

Angular Brushing of Extended Parallel Coordinates

Helwig Hauser, Florian Ledermann, and Helmut Doleisch
Hauser@VRVis.at, flo@subnet.at, and Doleisch@VRVis.at

VRVis Research Center in Vienna, Austria
<http://www.VRVis.at/vis/>

Abstract

In this paper we present several extensions to the well-known InfoViz technique of parallel coordinates, mainly concentrating on brushing and focus+context visualization. First, we propose angular brushing as a new approach to high-light rational data-properties, i.e., features which depend on two data dimensions (instead of one). We also demonstrate smooth brushing of parallel coordinates as an intuitive tool for specifying non-binary degree-of-interest functions (then used for F+C visualization). Thirdly, we show how composite brushes provide lots of flexibility during data exploration. Additionally, we also present several further and more general extensions to parallel coordinates which there are (a) histograms to be used as axis overlays, (b) interaction features such as axis re-ordering, flipping, scaling and panning, and (c) detail on demand, implemented as a mouse-over effect.

Keywords: information visualization, parallel coordinates, brushing, linear correlations, focus+context visualization

1. Introduction

For many years already, parallel coordinates [11, 13, 12] are well-known and an often used technique in information visualization (InfoViz). Multi-dimensional data-sets are visualized by drawing a poly-line – for every n -dimensional data-point – across coordinate axes which are laid out in parallel, side-by-side (see figure 1 for an example, four out of a dozen dimensions shown). The poly-lines intersect the parallel coordinate axes at points which correspond to the respective values (dimensions) of the particular data-points. Thereby, up to a dozen of data dimensions (or even more) are well visualized, especially when the visualization is combined with proper interaction features such as real-time re-ordering of coordinate axes or interactive brushing in single dimensions [15]. If used as an interactive tool, parallel coordinates are especially effective in visualizing correlations between neighboring axes, outliers, etc.

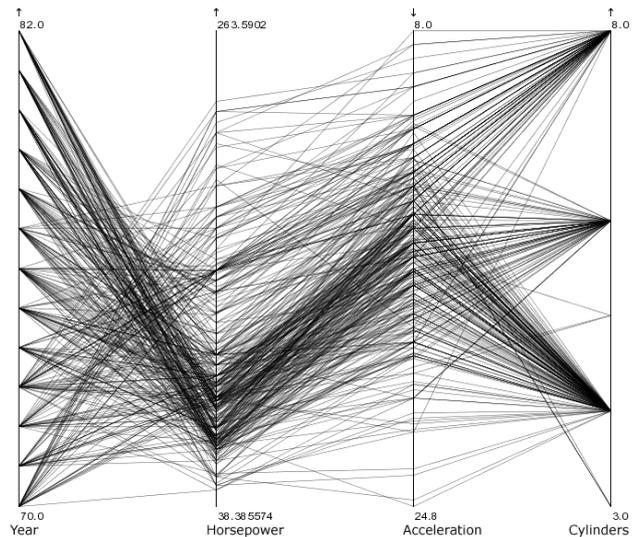


Figure 1. Parallel coordinates, a sample view: the cars data-set is shown, four (out of twelve) data dimensions are visualized.

Brushing – as already mentioned above, brushing is a very effective technique for specifying an explicit focus during information visualization [1, 20, 22]. The user actively marks sub-sets of the data-set as being especially interesting, for example, by using a brush-like interface element. If used in conjunction with multiple linked views, brushing can enable users to understand correlations across multiple dimensions [1, 2, 10, 9].

Focus+context visualization – brushing also is very useful to steer a drill-down into visualizations of really big data-sets [17] – by specifying a (limited and limiting) focus, more details can be shown for the selected data-points. This relates to another very important InfoViz concept which is focus-plus-context (F+C) visualization [8, 3]. The basic idea of F+C visualization is to jointly visualize some parts of the data in detail – usually those parts which are of prime

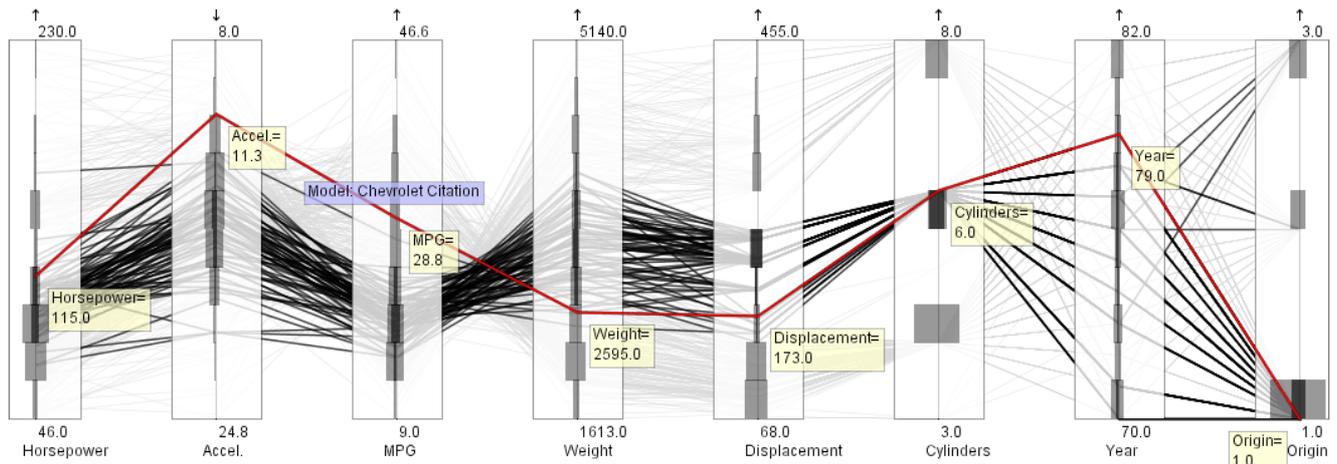


Figure 2. Extended parallel coordinates, a sample view of the *cars* data-set: cars with six cylinders were emphasized through brushing, histograms are laid over axes, and one data-point is shown with all details.

interest at a particular point in time – and also show the rest of the data-set, but in reduced style (smaller, accumulated, etc.) – this is done to maintain the user’s orientation and to ease navigation in very large data-sets.

Degree-of-interest functions – for specifying which parts of the data are to be drawn in detail, i.e., for specifying the focus, a degree-of-interest (DOI) function is used [8], usually assigning a value of 1 or 0 to each data-point, depending on whether the data-point is of current interest or not. Brushing is an effective method to actively (and explicitly) assign values of 1 to data-points of interest [1, 22].

Brushing extensions – in addition to standard brushing, several useful extensions have been proposed previously. The introduction of composite brushes [15], for example, enabled users to more specifically define their focus. However, it was also shown that interaction elements of the user interface must be chosen carefully due to the very high number of possible ways of compositing a complex brush [22]. The combination of brushing and non-binary degree-of-interest functions [15] proved to be useful for data-sets with gradual changes in at least one of their data attributes, for example, data from scientific simulation [4]. Also, rather complex extensions to brushing have been proposed as, for example, structure-based brushes, dealing with hierarchical data-sets [6].

In this paper, we propose several small but very effective extensions to brushing as an interaction technique for parallel coordinates (like angular brushing or smooth brushing, see section 2). Thereby, parallel coordinates can be furthermore improved as an effective tool for information visualization and for visual data mining.

General extensions to parallel coordinates – during the long time that they have been an integral part of InfoViz research, several extensions to the original parallel coordinates have been presented. A hierarchical extension was proposed [5], for example, integrating automatic clustering to deal with really big data-sets. Furthermore, a number of 3D extensions to parallel coordinates were presented [21], including extruded parallel coordinates or linking with wings, for example. Also, higher-order parallel coordinates were proposed [18], using higher-order splines to interpolate values on more than two axes.

In this paper, we also present a number of further general extensions to parallel coordinates, which we have found useful during data investigation (such as adding histograms on single axes or a detail-on-demand option, see figure 2 and section 3).

The remainder of this paper is structured as follows: first we present several extensions which special focus on brushing (section 2). Then, we continue with further, more general extensions to parallel coordinates (section 3). Later in this paper, we also shortly present some implementation issues, talking about fast rendering, for example (section 4). The application itself and some results from visual data analysis are demonstrated in section 5.

2. Extended Brushing for Parallel Coordinates

As already discussed above shortly, brushing effectively enables the user to (explicitly) specify his or her focus. In parallel coordinates, this usually is accomplished by marking a certain data sub-set of interest according to values within a single data dimension. In the *cars* data-set [19],

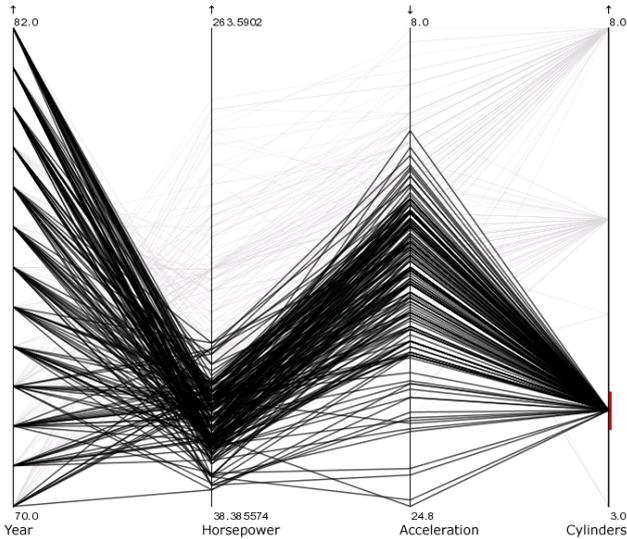


Figure 3. Brushing parallel coordinates, an example: all cars with four cylinders are marked and emphasized (only four data dims. shown).

for example, a user could focus on cars with four cylinders by brushing the respective portion of the *cylinders*-axis. See figure 3 for a sample view – note, that the third axis (“Acceleration”) was flipped, since values are given in seconds ’til 60 MPH, and therefore a smaller number represents a higher acceleration.

In our application of visually investigating multi-dimensional and large data-sets originating in a physically-based simulation of dynamic processes (computational fluid dynamics, CFD, for example, see section 5 for more details), we found standard brushing of parallel coordinates very useful, however, we needed to go further. The requirement was to specify a focus not only in dependence of one data attribute (one dimension), but to do so with respect to at least two of them – like brushing all those data-points which feature a *z*-velocity larger than the respective *x*-velocity. The consequence was to introduce a new brushing technique which we called *angular brushing*.

2.1. Angular Brushing of Parallel Coordinates

One strength of parallel coordinates is its effectiveness of visualizing relations between coordinate axes. Bringing axes next to each other in an interactive way, the user can investigate how values are related to each other with special respect to two of the data dimensions. This way, it is not only possible to see whether data-points exhibit high or low values in single dimensions, but also their relation can be understood. More specifically, the slope of the line-segments in-between two axes tells the user, whether there

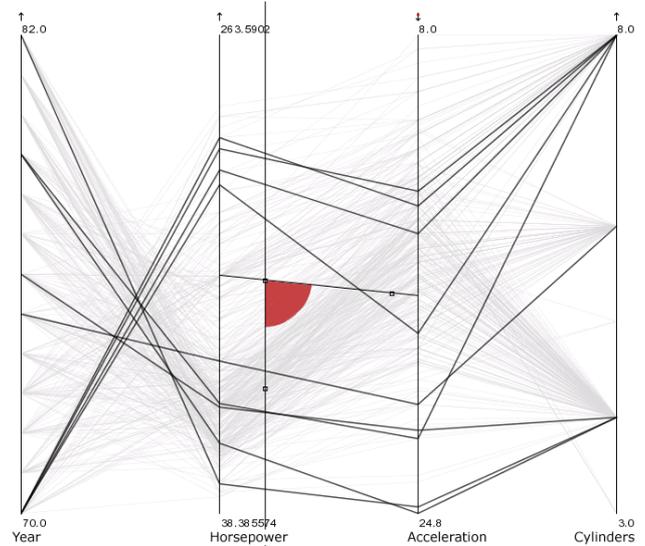


Figure 4. Reading between the lines: whereas most line-segments go up in-between the 2nd and the 3rd axis (visualizing a positive correlation of values there), just a few go down – those have been emphasized through angular brushing.

is a positive or negative correlation in-between values (of course assuming that the user set up a proper mapping). Outliers also can be easily found with respect to two data dimensions – when all but a few line-segments have a positive slope, then the few others clearly stand out (see figure 4 for an illustration).

In this paper, we demonstrate how this feature of parallel coordinates easily can be exploited for an extended brushing technique, i.e., angular brushing. In addition to standard brushing, which primarily acts along the (parallel) axes, we enable the space in-between axes for brushing interaction, as well. The user can interactively specify a sub-set of slopes which then yields all those data-points to be marked as part of the current focus, which exhibit the matching correlation in between the brushed axes. See figure 4 for an example, where all negative slopes have been marked in-between the second and third axis.

Angular brushing as described above is a useful extension to parallel coordinates, also because it very well corresponds to the effective visual cues provided by parallel coordinates. Inherently, angular brushing opens the extended field of brushing multi-dimensional properties of high-dimensional data-sets. In addition to composite brushes which are composed multiple single-axis brushes by the use of logical operators (see also below in section 2.3), angular brushes allow the selection of rational properties with respect to two of the data dimensions. When compared to

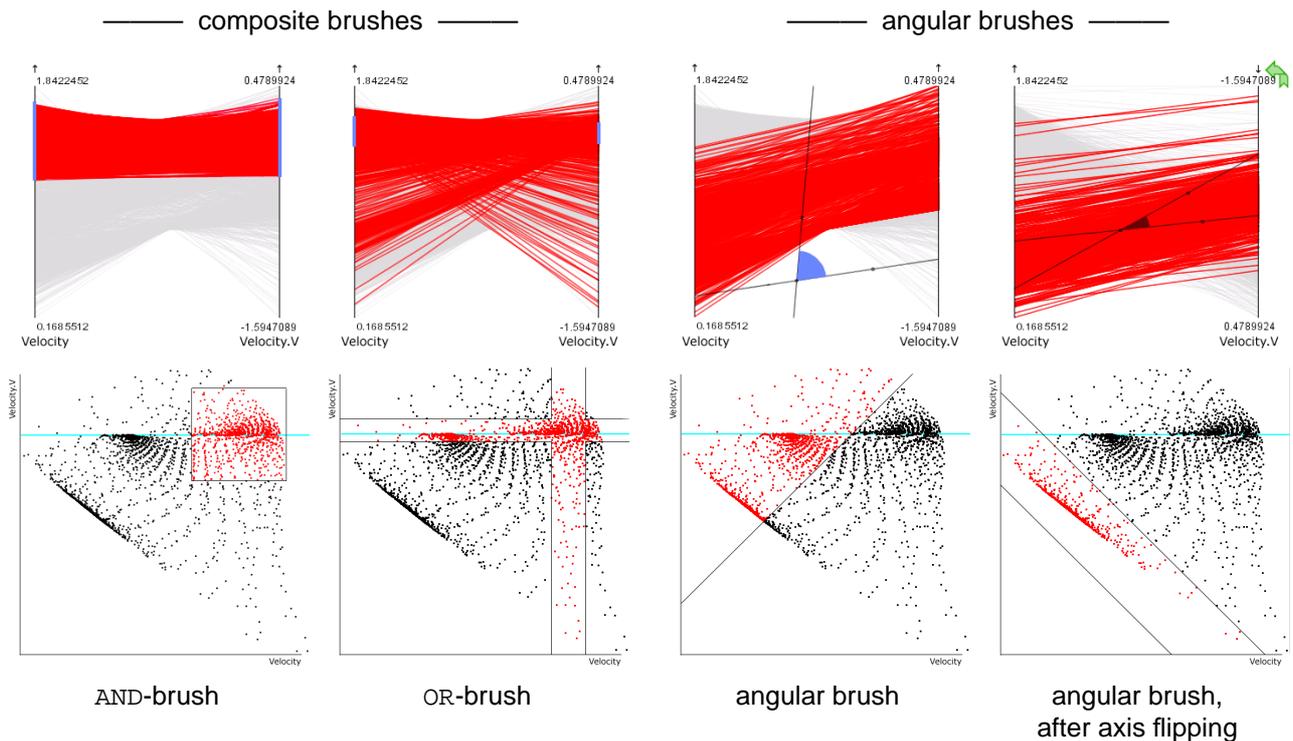


Figure 5. Comparing composite brushes (on the left) and angular brushes (on the right), using parallel coordinates (top row) and scatterplots (bottom row): composite brushes address “horizontal/vertical” features, whereas angular brushes emphasize “diagonal” feature.

composite brushes by the use of a scatterplot visualization, we clearly see that brushes based on logical operators select sub-sets which are aligned with the display axes, whereas angular brushes select sub-sets which are aligned with the diagonals when visualized in a scatter-plot (see also figure 5 for comparison).

2.2. Smooth Brushing and Non-Binary DOIs

Many well-known F+C techniques in InfoViz such as fish-eye views [7, 8], for example, do not use a discrete distinction between focus and context, but allow a multi-valued or even continuous transition, which inherently supports the mental connection between data-points in focus and their context. This corresponds to a degree-of-interest function, which non-binarily maps into the $[0, 1]$ -range.

Often, such a non-binary DOI function is defined by means of spatial distances, i.e., the DOI-value reflects the distance of a data-point from a so-called center-of-interest (COI), which in these cases often is specified explicitly, for example, by pointing to it. In the implicit case, i.e., when querying is used for the specification of the DOI function, fuzzy logic can be used in the style of probabilistic responses.

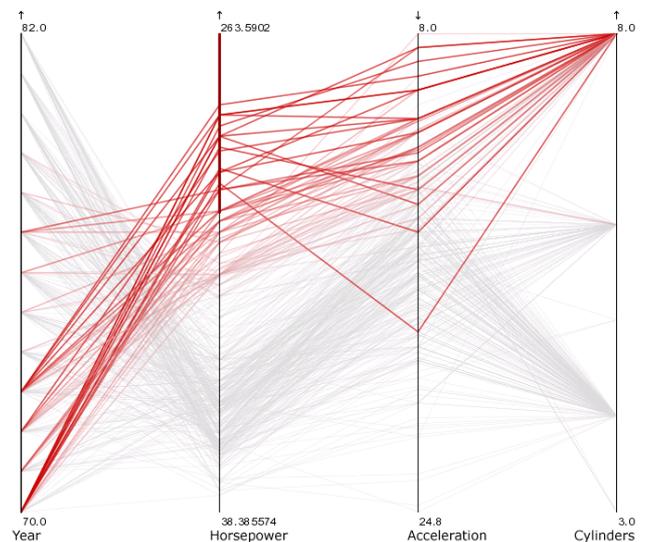


Figure 6. Smooth brushing, an example: note the gradual changes of drawing intensity which reflect the respective degree of interest, after smooth brushing of the 2nd axis.

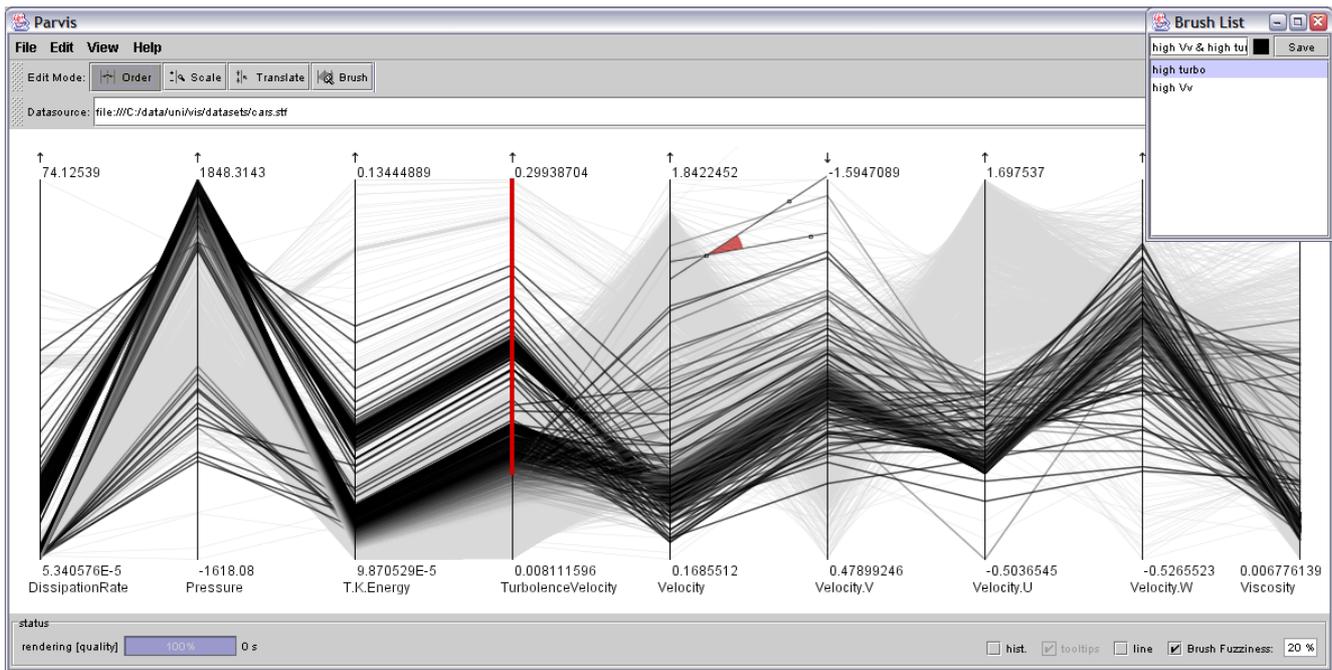


Figure 7. An example of a composite brush (AND): a CFD data-set is shown with all data-points high-lighted, which exhibit high values of turbulent velocity (single-axis brush, smoothly along “TurbulenceVelocity”-axis) and relatively low y -velocity (“Velocity.V”) when compared to 3D-velocity (angular brush after axis flipping).

Brushing also easily combines with non-binary DOI functions [15], i.e., when smoothly delimited brushes are used. Thereby, at the brush boundaries, fractional DOI-values between 0 and 1 are assigned. When composite brushes are composed through the use of logical operators, fuzzy logic is employed to aggregate final DOI-values for every single data-point. In previous work, we already demonstrated the useful application of smooth brushing in the F+C visualization of multi-dimensional data, originating in 3D flow simulation [4].

In conjunction with parallel coordinates we extend the interactively specified brush by a certain, user-defined percentage to accommodate a smooth gradient of DOI-values at its borders. When combining smooth brushes through logical operators, we apply the Łukasiewicz norm [16], known from fuzzy logic as a comparably large norm [14], i.e., a norm which (when applied repeatedly) only slowly converges to 0 and therefore better conserves the transitional DOI-gradients, for computing AND- as well as OR-combinations. See figure 6 for an example of smoothly brushed parallel coordinates.

2.3. Composite Brushes for Parallel Coordinates

Shortly after the original idea of brushing, also the logical combination of brushes was proposed as a useful extension [15]. As soon as the user’s imagination about the

features he or she wants to see within a data-set gets more and more concrete, a need for more flexible and multi-dimensional descriptions of features arises. User-centered specifications like *all those cars which are fast but do not cost a lot*, for example, need to be supported by visualization. Composite brushes, i.e., brushes which are composed of simple ones by the use of logical operators, are an intuitive solution to the above mentioned requirement.

In our application (see also section 5), composite brushes were to be enabled on the basis of single-axis brushes, angular brushes (see above, section 2.1), and smooth brushes (see above, also). We therefore implemented the following features to end up with a consistent solution:

Multiple brushes – whenever a new brushing interaction is triggered, a new brush is instantiated, represented by a new entry in a separate brush list (separated part of the user interface). New brushes can be named by simply editing the new entry in the brush list. Hereby, a useful overview is provided while still enabling the user to switch between previously defined brushes (to compare them, for example), or to combine them to form composite brushes.

Composite brushes – single brushes easily can be combined to form composite brushes through the application of logical operators. A nice little feature in our

application is that new (composite) brushes automatically derive their names from the brushes involved in the composition. Composite brushes also can be constructed by adding new constraints interactively, using modifier keys through interaction. As already described above, composite brushes are computed based on the Łukasiewicz norm, when smooth brushes are involved (see also figure 7).

Support of linking – in our application, DOI-values are considered to be first-rank data-derivatives, i.e., brushes can be loaded from and saved to disk, or exchanged with other views, for example. Thereby, the here described parallel coordinates also can be used as one of several linked views, supporting a *linking-and-brushing* style of application. See figure 8 for an example where parallel coordinates were combined with a 3D scientific view. In this example, a scatter-plot was used for brushing and the parallel coordinates as well as the 3D view are linked for F+C visualization.

3. Further Extensions for Parallel Coordinates

In addition to the brushing extensions as described above, we also developed several further extensions to parallel coordinates, which we found useful during data investigation. Next to simple but yet very useful interaction features such as coordinate flipping, the addition of histograms along all of the parallel axes was found to be especially useful.

3.1. Histograms over Parallel Coordinates

Similar to a previously published solution [22], we also compute histograms for all the axes shown in the parallel coordinates view. This is especially useful, when many data-points are to be shown. While users might be troubled with resolving clutter problems with an over-full display, histograms give a very intuitive clue on where data-points accumulate along coordinate axes.

Depending on a user-defined sub-division, multiple bins of equal extension are generated, and the data-points are sorted into these bins during the build-up of the parallel coordinates view. Afterwards, a semi-transparent representation of the bins' cardinality is plotted on top of the parallel coordinates view, comparable to a histogram along this data dimension. See figure 2 for an example of parallel coordinates plus histogram visualization along axes.

3.2. Flexible Layout with Parallel Coordinates

From the early beginning of parallel coordinates on, interaction was a crucial component of parallel coordinates views to enable the user to work with the view and to interactively explore the data under investigation.

In our application, we implemented a number of partly obligatory and partly new features in conjunction with the goal of providing a flexible layout the user can work with:

Axes re-ordering – probably most important is that the user can interactively re-order the layout of the axes in use. In our application, single axes can be dragged around, and automatically position themselves right in the gap (between two other axes, or at the boundary), where the user drops them.

Flipping of axes – also very useful for data investigation is an option which allows users to flip certain axes, i.e., to require a top-down ordering of values instead of the standard bottom-up style. By doing so, correlations between axes can be better investigated. Also, this option is especially important in conjunction with angular brushing as it enables the user to brush sub-sets, which – in a scatter-plot – would be aligned along the sub-dominant diagonal (see figure 5).

Changing the mapping – the default in our application is to use a min-max mapping of values to coordinate axes, i.e., minimal and maximal values are extracted before mapping, then values are linearly mapped such that minimal values come lowest whereas maximal values are mapped to the top of the coordinate axes. Additionally, two other modes can be used: (a) The user can enforce that zero is mapped to the same height along all the coordinate axes. (b) The min-max mapping can be correlated along all the axes, resulting in a consistent mapping for of them. Of course, if not all of the scales are comparable, this mode can result in a roughly distorted view.

Axis scaling and panning – the user can interactively scale and pan the mapping for single axes. On the first sight, this only adds more flexibility with adjusting the view layout. In our application, this option also provides important means when working in-between dimensions. Especially in conjunction with angular brushing, the ability to scale and pan one of the two axes (later to be brushed), proved to be very useful.

Deletion and addition of axes – in addition to interactive re-ordering of axes as shortly described above, we also enable the user to interactively delete single axes from the view. This is used to clarify the view, when not all of the information needs to be shown concurrently. We also allow the addition of axes to the view, even if they are already in use: often, single data dimensions are of special interest to the user and then it is very useful to have such an axis repeatedly in the view, especially as correlations in-between axes are best visualized when axes are next to each other.

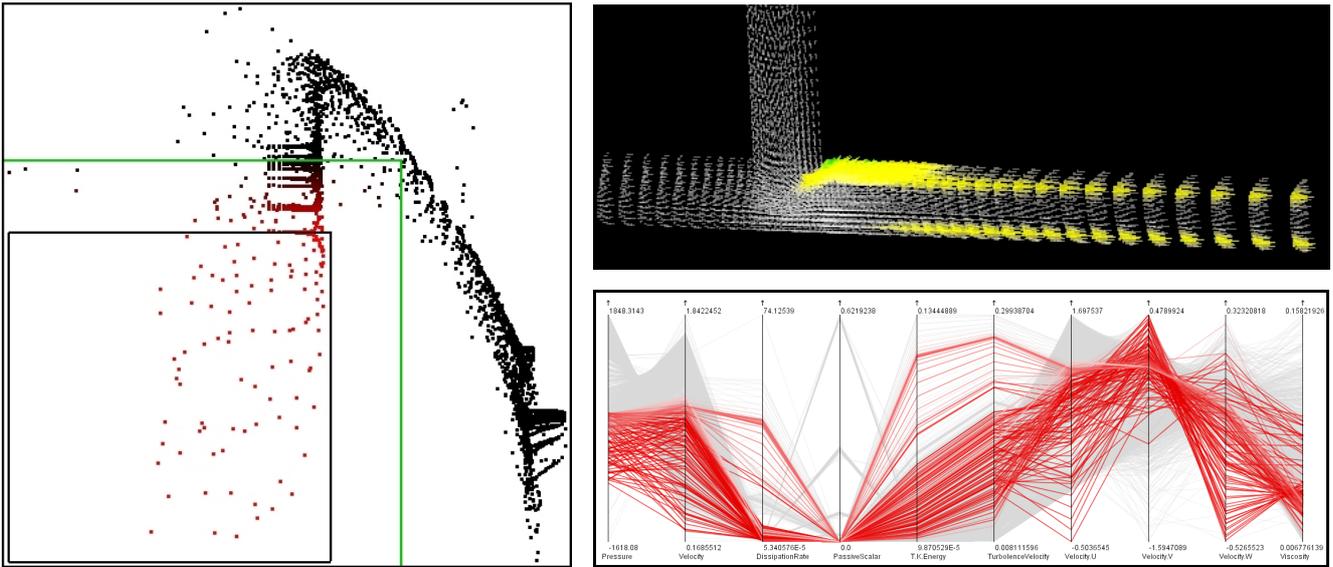


Figure 8. Linking and brushing, an example: in a scatter-plot (shown on the left side), smooth brushing was used to mark data-points of low pressure and low velocity; a linked 3D view (on the top right) shows the same data with the brushed data-points high-lighted; thirdly, the parallel coordinates view (on the lower right) also shows the same data, also high-lighting the brushed sub-set.

3.3. Detail on Demand

We also implemented a detail-on-demand feature, which we realized with a mouse-over effect. Thereby the user can interactively browse through the display and whenever the mouse moves over a poly-line, all the details are shown for the respective data-point. We also make use of this feature to display textual data attributes, which by default are not drawn as axes at all (problem of mapping).

4. Implementation

To deal with real-world data-sets – in our case consisting of 10^4 – 10^6 data-points each – we quickly experienced the limitations of a brute-force implementation of parallel coordinates. Simply drawing multiple line-segments for each and every data-point already poses quite a load on the graphics hardware – for data-sets as in our application, quickly millions of line-segments need to be drawn. We also needed semi-transparency, coloring, and anti-aliased line-segments to nicely integrate all of the features described in this paper.

In the following, we shortly describe some of the approaches we utilized to still provide real-time interactivity:

Exploiting coherence – during interaction, the application seeks to re-use as much of the current visual output as possible. When single axes are flipped, for example, only the inter-axis space on the left- and right-hand side of the flipped axis are re-rendered, all the rest is reused.

Progressive rendering – for immediate feed-back the application uses a preview-style of rendering, i.e., without anti-aliasing or semi-transparency. As soon as there is time for better rendering, the application computes a high-quality result and replaces the preview with the improved image.

Bi-threaded implementation – to de-couple rendering and interaction, we use a bi-threaded implementation. One thread serves the user interface and guarantees interactive response, while the other thread deals with rendering, which sometimes lasts for a number of seconds, especially for bigger data-sets.

5. Application and Results

In our application we investigate multi-dimensional data-sets which originate in CFD simulations (run by one of our partners), which usually exhibit a dozen of data dimensions, or more. During flow simulation, values like temperature, pressure, velocity, dissipation rate, and many others, are computed for all cells within a detailed CFD grid. Data like this needs careful analysis which usually is non-trivial, not only because the data is of high dimensionality, often laid out in 3D space or even time-dependent.

Parallel coordinates, with all the extensions presented in this paper, proved to be very useful for data investigation. The interactive character of the here described implementation allows for fast and flexible data exploration, even when

simultaneously investigating multiple dimensions. Linking parallel coordinates and other types of views (like a 3D SciViz view, shown in figure 8 on the top right) and using our brushing features eased data exploration. Whereas parallel coordinates are used to find interesting data-points, a linked 3D view can simultaneously show where those data-points reside in 3D space. Also the combination with scatter-plots was found to be useful, primarily because composite brushes are specified more easily in a scatter-plot than in parallel coordinates.

Figures 5, 7, and especially 8 show results of data exploration by the use of parallel coordinates. The data shown is the result of a simulation of flow through pipes shaped as a T-junction. Flow is coming from the left and the top (when referring to figure 8, top left) and leaving the T-junction to the right. In figure 7, a composite AND-brush was used to mark all those data-points which, on the one hand, exhibit rather high values of turbulence (smoothly brushed) and which are part of flow regions that are well aligned with the negative y -axis (“Velocity.V”), on the other hand.

6. Summary and Conclusions

In this paper, we have presented several useful extensions to parallel coordinates, especially focussing on brushing interactions. We proposed *angular brushing* as a new technique to effectively select data sub-sets which exhibit a data correlation along (at least) two axes. We also showed how *smooth brushing* can be used, even in combination with *composite brushes*. We also showed how histograms can be used as axis overlays to further support data exploration.

Concluding, we may say that we re-experienced the advantages of an interactive visualization tool when doing data exploration. We also conclude that the worth of a parallel coordinate visualization increases drastically when interaction features like axis re-ordering, etc., are added.

Acknowledgements

Parts of this work have been done in the scope of the basic research on visualization (<http://www.VRVis.at/vis/>) at the VRVis Research Center (<http://www.VRVis.at/>) in Vienna, Austria, which is funded by an Austrian governmental research program called Kplus. The authors also thank AVL (<http://www.AVL.com/>), i.e., one of VRVis’ partner companies, for providing real-world data to evaluate our software prototype. Last, but not least, the authors thank Robert Kosara for his valuable hints during the preparation of this paper.

References

[1] R. Becker and W. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[2] A. Buja, J. A. McDonald, J. Michalak, and W. Stuetzle. Interactive data visualization using focusing and linking. In *Proc. of IEEE Visualization ’91*, pages 156–163.

[3] S. Card, J. MacKinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, 1998.

[4] H. Doleisch and H. Hauser. Smooth brushing for focus+context visualization of simulation data in 3D. In *Proc. of WSCG 2002*, Plzen, Czech Republic.

[5] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proc. of IEEE Visualization ’99*, pages 43–50.

[6] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Structure-based brushes: A mechanism for navigating hierarchically organized data and information spaces. *IEEE Trans. on Visualization and Computer Graphics*, 6(2):150–159, 2000.

[7] G. W. Furnas. The FISHEYE view: A new look at structured files. Technical Memorandum #81-11221-9, Bell Laboratories, Murray Hill, New Jersey 07974, U.S.A., 1981.

[8] G. W. Furnas. Generalized fisheye views. In *Proc. of the ACM CHI ’86 Conf. on Human Factors in Computing Systems*, pages 16–23, 1986.

[9] D. Gresh et al. WEAVE: A system for visually linking 3-D and statistical visualizations, applied to cardiac simulation and measurement data. In *IEEE Vis. 2000*, pages 489–492.

[10] C. Henze. Feature detection in linked derived spaces. In *Proc. of IEEE Visualization ’98*, pages 87–94.

[11] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–92, 1985.

[12] A. Inselberg. A survey of parallel coordinates. In H.-C. Hege and K. Polthier, editors, *Mathematical Visualization*, pages 167–179. Springer Verlag, Heidelberg, 1998.

[13] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multidimensional geometry. In *Proc. of IEEE Visualization ’90*, pages 361–378.

[14] E. P. Klement, R. Mesiar, and E. Pap. *Triangular Norms*, volume 8 of *Trends in Logic*. Kluwer Academic Publishers, Dordrecht, 2000.

[15] A. Martin and M. O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *Proc. of IEEE Visualization ’95*, pages 271–278.

[16] B. Schweizer and A. Sklar. *Probabilistic Metric Spaces*. Noth-Holland, New York, 1983.

[17] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. of the IEEE Symp. on Visual Languages*, pages 336–343, 1996.

[18] H. Theisel. Higher order parallel coordinates. In *Proc. of the 5th Fall Workshop on Vision, Modeling, and Visualization*, pages 415–420, Saarbrücken, Germany, November 2000.

[19] P. Vlachos and M. Meyer. StaLib – data, software and news from the statistics community. <http://lib.stat.cmu.edu/>, 2002.

[20] M. O. Ward. XmdvTool: Integrating multiple methods for visualizing multivariate data. In *Proc. of IEEE Visualization ’94*, pages 326–336.

[21] R. Wegenkittl, H. Löffelmann, and E. Gröller. Visualizing the behavior of higher dimensional dynamical systems. In *Proc. of IEEE Visualization ’97*, pages 119–125.

[22] G. J. Wills. 524,288 ways to say “this is interesting”. In *Proc. of IEEE Symp. on Information Visualization*, pages 54–61, 1996.