State of the Art of Parallel Coordinates

J. Heinrich[1] and D. Weiskopf[1]

[1]Visualization Research Center, University of Stuttgart

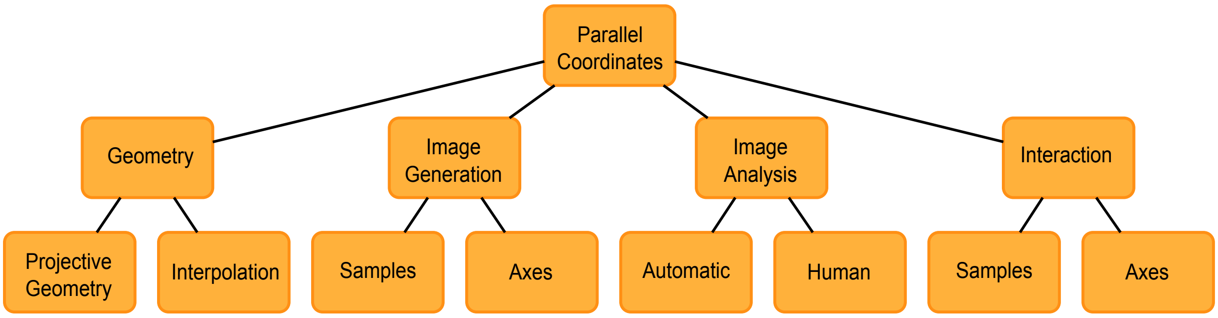


Figure 1: Taxonomy of topics for parallel coordinates in the scientific literature. The first-level nodes each represent a section in this paper, where the scope and definition of each topic will be explained.

Abstract

This work presents a survey of the current state of the art of visualization techniques for parallel coordinates. It covers geometric models for constructing parallel coordinates and reviews methods for creating and understanding visual representations of parallel coordinates. The classification of these methods is based on a taxonomy that was established from the literature and is aimed at guiding researchers to find existing techniques and identifying white spots that require further research. The techniques covered in this survey are further related to an established taxonomy of knowledge-discovery tasks to support users of parallel coordinates in choosing a technique for their problem at hand. Finally, we discuss the major challenges in constructing and understanding parallel-coordinates plot and provide some examples from different application domains.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Parallel coordinates is a widely used visualization technique for multivariate data and high-dimensional geometry. Since their first appearance in the scientific literature in the context of Nomography [Mau85], parallel coordinates have become a well-known visualization for exploratory data analysis [Weg90] and visual multidimensional geometry [Ins09]. The theory of parallel coordinates has been developed rigorously and the point–line duality has been successively generalized to higher dimensions [Ins85]. There are many visualizations that are related to parallel coordinates either by sharing the typical parallel layout of axes or the mapping of data samples to lines, as in stock-market diagrams, temperature forecasts, *N and M plots* [DF83] or *Andrews plots* [And72]. The relation to such familiar diagramming techniques is certainly one of the reasons of the rising popularity of parallel coordinates: the number of publications with the term "parallel coordinates" in the title has been rising steadily from 14 in the year 1991 to approximately 543 in 2011, with a total of 5620 publications as reported by Google scholar on the 15th of December, 2012.

This paper presents a survey of recent developments of parallel coordinates with a strong focus on visualization techniques and is aimed to complement Inselberg's textbook [Ins09], which represents the state-of-the-art of parallel-coordinates theory. The contributions of this work are

- A taxonomy and survey of techniques with respect to modeling, visualizing, understanding, and interacting with parallel coordinates.
- A classification of common tasks in knowledge discovery with respect to our taxonomy.
- A discussion of the challenges for visualizing parallel coordinates.
- A pointer to the literature for the aspects covered by the taxonomy.
- An overview of applications of parallel coordinates in various domains from the life sciences and engineering.

Note that we deliberately retain from comparing the visualization techniques presented here with other methods (including the original parallel coordinates) nor do we evaluate or validate the methods with respect to performance or applicability to real data, as this would be out of the scope of this state-of-the-art report. The intend of this work is to give an overview of existing visualization techniques for parallel coordinates and to provide pointers into the literature for further information.

The taxonomy given in Figure 1 was established from the scientific literature about various topics regarding parallel coordinates. It is targeted at identifying research directions and providing a classification scheme at different levels of abstraction. This is helpful as a guide for (i) scientists to identify areas that require further research and for (ii) users of parallel coordinates to provide an overview of available techniques and possible challenges. At the top-level, we distinguish between *geometric models* as the theoretical foundation of parallel coordinates from the more technical parts dedicated to *image generation* and *image analysis*.

In addition to the taxonomy, we identified a set of challenges a user might be faced with when working with parallel coordinates. We summarize these challenges and provide links to the sections of this work and to the literature in order to address them. Finally, we present a set of selected applications by domain to give examples of the wide range of data types that have been visualized with parallel coordinates.

We use an established taxonomy [FPSS96a] to relate the techniques covered in the following sections to a set of high-level tasks that support the knowledge-discovery in databases (KDD).

Classification is the task of mapping data samples to a set of predefined classes. A typical technique in interactive visualization environments that supports the classification of samples is brushing (Section 5.1.1). Brushing is typically used to select data points which are then subject to further processing, such as learning a classifier [AA99, TFA*11].

Regression is a common task for predicting the values of a dependent variable with respect to one or more independent variables. Parallel coordinates can be used for "visual regression" [WL97] or to visualize statistical properties of regression models [UVW03, DHNB09, SSJKF09].

Clustering is the identification of sets of data items exhibiting similar characteristics. There is a wide range of automatic clustering techniques which typically depend on the similarity measure being used. Parallel coordinates can be used for "visual clustering", i.e. to find groups of similar points based on visual features such as the proximity of lines or line density. Another application for which parallel coordinates are frequently used is the visualization of precomputed clusters and their characteristics, typically using color or geometry-based visual cues.

Summarization refers to the computation of aggregated data and usually involves loss of information. Visualization is considered a summarization technique in KDD because it requires multivariate data to be projected to two dimensions. From a visualization viewpoint, the presentation of an overview is what probably best describes the summarization task. This is an important task and the starting point of the information-seeking mantra [Shn96]. There are many approaches to show aggregated information in parallel coordinates, either as additional visual items, or by representing sets of items using alternative visual encodings such as envelopes of lines [Ins09] or density [MW91, HW09] (see Section 3.1).

Dependency-modeling is the process of establishing qualitative or quantitative dependencies between variables. Linear correlation between two variables is the most common dependency that can be visualized in parallel coordinates as a result of the point–line duality. The quantification of dependencies is an important measure for determining the relative importance of dimensions that can be used to order axes in parallel coordinates. The axis-ordering problem is discussed in more detail in Sections 3.2, 5.2, and 6.2.

Change and deviation detection includes the detection and visualization of outliers or other anomalies of the data with respect to some previously known measure. For example, data samples can be classified as outliers using a density estimation [NH06] based on parallel coordinates of the raw data. The detection of abnormal behavior using parallel coordinates is also an important task in process control applications [DEN12].

2. Geometry

A coordinate system provides a scheme for locating points given its coordinates and vice versa. The choice of coordinate system is therefore an important step in visualizing data,

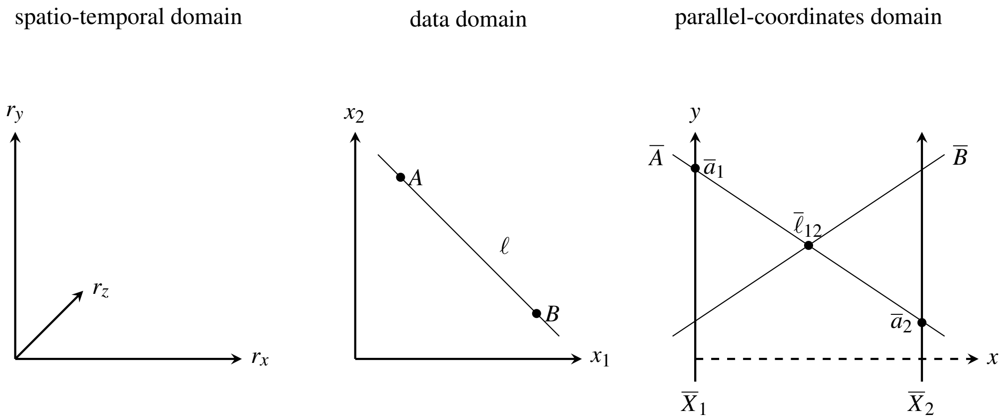


Figure 2: The notation used for different domains. The spatio-temporal domain (left) describes events in space and time with up to 4 dimensions. Many datasets describing abstract data are defined in the data domain (middle) with a finite number of dimensions. The parallel-coordinates domain (right) refers to the xy-plane in Cartesian coordinates which is used to construct a parallel-coordinates system. See Section 2 for an explanation of the point–line duality illustrated above.

as it transforms the geometry representing the data that is being visualized. With coordinate transformations, straight lines (e.g. in Cartesian coordinates) can be mapped to curves (e.g. in polar coordinates) or to points (e.g. in parallel coordinates). The choice of coordinate system determines the patterns exhibited by a visualization to a large part and therefore it is important to know how to "read" it. After introducing the notation used in this work, the construction of parallel coordinates is briefly described and two models that can be used for the transformation of data points from Cartesian coordinates to parallel coordinates are discussed.

Parallel coordinates can be used to visualize geometry that represents data in multiple domains. Here, the term "domain" is used as a synonym for the domain of a function, i.e. the set of values for which a function is defined. Some domains will be used frequently and are thus assigned a meaningful name as well as consistent labels to help the reader connect a symbol used in an equation to the respective domain. The notation of Inselberg [Ins09] is adopted to distinguish between Cartesian and parallel coordinates with respect to the following domains (see also Figure 2):

- The **spatio-temporal domain** represents the set of four-dimensional real values $\mathbb{R}^4$ describing events in space and time as well as any projection thereof to lower-dimensional subspaces (such as time only). Events are represented by data points referred to as *spatial*, *temporal*, or *spatio-temporal* data. A point $P = (r_x, r_y, r_z, t) \in \mathbb{R}^4$ is denoted using xyz-coordinates plus t for the time-dimension. Lines and curves are denoted with lowercase letters. The vector $\mathbf{p} = (r_x, r_y, r_z, t)^T$ is also lowercase with bold typeface.
- The **data domain** represents the set of N-dimensional real values $\mathbb{R}^N$, $N \in \mathbb{N}^+$. Data defined in the data domain usually depicts *non-spatial* or *abstract* data such as *observations* drawn from random variables. The position of points $X = (x_1, x_2, ..., x_N)$ in the data domain is determined using indexed coordinates, as N may take any natural number greater than zero. Unless stated otherwise, indexed lowercase letters denote the respective *dimension*, such that x_1 refers to the first dimension of the data domain. For lines, curves, and vectors, the same notation as for the spatio-temporal domain is used.
- The **parallel-coordinates domain** is represented by the xy-plane in $\mathbb{R}^2$. It is of special interest as its representation in Cartesian coordinates enables the construction of parallel coordinates, for which it forms the *embedding coordinate system*. The representation of a point $\overline{\ell} = (x, y)$ in the parallel-coordinates domain therefore uses only the x and y coordinates of the spatio-temporal domain. Note that lowercase letters with a bar refer to points while capital letters with a bar denote lines. This notation was proposed by Inselberg [Ins09] in order to emphasize the *dualities* between data domain and parallel-coordinates domain.

Note that many datasets in data mining and statistics are described exclusively by points in the data domain, as they have no spatial or temporal embedding. Examples are car statistics, credit card transactions, etc.

2.1. Constructing Parallel Coordinates

Parallel coordinates are constructed by placing axes in parallel with respect to the embedding 2D Cartesian coordinate system in the plane (the parallel-coordinates domain). While the orientation of axes can be chosen freely, the most common implementations use horizontal (parallel to the x-axis) or vertical (parallel to the y-axis) layouts. The choice of lay-

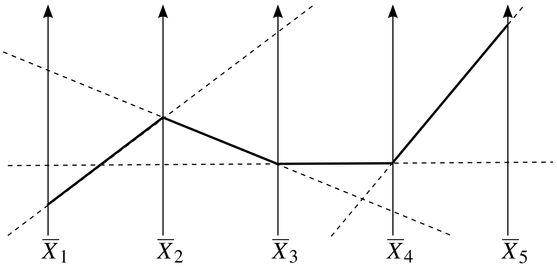


Figure 3: Constructing parallel coordinates with five dimensions represented by $N = 5$ vertical lines. Points in the plane are represented by lines joining the corresponding coordinates at the respective axes. Typically, only the line segments between the axes are drawn (represented by the bold polyline).

out depends on the number of axes, the range of the data, the dimensions of the screen, and the personal preference. For reasons of simplicity and consistency, vertical axes will be used throughout this document unless stated otherwise. For N-dimensional geometry, this results in N copies of the y-axis

$$\overline{X}_i : x = d_i, i = 1,2,..,N$$

where the N-vector $\mathbf{d_N} = (d_1, d_2, ..., d_i, ..., d_N)^T$ is used to denote the axis spacing as the distance of the i-th axis to the y-axis at $x = 0$. With this setting, $\frac{N(N-1)}{2}$ *pairs of axes* are obtained which will also be referred to as *segments*. Note that for a given $\mathbf{d_N}$, there are $N-1$ *adjacent* pairs of axes, as illustrated in Figure 3. For a discussion of the order to choose for the axes, please refer to Sections 3.2 and 6.2.

2.2. Projective Plane Model

The *point–line duality* in the plane [Ins85] is only briefly summarized here. A more detailed description including analytic proofs and the representation of hyperplanes and p-flats in $\mathbb{R}^N$ are given elsewhere [Ins85, ID90, Weg90, Ins09].

For $N = 2$, let $\mathbf{d_2} = (0, d)$ describe a two-dimensional parallel-coordinates system as in Figure 2. Then, a point $A = (a_1, a_2) \in \mathbb{R}^2$ of the corresponding data domain is represented in parallel coordinates by the line joining $(0, a_1)$ and (d, a_2)

$$\overline{A} : y = \frac{a_2 - a_1}{d} x + a_1, d \neq 0. \tag{1}$$

A set of points all located on the line

$$\ell : x_2 = m x_1 + b$$

is represented by a set of lines in parallel coordinates that intersect at the *indexed point*

$$\overline{\ell}_{12} : \left(\frac{d}{1-m}, \frac{b}{1-m} \right), m \neq 1.$$

Here, indexes denote axes or dimensions, and $\overline{\ell}_{ij}$ is a point in the $\overline{X}_i\overline{X}_j$ coordinate system. Similarly, points $\overline{p}_i$ with a single index are always located *on* the corresponding axis $\overline{X}_i$. For the sake of clarity, indexes will be omitted if the corresponding dimensions are obvious from the context, in particular for discussions of two-dimensional parallel-coordinate systems.

Note that the horizontal position of $\overline{\ell}$ only depends on the axis spacing and the slope of ℓ. For the common case $d > 0$, $\overline{\ell}$ is located

- left of $\overline{X}_1$ if $m > 1$
- right of $\overline{X}_2$ if $1 > m > 0$ and
- between $\overline{X}_1$ and $\overline{X}_2$ if $0 > m$.

So far, this formulation provides a mapping of points to lines and vice versa for all lines in the data domain with $m \neq 1$ and for all lines in the parallel-coordinates domain that are not vertical, such as the axes. In order to resolve those special cases and complete the duality, both the data domain and the parallel-coordinates domain are considered *projective planes* $\mathbb{P}^2$ that allow us to map the line $\ell : x_2 = x_1 + b$ with $m = 1$ in the data domain to the *ideal point* $\overline{\ell}_\infty$ in parallel coordinates where the set of parallel lines with slope b/d intersect. Likewise, the vertical line $\overline{P}_m^\infty : x = \frac{d}{1-m}$ in parallel coordinates maps to the set of parallel lines (or the ideal point) P_m^∞ with slope m in the data domain. Figure 4 illustrates ideal points in both domains.

Based on the point–line duality, other mappings can be expressed using the *envelope* of lines in parallel coordinates. For example, $\overline{\ell}$ is the envelope of all intersecting lines and is dual to the line ℓ as shown above. Inselberg further uses

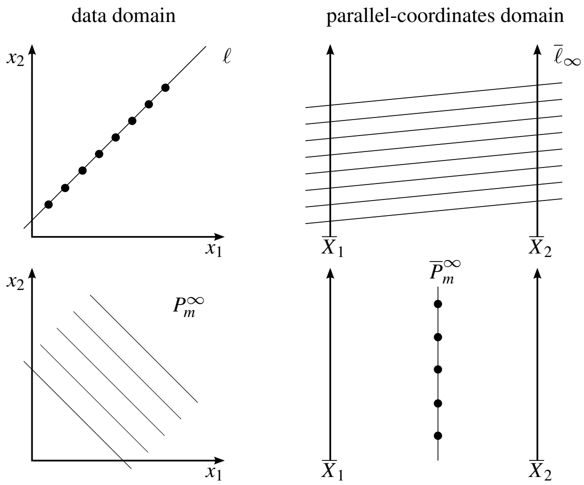


Figure 4: The line with slope $m = 1$ in the data domain is mapped to the ideal point $\overline{\ell}_\infty$ in parallel coordinates (top). The vertical line $\overline{P}_m^\infty : x = \frac{d}{1-m}$ in parallel coordinates is represented by the ideal point P_m^∞ with slope m in the data domain. Both domains are considered projective planes.

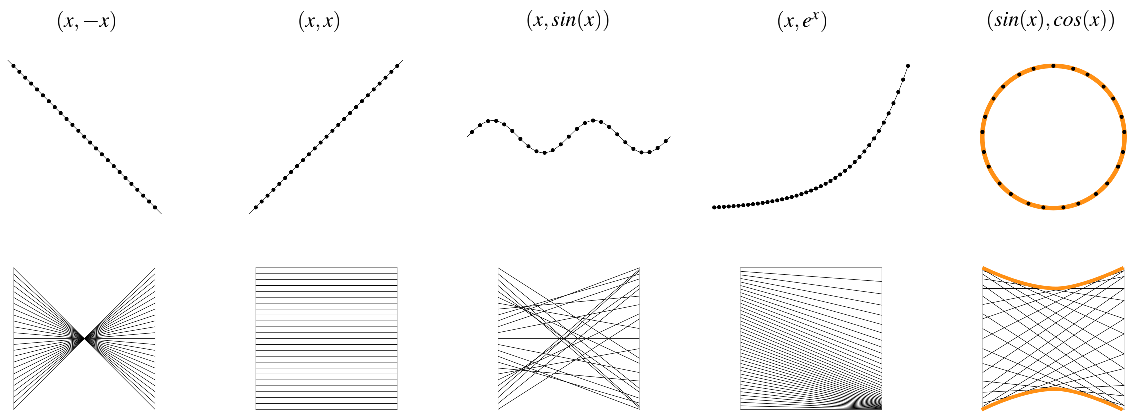


Figure 5: Common patterns in Cartesian coordinates (top) and their dual representation in parallel coordinates (bottom). The envelope of lines is highlighted for the ellipse–hyperbola duality.

envelopes to establish a *curve–curve duality* between Cartesian and parallel coordinates. Here, a curve c is mapped point-wise from the data domain to lines in the parallel-coordinates domain resulting in the *line-curve* $\bar{c}$. The envelope of the line-curve now describes a point-curve in parallel coordinates. As an example, ellipses in Cartesian coordinates are mapped to hyperbolas in parallel coordinates, as can be seen in Figure 5. The ellipse–hyperbola duality has implications for the visualization of Gaussian distributions [MW91, FKLI10, HBW11] in parallel coordinates.

Another duality that has implications for brushing (see Figure 10, page 11) is the *rotation–translation* duality. Translating a point in parallel coordinates along the x-axis changes the slope of its dual line in the data domain, and vice versa. Similarly, rotating a line in parallel coordinates about a point results in the dual point to move along the line dual to the point of rotation. Please refer to Inselberg [Ins09] for details.

2.3. Interpolation Model

Given N parallel axes, the polyline that is typically used to represent a point $A \in \mathbb{R}^N$ can also be obtained using a piecewise linear interpolation of the respective indexed points $\bar{a}_i, i = 1,2,...,N$ located on the axes. For example, the line $\bar{A}$ in Figure 2 can be computed by linearly interpolating the points $\bar{a}_1$ and $\bar{a}_2$.

In analogy to Section 2.2, let $N = 2$ and $\mathbf{d_2} = (0,d)^T$. Then, Equation (1) for the representation of a point $A = (a_1,a_2)$ in parallel coordinates can also be written as

$$\bar{A} : y = \frac{1-x}{d} a_1 + \frac{x}{d} a_2, x \in [0,d].$$

The interpolation model allows for a wide range of different visual mappings from points in Cartesian coordinates to lines and curves in parallel coordinates, as any scheme that interpolates the indexed points $\bar{p}_i$ at the axes can be employed (see Section 3.1.3 for an overview on curves). For example, the interpolation model with linear interpolation can be used to produce the same patterns as in Figure 5 and it can be shown that a line in Cartesian coordinates is always mapped to a point in parallel coordinates, regardless of the interpolation model applied [Mou09]. See Moustafa [Mou11] and references therein for a more detailed discussion of the interpolation model and its properties.

3. Image Generation

For multivariate data with $N > 2$, N axes are placed in parallel as described in Section 2.1. Applying the point–line duality to a N-dimensional point for every adjacent pair of axes results in $N-1$ lines (dashed in Figure 3), each representing a projection of the point to the corresponding plane. Restricting the mapping to segments results in a *polyline* intersecting all axes at the respective coordinates (bold in Figure 3) and constitutes the most common visualization for N-dimensional points in parallel coordinates. In terms of the visualization pipeline [HM90], the dashed-line representation and the polyline-representation constitute different geometric mappings. Further mapping and rendering techniques for *image generation* are presented in this section.

Many parallel-coordinate visualizations are composed of several *layers*, each of which may be computed independently. While we could consider using one layer for every line or geometric object, we will distinguish only two main layers here: one layer for the *data points* (which are typically mapped to polylines) and one for the axes. Other frequently used layers are

- brushes or any other object used for interaction with the plot,
- axis overlays such as boxplots or ellipses,
- any other geometry that is mapped to the final image.

A *parallel-coordinates system* is usually visualized using the axis layer only. A *parallel-coordinates plot* is a visualization of the sample layer with optional axis layer. A *composite parallel-coordinates plot* is a parallel-coordinates plot with any additional layer as described above.

In the following, different mapping and rendering approaches for the two main layers are described.

3.1. Samples

This section discusses various visual encodings in the parallel-coordinates domain for N-dimensional data points (defined in the data domain). It is important to note that the geometric mappings presented in the following are the objects used for visualization in the final parallel-coordinates plot and do not refer to objects in the N-dimensional data domain. For a discussion of the representation of multidimensional lines, planes, p-flats, curves, etc. in parallel coordinates, please refer to the respective chapters in Inselberg's book [Ins09]. Also note that, with some exceptions, most of the mappings are constructed using one of the models described in Section 2.

The following subsections describe two fundamentally different approaches for the visualization of a set of data points. **Geometry-based** approaches use geometric objects such as points, lines, curves, or polygons as a mapping for individual data samples or groups of samples. The analysis task thereby varies from the visualization of correlation over the detection of outliers to the characterization of clusters over multiple dimensions, among others.

Density or density estimates of the input data can be visualized implicitly or explicitly. Implicit density visualizations are based on the proximity of geometric objects. Depending on the sample size and the shape of the (true, but typically unknown) distribution, geometry-based visualizations represent both the raw data and the respective density or density estimate. Due to the potential overlap of visual items, however, these approaches may fail to convey useful information, in particular if the data is very large. In contrast, **density-based** approaches explicitly visualize a continuous density function of the underlying data instead of discrete samples. Figure 6 illustrates examples of explicit density visualizations for univariate, bivariate, and multivariate data.

Computing and visualizing densities is a typical summarization task, as it is used to show aggregated information about the raw data. In addition, the estimation of a probability density is closely related to the clustering task [FPSS96a].

3.1.1. Points

Points in the parallel-coordinates domain may represent points, lines, planes, hyperplanes, or p-flats with $p \in \mathbb{N}^+$ of the data domain. In order to distinguish different point-representations, Inselberg introduces the notation of *indexed points* [Ins09]. Points with one index represent one-dimensional projections of the data domain. An N-dimensional point P in the data domain is mapped to N indexed points $\bar{\ell}_i : (d_i, p_i)$ in the parallel-coordinates system. This can be used to represent marginal distributions on the axes, similar to a set of N one-dimensional scatterplots (also referred to as dot plot). Points with two indices $\bar{\ell}_{ij}$ represent lines of the respective $x_i x_j$-plane in the data domain, as described by the point–line duality in Section 2.2. For the generalization of this scheme to p-flats, see Chapter 5 in reference [Ins09].

The density of points with two indices can be used to detect lines in images [ICD97, DHH11]. Here, the data domain represents a grayscale image composed of pixels that are mapped to lines in a parallel-coordinates system with two axes for the horizontal and vertical pixel coordinates. Then, the density of intersecting points is evaluated, where high density regions or clusters are used as an indication of a line in the corresponding image. To capture lines with positive slopes, the first axis (e.g. for the horizontal position of pixels) is negated and appended to the parallel-coordinates system.

To combine the advantages of scatterplots and parallel coordinates, points have also been used to embed scatterplots between adjacent axes [YGX*09, HW10]. The respective point coordinates are determined by rotating either one of the axes by 90 degrees [YGX*09] or both axes by 45 degrees [HW10] to obtain the corresponding Cartesian coordinate system.

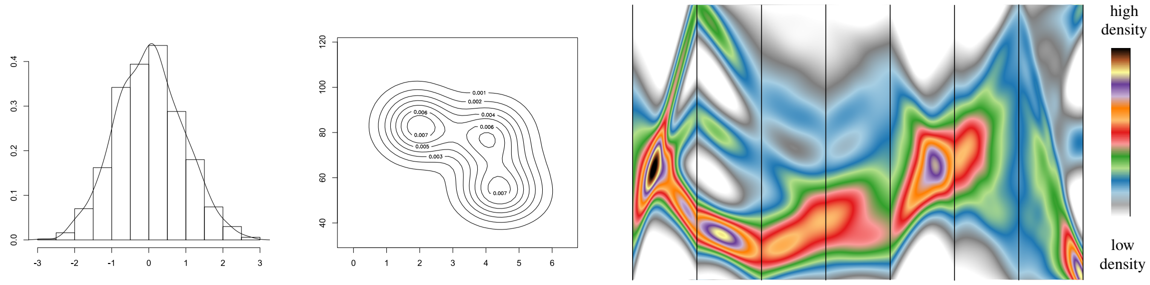
3.1.2. Lines

Due to the point–line duality, lines are the most common visual mapping for parallel coordinates. As described in Section 2, N-dimensional points are represented with a polygonal line intersecting each of the N axes at the respective coordinates (Figure 3 illustrates this scheme).

3.1.3. Curves

Using the interpolation model introduced in Section 2.3, the polyline resulting from connecting lines at the axes can also be described as a non-smooth, C^0 continuous curve that is not differentiable at the axes. Several authors proposed using smooth, C^n continuous curves with $n > 0$ to

1. visualize multiple, and higher-order correlations [The00, MW02],
2. facilitate line tracing [MW02, GK03, YGX*09, HW10, HLKW12],



(a) Histogram (b) Contour plot (c) Density-based parallel coordinates

Figure 6: Density visualizations for univariate (a), bivariate (b), and multivariate (c) data. The histogram shows a density estimate with discrete *bins* and an overlayed continuous density estimate reconstructed using a Gaussian kernel. The contour plot for bivariate data shows isolines for a density estimated from point-data using a 2D Gaussian kernel. Density-based parallel coordinates are computed from pairs of 2D density fields. A colormap was applied to the density in parallel coordinates.

3. enable the detection of overplotted line segments [GK03], and
4. visualize clusters using bundling [ZYQ*08, MM08, HLKW12].

Piecewise cubic B-splines can be used to visualize multiple pairwise correlations [The00] by choosing two “main axes” with an arbitrary number of additional axes placed in-between. Andrews plots [And72] can be obtained using an interpolation model with Fourier bases [MW02]. Other functions forming an orthonormal basis can be used to emphasize quantization effects on the axes and to detect second-order structures [MW02]. Piecewise quadratic and piecewise cubic interpolation models were proposed [GK03, MM08, HW10] to enforce tangents at a point $\overline{p}_i$ to be parallel to the line $\overline{\overline{p}_{i-1}\overline{p}_{i+1}}$. This model also resolves ambiguities if curves intersect axes orthogonally [HW10]. Many interactive implementations further add a parameter [HW10, HLKW12] to control the amount of smoothing. All these techniques guarantee curve smoothness and mitigate the line-tracing problem (see Section 6.3) by assigning different trajectories to curves that intersect at an axis.

3.1.4. Bundling

Curves can also be used for edge bundling [Hol06] to visualize clusters in parallel coordinates [ZYQ*08, MM08, HLKW12]. Here, a *bundle* represents all data samples belonging to a cluster defined a-priori [MM08] or emerging from the bundling algorithm [ZYQ*08]. Bundles can be visualized *implicitly* as a set of curves [MM08, ZYQ*08, HLKW12] or *explicitly* using polygons [MM08]. In both cases, the visual signature of a bundle is constructed by “attracting” one [MM08] or more [ZYQ*08] control points from individual curves toward a point that represents the respective cluster, such as the cluster centroid [MM08, HLKW12].

3.1.5. Polygons

Another mapping that readily supports the *summarization task* is from sets of points in the data domain to envelopes and quadrilaterals in the parallel-coordinates domain. This is also an example of the *explicit visualization* of sets or clusters, where the visual mapping for a group of data points is chosen prior to the rendering step and usually involves one or more filtering steps from the raw data (such as clustering the data). Given a set of data samples in the data domain contained in an N-dimensional convex hypersurface, Inselberg [Ins85] suggests drawing the envelope of the respective polygonal lines in parallel coordinates. Then, any point interior to the hypersurface in the data domain is represented by a polyline that is also interior to the envelope in parallel coordinates. Fua et al. [FWR99b] render convex quadrilaterals resembling the axis-aligned bounding box of a cluster in the data domain. The same geometric mapping can be used with different shadings for classification rules [HC00], fuzzy points [BH03], sets and subsets [AA04], contingency tables [BKH05, KBH06], binned data [NH06], multivariate time-series [JLC07], and quartiles [Mou11]. Non-convex quadrilaterals can also be used to indicate negative correlations [JLC07, ZMM12]. Other mappings, in shape similar to envelopes, evolved from bundling [MM08] and the visualization of line densities (see next section).

3.1.6. Density

In many cases, the density function

$$\sigma : \mathbb{R}^N \longrightarrow \mathbb{R}$$

describing the distribution of a (possibly multivariate) data sample cannot be reconstructed, but has to be estimated from data. A well-known probability density estimate for a univariate dataset $X = (x_1, x_2, \ldots, x_n)$ is the histogram (the term “histogram” is used both for a function representing a density estimate as well as for the visualization using rectangular "bins" (Figure 6), as proposed by Pearson [Pea95]) that Scott and Sain [SS05] define as

$$\sigma(x) = \frac{v_k}{nh}, x \in B_k, \tag{2}$$

where h is the (uniform) bin width for all bins $B_k, k \in \mathbb{N}$ and v_k is the number of observations falling in bin B_k. The histogram illustrated in Figure 6 (left) was computed using Equation (2). For the bivariate case, σ is defined on a two-dimensional domain $\sigma : \mathbb{R}^2 \longrightarrow \mathbb{R}$ and the bins B_k represent areas (usually rectangular) instead of intervals. The process of constructing such a 2D histogram is sometimes also referred to as *binning*. For visualization in the data domain, binned data is usually mapped to color. Hence, the model of a histogram is based on counting the number of samples per line segment in 1D or per area in 2D. The density as computed in Equation 2 can be thought of as the probability of observing a data point in B_k, and the total probability of observing a point in any bin equals one. A more general density estimate for multivariate data and arbitrary *kernels* reads:

$$\sigma(\mathbf{x}) = \frac{1}{nh^N} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x_i}}{h}\right) \tag{3}$$

where K is the respective kernel with *bandwidth parameter* h. Figure 6 illustrates the histogram with discrete bins and a continuous density estimate using Equation (3) with Gaussian kernels.

Similar to the implicit point-density model for Cartesian coordinates, a *line density* is implicitly encoded in parallel coordinates by the proximity of lines. A common approach to compute the density

$$\varphi : \mathbb{R}^2 \longrightarrow \mathbb{R}, (x,y) \mapsto \varphi(x,y)$$

explicitly in parallel coordinates at any given *point* $\bar{\ell} = (x,y)$ is to employ the same binning strategy [Nov04, JLJC05, HW10, DK10] as for the point-wise density computations in scatterplots. Here, the number of lines intersecting a 2D bin is evaluated instead of the points contained in the bin. Note that rectangular bins should not be confused with *pixels* [Smi95].

Binned densities can also be transformed to parallel coordinates using a scattering approach: quadrilaterals are rendered instead of lines, each representing a rectangular bin mapped from the data domain [AdOL04, NH06]. Here, the shading of quadrilaterals either reflects the density of the respective 2D bin (constant shading) or can be interpolated between one-dimensional density estimates corresponding to the respective axes [RTT03]. The final density at a point in parallel coordinates is then computed as the sum over the sample-contributions. This is typically implemented using additive blending.

Alpha-blending is used by many authors (assuming alpha-blending whenever the term *opacity* is used and the exact blending mode is not mentioned). Here, the contribution of the density of a line to φ at a point in parallel coordinates decreases exponentially with increasing number of contributions and thus does not converge to the line-density model. The advantage of this technique is that normalization is not required, as σ is bounded and converges asymptotically to a maximum value (typically 1 or 256). It is further important to note that alpha-blending is non-associative, i.e. the value of φ depends on the order of lines being rendered, for non-uniform distributions of α.

As Miller and Wegman point out [MW91], however, 2D binning in the parallel-coordinates domain might result in lines being counted in multiple bins, which violates the requirements of a probability density function to integrate to one. Instead, the probability of observing a line should be equal for any horizontal position, such that line density should be based on counting lines on vertical intervals instead of two-dimensional areas. A closed-form solution for bivariate normal and uniform distributions was given by Miller and Wegman [MW91]. Figure 7 compares the traditional, constant-density, line-based rendering with binning in the data domain and the approach proposed by Miller and Wegman. A density plot of the cars dataset obtained with Gaussian kernels in the data domain and the transformation to line density is illustrated in Figure 6. In addition, a colormap has been applied to the density field.

The model of continuous scatterplots [BW08] for the mass-conserving transformation of density from the spatio-temporal domain to the data domain was also extended to parallel coordinates [HW09]. A closed-form solution for the computation of continuous parallel coordinates from any two-dimensional density field and a discussion of different numerical and analytic integration approaches was presented by Heinrich and Weiskopf [HW09]. Figure 8 compares a discrete density-based parallel-coordinates plot with continu-

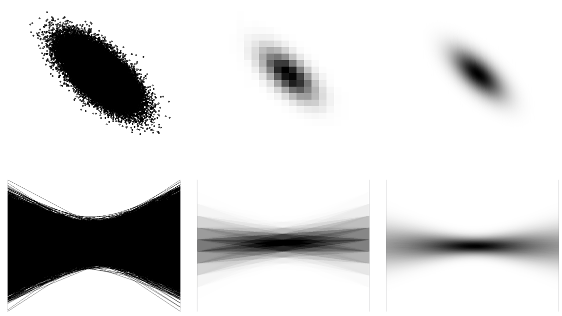
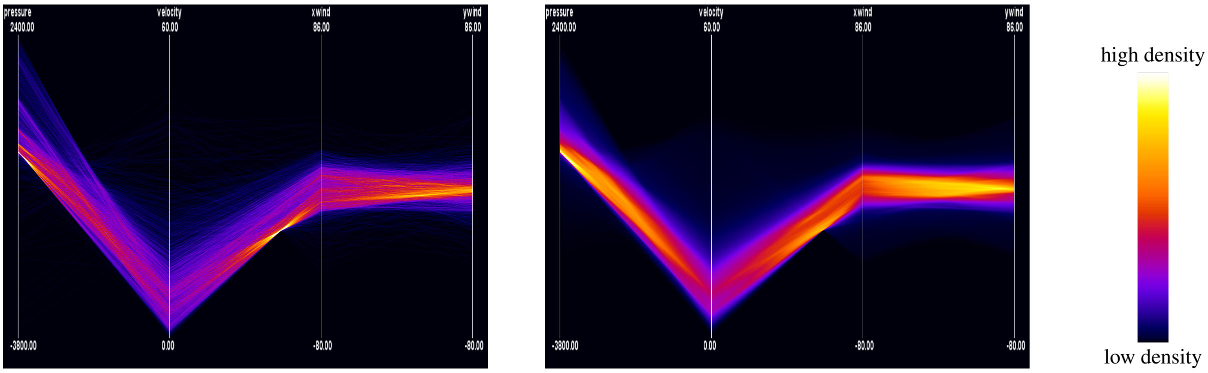


Figure 7: A sample of 100000 observations drawn from a bivariate normal distribution (top, left) rendered using traditional parallel coordinates (bottom, left), binned parallel coordinates (bottom, center), and the line-density model proposed by Miller and Wegman [MW91] (bottom, right). The dual pattern for each approach in Cartesian coordinates is shown in the top row.



(a) Discrete density-based parallel coordinates

(b) Continuous parallel coordinates

Figure 8: Discrete parallel coordinates computed using the binning approach as described in Section 3.1.6 and continuous parallel coordinates for the "hurricane Isabel" dataset at a spatial resolution of $25 \times 25 \times 5$. The Gaussian distribution at low velocities is apparent in continuous parallel coordinates but can not be seen in the discrete version. The high-density region at low pressure and low velocity constitutes the eye of the hurricane.

ous parallel coordinates for the same dataset. Splatting and progressive refinement [HBW11] can be applied to speed up computation time and allow for arbitrary interpolation schemes.

Using independent bivariate density estimates in the data domain for each pair of axes produces footprints with a potentially different density for every segment in parallel coordinates. As a result, the rendered primitives might not be visually traced over all axes, losing visual coherence. To accommodate for this, Moustafa [Mou09] quantizes densities in the parallel-coordinates domain between adjacent axes and accumulates the binned frequencies for each data point. After normalization, polylines are rendered in order of ascending cumulated frequencies, i.e. the data point with the highest overall frequency (over all 2D-projections) is drawn last. Distance-based weighting schemes [ZCQ*09, HW10] were also used to construct a multidimensional density function which is then used to assign a density to polylines. Anisotropic diffusion of noise textures [MHDG11] were also proposed to visualize line orientations for density-based parallel coordinates computed from discrete samples.

Similar to curve-based parallel coordinates, density can also be used to resolve ambiguities, as illustrated in Figure 9.

3.2. Axes

Axes are an important part of a parallel-coordinates plot that fulfill many purposes: they implicitly visualize the axis spacing $\mathbf{d_N}$, help an observer read off the value of a sample, and serve as a visual anchor for labels, ticks, and other overlays. Axes are usually mapped to straight lines and rendered solid and fully opaque. Labels are typically attached either at the top or at the bottom of an axis and can be rotated in order to save space. As with axes in other coordinate systems, arrows can be used to indicate the direction of increasing values. Axes are often enriched, or composited, with additional information about the respective dimension. Common examples of such overlays are histograms [HLD02, GPL*11, WGJL12] or boxplots [Sii00, The03]. Other mappings for axes are curves [QCX*07, WGJL12] for the representation of polar coordinates and tag clouds [CVW09] for the visualization of word-frequencies.

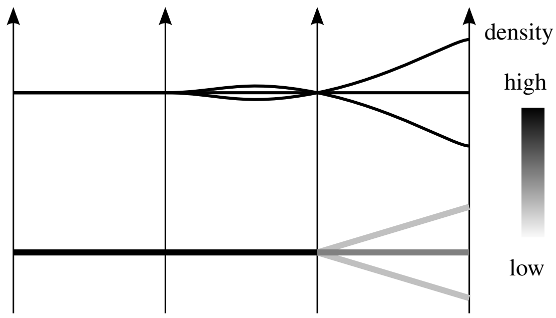


Figure 9: Some ambiguities cannot be resolved entirely using curves (as in Figure 13), as tangents only depend on adjacent axes. The lines appear as one in the leftmost segment using curves (top). The density representation (bottom) reveals two different densities, where the horizontal line appears darker than the other two. Assuming equal and constant densities for each sample, this means that at least two samples are contributing to the density of the horizontal line. In conclusion, the plot must be showing at least four samples instead of three as the top plot suggests.

3.3. 3D Plots

Several approaches to rendering axes [JCJ05, LJC09] and samples [WLG97, FCI05, RWK*06, DWA10, Mou11, WGJL12] in 3D are known for parallel coordinates. The placement of axes on a plane in a 3D world allows one to visualize multiple 2D parallel-coordinates plots without duplicating axes. For the visualization of sets of parallel-coordinates plots, such as for timepoints of dynamical systems [WLG97] or the expression of genes at different spatial positions [RWK*06], stacking the single plots along the third axis [WLG97, RWK*06, DWA10] or rotating the parallel-coordinates domain around a shifted x-axis [FCI05] was proposed. While 3D representations allow more flexibility by adding one degree of freedom to the visualization, they also introduce occlusion and distortion by projection.

4. Image Analysis

This section presents work related to parallel coordinates in an image analysis context. Here, image analysis refers to any process that uses parallel coordinates or a parallel-coordinates plot as input. Examples are the visual perception of a parallel-coordinates plot by a human observer, e.g. in a data-analysis task, or the processing by a computer algorithm, e.g. for automatic feature detection.

Some formal evaluations compare traditional parallel coordinates with other visualizations, namely scatterplots [LMvW10, HW10, KZZM12] and stardinates [LMP05]. It was shown that humans perform better using scatterplots than parallel coordinates in visual correlation analysis [LMvW10] and cluster identification [HW10] tasks. The former study investigated the participants' ability to estimate the Pearson correlation of two random variables in scatterplots and parallel coordinates, while the task in the latter study was to estimate the number of clusters in a dataset. The same task was shown to be effective using bundled parallel coordinates [HLKW12]. The performance of estimating the coordinate value of a given N-dimensional point at a given dimension was found to be better using parallel coordinates than scatterplots for small datasets [KZZM12]. The perception of patterns in the presence of different levels of noise was investigated by Johansson et al. [JFLC08]. They found out that patterns in parallel coordinates can be identified with a probability of 70.7% if approximately 13% noise was added to the signal. The patterns were created using a sample of 300 points from five different signals, including linear and sinusoidal functions. Other studies showed that parallel coordinates are effective in querying databases [SR06] and alarm filtering [AR11]. Finally, there is evidence that understanding patterns in parallel coordinates can be learned quickly [SLHR09].

Parallel coordinates have also been used for the automatic detection of lines [ICD97, DHH11] and other features [LT11] of the data domain as well as for the computation of metrics for visual abstraction [JC08] and for the ranking of 2D plots [DK10] (see also Section 3.2). Line detection in images can be realized using the density-based mapping approaches presented in Section 3.1.6. Rendering a line for every pixel of a grayscale input image with the respective density results in a parallel-coordinates plot similar to the example in Figure 7. The density at a point in parallel coordinates now reflects the density of the dual line of the image. Note that, in order to detect lines with positive slopes (with points in the parallel-coordinates domain located to the left or right of the axes), one of the spatial axes has to be inverted and added to the plot [DHH11].

5. Interaction

Interaction plays an important role to enhance perception for dataset exploration and visual data mining [FdOL03]. It enables the user of a software to change parameters interactively and get immediate feedback from the system. In the KDD process, interaction allows the user to modify each step of the pipeline individually, from the acquisition of a new dataset over changing normalization parameters to defining new visualizations. According to the information-seeking mantra [Shn96], the user of a data-analysis system should gain an *overview first*, with the option to get *details on demand*. The previous sections illustrate how static images of parallel-coordinates plots are used for tasks such as summarization, dependency modeling, or cluster detection. *Interactive parallel coordinates* further support these tasks and enable the exploration of a dataset.

There are many interactions possible with parallel coordinates, as any free parameter of any technique presented in the previous sections could be changed interactively. For this reason, only interactions compatible with the traditional parallel-coordinates plot are considered here, based on the geometric framework of Section 2. While others classified interactions with parallel coordinates by task [AA01, SR06], the same taxonomy as in Section 3 is used here to distinguish between interactions with samples and axes.

5.1. Interacting with Samples

5.1.1. Brushing

A common interaction technique used in statistical graphics is the *brushing* of samples, which was introduced for the *masking* and *isolation* of data points in scatterplots [FFT75]. Brushing is an operation that allows the user to select a subset of samples by means of a *brush* [BC87], which originally referred to an axis-aligned rectangle for selections in scatterplots. The selected set of points is then used as input for subsequent operations, such as highlighting, labeling, replacing, deleting, and many more [BC87, BCW87]. A particularly important task supported by highlighting brushed

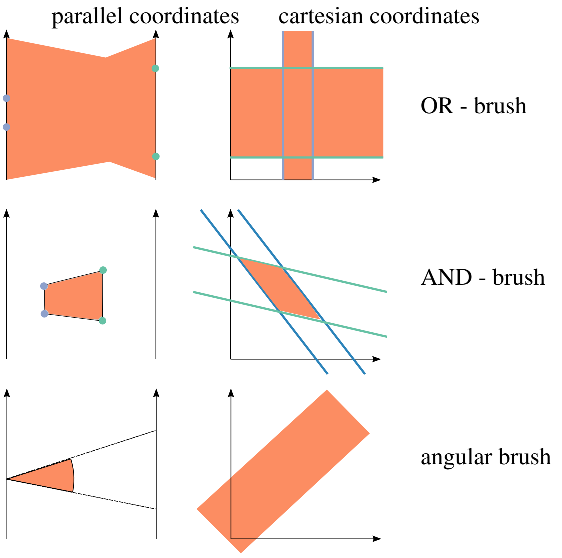


Figure 10: Brushing in parallel coordinates. Top: Axis-aligned brushing. A brush on one axis (e.g. left axis, bounded by the blue points) corresponds to a one-dimensional interval brush in the data domain (bounded by the dual lines). The union (OR) with another interval (green) results in the orange brush. The intersection (AND) of two intervals is shown in the center. Translating the blue and green points in parallel coordinates results in a rotation of the dual region in Cartesian coordinates. The bottom row illustrates the dual of an angular brush to the data domain, which corresponds to a set of ideal points $\bar{\ell}_\infty$ as in Figure 4.

samples is the visual linking of data samples between multiple graphical representations (*brushing and linking*), as in the scatterplot-matrix [Har75, BC87]. Brushing can further be direct and indirect [MW95], be composed of logical operations [MW95] or graphs [Che03], and be applied to dimensions instead of samples [TFH11]. As most of those concepts are applicable to parallel coordinates as well, the discussion will be restricted to the geometry of brushes and methods specifically designed for parallel coordinates.

An axis in the parallel-coordinates domain represents a set of parallel lines (or the ideal point) in the data domain [Ins09]. Brushing a point on an axis is thus equivalent to the selection of a line (i.e. all points on a line for discrete data) in the data domain. In addition, these lines are perpendicular to the respective axis in the data domain, such that the brush depends only on one dimension. Accordingly, a *range* on an axis in parallel coordinates results in an interval on the respective dimension in the data domain (such as the blue and green intervals on the axes in the topmost illustration in Figure 10). Extending such a one-dimensional brush to multiple axes enables the construction of higher-dimensional brushes [MW95] using logical operators [War94, AA99, HLD02] or graphs [Che03]. For instance, the *AND*-operation can be used to subsequently build a convex polygon in parallel coordinates which is dual to a hypercube in the data domain.

Exploiting the rotation–translation duality, line-based and polygon-based brushes can also be employed in the space between axes. As indicated in Figure 10, translating the blue and green points in parallel coordinates results in a rotation of the corresponding area in Cartesian coordinates.

Another brush that can be used to select samples in parallel coordinates is based on the slope of lines between adjacent axes. With angular brushing [AA99, HLD02], a range of angles in parallel coordinates (e.g. relative to the horizontal) can be used to define a set of ideal points $\bar{\ell}_\infty$ as a brush. In contrast to axis-aligned brushing, angular brushing enables a line-based brush in the data domain and thus further allows for the selection of lines with positive slopes in the data domain without the need to flip axes (see also Section 5.2).

For large datasets, hierarchical brushes using wavelets [WB96] and hierarchical clustering [FWR99b, FWR99a, FWR00] have been proposed. Here, brushed samples are aggregated in a balanced [WB96] or unbalanced [FWR99b] tree that can be navigated in discrete steps by defining the current depth [WB96] or continuously with arbitrary cuts [FWR00]. Both techniques give the user control over the current level-of-detail (LOD).

Traditional brushing can be expressed as binary function assigning either 0 or 1 to every sample in the dataset. Smooth brushing [MW95, HLD02, FKLI10] uses a continuous functions instead and allows one to express a certain degree-of-interest to any point (line) in the data (in parallel coordinates). However, composites are more difficult to compute using smooth brushes [MW95, HLD02].

Brushing in parallel coordinates can be supported by haptic feedback, e.g. by projecting a parallel-coordinates system on a mixing-board interface [CBS*07]. Bimanual interaction was found to be helpful for exploration and can also be used for angular brushing with touch interfaces.

5.2. Interacting with Axes

The position of axes in a parallel-coordinates plot has a high impact on the patterns emerging from the visualization of samples, as they define the scheme for locating an individual sample in the parallel-coordinates system. Translating axes changes the order of variables and the spacing in-between. The scaling determines the range of values that intersect an axis and provides a mechanism for flipping axes. Both operations, translation and scaling, cover a wide range of interactions that have been proposed for parallel coordinates.

5.2.1. Translation

The absolute horizontal position of axes $\mathbf{d_N}$ in parallel coordinates is a free parameter of the visualization and does not affect the validity of the point–line duality. The relative distance between adjacent axes is usually chosen to be uniform, as this configuration puts equal emphasize on all pairwise variable relations. However, in some cases it is beneficial to move axes horizontally, e.g. to investigate a particular pattern in detail (by exploiting the additional space gained for one pair of axes if another axis is translated horizontally), or to manually rearrange the axis order. Axis translation is often implemented as a *drag-and-drop* operation, where a uniform axis spacing is reconstructed after releasing an axis.

Translating axes and associated sample coordinates in the vertical direction can be useful to align a set of axes to a common scale or a common value [AA01].

5.2.2. Scaling

As with most statistical plots, patterns emerging in parallel coordinates depend on the scale of variables and axes. The default range of values represented on an axis is bounded by the minimum and maximum values of the corresponding variable, i.e. the smallest value will always intersect the axis at the bottom and the largest value at the top. While this setting allows us to see patterns in data of different units, it is not suited to compare values of equal units if the range of measurements differ between axes. Here, a uniform scale on all axes might be a better solution. Axis scaling is equivalent to applying a function to all values of the respective variable and has also been referred to as *dimension zooming* [FWR99b]. Scaling can be used to align axes to a a common base [AA01], such that one sample is represented as a horizontal line. This allows the user to visually estimate the similarity of other samples with respect to a reference.

A special case of scaling is the *flipping* of axes. Flipping negates all values of the respective dimension, which has the effect of reversing the relation of positive values at the top and negative values at the bottom. As a result, the slopes of lines are also negated as well as the patterns for negative and positive relations. Hence, a set of parallel lines indicating a positive correlation is transformed to a negative correlation, which can be represented as a point in parallel coordinates. This is particularly useful for systems searching for points in a parallel-coordinates plot, e.g. for the automatic detection of lines in the data domain [ICD97, DHH11]. Here, a two-dimensional data domain is represent using three axes, say $\overline{X}_1$,$\overline{X}_2$, and $\overline{X}_1'$ in parallel coordinates, where $\overline{X}_1'$ denotes the flipped $\overline{X}_1$. Now, the intersection of two lines will always occur within one of the segments.

6. Challenges

As we have seen in the previous sections, many decisions have to be made in order to find the "right" way to visualize (Section 3), interact (Section 5), or analyze (Section 4) parallel-coordinates plots. Similarly, the research conducted in the area of parallel coordinates may be categorized by visualization or interaction techniques, analysis tasks, applications, or *challenges*. While the challenge is clearly defined by a particular question or data analysis task (e.g. "find outliers in the data"), many authors motivate their work implicitly or explicitly by addressing some sort of "drawback" of a particular visualization. A good example of such a deficiency is "the clutter" in parallel coordinates, and the corresponding challenge is to reduce it. While there are objective measures for clutter [ED06], a subjective quantification of clutter in practice usually depends on the context and individual experience of the observer with the respective visualization. In many cases, no particular analytical task is addressed explicitly by reducing the clutter, although diverse findings such as clusters, outliers, or other patterns can be revealed by doing so. As a consequence, many researchers were faced with the following challenges when visualizing data with parallel coordinates:

- **Overplotting** occurs in parallel coordinates if lines potentially occlude patterns in the data.
- The **order of axes** implicitly defines which patterns emerge between adjacent axes.
- The **line-tracing** problem occurs if two or more lines intersect an axis at the same position.
- **Nominal and ordinal data** such as sets and clusters have to be mapped to a metric scale before it can be visualized in parallel coordinates.
- **Time series** are special in that time points, if interpreted as dimensions, have a fixed order.
- **Uncertain** data is another challenge for visualization, and there are approaches for the visualization of uncertainty in parallel coordinates.

6.1. Overplotting

The most prominent challenge in parallel coordinates is the clutter produced by a large number of lines, which potentially hide the patterns contained in the data. Lines need more ink than points such that the total mass of data appears larger in parallel coordinates than in scatterplots.

While many authors use the term "clutter" as a synonym for "density" [ED06, ED07], it is important to note that a dense display can reveal important information as well, even without any modification to the traditional parallel-coordinates plot [Ins09]. Here, we loosely define "clutter" as a parallel-coordinates plot that does not reveal any pattern useful to the observer.

The clutter reduction techniques for parallel coordinates can be categorized into data-driven and screen-based approaches. The former refers to algorithms that operate on the data *before* mapping- and rendering in terms of the visualization pipeline and do not affect the visualization. The latter are methods that modify parameters of those two stages. Hence, clustering the data and visualizing only the cluster

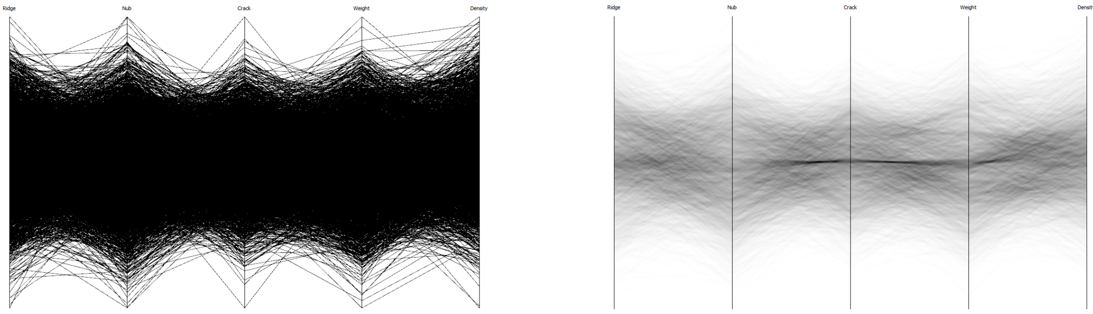


Figure 11: A frequently used synthetic dataset for parallel-coordinates plots about the geometric features of pollen grains. The large number of lines (3848 observations, left) may hinder the perception of patterns in the data. Note, however, that the hyperbolic shape of the envelope hints at normally distributed data. Density-based approaches (right) may reveal patterns that are not visible otherwise. The cluster of samples that appears in the center forms the word "EUREKA" if viewed in a scatterplot.

centroids in traditional parallel coordinates is an example of a data-driven clutter-reduction approach, while zooming into the image is a screen-based approach that might have different effects for different visualizations.

Some approaches to clutter reduction in parallel coordinates are discussed using a slight modification of an established taxonomy [ED07]. The methods are grouped in filtering, aggregation, and spatial distortion techniques.

Filtering is an operation that removes signals from its input. A filter reduces the number of lines to be rendered. In this sense, dynamic querying [Shn94] is a filter, if implemented with brushing (Section 5.1.1), which reduces clutter by putting the filtered lines *in focus* using some highlighting mechanism. Combining simple brushes using logical operators [MW95, AA99] further allows the user to formulate rather complex queries that might even achieve faster and more accurate results using parallel coordinates than using a Structured Query Language (SQL) [SR06]. Another type of filter uses sampling at lower rates than for the input data and has been suggested to reduce the actual number of lines to be rendered [ED06] depending on the density (Section 3.1.6). This approach assumes that subsets of the data may represent the dominant features if sampled appropriately. Clearly, it depends on the sampling strategy and the density estimation technique [ED06].

Aggregation refers to the computation of sum or integral of a subset of data and can be performed in the data domain and in the parallel-coordinates domain. There are many different ways to aggregate data and to render the resulting *aggregate items* [EF10]. To reduce clutter aggregates are rendered instead of individual samples. Typical aggregate items computed in the data domain are the mean [Sii00, HLKW12, HHD*12], median [RZH12], or cluster centroid [FWR99b] of a subset of samples. The range of visual mappings for aggregate items covers those discussed in Section 3. Traditional polylines [Sii00] and curves [MM08, HLKW12] can be used either alone [Sii00] or as an overlay [HHD*12] if no information about the distribution of the subset is available. Polygons [FWR99b, AA04, RZH12], histograms, or boxplots on the axes provide means to visualize the extent and distribution of subsets. Clusters can also be visualized using bundles. Hierarchical data structures [FWR99b, RZH12] can further be exploited to render lines or aggregate items at different levels of detail or to progressively refine the final visualization. The computation of a density (Section 3.1.6) is often referred to as a clutter-reduction technique as it is particularly useful to reveal dense areas and clusters in the data (Figure 11).

Spatial distortion techniques apply a transformation to the viewport. The most common representatives are fisheye views and the traditional, linear zoom. Distortion can help resolve uncertainty about line crossings, clarify dense areas, and brush individual lines with a pointing device. In addition, horizontal distortion (changing the axis-spacing vector) affects angles and slopes of lines, which can have an impact on the accuracy of judging angles [CM84, CM87, GW12].

In parallel coordinates, axis scaling (Section 5.2.2) can achieve the same effect as spatial distortion by rescaling the data at adjacent axes using the same function. However, axis scaling is performed in the data domain and further allows one to use different scales for each axis. Axis scaling thus belongs to the class of *line-displacement* techniques for clutter reduction.

Dimensional reordering in parallel coordinates is the same as axis translation (Section 5.2.1). Reordering the axes in a parallel-coordinates plot may reduce clutter by revealing patterns (e.g. of correlation) that might have been hidden before. An overview of axis-reordering techniques is given in Sections 5.2.1 and 3.2.

6.2. Axis Order

Since parallel coordinates were introduced [Mau85], axes are placed in parallel with different preferences for a hori-

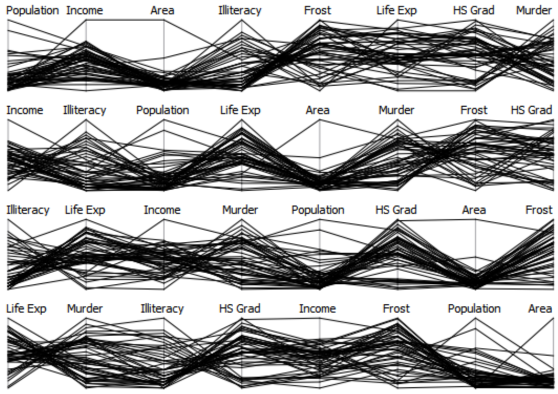


Figure 12: Different axis orders exhibit different patterns of correlation. The 8-dimensional census dataset [BCW88] shows several statistics of the 50 states of the United States of America and is layed out in the Parallel Coordinates Matrix (PCM) [HSW12] such that every pair of axes appears exactly once. The topmost plot shows a negative correlation between "Illiteracy" and "Frost", while the bottom plot indicates that "Life Exp" is negatively correlated with "Murder". Taking a close look at the "HS Grad" axes, we find that there is a cluster of states having a low rate of high-school graduates. Also, the bottom row indicates a negative correlation between "Illiteracy" and "HS Grad". A moderately negative correlation seems to be between "Life Exp" and "Illiteracy" in the third row, as well as a positive correlation with "Income".

zontal [Ins85] or vertical [Weg90] layout. Independent of the orientation, the order of axes affects the patterns revealed by a parallel-coordinates plot [Weg90] (see Figure 12). As there are $N!$ possible orderings for N axes, many researchers addressed the *axis order problem* in their work. While most of the papers deal with using some measure to score an ordering of axes, others build on that and discuss how to visualize multiple orderings in a single display.

Considering two-dimensional relations, where the order of N axes defines the pairwise plots of the full parallel-coordinates plot independently of the orientation, it is useful to model these relations in a graph-theoretic framework [Weg90, QCX*07, HO10, ZMM12] where vertices $V = \{x_i | i = 1,\ldots,N\}$ represent axes and edges $E = \{\{x_i, x_j\} | i, j = 1,\ldots,N\}$ represent pairwise plots of axes. Now, the *complete graph* K_N models the set of all pairwise relations between N dimensions and $|E| = \frac{N(N-1)}{2}$. Note that a parallel-coordinates plot can be constructed by following a *path* in K_N and laying out axes in parallel according to the order of nodes in the path. In particular, the traditional parallel-coordinates plot corresponds to a *Hamiltonian path* in K_N, i.e. a path that visits every node exactly once. See Hurley and Oldford [HO10] for an excellent treatment of graph-theoretic approaches to the pairwise display of variables.

There are different ways to visualize *all* pairwise relations in parallel coordinates using the previously described graph model. In general, it suffices to find an Eulerian trail [HO10] visiting all edges in K_N and laying out the axes in parallel coordinates accordingly. For $N = 2m+1, m \in \mathbb{N}$, no such trail exists, and some redundancy has to be tolerated by visiting some edges twice. For some applications, it is necessary to add another constraint to the problem of visualizing all pairwise relations by requiring subpaths to be Hamiltonian and of length N. In other words, all pairwise relations should be visualized in sets of N-dimensional parallel-coordinates plots, where every plot contains all N axes of the input dataset. Such a *Hamiltonian decomposition* of the complete graph K_N into m Hamiltonian paths for $N = 2m$ and m Hamiltonian cycles for $N = 2m+1$ can be used to visualize all pairwise relations in a single parallel-coordinates plot [HO10] (with some edges visited twice for $N = 2m$) or in a matrix-layout [HSW12] as in Figure 12 (with some vertices visited twice for $N = 2m+1$). Other matrix-based visualizations of multiple parallel-coordinates plots use latin-squares [VMCJ10], ranked displays [TAE*09, AEL*09] and manual orderings [CvW11].

With increasing N, all approaches to enumerate and visualize multiple paths will become impractical at some point, either due to the computational complexity or the limited screen real-estate. Then, a choice has to be made to decide which axis order to prefer. This problem can be translated to the graph-model by weighing edges with a distance measure $d : (x_i, x_j) \longrightarrow \mathbb{R}$ and order paths by their total edge weight. The metrics for ordering axes in parallel coordinates can be grouped into data-space measures [ABK98, YPWR03, Guo03, ZLTS03, PWR04, Hur04, WAG06, JKL*09, HO10, FR11, ZK12] defined in the data domain and image-space measures [TAE*09, AEL*09, DK10, TAE*11] defined in the parallel-coordinates domain. Data-space metrics are well-known from statistics and data mining and include the Euclidean distance, Pearson correlation, Kendall's τ, etc. In contrast, image-based metrics measure the slope of lines, their overlap (density), the number of line crossings and -angles, convergence, etc. Screen-based metrics [NH06, DK10] operate on the rasterized image of a parallel-coordinates plot and further incorporate the current screen-resolution when computing a measure. The most common tasks being supported by both types of measures are correlation analysis [Hur04, JKL*09, HO10, FR11, ZK12], clustering of data points [Guo03, TAE*09, AEL*09, JKL*09, TAE*11, FR11, ZK12], clustering of dimensions [ABK98, Hur04], clutter reduction [PWR04], dimensionality reduction [YPWR03, JKL*09], and outlier detection [WAG06, JKL*09]. Note that all measures can be applied before or after rasterization in the respective domain, which allows one to include the current resolution into the computation of a metric. As even finding the single Hamiltonian path/cycle with the smallest edge weight is NP-hard [HO10], heuristics [ABK98,Hur04,HO10] or manual path-selection [QCX*07,ZMM12] can be used instead.

Other approaches were proposed to order axes according to higher-order measures [The00, JKL*09, FR11], clustering [IA99, YPWR03], or 3D-parallel-coordinates plots [LJC09]. Without changing the order of axes, a grand tour can be used with parallel coordinates to traverse different projections of the data.

6.3. Line Tracing

The line-tracing problem in parallel coordinates is a special case of the *linking problem* in statistical graphics [CM84]. Given two data points $a = (a_1,a_2,a_3)^T \in \mathbb{R}^3$ and $b = (b_1,b_2,b_3)^T \in \mathbb{R}^3$ and two 2D plots relating x_1 with x_2 and x_2 with x_3. Linking a with b is the task to relate the lower-dimensional projections with each other by some visual means. For a single polygonal line, parallel coordinates inherently solve the linking problem. However, if a and b coincide on one dimension, e.g. $a_2 = b_2$, it is impossible to visually link the points. This is demonstrated in Figure 13, where it is not possible to assign all line segments unambiguously to a data point. There are basically two approaches to mitigate the linking problem for parallel coordinates. Using different colors to distinguish different points is a popular solution. However, this approach does not scale well with the number of points as it is difficult for the human visual system to reliably distinguish more than twelve colors [War04]. The other technique is to use curves instead of lines (see Section 3.1.3 for a review of the different implementations using curves). In contrast to lines, curves provide at least C^1 continuity and thus support the Gestalt principle of continuity. The disadvantage of using curves is the distortion of values between axes, such that some of the geometric properties as presented in Section 2 are not valid. However, other statistical properties of curve-based parallel coordinates were shown to be useful for pattern recognition [Mou11].

6.4. Sets and Clusters

The previous section presented clustering as a clutter-reduction technique. The focus of this section is the visualization of pre-clustered data with parallel coordinates. Here, the motivation for clustering is not to reduce clutter but to visualize patterns or anomalies within or between *sets of data*. For metric data, some of the techniques presented in the previous section about aggregation are applicable, i.e. the representation of a cluster by its mean value (or *centroid*). However, sets are not necessarily metric data and are often used to categorize a data*set*. A simple but effective method to distinguish a small set of categories are colors. If the color channel can not be used, bundling has been shown to work well for the identification of clusters while having a low impact on the effectiveness of the estimation of

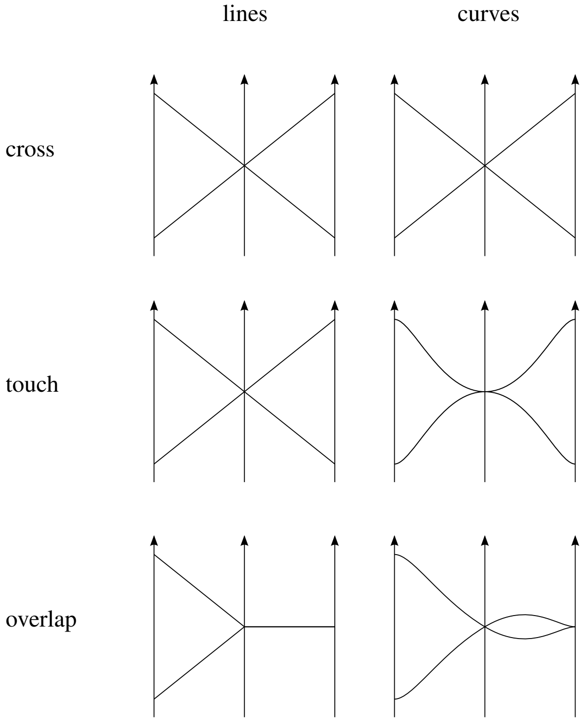


Figure 13: Ambiguities for tracing lines can partly be solved using curves instead of lines. For a pair of lines crossing at the midway axis (top), the curve model exhibits the same pattern. If lines touch instead (middle), a different pattern emerges. Note that without knowing the underlying model, it is still not possible to visually trace the lines. For lines that coincide (bottom), a smooth representation succeeds in disambiguating the samples.

correlations [HLKW12]. Other approaches based on geometry are to map clusters to envelopes [Mou11] or bounding-boxes [FWR99b].

6.5. Time Series

Time series are frequently visualized using line plots, where a single line or curve represents the progression or change of a data point over time. These plots can be constructed with the linear interpolation model of Section 2.3, simply by labeling the dimensions of the data domain as the time points of a time series. Using this model, time-series plots are a special case of parallel-coordinates plots, with the restriction to a common scale on every axis and a fixed ordering of dimensions. This has implications in both directions—from time-series plot to parallel-coordinates plot and vice versa. On the one hand, some of the results that were presented for parallel coordinates might also be valid for the interpretation of time-series plots. On the other hand, one of the reasons of the popularity of parallel coordinates might be the familiar visual pattern of a line interpolating a set of points that is long known from time-series plots such as stock market diagrams or the temperature forecast. While both types of visualization are expressed using similar visual mappings, the underlying model is different, as time points are samples of a one-dimensional continuous domain, whereas the axes in parallel coordinates represent one dimension each.

Several authors combined the visualization of time series and parallel coordinates. A simple but effective technique is to append data dimensions as axes to a time series plot [DHNB09], which enables the brushing of data samples with respect to the additional variables. Interchanging axes with "profiles" in a time-series plot allows for a truly multivariate interpretation of time-series data: here, an axis represents a measurement, dimension, or variable, while a data point is mapped to a polyline. Inselberg maps time to an axis in parallel coordinates and visualizes aircraft trajectories with indexed points in parallel coordinates [Ins01]. Although samples from time-series are intrinsically ordered, the order in which data samples are rendered in parallel coordinates has no effect on the final visualization unless α-blending or a density model is applied. *Temporal parallel coordinates* [JLC07] respect the order of time points and render a constant-density polygon for two consecutive samples. This corresponds to a nearest-neighbor interpolation of values in the data domain. The density approach is scalable and allows one to put more emphasize on large gradients, i.e. for which data dimensions the total amount of change is highest.

Another technique for the visualization of time-series in parallel coordinates employs animation [BS04, The06]. Here, a single parallel-coordinates plot is computed for every time step. A series of frames can then either be animated automatically or explored by the user stepwise.

6.6. Uncertainty

Uncertainty is a term that is difficult to define, and it is not the purpose of this section to do so. For the upcoming discussion, uncertainty may refer to variance, error, precision, or noise. We will shortly review how uncertainty may be introduced by visualizing data with parallel coordinates, how it can be addressed, and how a given, quantitative measure of uncertainty may be visualized along with the primary data in parallel coordinates.

According to the taxonomy of Dasgupta et al. [DCK12], uncertainty in parallel-coordinates plot occurs in the process of encoding the image of a parallel-coordinates plot as well as in the decoding steps involved when processed by humans or machines. For humans, additional uncertainty may arise individually due to different aspects of cognition. For example, an experienced user might be less uncertain about the recognition of patterns in parallel coordinates than a novice. Encoding and decoding directly relate to what I termed *image generation* and *image analysis*, for which we briefly discuss uncertainty.

The encoding of a parallel-coordinates plot introduces uncertainty in different stages [DCK12] of the transformation from the data domain to the parallel-coordinates domain. Despite from the loss of information due to the projection of a high-dimensional dataset to a set of 2D spaces, *visual uncertainty* may have a variety of sources in parallel coordinates. The user-driven filtering of dimensions and the application of algorithms in the data-mapping stage introduces uncertainty regarding the *completeness* (sample size) and the *configuration* (axis ordering) of the plot. Note that, in the KDD pipeline [FPSS96b], data mapping refers to the transformation step. The rendering of a parallel-coordinates plot causes further uncertainty as it involves sampling lines or line-densities on a regular grid (the pixels). This step depends on the resolution of the screen (the sampling frequency), the sampling kernel (usually a rectangular function), and the reconstruction kernel (rectangular for opaque lines). These parameters influence the *precision* in the visual mapping of data samples to lines. Visualizing aggregated information such as clusters instead of individual samples decreases the *granularity* and with that increases the uncertainty of the visual representation. Granularity is a common parameter subject to interaction and is often controlled by detail-on-demand operations (see also Section 5).

The analysis of an image of parallel coordinates consists of decoding the information contained in the sampled representation of the plot. In order to discuss the theoretical aspects of uncertainty associated with perception and cognitive processing of a parallel-coordinates plot, however, a perfect reconstruction of lines has to be assumed. Then, the human visual system introduces uncertainty when sampling the image, for the same reasons as above. The traceability of lines at the intersection with axes is yet another source of uncertainty that occurs for many visualizations where overlap is possible [EF10]. This type of uncertainty can be reduced with brushing (Section 5.1.1) and curves (Sections 2.3 and 3.1.3). A related problem is the identification of lines in heavily cluttered displays with the special case of overlapping lines. While the former can be resolved geometrically with increasing resolution or by scaling, the latter can be detected using density or transparency (Section 3.1.6).

The representation of uncertainty in parallel coordinates has been addressed by few researchers. One approach is to model data points with a normal distribution and to map these to parallel coordinates with respect to the point–line duality (see Section 3.1.6). The resulting image resembles a probability distribution of lines in parallel coordinates, such that uncertain points are de-emphasized while certain values appear more salient.

7. Applications

This section provides references to some of the many applications of parallel coordinates in the life sciences and engineering domains. Due to the large amount of publications using parallel coordinates, we can only provide for a short and non-exhaustive list of selected applications per domain.

7.1. Life Sciences

Parallel coordinates were used in a number of applications in different fields of the life sciences, including biology [KERC09, GPL*11], bioinformatics [DHNB09], systems biology [BMGK08, HHD*12], genetics [RWK*06], functional genomics [MWS*10], neurophysiology [TCMR05, tCM07] or computational chemistry [Bec97]. For gene expression data, parallel coordinates can be used to visualize the "profile" of genes (lines) over a set of conditions (axes). This is the natural counterpart to the heatmap [ESBB98], where rows represent genes and columns represent conditions. Note that in many occasions, conditions represent time steps which makes the corresponding parallel-coordinates plot a time-series visualization, where the order of axes is fixed. However, appending statistical dimensions to "profile plots" is a useful technique for model verification and querying statistical properties [DHNB09]. Gene profiles can also be visualized in parallel coordinates to help experts establish functional relationships of the expression of genes to their spatial location [RWK*06, MMD*10]. In conjunction with metabolic networks, parallel coordinates were used for the visualization of network parameters for single cells [BMGK08] and cell populations [HHD*12].

7.2. Engineering

Data in engineering applications often consists of multi-attribute samples given in the spatio-temporal domain, that need to be analyzed with respect to the time and place they were taken. Hence, linking parallel-coordinates plots (representing the multi-attribute data) to maps and 3D-visualizations (representing the spatial domain) was shown to be useful, e.g. to compare census data of different countries [AA01], visualize health statistics [Eds03], analyze traffic [GWY*11] or computational fluid dynamics (CFD) data such as weather simulations [DMH04, BBP08] or nasal air flow [ZMH*09]. Continuous parallel coordinates [HW09] were used to detect features in CFD data [LT11]. Parallel coordinates can also be used to navigate high-dimensional parameter spaces in volume-rendering applications [PBM05,TPM05,CBS*07] and to design multidimensional transfer functions [GXY11]. The exploration of high-dimensional parameter spaces is another frequent application of parallel coordinates, e.g. for aircraft- and car design [GBS*99, BPFG11] or diesel injection systems [MJJ*05, MHDG11]. Parallel coordinates were further applied for process control in chemical plants [AW06, CWVB11, DEN12], for alarm filtering for industrial systems [AR11], and for air traffic conrol [Ins01]. Another security related application of parallel coordinates is the detection of network attacks [CLK09, KLCM09, TNSa11, PMSN11].

8. Conclusion and Future Work

Although parallel coordinates were first published as early as 1885, they have become popular in the visualization community only recently. In many other domains, parallel coordinates are either unknown or considered an expert tool that requires much effort to work with. This report illustrates the variety of research conducted in modeling, creating, understanding, and interacting with parallel coordinates. Our taxonomy covers most of the aspects of parallel coordinates that are not covered in other surveys [Ins09,Mou11] and provides an abstraction for the classification of research topics. The taxonomy can be used to identify areas that require further research and to find techniques that have been successfully applied for a variety of challenges and tasks. However, we deliberately retained from evaluating the techniques with respect to their applicability, correctness, usability for real datasets, and performance compared to other visualization techniques, as this would be out of the scope of this state-of-the-art report.

In the future, a quantification of the research conducted within the individual topics would be desirable to improve the white-spot analysis and find underrepresented research areas. Also, an evaluation of existing tools and systems that employ parallel coordinates is required to identify shortcomings and technical issues with respect to the implementation of the techniques surveyed here.

References

[AA99] Avidan T., Avidan S.: Parallax-a data mining tool based on parallel coordinates. *Computational Statistics 14*, 1 (1999), 79–90. 2, 11, 13

[AA01] Andrienko G., Andrienko N.: Constructing parallel coordinates plot for problem solving. In *Proceedings of the 1st International Symposium on Smart Graphics* (2001), ACM New York, pp. 9–14. 10, 12, 17

[AA04] Andrienko G., Andrienko N.: Parallel coordinates for exploring properties of subsets. In *Proceesings of the Second International Conference on Coordinated and Multiple Views in Exploratory Visualization* (2004), IEEE, pp. 93–104. 7, 13

[ABK98] Ankerst M., Berchtold S., Keim D. A.: Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Proceedings of the IEEE Symposium on Information Visualization* (1998), pp. 52–60. 14, 15

[AdOL04] Artero A. O., de Oliveira M. C. F., Levkowitz H.: Uncovering clusters in crowded parallel coordinates visualizations. In *Proceedings of the IEEE Symposium on Information Visualization* (2004), Ward M. O., Munzner T., (Eds.), pp. 81–88. 8

[AEL*09] ALBUQUERQUE G., EISEMANN M., LEHMANN D., THEISEL H., MAGNOR M.: Quality-based visualization matrices. In *Proceedings of Vision, Modeling, and Visualization* (2009), pp. 341–349. 14

[And72] ANDREWS D. F.: Plots of high-dimensional data. *Biometrics 28*, 1 (1972), 125–136. 1, 7

[AR11] AZHAR S. B., RISSANEN M. J.: Evaluation of parallel coordinates for interactive alarm filtering. In *Proceedings of the 15th International Conference on Information Visualisation* (2011), pp. 102–109. 10, 17

[AW06] ALBAZZAZ H., WANG X. Z.: Historical data analysis based on plots of independent and parallel coordinates and statistical control limits. *Journal of Process Control 16*, 2 (2006), 103–114. 17

[BBP08] BLAAS J., BOTHA C., POST F.: Extensions of parallel coordinates for interactive exploration of large multi-timepoint data sets. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1436–1451. 17

[BC87] BECKER R. A., CLEVELAND W. S.: Brushing scatterplots. *Technometrics 29*, 2 (1987), 127–142. 10, 11

[BCW87] BECKER R. A., CLEVELAND W. S., WILKS A. R.: Dynamic graphics for data analysis. *Statistical Science 2*, 4 (1987), 355–383. 10

[BCW88] BECKER R. A., CHAMBERS J. M., WILKS A. R.: *The New S Language: A Programming Environmentfor Data Analysis and Graphics*. Wadsworth & Brooks/Cole Advanced Books & Software, 1988. 14

[Bec97] BECKER O. M.: Representing protein and peptide structures with parallel-coordinates. *Journal of Computational Chemistry 18*, 15 (1997), 1893–1902. 17

[BH03] BERTHOLD M., HALL L.: Visualizing fuzzy points in parallel coordinates. *IEEE Transactions on Fuzzy Systems 11*, 3 (2003), 369 – 374. 7

[BKH05] BENDIX F., KOSARA R., HAUSER H.: Parallel sets: Visual analysis of categorical data. In *Proceedings of the IEEE Symposium on Information Visualization* (2005). 7

[BMGK08] BARSKY A., MUNZNER T., GARDY J., KINCAID R.: Cerebral: Visualizing multiple experimental conditions on a graph with biological context. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1253–1260. 17

[BPFG11] BERGER W., PIRINGER H., FILZMOSER P., GRÖLLER E.: Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. *Computer Graphics Forum 30*, 3 (2011), 911–920. 17

[BS04] BARLOW N., STUART L.: Animator: a tool for the animation of parallel coordinates. In *Proceedings of the Eighth International Conference on Information Visualisation (IV'04)* (2004), pp. 725–730. 16

[BW08] BACHTHALER S., WEISKOPF D.: Continuous scatterplots. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1428–1435. 8

[CBS*07] CRIDER M., BERGNER S., SMYTH T. N., MÖLLER T., TORY M. K., KIRKPATRICK A. E., WEISKOPF D.: A mixing board interface for graphics and visualization applications. In *Proceedings of Graphics Interface 2007* (2007), pp. 87–94. 11, 17

[Che03] CHEN H.: Compound brushing. In *Proceedings of the IEEE Symposium on Information Visualization* (2003), IEEE, pp. 181–188. 11

[CLK09] CHOI H., LEE H., KIM H.: Fast detection and visualization of network attacks on parallel coordinates. *Computers & Security 28*, 5 (2009), 276–288. 17

[CM84] CLEVELAND W. S., MCGILL R.: Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association 79*, 387 (1984), 531–554. 13, 15

[CM87] CLEVELAND W. S., MCGILL R.: Graphical perception: The visual decoding of quantitative information on graphical displays of data. *Journal of the Royal Statistical Society. Series A (General) 150*, 3 (1987), 192–229. 13

[CVW09] COLLINS C., VIEGAS F., WATTENBERG M.: Parallel tag clouds to explore and analyze faceted text corpora. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology* (2009), Stasko J., Jarke J. van Wijk, (Eds.), pp. 91–98. 9

[CvW11] CLAESSEN J. H. T., VAN WIJK J. J.: Flexible linked axes for multivariate data visualization. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2310–2316. 14

[CWVB11] COMFORT J. R., WARNER T. R., VARGO E. P., BASS E. J.: Parallel coordinates plotting as a method in process control hazard identification. In *Proceedings of the 2011 IEEE Systems and Information Engineering Design Symposium* (2011), pp. 152–157. 17

[DCK12] DASGUPTA A., CHEN M., KOSARA R.: Conceptualizing visual uncertainty in parallel coordinates. *Computer Graphics Forum 31*, 3 (2012), 1015–1024. 16

[DEN12] DUNIA R., EDGAR T. F., NIXON M.: Process monitoring using principal components in parallel coordinates. *AIChE Journal* (2012), 1–12. 2, 17

[DF83] DIACONIS P., FRIEDMAN J.: M and n plots. In *Recent Advances in Statistics* (1983), Academic Press, New York, pp. 425–447. 1

[DHH11] DUBSKA M., HEROUT A., HAVEL J.: PClines - line detection using parallel coordinates. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2011). 6, 10, 12

[DHNB09] DIETZSCH J., HEINRICH J., NIESELT K., BARTZ D.: SpRay: a visual analytics approach for gene expression data. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology* (2009), pp. 179–186. 2, 16, 17

[DK10] DASGUPTA A., KOSARA R.: Pargnostics: Screen-space metrics for parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1017–1026. 8, 10, 14

[DMH04] DOLEISCH H., MUIGG P., HAUSER H.: *Interactive Visual Analysis of Hurricane Isabel with SimVis*. Tech. Rep. TR-VRVis-2004-058, VRVis Research Center, 2004. 17

[DWA10] DANG T. N., WILKINSON L., ANAND A.: Stacking graphic elements to avoid over-plotting. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1044 –1052. 10

[ED06] ELLIS G., DIX A.: Enabling automatic clutter reduction in parallel coordinate plots. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 717–724. 12, 13

[ED07] ELLIS G., DIX A.: A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1216 –1223. 12, 13

[Eds03] EDSALL R. M.: The parallel coordinate plot in action: design and use for geographic visualization. *Computational Statistics & Data Analysis 43*, 4 (2003), 605–619. 17

[EF10] ELMQVIST N., FEKETE J. D.: Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Transactions on Visualization and Computer Graphics 16*, 3 (2010), 439–454. 13, 16

[ESBB98] EISEN M. B., SPELLMAN P. T., BROWN P. O., BOTSTEIN D.: Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences 95*, 25 (1998), 14863–14868. 17

[FCI05] FANEA E., CARPENDALE S., ISENBERG T.: An interactive 3D integration of parallel coordinates and star glyphs. In *Proceedings of the IEEE Symposium on Information Visualization* (2005), Stasko J., Ward M., (Eds.), IEEE Computer Society. 10

[FdOL03] FERREIRA DE OLIVEIRA M., LEVKOWITZ H.: From visual data exploration to visual data mining: A survey. *IEEE Transactions on Visualization and Computer Graphics 9*, 3 (2003), 378–394. 10

[FFT75] FISHERKELLER M. A., FRIEDMAN J. H., TUKEY J. W.: PRIM-9: an interactive multi-dimensional data display and analysis system. In *Proceedings of Pacific 75, ACM Regional Conference* (1975). 10

[FKLI10] FENG D., KWOCK L., LEE Y., II R. M. T.: Matching visual saliency to confidence in plots of uncertain data. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 980–989. 5, 11

[FPSS96a] FAYYAD U., PIATETSKY-SHAPIRO G., SMYTH P.: From data mining to knowledge discovery in databases. *AI Magazine 17*, 3 (1996), 37. 2, 6

[FPSS96b] FAYYAD U., PIATETSKY-SHAPIRO G., SMYTH P.: The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM 39*, 11 (1996), 27–34. 16

[FR11] FERDOSI B. J., ROERDINK J. B. T.: Visualizing High-Dimensional structures by dimension ordering and filtering using subspace analysis. *Computer Graphics Forum 30*, 3 (2011), 1121–1130. 14, 15

[FWR99a] FUA Y.-H., WARD M., RUNDENSTEINER E.: Navigating hierarchies with structure-based brushes. In *Proceedings of the IEEE Symposium on Information Visualization* (1999), Graham Wills, Daniel Keim, (Eds.), IEEE Computer Society Press, pp. 58–64. 11

[FWR99b] FUA Y. H., WARD M. O., RUNDENSTEINER E. A.: Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of IEEE Visualization* (1999), Ebert D. S., Gross M., Hamann B., (Eds.), pp. 43–50. 7, 11, 12, 13, 15

[FWR00] FUA Y.-H., WARD M., RUNDENSTEINER E.: Structure-based brushes: a mechanism for navigating hierarchically organized data and information spaces. *IEEE Transactions on Visualization and Computer Graphics 6*, 2 (June 2000), 150 –159. 11

[GBS*99] GOEL A., BAKER C., SHAFFER C., GROSSMAN B., HAFTKA R., MASON W., WATSON L.: VizCraft: a multidimensional visualization tool for aircraft configuration design. In *Proceedings Visualization '99* (1999), pp. 425–555. 17

[GK03] GRAHAM M., KENNEDY J.: Using curves to enhance parallel coordinate visualisations. In *Proceedings of the Seventh International Conference on Information Visualization* (2003), pp. 10–16. 6, 7

[GPL*11] GENG Z., PENG Z., LARAMEE R., WALKER R., ROBERTS J.: Angular histograms: Frequency-based visualizations for large, high dimensional data. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2572 –2580. 9, 17

[Guo03] GUO D.: Coordinating computational and visual approaches for interactive feature selection and multivariate clustering. *Information Visualization 2*, 4 (2003), 232–246. 14

[GW12] GEHLENBORG N., WONG B.: Points of view: Power of the plane. *Nature Methods 9*, 10 (2012), 935–935. 13

[GWY*11] GUO H., WANG Z., YU B., ZHAO H., YUAN X.: TripVista: triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection. In *Proceedings of the Pacific Visualization Symposium* (2011), pp. 163–170. 17

[GXY11] GUO H., XIAO H., YUAN X.: Multi-dimensional transfer function design based on flexible dimension projection embedded in parallel coordinates. In *Proceedings of the IEEE Pacific Visualization Symposium 2011* (2011), Giuseppe Di Battista, Jean-Daniel Fekete, Huamin Qu, (Eds.). 17

[Har75] HARTIGAN J.: Printer graphics for clustering. *Journal of Statistical Computation and Simulation 4*, 3 (1975), 187–213. 11

[HBW11] HEINRICH J., BACHTHALER S., WEISKOPF D.: Progressive splatting of continuous scatterplots and parallel coordinates. *Computer Graphics Forum 30*, 3 (2011), 653–662. 5, 9

[HC00] HAN J., CERCONE N.: RuleViz: a model for visualizing knowledge discovery process. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2000), pp. 244–253. 7

[HHD*12] HASENAUER J., HEINRICH J., DOSZCZAK M., SCHEURICH P., WEISKOPF D., ALLGÖWER F.: A visual analytics approach for models of heterogeneous cell populations. *EURASIP Journal on Bioinformatics and Systems Biology 2012*, 1 (2012), 4. 13, 17

[HLD02] HAUSER H., LEDERMANN F., DOLEISCH H.: Angular brushing of extended parallel coordinates. In *Proceedings of the IEEE Symposium on Information Visualization* (2002), Wong P. C., Andrews K., (Eds.), pp. 127–130. 9, 11

[HLKW12] HEINRICH J., LUO Y., KIRKPATRICK A. E., WEISKOPF D.: Evaluation of a bundling technique for parallel coordinates. In *Proceedings of International Conference on Information Visualization Theory and Applications* (2012), Springer. 6, 7, 10, 13, 15

[HM90] HABER R. B., MCNABB D. A.: Visualization idioms: A conceptual model for scientific visualization systems. *Visualization in scientific computing 74* (1990), 93. 5

[HO10] HURLEY C. B., OLDFORD R. W.: Pairwise display of high-dimensional information via eulerian tours and hamiltonian decompositions. *Journal of Computational and Graphical Statistics 19* (2010), 861–886. 14, 15

[Hol06] HOLTEN D.: Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 741–748. 7

[HSW12] HEINRICH J., STASKO J., WEISKOPF D.: The parallel coordinates matrix. In *EuroVis - Short Papers* (2012), pp. 37–41. 14

[Hur04] HURLEY C. B.: Clustering visualizations of multidimensional data. *Journal of Computational and Graphical Statistics 13*, 4 (2004), 788–806. 14, 15

[HW09] HEINRICH J., WEISKOPF D.: Continuous parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1531–1538. 2, 8, 17

[HW10] HOLTEN D., WIJK J. J. V.: Evaluation of cluster identification performance for different PCP variants. *Computer Graphics Forum 29*, 3 (2010), 793–802. 6, 7, 8, 9, 10

[IA99] INSELBERG A., AVIDAN T.: The automated multidimensional detective. In *Proceedings of the IEEE Symposium on Information Visualization* (1999), Graham Wills, Daniel Keim, (Eds.), pp. 112–119. 15

[ICD97] INSELBERG A., CHATTERJEE A., DIMSDALE B.: System using parallel coordinates for automated line detection in noisy images, 1997. U.S. Classification: 382/281 International Classification: G06K 936; G06K 900; G06F 1714. 6, 10, 12

[ID90] INSELBERG A., DIMSDALE B.: Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st Conference on Visualization* (1990), Kaufman A., (Ed.), pp. 361–378. 4

[Ins85] INSELBERG A.: The plane with parallel coordinates. *The Visual Computer 1*, 4 (1985), 69–91. 1, 4, 7, 14

[Ins01] INSELBERG A.: Conflict detection and planar resolution for air traffic control. In *Proceedings of IEEE Intelligent Transportation Systems* (2001), pp. 1200–1205. 16, 17

[Ins09] INSELBERG A.: *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer, 2009. 1, 2, 3, 4, 5, 6, 11, 12, 17

[JC08] JOHANSSON J., COOPER M.: A screen space quality method for data abstraction. *Computer Graphics Forum 27*, 3 (2008), 1039–1046. 10

[JCJ05] JOHANSSON J., COOPER M., JERN M.: 3-dimensional display for clustered multi-relational parallel coordinates. In *Proceedings of the 9th International Conference on Information Visualization* (2005), Ebad Banissi, Muhammad Sarfraz, Jonathan C. Roberts, Bowen Loften, Anna Ursyn, Remo Aslak Burkhard, Angela Lee, Gennady Andrienko, (Eds.), IEEE Computer Society Press, pp. 188–193. 10

[JFLC08] JOHANSSON J., FORSELL C., LIND M., COOPER M.: Perceiving patterns in parallel coordinates: determining thresholds for identification of relationships. *Information Visualization 7*, 2 (2008), 152–162. 10

[JKL*09] JOHANSSON S., KNAVING K., LANE A., JERN M., JOHANSSON J.: Interactive exploration of ingredient mixtures using multiple coordinated views. In *Proceedings of the 13th International Conference on Information Visualisation* (2009), IEEE Computer Society Press, pp. 210–218. 14, 15

[JLC07] JOHANSSON J., LJUNG P., COOPER M.: Depth cues and density in temporal parallel coordinates. In *Proceedings of the Eurographics/IEEE-VGTC Symposium on Visualization* (2007), pp. 35–42. 7, 16

[JLJC05] JOHANSSON J., LJUNG P., JERN M., COOPER M.: Revealing structure within clustered parallel coordinates displays. In *Proceedings of the IEEE Symposium on Information Visualization* (2005), pp. 125–132. 8

[KBH06] KOSARA R., BENDIX F., HAUSER H.: Parallel sets: Interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization and Computer Graphics 12*, 4 (2006), 558–568. 7

[KERC09] KEEFE D., EWERT M., RIBARSKY W., CHANG R.: Interactive coordinated multiple-view visualization of biomechanical motion data. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1383–1390. 17

[KLCM09] KIM H., LEE I., CHO J., MOON J.: Visualization of network components for attack analysis. In *Proceedings of the IEEE Symposium on Computational Intelligence in Cyber Security* (2009), pp. 1–8. 17

[KZZM12] KUANG X., ZHANG H., ZHAO S., MCGUFFIN M. J.: Tracing tuples across dimensions: A comparison of scatterplots and parallel coordinate plots. *Computer Graphics Forum 31*, 3 (2012), 1365–1374. 10

[LJC09] LIND M., JOHANSSON J., COOPER M.: Many-to-many relational parallel coordinates displays. In *Proceedings of the 13th International Conference on Information Visualisation* (2009), pp. 25–31. 10, 15

[LMP05] LANZENBERGER M., MIKSCH S., POHL M.: Exploring highly structured data: A comparative study of stardinates and parallel coordinates. In *Proceedings of the Ninth International Conference on Information Visualisation (IV'05)* (2005), pp. 312–320. 10

[LMvW10] LI J., MARTENS J.-B., VAN WIJK J. J.: Judging correlation from scatterplots and parallel coordinate plots. *Information Visualization 9*, 1 (2010), 13–30. 10

[LT11] LEHMANN D., THEISEL H.: Features in continuous parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 1912–1921. 10, 17

[Mau85] 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, 1885. 1, 13

[MHDG11] MUIGG P., HADWIGER M., DOLEISCH H., GRÖLLER E.: Visual coherence for large-scale line-plot visualizations. *Computer Graphics Forum 30*, 3 (2011), 643–652. 9, 17

[MJJ*05] MATKOVIC K., JELOVIC M., JURIC J., KONYHA Z., GRACANIN D.: Interactive visual analysis and exploration of injection systems simulations. In *Proceedings of IEEE Visualization* (2005), pp. 391–398. 17

[MM08] MCDONNELL K., MUELLER K.: Illustrative parallel coordinates. *Computer Graphics Forum 27*, 3 (2008), 1031–1038. 7, 13

[MMD*10] MEYER M., MUNZNER T., DEPACE A., PFISTER H., MEMBER S.: MulteeSum : A tool for comparative spatial and temporal gene expression data. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 908–917. 17

[Mou09] MOUSTAFA R. E. A.: QGPCP: quantized generalized parallel coordinate plots for large multivariate data visualization. *Journal of Computational and Graphical Statistics 18*, 1 (2009), 32–51. 5, 9

[Mou11] MOUSTAFA R.: Parallel coordinate and parallel coordinate density plots. *Wiley Interdisciplinary Reviews: Computational Statistics 3*, 2 (2011), 134–148. 5, 7, 10, 15, 17

[MW91] MILLER J. J., WEGMAN E. J.: Construction of line densities for parallel coordinate plots. In *Computing and Graphics in Statistics*, Buja A., Tukey P. A., (Eds.), vol. 36. Springer-Verlag, New York, 1991, pp. 107–123. 2, 5, 8

[MW95] MARTIN A. R., WARD M. O.: High dimensional brushing for interactive exploration of multivariate data. In *Proceedings of the 6th IEEE Visualization Conference* (1995), pp. 271–278. 11, 13

[MW02] MOUSTAFA R., WEGMAN E. J.: On some generalization to parallel coordinate plots. In *Seeing a Million – A Data Visualization Workshop* (2002), pp. 41–48. 6, 7

[MWS*10] MEYER M., WONG B., STYCZYNSKI M., MUNZNER T., PFISTER H.: Pathline: A tool for comparative functional genomics. *Computer Graphics Forum 29*, 3 (2010), 1043–1052. 17

[NH06] NOVOTNY M., HAUSER H.: Outlier-preserving Focus+Context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 893–900. 2, 7, 8, 14

[Nov04] NOVOTNY M.: Visually effective information visualization of large data. In *Proceedings of the 8th Central European Seminar on Computer Graphics* (2004). 8

[PBM05] PRADHAN K., BARTZ D., MUELLER K.: *SignatureSpace: A Multidimensional, Exploratory Approach for the Analysis of Volume Data*. Technical report, University of Tuebingen, 2005. 17

[Pea95] PEARSON K.: Contributions to the mathematical theory of evolution. II. skew variation in homogeneous material. *Philosophical Transactions of the Royal Society of London. 186* (1895), 343–414. 8

[PMSN11] PROMRIT N., MINGKHWAN A., SIMCHAROEN S., NAMVONG N.: Multi-dimensional visualization for network forensic analysis. In *Networked Computing (INC), 2011 The 7th International Conference on* (Sept. 2011), pp. 68 –73. 17

[PWR04] PENG W., WARD M., RUNDENSTEINER E.: Clutter reduction in multi-dimensional data visualization using dimension reordering. In *Proceedings of the IEEE Symposium Information Visualization* (2004), pp. 89–96. 14

[QCX*07] QU H., CHAN W.-Y., XU A., CHUNG K.-L., LAU K.-H., GUO P.: Visual analysis of the air pollution problem in hong kong. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1408–1415. 9, 14, 15

[RTT03] RODRIGUES J., TRAINA A., TRAINA C.: Frequency plot and relevance plot to enhance visual data exploration. In *Proceedings of the XVI Brazilian Symposium on Computer Graphics and Image Processing* (2003), pp. 117–124. 8

[RWK*06] RÜBEL O., WEBER G. H., KERÄNEN S. V. E., FOWLKES C. C., HENDRIKS C. L. L., SIMIRENKO L., SHAH N. Y., EISEN M. B., BIGGIN M. D., HAGEN H., SUDAR D., MALIK J., KNOWLES D. W., HAMANN B.: PointCloudXplore: visual analysis of 3D gene expression data using physical views and parallel coordinates. In *Proceedings of the Eurographics/ IEEE-VGTC Symposium on Visualization* (2006), pp. 203–210. 10, 17

[RZH12] ROSENBAUM R., ZHI J., HAMANN B.: Progressive parallel coordinates. In *Proceedings of the IEEE Pacific Visualization Symposium 2012* (2012), pp. 25–32. 13

[Shn94] SHNEIDERMAN B.: Dynamic queries for visual information seeking. *IEEE Software 11*, 6 (1994), 70–77. 13

[Shn96] SHNEIDERMAN B.: The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages* (1996), pp. 336–343. 2, 10

[Sii00] SIIRTOLA H.: Direct manipulation of parallel coordinates. In *Proceedings of the IEEE International Conference on Information Visualization* (2000), IEEE, pp. 373–378. 9, 13

[SLHR09] SIIRTOLA H., LAIVO T., HEIMONEN T., RAIHA K. J.: Visual perception of parallel coordinate visualizations. In *Proceedings of the 13th International Conference on Information Visualisation* (2009), IEEE, pp. 3–9. 10

[Smi95] SMITH A. R.: *A Pixel Is Not A Little Square, A Pixel Is Not A Little Square, A Pixel Is Not A Little Square! (And a Voxel is Not a Little Cube*. Tech. rep., Technical Memo 6, Microsoft Research, 1995. 8

[SR06] SIIRTOLA H., RAIHA K.: Interacting with parallel coordinates. *Interacting with Computers 18*, 6 (2006), 1278–1309. 10, 13

[SS05] SCOTT D., SAIN S.: Multidimensional density estimation. *Handbook of Statistics 24*, 2004 (2005), 229–261. 8

[SSJKF09] STEED C. A., SWAN J. E., JANKUN-KELLY T., FITZPATRICK P. J.: Guided analysis of hurricane trends using statistical processes integrated with interactive parallel coordinates. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology 2009* (2009), pp. 19–26. 2

[TAE*09] TATU A., ALBUQUERQUE G., EISEMANN M., SCHNEIDEWIND J., THEISEL H., MAGNORK M., KEIM D.: Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology* (2009), pp. 59 –66. 14

[TAE*11] TATU A., ALBUQUERQUE G., EISEMANN M., BAK P., THEISEL H., MAGNOR M., KEIM D.: Automated analytical methods to support visual exploration of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics 17*, 5 (2011), 584–597. 14

[tCM07] TEN CAAT M., MAURITS N. M.: Design and evaluation of tiled parallel coordinate visualization of multichannel EEG data. *IEEE Transactions on Visualization and Computer Graphics 13*, 1 (2007), 70–79. 17

[TCMR05] TEN CAAT M., MAURITS N. M., ROERDINK J.: Tiled parallel coordinates for the visualization of time-varying multichannel EEG data. In *Proceedings of the Eurographics/IEEE VGTC Symposium on Data Visualization* (2005), p. 61. 17

[TFA*11] TAM G. K. L., FANG H., AUBREY A. J., GRANT P. W., ROSIN P. L., MARSHALL D., CHEN M.: Visualization of Time-Series data in parameter space for understanding facial dynamics. *Computer Graphics Forum 30*, 3 (2011), 901–910. 2

[TFH11] TURKAY C., FILZMOSER P., HAUSER H.: Brushing dimensions a dual visual analysis model for high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2591–2599. 11

[The00] THEISEL H.: Higher order parallel coordinates. In *Vision, Modeling, and Visualization* (2000), pp. 415–420. 6, 7, 15

[The03] THEUS M.: Interactive data visualization using mondrian. *Journal of Statistical Software 7*, 11 (2003), 1–9. 9

[The06] THERON R.: Visual analytics of paleoceanographic conditions. In *Proceedings of the IEEE Symposium On Visual Analytics Science And Technology* (2006), pp. 19–26. 16

[TNSa11] TRICAUD S., NANCE K., SAADE&#X0301 AND P.: Visualizing network activity using parallel coordinates. In *Proceedings of the 44th Hawaii International Conference on System Sciences* (2011), pp. 1–8. 17

[TPM05] TORY M., POTTS S., MÖLLER T.: A parallel coordinates style interface for exploratory volume visualization. *IEEE Transactions on Visualization and Computer Graphics 11*, 1 (2005), 71–80. 17

[UVW03] UNWIN A., VOLINSKY C., WINKLER S.: Parallel coordinates for exploratory modelling analysis. *Computational Statistics & Data Analysis 43*, 4 (2003), 553–564. 2

[VMCJ10] VIAU C., MCGUFFIN M. J., CHIRICOTA Y., JURISICA I.: The FlowVizMenu and parallel scatterplot matrix: Hybrid multidimensional visualizations for network exploration. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1100–1108. 14

[WAG06] WILKINSON L., ANAND A., GROSSMAN R.: High-dimensional visual analytics: Interactive exploration guided by pairwise views of point distributions. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (2006), 1363–1372. 14

[War94] WARD M. O.: XmdvTool: integrating multiple methods for visualizing multivariate data. In *Proceedings of the Conference on Visualization '94* (1994), pp. 326–333. 11

[War04] WARE C.: *Information visualization: perception for design*, vol. 22. Morgan Kaufmann, 2004. 15

[WB96] WONG P. C., BERGERON R. D.: Multiresolution multidimensional wavelet brushing. In *Proceedings Visualization '96* (1996), pp. 141–148. 11

[Weg90] WEGMAN E. J.: Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association 85*, 411 (1990), 664–675. 1, 4, 14

[WGJL12] WALKER J., GENG Z., JONES M., LARAMEE R. S.: Visualization of large, time-dependent, abstract data with integrated spherical and parallel coordinates. In *EuroVis - Short Papers* (2012), Eurographics Association, pp. 43–47. 9, 10

[WL97] WEGMAN E. J., LUO Q.: High dimensional clustering using parallel coordinates and the grand tour. In *Classification and Knowledge Organization* (1997), pp. 93–102. 2

[WLG97] WEGENKITTL R., LOFFELMANN H., GROLLER E.: Visualizing the behaviour of higher dimensional dynamical systems. In *Proceedings Visualization '97* (1997), IEEE, pp. 119–125. 10

[YGX*09] YUAN X., GUO P., XIAO H., ZHOU H., QU H.: Scattering points in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1001–1008. 6

[YPWR03] YANG J., PENG W., WARD M. O., RUNDENSTEINER E. A.: Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In *Proceedings of the IEEE Symposium on Information Visualization* (2003), pp. 105–112. 14, 15

[ZCQ*09] ZHOU H., CUI W., QU H., WU Y., YUAN X., ZHUO W.: Splatting the lines in parallel coordinates. *Computer Graphics Forum 28*, 3 (2009), 759–766. 9

[ZK12] ZHAO X., KAUFMAN A.: Structure revealing techniques based on parallel coordinates plot. *The Visual Computer 28*, 6 (2012), 541–551. 14

[ZLTS03] ZHAO K., LIU B., TIRPAK T., SCHALLER A.: Detecting patterns of change using enhanced parallel coordinates visualization. In *Proceedings of the Third IEEE International Conference on Data Mining* (2003), pp. 747–750. 14

[ZMH*09] ZACHOW S., MUIGG P., HILDEBRANDT T., DOLEISCH H., HEGE H.-C.: Visual exploration of nasal airflow. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1407–1414. 17

[ZMM12] ZHANG Z., MCDONNELL K., MUELLER K.: A network-based interface for the exploration of high-dimensional data spaces. In *Proceedings of the Pacific Visualization Symposium* (2012), pp. 17–24. 7, 14, 15

[ZYQ*08] ZHOU H., YUAN X., QU H., CUI W., CHEN B.: Visual clustering in parallel coordinates. *Computer Graphics Forum 27*, 3 (2008), 1047–1054. 7