# Skeleton-based Scagnostics

José Matute, Alexandru C. Telea, and Lars Linsen

**Abstract**—Scatterplot matrices (SPLOMs) are widely used for exploring multidimensional data. Scatterplot diagnostics (scagnostics) approaches measure characteristics of scatterplots to automatically find potentially interesting plots, thereby making SPLOMs more scalable with the dimension count. While statistical measures such as regression lines can capture orientation, and graph-theoretic scagnostics measures can capture shape, there is no scatterplot characterization measure that uses both descriptors. Based on well-known results in shape analysis, we propose a scagnostics approach that captures both scatterplot shape and orientation using skeletons (or medial axes). Our representation can handle complex spatial distributions, helps discovery of principal trends in a multiscale way, scales visually well with the number of samples, is robust to noise, and is automatic and fast to compute. We define skeleton-based similarity metrics for the visual exploration and analysis of SPLOMs. We perform a user study to measure the human perception of scatterplot similarity and compare the outcome to our results as well as to graph-based scagnostics and other visual quality metrics. Our skeleton-based metrics outperform previously defined measures both in terms of closeness to perceptually-based similarity and computation time efficiency.

## 1 Introduction

Scatterplot matrices (SPLOMs) are one of the oldest, and still frequently used, tools for exploring multidimensional data. Their attractiveness comes from the fact that they re-use the 2D scatterplot metaphor, which is very familiar and easy to understand for a wide spectrum of users, in a small-multiples setting. However, SPLOMs require visual space which is quadratic in the number of dimensions, and as such do not scale well visually to datasets having more than roughly ten dimensions. Scatterplot diagnostics, or *scagnostics*, attack this problem by essentially selecting a subset of 'interesting' scatterplots from the $d^2$ existing ones for a $d$-dimensional dataset, and presenting this subset to the user. The effectiveness of a scagnostic approach is, thus, directly linked to its ability to quantify the degree of interestingness that a scatterplot presents.

Many measures have been proposed to quantify the above-mentioned interestingness. Roughly, these can be divided into measures that capture the *correlation* of the variables in a scatterplot [54, 64], and measures that capture the data pattern's *shape* [59, 63]. However, to our knowledge, no measure combines both aspects in a way that is generic for a wide class of patterns that can occur in scatterplots.

In this paper, we propose such a novel measure. We base our work on skeleton-based descriptors, which have been used for a long time in shape analysis to reason about, and analyze, 2D shapes [48]. State-of-the-art variants of such descriptors (i) accurately capture shape geometry, orientation, and topology, (ii) provide the captured information in a multiscale way, making them thus noise-resistant and allowing one to filter details under a desired scale, and (iii) can be computed automatically and in real-time for large input images. We adapt skeleton-based descriptors to handle scatterplots representing complex spatial point distributions and help discovering principal trends present in the scatterplot, while keeping the attractive scalability and ease-of-use of the underlying techniques. Our descriptors can be easily added to any SPLOM visualization, as they only require access to the underlying 2D scatterplot images.

We present a scagnostics approach that is based on skeleton computations from scatterplots. Our individual contributions can be summarized as: (C1) *Skeletons as descriptors*: We use skeletons as shape descriptors of scatterplots and define distance measures between scatterplots based on skeleton similarity. The skeleton computation is based on a multiscale approach that is automatic, efficient, and robust. Our distance measure captures shape and orientation of the scatterplots, see Section 3. (C2) *Comparison to perception-based distances*: We perform a user study to measure perceptual distances between scatterplots. The outcome is compared against our skeleton-based distances and other state-of-the-art methods. We show that our approach is closest to the perception-based distances, in addition to being the computationally most efficient, see Section 4. (C3) *Skeletons as summarization*: We use skeletons as a visual representation for encoding the main features in scatterplots. This sparse visual representation reduces the rendering complexity and is, therefore, amenable for small-multiples approaches such as SPLOMs. We incorporate the skeleton-based summarization in an interactive visual SPLOM analysis, where filtering, selection, and re-ordering is based on the skeleton-based scagnostics, see Section 5.

## 2 Related Work

Related work can be categorized into approaches for characterizing scatterplots and those for summarizing scatterplots.

### 2.1 Scatterplot characterization

Let $D = \{\mathbf{p}_i\} \subset \mathbb{R}^d$, $1 \leq i \leq N$ be a $d$-dimensional dataset with $N$ points $\mathbf{p}_i$. Let $S = \{\mathbf{x}_i\} \subset \mathbb{R}^2$ be a scatterplot generated from $D$, i.e., where $\mathbf{x}_i$ represents the data point $\mathbf{p}_i$. Such scatterplots can be created in many ways, e.g., by selecting two dimensions from $\{1 \dots d\}$, or by dimensionality reduction, also called Multidimensional Projection (MP) [50]. For our work, both types of scatterplots are in scope. To characterize $S$, one can study its actual 2D point distribution. Alternatively, one can transform $S$ to a so-called feature space, i.e., extract higher-level measures from $S$ that allow an easier, more global, and more insightful reasoning about $S$.

Tukey and Tukey [59] proposed an initial characterization for 2D scatterplots called *scagnostics* for 'scatterplot diagnostics'. The proposed features included geometric graph analysis, computing principal curves, and kernel-based measures. An underlying probability function was assumed when calculating these features.

Wilkinson et al. [63] refined the above characterization measures for scatterplots. Their aim was to generate a small number of features that could characterize the many types of possible point distributions in $S$ on a common scale. Attention was dedicated to scalability, as several features proposed earlier [59] were prohibitively expensive for large $N$ (sample count). They proposed nine feature classes based on geometric graphs defined by the scatterplot: outlying, skew, clumpy, sparse, striate, convex, skinny, stringy, and monotonic. These measures are based on the shape of $S$ and do not take into account its orientation

- *José Matute and Lars Linsen are with the Institute of Computer Science at the University of Münster, Germany. E-mail: {matutefl | linsen} @uni-muenster.de.*
- *Alexandru C. Telea is with the Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, PO Box 407, 9700 AK Groningen, The Netherlands. E-mail: a.c.telea@rug.nl.*

*Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx*

in the embedding 2D space. The monotonicity feature is the closest to describe orientation as it estimates whether the relationship in $S$ follows an entirely non-increasing or non-decreasing trend. However, different scatterplots with different orientations with respect to the embedding 2D axes, and with different amount of noise, may produce the same monotonicity value.

More recently, additional features have been developed to characterize scatterplots. Sips et al. [49] proposed two quantitative measures – distance consistency and distribution consistency. These measures can be used to compute the class consistency, i.e., the visual separation between projected classes (labeled points) in a scatterplot. Distance consistency is defined as the distance to the classes' barycenters; distribution consistency is based on the spatial distribution of $\mathbf{x}_i$. Related to the above, several metrics have been defined to characterize the *quality* of scatterplots that represent MPs of high-dimensional data. Simple global metrics include normalized stress and neighborhood preservation plots [37], distance preservation plots [25], cluster segregation measures [44,45], and projection precision scores [42]. More refined metrics can show how MPs preserve distances [3,20,29,31] or neighborhoods [32] locally over $S$. However, such metrics are less interesting in our context, as they are specific to scatterplots obtained by MP only.

Guo [17] proposed a maximum conditional entropy (MCE) that computes the maximum between rows and columns in a 2D nested-means discretized grid. This measure was aimed for the detection of point clusters in 2D scatterplots representing MPs. Tatu et al. [54] proposed several features for characterizing scatterplots of both classified and unclassified data. Similar to Sips et al., a class distance measure and a histogram-based density measure are proposed for classified data. For unclassified data, a Rotating Variance Measure (RVM) was proposed to find (non)linear correlations between dimensions. Shao et al. [46] proposed grouping scatterplots according to *motifs*, i.e., locally similar scatterplot segments. A motif dictionary is generated by clustering local scatterplot segments through $k$-means and an interest measure is defined based to the motif's uniqueness in the dataset. They remarked that the selection of an appropiate dictionary size is crucial for subsequent analysis steps.

Yates et al. [64] proposed a categorization based on Boolean logical implication. The method categorizes scatterplots into eight different classes. It defines four quadrants in the scatterplot based on the distribution of the points and on a discretization step, and assigns the scatterplot to a class defined by which quadrants are filled. Such approaches have limited characterization power as they aim to detect a single condition – detection and separability of classes or, alternatively, detection of correlations. In the specific case of Yates et al., characterization power is further limited as that approach produces categorical, rather than continuous, values.

A different direction to characterize scatterplots relates to studying how humans perceive them. In this direction, much of the related work is task-based, i.e., focuses on seeing how one can infer insight from a scatterplot that help solving a given problem or answering a given question. Within this area, a large number of papers focus on tasks related to classification (labeled) data, e.g., the identification of class clusters and their visual separation [13,14,44]. In the same direction, Tatu et al. [55] studied the perceived quality of 2D projections based on cluster separability and cluster density. Rensink et al. [40] studied how correlation is perceived in similar scatterplots.

Yet another direction is the *comparison* of scatterplots. Globally put, if one can automatically measure how similar two scatterplots are, one can next classify (characterize) such scatterplots based on their similarity to given templates. Similarly, if one can measure how similar two scatterplots are perceived to be (by an observer), then one can infer how users interpret a SPLOM 'served' to them by scagnostics methods. Looking at plot similarity, Pandey et al. [35] derived key perceptual features where a set of scatterplots were grouped based on their perceptual judgment of similarity. Density, spread, and orientation were found to be the three most dominant features that determine the perceived scatterplot similarity. The need for developing additional perceptually-balanced measures was also outlined in [35]. Separately, Dang and Wilkinson [9] defined a scatterplot similarity measure based on Euclidean distance in feature space.

Scherer et al. [41] defined a set of representative functional models where the relative fitness of a scatterplot to each of these models is computed. A regressional feature vector (a probability density function for the functional model set) and a regressional coefficient feature vector (the calculated coefficients for each functional model) are then used for (dis)similarity computation. The dissimilarity is defined as a weighted distance sum between the two previously defined scatterplots feature vectors. A weighting factor is used to assign importance to the overall functional form or the functional coefficients. Similar to the approach by Shao et al. [46], the discriminative power is based on the algorithm's initial configuration, in this case of the chosen functional models.

Closest to our evaluation part, Albuquerque et al. [1] developed a perception-based visual quality measure used to compare scatterplots. For this, a perceptual user-study was performed on a training set of scatterplots, yielding a perceptual similarity ranking. To compare new scatterplots, a principal component analysis (PCA) projection of the training set and the new scatterplot was computed, and the nearest $k$ elements in the PCA projection were used to interpolate the new scatterplot value. A key drawback of this approach to compare scatterplots is the need to train the similarity metric on a well-constructed, comprehensive, training set that should capture well the space of all possible scatterplots.

Our approach follows the idea of scagnostics, but we propose a novel skeleton-based descriptor and a respective similarity measure that capture well shape and orientation of scatterplots and can be computed efficiently. We compare our approach against the work by Wilkinson et al. [63] and Tatu et al. [54], which following the discussion above we consider the state of the art for scatterplot characterization in scagnostics, see Section 4.

## 2.2 Scatterplot summarization

A related direction to scatterplot characterization is *summarization*, or the way in which $S$ is to be drawn. Indeed, knowing more about the characteristics of $S$ helps one emphasizing these in the resulting visualization. This also helps visual scalability, i.e., cases where one cannot draw the entire SPLOM at full level of detail due to the many dimensions $d$ present. In such cases, we want to show a summary that reflects a scatterplot's essential characteristics.

Besides the basic drawing of the cloud of points $\mathbf{x}_i$, extra information can be added at the level of points, scatterplots, or an entire SPLOM. At the point level, one can add more dimensions encoded in color, transparency, shading [8], or glyph shape. This works well for relatively low sample counts $N$ or when one subsamples $S$ to create more visual space to encode data for groups of related points, e.g. via tag clouds [38]. Given a large enough sample count $N$, overplotting eventually occurs in $S$ and single sample properties will not be distinguishable. At the scatterplot level, this can be mitigated by spatially aggregating values via density maps [31,42]. At an even higher aggregation level, a functional boxplot can be computed by binning over data ranges and calculating summary statistics over bins [51]. For large SPLOMs, a focus-plus-context approach is used, as single scatterplots become too small to understand. For example, Yates et al. [64] proposed glyphs based on the class categorization.

Summarization and characterization are linked by *principal curves* [18]. The key idea is to summarize a 2D scatterplot by a 1D curve that passes 'through the middle of the data' locally, thereby generalizing the concept of linear regression or the concept of PCA to using curves, and improving upon parametric nonlinear regressors by making the curve independent on the orientation of the scatterplot with respect to the 2D coordinate axes. Given a scatterplot $S$, principal curves $\lambda : \mathbb{R}^+ \to \mathbb{R}^2$, represented in non-parametric form, are computed by initializing a discrete polyline representation $\Lambda = \{\mathbf{l}_j\}$ to the largest principal component of $S$ and next iteratively adapting the polyline so as to minimize the sum of squared distances from $\mathbf{x}_i \in S$ to the closest $\mathbf{l}_j \in \Lambda$. While principal curves can summarize complex scatterplots better than PCA, their computation is not guaranteed to converge, strongly depends on proper initialization, and does not work for scatterplots whose overall shape cannot be approximated well by a single curve, i.e., whose data are described by multiple trends. Most of these limitations (but not the key last one) are removed by further refinements of principal curves [34]. Figures 1 a) and b) show the principal curves computed with the mentioned methods ( [18] and [34], respectively) for a 2D noisy spiral-like scatterplot. It can be observed that the method in a) [18] wrongly fits the curve to the scatterplot in areas where gaps are small; the method in b) [34] is better, but is still very expensive, cf. Section 3.2, .

Reddy et al. [39] extended principal curves into so-called data skeletons or Principal Trees, which summarize scatterplots where multiple trends occur. For this, the dataset $D$ is partitioned into $k$ clusters, and next a minimum spanning tree (MST) is computed from their centroids. The data are then filtered to contain only clusters traversed by the MST from one endpoint to another. For $m$ such endpoints, a total of $m(m-1)/2$ principal curves is computed. However, the method has several problems. If a too low $k$ value is used, the summarization will not capture the overall shape of the data. Also, the method needs to evaluate a high number of principal curves – $O(m^2 d^2)$ for a SPLOM with $d$ dimensions – which makes it slow for high-dimensional datasets.

The idea of a 'skeletal' representation of datasets was also explored by Gerber et al. [16]. Here, a simplified geometric representation of high-dimensional data is proposed based on regression curves and dimensionality reduction. The set of regression curves represent a topology-based skeleton of the data. While this approach cannot be directly applied to 2D scatterplots, it can be used to simplify density maps computed from scatterplots.

In the remainder of this paper, we extend the set of tools for scatterplot characterization with several descriptors based on shape skeletons. As we will show, these are simple and fast to compute, and capture dissimilarity between scatterplots, as perceived by humans, better than other existing descriptors listed above.

## 3 SKELETON-BASED SCAGNOSTICS

Towards establishing skeleton-based scatterplot characterizations and scagnostics, we first describe the concept of skeletons as shape descriptors (Section 3.1), then detail how we construct skeletons from scatterplots robustly, efficiently, and automatically (Section 3.2), and finally propose distance measures between scatterplots based on skeleton descriptors (Section 3.3).

### 3.1 Shape descriptors

Scatterplot characterization can be seen as a particular case for the more general field of shape analysis, which deals with the representation, quantification, characterization, and classification of general 2D and 3D shapes [30]. While scatterplots are, formally speaking, not compact subsets of $\mathbb{R}^2$ (as shapes are), we motivate the connection by the fact that (a) for high point counts $N$ and small scatterplot drawings, such as in large SPLOMs, a scatterplot's visual depiction typically converges to a dense representation; and (b) the way humans perceive such a scatterplot is by means of the same type of visual features (e.g. size, skewness, orientation, genus, curvature, thickness) as when looking at more general 2D shapes. Hence, it is interesting to adapt shape descriptors known in the computer vision literature to characterize scatterplots.

Medial axes, also called *skeletons*, are such a powerful descriptor [48]. To define them, we introduce first the distance transform [7] of a shape $\Omega \subset \mathbb{R}^2$ with boundary $\partial\Omega$

$$DT_\Omega(\mathbf{x} \in \mathbb{R}^2) = \min_{\mathbf{y} \in \partial\Omega} \|\mathbf{x} - \mathbf{y}\|. \tag{1}$$

Then, the skeleton of $\Omega$ is defined as

$$S_\Omega = \{\mathbf{x} \in \Omega | \exists \mathbf{f}_1 \in \partial\Omega, \mathbf{f}_2 \in \partial\Omega, \mathbf{f}_1 \neq \mathbf{f}_2, \|\mathbf{x}-\mathbf{f}_1\| = \|\mathbf{x}-\mathbf{f}_2\| = DT_\Omega(\mathbf{x})\} \tag{2}$$

and represents the locus of maximally-inscribed disks in $\partial\Omega$. The points $\mathbf{f}_i$ are the so-called *feature points* of a skeleton point $\mathbf{x}$, i.e., the closest points on $\partial\Omega$ to it. The mapping $FT_\Omega$ associating to a shape point $\mathbf{x} \in \Omega$ its closest (feature) points on $\partial\Omega$ is called the feature transform of $\Omega$ [19]. The pair $(S_\Omega, DT_\Omega)$ is called the Medial Axis Transform (MAT) of the shape $\Omega$, and fully encodes the geometry and topology of this shape, being a dual representation, and a lower-dimensional one, to the boundary-based one. Skeletons are typically very sensitive to $\partial\Omega$, i.e., small perturbations of the latter can create large changes in the former, typically known as spurious branches [48]. To address this in practice, one computes a so-called importance metric $\rho : S \to \mathbb{R}^+$ which next allows computing *regularized* skeletons $S_\Omega^\tau = \{\mathbf{x} \in S_\Omega | \rho(\mathbf{x}) \geq \tau\}$. A well-known such metric, which we use next, sets $\rho(\mathbf{x})$ for $\mathbf{x} \in S_\Omega$ to the shortest path along $\partial\Omega$ between the feature points $\mathbf{f}_1$ and $\mathbf{f}_2$ of $\mathbf{x}$. This allows interpreting $S_\Omega^\tau$ as the skeleton in which all branches of $S_\Omega$ caused by boundary details shorter than $\tau$ length-units have been removed [58]. Besides making their computation robust to noise, regularization also allows extracting *multiscale* skeletons which describe a shape at a user-chosen level of detail [15, 52, 56, 58]. Multiscale MATs of shapes represented as images of resolutions up to $1000^2$ pixels can be computed accurately and automatically in under one second on the CPU [58] and milliseconds on the GPU [12]. Besides shape analysis, MATs have been used in information visualization for graph bundling [12, 57], interactive semantic lenses [21], and visualizing the quality of MP scatterplots [31, 32].

Skeletons have also intriguing (and not fully explored) connections with principal curves. Alternative to Eqn. 2, they can be defined as the local *maxima* of $DT_\Omega$ [52, 58]; and principal curves are defined as the *minima* of a related distance function [18], or *ridges* of the probability density function $S$ [34]. Separately, skeletons are intimately related to graph bundling [12] which, in turn, is identical to the mean shift operator well known in image processing [6], see [22, 60]; on the other side, principal curves can be also defined by mean shift, and have also been used to compute approximations to skeletons for character recognition [26]. As such, skeletons can be seen as a generalization of principal curves for more complex, articulated, shapes. And since principal curves have been well proven to summarize scatterplots, we state a similar potential for skeletons. This is demonstrated next.

### 3.2 Skeleton construction

To construct a skeleton, we need a *compact* 2D shape $\Omega$ embedded in $\mathbb{R}^2$. We construct such a shape from a scatterplot $S$ by computing the discrete kernel density estimation (KDE) of $S$, modeled as a grayscale pixel image

$$I(\mathbf{x} \in \mathbb{R}^2) = \sum_{\mathbf{y} \in S} K\left(\frac{\|\mathbf{x}-\mathbf{y}\|}{R}\right) \tag{3}$$

where $K : \mathbb{R} \to [0,1]$ is a radial basis function like a Gaussian or Epanechnikov (parabolic) kernel of radius $R$. Figure 1c shows $I$ computed for the spiral scatterplot shown in Figs. 1a,b. Identical KDE estimations are used for image clustering [6], graph bundling [22, 60], and 2D cluster detection [31]. The radius $R$ is set to the average distance $\delta$ of a point in $S$ to its nearest-neighbor. This allows upper thresholding $I$ at a value $\varepsilon > 1$ to yield a shape $\Omega$ that contains most points in $S$ and is compact (see Fig. 1d showing $\Omega$ for the spiral scatterplot). Higher $\varepsilon$ values preserve only denser $S$ regions in $\Omega$; lower $\varepsilon$ values capture more of $S$. Outlying points in $S$, which are farther away from nearest neighbors than $\delta$, are ignored, thereby providing a simple but robust way to capture, or *summarize*, the essence of $S$.

Having $\Omega$, we now compute its MAT (following Eqns. 1, 2) using the fast and accurate method [12] which implements the earlier multiscale skeletonization in [58] on the GPU using NVIDIA's CUDA, see Fig. 1e for the distance transform $DT_\Omega$. Next, we regularize $S_\Omega$ using the boundary-length importance metric $\rho$ in [58], already outlined in Section 3.1. Setting $\tau \approx \pi R$ effectively removes all undulations on $\partial\Omega$ caused by the setting of $R$. Figure 1f shows $S_\Omega^\tau$ for the spiral dataset. As visible, this is very close to the principal curves computed by [18] and [34] (Figures 1a,b respectively). However, the complexity of the *fast* algorithm to compute principal curves in [34] is $O(N^2)$ for $N$ points in $S$. The complexity of the method in [18] is not detailed, but this method is much slower: In detail, computing the principal curves in Figures 1 a) and b) takes about 5 seconds ( [18]) and 20 seconds ( [34]), respectively. In contrast, our method, including KDE estimation and

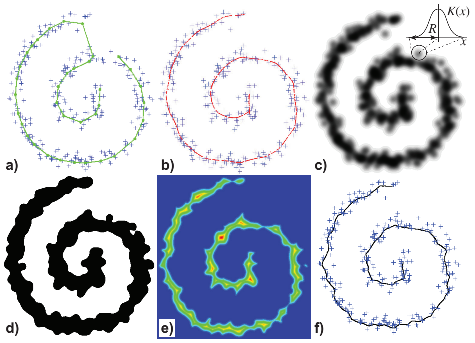

Fig. 1. Principal lines (a,b) vs. skeletons (f) of scatterplots. Skeleton computation includes generating greyscale image $I$ (c), compact shape $\Omega$ (d), and distance transform $DT_\Omega$ (e).

regularized skeletonization, operates in $O(N)$. Practically, we compute the skeleton in Figure 1f) in a few *milliseconds* on a modern PC.

The above procedure delivers the full MAT (skeleton $S_\Omega$, distance transform $DT_\Omega$) and also the feature transform $FT_\Omega$. The process is simple, automatic, and testedly robust, as shown by its earlier use for computing skeletons of curve sets [12, 57] and 2D scatterplots [31].

### 3.3 Dissimilarity Measures

As outlined in Section 2, an important class of methods for scatterplot characterization is based on measuring the *dissimilarity* of the scatterplot shapes. We propose two such novel measures based on the skeletal descriptors introduced in Section 3.2, as follows.

**Hausdorff Distance:** This metric is often used in computer vision and shape analysis to compare two shapes [10]. Given two shapes $\Omega_1$ and $\Omega_2$, their Hausdorff distance is defined as

$$D_H(\Omega_1,\Omega_2) = \max\left(\max_{\mathbf{x}\in\Omega_1}(DT_{\Omega_2}(\mathbf{x})), \max_{\mathbf{x}\in\Omega_2}(DT_{\Omega_1}(\mathbf{x}))\right) \qquad (4)$$

where $DT_\Omega$ is the distance transform of a shape $\Omega$ given by Eqn. 1. Computing $D_H$ is much faster if we consider, instead of two shapes, their skeletons $S_{\Omega_1}$ and $S_{\Omega_2}$ in Eqn. 4, as the skeleton of a shape is typically much smaller, pixel-count wise, than the shape itself. Moreover, our skeletonization framework already allows for a fast computation of distance transforms. $D_H$ accounts for the form of a shape, as its skeleton branches capture its part-whole structure and geometry thereof. However, $D_H$ cannot easily model the similarity of two scatterplots in terms of the amount of correlation they contain.

**Fréchet Distance:** Elongated shapes can be described well by their *centerlines*, which are non-self-intersecting curves locally centered within the shape. The principal curves introduced in Section 2.2 are one instance hereof. For a shape $\Omega$, we compute its centerline as the longest path $\pi \in S^\tau_\Omega$ between any two endpoints of branches of $S^\tau_\Omega$, Given two such 2D paths $\pi_1$ and $\pi_2$ for two shapes, we define their similarity as the Fréchet distance $D_F(\pi_1,\pi_2)$ between them, given by

$$D_F(\pi_1,\pi_2) = \inf_{\pi_1,\pi_2} \max_{t\in[0,1]} \|\pi_1(t)-\pi_2(t)\| \qquad (5)$$

where the curves $\pi_1$ and $\pi_2$ are parametrized over $t \in [0,1]$. We evaluate Eqn. 5 over the pixel-chain representations of $p_i$ extracted from the image skeleton $S^\tau_\Omega$, using the method by Alt and Godau [2]. Compared to the Hausdorff distance $D_H$, $D_F$ better 'pairs' points from the compared centerlines, avoiding one-to-many pairings. However, $D_F$ cannot be readily used for entire skeletons that have multiple branches, as it only captures the main trend in the scatterplot, as given by the longest skeletal path.

## 4 Evaluation

We next explore how well the distance measures introduced in Section 3.3 succeed in capturing the similarity of scatterplots in a SPLOM, as perceived by a human user. For this, we measure how humans perceive similarity between scatterplots (Section 4.1) and compare these values with our distance measures, and other well-known scagnostic distance measures (Section 4.2), computed automatically from the scatterplot shapes (Section 4.3).

### 4.1 Experiment

To measure similarity perception, we performed the following experiment. First, we generated a range of synthetic scatterplots having each $N = 1000$ points, by manipulating four parameters controlling the point distribution: rotation, skewness, linear-to-quadratic shape interpolation, and amount of Gaussian noise. Visual inspection was used to filter out plots that were seen as being too similar, as we next aim to test how good our distance metrics are when comparing reasonably different scatterplots (the hard case). We also added a few scatterplots of actual data from the Abalone dataset, UCI Repository (see Section 5 next). This yielded 29 scatterplots used in the experiment (Figure 2), images are available in the supplementary material . As visible, these span a quite wide range of shapes.

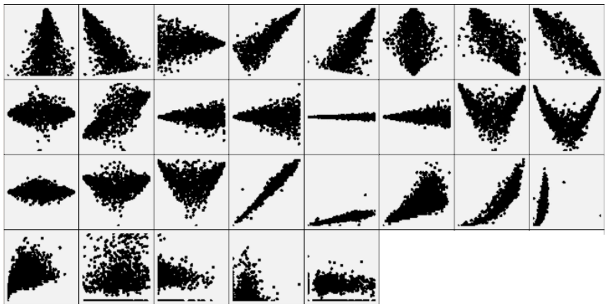

Fig. 2. Base scatterplots representing a wide range of possible point distributions used in the distance-perception experiment.

We asked users to visually compare the 29 scatterplots and rank pairs of plots in terms of six similarity levels: identical, very similar, somewhat similar, somewhat dissimilar, very dissimilar, and opposite. This agreement ranking scheme was chosen to avoid response bias [43, 61] caused by a numerical bipolar and/or asymmetric scale. The levels were then mapped to a 0 to 5 range for their statistical analysis where 0 denotes identical pair of plots.

To perform this, we used Amazon's Mechanical Turk (MTurk) platform [36], an online labor market for Human Intelligence Tasks (HITs). MTurk users, called workers, are given a single task to fulfill in exchange for monetary compensation. Possible tasks range from verification and data entry to image classification and categorization, the latter being close to our task. MTurk has been successfully used as a source of participants for studies given that it simplifies the creation of a large participant pool, a compensation system, and offers a simple study design [5, 33]. Each pair of scatterplots is defined as an item for the workers. Each unique worker is allowed to work on at least one item. MTurk creates batches where a group of items are passed to two different random workers and records their answers and (dis)agreements. At most two workers can be assigned to each batch, so multiple batches must be created for a high enough sample of users. Given that workers are allowed to work on different batches, a single item may be handled by a unique user more than once. We created ten batches with the maximum allowed of two workers each. In the worst case, each item in a batch would be handled by only two unique users once. In our study, a total of 40 unique workers analyzed the items with an average of 13 unique users per item, a minimum of 10 and maximum of 17.

We calculated next the intra-user variance in ranking per item. The average intra-user variance per scatterplot image for unique users with multiple answers per item was 0.246, with a standard deviation of 0.496. The inter-user standard deviation in ranking was 0.784. The inter- and intra- user values show good agreement between users as both deviations lie within one ranking level. We generated a plot pair from a single base scatterplot as a sanity check and workers did, indeed, rank it on average as $< 0.05$(identical). Figure 3 shows the scatterplot pairs found to have the largest perceptual dissimilarity and the largest deviation in ranking dissimilarity, respectively. We see that the

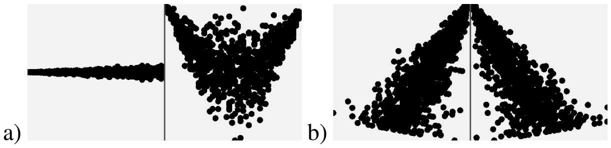

Fig. 3. a) Scatterplot pair with largest perceptual dissimilarity found ($avg = 4.29$, $stddev = 0.59$). Considered either *Very Dissimilar* or *Opposite* (14/15 users).b) Scatterplot pair with the largest deviation ($avg = 2.30$, $stddev = 2.19$) caused by mirroring symmetry with users considering it *Very Similar* and *identical* (7/13) or *Opposite* (5/13).

largest dissimilarity (Figure 3a) was found to be 4.29 on average, i.e., somewhere between the ranks of 4 (very dissimilar) and 5 (opposite), with a quite good consensus ($stddev = 0.59$). The largest deviation in perceptual similarity was found between the plots in Figure 3b, where the average similarity was 2.3, but with a large standard deviation of 2.19, i.e., two levels on our six-point scale, which says that there was disagreement between users in rating the similarity.

For more insight into the perceptual dissimilarity, we constructed a dissimilarity matrix having one entry per plot pair of the SPLOM, with the average ranking for that pair as value computed over all users who ranked the pair. We use this matrix to project the plots, seen as high-dimensional points, by multi-dimensional scaling (MDS), a commonly used MP technique. Figure 4 shows the result, where perceptually similar plots are placed close to each other. We see how orientation plays a role in the perception of similarity – for instance, plots in the top-left of Figure 4 are relatively horizontal, while plots in the bottom-left are diagonal. Separately, we see how shape influences the perceived similarity, by noticing the decreasing quadratic behavior of the scatterplots clustered middle-right in Figure 4 to those top-right. Finally, we see how the horizontal axis of the figure encodes spread in the plots, with widely spread points in the plots to the right and tight line-like point distributions in the plots to the left, respectively.

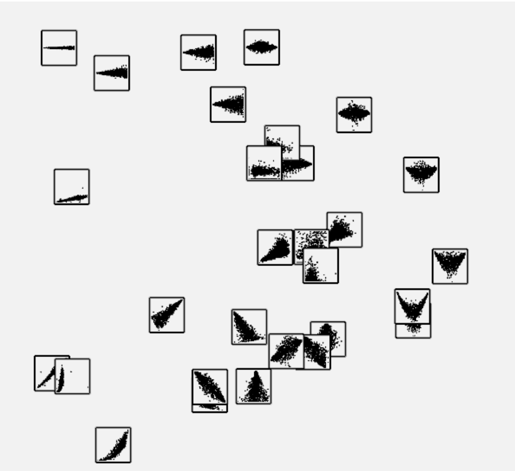

Fig. 4. MDS of the experiment perceptual dissimilarity matrix between scatterplot pairs. Changes in the scatterplots can be observed, such as the linear-to-quadratic behavior (middle-right to top-right) and rotational effect (top-left to bottom-left).

## 4.2 State-of-the-art Dissimilarity Measures

We want to compare our distance metrics introduced in Section 3.3 with existing state-of-the-art scagnostic metrics, which we describe in the following for the reader's convenience.

**Scagnostics Distance:** Wilkinson et al. [63] defined nine measures for characterization of scatterplots based on three geometric graphs: the minimum spanning tree MST, the convex hull ($H$), and the alpha shape graph ($A$) [11] of the point-set $S$, see Table 1. Here, the length of a tree is the sum of the lengths of all its edges; $q_i$ is the $i^{th}$ percent quartile of the edge-length distribution in the MST; $V$ are the nodes of the MST; $V^{(1)}$ and $V^{(2)}$ are nodes in $V$ of degree 1 and 2, respectively; $e(a,b)$ is the 2D vector corresponding to an edge with vertices $a$ and $b$ in the MST; $\mathcal{I}$ is an indicator function returning 1 if the argument is true and 0 otherwise; and $r_S$ is the Spearman rank correlation coefficient of the $x$ and $y$ coordinates of the scatterplot points.

| Measure | Definition |
|---|---|
| $c_{outlying}$ | $length\left(MST_{outliers}\right)/length\left(MST\right)$ |
| $c_{skew}$ | $\left(q_{90}-q_{50}\right)/\left(q_{90}-q_{10}\right)$ |
| $c_{clumpy}$ | $\max_j\left[1-\max_k\left(length(e_k)\right)/length(e_j)\right]$ |
| $c_{sparse}$ | $q_{90}$ |
| $c_{striate}$ | $\frac{1}{\lvert V\rvert}\sum_{v\in V^{(2)}}\mathcal{I}\left(\frac{e(v,a)}{\lVert e(v,a)\rVert}\cdot\frac{e(v,b)}{\lVert e(v,b)\rVert}<-0.75\right)$ |
| $c_{convex}$ | $area(A)/area(H)$ |
| $c_{skinny}$ | $1$ - $\sqrt{4\pi\ area\left(H\right)}/perimeter(A)$ |
| $c_{stringy}$ | $\lvert V^{(2)}\rvert/(\lvert V\rvert-\lvert V^{(1)}\rvert)$ |
| $c_{monotonic}$ | $r_S^2$ |

Table 1. Scagnostics measures from Wilkinson et al. [63].

Once these nine measures are computed, a scatterplot can be seen as a point in $\mathbb{R}^9$. Next, given that all measures are computed on the same scale, dissimilarity can be computed by the squared Euclidean distance in $\mathbb{R}^9$, following Dang et al. [9]. We refer to this as Scagnostics Dissimilarity. While capturing many aspects of shape, this approach has limitations. First, computing $H$ and $A$ is quite sensitive to outliers, which need to be removed by 'peeling' points located on $H$ and $A$, respectively, until these contours change little [63]. Our skeleton descriptors remove the influence of outliers by the joint effect of importance-based skeleton pruning and KDE density thresholding, see Section 3.2. Second, implementing these measures [63] is quite involved, as it requires computing the MST, convex hull, and alpha shape, with a binning scheme for performance considerations. We argue that our skeleton-based descriptor is simpler to realize.

**RVM Distance:** The Rotating Variance Measure (RVM) aims to find linear and nonlinear correlation between dimensions. To do this, Guo [17] notes that a scatterplot exhibiting a strong correlation is one whose density field contains a small high-value band; conversely, a lack of correlation corresponds to a field with local maxima spread throughout the image. Thus, the scatterplot is first transformed to a density field $I$. Tatu et al. [54] compute the density $I(\mathbf{x})$ of a pixel $\mathbf{x}$ as the inverse of the distance of $\mathbf{x}$ to its $k^{th}$ nearest neighbor. Alternatively, we could have computed $I$ as in Eqn. 3, but we stick to the definitions by Tatu et al. for a fair comparison. Having $I$, we can compute the RVM of a scatterplot $S$ as

$$RVM(S) = \frac{1}{\sum_x \min_y v\left(x,y\right)} \tag{6}$$

where $v\left(x,y\right)$ is the minimal mass distribution centered at pixel $\mathbf{x} = (x,y)$ along different directions, *i.e.*

$$v(\mathbf{x}) = \min_{\theta\in[0,\pi]} \frac{\sum_{\alpha\in[-L,L]} \alpha I(\mathbf{x}+\alpha\mathbf{u}(\theta))}{\sum^{s}_{\alpha\in[-T,T]} I(\mathbf{x}+\alpha\mathbf{u}(\theta))} \tag{7}$$

where $\mathbf{u}(\theta)$ is a 2D unit vector with orientation $\theta$, $\alpha$ is a distance in the range $[-L,L]$ along the direction $\mathbf{u}$, and $I(\mathbf{x})$ is the density at position $\mathbf{x}$ (Eqn. 3). Eqn. 7 is trivially parallelizable on CUDA as $v(\mathbf{x})$ is independent for each pixel $\mathbf{x}$. Finally, the RVM dissimilarity of two scatterplots $S_1$ and $S_2$ is defined by

$$D_{RVM}\left(S_1,S_2\right) = \left|RVM\left(S_1\right) - RVM\left(S_2\right)\right|. \tag{8}$$

Parameter settings for computing $D_{RVM}$ are discussed further in Section 4.5.

## 4.3 Quantitative Comparison of Dissimilarity Metrics

We define the perceptually based dissimilarity matrix as our ground truth, i.e., as the values that the metrics computed automatically from scatterplots should emulate. To gauge the (dis)agreement of the two, we compute our proposed metrics $D_F$ and $D_H$ (Section 3.3) and the existing metrics $D_S$ and $D_{RVM}$ (Section 4.2) and compare them with the ground truth. To allow for such a comparison, we normalize the dissimilarity values to the unit interval (using minimum and maximum values, where the minimum is always 0 due to the sanity check). We computed the element-wise absolute difference to the ground truth, results are shown in Fig. 5. We see that both our measures $D_F$ and $D_H$ yield the best agreement with the ground truth, i.e., lowest values of average and maximum differences. We verified this further by computing the Frobenius norm $F$ of the differences between the perceptual dissimilarity matrix and the four matrices computed with the four measures obtaining $F = 16.676$ for $D_S$, $F = 16.906$ for $D_{RVM}$, $F = 10.119$ for $D_H$, and $F = 11.5043$ for $D_F$. These results show that our proposed dissimilarity metrics $D_H$ and $D_F$ perform better than the existing well-accepted metrics $D_S$ and $D_{RVM}$ in terms of closeness to perceptual distances. Moreover, we see that our original hypothesis, i.e. the fact that skeleton descriptors are effective in capturing the perceptual similarity of scatterplots, does indeed hold, as both $D_F$ and $D_H$ are computed based on skeletons. In fact, $D_H$, which follows the skeletal information closest, offers the best agreement with the perceptually-measured distances.

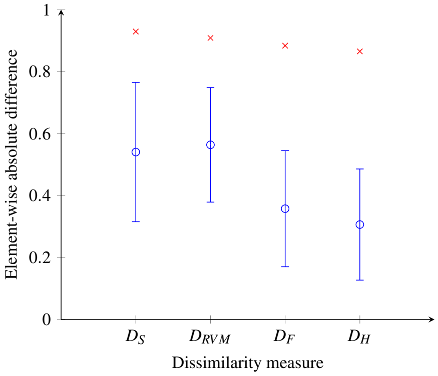

Fig. 5. Element-wise absolute differences averages and standard deviation between dissimilarity measures and perceptual similarity matrix. Proposed methods have lower average element-wise difference. Red: maximum element-wise difference.

Human perception of scatterplots, as shown in Figure 4, may not be invariant to mirroring and rotational transformations. While $D_S$ and $D_{RVM}$ are invariant to mirroring transformations, our proposed measures based on skeleton descriptors are not. In case of rotational transformations, the monotonicity feature in $D_S$ may change. This leaves $D_{RVM}$ as the only rotational invariant metric in our comparison.

In the following, we want to discuss a few interesting cases. For Figure 3b, users were not able to agree on the scatterplot-pair similarity. The mirroring symmetry visible in this figure caused a large standard deviation in ranking. This strong orientation effect on similarity is captured also by the shape metrics we propose, and best by $D_H$: The average normalized perceptual dissimilarity for the plots in Figure 3b is 0.54, while the normalized Hausdorff dissimilarity $D_H$ is 0.74, which is in line with the user study. In contrast, the normalized Scagnostics Dissimilarity $D_S$ is only 0.007. This indicates that $D_S$ finds these plots to be very similar, which contradicts the users' perception. Figure 6 shows another example. For this scatterplot pair, $D_{RVM}$ delivers that the plots are very similar ($D_{RVM} = 0.01$), given that both distributions can be characterized by a narrow high-density band. In increasing dissimilarity order, the other computed metrics are $D_S = 0.47$, $D_F = 0.58$, and $D_H = 0.82$, where the latter two are equally close to the perceptual distance 0.7 and much closer than the other two metrics.

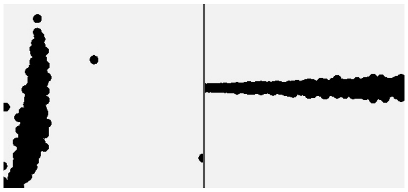

Fig. 6. Performance of computed distance metrics for comparing two plots having a perceptual distance of 0.7: $D_{RVM} = 0.01$, $D_S = 0.47$, $D_F = 0.58$, and $D_H = 0.82$. Narrow high-density plots are considered similar by $D_{RVM}$ regardless of orientation.

### 4.4 Visual Comparison of Dissimilarity Metrics

To gain more insight on how $D_F$ and $D_H$ actually compare scatterplots as compared to $D_S$ and $D_{RVM}$, we show the respective dissimilarity matrices using MDS, similarly to how we did it in Figure 4 for the perceptual distances. Figure 7a shows the MDS projection for $D_F$. We can clearly see the effect of scatterplot orientation: Starting from the top in counter-clockwise direction, we see clusters of scatterplots with roughly vertical, positive-slope, horizontal, and negative slopes. The linear-to-quadratic plot shape variation is captured in the lower-right corner (red inset in Figure 7a).

As explained in Section 3.3, the Hausdorff distance $D_H$ improves upon $D_F$ by also capturing scatterplot shape, apart from the main trend. Figure 7b shows this. The plot orientation change is captured by the vertical axis. In contrast to the $D_F$ projection, $D_H$ also allows capturing the scatterplot spread – if we focus on the upper part of Fig. 7b, we can see the change from right-skewed towards a left-skewed distributions. In contrast to the above, $D_{RVM}$ (Fig. 7c) mainly separates plots having a narrow spread (thin shapes, more to the right in the figure) from wide-spread plots (thick shapes, center and left in the figure). However, plots having different shapes (A) or different orientations (B) are seen as similar. Finally, Figure 7d shows the MDS plot of the Scagnostic distance $D_S$. We see how plots appear to be grouped based on skinniness (similar to $D_{RVM}$), with very skinny ones being separated from the others (A). However, orientation and/or correlation are not well captured; for instance, horizontal plots are seen as similar (B), but so are plots having strong direct and inverse correlations (C). Overall, this visual analysis strengthens the quantitative comparison from Sec. 4.3 in telling us that $D_F$ and especially $D_H$ capture perceived plot similarities better than $D_S$ and $D_{RVM}$.

### 4.5 Computational Performance Comparison

We detail how we implemented the four distance metrics introduced in Section 3.3 to ease replication and compare computational performance figures.

For RVM and Scagnostics Dissimilarity we used binning on a $75 \times 75$ grid to improve efficiency, as suggested in literature [63]. The first three steps of the RVM measure, namely density estimation, mass distribution computation (Eqn. 7), and pixel-wise minimum calculation (Eqn. 6) were implemented using CUDA. Angles $\theta \in [0, \pi]$ were sampled with a step of $0.1\pi$ radians. The oriented line segment of length $2L$ was sampled at each consecutive pixel. For all scatterplots, $D_{RVM}$ computation took between 45.5 and 82.1 $ms$. Arguably, this is also the most complex measure to implement.

For $D_S$, the main bottleneck is the computation of the MST graph. We used for this the VTK Toolkit [27]. Computing the MST using a $75 \times 75$ grid for one of the base scatterplots in Figure 2 takes between 293 and 1,075 $ms$. Reducing the bin resolution to $60 \times 60$ lowers this to the range of 122 to 446 $ms$. Overall, $D_S$ is the slowest measure to compute.

Since $D_F$ and $D_H$ are both skeleton-based, they share several computation steps. As outlined in Section 3.2, we compute the MAT (regularized skeleton and distance transform) using the fast CUDA-based method [12] (which is publicly available). For the base scatterplots, this took between 15.8 and 21.6 $ms$ per plot. Given that the skeletonization process already computes the distance transform $DT_S$ of a skeleton, computing $D_H$ via Eqn. 4 requires a simple maximization of $DT_{S_1}$ over the pixels of $S_2$, which takes under 1 $ms$ per plot. For $D_F$, we compute the longest path between endpoints of a skeleton $S$ using Dijkstra's algorithm on the pixel connectivity graph of $S$ as implemented in VTK. This took between 4.2 and 47.8 $ms$ per plot. This large variance is

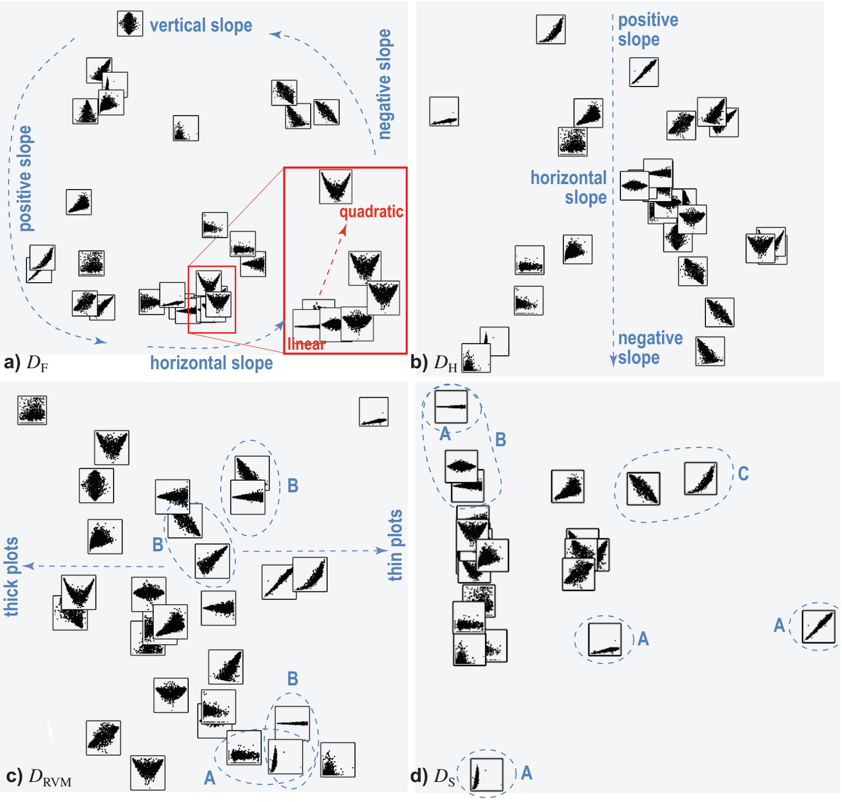

Fig. 7. MDS of dissimilarity matrices between scatterplot pairs computed with our Fréchet (a) and Hausdorff (b) distances, and with the RVM (c) and Scagnostics (d) distances. Zoomed inset (red, (a)) captures linear-to-quadratic plot behavior.

caused by the large variance in number of skeleton endpoints $E$, given that the number of paths between endpoints is $E^2$. Having the longest path, $D_F$ is computed in under 1 *ms* per plot. Overall, $D_H$ took between 16.8 and 22.6 *ms* per plot, and $D_F$ took between 21.4 and 68.1 *ms* per plot, respectively.

Given the above, we see that our skeleton-based measures $D_H$ and $D_F$ are the first- and second-fastest dissimilarity metrics, respectively. Given they also approximate the perceptual distance best (Section 4.3), we argue that they are both effective and efficient ways to characterize scatterplots in a scagnostics context.

## 5 Visual Analysis

So far, we have shown how skeleton-based descriptors can effectively and efficiently capture the perceptual similarity of scatterplots in a SPLOM. However, they can be used for more. We show next how we leverage them to create summarized visual representations for the point distributions in scatterplots. These representations can be used to quickly scan a SPLOM to find interesting (or similar) patterns in the data. As a case study, we use the Abalone dataset from the UCI Repository, which contains $N = 4177$ instances describing snails along eight numerical and one categorical anatomical dimensions [4, 62]. In all cases, we display the original scatterplots in the upper-triangular part of the SPLOM, and our summarizations (of the same scatterplots, i.e., not flipped along the diagonal) in the corresponding lower-triangular part. This way, one can compare the scatterplots, respectively summarizations, among themselves, but also link a scatterplot to its summarization (see Fig. 9 and further).

### 5.1 Visual Encoding

We first describe the proposed scatterplot summarizations. For each of them, we outline the task(s) it is aimed to support.

**Model comparison:** Bivariate relationships can be explored via scatterplots. Mathematical or sketch-based models can be defined and compared against the principal data trends. High model-data agreement means a high predictive power for the model. We support visually assessing this agreement by color mapping the distance from the skeleton to a given model at every skeleton point $\mathbf{x}$. If the model is described by a 2D curve $M$ given e.g. as $y = f(x)$, then this distance is $DT_M(\mathbf{x})$, the distance transform (Eqn. 1) of $M$ indexed over the skeleton points $\mathbf{x}$. Figure 9 (blue cells) shows this for the model $f(x) = 0.6x^2 + 0.3x$, using a color map from dark-green (low) to yellow (medium-low) to orange (medium-high) to red (high). Dark-green colors encode a low distance, thus, a good model-data agreement, such as in the SPLOM columns 1 and 3. In contrast, column 2 shows warmer colors, thus high disagreement. In the red inset, we change the model to $g(x) = 3x$ for a subset of the SPLOM (four scatterplots). We now see a much higher disagreement (red skeletons) for the top two of these plots (A,B). This was to be expected, since those scatterplot shapes are clearly more similar to a quadratic curve than to a straight line. We also support sketching any model interactively by drawing a curve (represented as a sequence of connected pixels) in a small pop-up window and comparing the scatterplots in the SPLOM against the drawn model, see accompanying video.

**Data summarization:** So far, our skeletons summarize only the *shape* of a scatterplot $S$. We extend this visual encoding by annotating each skeleton point $\mathbf{x}$ with information from all scatterplot points $\mathbf{y}$ that $\mathbf{x}$ summarizes. For this, we map each $\mathbf{y}$ to its closest $\mathbf{x} \in S_\Omega^\tau$. Note that this mapping is many-to-one in areas where the scatterplot shape $S$ is convex, one-to-many mapping where $S$ is concave, and one-to-one where $S$ is straight [48]. We easily compute this mapping as it is equal to the feature transform $FT_{S_\Omega^\tau}$ (see Section 3.1), or alternatively by gathering all $\mathbf{y}$ located on lines perpendicular to the tangent to $S_\Omega^\tau$ at $\mathbf{x}$. We next aggregate the values of all $\mathbf{y}$ found for a given skeleton point $\mathbf{x}$ and display the result on $\mathbf{x}$ using the same color mapping as used above for model comparison. We next propose two such aggregations.

*Density mapping:* We aggregate the densities $I(\mathbf{y})$ of scatterplot points (Eqn. 3) weighted by the inverse distance $1/\|\mathbf{x} - \mathbf{y}\|$ so that points closer to $\mathbf{x}$ have a higher contribution. Figure 8a shows how this makes skeletal summarizations represent well both the shape and local density of the scatterplots. We see, for example, a high-density area

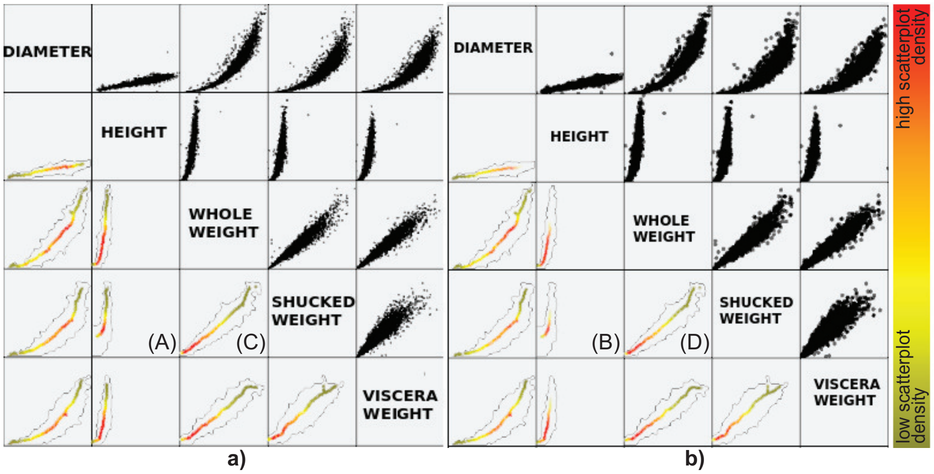

Fig. 8. (a) Density mapping summarization. Color indicates local scatterplot density, in a direction orthogonal to the skeleton. (b) Combined mapping summarization of local data-fit to a model (opacity) and local data density (color).

of abalones with low to mid-range *height* (thus a skewed distribution), while for *diameter* the distribution is much more centered (normal distribution).

*Combined mapping:* We can map two scalar properties of the scatterplot $S$ to two attributes, e.g., color and opacity, of the skeleton points. Figure 8b maps the model-data agreement explained earlier to opacity (for a model $f(x) = 0.6x^2 + 0.3x$), and point density to color, respectively. This shows both *how much* and *where* data agrees to a model. Low-agreement areas naturally filter out (due to transparency), helping to focus on high-agreement areas. In these, color tells how many points support the agreement, so how strong the agreement is. For instance, we see that while *height* is strongly correlated with *shucked weight* (red colors on skeleton in Fig. 8a, detail A), this correlation poorly fits our quadratic model $f$ (few opaque points on skeleton in Figure 8b, detail B). In contrast, the correlation of *whole weight* with *shucked weight* fits the model very well: The skeleton in Figure 8a (detail C) shows about the same number of points as the one in Fig. 8b (detail D).

*Other encodings:* Besides the above, other encodings of scatterplot data on the skeleton are easily possible. For instance, we can immediately encode the *local thickness* of the scatterplot by simply considering the values of $DT_{S^\tau_\Omega}$ over the scatterplot points $\mathbf{y}$. This elegantly subsumes the $D_{RVM}$ metric (Section 4.2) in our framework. Additionally, we can quantify the jaggedness of a scatterplot using the number of branches of its skeleton [48]. For space limitation reasons, we do not detail such options further.

### 5.2 Interactive Exploration

For SPLOMS with many dimensions $d$, appropriate interaction techniques must be provided so one can easily select a subset of interest to explore further. Our visual encodings presented so far allow exploring bivariate relationships for a small number of scatterplots. Visual exploration of SPLOMs often uses overview abstractions and/or interaction based on the 'interestingness' of representative plots. We provide interaction methods for interactive visual analysis based on dissimilarity measures for locating similar plots and for model evaluation, as follows.

**Scatterplot querying:** We extend the model-fit functionality described above by applying it to the level of *scatterplots* within a SPLOM instead of each *point* in a scatterplot skeleton. For this, we color-code the fit of a scatterplot with a given model on the scatterplots' backgrounds. This helps finding those scatterplots in the SPLOM having high and/or low fits. Once found, these can be explored in detail using the pixel-level mechanisms outlined in Section 5.1. We can also filter scatterplots according to their normalized fit to the model. Alternatively, we can select a scatterplot of interest $S$ and then only show the other scatterplots (from a large SPLOM, whose entire display would not fit the screen) that are more similar to $S$ than a user-given threshold $\varepsilon$. Figure 10a shows an example for the model $f(x) = 0.6x^2 + 0.3$. The Hausdorff distance $D_H$ (which, as shown in Section 4, best captures perceptual distance) of the scatterplots to the model is computed and color-coded using a colormap from light green (similar) to light orange (dissimilar), where brighter colors are used to prevent skeleton visibility issues. We define a threshold $\varepsilon = 0.10$ of the normalized dissimilarity to show scatterplots that fit the model well enough to allow for predictions. This threshold value corresponds to a point between our perceptual defined experimental range of *Identical* and *Very Similar*. Hence, scatterplots below this threshold value would be considered perceptually very similar by users.

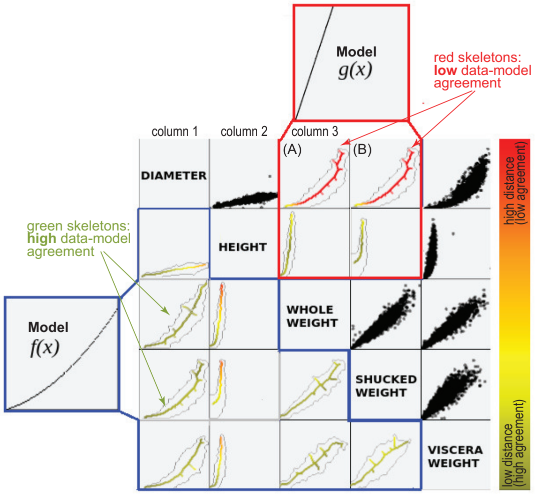

Fig. 9. Summarized agreement of a scatterplot with an analytic model. Blue SPLOM cells show the data fit to $f(x) = 0.6x^2 + 0.3x$. Red cells show the data fit to $g(x) = 3x$. Colors map the model-data fit from dark-green (low) to red (high).

We see that only the *diameter vs. whole weight* scatterplot, outlined blue in Figure 10, fits the model well. Other scatterplot cells that are still close to the model show up in greenish color; orange indicates scatterplot cells that have a much shallower or steeper slope and are thus very far from the model. In contrast, in Figure 10b we select the scatterplot *whole weight vs. shucked weight* (marked red), and show all other scatterplots more similar to it than $\varepsilon$. Three such scatterplots are found, outlined blue in the figure. As visible, these are indeed very similar in shape, thickness, and orientation to our selected plot. Both cases in Figure 10 can be thought of *querying* for similar scatterplots in a large SPLOM given either a model (Figure 10a) or an example (Figure 10b). For an example in a larger SPLOM (50 dimensions) see accompanying video. As such, our technique is similar to approaches well known in shape retrieval, such as query-by-example [53, 65]. Not surprisingly, at this point of reading, we see that many such methods

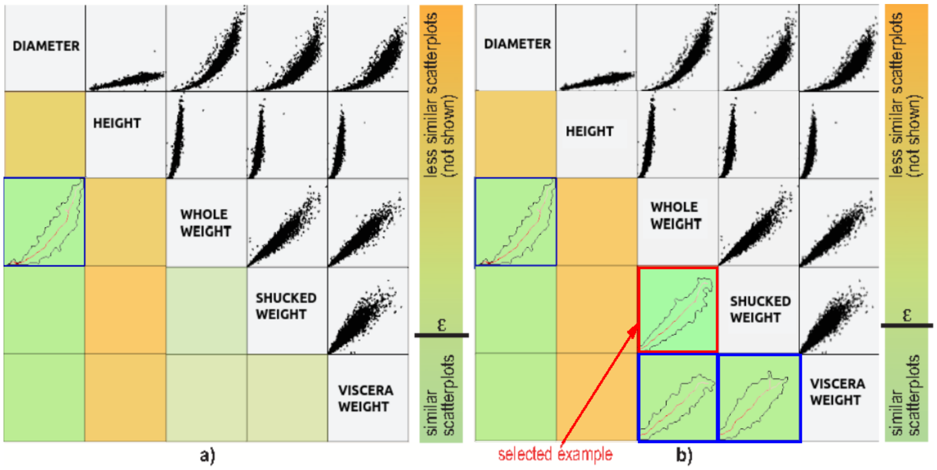

Fig. 10. Focus-and-context shows only scatterplots fitting a model (a) or fitting a given scatterplot example (b) better than a given user threshold $\varepsilon$. In both cases, the scatterplot cells returned by the 'query' are outlined in blue.

also use skeleton descriptors. On a different note, our visual encoding of scatterplot querying presented above follows the well-known Shneiderman mantra [47]: *overview* (we show the SPLOM with color-coded dissimilarities on background); *zoom-and-filter* (select a plot of interest for more information); and *details on demand* (we next show this plot, and the most similar ones to it, at high resolution).

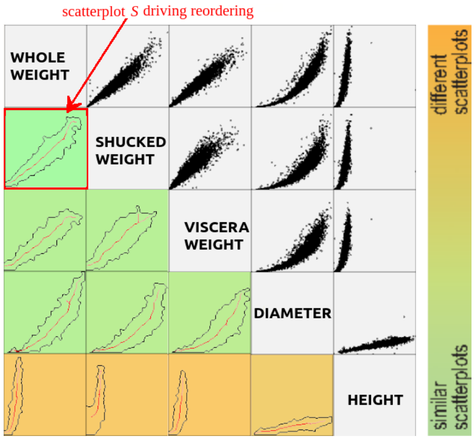

Fig. 11. Reordering of SPLOM based on similarity with the scatterplot outlined red. Background color shows scatterplot similarities.

**Reordering:** Reordering rows/columns of SPLOMs is a common task in optimization of SPLOM analysis. We propose doing this based on the dissimilarity $D_H$ to a selected scatterplot $S$. The first two dimensions to be placed in the new SPLOM (left-to-right, top-to-bottom) are the ones $S$ uses. Next, we add to the new SPLOM the dimension that minimizes the average dissimilarity to $S$ given the already placed dimensions, and so on, until all dimensions $d$ have been accounted for. Figure 11 shows the SPLOM reordering with respect to the scatterplot *diameter vs. whole weight*, which was also selected in Figure 10a. As visible, scatterplots close to the one driving the reordering (marked red) are more similar to it than ones being further in the SPLOM. We also color the scatterplot-cell backgrounds by the similarity $D_H$. We see that, indeed, cells close to $S$ are more similar. More valuably, we can now *explain* what makes scatterplots different: For instance, we see that all scatterplots on the bottom row (orange) are much more different than the other ones (green). Hence, the key dimension distinguishing scatterplots in the SPLOM from the selected $S$ is *height*.

## 6 Discussion and Conclusion

We have proposed a novel way to quantify and represent scatterplots in a SPLOM in a summarized manner. For this, we leverage properties of skeleton-based descriptors, well-known and used since long in computer vision and shape analysis. We first use skeletons to summarize complex scatterplot shapes in a compact curve-set, and show that this representation is closer to perceptual distances between a wide variation of scatterplots, and faster to compute, than established scatterplot similarity metrics. This allows us next to use skeleton-based descriptors to summarize more information about the scatterplot, such as the local point density and confidence-fit to analytic or sketch-based models.

The main limitation of our approach described in Section 3.1 is the assumption that a scatterplot's visual depiction typically converges to a dense visual representation. This is not always the case, e.g. for scatterplots having just a few points, or scatterplots where there are large gaps between highly concentrated point clusters. However, we argue that the first case is less interesting from a scagnostics perspective, while the latter one can be easily handled by existing cluster-based scagnostics techniques. Separately, we note that the skeleton representation is unable, without the aid of our additional data summarization visual encoding, to caputre the density distribution in a scatterplot. Density-weighted skeletal (dis)similarity measures may be further investigated to improve the comparison of point distributions with locally different point densities.

To our knowledge, this is the first time that shape skeletons have been used to summarize *data*, rather than just *shape*, in information visualization (or, for that matter, in other fields, too). Additionally, we use skeletons to depict the encoded information in a compact way, thereby enriching the palette of available scagnostic techniques. We show how such descriptors can be leveraged to easily pose model-fit queries including sketch-based models and queries-by-example to find relevant scatterplots in a large SPLOM. Our proposal applies to any SPLOM, requiring only access to the individual 2D scatterplots; robust to noise, given known skeleton regularization properties; automatic, requiring no parameter tweaking; and fast and simple to compute, given existing GPU parallelizations of 2D skeletonization [12]. At a higher level, our work shows how bridges can be formed (and exploited) between information visualization and the more classical shape analysis, image processing, and shape retrieval domains.

Future work envisages integrating our skeleton-based descriptors in existing frameworks for SPLOM exploration [9, 28], adding more data attributes to the skeleton to better *capture* scatterplot similarity, enhancing the skeleton visualization to better *explain* this similarity, and extending this approach to 3D scatterplots, based on recent breakthroughs in fast-and-accurate 3D skeleton computation [23, 24].

## Acknowledgments

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under contract LI-23/1 and the project MOTO (H2020-SESAR-2015-1), grant 699379, offered by the European Commission.

## References

[1] G. Albuquerque, M. Eisemann, and M. Magnor. Perception-based visual quality measures. In *Proc. IEEE VAST*, pp. 13–20, 2011.

[2] H. Alt and M. Godau. Computing the fréchet distance between two polygonal curves. *Intl J Comp Geom Appl*, 5(1):75–91, 1995.

[3] M. Aupetit. Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing*, 10(7-9):1304–1330, 2007.

[4] C. Blake and C. J. Merz. UCI repository of machine learning datasets, 1998.

[5] M. Buhrmester, T. Kwang, and S. D. Gosling. Amazon's mechanical turk a new source of inexpensive, yet high-quality, data? *Perspect Psychol Sci*, 6(1):3–5, 2011.

[6] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE TPAMI*, 24(5):603–619, 2002.

[7] L. Costa and R. Cesar. *Shape Analysis and Classification*. CRC Press, 2001.

[8] R. da Silva, P. Rauber, R. Martins, R. Minghim, and A. Telea. Attribute-based visual explanation of multidimensional projections. In *Proc. EuroVA*, 2015.

[9] T. N. Dang and L. Wilkinson. Scagexplorer: Exploring scatterplots by their scagnostics. In *Visualization Symposium (PacificVis), 2014 IEEE Pacific*, pp. 73–80. IEEE, 2014.

[10] M. de Berg, M. van Krefeld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications,* $2^{nd}$ *edition*. Springer, 2000.

[11] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE T Inform Theory*, 29:551–559, 1983.

[12] O. Ersoy, C. Hurter, F. Paulovich, G. Cantareira, and A. Telea. Skeleton-based edge bundles for graph visualization. *IEEE TVCG*, 17(2):2364 – 2373, 2011.

[13] R. Etemadpour, R. C. da Motta, J. G. de Souza Paiva, R. Minghim, M. C. F. de Oliveira, and L. Linsen. Role of human perception in cluster-based visual analysis of multidimensional data projections. In *Information Visualization Theory and Applications (IVAPP), 2014 International Conference on*, pp. 276–283. IEEE, 2014.

[14] R. Etemadpour, R. Motta, J. G. de Souza Paiva, R. Minghim, M. C. F. de Oliveira, and L. Linsen. Perception-based evaluation of projection methods for multidimensional data visualization. *IEEE TVCG*, 21(1):81–94, 2015.

[15] A. Falcão, J. Stolfi, and R. Lotufo. The image foresting transform: theory, algorithms, and applications. *IEEE TPAMI*, 26(1):19–29, 2004.

[16] S. Gerber, P.-T. Bremer, V. Pascucci, and R. Whitaker. Visual exploration of high dimensional scalar functions. *IEEE TVCG*, 16(6):1271–1280, 2010.

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

[18] T. Hastie and W. Stuetzle. Principal curves. *J. American Statistical Association*, 84(406):502–516, 1989.

[19] W. Hesselink and J. Roerdink. Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform. *IEEE TPAMI*, 30(12):2204–2217, 2008.

[20] N. Heulot, M. Aupetit, and J.-D. Fekete. ProxiLens: Interactive exploration of high-dimensional data using projections. In *Proc. EuroVis Workshop on Visual Analytics using Multidimensional Projections*, pp. 11–15, 2013.

[21] C. Hurter, O. Ersoy, and A. Telea. MoleView: An attribute and structure-based semantic lens for large element-based plots. *IEEE TVCG*, 17(12):2600–2609, 2011.

[22] C. Hurter, O. Ersoy, and A. Telea. Graph bundling by kernel density estimation. *CGF*, 31(3):435–443, 2012.

[23] A. Jalba, J. Kustra, and A. Telea. Surface and curve skeletonization of large 3D models on the GPU. *IEEE TPAMI*, 35(6):1495–1508, 2013.

[24] A. Jalba, A. Sobiecki, and A. Telea. An unified multiscale framework for planar, surface, and curve skeletonization. *IEEE TPAMI*, 38(1):30–45, 2016.

[25] P. Joia, F. V. Paulovich, D. Coimbra, J. Cuminato, and L. Nonato. Local affine multidimensional projection. *IEEE TVCG*, 17:2563–2571, 2011.

[26] B. Kegl and A. Kryzak. Piecewise linear skeletonization using principal curves. *IEEE TPAMI*, 24(1):59–74, 2002.

[27] Kitware, Inc. The VTK visualization toolkit, 2017. `www.vtk.org`.

[28] D. J. Lehmann, G. Albuquerque, M. Eisemann, M. Magnor, and H. Theisel. Selecting coherent and relevant plots in large scatterplot matrices. *Computer Graphics Forum*, 31(6):1895–1908, 2012.

[29] S. Lespinats and M. Aupetit. CheckViz: Sanity check and topological clues for linear and non-linear mappings. *Comput. Graph. Forum*, 30(1):113–125, 2011.

[30] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001, 1998.

[31] R. Martins, D. Coimbra, R. Minghim, and A. Telea. Visual analysis of dimensionality reduction quality for parameterized projections. *Computers & Graphics*, 41:26–42, 2014.

[32] R. Martins, R. Minghim, and A. Telea. Explaining neighborhood preservation for multidimensional projections. In *Proc. CGVC*. Eurographics, 2015.

[33] W. Mason and S. Suri. Conducting behavioral research on amazons mechanical turk. *Behavior research methods*, 44(1):1–23, 2012.

[34] U. Ozertem and D. Erdogmus. Locally defined principal curves and surfaces. *JMLR*, 12:1249–1286, 2011.

[35] A. V. Pandey, J. Krause, C. Felix, J. Boy, and E. Bertini. Towards understanding human similarity perception in the analysis of large sets of scatter plots. In *Proc. ACM CHI*, pp. 3659–3669, 2016.

[36] G. Paolacci, J. Chandler, and P. Ipeirotis. Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making*, 5(5):411–419, 2010.

[37] F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz. Least square projection: a fast high-precision multidimensional projection technique. *IEEE TVCG*, 14(3):564–575, 2008.

[38] F. V. Paulovich, F. Toledo, G. P. Telles, R. Minghim, and L. G. Nonato. Semantic wordification of document collections. *Computer Graphics Forum*, 31(3):1145–1153, 2012.

[39] C. K. Reddy, S. Pokharkar, and T. K. Ho. Generating hypotheses of trends in high-dimensional data skeletons. In *Proc. IEEE VAST*, pp. 139–146, 2008.

[40] R. A. Rensink and G. Baldridge. The perception of correlation in scatterplots. *Computer Graphics Forum*, 29(3):1203–1210, 2010.

[41] M. Scherer, J. Bernard, and T. Schreck. Retrieval and exploratory search in multivariate research data repositories using regressional features. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, pp. 363–372. ACM, 2011.

[42] T. Schreck, T. von Landesberger, and S. Bremm. Techniques for precision-based visual analysis of projected data. *Inf. Vis.*, 9(3):181–193, 2010.

[43] N. Schwarz, B. Knäuper, H.-J. Hippler, E. Noelle-Neumann, and L. Clark. Rating scales numeric values may change the meaning of scale labels. *Public Opinion Quarterly*, 55(4):570–582, 1991.

[44] M. Sedlmair and M. Aupetit. Data-driven evaluation of visual quality measures. In *Computer Graphics Forum*, vol. 34, pp. 201–210. Wiley Online Library, 2015.

[45] M. Sedlmair, A. Tatu, T. Munzner, and M. Tory. A taxonomy of visual cluster separation factors. *Comput. Graph. Forum*, 31(3):1335–1344, 2012.

[46] L. Shao, T. Schleicher, M. Behrisch, T. Schreck, I. Sipiran, and D. A. Keim. Guiding the exploration of scatter plot data using motif-based interest measures. *Journal of Visual Languages & Computing*, 36:1–12, 2016.

[47] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. ACM VL*, pp. 336–343, 1996.

[48] K. Siddiqi and S. Pizer. *Medial Representations: Mathematics, Algorithms and Applications*. Springer, 1999.

[49] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan. Selecting good views of high-dimensional data using class consistency. *Computer Graphics Forum*, 28(3):831–838, 2009.

[50] C. Sorzano, J. Vargas, and A. Pascual-Montano. A survey of dimensionality reduction techniques, 2014. *arxiv.org/pdf/1403.2877*.

[51] Y. Sun and M. G. Genton. Functional boxplots. *Journal of Computational and Graphical Statistics*, 20(2):316–334, 2011.

[52] A. Tagliasacchi, T. Delame, M. Spagnuolo, N. Amenta, and A. Telea. 3D skeletons: A state-of-the-art report. *CGF*, 35(2):573–597, 2016.

[53] J. Tangelder and R. Veltkamp. A survey of content based 3D shape retrieval methods. *Multimed Tools Appl*, 39:441–471, 2008.

[54] A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H. Theisel, M. Magnor, and D. Keim. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *Proc. IEEE VAST*, pp. 59–66, 2009.

[55] A. Tatu, P. Bak, E. Bertini, D. Keim, and J. Schneidewind. Visual quality metrics and human perception: an initial study on 2d projections of large multidimensional data. In *Proc. AVI*, pp. 49–56. ACM, 2010.

[56] A. Telea. Feature preserving smoothing of shapes using saliency skeletons. In *Proc. VMLS*, pp. 153–170. Springer, 2012.

[57] A. Telea and O. Ersoy. Image-based edge bundles: simplified visualization of large graphs. *Comput. Graph. Forum*, 29(3):843–852, 2010.

[58] A. Telea and J. J. van Wijk. An augmented fast marching method for computing skeletons and centerlines. In *Proc. VisSym*, pp. 251–259. Springer, 2002.

[59] J. W. Tukey and P. A. Tukey. Computer graphics and exploratory data analysis: An introduction. *The Collected Works of John W. Tukey: Graphics: 1965-1985*, 5:419, 1988.

[60] M. van der Zwan, V. Codreanu, and A. Telea. CUBu: Universal real-time bundling for large graphs. *IEEE TVCG*, 22(12):2550–2563, 2016.

[61] T. Wakita, N. Ueshima, and H. Noguchi. Psychological distance between categories in the likert scale: Comparing different numbers of options. *Educational and Psychological Measurement*, 72(4):533–546, 2012.

[62] S. Waugh. Extending and benchmarking cascade-correlation. *Dept of Computer Science, University of Tasmania, Ph. D. Dissertation*, 1995.

[63] L. Wilkinson, A. Anand, and R. Grossman. High-dimensional visual analytics: Interactive exploration guided by pairwise views of point distributions. *IEEE TVCG*, 12(6):1363–1372, 2006.

[64] A. Yates, A. Webb, M. Sharpnack, H. Chamberlin, K. Huang, and R. Machiraju. Visualizing multidimensional data with glyph sploms. *Computer Graphics Forum*, 33(3):301–310, 2014.

[65] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37:1–19, 2004.