Continuous Parallel Coordinates

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

Abstract—Typical scientific data is represented on a grid with appropriate interpolation or approximation schemes, defined on a continuous domain. The visualization of such data in parallel coordinates may reveal patterns latently contained in the data and thus can improve the understanding of multidimensional relations. In this paper, we adopt the concept of continuous scatterplots for the visualization of spatially continuous input data to derive a density model for parallel coordinates. Based on the point–line duality between scatterplots and parallel coordinates, we propose a mathematical model that maps density from a continuous scatterplot to parallel coordinates and present different algorithms for both numerical and analytical computation of the resulting density field. In addition, we show how the 2-D model can be used to successively construct continuous parallel coordinates with an arbitrary number of dimensions. Since continuous parallel coordinates interpolate data values within grid cells, a scalable and dense visualization is achieved, which will be demonstrated for typical multi-variate scientific data.

Index Terms—Parallel coordinates, integrating spatial and non-spatial data visualization, multi-variate visualization, interpolation.

✦

1 INTRODUCTION

Parallel coordinates have become a common technique for the visualization of high-dimensional data. In parallel coordinates, axes are aligned parallel to each other and data points are mapped to lines intersecting the axes at the respective value. The embedding of an arbitrary number of parallel axes into the plane allows the simultaneous display of many dimensions, providing a good overview of the data. However, while the representation of discrete data points as lines may reveal trends and patterns latently contained in the data, it also tends to clutter the view due to potentially heavy overplotting. In consequence, classical parallel coordinates do not scale well with sample size, making it difficult to use with large datasets. Despite the overdraw problem, typical information visualization techniques have been gaining importance for the analysis of scientific data, allowing for the detection of patterns which otherwise are difficult to spot.

For the visualization of large scientific data, we introduce *continuous parallel coordinates*. Here, data is typically defined on a 2-D or 3-D continuous domain, represented on a grid with respective interpolation or approximation schemes. Our method uses parallel coordinates to derive a continuous density description for such data. Although the input data field has to be defined on a continuous domain, the function describing it does not necessarily need to be continuous.

The main contribution of this paper is the mathematical model of density in parallel coordinates. Our definition of point density is based on "counting" discrete lines: we derive the point density by examining the limit process of lines intersecting an interval with indefinitely small vertical extent. Using this model, a relation of point densities from 2-D continuous scatterplots [2] to continuous parallel coordinates is derived.

Furthermore, we examine different numerical and analytical solutions for the computation of the model. Based on the point–line duality of scatterplots and parallel coordinates, the algorithms can be divided in two classes. In the *scattering* approach, a density description in parallel coordinates is obtained implicitly by sampling points from the input field. In contrast, the *gathering* approach computes the density by integration within the scatterplot.

Continuous parallel coordinates exhibit several benefits: (i) The visualization does not depend on the resolution of the data, as the available interpolation schemes are used to compute the continuous representation in parallel coordinates. (ii) In contrast to other frequency plot construction algorithms, our method is parameter-free: it does not rely on bucket size, binning, or texture resolution which are commonly used for the approximation of density. (iii) A continuous density model scales well with sample size and resolution, providing the basis for a visualization for which overplotting cannot occur. This makes parallel coordinates interesting for the analysis of large data, particularly in the field of scientific visualization.

2 RELATED WORK

Parallel-coordinates visualization utilizes a duality of points and lines: points in m-dimensional data space are represented as lines crossing m parallel axes in the 2-D domain of the parallel-coordinates plot. The advantage of parallel coordinates is that there is no fundamental limit on data dimensionality. Parallel coordinates were introduced by Inselberg [14, 15], and subsequently extended by Wegman [26]. The mathematical and geometric background of the point–line duality is reviewed in Section 3.

Unfortunately, parallel-coordinates visualization in its original version is subject to a couple of issues. One problem is the over-plotting of lines, in particular for large data sets. With the current trend toward applying statistical and information visualization techniques to scientific data [9], large-data visualization has become ubiquitous. A popular solution to the over-plotting problem is to replace opaque lines by a density representation [19, 27]. This strategy is applied in many, more recent publications as well. For example, features of the density plots can be visually extracted by appropriate gray-scale mappings [1] or general transfer functions [17]. Density-based visualizations can also be applied to frequency plots [23]. The recent work by Blaas et al. [7] specifically targets the visualization of multi-variate scientific data by density-based parallel coordinates. We share the application domain and also apply our technique to the same example test data set: the hurricane Isabel flow simulation from the IEEE Visualization 2004 Contest[1].

For the visualization of categorical variables, parallel sets [5] have been introduced as an extension to discrete parallel coordinates. However, previous work that deals with continuous density representations for the final visualization ignores the continuous nature of scientific input data: typically, data discretized via grid points are displayed, neglecting the reconstruction on the continuous domain. In contrast, we specifically consider the continuity of the domain with respective data reconstruction. The same basic approach can be applied to scatterplots [2] or histograms [8, 24]. The construction of continuous parallel coordinates requires substantial modifications and extensions compared to scatterplots and histograms because the duality of points and lines needs be considered (see Sections 3 and 4).

- *The authors are with VISUS (Visualization Research Center), Universität Stuttgart, Nobelstr. 15, 70569 Stuttgart, Germany, E-mail: {julian.heinrich, weiskopf}@visus.uni-stuttgart.de.*

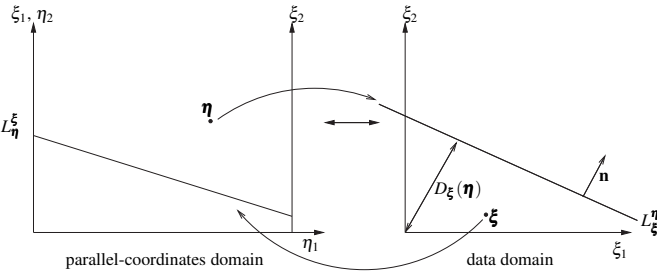[1]http://vis.computer.org/vis2004contest

Fig. 1. Parallel coordinates are constructed by placing parallel axes ξ_i on the $\eta_1\eta_2$-Cartesian coordinate system. A point in parallel coordinates is mapped to a line in the data domain and vice versa.

Clutter reduction for large-data visualization can be achieved by alternative approaches that are complementary to density plots and can be combined with those. For example, brushing-and-linking [4], originally developed for scatterplots, can be applied to parallel-coordinates plots in the form of angular brushing [13]. Another example is focus-and-context visualization with user-controlled lenses and adapted sampling of the data set [11]. Advanced four-level focus-and-context visualization, developed for the visualization of temporal features in large graph plots, shares aspects with density-based parallel coordinates and might be applied to them [20]. Alternatively, segmentation or clustering of the data might be included in order to separate distinct regions of the data: Johansson et al. [17] combine density plots with feature animation applied to clustered data; Novotny and Hauser [22] include the visualization of outliers and trends. Earlier work on cluster-based parallel coordinates includes aggregated visual representations in hierarchical plots [12], fuzzy cluster classification [6], and centroid visualization of clusters [27]. Finally, proximity in the visualization might be exploited by geometrically deforming the originally piecewise linear lines to curves [18, 28].

3 MATHEMATICAL MODEL

In this section, the mathematical model of continuous parallel coordinates is presented. After introducing the terminology and definitions used in this paper, the geometry of parallel coordinates is revisited. Then, a generic density model for parallel coordinates is derived.

The model of continuous parallel coordinates is based on the scalar density fields or continuous scatterplots [2] defined on an m-dimensional domain we will refer to as *data domain*. Following this terminology, another domain is introduced for the construction of continuous parallel coordinates: the *parallel-coordinates domain* defining a parallel-coordinates system in the Euclidean projective plane as introduced by Inselberg [14]. A nice feature of parallel coordinates is that the construction of the overall plot can be split into the construction of several independent parallel-coordinate systems for 2-D data, each emerging from a 2-D scatterplot. The final plot is then formed by placing the parallel axes consecutively on the plane. For m-dimensional data, this results in the computation of $m-1$ independent parallel-coordinate systems. Therefore, we will focus on 2-D data for the derivation of the mathematical model for continuous parallel coordinates.

3.1 Geometry of Parallel Coordinates

We briefly summarize parallel coordinates as presented in [14, 16], using our own notation. Parallel coordinates are constructed from a $\xi_1\xi_2$-Cartesian coordinate system by embedding the axes ξ_1 and ξ_2 in parallel onto another Cartesian coordinate system, the $\eta_1\eta_2$-Cartesian coordinate system (Figure 1). In order to distinguish points between data domain and parallel-coordinates domain, we will use the following notation throughout the rest of this paper. Generally, we discriminate between *attribute values* and their representation in the different coordinate systems. For any 2-D attribute, ξ_1 and ξ_2 denote the respective point coordinates in the data domain while η_1 and η_2 are used for point coordinates in the parallel-coordinates domain. If mappings of

multiple attributes have to be distinguished, superscripts are added to the respective coordinates. For example, a 2-D attribute $\mathbf{a}: (a_1, a_2)$ is mapped to the point $\boldsymbol{\xi}^{\mathbf{a}}: (\xi_1^{\mathbf{a}}, \xi_2^{\mathbf{a}})$ in the data domain. Dually, in the parallel-coordinates domain, the attribute $\mathbf{b}$ has coordinates $(\eta_1^{\mathbf{b}}, \eta_2^{\mathbf{b}})$ with respect to the $\eta_1\eta_2$-Cartesian coordinate system.

Following this notation, any point $\boldsymbol{\xi}: (\xi_1, \xi_2)$ in the data domain is mapped to a line segment between adjacent axes ξ_1 and ξ_2 in the parallel-coordinates domain:

$$L_{\boldsymbol{\eta}}^{\boldsymbol{\xi}}: \eta_2 = (\xi_2 - \xi_1)\eta_1 + \xi_1; \eta_1 \in [0, 1] \qquad (1)$$

Here, we set the distance between parallel axes ξ_1 and ξ_2 to one as proposed by Inselberg [14]. Note that we use subscripts to denote the domain in which the line is defined and superscripts for the parameter, i.e. the dual point to the line. Hence, the line in (1) is given with respect to the embedding $\eta_1\eta_2$-Cartesian coordinate system. In the data domain, equation (1) allows another interpretation. Here, it implicitly represents the line corresponding to the point $\boldsymbol{\eta}: (\eta_1, \eta_2)$ of the parallel-coordinates system with respect to the $\xi_1\xi_2$-Cartesian coordinate system. For this purpose, it may be interpreted as the projection of the vector $\boldsymbol{\xi}$ onto $\tilde{\mathbf{n}}$, which can be expressed by the dot product:

$$L_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}: \eta_2 = \tilde{\mathbf{n}} \cdot \boldsymbol{\xi} \qquad (2)$$

Note that $\tilde{\mathbf{n}} = (1 - \eta_1, \eta_1)^t$ is perpendicular to $L_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}$ and only depends on η_1.

The distance $D_{\boldsymbol{\xi}}$ of $L_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}$ to the origin is inherently contained in (2), but its computation assumes normalization of $\tilde{\mathbf{n}}$ to unit length, such that:

$$D_{\boldsymbol{\xi}}(\boldsymbol{\eta}) = \frac{\eta_2}{||\tilde{\mathbf{n}}||} = \frac{\tilde{\mathbf{n}}}{||\tilde{\mathbf{n}}||} \cdot \boldsymbol{\xi} \qquad (3)$$

Hence, the main conclusions of this section are two-fold: (i) the distance $D_{\boldsymbol{\xi}}(\boldsymbol{\eta})$ linearly correlates with η_2, the vertical position of the corresponding point in the parallel-coordinates domain and (ii) the slope $L_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}$ in the data domain only depends on η_1, the horizontal position of the corresponding point $\boldsymbol{\eta}$ in the parallel-coordinates domain.

3.2 Generic Density Model

Our proposed density model is based on mass conservation, assuming that (i) points in the data domain are given according to some density description, and (ii) the mapping of points from the data domain to lines in the parallel-coordinates domain does not change the number of points (lines), i.e. a point in the data domain corresponds to exactly one line in the parallel-coordinates domain and vice versa. As a consequence, a vertical line (or an interval) in the parallel-coordinates domain is mapped to a set of indefinitely dense *parallel* lines (or an area) in the data domain (see Figure 2). This can be used to derive a density description for points in parallel coordinates by examining the limit process at the transition of areas to lines in the data domain. With the assumptions (i) and (ii) stated above, the mass M covering an area $\Phi \subset \mathbb{R}^2$ in the data domain with density $\sigma: \mathbb{R}^2 \longrightarrow \mathbb{R}, \boldsymbol{\xi} \mapsto \sigma(\boldsymbol{\xi})$ is $M = \int_{\Phi} \sigma(\boldsymbol{\xi})\mathrm{d}^2\xi$. Considering the duality of points and lines, the density $\varphi: \mathbb{R}^2 \longrightarrow \mathbb{R}, \boldsymbol{\eta} \mapsto \varphi(\boldsymbol{\eta})$ of a point $\boldsymbol{\eta}$ in parallel coordinates is based on "counting" lines within an interval along the vertical axis. It can then be integrated to compute the mass of the covered interval Ω according to $\int_{\Omega} \varphi(\eta_1, \eta_2)\mathrm{d}\eta_2$. Assuming mass conservation, the mass of points (lines) does not change under the transformation from data domain to parallel-coordinates domain:

$$M = \int_{\Omega} \varphi(\eta_1, \eta_2)\mathrm{d}\eta_2 = \int_{\Phi} \sigma(\boldsymbol{\xi})\mathrm{d}^2\xi \qquad (4)$$

Here, we assume the density $\sigma(\boldsymbol{\xi})$ to be known for any $\boldsymbol{\xi}$ (see [2] for a derivation of densities in the data domain). Applying the fundamental theorem of calculus to (4) allows us to express the density in the parallel-coordinates domain in terms of σ:

$$\varphi(\eta_1, \eta_2) = \frac{\mathrm{d}M}{\mathrm{d}\eta_2} = \frac{\mathrm{d}}{\mathrm{d}\eta_2} \int_{\Phi} \sigma(\boldsymbol{\xi})\mathrm{d}^2\xi \qquad (5)$$
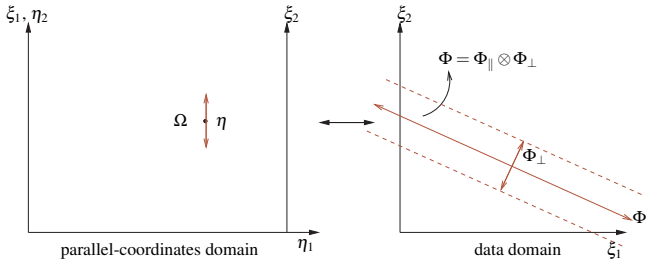
Fig. 2. The interval Ω containing $\boldsymbol{\eta}$ in the parallel-coordinates domain is mapped to the area stripe Φ containing $L_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}$ in the data domain. Since the slope of $L_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}$ is independent from η_2, the stripe has parallel border lines.

In order to compute the integral, we split Φ in two parts: one integration along the line $\Phi_{\parallel} = L_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}$ corresponding to $\boldsymbol{\eta}$ and another integration along the perpendicular direction $\Phi_{\perp}$ (see Figure 2). For this purpose, the $\xi_1 \xi_2$-coordinate system is rotated such that the rotated ξ_2-axis can be identified with the normal of the line $L_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}$. Then, we can use

$$\frac{\mathrm{d}D_{\boldsymbol{\xi}}}{\mathrm{d}\eta_2} = \frac{1}{||\tilde{\mathbf{n}}||} \tag{6}$$

as a result of (3) to transform the integration over $\Phi_{\perp}$ to an integration over Ω. Considering the limit process for indefinitely small intervals in the parallel-coordinates domain further eliminates integration over $\Phi_{\perp}$, such that the line density in a point (η_1, η_2) of the parallel-coordinates system is fully described by the integral over the corresponding line in the data domain:

$$\varphi(\eta_1, \eta_2) = \int_{L_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}} \frac{\sigma(\mathbf{L}_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}(\lambda))}{||\tilde{\mathbf{n}}||} \mathrm{d}\lambda \tag{7}$$

with $\mathbf{L}_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}(\lambda)$ being the arc-length parametrized line $L_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}$. Note that σ typically has finite support, although $L_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}$ is defined on an indefinite domain. A complete derivation of (7) is provided in the appendix.

3.3 Numerical Integration

Equation (7) describes the line density at any point in the parallel-coordinates domain as a line integral along its dual line in the data domain, where the function to be integrated is the respective point density σ of the scalar input field. In this section, two substantially different approaches to the numerical integration of (7) are briefly discussed.

A typical *gathering* technique is to sample φ in the parallel-coordinates domain followed by an evaluation of (7) in the data domain. Here, each sample $\boldsymbol{\eta}$ has a dual line $L_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}$ constituting the integration domain for the computation of $\varphi(\boldsymbol{\eta})$. Numerical integration now implies further sampling of σ over $L_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}$ and can be implemented using known techniques such as Monte Carlo integration or Riemann sums.

By exploiting the point–line duality, another approach to numerical integration of (7) is possible. Here, points are sampled from the data domain and the respective densities are *scattered* to line densities in the parallel-coordinates domain. The generic scattering algorithm using additive blending is

1: sample points $\boldsymbol{\xi}_i, i = 1, 2, \ldots, n$
2: **for all** $\boldsymbol{\xi}_i$ **do**
3:     setRGBAdrawColor$(1, 1, 1, \alpha)$
4:     drawLine$(L_{\boldsymbol{\eta}}^{\boldsymbol{\xi}_i})$
5: **end for**

A possible application of the scattering algorithm is to sample points on a regular grid on the data domain (step 1) and set $\alpha \leftarrow \sigma(\boldsymbol{\xi}_i)$, effectively resulting in a uniform sampling of the density function σ.

Note that point densities φ are then constructed implicitly by the superposition of lines with different density. Due to the linear model of (7), this leads to the same result as the gathering approach. Instead of sampling uniformly on a regular grid in the data domain, a random sampling strategy (with a uniform probability distribution) could be used to achieve an "implicit" Monte Carlo integration for the computation of density in parallel coordinates. Similarly, low-discrepancy sequences [21] could be used for sampling to obtain quasi Monte Carlo integration. Using σ in an importance sampling approach further improves performance compared with the standard or quasi Monte Carlo methods. In this case, a constant density α must be used for $L_{\boldsymbol{\eta}}^{\boldsymbol{\xi}_i}$, i.e. $\alpha \leftarrow const.$ in step 3 of the generic scattering algorithm. Sample points are drawn from a probability density function given by σ, up to a constant scaling factor. Now, the computation of $\varphi(\boldsymbol{\eta})$ at the sampling points $\boldsymbol{\eta}$ remains only a matter of counting the (weighted) lines intersecting with $\boldsymbol{\eta}$, which also is the basis of our mathematical model of continuous parallel coordinates. Note that φ depends on the number of samples and thus has to be normalized in order to properly compare the results.

In practice, many 2-D density fields are derived from higher dimensional input fields with known (sampling) densities, such as 3-D scalar fields, 3-D vector fields, or multi-attribute fields. Bachthaler and Weiskopf [2] denote the domain of such an input field as *spatial domain* and describe the transformation of density from the spatial domain to the data domain under the assumption of mass conservation. In consequence, the computation of continuous parallel coordinates using scattering may also be conducted on the spatial domain. Here, multi-dimensional points are sampled and mapped to polylines in parallel coordinates with $\alpha \leftarrow const$. This approach affects step 1 of the generic scattering algorithm, as points are now sampled according to the given density in the spatial domain (typically, constant density). This method and previous density-based methods (such as [17]) converge to the same basic computation with increasing grid resolution of the input field. Therefore, in the limit of infinitely high resolution of input data, continuous parallel coordinates and previous density-based representations yield the same result.

3.4 Triangulated Data

In this section, we provide an analytic solution to (7) for data given on tetrahedral grids in the spatial domain. Tetrahedral grids play an important role as simulation grids or as common ground for data exchange using the approximation of other grid structures by triangulation. Continuous scatterplots also support tetrahedral grids by exploiting the projected tetrahedra algorithm [25]. Under the assumption of mass conservation, spatial tetrahedra are projected to a set of triangles in the data domain, resulting in a triangulation of the density distribution with piecewise linear interpolation. Therefore, a piecewise computation of $\varphi(\boldsymbol{\eta})$ can be achieved by linear superposition of the contribution of all triangles intersecting the dual line $L_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}$. This approach is similar to the previously described scattering of densities, although in this case, triangles instead of points are mapped to parallel coordinates.

Figure 3 shows a possible footprint of a triangle Δ_{abc} from the data domain to the parallel-coordinates domain. The points $\boldsymbol{\xi}^{\mathbf{a}}$, $\boldsymbol{\xi}^{\mathbf{b}}$, and $\boldsymbol{\xi}^{\mathbf{c}}$ are mapped to lines $L_{\boldsymbol{\eta}}^{\mathbf{a}}, L_{\boldsymbol{\eta}}^{\mathbf{b}}$, and $L_{\boldsymbol{\eta}}^{\mathbf{c}}$ in the parallel-coordinates domain, as described in (1). For any vertical line $\eta_1 = \eta_1^{\boldsymbol{\omega}} = const.$, the intersections with $L_{\boldsymbol{\eta}}^{\mathbf{a}}, L_{\boldsymbol{\eta}}^{\mathbf{b}}$, and $L_{\boldsymbol{\eta}}^{\mathbf{c}}$ are $\eta_2^{\mathbf{a}}, \eta_2^{\mathbf{b}}$, and $\eta_2^{\mathbf{c}}$, as derived in (1). Without loss of generality, let

$$\eta_2^{\mathbf{a}} \leq \eta_2^{\mathbf{c}} \leq \eta_2^{\mathbf{b}}. \tag{8}$$

This means that for each triangle, we label its vertices such that (8) is true. Then, Δ_{abc} is divided in two subtriangles Δ_{aec} and Δ_{ebc}. Here, a case differentiation is necessary depending on the choice of $\eta_2^{\boldsymbol{\omega}}$. First, let $\eta_2^{\mathbf{a}} \leq \eta_2^{\boldsymbol{\omega}} \leq \eta_2^{\mathbf{c}}$ (highlighted red in Figure 3). The corresponding line $L_{\boldsymbol{\xi}}^{\boldsymbol{\omega}}$ in the data domain intersects Δ_{aec} in the points $\boldsymbol{\xi}^{\mathbf{f}}$ and $\boldsymbol{\xi}^{\mathbf{g}}$:

$$\mathbf{L}_{\boldsymbol{\xi}}^{\boldsymbol{\omega}}(\lambda) = \boldsymbol{\xi}^{\mathbf{f}} + \frac{\lambda}{t}(\boldsymbol{\xi}^{\mathbf{g}} - \boldsymbol{\xi}^{\mathbf{f}}) \tag{9}$$
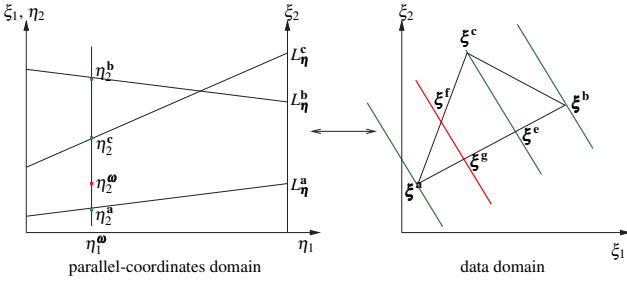
Fig. 3. Footprint of a triangle in parallel coordinates after transformation from the data domain. The point $\eta_2^{\boldsymbol{\omega}}$ and its dual line $L_{\boldsymbol{\xi}}^{\boldsymbol{\omega}}$ are highlighted in red. Assuming, $\eta_2^{\mathbf{a}} < \eta_2^{\mathbf{c}} < \eta_2^{\mathbf{b}}$, the triangle Δ_{abc} is divided in two sub-triangles Δ_{aec} and Δ_{ebc}.

with $t = ||\boldsymbol{\xi}^{\mathbf{g}} - \boldsymbol{\xi}^{\mathbf{f}}||$ and $\lambda \in [0, t]$ for the segment contained in Δ_{abc}. Due to the piecewise linear density distribution obtained from the projected tetrahedra algorithm, we can use that

$$\sigma(\mathbf{L}_{\boldsymbol{\xi}}^{\boldsymbol{\omega}}(\lambda)) = \sigma(\boldsymbol{\xi}^{\mathbf{f}}) + \frac{\lambda}{t}\left(\sigma(\boldsymbol{\xi}^{\mathbf{g}}) - \sigma(\boldsymbol{\xi}^{\mathbf{f}})\right) \quad (10)$$

such that the contribution $\varphi_{\mathbf{f}}^{\mathbf{g}}$ to $\varphi(\boldsymbol{\eta}^{\boldsymbol{\omega}})$ as computed according to (7) is:

$$\varphi_{\mathbf{f}}^{\mathbf{g}} = \int_0^t \frac{\sigma(\mathbf{L}_{\boldsymbol{\xi}}^{\boldsymbol{\omega}}(\lambda))}{||\tilde{\mathbf{n}}||} d\lambda = \frac{t}{2||\tilde{\mathbf{n}}||}\left(\sigma(\boldsymbol{\xi}^{\mathbf{f}}) + \sigma(\boldsymbol{\xi}^{\mathbf{g}})\right) \quad (11)$$

Now, we can use barycentric interpolation in subtriangle Δ_{aec} to obtain the density at the intersection points:

$$\sigma(\boldsymbol{\xi}^{\mathbf{g}}) = \sigma(\boldsymbol{\xi}^{\mathbf{a}}) + u(\sigma(\boldsymbol{\xi}^{\mathbf{e}}) - \sigma(\boldsymbol{\xi}^{\mathbf{a}})) \quad (12)$$

and

$$\sigma(\boldsymbol{\xi}^{\mathbf{f}}) = \sigma(\boldsymbol{\xi}^{\mathbf{a}}) + u(\sigma(\boldsymbol{\xi}^{\mathbf{c}}) - \sigma(\boldsymbol{\xi}^{\mathbf{a}})) \quad (13)$$

For the computation of u, distances of lines as derived in (3) can be used. Let $\Delta\eta_2^{\boldsymbol{\omega}} = \eta_2^{\boldsymbol{\omega}} - \eta_2^{\mathbf{a}}$ be the vertical distance of $\boldsymbol{\eta}^{\boldsymbol{\omega}}$ to $\boldsymbol{\eta}^{\mathbf{a}}$ in the parallel-coordinates domain. Similarly, let $\Delta D_{\boldsymbol{\xi}}^{\boldsymbol{\omega}} = D_{\boldsymbol{\xi}}(\boldsymbol{\eta}^{\boldsymbol{\omega}}) - D_{\boldsymbol{\xi}}(\boldsymbol{\eta}^{\mathbf{a}})$ be the distance of $L_{\boldsymbol{\xi}}^{\boldsymbol{\omega}}$ to $L_{\boldsymbol{\xi}}^{\mathbf{a}}$. Then, u can be derived using the intercept theorem in the data domain:

$$u = \frac{\Delta D_{\boldsymbol{\xi}}^{\boldsymbol{\omega}}}{\Delta D_{\boldsymbol{\xi}}^{\mathbf{c}}} = \frac{\Delta\eta_2^{\boldsymbol{\omega}}}{\Delta\eta_2^{\mathbf{c}}} \quad (14)$$

Note that $\eta_2^{\mathbf{a}} \le \eta_2^{\boldsymbol{\omega}} \le \eta_2^{\mathbf{c}}$ and thus $\lim_{\eta_2^{\mathbf{a}} \to \eta_2^{\mathbf{c}}} u(\eta_2^{\mathbf{a}}) = 1$ using l'Hôpital's rule so that (14) is defined even for $\Delta\eta_2^{\mathbf{c}} = 0$.

Similarly, for the computation of $\sigma(\boldsymbol{\xi}^{\mathbf{e}})$, barycentric interpolation within Δ_{abc} yields:

$$\sigma(\boldsymbol{\xi}^{\mathbf{e}}) = \sigma(\boldsymbol{\xi}^{\mathbf{a}}) + v(\sigma(\boldsymbol{\xi}^{\mathbf{b}}) - \sigma(\boldsymbol{\xi}^{\mathbf{a}})) \quad (15)$$

with

$$v = \frac{\Delta\eta_2^{\mathbf{c}}}{\Delta\eta_2^{\mathbf{b}}} \quad (16)$$

and $\Delta\eta_2^{\mathbf{b}} > 0$ (the special case $\Delta\eta_2^{\mathbf{b}} = 0$ will be treated later). Now, the final parameter to determine in order to solve equation (11) is $t = ||\boldsymbol{\xi}^{\mathbf{g}} - \boldsymbol{\xi}^{\mathbf{f}}||$, which can be obtained using the intercept theorem:

$$\frac{||\boldsymbol{\xi}^{\mathbf{g}} - \boldsymbol{\xi}^{\mathbf{f}}||}{||\boldsymbol{\xi}^{\mathbf{e}} - \boldsymbol{\xi}^{\mathbf{c}}||} = \frac{||\boldsymbol{\xi}^{\mathbf{g}} - \boldsymbol{\xi}^{\mathbf{a}}||}{||\boldsymbol{\xi}^{\mathbf{e}} - \boldsymbol{\xi}^{\mathbf{a}}||} \quad (17)$$

and thus:

$$t = u \cdot ||\boldsymbol{\xi}^{\mathbf{e}} - \boldsymbol{\xi}^{\mathbf{c}}|| \quad (18)$$

where $\boldsymbol{\xi}^{\mathbf{e}}$ is linearly interpolated similarly to (15).

Altogether, equation (11) resolves to a single expression depending only on the point coordinates $\boldsymbol{\eta}^{\boldsymbol{\omega}}$ in parallel coordinates and the densities at the triangle vertices:

$$\varphi_{\mathbf{f}}^{\mathbf{g}} = \frac{t}{2||\tilde{\mathbf{n}}||}\left((2 - u - uv)\sigma(\boldsymbol{\xi}^{\mathbf{a}}) + uv\sigma(\boldsymbol{\xi}^{\mathbf{b}}) + u\sigma(\boldsymbol{\xi}^{\mathbf{c}})\right) \quad (19)$$

The second case $\eta_2^{\mathbf{c}} \le \eta_2^{\boldsymbol{\omega}} \le \eta_2^{\mathbf{b}}$ is derived analogously by swapping indices a and b in equations (13), (12), and (14).

Note that both subtriangles Δ_{ebc} and Δ_{aec} may degenerate to a line if either $\eta_2^{\mathbf{c}} = \eta_2^{\mathbf{b}}$ or $\eta_2^{\mathbf{a}} = \eta_2^{\mathbf{c}}$. As these cases are covered by (14) and (16), there only remains the special case $\eta_2^{\mathbf{a}} = \eta_2^{\mathbf{c}} = \eta_2^{\mathbf{b}}$, where v is no longer defined. Here, Δ_{abc} degenerates to a line with three density values at the corresponding vertices, such that linear interpolation is not valid anymore. In this case, the density at $\boldsymbol{\eta}^{\boldsymbol{\omega}}$ according to the triangle-model can no longer be represented by a function in the parallel-coordinates domain. Instead, the degenerate triangle from the data domain maps to a single point in parallel coordinates. The associated density is represented by a delta distribution: $\varphi(\boldsymbol{\eta}) = M\delta(\boldsymbol{\eta} - \boldsymbol{\eta}^{\boldsymbol{\omega}})$, where M is the mass of the degenerate triangle which is conveniently determined by integration in the spatial domain.

4 IMPLEMENTATION

This section presents implementations of the different computational models introduced in Section 3.3 and 3.4. Each method will shortly be explained and applied to a test dataset comprising a single triangle with known density distribution in the data domain (see Figure 4) in order to evaluate the numerical quality of the different methods.

The implementations are based on C++ and OpenGL with GLSL. All calculations were performed using a 2048×2048 floating-point render target.

4.1 Triangulated Data

In Section 3.4, the contribution of the piecewise linear density given on a triangle to $\varphi(\boldsymbol{\eta})$ was reduced to a single equation depending only on $\boldsymbol{\eta}$ and the densities at the triangle vertices. This can be used to implement a rasterization of line densities in parallel coordinates. After projecting a tetrahedral mesh from the spatial domain to the data domain, the density distribution of each triangle is mapped to parallel coordinates according to (19). According to the linear density model, the total density $\varphi(\boldsymbol{\eta})$ can thus be computed using additive blending.

Using a floating-precision buffer as render target, the density is computed for each texel individually, such that the algorithm can easily be adapted for a GPU implementation. In particular, fast interpolation can be exploited for the computation of parameters to (19). Hence, the primitives have to be generated, such that the necessary parameters can be attached as texture coordinates. As can easily be seen in Figure 4, the footprint of a triangle in parallel coordinates consists of three lines, each representing one vertex of the triangle. In turn, each line may intersect each other line, such that a minimum of zero and a maximum of three intersections may occur. Dividing the horizontal axis at each intersection yields up to four *segments*, each consisting of two quadrilaterals. Rendering each quadrilateral with attached texture coordinates representing u, $\Delta\eta_2^{\mathbf{b}}$, and $\Delta\eta_2^{\mathbf{c}}$ then allows evaluation of equations (19) and (16) in a GPU fragment program. For the special case $\eta_2^{\mathbf{a}} = \eta_2^{\mathbf{c}} = \eta_2^{\mathbf{b}}$, we currently store a constant value in a separate channel of the render target in order to mark the corresponding pixel. In future, this may be considered for the final display. For a triangle Δ_{abc}, the algorithm consists of the following steps:

1. Determine all intersections of $L_{\boldsymbol{\eta}}^{\mathbf{a}}$, $L_{\boldsymbol{\eta}}^{\mathbf{b}}$, and $L_{\boldsymbol{\eta}}^{\mathbf{c}}$ and divide the horizontal axis into segments accordingly.

2. Determine upper and lower quadrilaterals (treat triangles as degenerate quadrilaterals) and attach parameters u, $\Delta\eta_2^{\mathbf{b}}$, $\Delta\eta_2^{\mathbf{c}}$ as texture coordinates to the corresponding vertices.

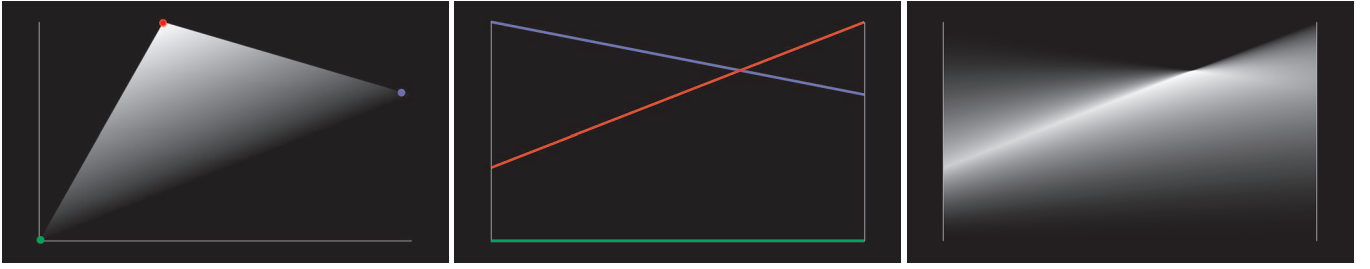3. Render quadrilaterals with fragment program enabled.

Fig. 4. The reference triangle with continuous density in the data domain (left), its footprint in the parallel-coordinates domain (middle) and the density plot using an analytic solution for triangulated data (right). The triangle vertices and respective lines in the footprint are marked red, green and blue. The density plots computed with numerical integration (gathering and scattering) are indistinguishable to the analytic solution. The respective l^2 distances are denoted in the main text.

Figure 4 shows the result of the implementation for the reference triangle, after density normalization to $[0,1]$. As this approach represents the analytic solution to the mathematical model of continuous parallel coordinates, it may also be considered as ground truth for comparison purposes. The fragment program used for the examples in this paper is available as supplemental material.

4.2 Numerical Integration

Given a 2-D scalar density field, the gathering approach presented earlier accumulates densities for each $\boldsymbol{\eta}$ along the dual line $L_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}$ in the data domain. In our implementation, we use the continuous reference triangle with densities stored in a floating-point render target to compute line integrals according to the gathering approach. Density values for parallel coordinates are stored in a floating-point render target of the same resolution. Then, for each texel in the parallel-coordinates domain, the dual line is sampled from the input field. In order to properly reconstruct σ, the sampling rate was set to the respective Nyquist rate. Due to the texel-based computation, the algorithm is perfectly suited for hardware-accelerated computation. As there is no visible difference to ground truth, we computed the l^2 norm of the difference vector of the respective render targets to obtain a quantitative distance measure. After normalization, the relative distance, i.e. $\frac{l^2}{N}$ with $N = 2048^2$, of the gathering approach to ground truth is approximately $1.2 \cdot 10^{-7}$. The error is negligible and, therefore, the gathering approach is an appropriate alternative to the analytic solution. The sources of the small difference between the numerical and the analytic solution include the sampled representation of the scatterplot, the numerical integration, and the interpolation when accessing the data domain. All these error sources depend on the resolution of the data-domain representation. Therefore, the quality of the numerical solution can be controlled by adapting the resolution of the intermediate scatterplot texture. In contrast to the analytic solution using triangulated data, the gathering approach does not depend on the size of the dataset, such that it may be used in a fast, although less accurate, implementation for the computation of continuous parallel coordinates. Note that, for the efficient rendering of continuous scatterplots, Bachthaler and Weiskopf [3] recently proposed adaptive techniques supporting a wide class of reconstruction filters, including trilinear interpolation. The fragment program used to compute 2-D continuous parallel coordinates from a continuous scatterplot texture is available as supplemental material.

A scattering approach was implemented according to the generic scattering algorithm presented in Section 3.3. Samples are drawn randomly on a triangle in the data domain using rejection sampling, i.e. observations are sampled from the surrounding rectangle, rejecting samples outside the triangle and linearly interpolating those accepted. Then, for each sample $\boldsymbol{\xi}_i$, the dual line $L_{\boldsymbol{\eta}}^{\boldsymbol{\xi}_i}$ in parallel coordinates is rendered as a white polyline with density being represented by the respective alpha value (i.e. $\alpha \leftarrow \sigma(\boldsymbol{\xi}_i)$). The overall density $\varphi(\boldsymbol{\eta})$ according to (7) is obtained by accumulating alpha values of each line intersecting $\boldsymbol{\eta}$, which is conveniently implemented using additive blending. After normalizing, the resulting image is finally low-pass filtered

using a Gaussian 5×5 kernel in order to compensate for aliasing artifacts. Again, there is no visible difference to ground truth. The relative l^2 difference to ground truth is approximately $2.75 \cdot 10^{-6}$, i.e. about one order of magnitude higher than for the gathering approach. This could be further improved by increasing the number of samples.

5 RESULTS

In this section, we compare discrete density-based and continuous parallel coordinates for a typical scientific visualization dataset. Further examples are available as supplemental material. Discrete parallel coordinates are created by drawing one polyline for each sample in the spatial domain. For continuous parallel coordinates, a 2-D density field is computed using the projected tetrahedra algorithm [2]. The resulting triangles in the data domain are then mapped to parallel coordinates as described in Section 4.1. In both approaches, a render-target texture is used to obtain floating-point precision for the computation of densities. In the case of discrete parallel coordinates, the density of a pixel is computed by counting the lines crossing that pixel. Before the content of the texture is written to the framebuffer, the densities are normalized to the same density range. Furthermore, we apply a logarithmic colormap to the normalized densities, such that low densities are shown in black/dark-blue, mid-density values are shown in red, and high-density values are mapped to yellow/white.

Figure 5 illustrates discrete and continuous 4-D parallel coordinates of the IEEE Visualization 2004 contest dataset "hurricane Isabel". The original data consists of 48 timesteps, each containing measurements of 11 attributes with a spatial resolution of $500 \times 500 \times 100$. For our comparison, we use the first timestep and four dimensions in three different spatial resolutions (original, and downsampled to $50 \times 50 \times 10$ and $100 \times 100 \times 20$). The visualized dimensions are the vertical spatial position (height), temperature, pressure, and wind velocity. Both temperature and pressure are contained in the original dataset, whereas wind velocity is computed from wind speed in x-, y-, and z-direction. Every dimension was normalized independently to the range $[0,1]$ before computation. Furthermore, tetrahedra containing invalid attribute data such as N/A-values were discarded.

The most prevalent character of the series of standard parallel coordinates in Figure 5 is the increasing amount of clearly visible clusters resulting from the discrete mapping of the vertical spatial coordinate (height). Only at high resolutions the true character of the first dimension can be revealed, indicating a linearly increasing function defined on a continuous domain. But, if only one of the plots were available, it could falsely be interpreted as a set of high-dimensional clusters with equal values on the first dimension. Continuous parallel coordinates do not suffer from this problem, as linear interpolation of values is inherently contained in the density model. This can nicely be seen in Figure 5, where the equal distribution of samples on the first dimension can already be observed at low resolutions. Note that this is a key information which is entirely missing in discrete parallel coordinates.

We observe that continuous parallel coordinates of low-resolution data rapidly converge to ground truth, i.e. plots computed from full-resolution data. In order to obtain a numerical measure for similarity,
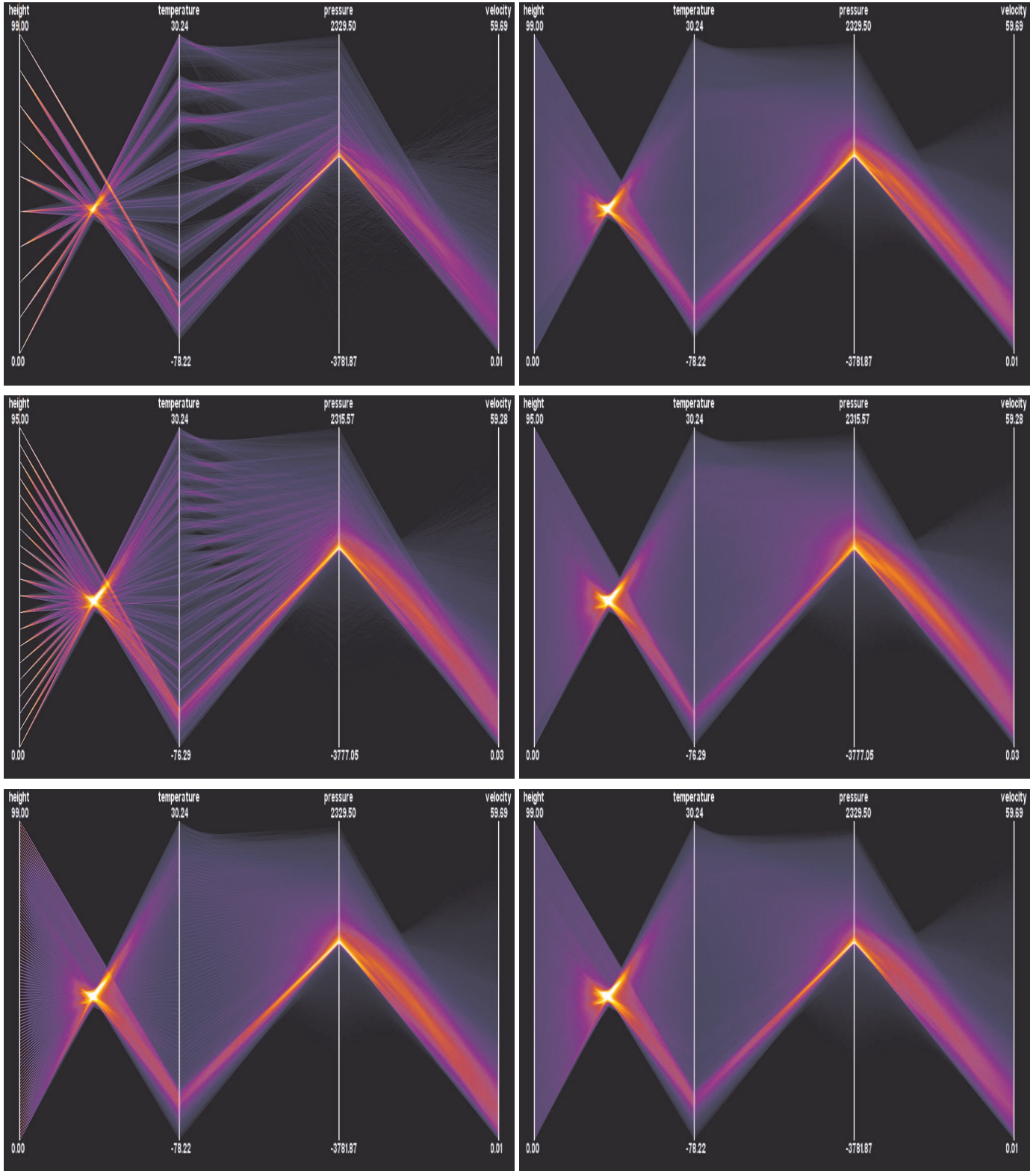
Fig. 5. Discrete and continuous parallel coordinates for the "hurricane Isabel" dataset at different spatial resolutions ($50 \times 50 \times 10$, $100 \times 100 \times 20$, $500 \times 500 \times 100$ from top to bottom). On the left side, discrete parallel coordinates are shown with the corresponding continuous version on the right side. Sampling artifacts stemming from the discrete mapping of the vertical spatial coordinate (height) lead to misrepresentation of key information in discrete parallel coordinates.
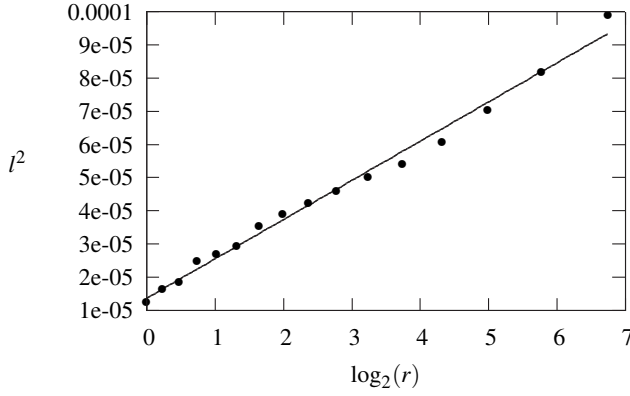
Fig. 6. Relation of the relative l^2 distance with spatial sampling rate $r = \frac{500 \cdot 500 \cdot 100}{n_x \cdot n_y \cdot n_z}$, where n_i denotes the number of samples in dimension i. Both l^2 as well as r are given relative to ground truth, i.e. to the full-resolution data set. In order to accentuate the exponential relation, a linear regression line in the logarithmic plot was computed.

the l^2-norm of the difference of density for different spatial sampling rates to the original dataset was computed with floating-point precision (Figure 6). The results show that difference decreases exponentially with increasing spatial sampling resolution. Furthermore, the largest l^2 value of 1e-04 is still very small, emphasizing that the main information contained in the data is already captured by low-resolution plots.

A performance comparison of discrete and continuous parallel coordinates is provided in Table 1. Although the gathering approach allows for highly interactive computation of continuous parallel coordinates while being independent of the spatial resolution, it depends on the computation of continuous scatterplots, which make up most of the total time needed to compute the final plot. More efficient rendering techniques have been proposed recently by Bachthaler and Weiskopf [3] and may be used to accelerate our approach as well.

6 CONCLUSION AND FUTURE WORK

We have presented continuous parallel coordinates for multi-variate data defined on a continuous domain. The construction of such a high-dimensional density field relies on the concept of two-dimensional continuous scatterplots that are mapped to the parallel-coordinates system using point–line duality. We have derived a mathematical density model based on mass conservation during the mapping from spatial to data and parallel-coordinates domains. The consecutive application of this mapping allows for an arbitrary number of data dimensions. Different numerical integration techniques for the computation of the density model have been presented. We have shown that both gathering and scattering techniques can be used for the approximation of density in parallel coordinates. For triangulated data, an analytic solution has been provided.

An important benefit of continuous parallel coordinates is that typical sampling artifacts do not occur. Distracting patterns are removed

which are not contained in the data, but emerge from the dependency of discrete parallel coordinates on the sampling rate in the spatial domain. In contrast, continuous parallel coordinates are largely independent of the resolution: plots generated from low-resolution data are very similar to the full-resolution version. However, the accuracy of the plots from coarsened data depends on the interpolation function used in the reconstruction step. Hence, the algorithm presented in section 4.1 using linear interpolation will therefore produce less accurate results for higher-order characteristics.

This behavior demonstrates the fundamental aggregation character of density-based parallel coordinates. Like other statistical visualization techniques, such as histograms, this approach is robust under sampling effects and other external influences, capturing the essence of a dataset. It is important to note that although sparse data probably benefits most from our method, sampling artifacts can also occur from high-resolution data which are guaranteed to be removed by continuous parallel coordinates. Another practical advantage of continuous parallel coordinates is the scalability with increasing data set size: the overplotting problem is avoided without the need for parameters such as bucket-size or any other density approximation technique.

Apart from differences regarding the sampling of the data, however, continuous parallel coordinates share most of the advantages and problems of discrete parallel coordinates. Many of the improvements and extensions to parallel coordinates presented in recent work can thus be applied to continuous parallel coordinates without restrictions. For instance, parallel sets could be used in conjunction with continuous parallel coordinates in order to join both categorical and continuous variables in a single plot. In principle, interactive techniques such as brushing are also applicable to continuous parallel coordinates. Smooth brushing [10] is particularly interesting for continuous data representations, as a density gradient can directly be obtained from the plots. However, methods depending on individual lines such as angular brushing [13] cannot be used.

In the limit process, continuous parallel coordinates share the same visual signature with classic density plots, where the characteristics of parallel coordinates are fully captured but single lines cannot be perceived. Using brushing, however, the line structure of discrete parallel coordinates can be reconstructed in a controlled manner by sampling the continuous version.

In future work, further application areas could be explored and the usefulness of our visualization technique could be investigated by application-oriented studies. We expect that applications with large scientific data sets might benefit most from continuous parallel coordinates. Other aspects of future work could include investigating analytic solutions to the computation of density for non-triangulated data and non-linear interpolation schemes using continuous scatterplots [3] and direct mapping of datasets from the spatial domain. The efficiency of rendering parallel coordinate plots could be improved for the analytic solution by porting the geometry computations to the GPU and for numerical integration by incorporating hierarchical and adaptive techniques for the rendering of continuous scatterplots [3]. Finally, the investigation of interactive, density-based brushing techniques is an important task to be conducted in the future.

APPENDIX

This section provides the derivation of (7), the line density of a point $\boldsymbol{\eta}$ in parallel coordinates.

Assuming mass conservation, the mass M of the interval Ω in the parallel-coordinates domain and the area Φ in the data domain must be equal (see Figure 2)

$$M = \int_{\Omega} \varphi(\boldsymbol{\eta}) \mathrm{d}\eta_2 = \int_{\Phi} \sigma(\boldsymbol{\xi}) \mathrm{d}^2 \xi \qquad (20)$$

Applying the fundamental theorem of calculus yields

$$\varphi(\eta_1, \eta_2) = \frac{\mathrm{d}M}{\mathrm{d}\eta_2} = \frac{\mathrm{d}}{\mathrm{d}\eta_2} \int_{\Phi} \sigma(\boldsymbol{\xi}) \mathrm{d}^2 \xi \qquad (21)$$

Now, the integration domain Φ is split in two perpendicular directions

Table 1. Computation time in ms for continuous scatterplots (CS), continuous parallel coordinates (CPC), and discrete parallel coordinates (PC) for different resolutions of the hurricane Isabel dataset. The measurements were conducted on a Linux PC with an Intel(R) Core(TM) 2 Quad CPU running at 2.4 GHz with 4 GB RAM and an NVIDIA GeForce 8800 GTX graphics card.

	$50 \times 50 \times 10$	$100 \times 100 \times 20$	$500 \times 500 \times 100$
CS	5 022	7848	661 164
CPC	5	4	4
PC	10	80	36 631

$\Phi_{\parallel}$ and $\Phi_{\perp}$. For this purpose, we define a rotation $v : \mathbb{R}^2 \longrightarrow \mathbb{R}^2, \boldsymbol{\xi} \mapsto v(\boldsymbol{\xi})$ that maps the unit vector $\boldsymbol{\xi_2}$ to $\mathbf{n} = \frac{\tilde{\mathbf{n}}}{||\tilde{\mathbf{n}}||}$:

$$v(\boldsymbol{\xi_2}) = \tilde{\boldsymbol{\xi_2}} = \mathbf{n} \tag{22}$$

Now, the transformation theorem for integrals can be applied to (20):

$$\int_{v(\phi)} \sigma(\tilde{\boldsymbol{\xi}}) \mathrm{d}^2 \tilde{\xi} = \int_{\phi} \sigma(v(\boldsymbol{\xi})) |\det(Dv(\boldsymbol{\xi}))| \mathrm{d}^2 \xi \tag{23}$$

where D denotes the respective Jacobian matrix. Note that, in our case, $|\det(Dv(\boldsymbol{\xi}))| = 1$. Now, splitting the region $v(\phi) = \phi_{\parallel} \otimes \phi_{\perp}$ remains only a matter of splitting integrals:

$$M = \int_{\phi_{\parallel}} \left[\int_{\phi_{\perp}} \sigma(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{\xi}_2 \right] \mathrm{d}\tilde{\xi}_1 \tag{24}$$

For the computation of the density follows:

$$\varphi(\eta_1, \eta_2) = \int_{\phi_{\parallel}} \left[\frac{\mathrm{d}}{\mathrm{d}\eta_2} \int_{\phi_{\perp}} \sigma(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{\xi}_2 \right] \mathrm{d}\tilde{\xi}_1 \tag{25}$$

In order to transform the integration along $\phi_{\perp}$ to an integration over Ω, we use that

$$\frac{\mathrm{d}D_{\boldsymbol{\xi}}}{\mathrm{d}\eta_2} = \frac{1}{||\tilde{\mathbf{n}}||} \tag{26}$$

which is a result of (3). Then, the inner integral of (24) yields the desired transformation to the parallel-coordinates domain:

$$\int_{\phi_{\perp}} \sigma(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{\xi}_2 = \int_{\Omega} \frac{\sigma(\tilde{\xi}_1, D_{\boldsymbol{\xi}}(\eta_2))}{||\tilde{\mathbf{n}}||} \mathrm{d}\eta_2 \tag{27}$$

With (25), the density in the parallel-coordinates domain then becomes:

$$\varphi(\eta_1, \eta_2) = \int_{\phi_{\parallel}} \frac{\sigma(\tilde{\xi}_1, D_{\boldsymbol{\xi}}(\eta_2))}{||\tilde{\mathbf{n}}||} \mathrm{d}\tilde{\xi}_1 \tag{28}$$

Returning to the original coordinate system finally describes the line density in a point $\boldsymbol{\eta}$ of the parallel-coordinates system by integrating over the corresponding line in the data domain:

$$\varphi(\eta_1, \eta_2) = \int_{L_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}} \frac{\sigma(\boldsymbol{\xi}(\lambda))}{||\tilde{\mathbf{n}}||} \mathrm{d}\lambda \tag{29}$$

with $\mathbf{L}_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}(\lambda)$ being the arc-length parametrized line $L_{\boldsymbol{\xi}}^{\boldsymbol{\eta}}$.

ACKNOWLEDGMENTS

REFERENCES

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

[2] S. Bachthaler and D. Weiskopf. Continuous scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1428–1435, 2008.

[3] S. Bachthaler and D. Weiskopf. Efficient and adaptive rendering of 2-D continuous scatterplots. *Computer Graphics Forum*, 28(3):743–750, 2009.

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

[5] F. Bendix, R. Kosara, and H. Hauser. Parallel sets: Visual analysis of categorical data. In *IEEE Symposium on Information Visualization*, pages 133–140, 2005.

[6] M. R. Berthold and L. O. Hall. Visualizing fuzzy points in parallel coordinates. *IEEE Transactions on Fuzzy Systems*, 11:369–374, 2003.

[7] J. Blaas, C. Botha, and F. Post. Extensions of parallel coordinates for interactive exploration of large multi-timepoint data sets. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1436–1451, 2008.

[8] H. Carr, B. Duffy, and B. Denby. On histograms and isosurface statistics. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1259–1266, 2006.

[9] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *IEEE Symposium on Visualization*, pages 239–248, 2003.

[10] H. Doleisch and H. Hauser. Smooth brushing for focus+context visualization of simulation data in 3D. *Journal of WSCG*, pages 147–155, 2002.

[11] G. Ellis and A. Dix. Enabling automatic clutter reduction in parallel coordinate plots. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):717–724, 2006.

[12] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *IEEE Visualization*, pages 43–50, 1999.

[13] H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing for extended parallel coordinates. In *IEEE Symposium on Information Visualization*, pages 127–130, 2002.

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

[15] A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *IEEE Visualization*, pages 361–378, 1990.

[16] A. Inselberg and B. Dimsdale. Multidimensional lines II: Proximity and applications. *SIAM Journal on Applied Mathematics*, 54(2):578–596, 1994.

[17] J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure within clustered parallel coordinates displays. In *IEEE Symposium on Information Visualization*, pages 125–132, 2005.

[18] K. T. 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 *Computing and Graphics in Statistics*, pages 107–123. Springer, New York, 1991.

[20] P. Muigg, J. Kehrer, S. Oeltze, H. Piringer, H. Doleisch, B. Preim, and H. Hauser. A four-level focus+context approach to interactive visual analysis of temporal features in large scientific data. *Computer Graphics Forum*, 27(3):775–782, 2008.

[21] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM (Society for Industrial and Applied Mathematics), Philadelphia, 1992.

[22] 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.

[23] J. F. Rodrigues, Jr., A. J. M. Traina, and C. Traina, Jr. Frequency plot and relevance plot to enhance visual data exploration. In *Computer Graphics and Image Processing*, pages 117–124, 2003.

[24] C. E. Scheidegger, J. Schreiner, B. Duffy, H. Carr, and C. T. Silva. Revisiting histograms and isosurface statistics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1659–1666, 2008.

[25] P. Shirley and A. Tuchman. A polygonal approximation to direct scalar volume rendering. *Computer Graphics*, 24(5):63–70, 1990.

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

[27] E. Wegman and Q. Luo. High dimensional clustering using parallel coordinates and the grand tour. *Computing Science and Statistics*, 28:361–368, 1997.

[28] H. Zhou, X. Yuan, H. Qu, W. Cui, and B. Chen. Visual clustering in parallel coordinates. *Computer Graphics Forum*, 27(3):1047–1054, 2008.