

Parallel Coordinates for Multidimensional Data Visualization:

Basic Concepts

Julian Heinrich | Commonwealth Scientific and Industrial Research Organisation
Daniel Weiskopf | University of Stuttgart

Multidimensional data is regularly produced by simulations or measurements in virtually any scientific or engineering field. One example is the set of data attributes computed in computational fluid dynamics (CFD): such simulations typically include the velocity field of the fluid along with temperature, pressure, and other quantities of the flow. Other examples include amino acid attributes of protein structures in molecular biology or multidimensional state variables gathered from sensor networks.

These examples show that it's important to understand the distribution and relationship of data values—even if we ignore their connection to the original 2D or 3D simulation domain.

Standard visualization techniques for data attributes include histograms (for one data dimension) and scatterplots (to show two data dimensions simultaneously). But what if we want to show more than two data dimensions? Then we can split them, showing them side by side. We can look at histograms individually or put several histograms for different data dimensions next to each other. Or we can use a collection of scatterplots for varying pairs of data dimensions. We can even do this systematically, putting all combinations of two data dimensions in a scatterplot matrix.

But this approach has a serious weakness: it doesn't show the connection between more than two data dimensions. Here, parallel coordinates come into play because they aren't restricted to displaying just two data dimensions. In fact, they're scalable in the number of dimensions visualized—we just add another parallel axis for any additional data dimension.

The notion of parallel coordinates dates back to Maurice d'Ocagne in 1885.¹ Parallel coordinates as a means of

multidimensional data visualization were developed and popularized by Alfred Inselberg² and Edward Wegman.³ In this article, we describe the basic concepts of parallel coordinates as a tool for multidimensional data visualization. To this end, we discuss the geometric construction of parallel coordinates, issues related to the visualization of large data typical for scientific and engineering disciplines, and how we can interact with the visual representation of multidimensional datasets.

We'll describe details of implementations and applications in a follow-up article in this department. For background reading on parallel coordinates, we recommend Inselberg's book⁴ and our state-of-the-art report.⁵

Geometry of Parallel Coordinates

How do we construct a parallel-coordinates plot? Let's start with a simple case: just two data dimensions. Figure 1 shows how a 2D data point is transformed from the domain of the scatterplot to the corresponding line in the domain of the parallel coordinates. The scatterplot domain is spanned by the two orthogonal axes of the two data dimensions, whereas the data axes are parallel in the domain of the parallel coordinates. The data point is mapped to a line in the parallel-coordinates domain so that this line intersects the two parallel-coordinates axes at the respective data values for the two data dimensions. If there's more than just one data point, we just have to draw several lines in parallel coordinates.

Now we know how to map a point from the scatterplot to parallel coordinates. Can we also transform points from parallel coordinates back to the scatterplot? In fact, we can. Figure 1a illustrates how a collection of points coincide on

a line in the scatterplot. When we transform these points to parallel coordinates, they lead to a set of lines that intersect in a single point (Figure 1b). Therefore, we can relate this common intersection point in parallel coordinates to the line in the scatterplot—in other words, this point is transformed back from parallel coordinates to the scatterplot. There’s a correspondence between points and lines in both ways: the point-line duality. (For lines with a positive slope in the scatterplot domain, the corresponding pattern in parallel coordinates is another set of lines; see Inselberg’s book⁴ for details on how to establish the duality for this special case.)

So far, we’ve restricted parallel coordinates to two data dimensions. However, it’s simple to extend them to n dimensions. Figure 2 shows that we just put more and more parallel axes next to each other. Each pair of neighboring parallel axes is handled as we explained above; here, we clip the line that corresponds to a data point so that the line segment stays within the bounds of the two parallel coordinates axes. Next, we construct a straight line segment for each pair of neighboring axes. This leads to a polygonal line that intersects the parallel coordinates axes at the respective data values. In other words, an n -dimensional data point is mapped to a polygonal line in parallel coordinates.

Visual Patterns

We’ve seen how we can transform back and forth between scatterplots and parallel coordinates. How can we use the geometry of the parallel coordinates to visualize and analyze multidimensional data? How can we read and interpret a parallel-coordinates plot?

Let’s have a look at typical visual patterns that you may find in parallel coordinates. First, we restrict ourselves to patterns in 2D parallel coordinates; examples with more data dimensions will follow later. The most striking pattern is the accumulation point in parallel coordinates when we plot data with negative linear correlation. Perfect linear correlation leads to a single accumulation point, following the point-line duality (Figure 3a). However, even more realistic cases with just approximate linear correlation still lead to some smeared-out accumulation region (see Figure 3b). We just need to search for such regions with many line intersections to spot negative correlation.

In contrast, positive correlation leads to parallel or fan-shaped lines in parallel coordinates (Figure 3c). This is a less striking visual pattern and thus can be more difficult to perceive than negative correlation,⁶ but we can nevertheless use it to understand the data.

Nonlinear data behavior leads to other visual patterns. These are sometimes easier to spot by observing the envelope of lines, which can be an indicator for further dualities based on the point-line duality, such as curve-curve dualities.⁴ A prominent example is the ellipse-hyperbola duality (Figure 3d): we can search for hyperbola patterns in parallel

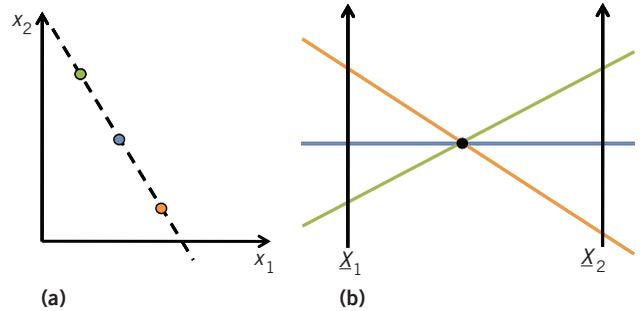


Figure 1. Geometry of parallel coordinates. (a) Data points in a scatterplot correspond to (b) lines in parallel coordinates: these lines intersect the two parallel axes of the parallel coordinates at the respective data values. Point-line duality is established as points coinciding on a line (dashed) in the scatterplot map to lines that intersect at a point (solid black dot) in parallel coordinates.

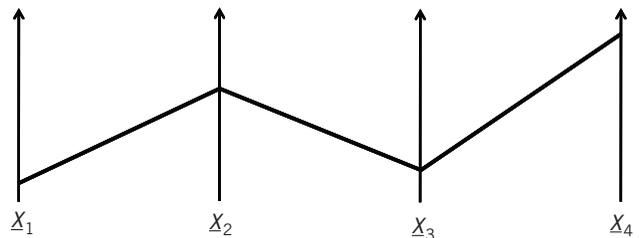


Figure 2. Parallel coordinates for multiple data dimensions and one data point, shown in the form of a polygonal line.

coordinates to find elliptical shapes in the scatterplot that might indicate a spherical or elliptical cluster of data points. Another pattern originates from data consisting of separate clusters: Figure 3e shows an example with two clearly separated clusters, visible in the scatterplot and the parallel-coordinates plot alike.

It’s clear that these visual patterns might not be obvious right from the beginning. One issue is that you have to learn how to read parallel coordinates. In fact, with increasing exposure to and experience with working with parallel coordinates, you will become more proficient in identifying relevant visual patterns.

Interacting with Parallel Coordinates

We’ve seen that negative correlation leads to very strong visual patterns in the form of accumulation points or regions. These patterns are less pronounced for positive correlation.⁶ By negating one parallel-coordinates axis, we can turn positive into negative correlation. Therefore, negating is a relevant interaction element. Similarly, the user should be supported in scaling data values to put them into the right perspective.

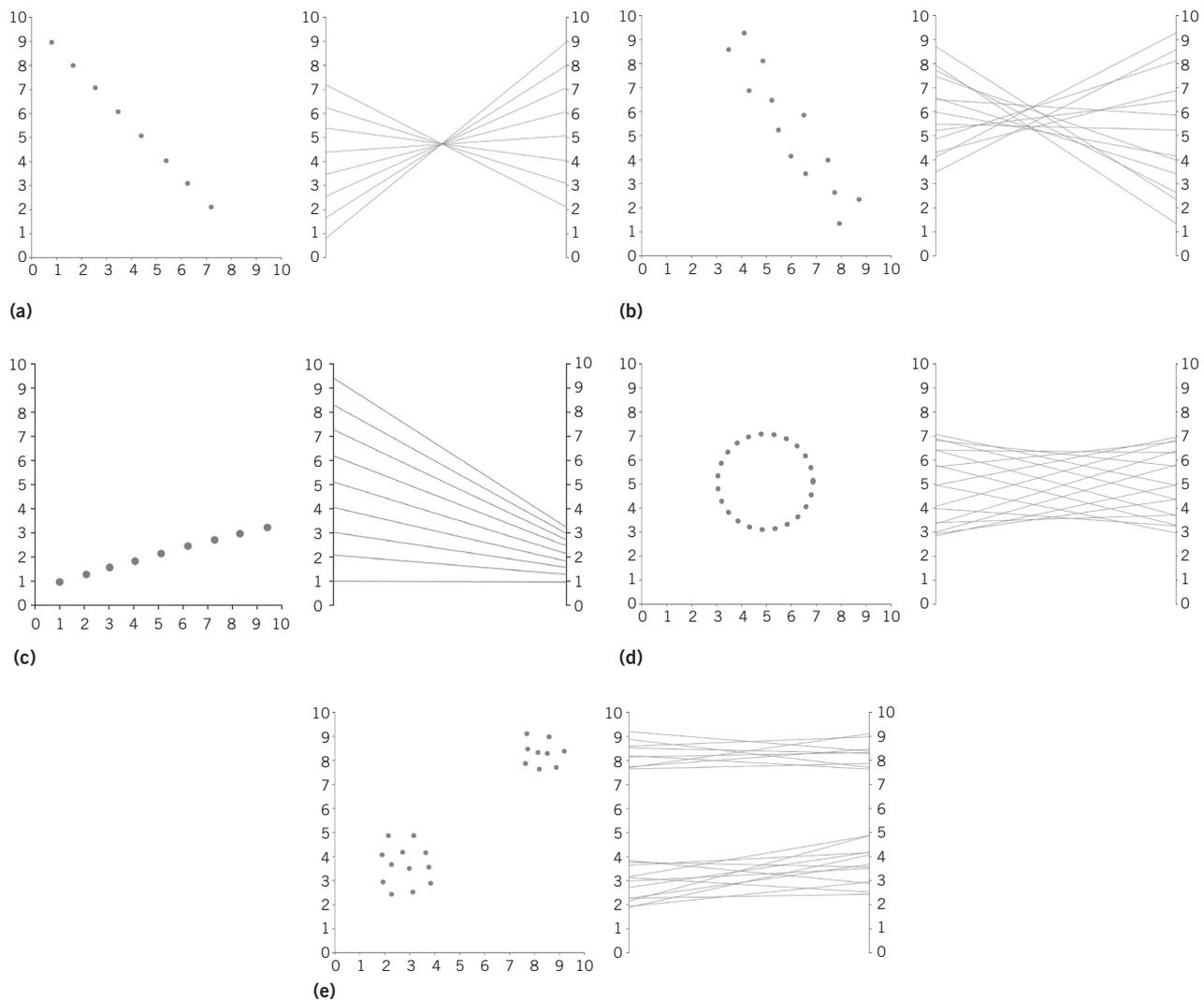


Figure 3. Visual patterns in scatterplots and parallel coordinates: (a) perfect negative correlation, (b) less pronounced negative correlation, (c) perfect positive correlation, (d) ellipse-hyperbola duality, and (e) two clusters.

Let's have a look at n -dimensional parallel coordinates. We want to follow the distribution of the data values across data dimensions, essentially reading along the polygonal lines in the plot. It's very useful to select and highlight subsets of the data in parallel coordinates, as in Figure 4, which features data collected by Edgar Anderson.⁷ With highlighting, we can easily follow sets of polylines across the data dimensions. The selection is typically done in an interactive fashion; often, the user marks interesting areas in the plot by mouse interaction. This selection process is also called *brushing*. Parallel coordinates come with interesting geometric properties, including the point-line duality. Therefore, we have several options of how we might want to brush data entries. One approach lets us brush data

values or ranges thereof on an axis in parallel coordinates: axis-aligned brushing (Figure 5a). An alternative method is angular brushing⁸: a range of angles is selected in parallel coordinates (Figure 5c). In this way, for example, we can highlight positive slopes (correlations) in the data. Yet another approach supports polygon-shaped brushes in the parallel-coordinates domain, corresponding to rotated boxes in the scatterplot domain (Figure 5b).

The full power of brushing emerges in combination with other visualization views, for example, 3D renderings of simulation results. The sidebar "Coordinated Multiple Views" provides more background information.

Even with brushing, it's clear that the visual patterns strongly depend on the order of the coordinate axes.

Correlations between two data dimensions are best shown with a pair of neighboring parallel-coordinates axes. What's the best order of axes? Some approaches use automatic data analysis to identify the most relevant pairs of axes and put them next to each other.⁹ Other approaches allow the viewer to change the axis order interactively or provide a set of individual plots to explore all possible pairwise correlations.¹⁰ Even with some (initial) automatic order, any useful parallel-coordinates visualization tool will support the user in modifying the axis order, to interactively explore the data.

Density Plots

We've been assuming that the actual visual representation of the parallel-coordinates plot is simple: just draw a polygonal line for each data point. More specifically, let's just draw opaque lines. We produced the examples shown so far by using this rendering approach. However, this way of rendering doesn't work for large datasets—large in the sense of many data points. If it is, then the parallel-coordinates plot will suffer from much overplotting and visual clutter. Unfortunately, the issue of large data is natural to most applications in science and engineering because simulations and measurements tend to deliver large numbers of data points.

One popular approach to large-data parallel coordinates replaces the opaque lines with semitransparent ones. In this way, we can see a varying density of lines: high density in regions of many overlapping lines, low density in areas covered by few lines. Such a density plot allows us to easily recognize the prominent visual patterns that belong to the most frequent characteristics of the data, as in Figure 6.

Density plots can be constructed by rendering lines with additive blending: we virtually count the number of times a pixel is covered by a polygonal line from the parallel-coordinates plots. Additive blending is sometimes replaced by alpha blending; this blending model isn't order-independent, that is, it allows us to emphasize the lines that are rendered late. This only occurs if lines have varying density or color. Another alternative uses the model of continuous parallel coordinates¹¹: it directly transforms densities from the (spatial) computational domain—such as the simulation grid—to densities in parallel coordinates.

We think that density plots are highly relevant and useful for the typical dataset sizes in science and engineering.

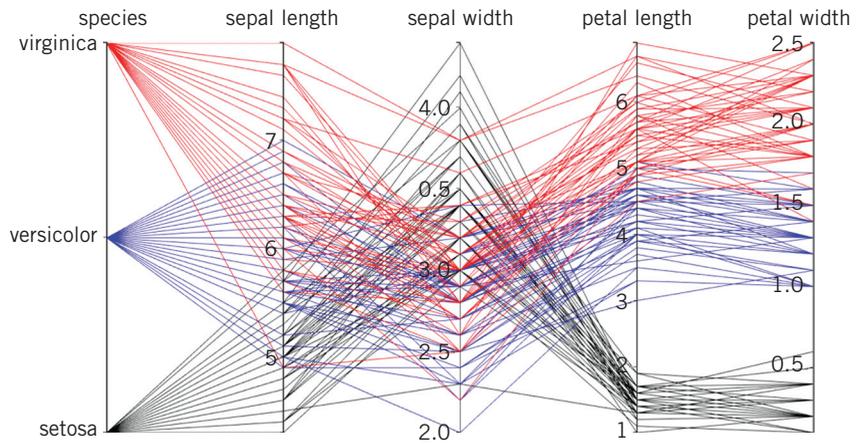


Figure 4. Brushing lets us compare the values of data samples or clusters over a set of dimensions. For this dataset (a popular test dataset of iris flower data collected by Edgar Anderson⁷), a non-metric variable was mapped to a number to represent it in parallel coordinates (leftmost axis). This is common practice, as it allows us to select groups of data samples. We highlighted three different groups with separate colors.

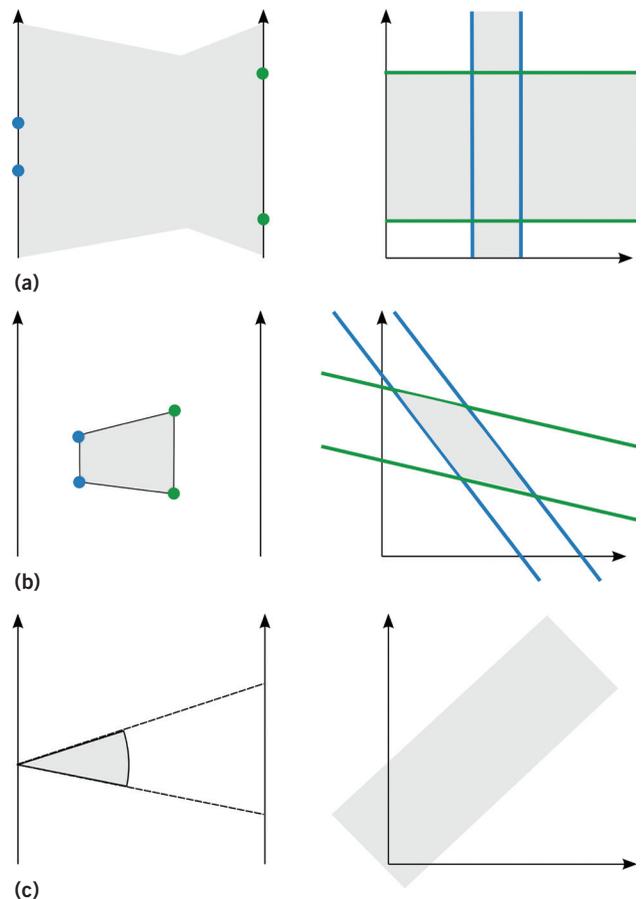


Figure 5. Different types of brushing in parallel coordinates (shown left) and the corresponding selection in the scatterplot (right): (a) axis-aligned brushing, (b) polygon-shaped brush, and (c) angular brushing.

Coordinated Multiple Views

Coordinated multiple views are a popular approach to designing visual interfaces that facilitate the visual analysis of complex data. The basic idea is that complex data can't be shown in a single visualization alone because there's just too much information to be conveyed in a single image. Instead, we can show different types of visualizations in multiple views. The multiple views can provide different perspectives on the very same data. But how can we build a combined, overall mental model of the data from several views? One strategy uses the concept of coordination: the views aren't completely independent from each other, but rather react to user interaction in a coordinated sense.

Brushing-and-linking is a popular approach to coordinated multiple views: visualization elements are brushed (that is, selected) in one view, and the selection is automatically applied to the other views, highlighting the corresponding elements in these views. Early work used brushing-and-linking to coordinate multiple scatterplots.¹ Brushing is commonly used in scatterplots; it can be extended to more complex types of brushes, for example, to high-dimensional brushes.²

It's even more interesting to link structurally different views: data in science and engineering often has some intrinsic spatial relationship, and therefore it's especially useful to link different views that complement the perspective of the spatial embedding

with the distribution of data values. Figure A illustrates this scenario.³ Another similar example is brushing-and-linking between scatterplots of data attributes from a simulation and the 3D visualization of the simulation. Helmut Doleisch and his colleagues⁴ provide several examples for such coupling. For further reading on coordinated multiple views, we recommend the survey by Jonathan Roberts.⁵

References

1. R.A. Becker and W.S. Cleveland, "Brushing Scatterplots," *Technometrics*, vol. 29, no. 2, 1987, pp. 127–142.
2. A.R. Martin and M.O. Ward, "High Dimensional Brushing for Interactive Exploration of Multivariate Data," *Proc. IEEE Visualization*, 1995, pp. 271–278.
3. J. Heinrich et al., "Visualising Intrinsic Disorder and Conformational Variation in Protein Ensembles," *Faraday Discussions*, vol. 169, no. 1, 2014, pp. 179–193.
4. H. Doleisch, M. Gasser, and H. Hauser, "Interactive Feature Specification for Focus+Context Visualization of Complex Simulation Data," *Proc. Joint Eurographics/IEEE TCVG Symp. Visualization*, 2003, pp. 239–248.
5. J.C. Roberts, "State of the Art: Coordinated & Multiple Views in Exploratory Visualization," *Proc. Int'l Conf. Coordinated & Multiple Views in Exploratory Visualization*, 2007, pp. 61–71.

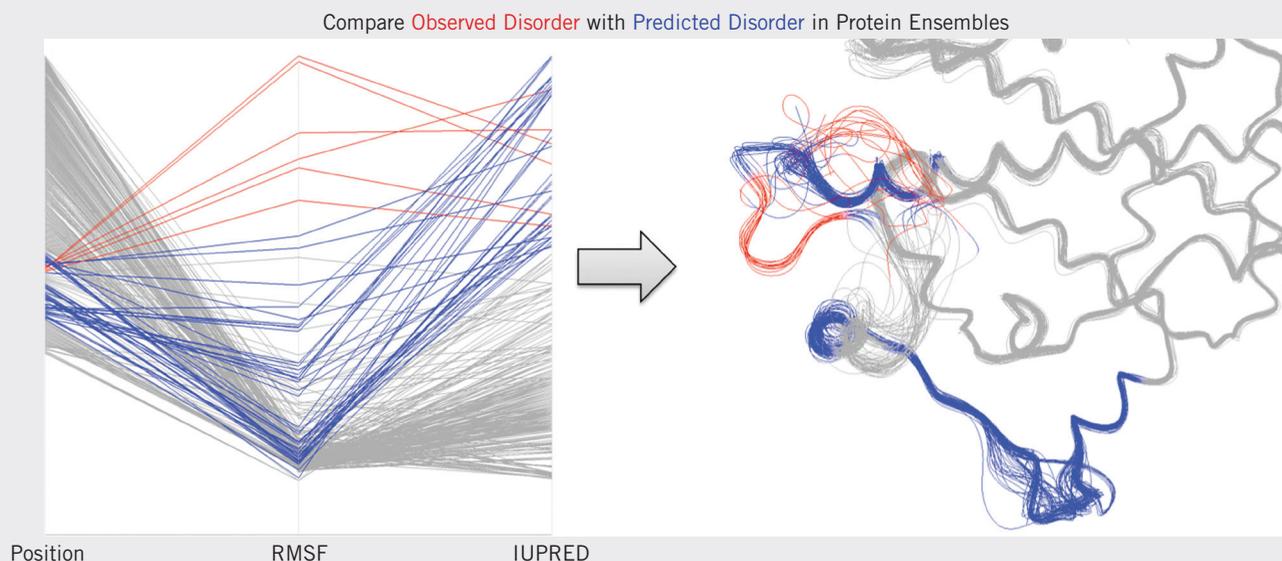


Figure A. Brushing-and-linking using parallel coordinates to highlight amino acids with properties as selected in parallel coordinates (left). The red brush indicates residues of protein structures that show a significant variation in their spatial position, corresponding to high values of the root mean-square fluctuation (RMSF). These residues are called "disordered." Only with linking to the 3D visualization (right) can we understand the spatial relationship of the disordered parts. From the parallel-coordinates plot, we can see that not all residues that are predicted to be disordered (blue brush, IUPRED score > 0.5) show a strong spatial variation (red brush). More details appear elsewhere, in the source for the image.³

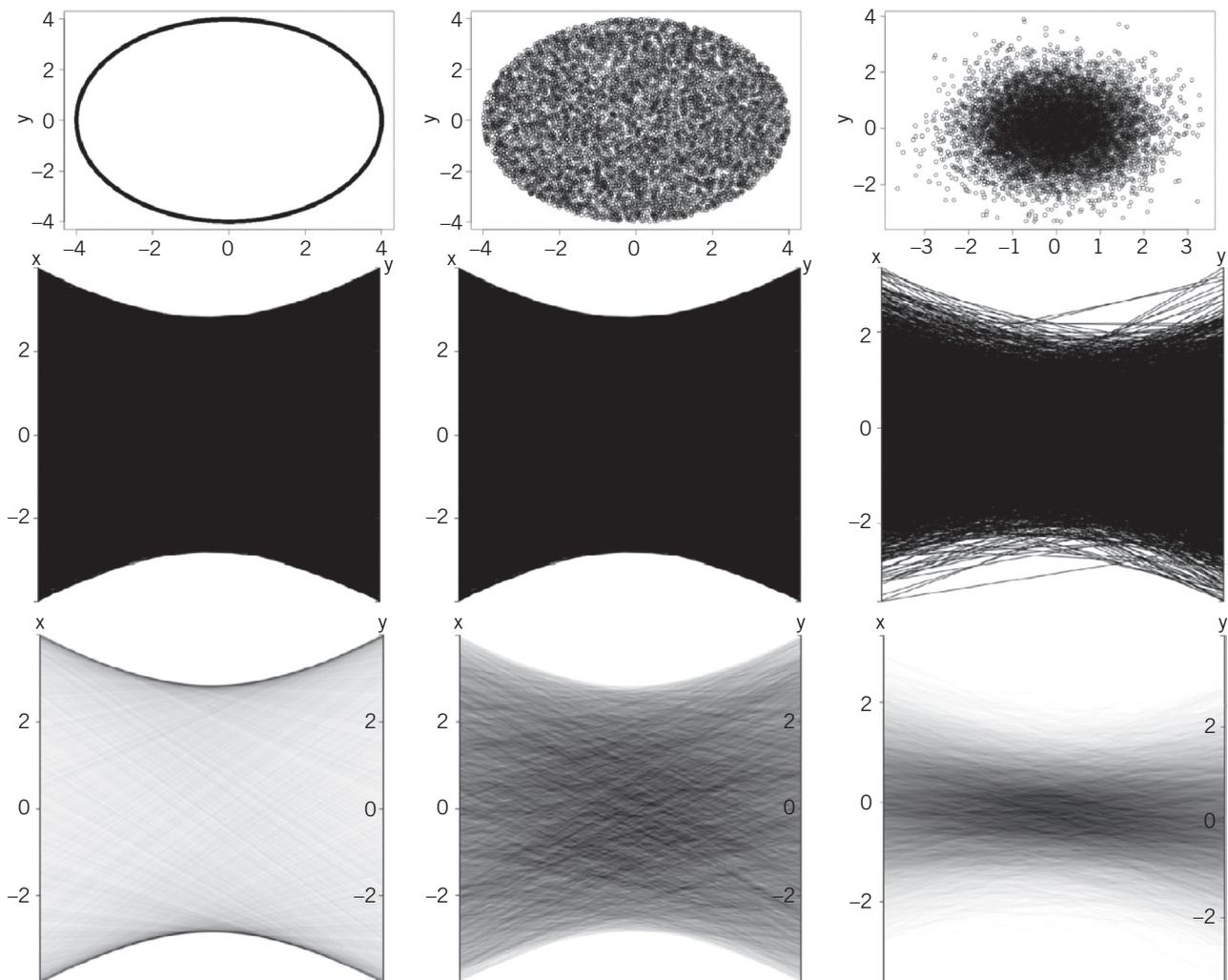


Figure 6. Density plots reveal otherwise hidden patterns in parallel coordinates. The top row shows scatterplots, the center row parallel coordinates with opaque lines, and the bottom row density plots of parallel coordinates. The patterns that emerge in parallel coordinates from samples taken from the circumference of a circle (left column) and from the circle area (center column) are identical for opaque line rendering (center row) but become distinguishable with density information (bottom row). The density plot also provides much more information of the distribution of lines than the opaque line rendering for data distributed with varying density (right column).

However, we would like to point out alternative strategies, such as reduction methods that work on the original data before it's visualized (for example, clustering, principal component analysis, and filtering). More details are in our report.⁵

Parallel coordinates are a great tool for visualizing multidimensional data. With the straightforward mapping from data points (in scatterplots) to lines (in parallel coordinates), the concept and basic implementation of parallel coordinates are simple. So, why aren't plots with parallel

coordinates as common as scatterplots? We think that one important reason is that the visual patterns are very different from those in scatterplots and require training before they become useful. A related issue is that some visual patterns are quite hard to perceive—especially the visual asymmetry between visualizing positive and negative correlation is striking and makes some aspects of the data less clear than with scatterplots. We also think that the need for interaction—in particular, for complex data—might add some problems with the acceptance. However, we're convinced that parallel coordinates provide a very useful, complementary view on multidimensional data. This is particularly true

for data that comes with relationships between more than two data dimensions.

You can find more background information and material for further reading on Inselberg's webpage (www.math.tau.ac.il/~aaisreal) and www.parallelcoordinates.de. We'll publish a follow-up article in this department with more details of implementations and practical applications of parallel coordinates. Stay tuned! ■

Acknowledgments

The dataset used in Figure 4 shows data collected by Anderson;⁷ we used the data distribution made available through the R Project (www.r-project.org).

References

1. M. d'Ocagne, *Coordonnées parallèles et axiales: Méthode de transformation géométrique et procédé nouveau de calcul graphique déduits de la considération des coordonnées parallèles*, Gauthier-Villars, 1885.
2. A. Inselberg, "The Plane with Parallel Coordinates," *Visual Computer*, vol. 1, no. 4, 1985, pp. 69–91.
3. E.J. Wegman, "Hyperdimensional Data Analysis Using Parallel Coordinates," *J. Am. Statistical Assoc.*, vol. 85, no. 411, 1990, pp. 664–675.
4. A. Inselberg, *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*, Springer, 2009.
5. J. Heinrich and D. Weiskopf, "State of the Art of Parallel Coordinates," *Eurographics 2013, State of the Art Reports*, 2013, pp. 95–116.
6. J. Li, J.-B. Martens, and J.J. van Wijk, "Judging Correlation from Scatterplots and Parallel Coordinate Plots," *Information Visualization*, vol. 9, no. 1, 2010, pp. 13–30.
7. E. Anderson, "The Irises of the Gaspe Peninsula," *Bulletin Am. Iris Soc.*, vol. 59, 1935, pp. 2–5.
8. H. Hauser, F. Ledermann, and H. Doleisch, "Angular Brushing of Extended Parallel Coordinates," *Proc. IEEE Symp. Information Visualization*, 2002, pp. 127–130.
9. A. Tatu et al., "Automated Analytical Methods to Support Visual Exploration of High-Dimensional Data," *IEEE Trans. Visualization and Computer Graphics*, vol. 17, no. 5, 2011, pp. 584–597.
10. J. Heinrich, J. Stasko, and D. Weiskopf, "The Parallel Coordinates Matrix," *EuroVis—Short Papers*, 2012, pp. 37–41.
11. J. Heinrich and D. Weiskopf, "Continuous Parallel Coordinates," *IEEE Trans. Visualization and Computer Graphics*, vol. 15, no. 6, 2009, pp. 1531–1538.

Julian Heinrich is a postdoctoral fellow at Digital Productivity, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia. His research interests include visualization, visual analytics, and bioinformatics. Heinrich has a PhD in computer science from the University of Stuttgart, Germany. Contact him at julian.heinrich@csiro.au.

Daniel Weiskopf is a professor at the University of Stuttgart, Germany. His research interests include visualization, visual analytics, GPU methods, computer graphics, and special and general relativity. Weiskopf has a PhD in physics from the University of Tübingen, Germany. He's a member of the IEEE Computer Society, ACM, Eurographics, and the Gesellschaft für Informatik. Contact him at weiskopf@visus.uni-stuttgart.de.



Keep up with the latest IEEE Computer Society publications and activities wherever you are.



@ComputerSociety
@ComputingNow

facebook

facebook.com/IEEEComputerSociety
facebook.com/ComputingNow

LinkedIn

IEEE Computer Society
Computing Now

YouTube

youtube.com/ieeecompulersociety

cn Selected articles and columns from IEEE Computer Society publications are also available for free at <http://ComputingNow.computer.org>.