

# Using Curves to Enhance Parallel Coordinate Visualisations

Martin Graham and Jessie Kennedy

School of Computing, Napier University, 10 Colinton Road, Edinburgh, EH10 5DT, UK

E-mail: {m.graham, j.kennedy}@napier.ac.uk

## Abstract

*This paper introduces a number of refinements to the Parallel Coordinates visualisation metaphor for multi-dimensional data. Firstly, the traditional set of poly-lines are replaced with a collection of smooth curves across the attribute axes, allowing individual data elements to be traced under certain limitations; normally impossible due to the 'crossing problem'. Then the notion of spreading out points on axes with a few discrete values is introduced, which leads onto a simple focus+context technique when the user selects values on such axes.*

## 1. Introduction

During the 1980's and early 90's, Inselberg and Dimsdale [1] introduced *Parallel Coordinates*, a representation of multi-dimensional information or data, in which multiple dimensions are allocated one-to-one to an equal number of parallel axes on-screen. An object in a data set is then mapped as a series of points, one per axis, with the position of each point on the axis being dependent on their value in the associated dimension. The points are then joined together by line segments from one axis to its immediate neighbour, forming a *poly-line* across the set of axes. This process is then repeated for each object in the information set.

Parallel Coordinates allows similar objects to be seen as having similar shapes and the basic technique has since been modified with a variety of additional features. For example, Siirtola's [2] version of Parallel Coordinates allows sub-ranges within dimensions to be brushed or selected to highlight particular groups of objects, and these groups can in turn be combined or filtered with other selections on other dimensions. The axes could also be rearranged to enable the user to order the dimensions as they saw fit. Fua *et al* [3] developed hierarchical parallel co-ordinates that showed representative paths for groups of similar objects, and used colour shading cues to indicate the spread of the object groups represented by these single lines, thus reducing the clutter and overhead of displaying the full set of co-ordinates. Falkman [4] extended the technique to 3 dimensions with a parallel plane visualisation, though unfortunately this also

combined the difficulty of following lines in dense parallel co-ordinate displays with the occlusion problems of 3D representations. Further interactions such as *angular brushing* have also recently been proposed by Hauser *et al* [5], which picks out objects with specific *trends* between two dimensions rather than objects which are grouped together by value in just one dimension. However, one remaining problem is that if two objects share the same value in a particular dimension, they will share the same point on the corresponding axis, and as such their respective poly-lines will appear to merge and then separate again. Without additional cues such as colour, it is impossible to determine which line is which after the merge and separation effect. Such a situation is shown in the line diagrams and parallel coordinate screenshot in Figures 1 and 2, with a close-up of the affected area in Figure 3.

## 2. Motivation

OPAL is an EC-funded research project whose aim is to help businesses quickly form or select project groups that fit the skill requirements and attribute profiles for project tenders. Part of this process will involve selecting individuals according to data carried on their CVs or elicited from online interviews and interactions. Thus, each person will have an associated information profile that forms an instance of a multi-dimensional data set, and so parallel coordinates were investigated as a possible interface metaphor for this data.

In Inselberg and Dimsdale's original work on parallel co-ordinates the information used was from geometrical spaces, and as such the dimensions visualised tended to be continuous; mostly representing real or extended vector spaces; and homogenous, in that the units for one dimension tended to be the same for other dimensions. In doing so, information was also encoded into the lines between the axes; they didn't just act as visual markers for following values and objects from one axis to the next. Hence, a line that sloped upwards from the X to the Y-axis on the parallel co-ordinates represented a point with a higher Y than X value. The intersection of two lines between axes in parallel co-ordinates represented information about the line between two points in

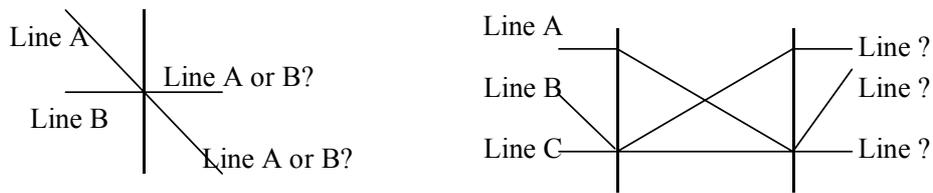


Figure 1. Crossover uncertainties result when lines share a point on an axis.

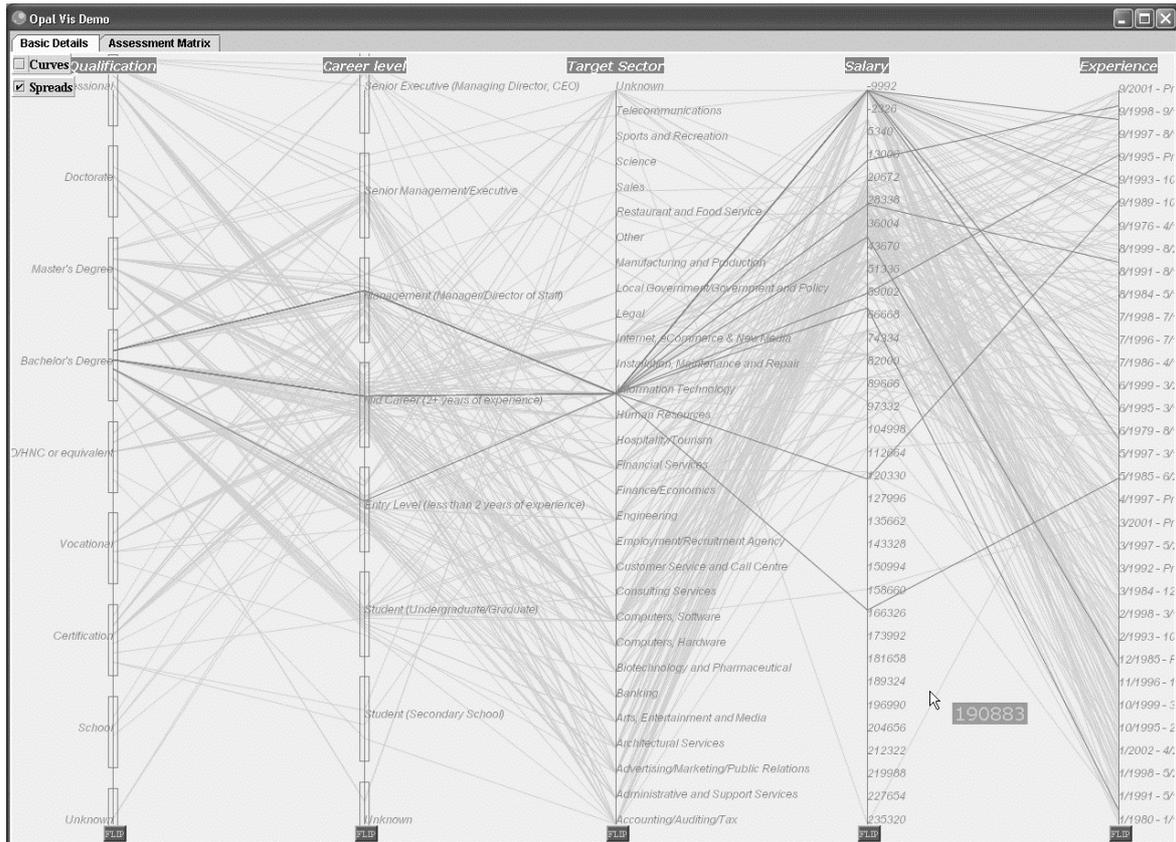


Figure 2. A crossover ambiguity in our CV data set.

geometrical space. However, when dealing with discrete, heterogeneous data dimensions, such as those found in information visualisation, these meanings are lost. A poly-line that slopes upwards from axis A to axis B does not mean that a particular object has a higher value in dimension B than A, as the number of discrete points on both axes may differ, and the units used on each dimension may be impossible to compare in any meaningful way. Dimensions themselves may have no particular ordering to enforce on an axis' scale, relying instead on an arbitrary scale such as alphabetical or numerical ranking. As such, if for abstract information sets, the lines have lost their usefulness apart from

representing continuity, we may as well attempt to use their ability to represent that continuity more fully.



Figure 3. Detail of Figure 2.

### 3. Parallel Coordinate Curves

This paper demonstrates a static method for resolving the difficulty of following lines that share common points on axes, the ‘cross-over’ problem. Our solution is based on the use of curves and the Gestalt principle of good continuation (Wertheimer [6]). Simply put, using a smoothly graduating curve over the parallel co-ordinates instead of a zig-zagging line allows users to discern individual paths through these knots for reasonable numbers of lines.

Curves have been used in parallel co-ordinate visualisations previously by Theisel [7], to show correlations between non-adjacent axes. In their work, additional axes could be placed between two adjacent axes, and the values of the objects on these dimensions act as control points for curves, pulling the curve towards them. Nesbitt and Friedrich [8] used Gestalt principles to improve the animation of dynamic graphs, and Bartram [9] used the Gestalt ‘law of common fate’ (objects moving in a similar manner appear to be grouped) to use animation as a filtering and brushing technique on a scatter plot style visualisation.

In our approach though, we use curves solely to help differentiate poly-lines that cross at axes, an occurrence that increases dramatically when using axes with a few discrete values. This technique works well in isolation on a few lines as does brushing, but performs more strongly in conjunction with brushing to distinguish object representations. Indeed, as a spatial variable, curvature can be used in conjunction with any of the previous filtering and brushing methods that distinguish poly-lines with colour variables. The continuous curve approach also has the benefit of reducing the need to have the pixel-accurate steady mouse hand that is sometimes necessary with the brushing interaction techniques in parallel coordinate displays. Figures 4 and 5 show the effect of applying curves to the same data shown in Figures 1 and 2 respectively. Note that the technique will not work for large numbers of crossings, as the situation in those circumstances is more akin to the general line-crossing problem shown in graph drawing by Purchase [10].

Our instance of a parallel coordinate visualisation runs over a data set of CVs, showing simple, singular attributes such as expected wage, work sector and highest educational qualification achieved. Some of these dimensions, such as educational level, have only a few discrete points, upon which many objects share a value. In the visualisation, this maps to poly-lines converging upon a few points on these particular axes.

In standard parallel co-ordinates, finding which entering line maps to which exiting line is a matter of chance. For  $N$  poly-lines converging and then diverging to and from a particular point, a user would have a 1 in  $N$

chance of following a particular poly-line correctly without additional cues. With a reasonable number of curves, the human visual processing system can follow the most likely smooth path across the point of intersection, the gradients of the curves giving visual clues as to the relationships, and shapes such as asymptotic inflections can be rapidly dismissed as candidates. The smooth curves produced in our visualisation enable simple situations such as 4 or 5 crossing curves to be quickly resolved visually without brushing, and if the situation is more complicated, exploratory brushing can at least then be concentrated on the more likely candidates.

In theory, a series of  $n$  points can be connected by a single curve described by a polynomial series reaching a highest power of  $n-1$ . However, Java 1.4 supplies ready-made classes for quadratic and cubic curves, so we use these and seamlessly link them together to produce the higher order curve over the  $n$  attribute values. This also circumvents the problem that discovering a polynomial that passes exactly through a set of  $n$  points involves solving a system of  $n+1$  linear equations. This is described as computationally expensive by Davies and Samuels in their introductory text [11], a situation that is usually inappropriate for any rapid, interactive system. Their text also contains a more formal, mathematical explanation of the basics of curve drawing.

The curve segments contain two endpoints, which map to the points on the axes, and either one or two control points, which influence the shape of the curve. Mathematically, two control points produce a cubic equation using their co-ordinate values as factors, whilst one control point produces a quadratic equation that describes the shape of the curve. Visually, the control points appear to pull the curve towards them. Smoothness between adjoining curve sections (i.e. making the tangents at the end points of abutting curves parallel) is achieved by making sure the last control point of the preceding curve segment, the joint endpoint co-ordinate that sits on the attribute axis, and the first control point of the following curve segment form a straight line.

With the curved representations, the parallel coordinate display now resembles the oscilloscope-style plots of Andrews’ curves [12] for multi-dimensional data, though Andrews produced his representations using a continuous Fourier-style function over the relevant data points.

The decision on whether to use a quadratic or cubic curve for a particular section is made on the basis of the general shape formed by that section and the two adjacent sections. If the vertical displacement to be covered by the current section is outside the range delimited by the displacements for the adjacent sections then an inflection has been detected and a cubic curve will be used. Though a series of quadratic curves would always be sufficient to produce a smooth curve, a cubic section can provide an

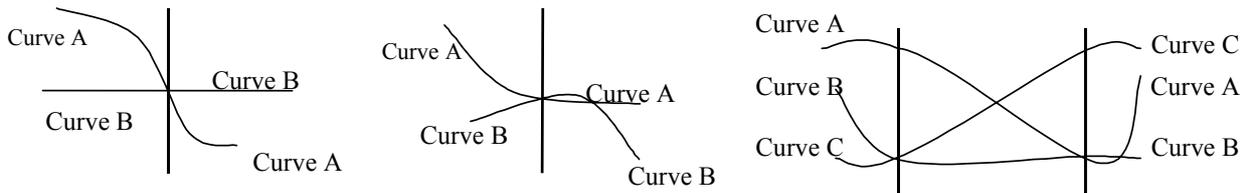


Figure 4. Curves make the crossings easier to resolve

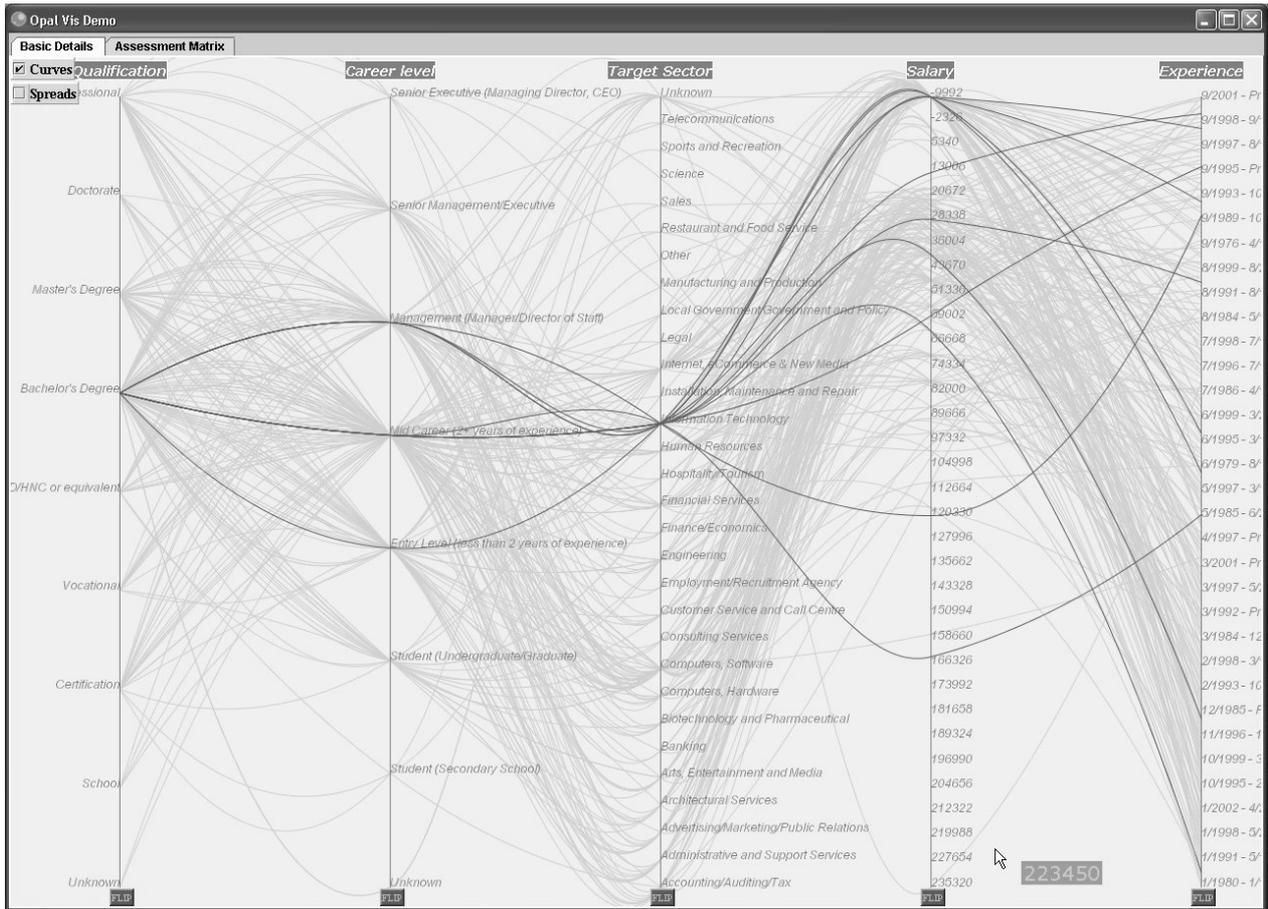
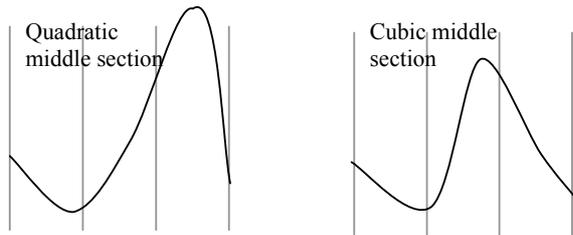


Figure 5. As per Figure 2, but with curves instead of straight lines

inflection that would otherwise need to be drawn by a combination of two adjacent quadratic curves, as shown in Figure 7. In these circumstances, drawing solely with quadratic curves can sometimes produce wildly oscillating curves whose paths go off-screen.

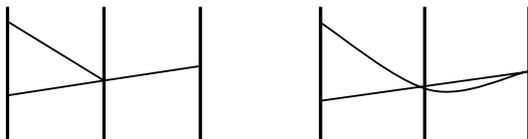


Figure 6. Close up of Figure 5. Brushing confirms the continuation.



**Figure 7. Cubic sections can stop curves oscillating by allowing them to perform inflections between axes.**

One other feature of using curved sections in parallel co-ordinates is an apparent increase in the number of objects displayed in the visualization. For instance, in the left-hand side of Figure 8 there appears to be only one line between the second and third axes, but in the curved representation on the right-hand side there are now two curved sections in the same area. This occurs because curves are a function not only of the steepness of the original line between two adjacent axes, but also of the gradient of the previous line section (as to ensure a smooth curve over all the axes, the last and first control points of the preceding and following curves respectively must lie on a straight line with the shared endpoint). Thus, two objects that had the same values on two adjacent axes and would overlay exactly when using straight lines, would now diverge slightly in the middle of this section if they had different values on the directly preceding axis. This can be viewed as either a useful or distracting effect, depending on circumstance; distracting in that it increases the number of perceived objects on screen, but useful in that it gives a more realistic presentation of the amount of data that is present.



**Figure 8. Curves mean that object representations tend to resolve from each other over shared values**

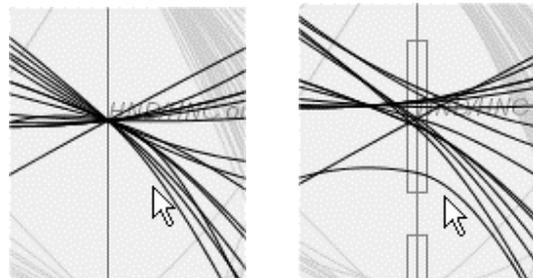
Understandably, drawing curves takes more time and processing than drawing simple straight lines. However, by using efficient repainting techniques, such as only redrawing curves that have changed state to or from the background state, and only recalculating curves when coordinates change due to axis reordering, disabling or flipping, the rate of interaction for tasks such as brushing is still high, which has been tested on data sets of up to 4,000 items on a 2 GHz PC.

#### 4. Spreading points and Focus & Context

Although the curves clearly aided the separation of items, it was found that with many curves it became difficult to differentiate them if they were bunched close together along their paths. Methods of separating such curves were needed, and two complementary techniques were developed, which can also be applied to traditional parallel co-ordinates.

Firstly, it was noticed grouping was greater around coordinates on axes with relatively few values, as the curves converged on a few designated points on that axis, with much of the axis composed of empty space in between. Thus, if there was empty space available, we decided to 'spread' the point out to become a short line along the axis, with a bounding box to clarify that this length still formed one and the same value.

Figure 9 demonstrates the effect this has, with instead of one complicated crossing point, many simpler crossing points being formed. The spread is calculated by moving the axial crossing point for each line to a point in the bounding box that is proportional to its average position in the preceding and following axes. Thus the curve at the bottom of the bounding box in Figure 9 is placed there because it crosses close to the bottom bounds of the preceding and following axes. Also, as the crossing points tend to drift away slightly from the axis, it is also possible to follow straight-line co-ordinates through these situations for small numbers of items, as in Figure 10.



**Figure 9. Before and after spreading is introduced at an axis point.**

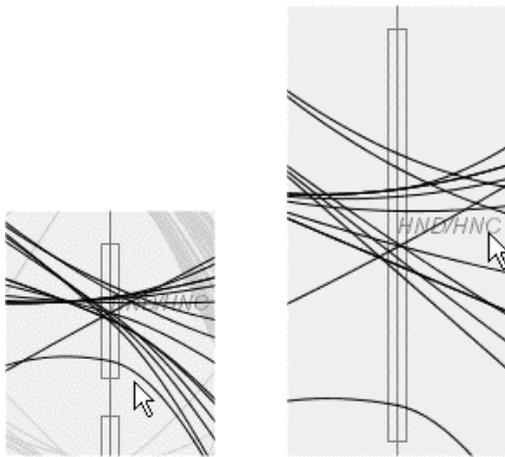


**Figure 10. Spreading with traditional straight line representations.**

Following on from this, in some axes it was noticed that there was not enough space to perform the spread effect even when that axis had a relatively small number of discrete values,

due to a space being necessary between the points themselves to show visual separation. It was then decided to implement a simple focus+context technique on the axes, with ranges of selected values being given more space to the detriment of unselected items. This would give the selected items the necessary space to perform the spreading technique. It would also further separate curves on axes that already had enough space to spread out shared points.

An example of the difference this makes is shown in Figure 11, where a chosen data point has expanded under the focus and context effect and a previously impenetrable knot of crossings has been loosened to the point where individual curves can be discerned.



**Figure 11. Before and after a focus and context effect is introduced.**

Andrienko and Andrienko [13] described a related problem in that exceptional outliers could skew a scale such that the vast majority of data points on an axis were squeezed into a minority of the space. Their solution involved a statistical analysis of the attribute data, dividing the axes into portions according to quartiles or standard deviations from the mean of the data. These scaled points were also aligned across the different axes, so that a line drawn through the mean or first/third quartile points for all attributes formed a straight line across all the corresponding axes on-screen.

## 5. Future Work

Initial user testing has begun by observing a handful of representative users acting on sample tasks with the visualisation. They were able to follow curves across screen for individual elements given reasonable numbers of items. This was especially apparent if the item in question was an outlier with an unusual path across the

axes. Further feedback has resulted in a number of suggestions that may be incorporated into future development, such as having on-screen formal descriptions of the current selection and its result (i.e. “where SALARY < 60000 and SECTOR = Finance: 125 Matches”).

Interaction with the parallel coordinate view may well be served better by using a mixture of curved and straight-line representations for the data items, with selected and brushed items being represented by the curves, and background items by traditional poly-line representations. This would not only reduce clutter on-screen, as the straight-line descriptions of background items will tend to overlap, but the speed of the interaction may also be slightly improved as only a subset of items need to be drawn as the more computationally expensive curves. This method will be investigated in the course of future work.

## 6. Conclusions

This work successfully combines a number of techniques for use with parallel coordinate visualisations that allow individual paths, and thus elements in an information set, to be distinguished when samples of the information set have been selected. These include using curves to produce paths with continuous gradients across axes, spreading out points on axes that have few discrete values, and a simple focus and context technique for then giving further space to points of interest on axes. The latter two techniques can also be used on standard parallel co-ordinates that display using straight lines, and all can be used in conjunction with the common colour coding techniques used when selecting and brushing items in parallel coordinate visualisations.

Early user testing has shown that curves can be followed across axes on-screen without the need to use brushing, an interaction technique for which pixel-perfect accuracy is sometimes needed. Following curves was easiest when the elements were outliers and had the most distinct curvatures, and these elements in many tasks are the most interesting cases to examine. Future work will consist of refining the technique with users, and to discover when the use of curves is most appropriate.

## 7. Acknowledgements

This work is part of the OPAL research project, funded by the EC under the Information Society Technologies programme (Ref: IST-2000-28295).

## 8. References

- [1] A. Inselberg and B. Dimsdale, "Parallel Coordinates: A Tool for Visualizing Multidimensional Geometry," *Proc. IEEE*

*Visualization 1990*, IEEE Computer Society Press, San Francisco, California, USA, October 23-25, 1990, pp. 361-378.

[2] H. Siirtola, "Direct Manipulation of Parallel Coordinates," *Proc. Information Visualization IV'00*, IEEE Computer Society Press, London, UK, July 19-21, 2000, pp. 373-378.

[3] Y.-H. Fua, M.O. Ward and E.A. Rundensteiner, "Hierarchical Parallel Coordinates for Exploration of Large Datasets," *Proc. IEEE Visualization '99*, IEEE Computer Society Press, San Francisco, California, USA, October 24-29, 1999, pp. 43-50.

[4] G. Falkman, "Information visualization in clinical Odontology: multidimensional analysis and interactive data exploration," *Artificial Intelligence in Medicine*, vol. 22, no. 2, 2001, pp. 133-158.

[5] H. Hauser, F. Ledermann and H. Doleisch, "Angular Brushing of Extended Parallel Coordinates," *Proc. To appear in IEEE InfoVis 2002*, IEEE Computer Society Press, Boston, Massachusetts, USA, October 28-29, 2002, pp.

[6] M. Wertheimer, *Laws of Organization in Perceptual Forms*. In *A source book of Gestalt psychology*. W. Ellis. London, Routledge & Kegan Paul, 1938, pp. 71-88.

[7] H. Theisel, "Higher Order Parallel Coordinates," *Proc. Vision, Modeling and Visualization 2000*, Aka, Berlin, Saarbrücken, Germany, November 22-24, 2000, pp. 119-125.

[8] K.V. Nesbitt and C. Friedrich, "Applying Gestalt Principles to Animated Visualizations of Network Data," *Proc. IV 2002*, IEEE Computer Society Press, London, UK, 10-12 July, 2002, pp. 737-743.

[9] L. Bartram and C. Ware, "Filtering and brushing with motion," *Information Visualization*, vol. 1, no. 1, 2002, pp. 66-79.

[10] H.C. Purchase, "Effective information visualisation: a study of graph drawing aesthetics and algorithms," *Interacting with Computers*, vol. 13, no. 2, 2000, pp. 147-162.

[11] A. Davies and P. Samuels, *An Introduction to Computational Geometry for Curves and Surfaces*, Oxford University Press, Oxford, UK, 1996, p. 213.

[12] D.F. Andrews, "Plots of high dimensional data," *Biometrics*, vol. 28, no. 1, 1972, pp. 125-136.

[13] G. Andrienko and N. Andrienko, "Constructing Parallel Coordinates Plot for Problem Solving," *Proc. 1st International Symposium on Smart Graphics*, ACM Press, Hawthorne, New York, USA, March 21-23, 2001, pp. 9-14.