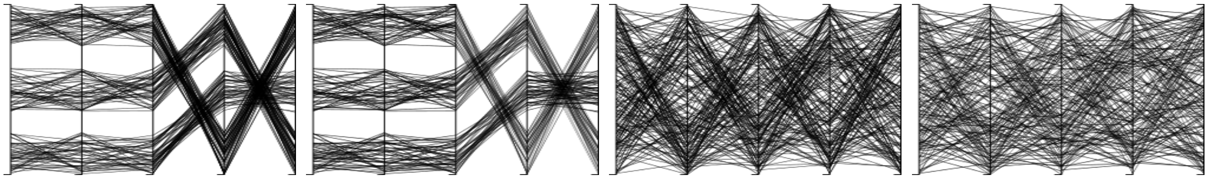
Slope-Dependent Rendering of Parallel Coordinates to Reduce Density Distortion and Ghost Clusters

David Pomerenke, Frederik L. Dennig, Daniel A. Keim, Johannes Fuchs, and Michael Blumenschein
University of Konstanz, Germany
firstname.lastname@uni-konstanz.de



(a) Regular rendering (b) Slope-dependent rendering (c) Regular rendering (d) Slope-dependent rendering

Figure 1: **Comparison of regular parallel coordinates with our slope-dependent polyline rendering.** Parallel coordinates face two problems, which are inherent in the technique: (a) depicts three clusters of the same diameter and size across all dimensions. *Diagonal changes of the clusters are visually more prominent*, as diagonal lines are rendered more closely. (c) shows 200 data points of uniform random clutter/noise in all dimensions. *Zig-zag clusters are visible* as diagonal lines and are perceived as clusters, although there are no such clusters in the data (ghost clusters). We propose to render each line segment based on its slope between two axes. As a result, clusters are not distorted by their shape (b), and the ghost clusters effect is reduced (d).

Abstract

Parallel coordinates are a popular technique to visualize multi-dimensional data. However, they face a significant problem influencing the perception and interpretation of patterns. The distance between two parallel lines differs based on their slope. Vertical lines are rendered longer and closer to each other than horizontal lines. This problem is inherent in the technique and has two main consequences: (1) clusters which have a steep slope between two axes are visually more prominent than horizontal clusters. (2) Noise and clutter can be perceived as clusters, as a few parallel vertical lines visually emerge as a ghost cluster. Our paper makes two contributions: First, we formalize the problem and show its impact. Second, we present a novel technique to reduce the effects by rendering the polylines of the parallel coordinates based on their slope: horizontal lines are rendered with the default width, lines with a steep slope with a thinner line. Our technique avoids density distortions of clusters, can be computed in linear time, and can be added on top of most parallel coordinate variations. To demonstrate the usefulness, we show examples and compare them to the classical rendering.

Keywords: Parallel coordinates, rendering, density distortion.

1 Introduction

Parallel coordinates plots (PCPs) [11] are a well-researched visualization technique for multi-dimensional data. Studies have shown that they can be learned easily by non-visualization experts [15, 16] and used in practice in various domains like finance [1], traffic safety [7], and network analysis [17]. Compared to related techniques such as scatter plot matrices and projections, PCPs have the advantage to identify, explore, and understand patterns across multiple dimensions. Cluster identification is, among others, one of the most common tasks for parallel coordinates [2].

Every data record is represented by a single polyline, spanning across the different axes/dimensions of the dataset. Polylines running close together are considered a cluster as they have similar values across the dimensions. In Fig. 1 (a), we can see three clusters spanning across the dataset. Between dimensions 1–3, the clusters are *horizontal*, meaning that the data values are approximately the same within all dimensions. Across dimensions 3–5, the clusters are *diagonal*, changing their values and cluster center, and have a steep *slope*. We can easily see a general problem of the PCP technique: diagonal changes of clusters are visually more prominent than horizontal trends. Assuming all polylines have the same line thickness, there are two reasons for the problem: Diagonal lines need more area and pixels, and the actual space between parallel lines is smaller for diagonal clusters compared to horizontal ones. As a consequence, there is a density distortion of clusters based on the slope or angle of the cluster. A second problem, also based on these rendering artifacts, are so-called *ghost clusters*. Fig. 1 (c) depicts a dataset with 200 points, randomly and uniformly distributed across all dimensions. One can "see" two zig-zag patterns indicating two clusters. However, the data does not contain any specific structure – in particular, no clusters. This problem is not only relevant in pure clutter (or noise) datasets but also influences the perception of clusters in datasets that contain a limited amount of clutter and noise along with relevant patterns. *Ghost clusters* and distorted cluster density are related to human bias, but the core problem is based in the PCP technique. It can also occur in other variants of PCPs (e.g., different colors and transparency for lines, and edge-bundling).

We make two contributions: (1) we formalize the problem and show its impact. (2) we propose a novel approach which renders each line segment based on the slope between two dimensions. Horizontal lines are rendered with the default line thickness. Diagonal lines are rendered thinner. Two examples are depicted in Fig. 1 (b) and (d). The technique can be computed in linear time and applied on top of most PCP variations. The approach by Zhou et al. [20] is closest to our work. It blends polylines based on their local neighborhood, which reduces the influence of noise but still suffers from the distortions caused by the over-emphasis on diagonal lines.

All material of this paper is available at https://osf.io/sy3dv.

2 Related Work and Research Gap

Plenty of approaches try to reduce clutter and highlight patterns in PCP generally. However, to the best of our knowledge, a formalization of the pattern distortion based on the polyline slope is missing, and none of the existing approaches specifically target this limitation.

2.1 Sampling and Filtering Techniques

The basic premise for the use of sampling (and filtering) techniques is that with less data, the degree of clutter and overplotting decreases, while the general structures, typically represented by many data records, remain in the PCP [10]. The taxonomy by Ellis & Dix [6] provides a categorization of clutter reduction methods, including sampling, filtering, and clustering, as well as visual techniques such as point size and opacity. Sampling often removes relevant data records or dimensions, and in this way reduces the truthfulness of the sampling concerning the dataset in its entirety. Our technique reduces clutter by counterbalancing the distortion artifact inherent to PCPs. It can be applied to a sampled or filtered subset of the data if the dataset exceeds the size visualizable in a PCP. Dependent on the data, our technique increases the amount of data displayable in a given PCP by deemphasizing diagonal polyline segments.

2.2 Axes Reordering and Dimension Reduction

Another approach to minimize clutter in PCPs is to reorder the dimension axes or reduce the number of displayed dimensions. For example, Pargnostics by Dasgupta and Kosara [5] describes a set of quality metrics for PCPs which can be minimized or maximized (e.g., the number of line-crossings and parallelism). The authors also suggest the flipping of axes to reduce the number of line-crossings or diagonal clusters. The survey by Behrisch et al. [4] discusses a large number of quality metrics as objective functions for axes reordering. Axes reordering, dimension reduction, and axes flipping can reduce ghost clusters by favoring horizontal structures. Depending on the data, however, it cannot be avoided entirely. Axes reordering is highly dependent on the data and analysis task. It is an orthogonal concept to our approach and can be combined with it.

2.3 Density- and Cluster-based Rendering

Clusters and other patterns can also be highlighted by density-distributed rendering. The general idea is to render PCPs as density distributions rather than individual polylines. Johansson et al. [12] measure the density based on the number of overlapping polylines per pixel. This notion of density serves as input to a transfer function that allows highlighting areas according to their local density. Heinrich & Weiskopf [9] apply the concept of continuous scatterplots [3] to PCPs to derive a density model and thus interpolate the data. The resulting rendering is specifically useful for cluster identification. The work by Palmas et al. [14] provides a different approach, which bundles edges according to class membership. The resulting bundles are rendered as polygonal strips. Density- and cluster-based rendering may hide the underlying individual records and often require class labels to achieve a useful coloring or edge-bundling. While these approaches reduce clutter, they do not avoid the density distortion of clusters.

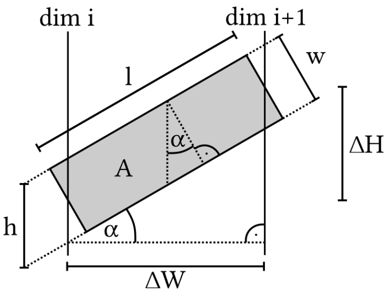


Figure 2: **Geometry of a polyline segment.** In regular PCPs, width w is constant and length l, height h and area A are dependent on α.

2.4 Polyline Modifications

A common technique is to modify the polylines of PCPs, specifically the overall line width, opacity, color, and shape. One example is the edge-bundling approach by Heinrich et al. [8], which bundles polylines according to class membership and thus reshapes the line. The work by Zhou et al. [20] called line splatting is most closely to ours. Line splatting is iteratively adjusting the opacity of lines based on the local neighborhood. Users can interactively change the degree of polyline and segment splatting. In contrast to Zhou et al. [20], our work tries to mitigate the visual distortions intrinsic to PCPs, such as the perceived density of clusters and the effect of ghost clusters.

3 Problem Statement and Impact on PCP Patterns

We formalize the line geometry of parallel coordinates and describe their effects on density distortions and ghost clusters.

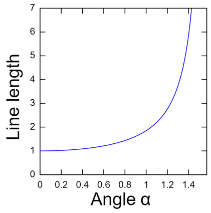
3.1 Line Geometry in Parallel Coordinates

In standard PCPs, a polyline segment has a constant line width $w \in \mathbb{R}^+$, also called thickness or stroke width. As depicted in Fig. 2, the slope of a segment is defined by the angle $\alpha \in [0, \frac{\pi}{2})$ between the horizon and the segment. ΔW denotes the space between the dimension axes and ΔH indicates the difference of data values. In contrast to w, the line height is slope-dependent: $h = w \cdot \cos^{-1}(\alpha)$, with $\alpha = \tan^{-1}(\Delta H/\Delta W)$. The area A of a segment is defined as $h \cdot \Delta W$ and the length l is defined as $\Delta W \cdot cos^{-1}(\alpha)$.

3.2 Slope-dependent Distortion of Polylines

In parallel coordinates, *horizontal clusters* correspond to a set of data points with a strong positive correlation in a subset of values across dimensions. Visually, these clusters have roughly horizontal cluster boundaries and only small line slopes. An example is depicted in dimensions 1–3 of Fig. 1 (a). In contrast, *diagonal clusters* correspond to data points with similar values within, but a strong variation between dimensions. Visually, these clusters have steep cluster boundaries and high line slopes. The last three dimensions in Fig. 1 (a) present examples. Horizontal and diagonal clusters are not defined in a precise way, and there is a smooth transition between them. *Visual cluster density* refers to the share of colored pixels within a line cluster. A large number of densely packed colored pixels induce a dense cluster and vice versa. The following effects characterize the emerging distortions influencing visual cluster density.

Increase of Line Length and Area. Line length l, line-height h and line surface area A depend on the angle α, with the exponential relationship $A \propto h \propto l \propto cos^{-1}(\alpha)$ shown by the figure on the right. This dependency affects the perception of clusters. Large line slopes imply larger surface areas (= more pixels, lower data-to-ink ratio [18])



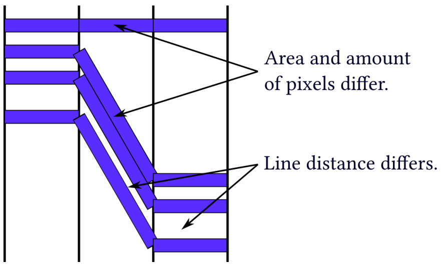


Figure 3: **Effect of angle α on PCP lines.** (1) Diagonal lines have a *higher line surface area* (= more pixels) compared to horizontal lines. (2) Diagonal lines have a *smaller distance between lines.*

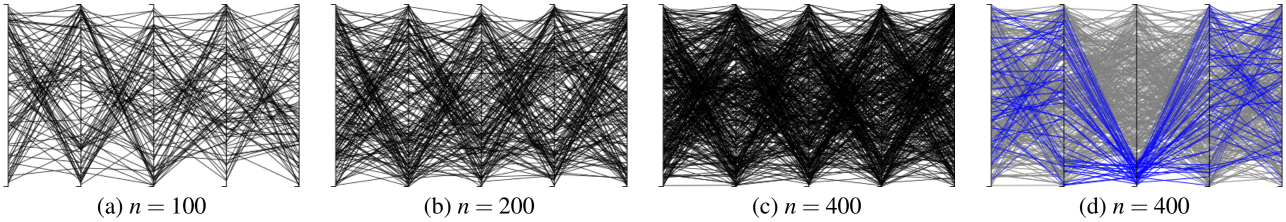

(a) $n = 100$ (b) $n = 200$ (c) $n = 400$ (d) $n = 400$

Figure 4: **Ghost clusters in uniformly distributed random data points.** The number n of polylines is increased from (a) to (c). (d) = (c) but the data points of a ghost cluster are highlighted to demonstrate that they are indeed uniformly distributed even though (c) indicates otherwise.

and therefore a more prominent line. The emphasis translates from lines to clusters, so that diagonal clusters are more noticeable than horizontal clusters. This effect is depicted in the top of Fig. 3.

Decrease of Line Distance. Large line slopes in diagonal clusters reduce the space between lines and increases the perceived density of the cluster as lines may overlap, and the background vanishes. The orthogonal distance $d_\perp$ between two parallel lines is depends on the angle α, with $d_\perp = d_h \cdot \cos(\alpha)$, where d_h is the distance of the intersections of both lines with a dimension axis. This effect creates the perception that the lines are cohesive as shown in Fig. 3.

3.3 Visual Distortion of Cluster Densities

The Gestalt law of proximity [13, 19] indicates that the density of lines translates to a perception of cohesiveness and thereby enables users to recognize clusters in PCPs. Classical PCPs put undue emphasis on diagonal clusters, which is facilitated by the *increase of line lengths* and *decrease of line distances*. This contradicts the data-ink ratio coined by Tufte [18], which describes the proportion of ink devoted to the actual data relative to the total amount of ink. Thus, it adds unnecessary distortion: Diagonal clusters are emphasized more than horizontal clusters. Classical PCPs, therefore, induce a systematically inaccurate perception of clusters, when the observer would expect that the visualization is inherently neutral in this respect. We can see the effect in Fig. 1 (a), where diagonal and horizontal clusters receive a significantly different emphasis.

3.4 Ghost Clusters

The rendering effects caused by the different slopes of the polyline segments can also produce artificial patterns in parallel coordinates plots. Fig. 4 (a–c) show three PCPs with uniformly distributed random data points, i.e., there is no structure in the data. One can easily see that a *zig-zag pattern*, alternating between high and low values is visually present. The corresponding polylines seem to be parallel and close together, forming two clusters. With an increasing number of data points, the "clusters" are perceptually stronger. In Fig. 4 (d), we mark one apparent cluster and highlight its polylines across the different dimensions. One can see that the data is indeed randomly distributed and not forming a cluster across the dimensions. We define these visible, but non-existing patterns as *ghost clusters*. Ghost clusters are not only a problem of datasets with clutter or noise. Also, in structured datasets, ghost clusters can be present and influence the interpretation of the data.

4 Slope-Dependent Rendering of Lines

To overcome the distortion of cluster densities and potential ghost clusters, we propose to render the polyline segments based on their angle α. The general idea is to render horizontal lines with the default width and diagonal lines with a thinner line. As a result, we increase the space between vertical lines and decrease the surface area, i.e., the number of pixels to draw a line. In the ideal case, all line segments should end up with the same area and the same distance between the segments. To achieve the same area for all line segments, the width w of the polyline segments needs to be scaled based on their length l. As the line length $l = \Delta W/cos(\alpha)$ is dependent on α, the desired width ω also needs to depend on α. We interpret all lines as parallelograms with an equal and constant area A and thus equal and constant side length $h \in \mathbb{R}^+$ which is *independent* of α (Fig. 2). The height of this parallelogram corresponds to the desired α-dependent width ω, leading to $A = l \cdot \omega = \Delta W/cos(\alpha) \cdot (h \cdot cos(\alpha))$. This results in the angle-dependent line width

$$\omega = h \cdot \cos(\alpha) \tag{1}$$

The angle-dependent width ω can be generalized, allowing us to weaken or strengthen the adjustment of the line width

$$\omega = h \cdot \cos^P(\alpha) \tag{2}$$

where parameter $P \in \mathbb{R}$ determines the adjustment strength. Our approach applies to pixel- and vector-based rendering techniques.

4.1 Choosing the Adjustment Strength

$P = 0$ corresponds to classical PCP rendering, where all lines have the same width. $P = 1$ corresponds to rendering with equal line heights resulting in the same surface area A for all polylines. However, it does not fully correct the decreased line distances. Thus, we allow $P > 1$ as over-adjustment to further compensate overplotting of lines with strong slopes. In particular, the parameter P can be freely adapted to the degree of clutter, and the properties of the dataset. We want to highlight that our slope-dependent rendering can fully overcome the problem of different line surface area ($P = 1$), but the issue of varying distance between polylines can only be reduced with $P > 1$. Based on these geometric properties, we recommend $P = 1$ for truthful representation. However, many properties of a PCP and dataset influence the quality of the rendering (see Sec. 4.2), therefore an over-adjustment ($P > 1$) may be necessary. Our tests with various synthetic and real-world datasets showed that $P \approx 2$ is an upper bound for most applications.

In Fig. 5, we apply our technique to a synthetic dataset and uniform random noise. We achieve a balanced emphasis of horizontal and diagonal clusters for $P = 1$ and an over-emphasis of horizontal lines for $P = 2$. Ghost clusters are also reduced for $P = 1$ because their density is corrected. However, the effect of smaller line distance cannot be avoided, and ghost clusters are still visible. We can compensate for the line distance effect by over-adjusting the line area effect (e.g., $P = 2$), nearly eliminating the ghost clusters, but introducing an over-emphasis of horizontal lines.

4.2 Influence of PCP Properties and Parameters

The following parallel coordinates parameters influence the impact of ghost clusters and the distortion of cluster densities and should be taken into account when applying the slope-dependent rendering.

PCP Size, Axis Height and Spacing. The overall size of a PCP has a direct impact on the axis height and spacing ΔW between the axes. Axis height and ΔW determine the range of α: Long axes and tight spacing, caused by high-dimensionality, increase the angles and distort cluster densities and increase the likelihood of ghost clusters.

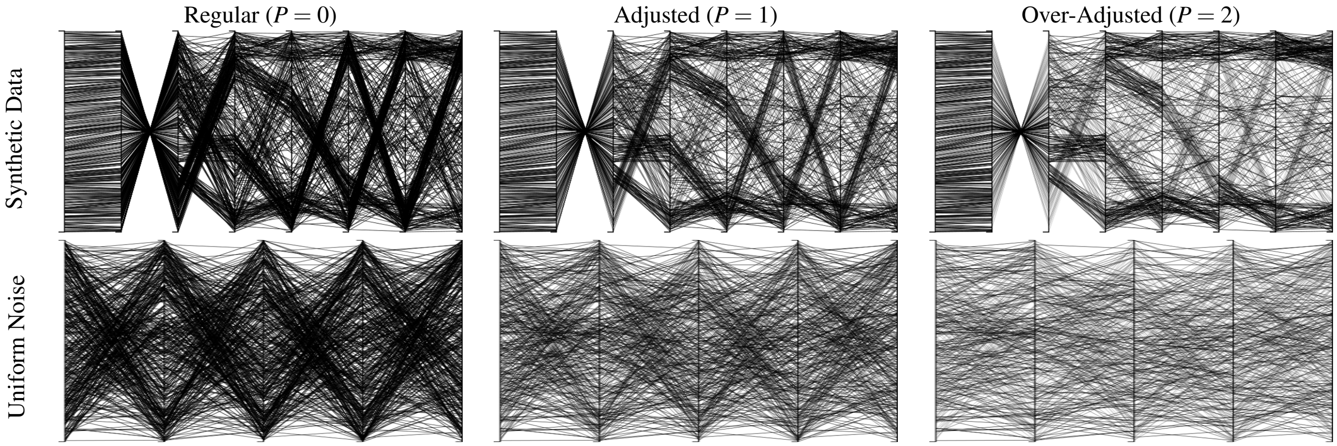
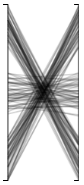


Figure 5: **Effect of parameter P on pattern visualization in synthetic data with uniformly distributed background noise, and in uniformly distributed random data only.** Regular rendering ($P = 0$) significantly over-emphasizes diagonal clusters and causes the occurrence of ghost clusters. For $P = 1$, all clusters are equally emphasized, and the effect of ghost clusters is strongly mitigated. For $P = 2$ the distortion is reverted, and horizontal clusters are over-emphasized. Simultaneously, ghost clusters are further reduced.

Default Line Width. Manipulating the constant line-height h influences the detail and the clarity of the PCP. Thick lines increase the problem of overplotting, in particular for diagonal lines and clusters. Thin lines are more distinguishable and therefore produce more salient visualizations. The result of the slope-dependent rendering depends on the default line width, typically determined by the user. The default width directly influences the area covered by each line segment. It is advisable to consider a manual adaptation of the constant line-height h before applying a slope-dependent rendering.

Data Volume. The number of data records influences the visual representation a PCP and is strongly related to its size and the default line width. A high data volume visualized with a small PCP and/or a thick line width increases the problem of overplotting, but also the distortion of cluster densities and ghost clusters. For example, Fig. 4 shows how the dataset size increases the perception of ghost clusters. Therefore, these properties should be optimized for a given dataset before applying the slope-dependent rendering.

Line Color and Transparency. When no transparency is used, then the color of the polylines does not affect PCPs and therefore also not our approach. Transparency can be used to avoid clutter and overplotting but introduces another artifact, which negatively influences the perception of patterns. Crossing lines introduce a darker color, which may be interpreted as a cluster. Combined with the slope-dependent rendering, new ghost clusters may occur, while other patterns may vanish: Adjusting the transparency of lines based on their slopes, as opposed to the line width, is not useful.



5 Discussion and Future Work

To test the effectiveness of our slope-dependent rendering, we implemented a tool which is available on our website[1]. Users can upload their data, or try out various synthetic and real-world datasets, comparing the results of classical and slope-based rendering. During our testing with the implementation, we found out that our slope-dependent line adjustment technique performs well on various datasets, reduces ghost clusters, and counterbalances distortions. We also tested the impact of our approach with other patterns, such as positive and negative correlations (Fig. 5). While positive correlations are not affected even with a large P value ($P = 2$), the slope-dependent rendering influences the diagonal lines of negative correlation. We found that negative correlations also remain visible. However, the line representing data points at the ends of the dimension ranges are drawn with a small line width, making the visibility of this pattern susceptible to large P values ($P = 2$).

Our approach can be combined with other techniques, such as axes reordering and dimension reduction, as they do not manipulate the polylines of a PCP. It can also be combined with polyline modifications like edge-bundling. However, the line width should then be calculated relative to the line length rather than the slope. As described above, various PCP properties generally influence the visual distortion and ghost clusters in PCPs. To achieve optimal results, these parameters should be optimized before the slope-dependent rendering is applied, and focus on the reduction of overplotting and the average angles of polylines.

A careful selection of the parameter P is necessary. The usefulness of a particular P depends on many general PCP properties, as well as data characteristics such as the number of data records and dimensions. Therefore, P cannot be determined fully automatically based on a fixed parameter. However, we envision an algorithm which measures the density distribution, overlapping, and distortion and automatically selects an appropriate P to achieve a reliable representation of the data. We want to address this algorithm as part of future work. Furthermore, we want to evaluate the usefulness of our approach, in particular in comparison to other methods, by conducting a quantitative user study.

6 Conclusion

We formalize two general problems of parallel coordinates: The density of clusters are often distorted and non-existing ghost-clusters emerge. As a solution, we propose a novel rendering technique for the polyline segments: The line width is adjusted according to the angle of each line segment. Our method can be computed in linear time, depends on a single parameter, and can be combined with many existing parallel coordinates' variations.

Acknowledgments

We thank the anonymous reviewers for their feedback. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within the projects A03 of TRR 161 (Project-ID 251654672) and Knowledge Generation in VA (Project-ID 350399414).

[1] See `http://subspace.dbvis.de/pcp-adjustment` for the tool and `https://github.com/davidpomerenke/slope` for code and data.

References

[1] J. Alsakran, Y. Zhao, and X. Zhao. Tile-based parallel coordinates and its application in financial visualization. In *Visualization and Data Analysis*, 2010. doi: 10.1117/12.838819

[2] G. Andrienko and N. Andrienko. Constructing parallel coordinates plot for problem solving. In *1st International Symposium on Smart Graphics*, pp. 9–14, 2001.

[3] S. Bachthaler and D. Weiskopf. Continuous scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1428–1435, 2008. doi: 10.1109/TVCG.2008.119

[4] M. Behrisch, M. Blumenschein, N. W. Kim, L. Shao, M. El-Assady, J. Fuchs, D. Seebacher, A. Diehl, U. Brandes, H. Pfister, T. Schreck, D. Weiskopf, and D. A. Keim. Quality metrics for information visualization. *Computer Graphics Forum*, 37(3):625–662, 2018. doi: 10.1111/cgf.13446

[5] A. Dasgupta and R. Kosara. Pargnostics: Screen-space metrics for parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1017–1026, 2010. doi: 10.1109/TVCG.2010.184

[6] G. P. Ellis and A. J. Dix. A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1216–1223, 2007. doi: 10.1109/TVCG.2007.70535

[7] Y. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *IEEE Conference on Visualization*, pp. 43–50, 1999. doi: 10.1109/VISUAL.1999.809866

[8] J. Heinrich, Y. Luo, A. E. Kirkpatrick, and D. Weiskopf. Evaluation of a bundling technique for parallel coordinates. In *Proceedings of the International Conference on Computer Graphics Theory and Applications and International Conference on Information Visualization Theory and Applications*, pp. 594–602, 2012.

[9] J. Heinrich and D. Weiskopf. Continuous parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1531–1538, 2009. doi: 10.1109/TVCG.2009.131

[10] J. Heinrich and D. Weiskopf. State of the art of parallel coordinates. In *Eurographics 2013 - State of the Art Reports,*, pp. 95–116. The Eurographics Association, 2013. doi: 10.2312/conf/EG2013/stars/095-116

[11] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985. doi: 10.1007/BF01898350

[12] J. Johansson, P. Ljung, M. Jern, and M. D. Cooper. Revealing structure within clustered parallel coordinates displays. In *IEEE Symposium on Information Visualization*, pp. 125–132, 2005. doi: 10.1109/INFVIS.2005.1532138

[13] K. Koffka. *Principles of Gestalt psychology*. Mimesis International, September 2014.

[14] G. Palmas, M. Bachynskyi, A. Oulasvirta, H. Seidel, and T. Weinkauf. An edge-bundling layout for interactive parallel coordinates. In *IEEE Pacific Visualization Symposium*, pp. 57–64, 2014. doi: 10.1109/PacificVis.2014.40

[15] H. Siirtola, T. Laivo, T. Heimonen, and K. Räihä. Visual perception of parallel coordinate visualizations. In *13th International Conference on Information Visualisation*, pp. 3–9, 2009. doi: 10.1109/IV.2009.25

[16] H. Siirtola and K. Räihä. Interacting with parallel coordinates. *Interacting with Computers*, 18(6):1278–1309, 2006. doi: 10.1016/j.intcom.2006.03.006

[17] K. Stockinger, K. Wu, S. Campbell, S. Lau, M. Fisk, E. M. Gavrilov, A. Kent, C. E. Davis, R. D. Olinger, R. J. Young, J. Prewett, P. M. Weber, T. P. Caudell, E. W. Bethel, and S. Smith. Network traffic analysis with query driven visualization SC 2005 HPC analytics results. In *Proceedings of the 2005 ACM/IEEE SC2005 Conference on Supercomputing*, 2005. doi: 10.1109/SC.2005.47

[18] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 2nd ed., 2001.

[19] C. Ware. *Information visualization: perception for design*. Morgan Kaufmann, 3rd ed., May 2012.

[20] H. Zhou, W. Cui, H. Qu, Y. Wu, X. Yuan, and W. Zhuo. Splatting the lines in parallel coordinates. *Computer Graphics Forum*, 28(3):759–766, 2009. doi: 10.1111/j.1467-8659.2009.01476.x