Direct Manipulation of Parallel Coordinates

Harri Siirtola

Human-Computer Interaction Group TAUCHI
Department of Computer and Information Sciences
FIN-33014 University of Tampere, Finland
+358-40-5488700
hs@cs.uta.fi

Abstract

This paper introduces two novel techniques to manipulate parallel coordinates. Both techniques are dynamic in nature as they encourage one to experiment and discover new information through interacting with a data set. The first technique, polyline averaging, makes it possible to dynamically summarise a set of polylines and can hence replace computationally much more demanding methods, such as hierarchical clustering. The other new technique interactively visualises correlation coefficients between polyline subsets, helping the user to discover new information in the data set. Both techniques are implemented in a Java-based parallel coordinate browser. In conclusion, examples of the use of these techniques in visual data mining are also explored.

1. Introduction

Parallel coordinates [4] is a two-dimensional technique to visualise multidimensional data sets. An *n*-dimensional data tuple

$$(x_1,x_2,x_3,\ldots,x_n)$$

is visualised in parallel coordinates as a *polyline,* connecting the points $x_1,x_2,x_3,..,x_n$ in *n* parallel *y*-axes, as can be seen in Figure 1. For a large set of tuples, this technique will produce a compact two-dimensional visualisation of the whole multidimensional data set.

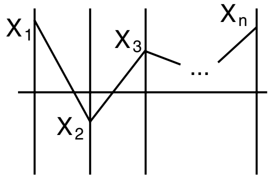


Figure 1: A tuple visualised with parallel coordinates.

In addition to being a space-efficient method to represent a large data set, parallel coordinate visualisation is also interactive. The user can select ranges from the *y*-axes and the lines traversing through the ranges will be emphasised, for instance, in colour. These ranges can be combined with other ranges using the standard logical connectors `AND`, `OR`, and `XOR`. This kind of set of connected ranges is corresponding to a query performed on a database management system. The user can also perform several independent, differently coloured queries simultaneously, which makes it simple to compare the results. This is an easy and intuitive method to visually query a data set, or to do visual data mining.

1.1. Interaction with parallel coordinates

Parallel coordinates are generally considered to be one of the standard techniques to visualise multidimensional data sets. What is not well established yet is the interaction with parallel coordinates and especially the dynamic aspects of interacting with them. Many of the existing implementations have a poor interface. Especially, direct manipulation of parallel coordinates is usually not supported. This paper explores the direct manipulation of parallel coordinates and proposes two new techniques.

The goal of rapid interaction or direct manipulation in visualisation is to give the user a chance to see more in a certain amount of time, or to detect interesting dynamic behaviour. The speed of the interaction can be divided into three levels the finest of which is roughly 0.1 seconds [1, p. 231]. This is the *cause and effect* limit, meaning that manipulating a visualisation control and getting a response within 0.1 seconds will connect these events. A user perceives the effect as a direct response to the manipulation. This is the form of interaction that is still underused in the manipulation of parallel coordinates.

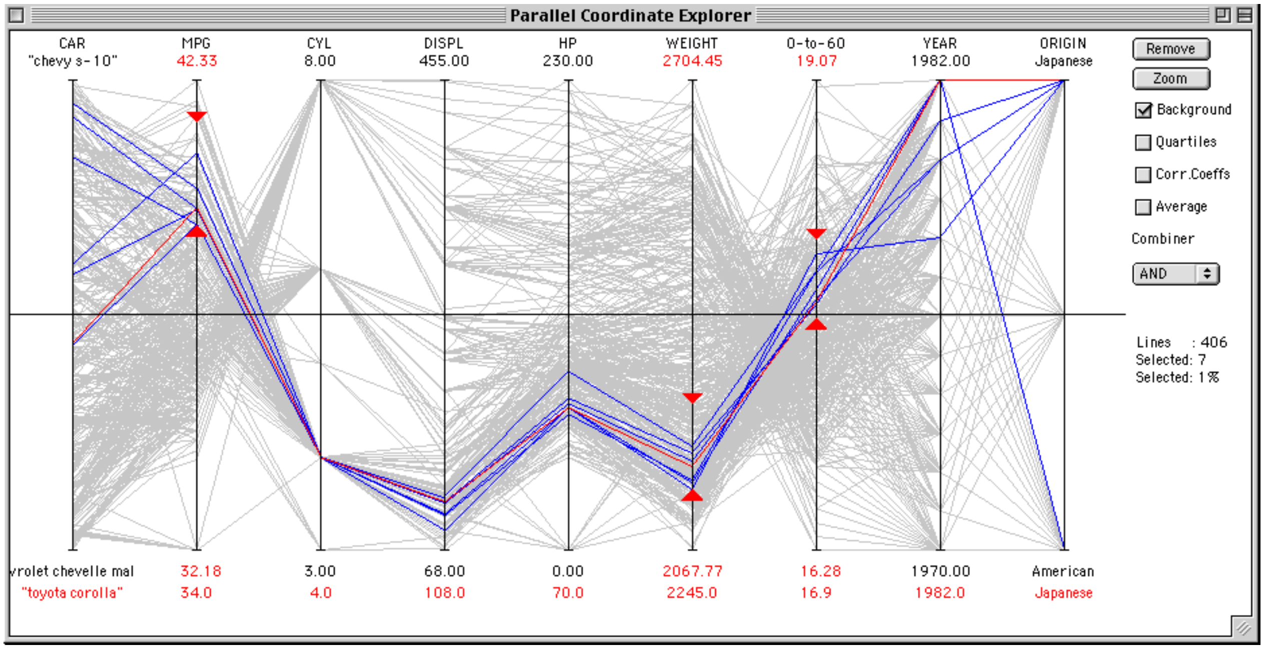
The latter part of this paper discusses what kind of useful dynamic interaction with parallel coordinates can be performed within the cause and effect limit. The amount of work that can be done within this limit is naturally dependent on the speed of the computer and the size of the data set.

The ideas presented here were implemented with Java on a personal computer (a 250 MHz G3 Macintosh) running a Java 1.1.8 virtual machine. A faster machine or a native-compiled implementation would naturally allow for the direct manipulation of much larger data sets.

1.2. Cars data set

The data set used in the experiment is a version of the widely available cars data set that has 9 dimensions and 406 polylines, or 3654 data items [2]. The cars data set contains data on all the cars road tested by the *Consumer Reports* magazine between 1971 and 1983. There are 9 variables in the cars data set: make and model of the car (`CAR`), fuel economy in miles per U.S. gallon (`MPG`), number of cylinders in the engine (`CYL`), engine displacement in cubic inches (`DISPL`), output of the engine in horsepower (`HP`), vehicle weight in U.S. pounds (`WEIGHT`), model year (`YEAR`), and origin of the car (`ORIGIN`).

American Statistical Association (ASA) has been using this data set in its annual expositions of statistical graphics technology, which is a shootout for new statistical data graphics ideas. What makes the cars data set interesting is the fact that it reveals the global influence of the 1973 oil crisis in car design.



☑ Background
☐ Quartiles
☐ Corr.Coeffs
☐ Average
Combiner
AND

Lines : 406
Selected: 7
Selected: 1%

Figure 2: A graphical query with parallel coordinates: cars that have mileage, weight and 0 to 60 mph acceleration between given limits. A tuple from a set fulfilling these conditions has been selected and is displayed at the bottom of the visualisation.

2. Manipulating a selection

Selecting or modifying a range from one of the parallel axes is the most frequent user action when interacting with a parallel coordinate visualisation. Direct manipulation can enhance this interaction by providing continuous feedback while the user is changing the range limits. Figure 2 shows a simple query that is displayed as a selection of polylines. Besides colouring the set of polylines according to the current ranges, the program can show the number and percentage of polylines currently selected.

When the selection changes dynamically during manipulation, there is a chance to observe behaviour that would not be revealed with slower updates. For example, selecting the full range of vehicle weights and lowering the upper limit gradually reveals that heavy vehicles come from all parts of the other dimensions – not just from the upper or lower parts of the other dimensions, as one might expect.

Another situation where continuous feedback is crucial is when you want to select a certain number of polylines from one of the axes. Suppose you want to divide the set of vehicles into two classes of equal number of members according to vehicle weight. Finding the correct spot on the `WEIGHT` axis without a continuous update of the percentage would cause many trials and errors.

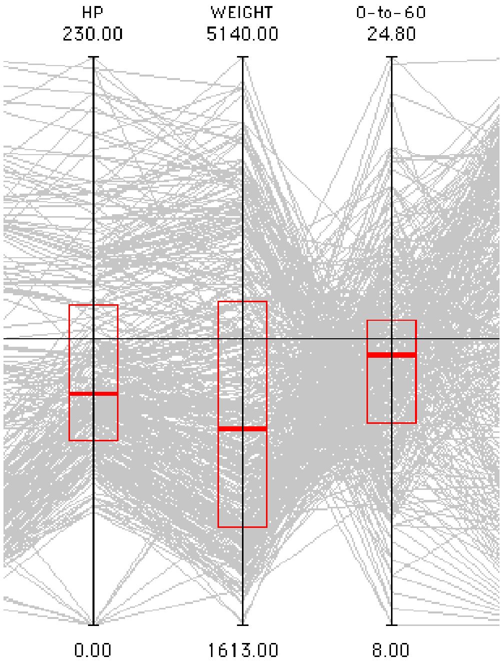


Figure 3: Displaying quartiles as a Box Plot.

Dividing a set of polylines into subsets can also be assisted statically by overlaying each axis with rectangles that show quartiles. In Figure 3 we have the axes of `HP`, `WEIGHT` and `0 to 60 mph` acceleration divided into four 25% subsets. This kind of overlay makes it easier to formulate queries where you either need to explore a certain number of data tuples of one dimension or need to see where the other selected polylines are located. A similar representation is known as Box Plot in statistical data graphics.

3. Summarising a set of polylines rapidly

With a large number of polylines, the parallel coordinates visualisation will be more difficult to read, because the lines that ovelap and are positioned close to each other appear as a solid surface. A number of solutions have been suggested to this problem, such as *dimension zooming* and *hierarchical clustering* [3].

Dimension zooming is simply an operation where we scale up the selected axis according to a selected range. Figure 4 illustrates the application of dimension zooming several times in succession, which reveals a more fine-grained distribution of lines crossing the axis.

Hierarchical clustering is a technique where polylines appearing close to each other are clustered together, forming variable-width bands. These bands are then illuminated according to the size of the cluster. This kind of technique calls for high quality rendering and requires a lot of the implementation. However, the result is aesthetically pleasing and can deal with a large number of polylines.

Technique of hierarchical clustering requires more computational power than is generally available during direct manipulation. A simpler approach is to simply average the lines in the current selection.

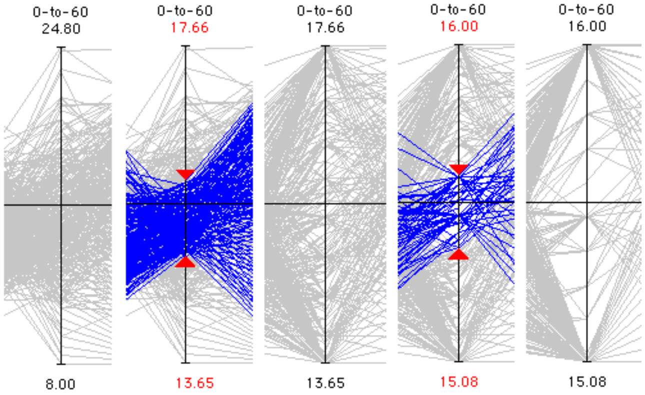
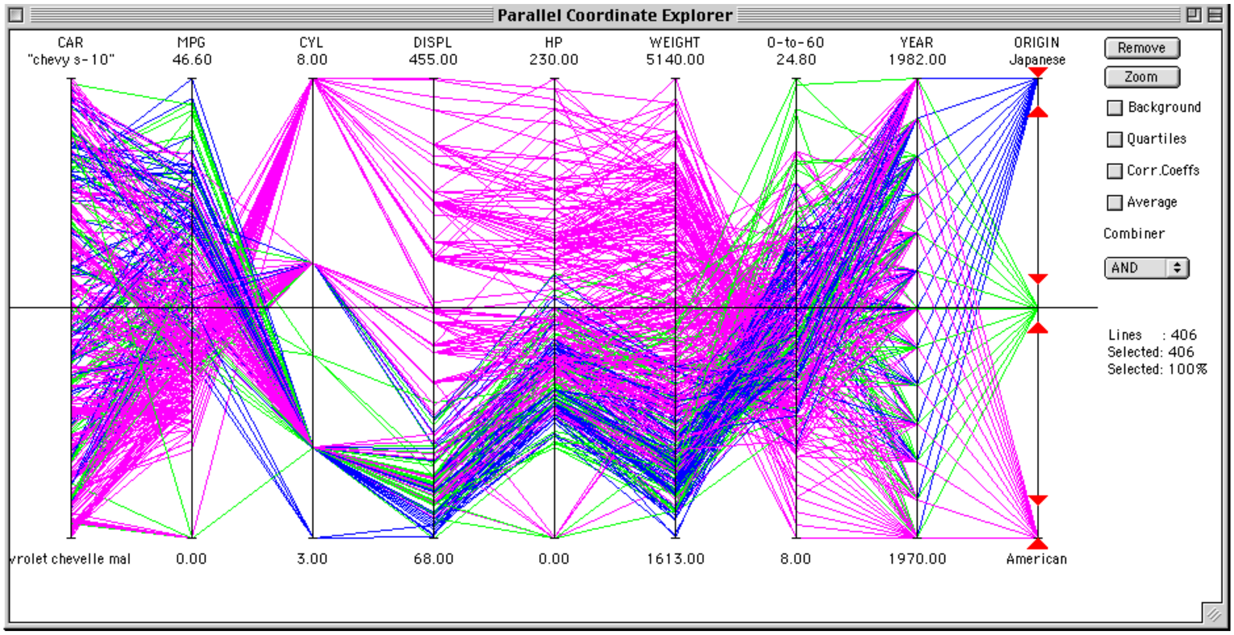


Figure 4: Dimension zooming applied in succession.

Although replacing the set of lines with their average hides much of the information, we get a quick overview of the data set and its subsets through direct manipulation of the ranges. The hidden information problem can be at least partially solved by showing also the standard deviation of a polyline set in a graphical form. In the current implementation the standard deviation is shown as a bar extending one normalised standard deviation unit over the average.

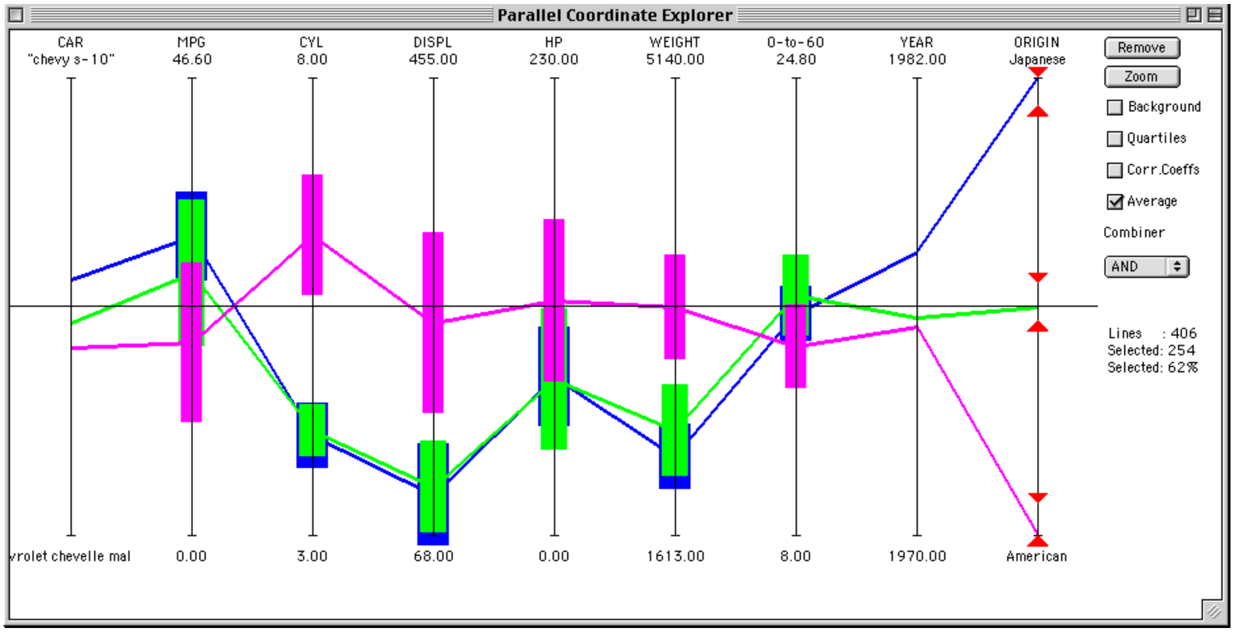


□ Background
□ Quartiles
□ Corr.Coeffs
□ Average
Combiner
AND

Lines : 406
Selected: 406
Selected: 100%

Figure 5: Comparison of American, European and Japanese cars.

Figure 5 and Figure 6 show a comparison of American, European and Japanese cars. In Figure 5 the data is visualised as any parallel coordinate browser would do, and in Figure 6 the polyline sets are replaced with lines traversing through their average values. In Figure 5 it is difficult to see how the subsets relate to each other, but in Figure 6 it is easy to see that European and Japanese cars are almost identical in regard to these properties, and that American cars are heavier, have larger engines, have better 0 to 60 mph performance and horsepower, and have inferior fuel economy.



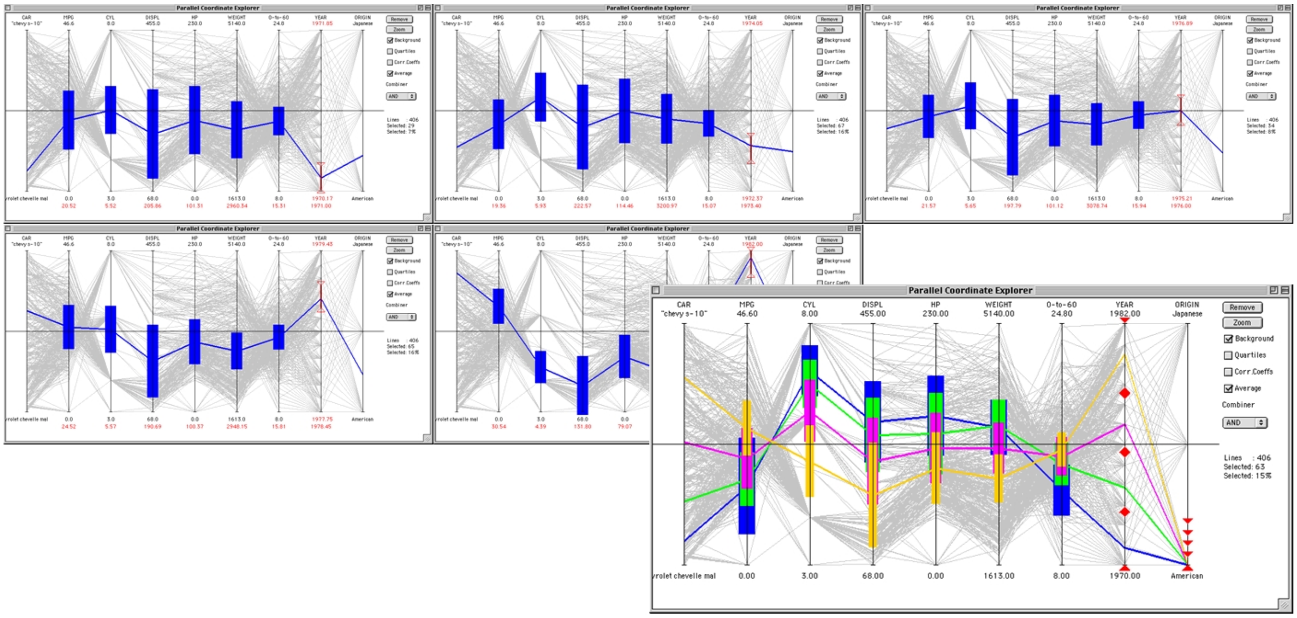
□ Background
□ Quartiles
□ Corr.Coeffs
☑ Average
Combiner
AND

Lines : 406
Selected: 254
Selected: 62%

Figure 6: Comparison of American, European and Japanese cars with polyline averaging.

One problem with polyline averaging is that a nominal scale axis will have meaningless averages. This is not a problem if users are aware of this and do not try to give meaning to these values.

Averaging is useful as a rapid and dynamic technique to explore the behaviour of a subset of polylines. Averaging is especially well suited for data that is not nominal, but discrete in nature. A typical example of this kind of data is an evaluation form where subjects rate something by giving it a number value , for instance from 1 to 5. This kind of data is difficult to explore with a parallel coordinates visualisation because of the many overlapping polylines, but averaging will summarise the selections and produce readable visualisations.



☑ Background
□ Quartiles
□ Corr.Coeffs
☑ Average
Combiner
AND

Lines : 406
Selected: 63
Selected: 15%

Figure 7: Animation showing changes in the performance and efficiency of American cars in 1970–1982.

Figure 7 shows how the direct manipulation of parallel coordinate ranges can be used to animate changes in the data set. In Figure 7 we animate how American cars have changed from 1970 to 1982. The animation is controlled by dragging the year selection upwards and continuously drawing the averaged polylines.

Several observations can be made from the animation. Before the year 1973 the mileage of the cars was slowly declining, but after the oil crisis there is a clear trend for better mileage. Other observations that can be made are the drift from 8 and 6 cylinder engines towards more economical 4 cylinder engines, the decline of engine displacement and horsepower, and the direction towards lighter vehicles.

The same information the animation portrays can also be illustrated by a static visualisation. The last slightly cut out screen in Figure 7 shows four parallel queries of American cars in four different time periods. The same observations are also evident in this visualisation.

4. Observing correlating subsets

It is sometimes difficult to observe the relative order of lines travelling through a pair of y-axes in a parallel coordinates visualisation. If they cross the axes in the same order, there is a positive correlation between the attributes, and if they cross in the opposite order, there is a negative correlation.

Figure 8 shows two situations where the correlation coefficients of two polyline subsets are animated. The correlation coefficient is drawn as a bar between the ranges, the positive coefficient as a green upwards-pointing bar and the negative coefficient as a red downwards-pointing bar. A green bar extending to the top of the polyline area indicates a correlation coefficient of +1, and correspondingly, a red bar extending to the bottom indicates a correlation coefficient of -1.

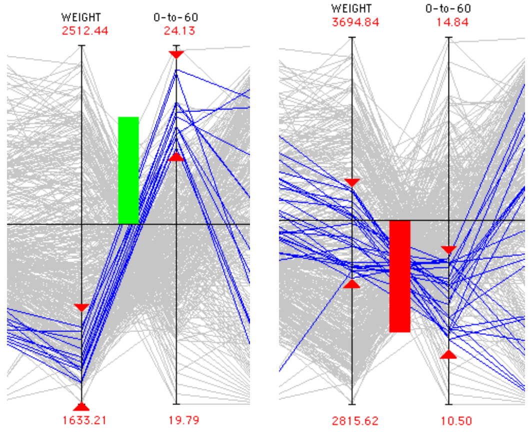
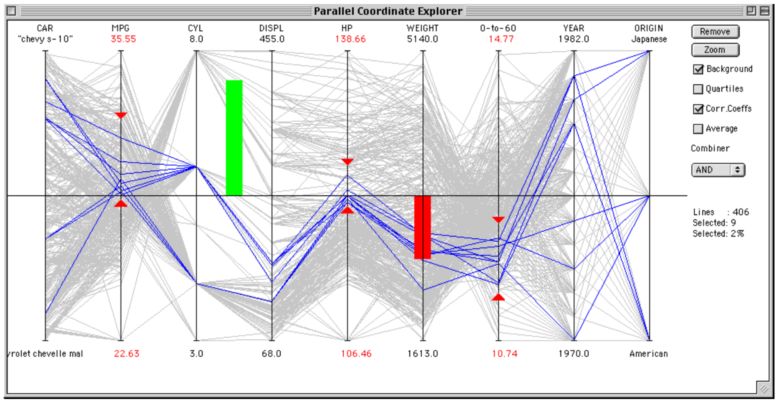


Figure 8: A subset with positive correlation on the left, and a subset with negative correlation on the right.

The example in Figure 8 shows that there are subsets with positive and negative correlation coefficients between variables `0 to 60 mph` and `WEIGHT`. In other words, in the former subset, a vehicle having a smaller weight also has a better `0 to 60 mph` performance, and in the latter subset the situation is the opposite. Discovering these subsets without animated correlation coefficients would be difficult.

Figure 9 displays an interesting situation from the cars data set. Overall, there is a remarkably strong negative correlation coefficient between `MPG` and `HP`, indicating that more powerful cars have lower mileage. However, the subset displayed in Figure 9 has a relatively strong positive correlation. A closer examination reveals that it contains mainly new, technically advanced cars like Nissan Maxima and 280Z, Toyota Cressida, and Saab 99. The oldest car in the subset is the venerable BMW 2002, a car with a low weight but a powerful engine.



☑ Background
□ Quartiles
☑ Corr.Coeffs
□ Average
Combiner
AND

Lines : 406
Selected: 9
Selected: 2%

Figure 9: A subset where a better mileage has positive correlation coefficient with a higher horsepower.

5. Conclusions

Two novel techniques to enhance parallel coordinate visualisations were presented in this paper. They were implemented in a Java-based parallel coordinate browser. The Java applet version of this browser is available at

`http://www.cs.uta.fi/~hs/pce/`

These techniques make it possible to perform new kind of queries with parallel coordinates, and encourage exploring a data set by providing better feedback. Specifically, they provide new ways to directly manipulate parallel coordinates.

The usefulness of these new interaction techniques must be verified with further user testing.

6. Acknowledgements

This work was supported by the Technology Development Centre TEKES (project 20287).

7. References

[1] S.K. Card, J.D. Mackinlay and B. Shneiderman. Readings in Information Visualization – Using Vision to Think. Morgan Kaufmann Publishers, San Francisco, CA, 1999.

[2] Cars data set. http://stat.cmu.edu/datasets/

[3] Y.-H. Fua, M.O. Ward and E.A. Rundensteiner. Hierarchical Parallel Coordinates for Exploration of Large Datasets, in Proc. of IEEE Conf. on Vis. '99, IEEE Comp. Soc., San Francisco, CA, 1999.

[4] A. Inselberg and B. Dimsdale. Parallel Coordinates: A Tool For Visualizing Multidimensional Geometry, in Proc. of IEEE Conf. on Vis. '90, 361-378. IEEE Comp. Soc., Los Alamitos, CA, 1990.