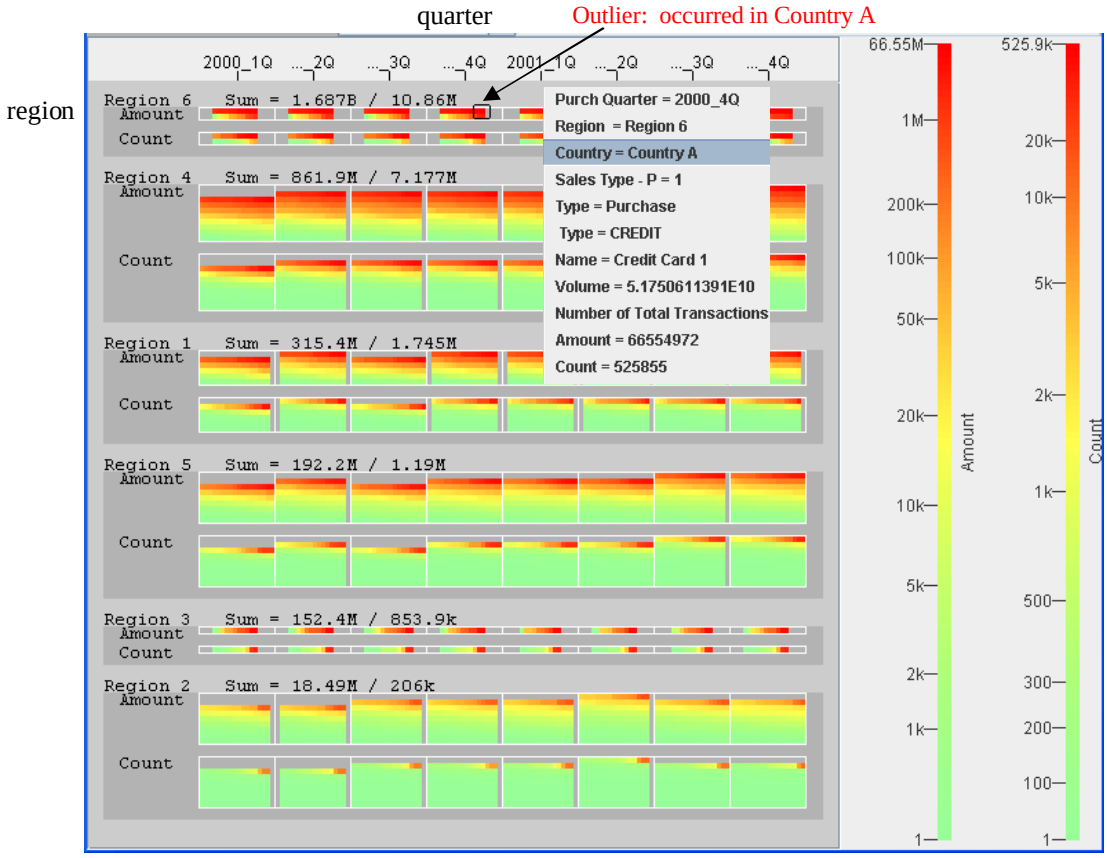


Figure 6: A Quarterly Credit Card Fraud Visual Time Series Map

6.2 Fraud Visual Content Query

To address the questions raised in the beginning of section 6, the analyst first identifies the outlier in Figure 6 (Country A, 2000_4Q). Then the analyst performs VisConQ and finds that this outlier has the highest Amount ($66,554,972) in Country A. Then the analyst selects Country A to find information which is associated to Country A. As a result of the query, VisConQ builds the Content Association Matrix shown in Figure 7A. From the matrix, the two visual time series maps shown in Figures 7B and 7C are generated and displayed to answer the questions.

Purch Qu...	Region	Country	S...	Type	Type	Name	Volume	Number of T...	Amount	Count	BIN Issu...
2000_4Q	Region 6	Country A	2	Cash	DEBIT	Credit Card 2	481,760,478	4742806	6622	8	District 1
2000_4Q	Region 6	Country A	2	Cash	DEBIT	Credit Card 5	3,061,605,865	34305847	55860	97	District 1
2000_4Q	Region 6	Country A	1	Purchase	DEBIT	Credit Card 5	3,527,397,184	73023226	3086274	23280	District 1
2000_4Q	Region 6	Country A	1	Purchase	CREDIT	Credit Card 2	4,936,135,083	44436574	5944238	34386	District 1
2000_4Q	Region 6	Country A	1	Purchase	DEBIT	Credit Card 1	41,790,307,707	1096028889	21167766	200588	District 1
2000_4Q	Region 6	Country A	1	Purchase	CREDIT	Credit Card 8	25,848,955,702	308197913	40326240	209300	District 1
2000_4Q	Region 6	Country A	1	Purchase	CREDIT	Credit Card 5	39,236,652,234	470529915	64884678	360953	District 1
2000_4Q	Region 6	Country A	1	Purchase	CREDIT	Credit Card 1	51,750,611,391	761114279	66554972	525855	District 1

Figure 7A: Country A: Content Association Matrix
(Contains all the information associated to Country A)

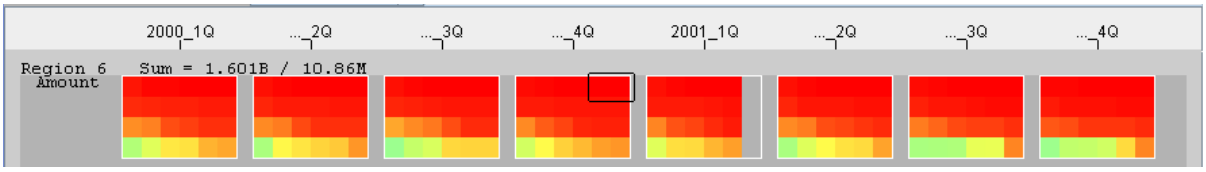


Figure 7B: Country A has the top rate of fraud over 2000-2001 (most red)
(High Fraud History with a highest amount occurred in 2000_4Q (rectangle))

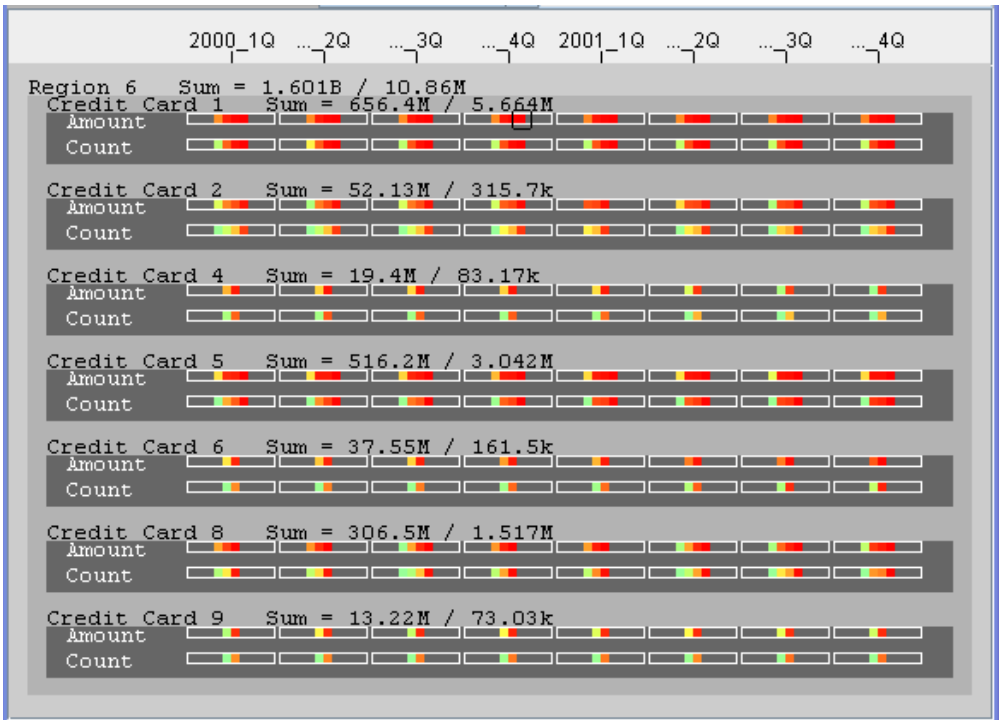


Figure 7C: Country A Credit Card Usage Analysis
People in Country A use seven types of credit cards-Credit Card 1, Credit Card 2, etc.
Credit Card 1 has the highest amount.
(at the top of the map, mostly red and burgundy)
Both 2000 4Q and 2001 4Q have higher frauds than the other quarters (red and pink).

Using the above information, the company is able to place strict control on certain countries, such as Country A, and certain credit cards, such as credit card 1. After identifying the sources of the fraud, the company will be able to take preventive action.

7. Conclusion

In this paper, we identified the current need for visualizing multi-attributes time series and simultaneously finding the associated information, relationships, and patterns over time. We presented the Visual Time Series and Visual Content Query techniques, two new methods that transform spreadsheet-like data into hierarchical visual time series charts and maps. To speed up the visual comparisons, time series are ordered and aligned according to their weight criteria, such as total, average, maximum, or correlation. To expedite the data analysis process, we added a new content query mechanism that constructs maps related to the content (value) of attributes beyond the ability to query on a data category. Our experimental studies show that the Visual Time Series techniques have significant advantages when discovering patterns, trends, and exceptions. Using Visual Content Query, data warehouse administrators can easily find related information on a selected data item, such as the history of a top fraud country for a period of time. Our future work will add multi-resolution and correlation algorithms to place multiple time series in one single display. Also, we will apply the Visual Map technique to other applications such as database query management and optimization.

Acknowledgements

Many thanks to Mei-Chun Hsu of HP Laboratories for her encouragement and suggestions and to Jörn Schneidewind from the University of Konstanz, Germany, for his initial work on Visual Maps. We also thank Rod Watson from HP-IT and Manish Bhardwaj from the HP consulting and services division for providing their comments and data.

References

[CW06] Q. Cui, M. Watthew: Measuring Data Abstraction Quality in Multiresolution Visualizations, IEEE Symposium on Information Visualization2006, Baltimore, MD.

[KG06] G. Kumar, M. Garland: Visual Exploration of Complex Time-Varying Graphs, IEEE Symposium on Information Visualization2006, Baltimore, MD.

[BAS05] P. Buono, A. Aris, B. Shneiderman: Interactive Pattern Search in Time Series, Visual Data Analysis Conference, CA. 2005.

[STH02] C. Stolte, D. Tang, P. Hanrahan: Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases, IEEE Symposium on Information Visualization 2002.

[TAS02] C. Tominski, J. Abello, C. Schuman: Axes-Based Visualizations for Time Series Data, IEEE Symposium on Information Visualization 2002.

[WS99] J. van Wijk, E. Selo: Cluster and Calendar Based Visualization of Time Series Data, IEEE Symposium on Information Visualization 1999.

[SH07] C. Stolte, P. Hanrahan, "Polaris: A System for Query, Analysis and Visualization of Multi-dimensional Relational Databases", IEEE Symposium on Information Visualization 2000, Salt Lake City, UT.

[KHD00] D. A. Keim, M. Hao, J. Ladisch, M. Hsu, U. Dayal: Cell Bar Charts: A New Technique for Visualizing Large Multi-Attribute Data Sets without Aggregation, IEEE Symposium on Information Visualization2000, San Diego, CA.

[KK94] D. A. Keim, H. P. Kriegel, "VisDB: Database Exploration Using Multidimensional Visualization, IEEE Computer Graphics and Applications, Sept. 1994.

[TAB06] Tableau Software, http//www.tableausoftware.com, 2006.