User Interfaces for the Exploration of Hierarchical Multi-dimensional Data

Mark Sifer[1]
School of Economics & Information Systems
University of Wollongong, Australia

ABSTRACT

A variety of user interfaces have been developed to support the querying of hierarchical multi-dimensional data in an OLAP setting such as pivot tables and more recently Polaris. They are used to regularly check portions of a dataset and to explore a new dataset for the first time. In this paper, we establish criteria for OLAP user interface capabilities to facilitate comparison. Two criteria are the number of displayed dimensions along which comparisons can be made and the number of dimensions that are viewable at once—visual comparison depth and width. We argue that interfaces with greater visual comparison depth support regular checking of known data by users that know roughly where to look, while interfaces with greater comparison width support exploration of new data by users that have no apriori starting point and need to scan all dimensions. Pivot tables and Polaris are examples of the former. The main contribution of this paper is to introduce a new scalable interface that uses parallel dimension axis which supports the latter, greater visual comparison width. We compare our approach to both recent table based and parallel coordinate based interfaces. We present an implementation of our interface SGViewer, user scenarios and provide an evaluation that supports the usability of our interface.

CR Categories: H.5.2 [User Interface]—Graphical user interface.

Keywords: Data exploration, OLAP, visualization, parallel coordinates.

1 INTRODUCTION

Concerns about data integrity and update performance have driven much database research, while user interfaces were often an add-on. Updates and queries are often done directly by applications, or via a standard language like SQL for ad-hoc queries, or via tools such as Query By Example that translate text queries to SQL. However, with On-Line Analytic Processing (OLAP) systems [11] there has been a reversal of concerns. Typical OLAP data does not change, as it is usually historical, rather a major concern is supporting the ad-hoc exploration of the data by an analyst or other users looking for trends or patterns at varying levels of detail, perhaps integrated with decision support applications or with data mining heuristics to show or locate results [7,9].

Two key requirements for OLAP systems are: (i) support for many dimensions, often four or more and (ii) scalability. A typical sales dataset can have time, product, location and sale staff dimensions, while a customer dataset could have age, sex, location, income, and household type. Four and five dimensions respectively. A sales dataset for a large national retailer over a one year period could reach billions of transactions.

A data cube of facts combined with dimension hierarchies is the standard model for OLAP data, which is often generated from relational data organised in a star schema. A data cube can be queried or restricted by slicing and dicing dimensions. It can be aggregated or deaggregated through roll-up and drill-down operations, while views of the cube can be altered via rotations. The standard interface for exploring data cubes is the pivot table [10], a multi-dimensional spreadsheet.

An interactive data exploration session may have specific goals or it may be open ended. A specific goal could be to lookup or compare particular trends or distributions such as how sales of a new product have been changing in different regions over the last few months. The goal of an open ended session could be to find unusual or interesting features in a dataset by surveying the data at increasing levels of detail followed by more detailed exploration of any features found. Finding features is also a goal of many data mining systems, but in this paper we are concerned with achieving it via an interface that provides an interactive visualisation.

Existing table based OLAP interfaces such as pivot-tables are appropriate for the former case. Table row and column axis can be chosen according to the dimensions of most interest and trends and distribution can be looked-up via the row and column dimension scales. Open ended exploration ideally requires an interface where all dimensions and all data are initially displayed so the data can be surveyed to determine where to look further. It also requires an interface that supports further analysis with look-up and comparison of proximate trend and distribution operations.

This paper presents an interface that uses parallel axis like parallel co-ordinates [8], designed for the latter requirements of open ended exploratory data analysis of the hierarchically structured multi-dimensional data found in OLAP systems.

Section 2 presents two interface styles (i) table based interfaces starting with scatterplots, then tables, pivot tables and Polaris and (ii) an orthogonal approach--parallel co-ordinates. Section 3 introduces our implementation. Sections 4 and 5 present our interface's parallel tree design and additional user scenarios. Section 6 presents an evaluation summary. Additional related work and conclusions are then given.

2 EXISTING APPROACHES

We review table based and parallel coordinate approaches for exploring multidimensional data.

2.1 Table-based Approaches

2.1.1 The Displays

One way to present multi-dimensional data is as points in an n-dimensional space. When there are two or three dimensions the space can be presented directly as scatterplots [3]. But when there are more than three dimensions either multi-dimensional scaling to reduce the number of dimensions or a user selection of presented dimensions is required. A further limitation of scatterplots is the number of points that can be usefully displayed and read by users. A computer screen has a limited number of pixels and the human eye has limited perceptual resolution. These are scale limitations.

[1]msifer@uow.edu.au

IEEE Symposium on Visual Analytics Science and Technology 2006
October 31 - November 2, Baltimore, MD, USA
1-4244-0592-0/06/$20.00 © 2006 IEEE

The problem of large datasets can be addressed by aggregation. Rather than showing each dataset fact as a point in a scatterplot, a group of facts can be shown as a point, a glyph, a bar or just a number that is proportional to the number of facts in the group. Groups can be created by dividing dimensions into regular meaningful intervals, then aggregating facts that sit in the same intervals. Figure 1 shows the result of applying this to a two dimensional scatterplot to convert it into a table.

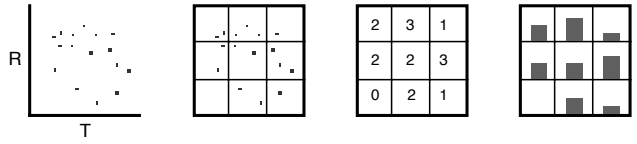

Figure 1. Scatterplot and table views of 2D data.

OLAP data has well-defined dimension hierarchies that divide dimensions into intervals where dimension hierarchy level determines interval granularity. The problem of presenting data with more than two dimensions as a 2D table can be addressed by showing a two dimensional slice and providing navigational controls to choose the row and column dimensions, or by unfolding the table so that more than two dimensions are visible. Figure 2(i) shows an example of the former, while 2(ii) shows an example of the latter. The standard OLAP interface the cross-tab supports both of these.

Figures 2(iii) and (iv) show two more options. Each table cell can itself contain a visualisation such as a chart or proportional coloured bar that conveys addition dimensions. Stolte et. al. use this in Polaris [19] and later in its commercial offshoot Tablue.

2.1.2 Reading the Displays

The purpose of a data visualisation is to facilitate the reading of certain relationships or features. Major tasks are:

- Looking up dimension values of a fact or group of facts.
- Comparing dimension values of facts or groups of facts.
- Identifying local clusters.
- Identifying the distribution of facts along a dimension.
- Comparing distributions or trends.

A fact's position in a scatterplot shows its dimension values. A fact's values are looked up by reading its position from each dimension axis. Clusters are apparent as collections of points in close proximity. Reading distributions is difficult, as it requires visually summing points while comparison of distributions is clearly even more difficult.

In a table the row and column of fact aggregates show dimension values. Looking up dimension values is done by reading off these row and column values. If a table is created by imposing a grid over a scatterplot the ability to see local clusters will be affected by the size of each grid cell and where in the grid a cluster lies. For example a small cluster that is on the border of both a row and a column will sit under four cells, dividing its facts amongst the cells. The small cluster will be difficult to notice when each of the divided portions is aggregated with other facts in their cells. The distribution of facts in a table can be read by glancing across a row or down a column; particularly if histogram bars are used to show table cell values as in figure 1(iii).

In summary the scatterplot is better for reading local clusters, but for reading distributions and comparing distributions tabular approaches are better. However, once scalability is considered, only pivot tables and Polaris provide support for many dimensions with navigation controls and support for large datasets with the aggregation of facts in table cells.

2.2 Parallel Coordinate Approach

A key limitation of the table-based displays is the number of dimensions that can be viewed at once. Even after using combinations of the techniques shown in figure 2, the limit is around three or four dimensions. An approach that does scale well with the number of dimensions is parallel co-ordinates [8].

A fact in a scatterplot is rendered as a point in orthogonal dimension axis. A fact in parallel co-ordinates is rendered as points on each parallel dimension axis that are joined to form a path. Figure 3 shows an example of a small cluster shown with scatterplot and parallel co-ordinates displays.

Parallel co-ordinates offer limited support for looking-up and reading distributions. When a point associated with a fact on dimension axis is chosen, only one dimension can be immediately read, the value of that point on the dimension axis. To read a fact's other dimension values, the fact's path through the other dimension axis must be followed. The distribution of facts along a dimension axis can be read by visually aggregating the paths crossing a dimension axis. Like reading distributions from a scatterplot, this is likely to be a very rough measure. Comparison of distributions is not possible as only one distribution can be seen along each dimension axis. Like the scatterplot this visualisation does not scale well for large datasets.

In summary parallel coordinate displays support the reading of local clusters for small datasets even when there are more than three or four dimensions, but they do not support the reading of distributions very well.

2.3 Display Metrics

Interactive capabilities and static visualisation displays determine the query power of an interface. Interactive features include zooming and filtering. Static visualisations include the scatterplots, tables and parallel coordinate displays described earlier. Next we present metrics for static visualisations of N-dimensional data (facts).

A dimension contour is a curve where each point along the curve has the same dimension value. For example a horizontal line in a scatterplot or a table row are dimension contours. The lookup depth of a displayed fact or aggregate is the number of independent dimension contours that pass through it defining its

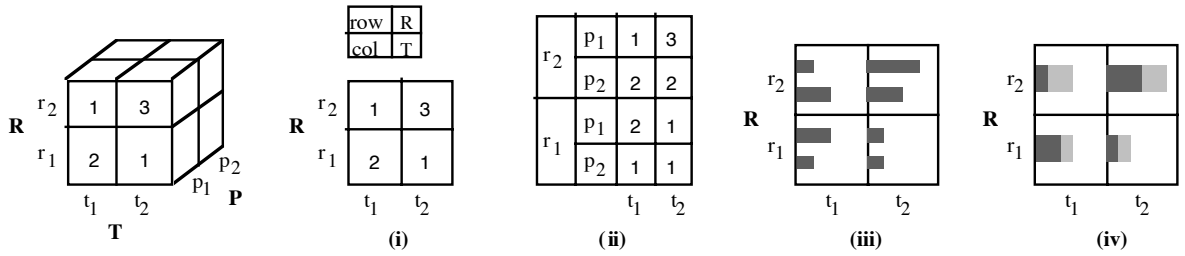

Figure 2. Pivot table, nested table and compound table views of three-dimensional data.

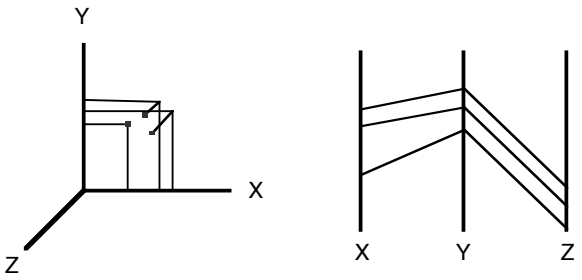

Figure 3. Scatter plot and parallel coordinate views.

dimension values. For example horizontal and vertical lines pass through each point in a scatter plot defining two dimension values, equating to a lookup depth of two.

Dimension contours may support comparison. A dimension contour supports comparison when only one dimension value changes across the contour. When a display shows aggregations of facts lookup depth can define the displays query power. While N dimensional data consists of individual facts having N dimension values, an aggregation of N dimensional data consists of aggregates which summarise facts whose dimension values are within an intersection of intervals from each dimension. That is, each aggregate results from a conjunction of dimension interval constraints. For example, the top left cell in figure 2(ii) aggregates the facts that satisfy r2 ^ p1 ^ t1. The number of dimensions that participate in the conjunction is given by the display's lookup query depth. The display shows the count or weighted (by a measure) sum of each aggregate in some manner.

The comparison depth of a fact or aggregate is the number of independent dimension contours that pass through it that support comparison. These contours allow the distribution of facts in the changing dimension to be read. For example both horizontal and vertical lines through a scatterplot or table support comparison giving a comparison depth of two. However columns in the cross-table shown in Figure 2(ii) do not support comparison as adjacent cells can vary in two dimensions. When row two and three in column one are compared both P and R dimensions change. Only rows in this cross-table support comparison, so cells have a comparison depth of only one. In addition to position other visual characteristics such as colour can increase comparison depth.

When lookup and comparison depth are uniform across the display these values define the display's lookup and comparison depth.

Table 1. Comparison of tabular displays and parallel co-ordinates for data with N dimensions

	Pivot table	Nested table	Polaris table	Parallel coords
Lookup depth	2	3	3	1
Comparison depth	2	1	3	1
Comparison width	2	1	3	N
Setup combinations	${}^{n}C_2$	${}^{n}C_3$	${}^{n}C_3$	1
Small dimensions	√	√	√	—
Aggregation support	√	√	√	—
Query refinement	√	√	√	—

The number of dimensions that can be looked-up is a display's lookup-up width. The number of dimensions along which comparisons can be made is a display's comparison width which usually matches the number of dimension axis.

When a display's lookup width is less than the number of data dimensions the user must choose a subset to display. Setup combinations is the number of choices, where ${}^{n}C_m$ is the number of combinations of m elements taken from a set of n elements.

Table one shows the metrics for table based displays and parallel coordinate displays. The display with the best metrics is the Polaris table shown in figure 2(iv). It has a comparison depth and width of three, better that the unfolded table that was shown in figure 2(ii) that has a comparison depth of one. A comparison depth of two is required to compare two distributions while a comparison depth of three allows a distribution to be compared along two other dimension axis allowing more distributions to be compared at a glance.

Parallel co-ordinates still have a number of strengths compared with the table based displays. They support data with many dimensions better as their lookup and comparison width increase with the number of data dimensions and the number of setup combinations remains one. However, with a comparison depth of one parallel co-ordinates cannot support comparison of distributions. They also have difficulty with small dimensions. For example, if a dimension has two values all paths must travel through two points on the dimension's axis, making it difficult to read.

We introduce an interface that like parallel coordinates presents dimension axis independently but unlike parallel coordinates also supports greater comparison depth, aggregation and query refinement. In combination, these are the capabilities needed to support exploration of OLAP data. These are the requirements that motivate our design.

3 SGVIEWER

Structured Graph Viewer (SGViewer) is an implementation of our interface design. Figure 4 shows SGViewer presenting sales by a distributor of electrical appliances. There were 365 orders for a total of 5254 items. Each order is for some quantity of one item type. The screenshot gives an overview of all orders, presented in five vertically stacked dimension trees: line, brand, outlet, market and price.

Each dimension tree has three or four levels. The top level is a single node representing all orders while the lowest level contains 365 nodes representing each order. The width of each order node is proportional to its item quantity. The lowest level of each dimension tree contains the same 365 order nodes with the same widths, but positioned according to their dimension value. The width of each intermediate level is proportional to the item quantity of the orders below it.

The line dimension tree contains three appliance categories: dishwashers, microwaves and stoves. The dishwasher category is slightly wider than the other two. A glance at the numeric suffixes shows about 2,000 dishwasher sales versus about 1,600 of microwaves and stoves. The brand dimension shows an even spread. The outlet dimension shows west coast cites San Diego and San Jose ordered more than the east coast cites New York and Boston. By market, most sales were to department stores, that ordered more than half of all items.

The price dimension shows the price of each item in an order. It is divided into intervals of 100 dollars then into intervals of 20 dollars. It shows the mid priced 300-400, 400-500 and 500-600

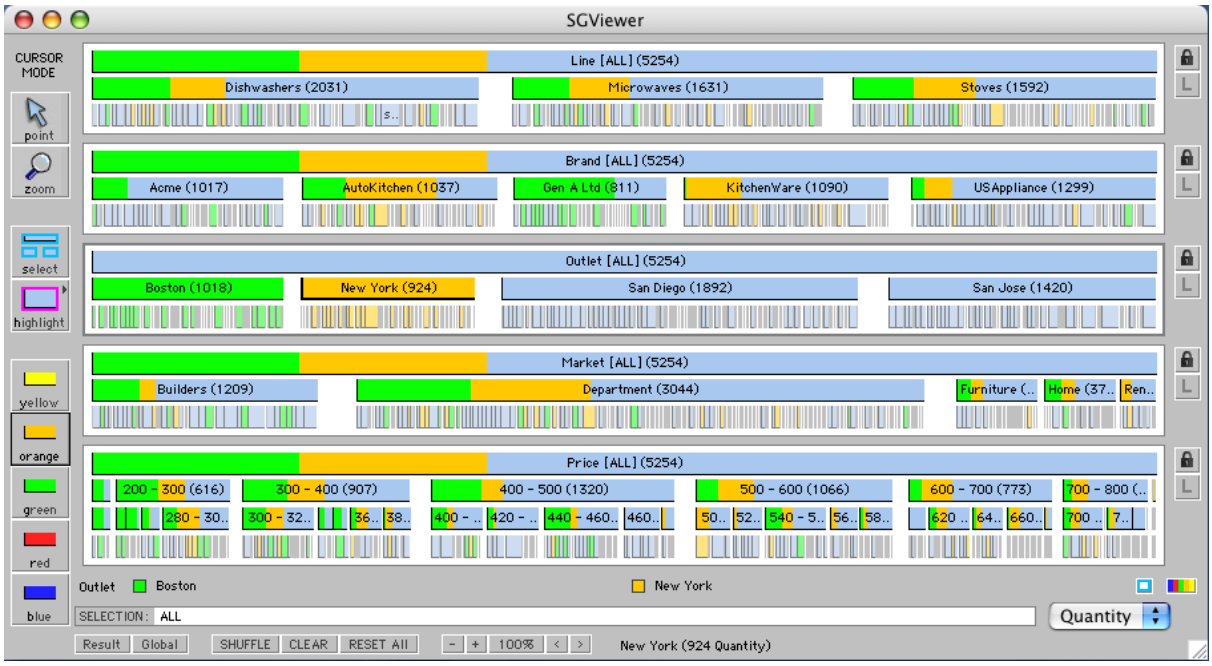

Figure 4. The SGViewer tool presenting sales data.

items were the main sellers, and within these intervals 300-320, 440-460 and 540-560 dollar items were significant.

The outlet dimension has been coloured, New York orders set to green and Boston orders set to orange and remaining orders left as default blue. The colouring of orders has been applied to all other dimensions as well, where each category's colouring shows the relative sum of order quantities. For example in the brand dimension, the category Gen A Ltd is mostly green, has no orange and some blue indicating the destination for its items was mostly Boston. While the KitchenWare brand has almost no observable green indicating almost no orders for Boston. A close look at the price dimension, shows that appliances for Boston were mostly at price points below 500 dollars, while appliances for New York were spread more evenly across the price range.

The relationship between the outlet dimension and other dimensions is shown by the pattern of colour partitioning. A pattern of implicit colour paths between outlet cities and each category in the other dimensions. Readability is maintained by applying the left to right colour sequence (green, orange then blue) used in the outlet dimension in all other dimension categories. However, unlike parallel coordinate displays which shows the relationship between adjacent axis, in figure 4 the relationship between adjacent dimensions such as line and brand can not be read easily. To do this, either line or brand would need to replace outlet as the active coloured dimension.

We used our MakeSGF tool to prepare the order data. It converted a CSV text file of order data into the Structured Graph Format (SGF) XML document that SGViewer inputs. MakeSGF is a Java application while SGViewer is a Java application/applet. The example order data also contained a profit column that can be selected in the viewer via the measure menu.

4 PARALLEL TREES

This section presents the three key design elements of our interface, and details how each contributes to the data exploration tasks that were identified earlier.

4.1 Dimension Value Scales

We describe fixed scales, proportional scales and proportional tree scales. A choice when presenting data is whether to use a fixed value scale or data dependant scale. A histogram uses a fixed value scale to present data frequency for one dimension. A pie chart uses a data dependant scale; the proportion of a pie for a given attribute value that shows relative frequency. Figure 5 (a) shows an example of the former. Figure 5(b) shows an example of a rectangular pie chart, a bargram [20] where relative width shows relative frequency.

The purpose of a dimension scale is to facilitate data lookup; that is to lookup the frequency at a given attribute value. For a fixed scale the lookup process is straightforward, one visually scans along a predetermined constant dimension axis for the desired attribute value then reads the number or bar height at that position. For a proportional scale the lookup process may not be straightforward. To lookup the frequency for a given attribute value one must read the bar titles which may be obscured if the bar is small or is missing if there was no data with that value.

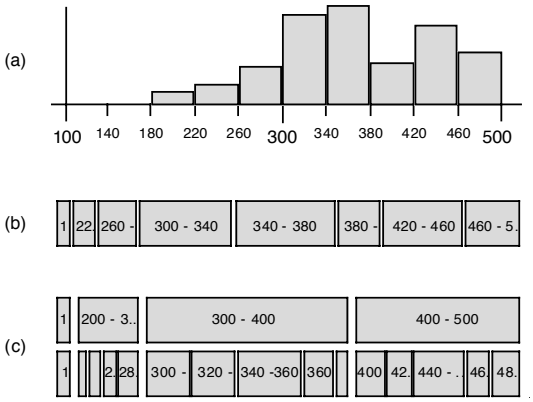

Figure 5. Three types of dimension value scales: (a) fixed (b) proportional and (c) proportional tree.

Figure 5(c) presents a proportional tree scale that addresses some of these issues. This scale can be read in a top-down and left to right fashion. The top level is divided into intervals of 100 units which are divided again into five intervals of 20 units. The 420-440 interval is located by finding the 400-500 interval then looking below it.

4.2 Dimension Relations

Parallel coordinates show the relationship between two or more dimensions by the pattern of paths through them and are often used to locate clusters. However they don't support the task of reading or comparing data distributions. For example in figure 6(a) to see the distribution of data along the Y-axis one needs to count paths entering each 5 unit interval and to see how the Y interval 5-10 is divided by X. This requires visually following each path. Figure 6(b) show a parallel set display [2] of the same data. It uses proportional scales for both X and Y dimensions. Paths between intervals show the size of the association. We observe that the largest Y interval is 5-10, which is evenly distributed between two X intervals 0-1 and 1-2. A problem with the parallel set visualisation is that as the number of intervals increases the increasing number of colour paths of varying thickness become difficult to read.

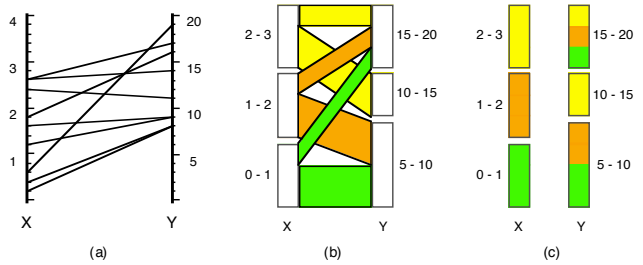

Figure 6. Parallel axis visualizations: (a) parallel coordinates (b) parallel sets and (c) our visualization.

Our solution is shown in (c) where the explicit colour paths are dropped and the axis itself is coloured. Paths between the axes are implicit, linking axis portions that have the same colour. This introduces a significant difference when there are three or more dimensions. Unlike parallel coordinates and parallel sets our visualisation only shows the relationship between the actively coloured dimension like X and all other dimensions, rather than between adjacent dimensions. Figure 4 showed examples of (c): the line, brand, market and outlet category dimensions. The price dimension uses a proportional tree scale in combination with colouring. Definition: these dimensions form a *parallel tree* over a common set.

Table 2. Sales data by product, size and time dimensions.

	Product	Size	Time	Sales
1	hat	med	Q1	100
2	hat	med	Q2	50
3	hat	large	Q1	150
4	hat	large	Q2	250
5	cap	med	Q1	120
6	cap	med	Q2	160
7	cap	large	Q1	110
8	cap	large	Q2	200

4.3 Filter Coordination

Each dimension axis is not just a static visualisation but can also be used as an interactive filter to restrict what is presented in other dimensions. Figure 7 presents an example of a progressive filter coordination applied to the sales data of 1140 items shown in table 2. Part (a) shows an overview of the data; a proportional scale for each dimension. To investigate time Q1 further, the Q1 category is selected restricting the product and size dimensions to Q1 sales as shown in (b). Note the sales count in product and size dimensions sums to 480 items, the Q1 total.

To drill-down further the Cap category is selected, restricting the remaining size dimension to Q1 and Cap sales as shown in (c). In general, selection in a dimension restricts the remaining dimensions; those dimensions that were selected later or remain un-selected. It is only the unset dimensions, such as product and size in (b) that show the proportions of a common result set and form a parallel tree from which relationships can be read. The result set is the conjunction of dimension selections, of one or more categories.

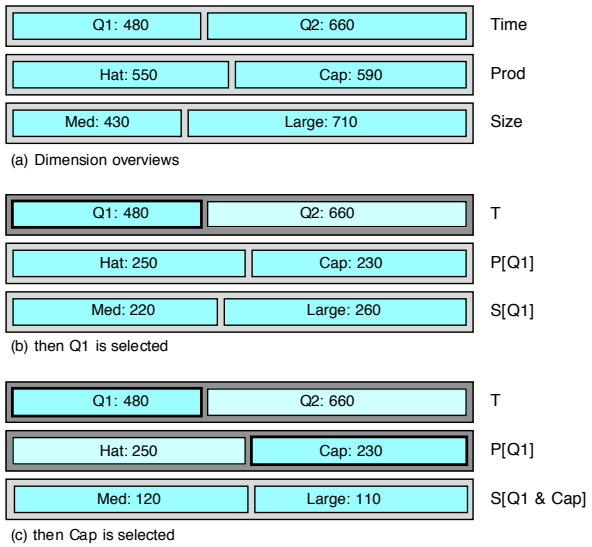

Figure 7. Progressive filter co-ordination.

A number of visual cues are used to show which dimensions have been restricted: background colour changes, box border changes and greying of categories that are not selected. Users can restrict dimensions in any order. However, if there are many dimensions and the user has restricted several they may lose track of the query sequence. For example, if they restrict the fourth, then the first and then third vertical dimension.

Our solution is a shuffle operation that reorders restricted dimensions vertically into their progressive query order. They are placed above any unset dimensions. We leave unset dimensions in their original order to minimise visual change when shuffling. Our design could be extended, to include an option that reorders unset dimensions based on correlations. This would be most useful when there are many dimensions.

5 User Scenarios

We demonstrate our interface with two scenarios: (i) the sales data shown in figure 4 is explored further with pattern division (an alternative to colour division) and filtering, and (ii) a network data example that contains deeper dimension trees.

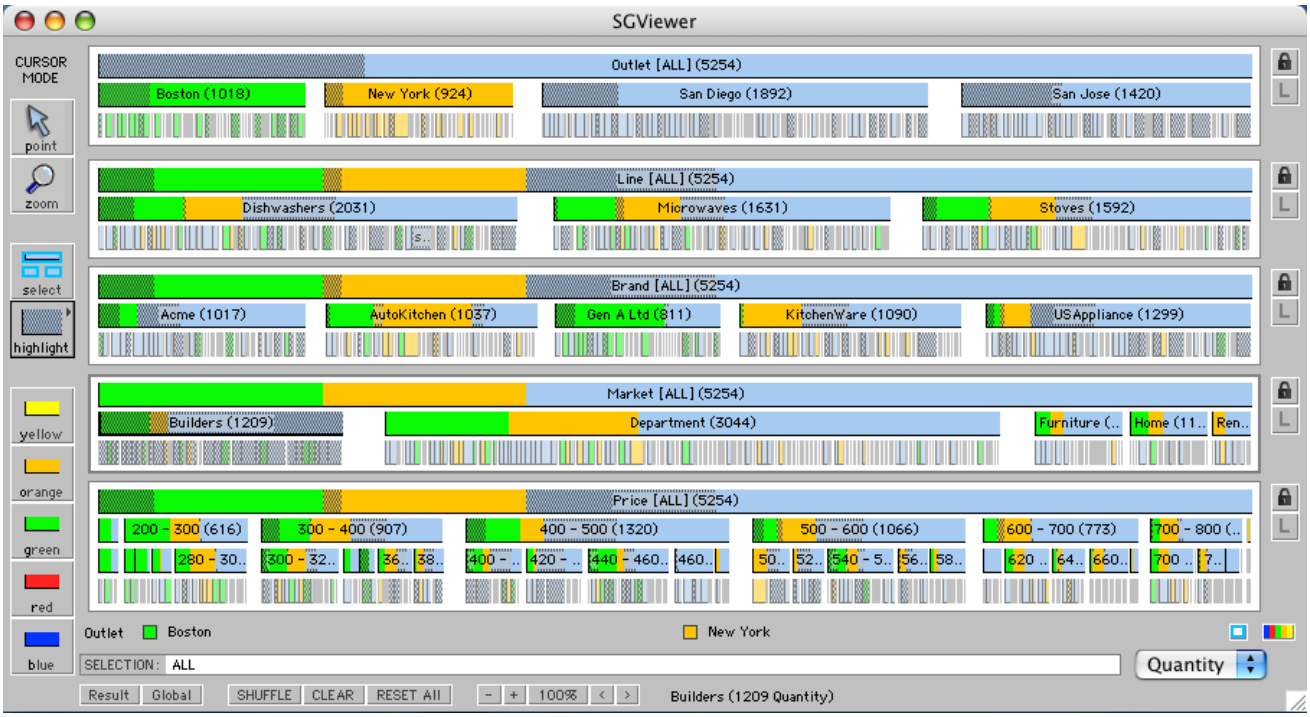

Figure 8. Sales data that is coloured by Outlet and pattern masked by Market.

5.1 Sales Data

Earlier we noted that our parallel tree visualisation shows the relationship between an actively coloured dimension and all other dimensions. In figure 4 this was between the outlet dimension and other dimensions. An interesting category in the market dimension was Builder; an industry market rather than a consumer market. We would like to see the relationship between Builder sales and line, that is which appliances builders have been buying. We could change the active coloured dimension to Market to see how the Builder colour is distributed across appliances. A less disruptive approach would be to drill-down into the data while maintaining as much context as possible.

One approach is to visually divide dimension intervals in a way that is similar to colour division but independent of it. Pattern mask is such an approach. Figure 8 shows the effect of applying a pattern mask to the Builders category. Each colour segment in a category or interval is divided in two: the portion which are builder sales and the rest. Only the green portion of Dishwashers is partly masked indicating that there were significant sales of Dishwashers to builders in Boston but none to builders in New York, while most Microwave and Stove sales were to east coast Builders.

Another approach is to restrict dimensions to sales by Builders, that is to use the filter technique described in section 4.3. Figure 9 shows the effect of selecting Builders. Each root dimension interval except Market has the suffix [Builders] to denote this restriction. The Outlet, Line, Brand and Price dimensions show 1209 Builder sales. We can see more detail. For example a thin orange band in Dishwashers indicates that a few but not zero sales to Builders in New York. We can also see the overall proportion of sales to builders was for Dishwashers.

Sales data can be restricted again to see a more detailed view of a contained subset. We are interested in the sale of mid priced appliances to Builders. The intervals 500-600 and 600-700 are selected. Figure 10 shows the result. The remaining three

Figure 9. Sales data after Builders is selected.

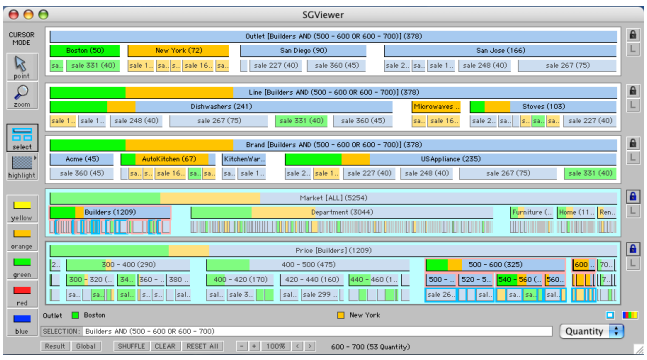

Figure 10. Sales data after market Builders then price range 500 – 700 dollars is selected.

dimensions: Outlet, Line and Brand are restricted to Builders and ($500 - $700), 378 sales. The line dimension shows most of these sales were for Dishwashers; this included one large order, sale-267 for 75 US Appliance brand dishwashers for a San Jose Outlet.

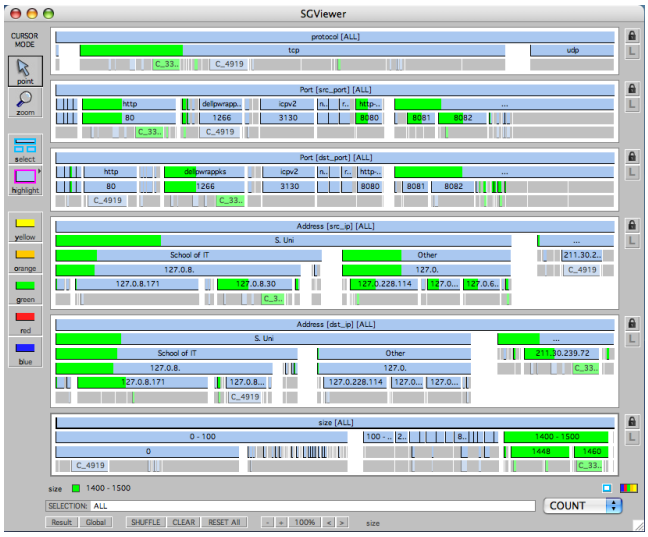

Figure 11. An overview of network traffic data in six dimensions where messages of size 1400-1500 are green.

5.2 Network Data

A network traffic example that makes greater use of our proportional tree scale was also explored. Each dimension except the first has several intermediate tree levels. We took a network dump of 200,000 messages with six dimensions: protocol, source port, destination port, source address, destination address and size. They were factored into 38,000 nodes where each node has the same dimension values, by our MakeSGF data preparation tool and then fed into SGViewer.

Figure 11 shows the result after messages with size 1400 - 1500 were selected green. Most messages used the tcp protocol. Most traffic was not http, which only accounted for about 10%. The size dimension shows that a large minority of messages where just headers with a content size of zero, while within the size range we selected, most messages were 1448 or 1460 bytes. Both size and address dimension trees show that even when the tree scale is unbalanced, large categories or intervals that are several levels deep can be read at a glance.

6 Evaluation Summary

6.1 Student Study

Fifty five first year software development students completed a user study of the viewer that took about one hour per person. It was conducted in several sessions over three days using an online survey that participants filled in as they did the study. A web log dataset of 3,000 visits was used. Participants were given a series of tasks to complete: seeing detail via zooming, reading time trends, reading data distribution, making selection to see a subset and using colour to make comparisons.

At the end of the study they were asked to rate each task using a scale of 1 (difficult) - 10 (easy). Students also had opportunities to make qualitative comments. Table 3 shows the average student ratings. In our own use of the viewer, we had explored datasets by first zooming and filtering, and only later applied colours to see proportions within selected subsets. We expected filtering and colouring tasks to get a similar rating.

In other qualitative comments/answers about 40% of students indicated the feature they most liked was the use of colours.

Table 3. Average rating of ease of use: 1 (hard) – 10 (easy)

Task	*Rating*
seeing detail via zooming	8.1
reading time trends	5.5
reading data distribution	5.9
making selections to see a subset	6.7
using colour to make comparisons	8.0

6.2 Company Evaluation

We provided SGViewer, the MakeSGF data preparation tool and short user guide to a large telecommunications company for external evaluation. A company staff member spent an extended period of time with the viewer. He found the tool simple to use, but judged that useful analysis of data, as presented by the viewer, required an ability to deal well with abstraction, and a familiarity with the data. The viewer presents data via labelled trees. Such trees will not be meaningful unless the user is familiar with the category labels. While its likely a data analyst or manager would likely satisfy these requirements a consumer may not.

The viewer design can be divided into: (i) coordination of the dimension panels including the use of colour, and (ii) the tree visualisation used in each panel. The latter could be made less abstract for consumers by offering additional panel visualisations that are more concrete such as geographic maps.

6.3 Individual Experience

We worked with a system administrator who used the viewer to explore network traffic logs. Our main motivation was to see how well the viewer supported larger datasets. For small datasets of up to 20,000 nodes his experience was consistent with the other studies. After an initial learning period he was able to use the viewer to read and query the network data. Where possible, he also preferred to use colouring to make comparisons rather than drilling-down by making filter selections. However, the zoom functions became harder to use as dataset sizes approached 200,000 aggregated nodes and hierarchies approached thousands of categories.

6.4 Discussion

The viewer used in the student study did not offer the general zoom controls, they could only zoom between subtrees in a dimension. Students rated this subtree navigation well with a value of 8.1. However both reading tasks were rated well below this, 5.5 and 5.9. Student comments indicate they sometimes found it difficult to make visual comparisons across a dimension tree level. For example date dimension bar heights were quite small. The viewer provided to the company and administrator offered the general zoom controls which they preferred to use. The administrator's main request was integration of a standard scroll bar to make comparisons and navigation across a zoomed dimension easier. We expect the use of large screens, a background grid and live scroll bars would address these issues.

The response for colouring tasks was very positive. It was rated 8.0 by the students, the equal easiest task, and reinforced by their qualitative comments. The administrator also preferred colouring to filtering. This indicates data exploration via proportionate colour division of dimensions was not only usable but the preferred approach. However it is an approach that can place substantial demands on zooming to support the reading of detail.

7 Related Work

Additional related work falls into several areas: extensions of parallel co-ordinates, tree visualisations, interactive query systems for exploring multi-dimensional data and our work.

Parallel co-ordinates have been extended in a number of ways. Fua et. al. [4] improved the scalability of parallel co-ordinates by deriving a hierarchical data clustering and then showing a selected level of cluster nodes as coloured paths. A graduated band that shows the extent of the cluster was added to these paths. Hauser [6] added aggregation support by placing histograms over each parallel axis to show data distribution. Our tree layout is a one dimensional space filling variation of TreeMaps [14] that shows subtree depth and is zoomable.

Many systems for interactive querying of multi-dimensional data have been developed. Dynamic query systems [1] uses selections in multiple widgets, one for each data dimension to locate subsets of the data. Table Lens [13] supports exploration of very large attribute tables with focus areas and zoom controls. FocusTable [18] and its' InfoZoom offshoot support incremental queries on large attribute value tables by successive selection of values in the rows of interest. Columns with values excluded by these selections are filtered out. Both FocusTable and Table Lens columns can be sorted by a selected attribute row, transforming that row into a value scale that shows (count) proportions. But this approach allows such tables to present only one continuous attribute value scale at a time. FocusTable also allows rows describing different granularities of an attribute to be grouped together with a tree outliner. When this is combined with resorting, proportions across a single attribute hierarchy can be shown. In contrast, SGViewer can show proportions across multiple attributes in multiple hierarchies.

Attribute explorer [17] and Query preview [12] systems display data distribution to guide progressive querying. At each query step Query previews show the distribution of intermediate results in multiple dimension at a single level of aggregation. Users avoid developing queries that will have an empty result. Wittenburg et. al. [20] described an extension of dynamic queries where each widget is a bargram that shows the distribution of relevant values.

Graham et. al. [5] have developed a system that uses parallel tree views linked via colour brushing. However colouring is applied only to leaves and the trees are not strictly proportional limiting their interface to smaller datasets. This paper extends previous SGViewer based work [15,16]. It is distinguished by the addition of colour and pattern partitioning to improve query power, and our comparison of user interfaces for exploring multi-dimensional data. The earlier work supported data distribution overviews and progressive filtering via panel selections only.

8 Conclusion

This paper introduced a number of metrics to compare interfaces for interactive data exploration of hierarchical multi-dimensional data of OLAP systems. In particular we contrasted interfaces that support detailed but narrower data views (table based) with interfaces that support wider data views (parallel coordinate based). We argued that wide interfaces are better suited to exploration of new or changing datasets where the dimensions of interest are not known, so having an immediate view of all dimensions is important.

However, existing parallel coordinate interfaces do not support OLAP data exploration as they do not support aggregation. We introduced an interface implemented as SGViewer based on parallel trees that does. We demonstrated its support for comparison of distributions via colouring and its support for drilling-down via progressive filtering. Our evaluation showed where there was a choice users preferred colour partitioning to filtering. We expect that SGViewer and table based OLAP interfaces are complementary. An analyst is likely to be best equipped by having wide and deep data exploration tools. The appropriate choice will depend on the nature of the exploration and the number of data dimensions.

Referernce

[1] Ahlberg C. and Shneiderman B., Visual information seeking: tight coupling of dynamic query filters with starfield displays. In Proc. CHI'94, ACM Press, 1995.

[2] Bendix F., Kosara R., and Hauser Helwig. Parallel sets: a visual analysis of categorical data. IEEE InfoVis, MN, USA, Oct. 23-25, 2005, 133-140.

[3] Cleveland W. and McGill M. Dynamic Graphics for Statistics. Wadsworth, Inc., 1988.

[4] Fua Y., Ward M. and Rundensteiner E. Structure based brushes: a mechanism for navigating hierarchically organised data and information spaces. IEEE trans. on visualization and computer graphics, Vol. 6. No. 1, April 2000, 150-59.

[5] Graham M., Watson M., and Kennedy J. Novel visualization techniques for working with multiple, overlapping classification hierarchies. Taxon 51. May 2002, 351-358.

[6] Hauser H., Ledermann F. and Doleisch H. Angular Brushing of Extended Parallel Co-ordinates. Proc. IEEE Symposium on Information Visualization, 2002, 127-130.

[7] Han J. OLAP Mining: An integration of OLAP with data mining. FIP Conference on Data Semantics (DS-7), Leysin, Switzerland, Oct. 1997, pp. 1-11.

[8] Inselberg A. The plane with parallel co-ordinates. The Visual Computer, 1(2), 1985, pp. 69-92.

[9] Keim D. Visual exploration of large data sets. Communications of the ACM, Aug. 2001, Vol. 44, No. 8, 38-44.

[10] Microsoft Excel -- User's Guide. Redmond, Washington. Microsoft, 1995.

[11] Pedersen T. and Jensen C. Multidimensional Database Technology. IEEE Computer 34(12), 2001.

[12] Plaisant C., Bruns T., Doan K. and Shneiderman B. Interface and data architecture for query previews in networked information systems. ACM Trans. on Information Systems, 17, 3, 1999, 320-341.

[13] Rao, R., and Card, S.K. The Table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In Proceeding of CHI'94, Boston MA, USA, 1994, 318-322.

[14] Shneiderman B. Tree visualisation with treemaps: a 2-d space filling approach. ACM Transactions on Graphics, 11(1), Jan. 1992, 92-99.

[15] Sifer M. A visual interface technique for exploring OLAP data with coordinated dimension hierarchies. Proc. ACM CIKM, New Orleans, Nov. 2003, 532-535.

[16] Sifer M. Filter coordinations for exploring multi-dimensional data. Journal of Visual languages and Computing, Elsevier. Vol. 17 Issue 2, April 2006, 107-125.

[17] Spence, R., Tweedie, L.: The Attribute Explorer: information synthesis via exploration. Interacting with Computers, 11(2), 1998, 137-146.

[18] Spenke, M., Beilken, C. and Berlage T., FOCUS: the interactive table for product comparison and selection. In Proceedings of UIST'96, ACM Press, 1996, 41-50.

[19] Stolte C., Tang D. and Hanrahan P. Query, Analysis, and Visualization of Hierarchically Structured Data using Polaris. In Procs. ACM SIGKDD, July 2002.

[20] Wittenburg K., Lanning T., Heinrichs M. and Stanton M. Parallel bargrams for consumer based information exploration and choice. Proc. ACM UIST'01, 51-60.