Vision, Modeling, and Visualization (2021)
B. Andres, M. Campen, and M. Sedlmair (Eds.)

Axes Bundling and Brushing in Star Coordinates

Hennes Rave, Vladimir Molchanov, and Lars Linsen

Westfälische Wilhelms-Universität Münster, Germany

Abstract

Visual analysis of multidimensional data commonly involves dimensionality reduction to project the data samples into a lower-dimensional visual space. Star coordinates (SC) provide a means to explore the multidimensional data distribution by interactively changing the linear projection matrix. While SC have the advantages of being intuitive, allowing for relating the data samples to their original dimensions, having low computation costs, and scaling well with the number of data samples, they have the disadvantages of not scaling well to larger number of dimensions and being restricted to linear projections. We address these short-comings by introducing novel SC interactions. First, interactive bundling of axes is proposed to reduce the number of dimensions. While bundles are fully customizable, the bundling interactions are supported by visualizations of correlation matrices and hierarchical axes clustering dendrograms. Second, we enhance classical region brushing in SC projections with axes brushing, which allows for multidimensional cluster selection, even if two (separable) clusters are projected to the same area of the visible space. Axes brushing is supported by visualizing 1D histograms of data distributions along the SC axes. Our brushing interactions alleviate the restriction of SC to linear projections. The integration of histograms into SC also eases other interactions such as moving axes to change the projection matrix. A user study evaluates how analysis tasks for labeled and unlabeled multidimensional data can benefit from our extensions.

CCS Concepts

• ***Human-centered computing*** → ***Visual analytics;***

1. Introduction

Analysis of multidimensional data often includes a *dimensionality reduction* (DR) step. Interactive steering of the parameters of the projection methods is often important for understanding the interplay of the parameters, dimensions, and projection spaces. Therefore, interactive visual analysis systems are desired for the exploration of multidimensional data. Star coordinates (SC) are a popular tool for multidimensional data projection. SC support interactive modification of the parameters of the linear projection by dragging and dropping the end points of the SC axes, which is an *intuitive* operation for exploring the space of linear projections. A re-projection of data using the updated operator can be performed at interactive rates even for relatively large datasets, since an application of a linear projection is *computationally lightweight*. The visual encoding via point rendering also *scales well to a larger number of points*, at least, in comparison to other multidimensional data visualization methods. These advantages make SC interaction a desirable approach for multidimensional data exploration.

However, there are downsides to using SC. Most prominently, they do *not scale well to large numbers of attributes*. From our experience, SC interactions are intuitive and effective for roughly up to 10 axes, but become impractical when the number of attributes becomes larger due to overplotting of axes and an overwhelming number of possibly interesting axes configurations. We address this issue by proposing an *axes-bundling strategy*, which reduces the large number of dimension axes to a customizable number of axes bundles. Interactive bundling is performed in a data-driven manner and supported by a visualization of the dimensions' correlation matrix and a dendrogram of a hierarchical dimension clustering. The user can interactively adjust the amount of bundling and the bundling strength, which induces a nonlinear component. Bundles can be handled and interacted with just like original axes in SC.

Given SC axes and bundles thereof, we propose flexible brushing interactions, where the commonly supported region brushing is complemented with *axes brushing*, which allows for brushing of (separable) clusters, even if they are not separated in the currently selected projection or in case of visual clutter. Axes brushing is supported by plotting 1D histograms along all axes or bundles. The 1D histograms allow for a more precise brushing as well as for more educated selections of axes/bundles configurations. To avoid overplotting of the (enriched) axes/bundles and the projected point distributions, we render them in separate linked views.

Our main contributions can be summarized as follows: (1) SC interactions with bundled axes for better scalability to larger number of dimensions, (2) an interactive data-driven bundling strategy for SC axes, (3) a non-linear SC projection layout for bundled axes with different bundling strengths, (4) more flexible SC brushing by complementing region brushing with axis and bundle brush-

DOI: 10.2312/vmv.20211365

ing, (5) enrichment of SC axes and bundles with 1D histograms for improved interaction experience, and (6) numerical tests and user studies demonstrating the effectiveness of our approaches.

2. Related Work

DR methods (pre)-process multidimensional data, usually optimizing the set of attributes by deleting negligible dimensions or by computing new significant data features. Sufficiently small number of attributes can be handled interactively. Recently, Sacha et al. [SZS*17] proposed a classification of interactive visualization methods and systems which exploit DR algorithms.

SC [Kan00, Kan01] are a prominent example of an interactive linear DR technique. SC allow for intuitive exploration of the projection space by changing the applied projection operator. Teoh and Ma [TM03] solved data classification tasks using SC. Molchanov et al. [MFL13a] proposed a continuous representation of the projected space when exploring volumetric multidimensional data, including a progressive rendering approach [MFL13b]. Lehmann and Theisel [LT16] generalized SC and RadViz [HGM*97] by minimizing a distortion measure and providing a data-dependent magic lens. Molchanov and Linsen [ML14] solved an inverse problem of finding a least-squares optimal SC constellation for a desired projection layout.

Overpopulated regions in scatterplots are subject to *overplotting*. Overdraw makes the data structure difficult to discern. The issue of overplotting was addressed over decades. A taxonomy of clutter reduction methods for information visualization was developed by Ellis and Dix [ED07]. Recently, Raidou et al. [RGE19] proposed a scatterplot relaxation technique based on pixel-based mappings. Zanabria et al. [ZNGN16] mitigated the SC visual clutter by grouping the SC axes using a clustering mechanism. In our work, we perform the next step and bundle the grouped axes, which reduces visual clutter and allows for interacting with resulting bundles. Hierarchical edge *bundling* was proposed by Holten [Hol06] to reduce clutter in graphs. Trajectory bundling [TE10, EHP*11, HET12, HEF*14, LHT17] is an effective technique for visual aggregation of (poly)linear elements in order to improve readability of plots.

Parallel coordinates [Ins85, ID90, Ins09] became a popular tool for multidimensional data visualization and exploration. Bundling in parallel coordinates was studied by McDonnell and Mueller [MM08], Heinrich et al. [HLKW12], and Palmas et al. [PBO*14]. Data selection in parallel coordinates is usually performed by area, lasso, or angular brushing, probing, and composite AND/OR brushes [REB*16]. Recently, Roberts et al. [RLS*19] proposed a set of high-order smart *brushing* techniques for parallel coordinates. While brushing on axes is common in parallel coordinates, brushing in SC is usually performed by selecting a region with a lasso tool. We complement this by also allowing for axes brushing, which we support by histogram visualizations.

3. Background

3.1. Star Coordinates

An arbitrary linear operator L mapping n-dimensional data into two-dimensional space can be represented by a $2 \times n$ matrix $L = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$, $\mathbf{z}_i = (x_i, y_i)^T$. The mapping itself is then a multiplication of matrix L with the multidimensional data samples $\mathbf{a}_j$ by $\mathbf{p}_j = L \cdot \mathbf{a}_j$, $j = 1, \ldots, m$, where $\mathbf{p}_j$ is the projected sample and m is the number of samples.

The two-dimensional vectors $\mathbf{z}_i$ corresponding to columns of L can be visually encoded as line segments starting at the origin and ending at points with coordinates $\mathbf{z}_i$. They form the axes of the *SC* system, which uniquely represents a given linear operator and can be used for interactive steering of L by the user. When the end point of an SC axis is interactively moved, the system changes the entries of the respective vector $\mathbf{z}_i$. An updated matrix L is immediately applied for remapping the multidimensional data. Thus, the user observes which impact the interaction with the SC axis has on the resulting projection layout. SC allow the user to perform effective data analysis for tasks such as outlier detection, cluster separation, feature sensitivity analysis, projection layout design [ML14], and classification [TM03, MCL15].

The *initial SC configuration*, by default, spreads the axes uniformly on the unit circle, which provides an easy access to each individual axis for the user interaction. It is convenient to depict the unit circle in the SC widget to have a visual reference for the lengths of the SC axes. Alternatively, the initial configuration can be chosen to represent the two leading dimensions of the principal component analysis, i.e., to reflect an intrinsic structure of the multidimensional data. Another option is to place axes according to their pairwise similarities [ZNGN16].

Since different dimensions may have different ranges, a data *normalization* step is usually applied in a pre-processing step. Normalization can be performed in various ways, e.g., by subtracting the mean and dividing by standard deviation. For our work, any normalization scheme can be used (but does not have to). For the presented examples, we simply scale all dimensions linearly to $[-1, 1]$.

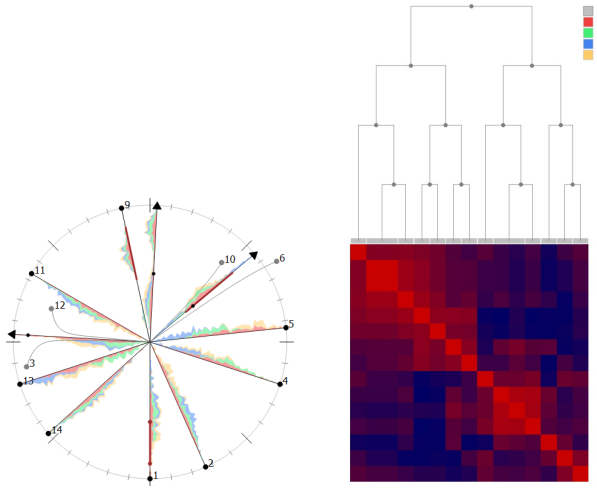
3.2. Bundling

The main purpose of bundling of graph edges or segments of polylines in parallel coordinates plot is to reduce visual complexity and to free the screen space making the visualization better readable for the user. In the examples of graphs and parallel coordinates plots, original unbundled elements are line segments connecting pairs of nodes determined by the given data. Bundling usually means a nonlinear distortion of such segments towards some curves. A curve of attraction commonly represents a mean behavior of a group (cluster) of original elements, which are being distorted. Generally, the distortion does not change the position of the end points of the segments. Thus, bundling is commonly a non-linear, end-point interpolating geometrical transformation of line segments towards a cluster-based target shape.

Technical implementations of the distortion transformation of the line segments may vary significantly. Generally, one has to balance the attraction strength that positively affects the occlusion reduction and the smoothness of the distorted trajectory which is responsible for the visual appeal and readability of the result. Ideally, such a balancing is parametrized. Since an optimal parameter setting may be data dependent, the search of the best configuration in interactive applications can or shall be left to the user.

One approach is to represent each deformed segment as a parametric curve of certain order, e.g., as a quadratic B-spline [MM08] or a C^1-continuous Bézier curve [HLKW12]. The bundling strength can then be controlled by adjusting the control points. Since the order of the resulting curve is fixed, the geometrical form of the bundled trajectories cannot be made arbitrary. Therefore, this method works well only if the attraction curve has a simple shape.

An alternative approach for deforming a line towards a curve is to discretize the line segment and evolve the discrete nodes. The new positions of the discrete nodes can be recomputed as a linear combination of their initial positions and the closest points on the skeleton curve [EHP*11]. The tightness of bundling defines the relative advection distance, i.e., is governed by the linear combination coefficients. Alternatively, one may interactively advect the discrete nodes along the gradient field of the distance functions induced by the skeleton curve using any integration scheme, e.g., the Euler scheme [HET12]. A smoothing step is required for removing small-scale advection artifacts caused by, e.g., discretization errors.



(a) *bundling and brushing* **(b)** *similarity matrix*

Figure 1: *(a) Bundling of SC axes on a synthetic dataset with* 15 *dimensions, where end points of conventional SC axes are shown as black dots and bundles are depicted as arrows. Bundling strength is visually encoded as small black dots on the bundling axis: Axes* 6 *and* 10 *are bundled with a lower bundling strength than axes* 3 *and* 12*. Axes* 7 *and* 8 *are bundled, but not shown (default) to reduce visual complexity. The axes are enhanced with stacked 1D histograms, where colors represent labels and histograms for bundles are aggregated over bundled axes. Brushing selections are shown in red. (b) Visualization of similarity matrix computed by Pearson correlation magnitude using a blue-to-red heatmap. The dendrogram above represents a hierarchical clustering of the dimensions. The dendrogram is used to define bundles by clicking at interior nodes, which are depicted by the grey horizontal bars at the leaves of the dendrogram. The color legend at the top right corner represents classes of labeled samples.*

4. Enhanced Scalable SC Interactions

4.1. SC Axes Bundles

Given a set of SC axes selected for grouping into a *bundle*, then the bundle is represented by a *bundle axis* and a set of *bundled axes*. The bundle axis can be operated just like an SC axis, i.e., it is characterized by its direction and length (or by the position of its end point, respectively). A bundle axis is visually represented by an arrow, see Figure 1a, while original SC axes are depicted with dots at their end points. On its creation, the bundle axis direction is chosen to be close to the first SC axis that was assigned to the bundle, where a deviation of five to seven degrees is introduced to avoid overplotting. The bundled axes are visually represented by curves that are obtained by a deformation of the axes, where the axes' endpoints are maintained. Next, we describe how we compute the deformation.

Each bundle $\mathbf{b}_k$ has a scalar-valued parameter α_k ranging from 0 to 1, which determines the *bundling strength*, where $\alpha_k = 0$ corresponds to the unbundled state and $\alpha_k = 1$ to the strongest bundling. Changing the value of α_k can be performed interactively by using the mouse wheel. When scrolling the mouse wheel, a black dot that moves along the bundle axis visualizes the currently chosen bundling strength, where smaller distances to the origin reflect a smaller bundling strength, see Figure 1a. By smoothly varying the bundling strength, we generate a continuous transition between classical SC and our bundled SC, where the bundling not only affects the rendering of the bundled axes but also adjusts the respective projection accordingly.

Since the attraction element $\mathbf{b}_k$ is just a linear segment, discretizing the original SC axes and advecting the resulting nodes towards $\mathbf{b}_k$, as it was described in Section 3.2, would have been an overly complicated procedure. Instead, we derive a simple closed-form analytical formula for the *deformation* of SC axes when bundled. Using B-splines is technically possible, but one would need to generate a proper curve parametrization, which is crucial for the resulting projection. Thus, we opted for a closed-form analytical solution.

Let us first assume that the end point of the bundle axis has coordinates $\mathbf{b}_k = (1, 0)$ and that $\mathbf{z}_i = (1, 1)$ is an original SC axis that is assigned to the bundle. Then, we define a transformation function

$$f(x) = (1 - \alpha_k)\, x + \alpha_k \, \frac{\left(e^{\beta_k (x^2 - 1)} - e^{-\beta_k}\right)}{1 - e^{-\beta_k}},$$

where $x \in [0, 1]$, $\alpha_k \in [0, 1]$ is the bundling strength, $\beta_k = (1 - \alpha_k + \varepsilon)^{-1}$, and ε is a small constant, which we set to 0.0005 in our experiments. The graph of $f(x)$ determines the shape of the bundled axis $\mathbf{z}_i$ with bundling strength α_k. For any parameter value x, it returns a two-dimensional point $(x, f(x))$. As the bundling strength increases, the SC axis' shape smoothly changes from the perfect line ($\alpha_k = 0$) to a curve in a small vicinity of the bundling axis. Thus, the user can interactively control the transition between the unbundled and bundled states of the axes.

In a general case, i.e., when dropping the assumption made above, the proposed bundling can be computed by applying a transformation T_{ki} of the coordinate system, which includes a rotation and a scaling. Let θ be the angle between bundle axis $\mathbf{b}_k$ and the

positive direction of the x-axis and $\mathbf{z}_i = (x_i, y_i)$. Then, an explicit formula for the transformation T_{ki} is

$$T_{ki} = \text{diag}\begin{pmatrix} (x_i\cos\theta - y_i\sin\theta)^{-1} \\ (x_i\sin\theta + y_i\cos\theta)^{-1} \end{pmatrix} \cdot \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix},$$

where diag(.) denotes a diagonal matrix with the given values as entries on the diagonal. Then, $T_{ki} \cdot \mathbf{z}_i = (1,1)$ and $T_{ki} \cdot \mathbf{b}_k \parallel (1,0)$. Putting it all together, the general formula for drawing the curve for axis $\mathbf{z}_i$ after bundling is given by $\mathbf{t}_{ki}(x) = T_{ki}^{-1} \cdot (x, f(x))^T, x \in [0,1]$.

As our motivation for introducing bundling was to reduce visual clutter, the bundled axes are only shown on demand, while the bundle axis is always depicted, see Figure 1a.

4.2. Non-linear Projection of Bundled SC

When performing a bundling of selected axes to a bundle $\mathbf{b}_k$, all bundled axes are deformed in a non-linear fashion as described above. For a consistent view on the data samples, their distribution needs to be deformed accordingly, leading to a non-linear projection. Overall, when assuming multiple bundles $\mathbf{b}_k$, $k = 1,\ldots,p$, the data samples $\mathbf{a}_j = (\mathbf{a}_{j1},\ldots,\mathbf{a}_{jn})$, $j = 1,\ldots,m$, are projected to the position $\mathbf{p}_j = \sum_{i=1}^{n} (\mathbf{t}_{1i} \circ \cdots \circ \mathbf{t}_{pi})(\mathbf{a}_{ji})$, where $\circ$ denotes the composition of functions. Of course, each bundle $\mathbf{b}_k$ only affects those dimensions, whose axes belong to the bundle. Thus, if axis $\mathbf{z}_i$ belongs to bundle $\mathbf{b}_k$, then all functions $\mathbf{t}_{li}$ will be the identity except for $l = k$.

The bundle axis $\mathbf{b}_k$ can be interacted with just like any of the original SC axes. Hence, when shortening or lengthening the bundle axis $\mathbf{b}_k$, the impact of the bundle decreases or increases accordingly. Similarly, changing the direction of the bundle axis $\mathbf{b}_k$ affects the projection. Whenever a bundle axis is interacted with, the non-linear projection is recomputed using the formula above. Effectively, changing the bundle axis is like changing all original SC axes belonging to the bundle.

4.3. Bundle Generation

Creation of a new bundle and assigning original SC axes to this bundle can be done manually, which gives full control to the user. However, when the number of dimensions is high, further aid is required. We support the generation of bundles by visualizing the correlation matrix of the given dimensions as well as the dendrogram of a hierarchical clustering approach. Interacting with these visualization allows for a top-down analysis strategy.

Using a *dendrogram* of a *hierarchical clustering* outcome allows the user to choose how many bundles the user wants to use for the SC. A top-down analysis would start with few bundles that can be refined upon demand. The dendrogram is created by hierarchical clustering on the dimensions with respect to some linkage scheme. For the experiments presented in this paper, we used single linkage. Hierarchical clustering has the advantage over other clustering approaches that no assumptions are made about cluster shapes, density, etc. and that no parameters need to be tuned. The only choice that needs to be made is where to cut the dendrogram for cluster generation, but this decision is what we deliberately give to the user to steer the process. The user just selects the respective cutting level in the dendrogram. Alternatively, the user can also define a similarity threshold using a slider, which leads to bundling all axes with similarities higher than the threshold.

We still need to decide how we compute pairwise similarities of dimensions. For comparing dimensions, it is common to compute their correlation, where the Pearson correlation is the most widely used scheme, which we also used for our implementation. However, the rationale behind the bundling is that axes with high correlations shall be bundled, no matter whether their correlation was positive or negative, i.e., we use the magnitude of the Pearson correlation as our similarity measure. Having negatively correlated axes in one SC bundle would simultaneously attract and repel points along the bundle axis. Therefore, we bundle highly negatively correlated SC axes, but give them opposite orientations within the bundle, which applies to both the projection computations and the bundled axis visualization.

To make an educated decision about which bundles to refine during a top-down approach, we support the user with detailed information about the cluster's homogeneity. We visually encode the pairwise Pearson correlation magnitude of all dimensions by drawing a heat map of the *correlation matrix* of all dimensions. Figure 1b shows a heatmap for a 15-dimensional dataset using a blue-to-red color map. The hierarchical clustering outcome determines the order of the dimensions. We place the dendrogram above the correlation matrix. The interface between the two visualizations is formed by horizontal bars that indicate which dimensions are currently grouped into bundles. Bundles can be created, modified, and deleted by interacting with the interior nodes and horizontal bars. The height of the nodes in the dendrogram merely encodes the clustering level and no correlation values, which makes the dendogram easier to read. The respective correlation values are displayed when hovering over the nodes with the mouse, cf. accompanying video. The bundling decisions based on hierarchical clustering and correlation visualization can be complemented with the user's expertise about the meaning of individual dimensions. The bundles, therefore, are completely customizable.

The correlation analysis can also be restricted to a subset of the given data samples, e.g., to a selected cluster of data samples using brushing (see below) or to a class in case of labeled data. Classes or data sample clusters are shown in the form of colored icons in the top right corner of the dendrogram view, where the grey icon corresponds to the entire dataset, see Figure 1b. The icons can be clicked to select the respective class or cluster and trigger a recomputation of the correlation matrix for the selection.

4.4. Smart Brushing in SC

Interactive selection and highlighting of subsets of the given data samples is an important tool in multidimensional data analysis for visually accentuating interesting patterns, data filtering, and performing manual classification. A common method for data selection is brushing, which in the SC projection domain is commonly performed using a lasso tool to select a region. However, if the projection layout is occluded or the target group of the projected data overlaps with other data, selection in the projection space is prac-

tically not possible, even if the group was separable in the multidimensional space.

We propose to complement the brushing interaction in SC by supporting brushing on the SC axes and bundles similar to brushing in parallel coordinates. However, while parallel coordinates to some extent allow for a comprehension of the 1D data distribution of each dimension, a meaningful exploratory brushing on SC axes is basically impossible. To compensate for this shortcoming, we support a *"smart" brushing* on SC axes by depicting 1D *histograms* of data distributions along each axis, see Figure 1a. Histograms are generated by summing values over equally sized bins. If desired, the number of bins can be interactively adjusted, but this was not necessary for any of the datasets we used. As all dimensions are normalized to the range $[-1, 1]$, the histograms can directly be mapped to the axes by adding a unity and scaling by the axes' half-lengths. Given the 1D histograms, clusters that separate along one of the dimensions can be easily spotted and selected. *Multidimensional clusters* (clusters that do not separate in a single dimension) can also be found by brushing on not perfectly separated peaks of a histogram in one dimension and refining the selection by brushing on further histograms in other dimensions, see Section 5. Also, axes brushing can be used in addition to region brushing.

When dealing with labeled data or when already having defined clusters interactively, the data distributions of all classes or clusters are shown by stacked histograms, where each class or cluster is given a different color, see Section 5. The stacked histograms also ease other SC interactions such as adjusting the projection layout. For example, when trying to separate classes, the histograms provide *hints* about which axes to interact with.

Brushing interaction on the axes is supported as follows. Hovering over an axis shows the value at the respective position. Right-clicking and dragging along the axis then shows the values of the respective interval. Brushing outwards (from origin to end point) adds the brushed interval to the selection. Brushing inwards (from end point to origin) removes the brushed interval from the selection. Multiple selections on one axis can be combined. Then, the brushing selects all data samples, whose value lies in the union of selected intervals for the brushed axis. Brushing on multiple axes selects all data samples, whose values lie in the union of selected intervals for all brushed axes. Brushing can be restricted to selected groups such as labeled classes or previously defined clusters. Brushing is always applied to the currently selected group.

When axes are *bundled*, the histograms of the SC axes that belong to the bundle are aggregated to be represented by a single histogram. Hence, bundles are handled like original SC axes. The brushing interaction on bundles selects all data samples, whose values lie in the union of selected intervals for all SC axes that belong to the bundle. If the bundle contains negatively correlated dimensions, the selection of an interval $[a,b]$ on the bundle axis is mapped to a selection of interval $[1-b, 1-a]$ for each negatively correlated dimension. Assuming that bundles represent highly correlated dimensions, bundling the dimensions before brushing can significantly reduce the workload. On the other hand, the histograms are also helpful for deciding on which axes to bundle, especially for labeled data. While the hierarchical clustering considers all data samples regardless of their label, the stacked histograms may provide further information about how labels are distributed.

5. Results

To test the effectiveness of our approach, we performed numerical tests and a user study using both synthetic datasets and datasets from the UCI machine learning repository [DG19]. including the *Wine*, *Breast Cancer Wisconsin (Diagnostic)* (WDBC), and *Mice Protein Expression* datasets. The datasets contain between 13 and 77 dimensions with 178 to 569 data points.

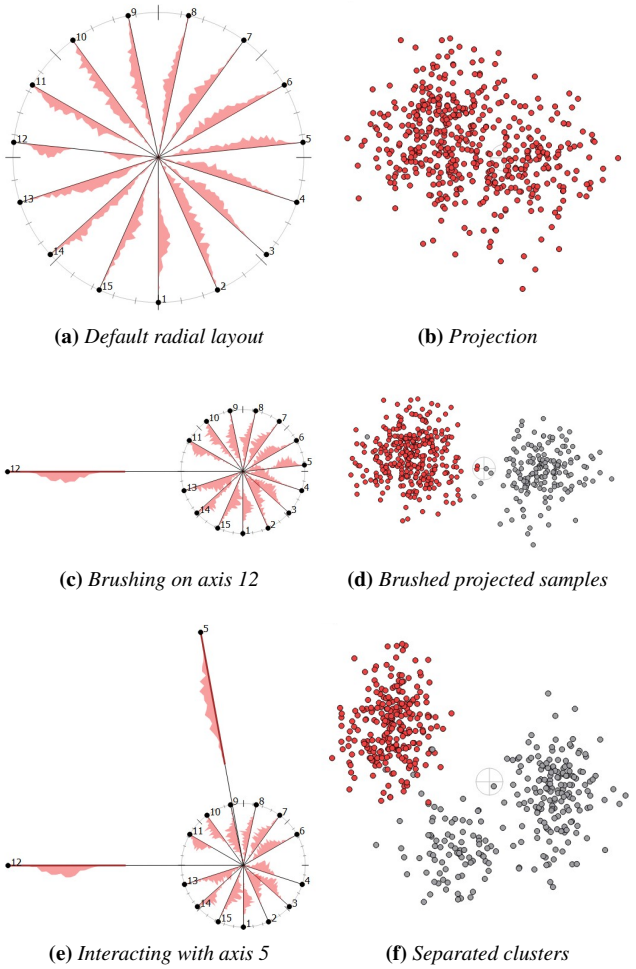
5.1. Numerical Experiments

In this section, we demonstrate the use of our proposed SC brushing and bundling methodologies to obtain analysis results for a few illustrative examples. Interactive sessions with our tool are shown in the accompanying video.

Brushing for Cluster Detection. We first demonstrate the use of smart brushing for cluster detection. We load a synthetic dataset with 15 attributes and 500 samples. The dataset is unlabeled, but contains three multidimensional Gaussian clusters. Figures 2a and 2b show the SC widget in its default state (radial layout) and the respective projection. The SC axes are enhanced with the 1D histograms. The histograms show a bimodal distribution on axis 12. Thus, we brush on the axis to select the data samples with higher values in that dimension, see Figure 2c. By elongating the axis, we give it more weight, which separates the brushed selection (red) from the unbrushed points (grey) in the projection in Figure 2d. When making a selection via brushing, the histograms get re-computed for the selected points only, see Figures 2c. We observe that for the selection, the histogram of axis 5 is bimodal too. Hence, we refine the point selection by brushing on axis 5 (selecting high values) and change the SC layout by elongating and rotating axis 5, see Figure 2e. We observe that the prior selection splits again into two clusters such that we overall have found a projection layout that separates all three clusters that are contained in the dataset, see Figure 2f. In summary, our smart brushing with histograms allowed us to detect the three clusters by only touching two of the 15 axes and to assign 496 out of 500 samples, i.e., 99.2% of data points, to correct clusters. Without the histograms and the brushing, one would have, in general, had to touch all 15 axes to observe their impact on cluster formations.

Scalability. Next, we demonstrate the effectiveness of bundling for reducing high dimensionalities. We analyze the *Mice Protein Expression* dataset, containing 77 dimensions and 552 data samples (after removing those with missing values). The protein expressions of several healthy and trisomic mice were measured. Some of them have been stimulated to learn (referred to as context-shock), while others have not (referred to as shock-context), and for some trisomic mice a drug was injected. In our analysis, we focus on the distinction between context-shock and shock-context mice. The analysis goal was to separate these two classes in the projection space.

Figures 3a and 3b show the initial SC configuration, the projection layout, and the dendrogram after loading the dataset and performing hierarchical dimension clustering. We observe that the SC



(a) *Default radial layout* **(b)** *Projection*

(c) *Brushing on axis 12* **(d)** *Brushed projected samples*

(e) *Interacting with axis 5* **(f)** *Separated clusters*

Figure 2: *Axes brushing for cluster detection on unlabeled synthetic dataset with* 15 *dimensions,* 500 *samples, and 3 clusters. (a) and (b) show default SC layout and resulting projection. Axis 12 with bi-modal distribution is brushed on and elongated in (c). Selected cluster (red) is shown in the projection in (d). Brushing and interacting with axis* 5 *in (e) refines the prior cluster, leading to a projection where the 3 clusters are visible in (f).*

configuration has a high visual complexity. Obviously, it would be a daunting task to interact with all 77 axes to find a good configuration. Instead, we performed an automatic bundling by applying (absolute) correlation threshold 0.4 to the dimension cluster hierarchy. All SC axes were bundled, which reduced the number of axes to a feasible amount. Here, the number of axes was reduced from 77 to 18, see Figures 3c. We interact with the bundle axes, which results in a much better separation of the two classes, see Figure 3d. The final silhouette coefficient is 0.697.

Even though the correlation threshold used was quite low (0.4), we were able to quickly separate the two classes. Had there been complications, dissolving any of the bundles would have been an option. In general, a structured analysis can be performed by starting with a low correlation threshold, which results in large bundles that can then interactively and iteratively be refined during the analysis process. This top-down approach using bundling (in conjunction with smart brushing on the bundle axes) makes the analysis of datasets with many dimensions in SC much more feasible.

5.2. User Study

We conducted a quantitative user study to evaluate our approach against classical SC.

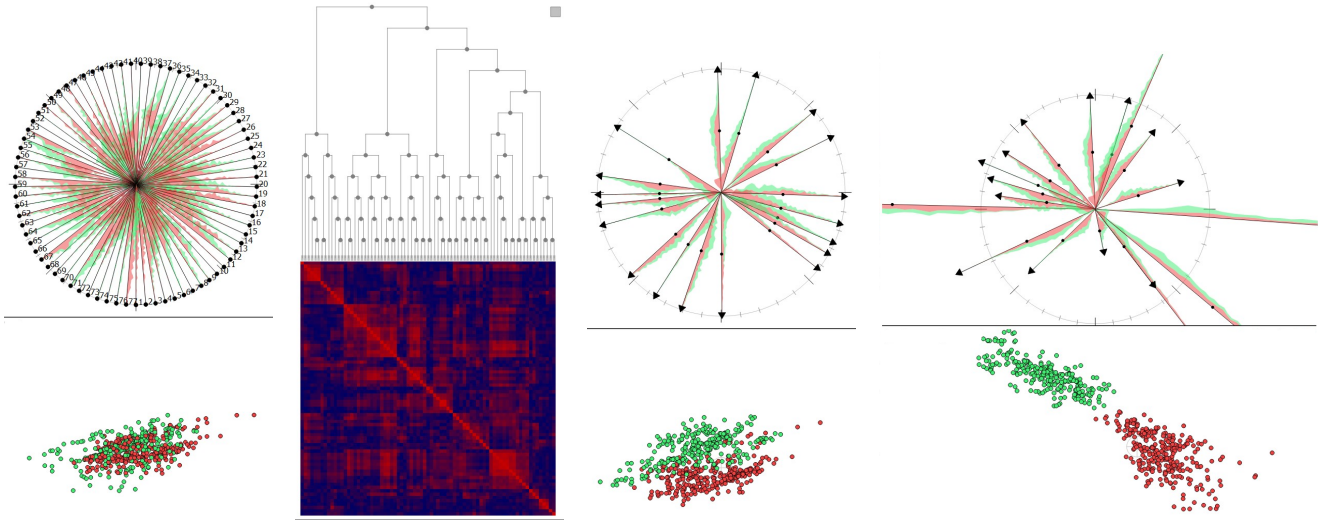
Set-up. A total of 20 participants (age 20 to 64 with an average of 36.25, 15 male) were recruited to perform three different tasks with classical SC and with our approach. To compare the scalability of the two approaches, each task was performed on a dataset with a lower (about 15) and a dataset with a higher (about 30) dimensionality. For each task, we measured the participants' performance, total time to complete the task, number of axis pickups, and confidence. We followed an intra-subject design, i.e., the participants were split into two equally sized groups, one starting with classical SC and the other with our approach. The interaction mechanisms of both approaches were explained in detail. Each session took approximately one hour.

Datasets. Ten of the twelve datasets used for the user study were synthetic datasets. The other two datasets were the Wine and the WDBC datasets. They were used for the *Class Separation* and *Labeling Missing Data* tasks, respectively. The synthetic datasets use isotropic Gaussian kernels that only differ by their means. All generated classes are separable. The description of all datasets are provided in Table 1 and can be found in the supplementary material. Their visualizations with the default radial SC layout (starting point for the subjects) are shown in the supplementary material as well.

Hypotheses. We tested the null hypothesis that our approach and classical SC perform equally well for each of the measures. Completion time, number of axis pickups, and user confidence were compared for every dataset while some measures are specific to a certain task.

Tasks. The subjects completed three analysis tasks as detailed below. The accompanying video shows examples of task completions.
Class Separation: Given a labeled dataset with three disjoint classes, the participants have to optimally separate them in the projection space. For this task, we compared silhouette coefficients.
Clustering: Given an unlabeled dataset containing three clusters, the participants have to find the clusters and select them in the projection space. For this task, we compared the number of correctly labeled samples.
Labeling Missing Data: Given a labeled dataset with three unlabeled samples, the participants have to assign missing labels. For this task, we compared the number of correct labels.

Statistical analysis. For the statistical analysis, we provide p-values, calculated using a two-sample unpooled t-test, and the effect sizes $S = (\mu_1 - \mu_2) \,/\, \sqrt{\frac{\sigma_1+\sigma_2}{n_1+n_2-2}}$, where μ_i is the mean, σ_i the standard deviation, and n_i the number of participants in the group. In our experiment, $n_1 = n_2 = 10$. We consider the effect size to be



(a) *Silhouette* 0.014 **(b)** *Initial* **(c)** *Silhouette* 0.205 **(d)** *Silhouette* 0.697

Figure 3: *Analysis of Mice Protein Expression dataset with 77 dimensions, 552 samples, and two classes. (a) Initial SC layout and projection with two classes highlighted in red and green. (b) Similarity matrix and dendrogram for 77 dimensions. (c) Result of automatic bundle selection. (d) Final separation of classes with silhouette coefficient* 0.697.

	Class Separation				Clustering				Labeling Missing Data			
Dataset	Wine	A15_2	A30_1	A30_2	B15_1	B15_2	B30_1	B30_2	C15_1	C15_2	WDBC	C30_2
Dimensions	13	15	30	30	15	15	30	30	15	15	30	30
Samples	178	200	300	300	200	200	300	300	200	200	569	300
Classes	3	3	3	3	3	3	3	3	4	4	2	4

Table 1: *Descriptions of datasets for user study.*

significant, if its absolute value is greater than 0.8, and the p-value to be significant, if it is smaller than significance level 0.1.

Results. All statistical information about our analysis is provided in the supplementary material. For many tests, the null hypothesis could not be rejected. Concerning *completion time*, only for dataset C15_1 our approach was significantly faster than classical SC with an effect size of -0.97. For datasets C15_1, WDBC, C30_2, and B15_2 our approach resulted in significantly fewer *axis pickups*, with effect sizes of -2.41, -1.02, -1.04, and -1.01, respectively. No datasets used for the *Class Separation* task showed significant differences. The average *confidence* was, despite the higher complexity of our approach and the loss of detail when bundling, smaller only for the B30_1 dataset. In terms of *accuracy*, the silhouette coefficient was significantly higher for our approach for all four datasets in the *Class Separation* task with absolute effect sizes 1.82, 0.86, 1.43, and 1.02 and p-values 0.001, 0.082, 0.005, and 0.037. We therefore reject the null hypothesis in favor of our approach. No significant difference was found neither for correctly clustered samples nor for correctly sorted samples in tasks *Clustering* and *Labeling Missing Data*. There was no significant differences in our findings when comparing the results for 15- and 30-dimensional datasets.

Discussion and Conclusions. We conclude that the accuracy significantly improved when using our approach in comparison to standard SC for the *Cluster Separation* task. For the other two tasks, subjects achieved almost perfect results for both methods, i.e., there was no significant difference. We planned to repeat the experiments with more challenging datasets but due to social contact restrictions after the outbreak of the pandemic we had to post-pone further user studies. For several datasets, we observed that less axis interactions were necessary to complete the tasks when using our approach. It is also positive that the subjects did not feel less confident in their findings when using our tool despite the aggregation via bundling.

After completing a session, the subjects were given the chance to provide some general feedback regarding both tools. 15 subjects stated that the histograms helped them identify important axes, thus enabling a more methodological approach. 8 participants said that bundling helped them in identifying important axes, while 2 participants said it helped them to reduce the number of axes they had to handle. On the other hand, 2 participants found the correlation matrix and dendrogram to be unnecessary and suggested to only provide automatic bundling, and 2 participants did not find the histograms helpful at all.

6. Conclusions

We proposed enhanced interactions with SC axes. Axes bundling allowed us to reduce the dimensionality of the data, thus, enabling SC interactions for datasets with larger number of dimensions, where correlation investigations facilitated bundle selection. Enhancing axes with visualizations of stacked histograms allowed us to perform educated axes brushing operations for more efficient and effective cluster selection and class separation.

Acknowledgmnents

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) grant 360330772 (MO 3050/2-1).

References

[DG19] DUA D., GRAFF C.: UCI machine learning repository, 2019. URL: http://archive.ics.uci.edu/ml. 5

[ED07] ELLIS G., DIX A.: A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (Nov 2007), 1216–1223. doi:10.1109/TVCG.2007.70535. 2

[EHP*11] ERSOY O., HURTER C., PAULOVICH F., CANTAREIRO G., TELEA A.: Skeleton-based edge bundling for graph visualization. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2364–2373. doi:http://doi.ieeecomputersociety.org/10.1109/TVCG.2011.233. 2, 3

[HEF*14] HURTER C., ERSOY O., FABRIKANT S. I., KLEIN T. R., TELEA A. C.: Bundled visualization of dynamic graph and trail data. *IEEE Transactions on Visualization and Computer Graphics 20*, 8 (2014), 1141–1157. doi:10.1109/TVCG.2013.246. 2

[HET12] HURTER C., ERSOY O., TELEA A.: Graph bundling by kernel density estimation. *Computer Graphics Forum 31*, 3 (June 2012), 865–874. doi:10.1111/j.1467-8659.2012.03079.x. 2, 3

[HGM*97] HOFFMAN P., GRINSTEIN G., MARX K., GROSSE I., STANLEY E.: DNA visual and analytic data mining. In *Proceedings of the 8th Conference on Visualization* (Los Alamitos, CA, USA, 1997), VIS '97, IEEE Computer Society Press, pp. 437–442. doi:10.1109/VISUAL.1997.663916. 2

[HLKW12] HEINRICH. J., LUO. Y., KIRKPATRICK. A. E., WEISKOPF. D.: Evaluation of a bundling technique for parallel coordinates. In *Proceedings of the International Conference on Computer Graphics Theory and Applications and International Conference on Information Visualization Theory and Applications - Volume 1: IVAPP, (VISIGRAPP 2012)* (2012), INSTICC, SciTePress, pp. 594–602. doi:10.5220/0003821205940602. 2, 3

[Hol06] HOLTEN D.: Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (Sep. 2006), 741–748. doi:10.1109/TVCG.2006.147. 2

[ID90] INSELBERG A., DIMSDALE B.: Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st Conference on Visualization '90* (Los Alamitos, CA, USA, 1990), VIS '90, IEEE Computer Society Press, pp. 361–378. 2

[Ins85] INSELBERG A.: The plane with parallel coordinates. *The Visual Computer 1*, 2 (1985), 69–91. doi:10.1007/BF01898350. 2

[Ins09] INSELBERG A.: *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer-Verlag, Berlin, Heidelberg, 2009. doi:10.1145/1764810.1764834. 2

[Kan00] KANDOGAN E.: Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In *Proceedings of IEEE Information Visualization Symposium* (2000), pp. 4–8. 2

[Kan01] KANDOGAN E.: Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2001), KDD '01, ACM, pp. 107–116. doi:10.1145/502512.502530. 2

[LHT17] LHUILLIER A., HURTER C., TELEA A.: State of the art in edge and trail bundling techniques. *Computer Graphics Forum 36*, 3 (2017), 619–645. doi:https://doi.org/10.1111/cgf.13213. 2

[LT16] LEHMANN D. J., THEISEL H.: General projective maps for multidimensional data projection. *Computer Graphics Forum 35* (2016), 443–453. doi:10.1111/cgf.12845. 2

[MCL15] MOLCHANOV V., CHITIBOI T., LINSEN L.: Visual analysis of medical image segmentation feature space for interactive supervised classification. In *Eurographics Workshop on Visual Computing for Biology and Medicine, VCBM* (September 2015), pp. 11–19. doi:10.2312/vcbm.20151204. 2

[MFL13a] MOLCHANOV V., FOFONOV A., LINSEN L.: Continuous representation of projected attribute spaces of multifields over any spatial sampling. *Computer Graphics Forum 33*, 3 (June 2013), 301–310. doi:10.1111/cgf.12117. 2

[MFL13b] MOLCHANOV V., FOFONOV A., LINSEN L.: Frequency-based progressive rendering of continuous scatterplots. *Journal of WSCG 21*, 1 (July 2013), 49–58. 2

[ML14] MOLCHANOV V., LINSEN L.: Interactive Design of Multidimensional Data Projection Layout. In *Proceedings of EuroVis 2014 – Short Papers* (June 2014), pp. 25–29. doi:10.2312/eurovisshort.20141152. 2

[MM08] MCDONNELL K. T., MUELLER K.: Illustrative parallel coordinates. *Computer Graphics Forum 27*, 3 (2008), 1031–1038. doi:10.1111/j.1467-8659.2008.01239.x. 2, 3

[PBO*14] PALMAS G., BACHYNSKYI M., OULASVIRTA A., SEIDEL H. P., WEINKAUF T.: An edge-bundling layout for interactive parallel coordinates. In *Proceedings of the 2014 IEEE Pacific Visualization Symposium* (USA, 2014), PACIFICVIS '14, IEEE Computer Society, pp. 57—-64. doi:10.1109/PacificVis.2014.40. 2

[REB*16] RAIDOU R. G., EISEMANN M., BREEUWER M., EISEMANN E., VILANOVA A.: Orientation-enhanced parallel coordinate plots. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2016), 589–598. doi:10.1109/TVCG.2015.2467872. 2

[RGE19] RAIDOU R. G., GRÖLLER M. E., EISEMANN M.: Relaxing dense scatter plots with pixel-based mappings. *IEEE Transactions on Visualization and Computer Graphics 25*, 6 (June 2019), 2205–2216. doi:10.1109/TVCG.2019.2903956. 2

[RLS*19] ROBERTS R. C., LARAMEE R. S., SMITH G. A., BROOKES P., D'CRUZE T.: Smart brushing for parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics 25*, 3 (March 2019), 1575–1590. doi:10.1109/tvcg.2018.2808969. 2

[SZS*17] SACHA D., ZHANG L., SEDLMAIR M., LEE J. A., PELTONEN J., WEISKOPF D., NORTH S. C., KEIM D. A.: Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (Jan 2017), 241–250. doi:10.1109/TVCG.2016.2598495. 2

[TE10] TELEA A., ERSOY O.: Image-based edge bundles: Simplified visualization of large graphs. *Computer Graphics Forum 29*, 3 (2010), 843–852. doi:10.1111/j.1467-8659.2009.01680.x. 2

[TM03] TEOH S. T., MA K.-L.: StarClass: Interactive visual classification using star coordinates. In *SDM* (2003), Barbará D., Kamath C., (Eds.), SIAM, pp. 178–185. doi:10.1137/1.9781611972733.16. 2

[ZNGN16] ZANABRIA G. G., NONATO L. G., GOMEZ-NIETO E.: iStar (i*): An interactive star coordinates approach for high-dimensional data exploration. *Computers & Graphics 60* (2016), 107–118. doi:10.1016/j.cag.2016.08.007. 2