Parallel-Coordinates Art

Julian Heinrich, *Student Member, IEEE* and Daniel Weiskopf, *Member, IEEE Computer Society*

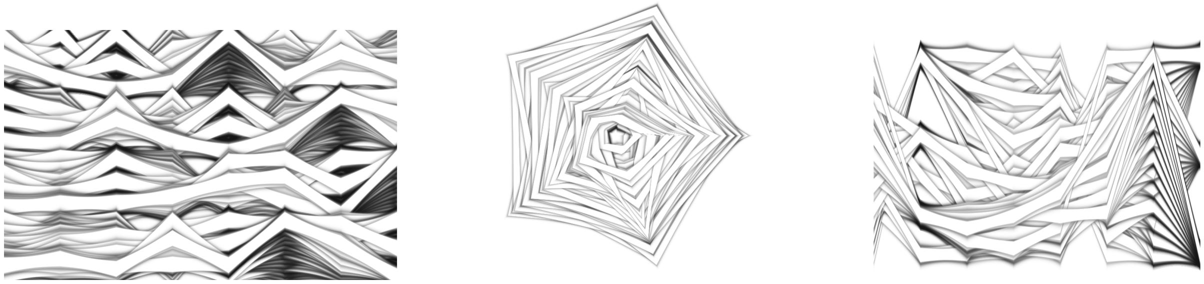

Fig. 1. Examples of artistic parallel coordinates: (left) parallel-coordinates matrix, (center) stardinates, (right) traditional parallel-coordinates layout.

Abstract—We employ common information visualization techniques such as parallel coordinates or stardinates as a tool for artistic image generation. Our technique is based on rendering footprints of continuous density distributions as they emerge from modeling data with uncertainty or estimating probability densities. By compositing density footprints with alpha-blending, we obtain visually pleasing images from high-dimensional input data. We further give examples of a number of controls that can be used to tailor the outcome of the algorithm.

Index Terms—Parallel coordinates, density estimation, infographics, art

1 Introduction

Typical information visualization techniques are employed to communicate quantitative or qualitative information visually [25] or to provide means for explorative data analysis [26]. The emerging diagrams are explicitly designed to *show the numbers* [4] with as few ink as possible and to focus on clarity and accuracy instead of aesthetics. As one particular example, parallel coordinates are based on projective geometry [13] and can be used to visualize multidimensional data in two dimensions using points, lines, or other geometry [9]. In this paper, we extend the traditional line-based rendering of parallel-coordinates plots to obtain artistic images and illustrate the effect of changing parameters such as the order or the layout of axes.

Our approach is based on modeling data points as probability density distributions, similar to the well-known kernel density estimation (KDE), a technique widely applied in statistical data analysis [24]. However, instead of computing a density that approximates the true (but typically unknown) *overall* probability density distribution of a given dataset from the contributions of many samples, we compute the footprint of each data point in parallel coordinates and employ alpha-blending to obtain an image of the dataset.

2 Related Work

There are many ways to create artistic images using well-known algorithms that produce fractals [2] or compute solutions to famous problems in computer science, such as the traveling-salesman problem [15]. In this paper, we exploit the theory of parallel coordinates, density estimation techniques, and image compositing, to devise a new rendering method that produces aesthetic visuals and allows for a wide range of controls.

Parallel coordinates and the point–line duality were first published in the context of nomography [17]. The mathematical model of parallel coordinates and many generalizations using projective geometry were further developed by Inselberg [13], who recently published the first textbook covering mathematical proofs, generalizations, and applications to data analysis for this multidimensional visualization technique [14]. Wegman introduced parallel coordinates for high-dimensional data analysis [27] and clustering [28]. A recent survey on visualization techniques for parallel coordinates has been presented by Heinrich and Weiskopf [9].

In order to mitigate the clutter of lines resulting from a large number of samples, various density-based approaches to rendering parallel coordinates were proposed in the visualization literature, including transparency [27] and binning [1, 21]. We employ a line-density model [19] to compute footprints of data samples in parallel coordinates by transforming the respective density distribution with the point–line duality. Similar models can be applied to visualize densities from spatiotemporal data [6, 8] and uncertain data [3] in parallel coordinates. While we share the density model for individual samples, we employ a different compositing algorithm and thus do not obtain a density distribution in parallel coordinates that resembles a density estimate of the input data.

McDonnell and Mueller [18] use edge bundling [11], opacity, and shading effects when rendering parallel coordinates. While they focus on showing structures in the data such as cluster density, our method is aimed at rendering visually pleasing images and can be used with any type of input data.

Stardinates [16] use a radial layout of axes with a similar representation of data points as polylines. We extend our density-based approach from parallel coordinates to the radial layout of stardinates.

To illustrate the generality of our approach, we demonstrate another extension of the technique to the parallel-coordinates matrix [7] that allows us to visualize all pairwise relations of a multivariate dataset in

• *The authors are with VISUS (Visualization Research Center), University of Stuttgart, Allmandring 19, 70569 Stuttgart, Germany, E-mail: {julian.heinrich, daniel.weiskopf}@visus.uni-stuttgart.de.*

Manuscript received 31 March 2013; accepted 1 August 2013; posted online 13 October 2013; mailed on 27 September 2013. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

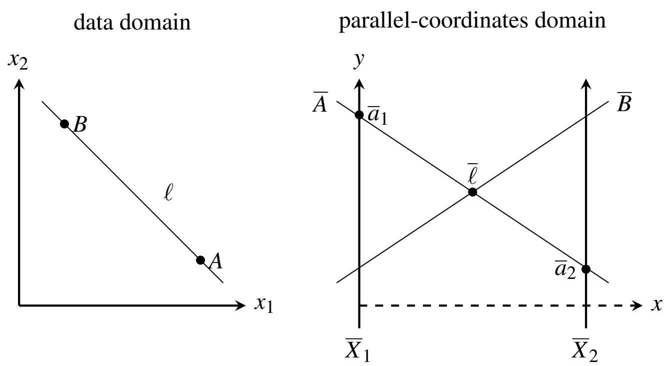

Fig. 2. Point–line duality between 2D Cartesian coordinates (left) and parallel coordinates (right): data points defined in the data domain are mapped to lines in the parallel-coordinates domain and vice versa.

a manner similar to the scatterplot matrix.

3 Mathematical Model

In this section, we briefly introduce the mathematical model and the construction of traditional parallel coordinates. Then, we explain the density model applied in our technique and the computation of footprints.

3.1 Data Domain

Multidimensional points are given in the *data domain*, which constitutes the set of N-dimensional real values $\mathbb{R}^N$, $N \in \mathbb{N}^+$ (Figure 2). We denote dimensions using indexed letters x_i with $0 < i \leq N$. A point $P \in \mathbb{R}^N$ is denoted by its coordinates $P = (p_1, p_2, \ldots, p_N)$ for dimensions x_i. The corresponding vector is $\mathbf{p} = (p_1, p_2, \ldots, p_N)^\mathrm{T}$.

3.2 Parallel-Coordinates Domain

A parallel-coordinates system for N-dimensional data is constructed by placing N copies of the y-axis

$$\overline{X}_i : \; x = d_i, \; 0 < i \leq N$$

at horizontal positions $\mathbf{d}_N = (d_1, d_2, \ldots, d_N)^\mathrm{T}$ with respect to the *embedding xy*-Cartesian coordinate system, as illustrated in Figures 2 and 3. Accordingly, the N-dimensional point $P \in \mathbb{R}^N$ is mapped to a polyline intersecting axes at the respective coordinates (d_i, p_i). Figure 3 illustrates the construction of a 5-dimensional parallel-coordinates system.

For $N = 2$, a point–line duality between Cartesian and parallel coordinates can be established [13, 17]. As shown in Figure 2, a point $A = (a_1, a_2)$ in Cartesian coordinates is represented as a line $\overline{A}$ in parallel coordinates, and a point $\overline{\ell}$ in parallel coordinates is mapped to a

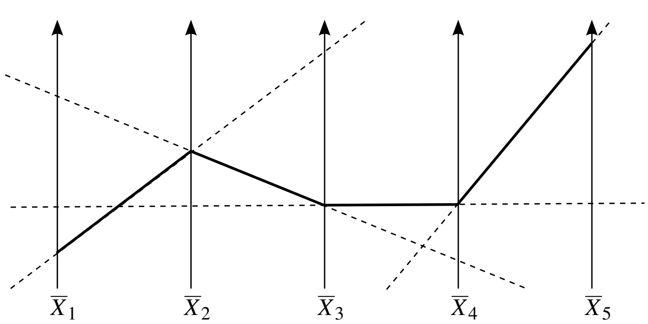

Fig. 3. A parallel-coordinates system with $N = 5$ dimensions. The bold polyline represents a five-dimensional point by joining the (stippled) lines between neighboring pairs of axes.

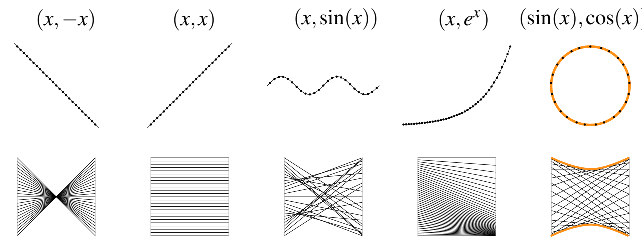

Fig. 4. Common patterns in Cartesian coordinates (top) and their dual representation in parallel coordinates (bottom). The envelope of lines is highlighted for the ellipse–hyperbola duality.

line ℓ in Cartesian coordinates. Figure 4 shows a number of patterns that can be derived from the point–line duality [14]. Of special interest for this work is the ellipse–hyperbola duality because it can be used to map Gaussian distributions to parallel coordinates.

3.3 Density Model

Instead of rendering discrete lines (as in Figure 5, left), we use two-dimensional normal distributions centered at the respective coordinates for each point P and pair of dimensions (i, j), $i \neq j$ to compute a density at any location $\mathbf{x} = (x_i, x_j)^\mathrm{T}$ with respect to the $x_i x_j$ Cartesian coordinate system:

$$g_p(\mathbf{x}) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{||\mathbf{x} - \mathbf{p}||}{2\sigma^2}\right] \qquad (1)$$

with mean vector $\mathbf{p} = (p_i, p_j)^\mathrm{T}$ and standard deviation σ.

Based on the point–line duality, density in parallel coordinates is obtained by transforming the footprint given in Equation (1) from the data domain to the parallel-coordinates domain, resulting in a *point-density* that can be evaluated at any point (x, y) in the parallel-coordinates system [19]:

$$h_p(x, y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(y - p_y)^2}{2\sigma^2}\right], \; x, y \in [0, 1] \qquad (2)$$

where $p_y = (1 - x)p_i + x p_j$ is linearly interpolated from the respective point-coordinates p_i and p_j at the axes. Note that we use normalized coordinates in the range $[0, 1]$ for the sake of simplicity. The construction of the overall plot can be split into the construction of $N - 1$ independent parallel-coordinates systems for two-dimensional points, each emerging from a two-dimensional data domain. The final plot is then formed by placing the parallel axes consecutively on the plane.

3.4 Footprint Computation

The footprint of a point in two-dimensional parallel coordinates can now be computed by evaluating Equation (2) in the parallel-coordinates domain for all consecutive pairs of axes. Figure 6 shows the density footprint of a point in Cartesian coordinates and its dual representation in parallel coordinates (note the hyperbolic shape of the contours in the footprint of parallel coordinates). To speed up computation, we pre-compute a single footprint for $\mathbf{p} = (0, 0)^\mathrm{T}$ and $\tilde{\sigma} = 0.01$ and store the result in a texture of fixed resolution. Then, for every data sample, the footprint is scaled and translated according to the point-coordinates of the current data point and a user-defined width w, effectively resulting in a scaled standard deviation $\sigma = \tilde{\sigma} w$. Another option is to render a screen-filling quad and evaluate Equation (2) for every pixel in a fragment shader. Note that the width of a footprint only depends on σ, which can be chosen freely to control the dimensions of individual footprints. A comparison of different values for σ is given in Figure 9.

4 Image Composition

In the previous section, the computation of footprints for individual points in parallel coordinates was presented. For a set of points, the re-

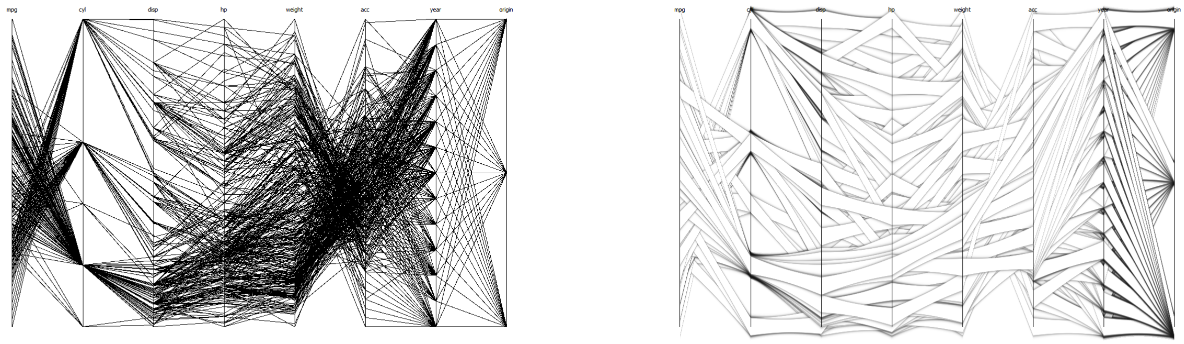

Fig. 5. Traditional parallel coordinates (left) compared to our technique (right) using density-based footprints. Both examples show the same data (a subset of the ASA cars dataset [23]). The hyperbolic shape of the footprints in combination with alpha-blending is visually more appealing, while more details and insight into the structure of the data can be seen from the discrete rendering.

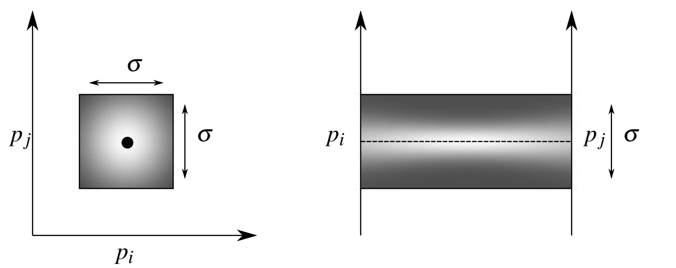

Fig. 6. Footprints of a normal distribution in Cartesian coordinates (left) and parallel coordinates (right). The density as computed by Equations (1) and (2) is mapped to a grayscale texture, where low values are mapped to black and high densities are mapped to white. The geometry used to render the footprints is obtained by mapping vertices of a quad located at $\mathbf{p} = (p_i, p_j)^{\mathrm{T}}$ to the corresponding parallel-coordinates system. Width and height of the quad are controlled by σ.

spective footprints are composited [22] into the final image, where different compositing operators yield different visual results (Figure 7). With additive blending, the contributions of each footprint to the final density are accumulated, which effectively produces a continuous parallel-coordinates plot [8, 19] (up to a constant scaling factor); here, individual footprints vanish and cannot be distinguished by their geometry. In contrast to additive blending, individual footprints remain visible when alpha-blending is used instead.

Figure 5 compares the traditional discrete rendering model with aesthetic parallel coordinates for the ASA cars dataset [23]. The same data is used in Figure 7 to illustrate the difference between our rendering model and the kernel density estimate for parallel coordinates, where color has been mapped to the final density. Note how the density estimation reveals dense and sparse areas that cannot be seen using our technique, as the superimposition of samples using alpha-blending makes the visual signature of the final image heavily depend on the order of samples being rendered. Figure 8 shows the same dataset with a different order of samples.

5 Artistic Control

There are many parameters that allow us to control how the final image is created. As with traditional parallel coordinates, these include the input data, the order and the layout of axes, as well as parameters of individual footprints such as color and size.

5.1 Layout

We first extend our rendering model to other layouts by employing simple transformations to the axes of single or multiple parallel-coordinates plots.

5.1.1 Polar Coordinates

Stardinates [16] can be obtained from parallel coordinates by transforming the embedding *xy*-coordinate system to polar coordinates and connecting lines between the emerging axes. The same footprints as obtained from Equation (2) can now be used to render the same hyperbolic shapes as for "linear" parallel coordinates, which is illustrated in Figure 10 for a set of cars and a dataset generated by the simulation of a parameterized biological reaction network [5]. Note that in order to circumvent the singularity emerging at the center of the coordinate system, we added a small offset to the minimum value for each axis, which effectively results in a "hole" at the center of artistic stardinates.

5.1.2 Matrix Layout

In parallel coordinates, the order of axes has a large impact on the final visualization. For data analysis, this is sometimes referred to as the axis-order problem [9]. As the order of axes determines which of all possible axis-aligned 2D projections of the N-dimensional input dataset are exhibited, choosing a "good" permutation is difficult and highly depends on the data-analysis task to be conducted. A straightforward solution is to enumerate all permutations and to visualize one parallel-coordinates system for each axis order. Due to the combinatorial explosion, however, this is usually not feasible, such that a number of heuristics have been proposed (see [9] for an overview). In order to visualize all pairwise relations of an N-dimensional dataset, the *parallel-coordinates matrix* [7] can be employed. This approach is based on a Hamiltonian decomposition of the complete graph K_N, where nodes represent dimensions and edges represent 2D relations between dimensions [12]. For N dimensions, a set of $\frac{N}{2}$ parallel-coordinates plots are obtained for $N = 2M$ and $\frac{N+1}{2}$ for $N = 2M + 1$. The parallel-coordinates matrix is then constructed by rendering a list of single parallel-coordinates plots in arbitrary order. As the matrix is independent of the rendering algorithm for each of the individual plots, our technique can easily be applied to the parallel-coordinates matrix, as illustrated in Figure 11.

5.2 Axis Order

While the matrix layout presented in the previous section is one possible method to circumvent the axis-order problem, both the traditional "linear" layout of axes as well as the radial layout used for stardinates depend on the order of axes. Figure 12 shows the effect of changing the axis order using the same input data.

5.3 Color

The color of each footprint is yet another source of variation that we exploited to control the appearance of artistic parallel coordinates. For data analysis, color is typically used to indicate group membership of lines to clusters or to encode other data-driven parameters and thus is required to adhere perceptual properties such as a uniform change in perceived importance. In the arts, any choice of color that is visually

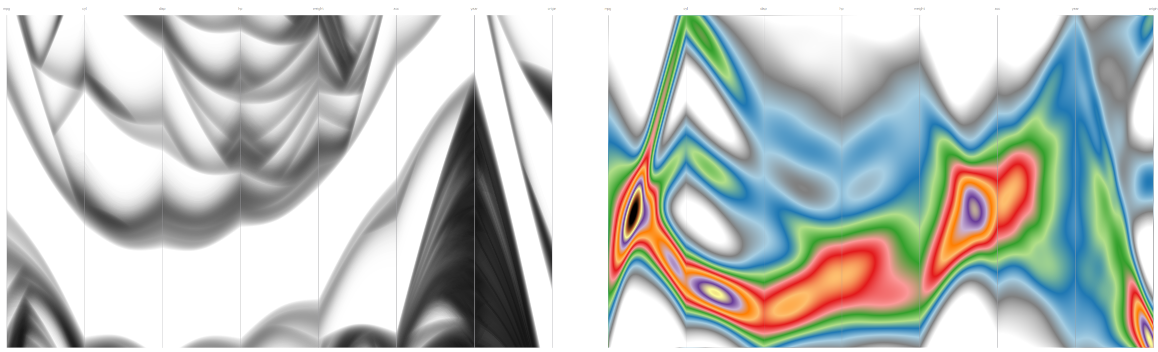

Fig. 7. The difference of employing alpha-blending (left) and additive blending (right). Again, the same subset of the ASA cars dataset was used to create both visualizations. The left image shows our technique with larger standard deviations than in Figure 5. For the right image, additive blending was employed to obtain a kernel density estimate of the data, which was further mapped to color for illustration.

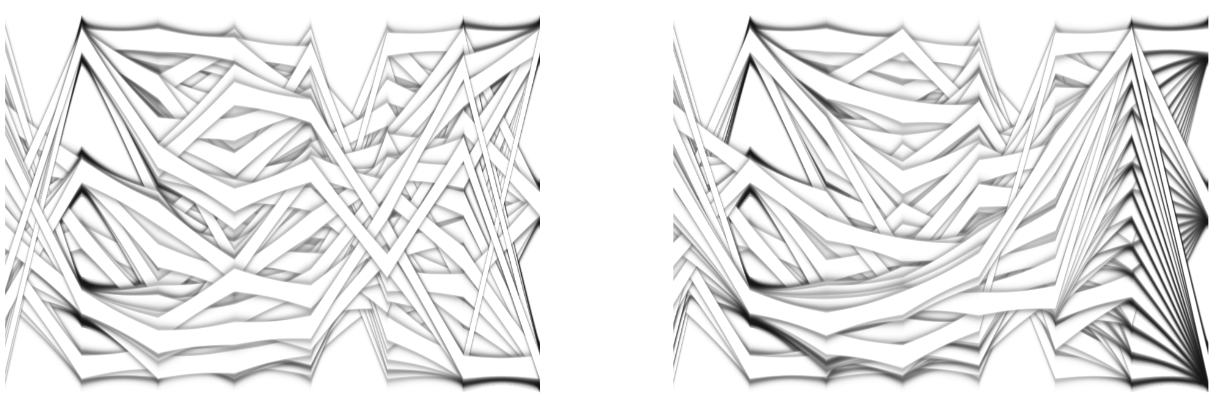

Fig. 8. Effect of the order of samples on the resulting image. Both visualizations were created using the same dataset, but with a different order of footprints. While the order was randomized for the left image, the parallel-coordinates plot on the right was created with data points grouped by the last (rightmost) axis.

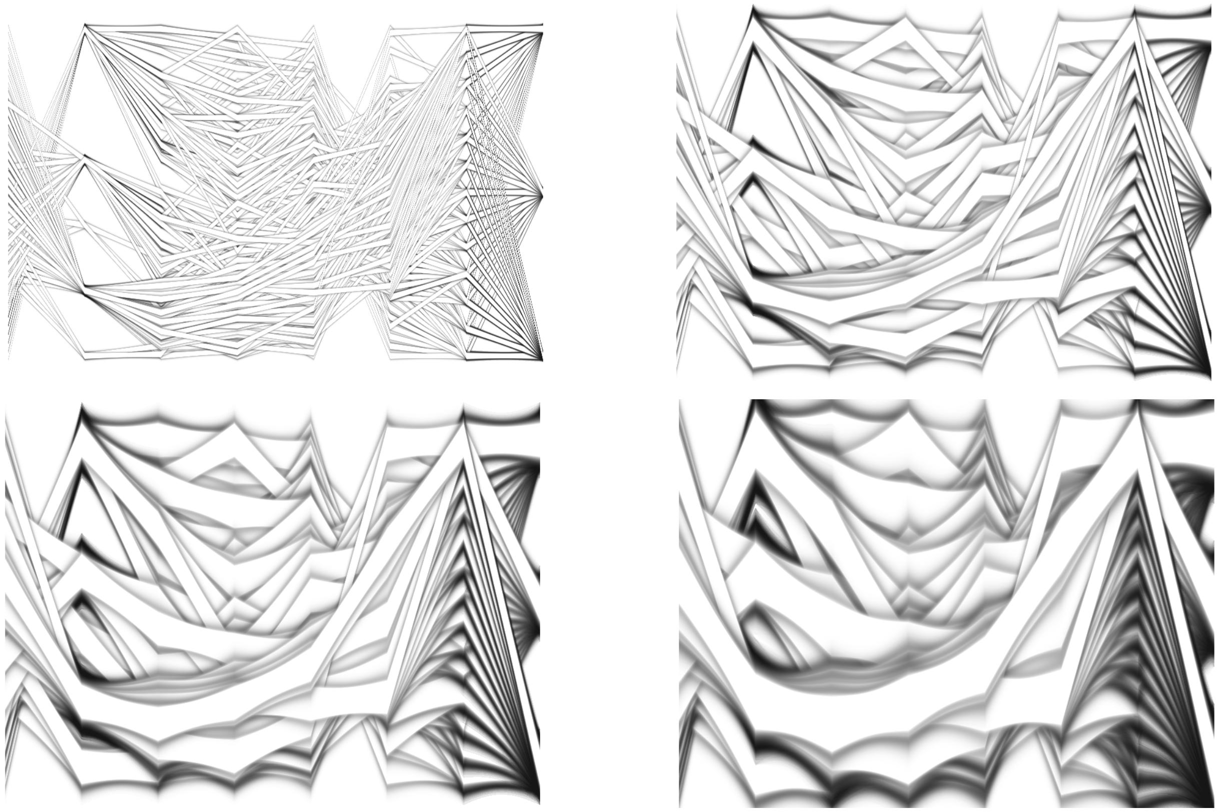

Fig. 9. The width of a footprint is determined by the standard deviation σ. The values for σ used here are: 0.025 (top left), 0.1 (top right), 0.15 (bottom left), and 0.25 (bottom right).

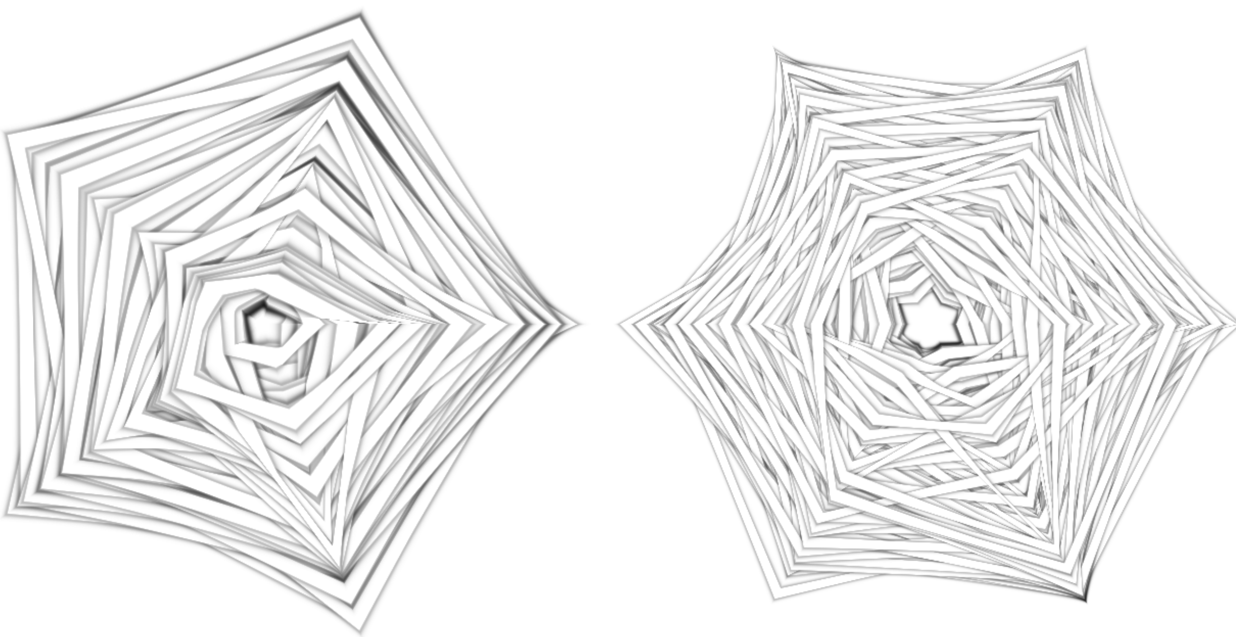

Fig. 10. Transforming the parallel-coordinates system to polar coordinates results in a radial representation of footprints. The left image was created from 5 attributes of cars and $\sigma = 0.1$, while the right image was rendered with $\sigma = 0.055$ from a dataset comprising six dimensions of a biological reaction network.

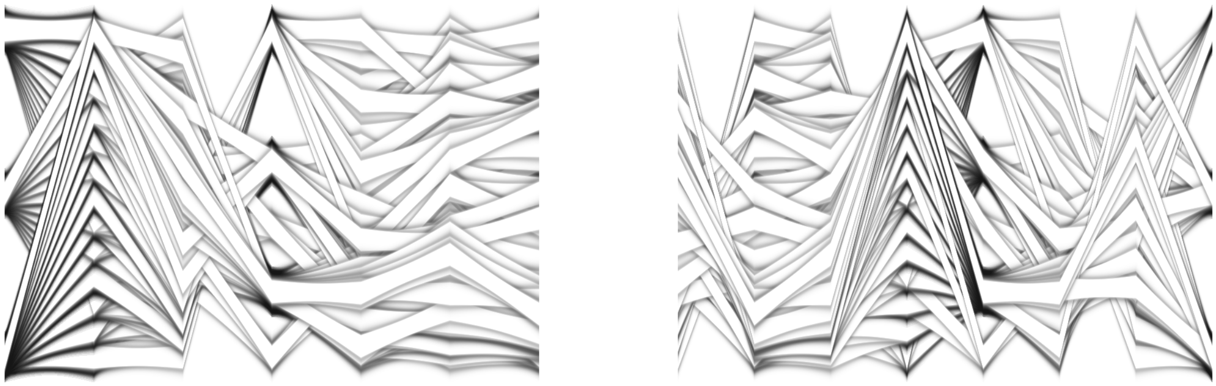

Fig. 12. The effect of the order of axes for the traditional layout of axes in artistic parallel coordinates.

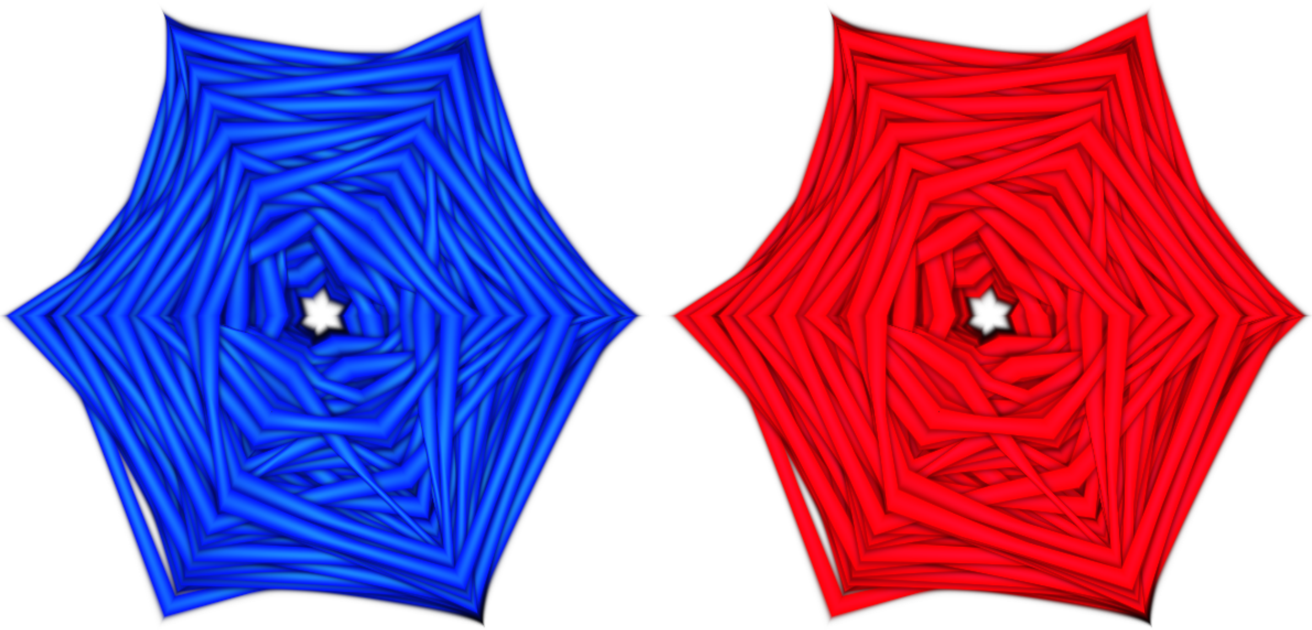

Fig. 13. The color of samples is another way of generating many different variants for artistic parallel coordinates. Note that a black background (not shown here) gives better results when viewed on a computer display, while a white background is generally more attractive on white paper.

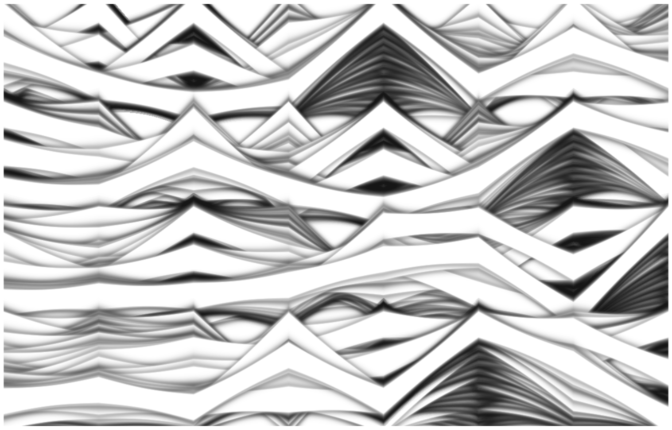

Fig. 11. Parallel-coordinates matrix of another cars dataset [10] using our technique. The matrix comprises three individual parallel-coordinates plots that are ordered in a list. By adjusting the vertical space between the plots, footprints start to overlap.

attractive can be used. Figure 13 provides some examples where a single color was applied to all footprints (by changing the color map that was used in Figure 6).

5.4 Input Data

Obviously, the input data is a parameter with a high impact on the visual result as it determines the position and orientation of individual footprints. In visual data-mining applications that employ parallel coordinates, characteristics of the visualization are typically used to find patterns in the data. For parallel-coordinates art where image aesthetics need to be controlled, the input data depends on the desired visual effects. As a further investigation of this dependency was out of the scope of this paper, we used only three datasets in this paper: two represent characteristics (as axes) of a set of cars (as lines or splats) taken from different sources [10, 23], and one is from a systems biology application [5] where axes denote parameters of a cell model and data points represent simulation results for individual cells.

Interactivity is another important aspect when working with parallel-coordinates, as it allows an analyst to change parameters of the visualization. For data-analysis tasks, most interaction techniques implement the selection of data points or the manipulation of axes. As the input data is just another parameter that allows the user to control the visual output of artistic parallel coordinates, it would certainly be possible to extract data from interacting with the user, e.g., by tracking the mouse input, and use that data as input to our algorithm.

6 Conclusion and Future Work

We presented a density-based rendering model for parallel coordinates that can be used to obtain visually pleasing images and showed how to control some of the parameters used in the model. In contrast to existing methods in statistical data analysis that typically aim to visualize (probability) density estimates in order to analyze data, our technique is not designed to reveal new insights into the dataset at hand but rather generates visually pleasing images. We showed how the method can be extended to other line-based visualizations such as stardinates. Other possible extensions are traditional line diagrams as they are used to convey time series such as stock return values or temperature forecasts. Other footprints can be computed using the same density-based model, such as ellipses for scatterplots or curves for parallel coordinates (see the overviews [9, 20] for more information on curve-based parallel coordinates and bundling).

The final image depends on the standard deviation σ, the order and layout of axes, and the order of data samples. While we have chosen to use a fixed value for the standard deviation for every sample and dimension, another source of variation could be added by assigning different values to σ for different samples or by changing the standard deviation between axes. The color of footprints and the background color can be adopted freely. Black background color gives good results on computer screens; we have chosen to use white as background for the production of the figures in this paper.

As mentioned in Section 5.4, we would like to investigate how the final images depend on the data and how the input data can be used to control the visual output of artistic parallel coordinates. In particular, time-varying input data could be used that would allow us to obtain smooth animations, e.g., in order to react to user input.

Acknowledgments

In part, this work was supported by Deutsche Forschungsgemeinschaft (DFG) within SFB 716 / D.5.

References

[1] A. O. Artero, M. C. F. de Oliveira, and H. Levkowitz. Uncovering clusters in crowded parallel coordinates visualizations. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 81–88, 2004.

[2] M. Barnsley. *Fractals Everywhere*. Academic Press Professional, Inc., San Diego, CA, USA, 1988.

[3] D. Feng, L. Kwock, Y. Lee, and R. M. Taylor II. Matching visual saliency to confidence in plots of uncertain data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):980–989, 2010.

[4] S. Few. *Show Me the Numbers: Designing Tables and Graphs to Enlighten*. Analytics Press, USA, 2nd edition, 2012.

[5] J. Hasenauer, J. Heinrich, M. Doszczak, P. Scheurich, D. Weiskopf, and F. Allgöwer. A visual analytics approach for models of heterogeneous cell populations. *EURASIP Journal on Bioinformatics and Systems Biology*, 2012(1):1–13, 2012.

[6] J. Heinrich, S. Bachthaler, and D. Weiskopf. Progressive splatting of continuous scatterplots and parallel coordinates. *Computer Graphics Forum*, 30(3):653–662, 2011.

[7] J. Heinrich, J. Stasko, and D. Weiskopf. The parallel coordinates matrix. In *EuroVis–Short Papers*, pages 37–41, 2012.

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

[9] J. Heinrich and D. Weiskopf. State of the art of parallel coordinates. In *STAR Proceedings of Eurographics 2013*. Eurographics Association, 2013.

[10] H. V. Henderson and P. F. Velleman. Building multiple regression models interactively. *Biometrics*, 37(2):391–411, 1981.

[11] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, 2006.

[12] C. B. Hurley and R. W. Oldford. Pairwise display of high-dimensional information via Eulerian tours and Hamiltonian decompositions. *Journal of Computational and Graphical Statistics*, 19:861–886, 2010.

[13] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(4):69–91, 1985.

[14] A. Inselberg. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer, NY, USA, 2009.

[15] C. S. Kaplan and R. Bosch. TSP art. In *Renaissance Banff: Bridges 2005: Mathematical Connections in Art, Music and Science*, pages 303–310, 2005.

[16] M. Lanzenberger, S. Miksch, and M. Pohl. The stardinates - visualizing highly structured data. In *Proceedings of the Seventh International Conference on Information Visualization*, pages 47–52, 2003.

[17] Maurice d'Ocagne. *Coordonnées Parallèles et Axiales: Méthode de transformation géométrique et procédé nouveau de calcul graphique déduits de la considération des coordonnées parallèlles*. Gauthier-Villars, Paris, France, 1885.

[18] K. McDonnell and K. Mueller. Illustrative parallel coordinates. *Computer Graphics Forum*, 27(3):1031–1038, 2008.

[19] J. J. Miller and E. J. Wegman. Construction of line densities for parallel coordinate plots. In A. Buja and P. A. Tukey, editors, *Computing and Graphics in Statistics*, volume 36, pages 107–123. Springer-Verlag, NY, USA, 1991.

[20] R. Moustafa. Parallel coordinate and parallel coordinate density plots. *Wiley Interdisciplinary Reviews: Computational Statistics*, 3(2):134–148, 2011.

[21] M. Novotny and H. Hauser. Outlier-preserving Focus+Context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):893–900, 2006.

[22] T. Porter and T. Duff. Compositing digital images. *Computer Graphics (Proceedings of SIGGRAPH)*, 18(3):253–259, 1984.

[23] E. Ramos and D. Donoho. The 1983 ASA data exposition dataset: Cars. http://lib.stat.cmu.edu/datasets/cars.data, 1983.

[24] D. Scott and S. Sain. Multidimensional density estimation. *Handbook of Statistics*, 24(2004):229–261, 2005.

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

[26] J. W. Tukey. *Exploratory Data Analysis*. Behavioral Science: Quantitative Methods. Addison-Wesley Publishing Company, Massachusetts California London Amsterdam Ontario Sydney, 1977.

[27] E. J. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85(411):664–675, 1990.

[28] E. J. Wegman and Q. Luo. High dimensional clustering using parallel coordinates and the grand tour. In *Classification and Knowledge Organization*, pages 93–102. Springer, Berlin Heidelberg, 1997.