This paper is © IEEE and appeared reformatted in *IEEE Trans. on Visualization and Computer Graphics, 20(3): 351-364, 2014. DOI 10.1109/TVCG.2013.101*

A Structure-Based Distance Metric for High-Dimensional Space Exploration with Multi-Dimensional Scaling

Jenny Hyunjung Lee, Kevin T. McDonnell, *Member, IEEE,* Alla Zelenyuk, Dan Imre and Klaus Mueller*, Senior Member, IEEE*

Abstract — Although the Euclidean distance does well in measuring data distances within high-dimensional clusters, it does poorly when it comes to gauging inter-cluster distances. This significantly impacts the quality of global, low-dimensional space embedding procedures such as the popular multi-dimensional scaling (MDS) where one can often observe non-intuitive layouts. We were inspired by the perceptual processes evoked in the method of parallel coordinates which enables users to visually aggregate the data by the patterns the polylines exhibit across the dimension axes. We call the path of such a polyline its *structure* and suggest a metric that captures this structure directly in high-dimensional space. This allows us to better gauge the distances of spatially distant data constellations and so achieve data aggregations in MDS plots that are more cognizant of existing high-dimensional structure similarities. Our bi-scale framework distinguishes far-distances from near-distances. The coarser scale uses the structural similarity metric to separate data aggregates obtained by prior classification or clustering, while the finer scale employs the appropriate Euclidean distance.

Index Terms— Information Visualization, Multivariate Visualization, Clustering, High-Dimensional Data, Visual Analytics

1 Introduction

THE recognition of relationships embedded in high-dimensional (multi-attribute) data remains a challenging task, and visual analytics has been identified as a powerful means to aid humans in this mission. Visual analytics appeals to the intricate pattern recognition faculties of the human visual system which can recognize relationships with ease when presented in a suitable visual manifestation [4]. One such paradigm, especially useful for the visualization of high-D data relationships on a 2D canvas amenable to human perception is *multi-dimensional scaling (MDS)* [15][24]. MDS seeks to visually group data objects so that similar objects are close to each other and dissimilar data objects are far away, as judged by some *similarity metric*. As such, MDS provides a good visual overview on the data.

However, when using these types of overview displays it is important to realize that relationships portrayed with MDS (or any other low-D embedding technique) are still only approximations. There are numerous ways to embed high-D data into 2D, and unless the high-D space is trivial, there are always data relationships that are being suppressed. While the protocol used to optimize the embedding certainly plays a significant role here, the similarity metric used to gauge the distance relationships plays another important part.

By far the most popular metric to guide 2D MDS (and other) layouts for the visualization of high-D data is the Euclidean distance. However, once the number of dimensions grows, the contribution of each coordinate to the Euclidian distance rapidly decreases and ultimately all high-D data points have similar distances from one another [2]. As a consequence, a low-D embedding computed from these distances is not overly robust to small distance perturbations and this and other peculiar phenomena associated with high-D space are commonly referred to as the *curse of dimensionality* [2]. In fact, it is already at relatively low dimensionality, say 10, that the use of the Euclidean distance as a means to gauge the spatial proximity of two distant points becomes questionable [3].

MDS is well suited to show proximity relationships in the data, however any quantitative information on the data points is lost. Hence, MDS is often used in conjunction with parallel coordinate (PC) plots [13] by which analysts can inspect the data at an attribute level. A PC plot is generated by erecting a set of parallel coordinate axes – one per attribute. Each data point then gives rise to a piecewise linear line called *polyline* which is defined by connecting the corresponding attribute values on these parallel axes. We shall call the path of such a polyline its *signature* or *structure*. By looking at these plots, users visually aggregate the data by the patterns the polylines exhibit across the dimension axes. The usefulness of parallel coordinates for practical applications executed by mainstream users has been demonstrated by Siirtola et al. [21].

- *Jenny Hyunjung Lee and Klaus Mueller are with the Visual Analytics and Imaging Laboratory, Center for Visual Computing, Computer Science Department, Stony Brook University, Stony Brook, NY. Email: {hyunjlee, mueller}@cs.sunysb.edu.*
- *Kevin T. McDonnell is with the Department of Mathematics and Computer Science, Dowling College, Oakdale, NY, Email: mcdonnek@dowling.edu*
- *Dan Imre is with Imre Consulting, Email: dimre2b@gmail.com*
- *Alla Zelenyuk is with the Chemical and Material Sciences Division, Pacific Northwest National Lab, Richland, WA. Email: alla.zelenyuk@pnnl.gov*

Typically there is no explicit support for analysts to assess the distances of these polyline patterns – they simply use their own human perceptual system to make such associations. We therefore ask – can we capture these perceptual processes into a distance metric that can then be used to drive the linked MDS layout? This in turn would be vastly beneficial because having the same underlying distance metric would make these two displays more mutually consistent and complementary. We propose such a metric in this paper.

In the following, Section 2 presents an overview and motivation of our work, Section 3 offers a discussion of related efforts, Section 4 presents background on embedding algorithms and Section 5 describes the theory of our metric. Section 6 presents the datasets and Section 7 describes our framework and compares it with other approaches. Section 8 provides a few case studies using our system, Section 9 presents a discussion, and Section 10 ends with conclusions and an outlook onto future work.

2 Overview and Motivation

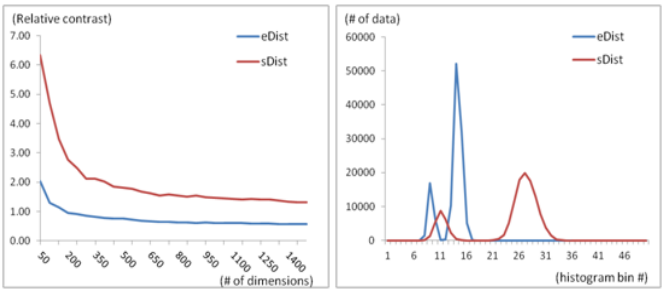
We seek a perception-motivated metric that can compare clusters in terms of the patterns they exhibit across their attribute (dimension) levels. This metric would capture differences in the structure of the dimension signature of two data points. In other words, we would regard two data points dissimilar if their dimension signatures had low correlation and different variances and means. An excellent metric that can be adapted to gauge this type of similarity is the *Structural Similarity Index (SSIM)* [25]. The SSIM is a perceptual metric popular for measuring the quality of compressed video and images, compared to some reference medium. We formally introduce our perceptual similarity metric, termed *sDist*, in Section 5.

We first ask – does our perceptual distance metric have good potential to yield a better distance measure than the Euclidean distance for high-D space? To determine this we can make use of the concept of *relative contrast* [1]:

$$\lim_{m\to\infty} \frac{dist_{\max} - dist_{\min}}{dist_{\min}} \to 0 \qquad (1)$$

where $dist_{max}$ and $dist_{min}$ are the minimum and maximum distances, respectively, in a given high-D data distribution, and m is the number of dimensions. So essentially, as m increases, the distances between pairs of data points become increasingly indistinguishable and this adversely affects the MDS layout. While this is a property of any distance metric, some will do better than others.

As a first experiment on gauging the effectiveness of our pattern-similarity distance (*sDist*) in comparison with the conventional Euclidean distance (*eDist*), we created a Gaussian-distributed dataset with 1,000 points and a varying number of intrinsic dimensions. For each metric, we computed the distances for all point pairs, determined $dist_{min}$ and $dist_{max}$ and normalized these differences by $dist_{min.}$ The results are plotted in Fig. 1a. We observe that *sDist* has a consistently higher relative contrast than *eDist*, for all dimensionalities. While this does not overcome the curse of dimensionality, it does produce a better distribution of point distances for the MDS layout.



(a) Non-clustered data (b) Clustered data

Fig 1: Comparing *eDist* (Euclidian distance) to *sDist* (pattern similarity distance). (a) Relative contrast as a function of dimensionality with un-clustered data – a single Gaussian distribution with 1,000 points. We observe that *sDist* has a higher value dynamic range than *eDist* throughout. This means that *sDist* offers a wider spread of distinct distances. (b) Distance distribution of clustered data – a mixture model of 5 Gaussians distributions with 200 points in 200-D space. The first peak is due to intra-cluster distances, while the second peak is due to inter-cluster distances. We observe that *sDist* has a larger separation between the two peaks. This means that *sDist* can distinguish clusters better than *eDist.*

Next we created an artificial Gaussian mixture dataset in 200-D with 5 clusters and 200 data points each. For each metric, we computed the histogram of distances for each point pair, normalized them by the maximum distance of each metric and plotted the curves shown in Fig. 1b. Both metrics exhibit a bimodal distribution of distances. The first mode is due to intra-cluster distances, while the second mode is due to inter-cluster distances. It is interesting to see that the two modes for our pattern-similarity metric are significantly more separated than for the Euclidean metric. This separation is sufficiently large and cannot be explained by any additional scaling effects – we already normalized the distances.

This example demonstrates that there are typically two modes in the distribution of point-pair distances: one for mutually close points located inside a cluster and one for far-spaced points that are part of different clusters. Clustering algorithms like *k*-means can identify points that are close even in high-D spaces, and in many cases classification tags already exist that can semantically group data points. So, while the overall density of points in the vast ocean of high-D space is low, there are dense islands of points (the clusters) that are mutually relatively close – close enough to gauge nearest-neighbor relations and proximity via the Euclidean distance. Using this distance at that scope is also the most appropriate one because it most accurately defines the small-scale and nuanced deviations among neighboring points. This recognition gives rise to what we call a *bi-scale metric* – it uses the Euclidean distance at the local (intra-cluster) scale and pattern-similarity at the global (inter-cluster) scale.

3 Related Work

A prominent method for high-D data visualization is via an $m \times m$ matrix of 2D scatterplots [11], but since multivar-

iate relationships are distributed across the matrix of plots, they can be difficult to discern. To overcome this problem, Nam and Mueller [17] devised an interactive user interface and framework that allows users to control multivariate dynamic scatterplots. Alternative to these direct projection methods, 2D embedding techniques "flatten" the high-D space to render the points optimally on a 2D canvas. Apart from the aforementioned MDS (used e.g. by Yang et al. [28]), also popular are methods bases on Linear Discriminant Analysis (LDA). An interesting recent work in the latter area is the LDA-based framework by Choo et al. [7][8] which decomposes the process into two stages. The first stage uses LDA to maximize the distance of cluster centroids. It achieves this well-spaced layout in part by shaping the clusters themselves compactly: in the limit, to a single point. Since LDA can only reduce the number of dimensions to $k - 1$ (k is the number of clusters, and typically $k - 1$ is greater than 2), the second stage gives users visual tools that allow them to explore and select two dimensions to be used for the final 2D scatterplot layout. Oesterling et al. [19] chose a similar two-stage framework as Choo et al. They also first use an LDA-based strategy to create an intermediate representation, but then generate a topology-based layout in the second stage. The topological features are determined by user-guided density filtering. We, on the other hand, aimed for an automated approach. Finally, both LDA-based approaches focus more on discrimination than on preserving distance relationships. Therefore, if there are many clusters, their data layouts might look uniformly distributed, suppressing true distance relationships.

There are also geodesic and kernelized distance metrics as used in popular algorithms such as Isomap [23], Locally Linear Embedding (LLE) [20], Diffusion Map **Error! Reference source not found.** and the random walk version of t-SNE [14]. All of these employ neighborhood graphs and measure the distance of two points as a geodesic path across this graph. Conversely, our metric measures the distances directly, without involving any other points.

Finally, as mentioned, the method of Parallel Coordinates [13] reduces a high-D data point to a piecewise linear curve. While the emerging ensemble of lines can reveal data patterns, it often occurs that a pattern of interest is fully or partially occluded by other data patterns. Interaction can help to isolate a desired pattern and so reduce the clutter, or one might hide unnecessary detail via analysis-informed illustrative abstraction [18].

4 Embedding Algorithms

In the following we provide more detail on MDS and LDA, which are referred to in later sections of this paper.

4.1 Multi-Dimensional Scaling (MDS)

The essence of MDS is to embed the set of high-D data points into low-D space – mostly 2D in visualization applications. We are given a set of n points $X = (x_1, x_2, \ldots, x_n)$ in m-dimensional space and compute from them an $n \times n$ distance (or similarity, adjacency) matrix with high-D distances $\delta_{ij} = \|x_i - x_j\|$. We then seek to reduce this matrix to an $n \times n$ distance matrix with 2D distances $d_{ij} = \|y_i - y_j\|$, where $y_{i,j}$, $0 \le i,j < n$, are the locations of the corresponding points on the 2D canvas. This comes down to the following optimization problem:

$$\min(y_1,..,y_n)\sum_{i<j}\left(\|y_i - y_j\| - \delta_{ij}\right)^2 \qquad (2)$$

Two main approaches exist to find this minimum. The first was devised by Torgerson [24] and is typically referred to as *classical MDS*. It seeks to obtain the embedding by fitting inner products, using Singular Value Decomposition (SVD) of XX^T. The other variant is based on the initial work of Kruskal [15] known as *distance scaling MDS*. It obtains the embedding by non-linear optimization, which is often achieved using a spring-model approach. For the classical method, since X is typically rather large, there are two popular algorithms, called *Landmark MDS* [22] and *Pivot MDS* [5] that only optimize for a representative subset of the points and then place the remaining points with respect to these locations. Several algorithms have sought to improve on both of these principal MDS approaches. However, their focus was primarily on reducing local minima for more accurate embeddings achieved at higher speed, but still using the Euclidean distance as a distance metric – in fact, the Euclidean distance has been the most often used distance metric for 2D MDS in visualization research. One of these techniques is Glimmer [12], which uses a sophisticated multi-level approach for classical MDS. Glimmer was inspired by the multigrid method devised for distance scaling MDS by Bronstein et al. [6]. It is in some sense related to the user-steerable MDS approach described by Williams and Munzner [27]. We employ Glimmer's multi-level MDS strategy, but instead of using the Euclidean distance for δ_{ij}, we utilize the more robust bi-scale pattern similarity/ Euclidean distance metric that is subject of this paper.

4.2 Linear Discriminant Analysis (LDA)

LDA aims to project the data from high-D space into an optimal lower-D space by maximizing the ratio of between-cluster variance and within-cluster variance. This guarantees maximal separability of clusters. Following the notation of Choo et al. [7], we define a dimension-reducing linear transformation G^T as:

$$G^T : x \in R^{m\times 1} \rightarrow z = G^T x \in R^{l\times 1} \qquad (3)$$

With m being the dimensionality of the original data space, G^T maps an m-dimensional data vector x in R^m to a vector z in l-dimensional space R^l ($m > l$). We call this reduced dimensional space *intermediate space*, since typically $l > 2$. Let us assume we have k classified clusters i, each with N_i points and centroid $c^{(i)}$. We can then define the within-cluster scatter matrix S_w and the between-cluster data scatter matrix S_b on the clustered data points $a_j, j \in N_i$:

$$S_w = \sum_{i=1}^{k}\sum_{j\in N_i}\left(a_j - c^{(i)}\right)\left(a_j - c^{(i)}\right)^T$$
$$S_b = \sum_{i=1}^{k} n_i\left(c^{(i)} - c\right)\left(c^{(i)} - c\right)^T \qquad (4)$$

where c is the overall centroid. Following the general LDA strategy, the approach of Choo et al. then maximizes $trace(G^T S_b G)$ and minimizes $trace(G^T S_w G)$ in the reduced dimensional space. This yields the desired embedding where the k clusters are optimally spaced apart, at the expense of compressing the data points inside the clusters. The two optimizations are simultaneously satisfied and can be approximated in a single form:

$$J_{b/w}(G) = \max trace((G^T S_w G)^{-1}(G^T S_b G)) \quad (5)$$

The solution, G_{LDA}, is a matrix in which the columns are the leading generalized eigenvectors u of the generalized eigenvalue problem:

$$S_b u = \lambda S_w u \quad (6)$$

Lastly, the m-dimensional data vectors are projected into the l-dimensional space. This space has dimensionality $l = k - 1$ at most and so does not produce the desired 2D layout as yet. This is somewhat of a shortcoming for LDA, and there are many choices how to go from l to 2.

Choo et al. offer two strategies for this. Their first method, called *Rank-2 LDA* [7], chooses the two dimensions with the largest leading generalized eigenvalues, while their second method [8] allows users to select the two dimensions via an interactive framework that uses a parallel coordinate display with bivariate scatterplots for each axis pair. We shall refer to this second, more general method as *Selected-2 LDA*.

As mentioned, all LDA-based approaches generally focus more on cluster discrimination than preserving distance relationships. Therefore, if there are many clusters, their data layouts might look uniformly distributed, suppressing true distance relationships. In contrast, our goal is to find a perceptually-motivated similarity metric that preserves the pattern that data items (clusters) have in high-D space. Using MDS with the pattern similarity distance metric also enables a well-separated global cluster layout in 2D, but unlike the two-stage LDA approach, this layout is optimized for direct 2D embedding and does not require user interaction. In addition, our local Euclidean distance metric – the second scale in our bi-scale framework – preserves the local cluster appearance well and does not appear compacted.

5 The Structure Based Distance Metric

As mentioned, we derive our new high-D distance metric from the *Structural Similarity Index (SSIM)* [25] which has found popular use in the quality assessment of compressed video and images. The SSIM is a refinement of the image quality index (UQI) [26]. Both metrics have been designed to quantify the difference between a degraded image – for example, by a compression algorithm – and a high-quality reference image. Their effectiveness has been amply verified in large-scale user studies [26]. In the following, we first describe the SSIM and then show and demonstrate its adaption to high-D data spaces.

5.1 The Structural Similarity Index (SSIM)

The Structural Similarity Index (SSIM) evaluates three image-centric measures – luminance L, contrast C, and structure S [25]. Formally:

$$\mathrm{SSIM}(x, y) = \left[\frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}\right]^{\alpha} \cdot \left[\frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}\right]^{\beta} \cdot \left[\frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3}\right]^{\gamma} \quad (7)$$

All measures are based on the image luminance values. The x and y stand for the two images to be compared, μ_x and μ_y are the mean values, σ_x and σ_y are the standard deviations, and σ_{xy} is the covariance of the two images. The constants c_1, c_2, and c_3 are typically small and prevent numerical instabilities when the main terms are close to zero. We use the settings suggested by the authors [25], which are $c_1 = (K_1 L)^2$, $c_2 = (K_2 L)^2$, $c_3 = c_2/2$, where $K_1 = 0.01$, $K_2 = 0.03$, and L is the dynamic range of the values.

The first SSIM term is the luminance comparator with a dynamic range of [0, 1]. It judges how similar the mean luminance is, and the best value of 1 can be obtained when both means are identical. The second term compares the contrast between two images. Its dynamic range is [0, 1] and it can be 1 only when both variances are identical. Finally, the third term is the structure comparator. It measures the degree of linear correlation between two images. Its dynamic range is [-1, 1], and the best value of 1 can be obtained when they are perfectly linearly correlated. This term evaluates the structural similarity after the differences in mean and contrast have been accounted for. The three components can be exponentially scaled with different weights, with $\alpha>0$, $\beta>0$ and $\gamma>0$, according each component's importance.

In practice [25], the SSIM metric is typically computed over a sliding 11×11 window, in conjunction with a Gaussian weighting with standard deviation of 1.5 samples, and then averaged into a single descriptive number. This models the property of the human eye to focus on small local image regions at a time. One obtains:

$$SSIM_{pooled} = \frac{1}{n_w}\sum_{i=1}^{n_w} SSIM(x_i, y_i) \quad (8)$$

where n_w is the number of sliding windows.

5.2 Adapting the SSIM to High-D Data Spaces

The SSIM and UQI metrics were originally devised in response to the inadequacy of the RMS error to capture the structural distortions that give rise to the perceived difference between two images. Since the RMS error is strongly based on the Euclidean distance, it is sensible to also use this metric to overcome the problems of the Euclidean distance in gauging the (dis)similarity of two high-D points. To the best of our knowledge, the SSIM/ UQI has never been used outside the image processing domain.

To make the analogy from the image domain to high-D data spaces let us recall our introductory discussion on the method of parallel coordinates. As mentioned, PC is a popular means to assess all (or a selected number of) attribute values of a high-D data point simultaneously in one display. Now, just like a gray-level image consists of pixel intensity patterns that span the spatial domain, a high-D data point visualized in PC consists of the pattern a polyline generates as it undulates across the parallel axes. We argue that the visual qualities human analysts

assess when comparing two polylines are quite related to those human observers employ when judging the difference of two images. Just like images, polylines have means and variances, and two polylines have a certain degree of correlation. So we can simply exchange the SSIM image-domain term of luminance with the more general term 'mean', but keep the terms 'contrast' and 'structure' because they are perceptually meaningful also for the high-D domain. Each vector of a high-D data point then plays the role of 'image' and its components (attributes) map to 'pixels'.

The correlation term σ_{xy} deserves special attention. In some scenarios, a perfectly negative correlation of -1 might be considered very similar to a perfectly positive correlation of 1 – after all one might just mirror the values of one of the two dimensions about the mean. This would then make data points with correlation close to 0 least similar. Conversely, in other scenarios a negative correlation may be considered rather dissimilar. We therefore provide two options – users may select either $|\sigma_{xy}|$ or $(\sigma_{xy}+1)/2$ to compute the correlation of two high-D points. In the paper, this latter setting will be used. Both will result in values within [0, 1].

5.2.1 SSIM Windowing: The Pooled Metric

Just like images, polylines span the spatial domain and are perceptually evaluated by focusing onto one local window at a time, however brief. This then motivates a pooled SSIM-like metric as formulated in (8). In experiments we found that a window size of 11 (dimensions, attributes) worked quite well and produced layouts with desirable qualities (more on this below). Thus given these strong analogies, an MDS layout driven by this metric is poised to arrange the high-D points on a 2D canvas in a manner quite similar to how they are perceived as polylines within a corresponding linked PC display.

The outcome of the windowed SSIM is affected by the order of the dimensions, and so we require a consistent and practice-informed strategy to determine the dimension order in this case. A useful measure for arranging the dimensions in a parallel coordinate plot is to ensure that neighboring dimensions are well correlated [1]. To achieve this, we have applied the approach recently proposed by Zhang et al. [30]. It uses an approximate traveling salesman scheme (TSP) via a genetic algorithm [16] to optimize the sum of pairwise correlations in a parallel coordinate plot (all selected dimensions must appear once and only once). Then by arranging highly correlated dimensions into close neighborhoods, the windowed SSIM will factor them together. We have chosen this approach over one that optimizes the SSIM itself – which would also have taken mean and variance into account – because these are dimension orderings that are often used in practice.

5.3 Using the Structure-Based Metric: A First Study

We shall now study the new metric more closely and also specifically examine the influence of the three SSIM terms. As mentioned, we can use the factors α, β and γ to weigh the influence of these terms to the overall metric outcome. In our study, we set them to either 1 or 0, which is equivalent to either keeping the corresponding component in the SSIM expression or not. Thus, there are 8 cases, which we capture in the two tables of Fig. 2a and b. Below these tables (Fig 2c) we show the corresponding parallel coordinate plots for case 1-8, each with the polylines of two data points. To the right of these plots are the corresponding MDS layouts generated using our SSIM-based metric to evaluate the distance between the two data points – hence we call it sMDS. There are four sMDS columns. The first three columns show the sMDS layouts using only one of the three individual SSIM terms – mean *M*, contrast (variance) *C*, or structure (correlation) *S*. The last column shows all of them with equal weighting – the full SSIM adapted to high-D data. All cases use $(\sigma_{xy}+1)/2$ to compute the correlation term.

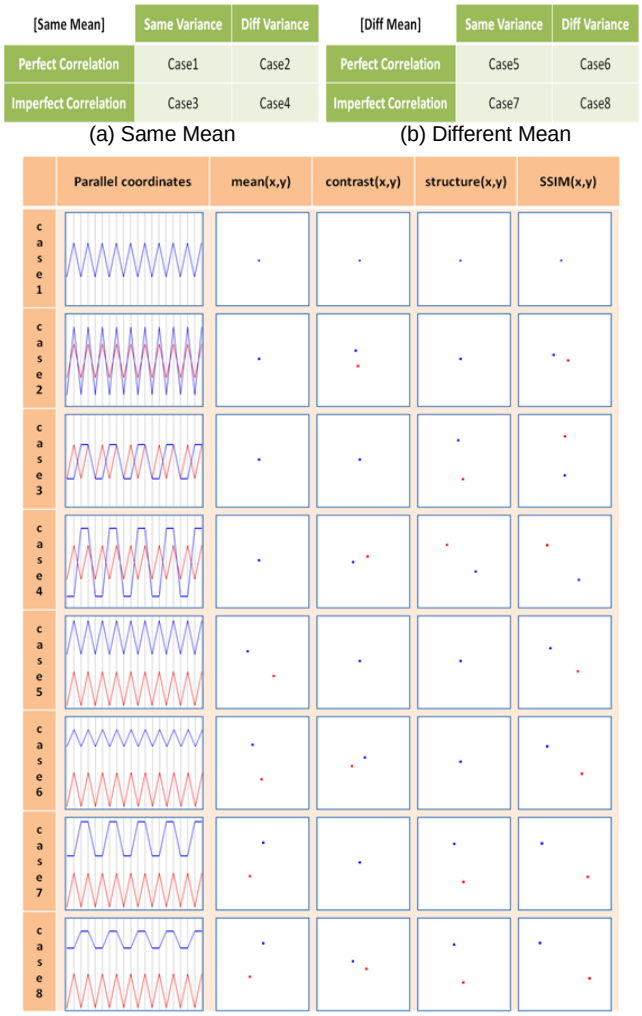
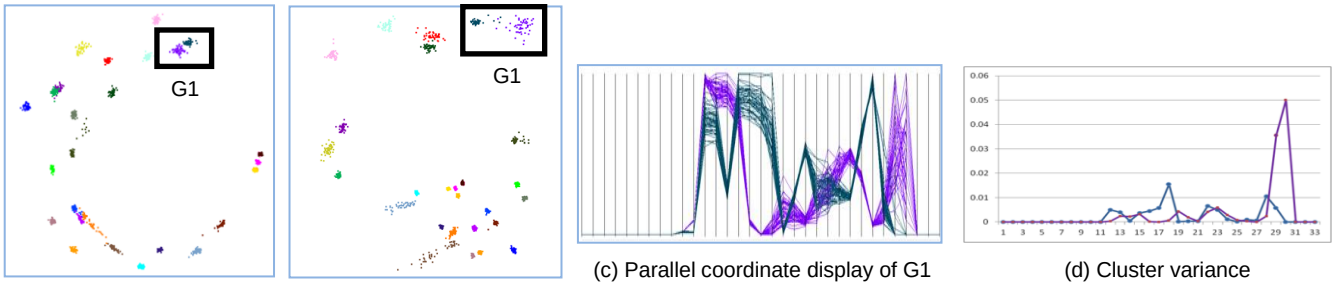


Fig 2: Exploration of the SSIM index. SSIM(x,y) consists of three terms – mean(x,y), contrast(x,y) and structure (x,y). (a) Four experimental cases with the same means. (b) Four experimental cases with different means. (c) Experiments for these eight cases. Each row shows one experimental case with a parallel coordinates display (first column), sMDS considering one of three terms at a time (columns 2-4), and sMDS with all three terms together (last column).



(a) Without windowing (b) With windowing (c) Parallel coordinate display of G1 (d) Cluster variance per dimension of G1

Fig 3: Exploration of the window approach with the Operating Systems dataset using sMDS for layout. (a) Without windowing – both clusters inside group G1 have a somewhat compact distribution, (b) with windowing – the purple cluster inside group G1 has a wider distribution than in (a); (c) parallel coordinates display of G1 and G2, (d) cluster variance per dimension for each of the two clusters in G1. The purple cluster has a rather dominant local variance profile around dimension 30 and this causes the larger spread in (b). On the other hand, these local effects are somewhat averaged out when factored in globally and therefore they play a lesser role for (a).

In Fig. 2a we keep the means identical which corresponds to the first four rows of Fig. 2c. The first case, Case 1, where the variances are identical as well and the correlation is 1, results in two overlapping points in all four displays. Case 2 also has a correlation of 1 but now the variances are moderately different. Correspondingly, the *C* term has the two data points moderately disparate, while they overlap for *M* and *S*. Conversely, the two polylines in Case 3 have the same variances but they are phase shifted and therefore less correlated. As a result, the data points are more disparate in the *S* layout, while they overlap for *M* and *C*. Finally, Case 4 also has a phase-shifted constellation but now the variances are moderately different as well. The sMDS layouts capture this correctly – both the *C* and *S* terms have the two points moderately disparate, while they overlap for *M*.

Next, In Fig. 2b we now also allow the means to change and we again consider the four most interesting cases in the lower four rows of Fig. 2c. In Case 5 the two polylines have identical variances and a correlation of 1. The sMDS layouts reflect this correctly – the two points are moderately disparate only in the *M* layout, while they overlap for *C* and *S*. In Case 6 the polylines have a correlation of 1 but also moderately different variances. Correspondingly, the two points are moderately disparate in the *M* and *C* layouts, while they overlap for *S*. Conversely, in Case 7 the two polylines have identical variances but now they have a different frequency. The sMDS layouts capture this – both *M* and *S* layouts have the two points disparate, while they overlap for *C*. Finally, Case 8 has all three SSIM terms dissimilar, which is properly reflected in the plots.

For all cases, each full SSIM plot reflects the combinations of the three individual terms appropriately. In particular, we observe that the points are further apart than in the individual layouts whenever there are two or three SSIM terms that are dissimilar at the same time.

We end by noting that all three SSIM terms are needed to fully appreciate cluster similarity. The correlation term is independent of cluster mean and extent, so the two clusters which have similar correlations with different means may still reside at very similar locations in high-D space. This may be interesting in some data analysis tasks, for example, one seeks to compare the behavior of different populations with possibly different sensitivities. If this is desired, one might just set $\alpha=\beta=0$ and only set $\gamma=1$. It is the beauty of the SSIM metric that it enables such comparisons with a simple change of weights. However, in the following experiments of this paper, we have set $\alpha=\beta=\gamma=1$ which requires similar data points with similar values (i.e. mean and variance) to be in similar high-D locations.

5.4 Effect of Windowing

To confirm the necessity of a sliding window to pool contributions we conducted an experiment using a dataset with 33 dimensions, 28 clusters and 50 data points each (this is the Operating Systems dataset described more closely in Section 6). Fig. 3a and b show the sMDS layout for region G1 with and without windowing, respectively. While the purple cluster already has a somewhat larger footprint without windowing (Fig. 3a), it is significantly more spread out when windowing is applied (Fig. 3b). And indeed, when looking at the parallel coordinate plot in Fig. 3c the purple cluster does seem to have more local variations compared to the navy cluster. Fig. 3d makes this clearer. Here we plot the cluster variance per dimension for each of the two clusters. We observe that the navy cluster has a higher level of variation around dimension 17, but this variation is dwarfed by the very large variation of the purple cluster around dimension 30 (Fig. 3d). At all other locations the variations are about the same. This explains the wider spread of the purple cluster in G1 of Fig. 3b. The spread is not as large in the non-windowed display of Fig. 3a because these effects are somewhat averaged out when computing the SSIM terms across the entire dimension spectrum.

We add that the windowing does not always produce a better spread in the distribution. Also, users may prefer to focus on global and not local similarity. Hence our system allows users to disable the windowing via a button. All of the results presented here use an 11-point window.

6 Datasets

As mentioned, we see our framework as a platform to visualize datasets that are either the output of some clustering algorithm or have been classified or generated by

some other means. In this paper we have used: (1) various artificial datasets generated by high-D Gaussian mixture modeling, (2) the Concrete Compressive Strength dataset from the UCI database (1,030 data points, 9 dimensions) [31], (3) a dataset with mass spectra of aerosol particles acquired by a state-of-the-art SPLAT (Single Particle Laser Ablation Time-of-Flight) mass spectrometer [29] divided into 4 clusters, each for a specific particle type – ANLA, NaCl, NaNO3 etc. (2,000 data points, 450 dimensions), (4) the Waveform Database Generator dataset from UCI (5000 points, 22 dimensions) [32], and (5) a dataset obtained for file system analysis, called the Operating Systems (OS) dataset. This dataset has 1,400 data points divided into 28 clusters, each for a specific file system operation – ALLOCATE, DELETE, RELEASE, etc. Each data point characterizes a system operation as a 33-D vector which is essentially a binned histogram of completion times. By collecting many observations over time, for different benchmarks and execution profiles, each file system operation has a characteristic histogram which can yield insight into the behavior of a particular file system, but also allows for the comparison of different file systems. Much of the research reported in this paper has been motivated from the interaction with the file system researchers – specifically the distance discrepancies that are inherent to conventional MDS layouts.

While the initial order of dimensions in the Concrete Compressive Strength dataset is arbitrary, the other datasets have a meaningful initial order. Therefore, the dimension ordering method is applied only to the Concrete Compressive Strength dataset.

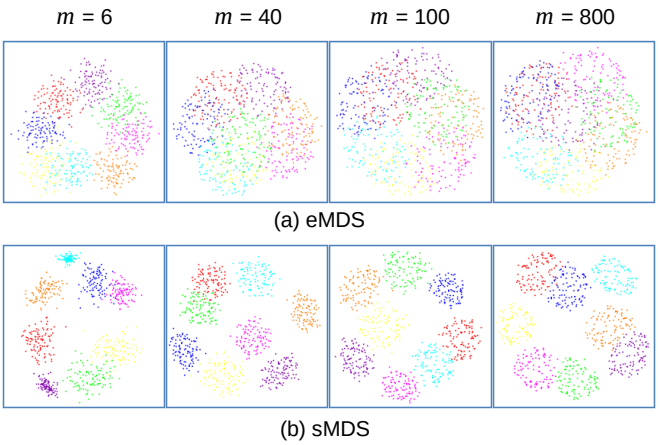


Fig 4: 2D scatter plots obtained by applying MDS with (a) the Euclidean metric (eMDS) and (b) the SSIM-based metric (sMDS) to plot a synthetic Gaussian mixture dataset with 8 clusters and 800 data points (100 points for each cluster), for a variety of dimensionalities (*m*). A different color corresponds to a different cluster. When the number of dimensions is 6, the clusters do not overlap significantly for either metric – however eMDS shows a few overlaps. For larger numbers of dimensions, eMDS leads to severe cluster overplotting while sMDS preserves and distinguishes the individual clusters well.

7 MDS With Structural Distance (sMDS)

Section 5.3 demonstrated the promise of sMDS via a small low-D toy example. We shall now examine its effectiveness using a larger synthetic dataset. Specifically, we compare two versions of MDS: (1) the conventional MDS (eMDS) using the Euclidean distance metric *eDist*, and (2) our sMDS using the new structural distance metric *sDist*. For this purpose we generated four Gaussian mixture datasets with *m*=6, 40, 100, and 800 dimensions, each with 800 data points divided equally into eight non-overlapping clusters. Each cluster was generated at random with identical variance, which yields clusters of similar distributions but different structure. Fig. 4 shows the corresponding 2D scatter plots obtained with eMDS (Fig. 4a) and our sMDS (Fig. 4b). We observe that for *m*=6 both eMDS and sMDS visualize the six clusters well, but that eMDS fails to separate the individual clusters once *m* increases, mapping points of different clusters into overlapping areas. But even for *m*=6 (first column), eMDS cannot completely separate the clusters – points from different clusters are intermixed and the structures of the individual clusters are lost. Conversely, sMDS has none of these problems and separates individual clusters even for *m*=800.

This shortcoming of eMDS could either be due to distortion or because the Euclidean metric cannot gauge the distance between points correctly. The first issue is unlikely for *m*=6 since the stress values are low enough to be ignored. For the second possible reason we recall Fig. 1b, where we distinguished between two distances: (1) the min (intra-cluster) distance and (2) the max (inter-cluster) distance. The most significant condition for non-overlapping clusters is that the distance between a pair of points that belong to the same cluster is clearly distinct from that of two points with different cluster memberships. However, the overlapping clusters for eMDS indicate that it is unable to distinguish the two distances in high-D. Hence, the distribution of pair-wise distances is uniform or near-uniform -- a hallmark of the curse of dimensionality. Conversely, the sDist distance metric of sMDS preserves the ability to differentiate between intra and inter-cluster points consistently even for high dimensionality. This indicates (again) that the sDist has a less uniform distribution of pair-wise distances and so alleviates the curse of dimensionality to some extent.

7.1 Comparing sMDS with Rank-2 LDA

As noted in Section 4.2, previous work has suggested the use of Rank-2 LDA to overcome the problems with overlapping clusters. While this has been shown to separate the clusters very well – with the shortcoming of having to first select two major dimensions for projection – the issue of correct placement of the clusters has not been discussed thus far. We shall examine this now, using the OS-dataset as an example. Fig. 5a shows a Rank-2 LDA projection, Fig. 5b zooms into the rectangular region marked C1, and Fig. 5c provides the associated parallel coordinates plot. Let us focus on the two clusters colored green and cyan and labeled *cc1* and *cc2* in the marked rectangular region in Fig. 5b. For ease of comparison, we have isolated the parallel coordinate plots for these two clusters in

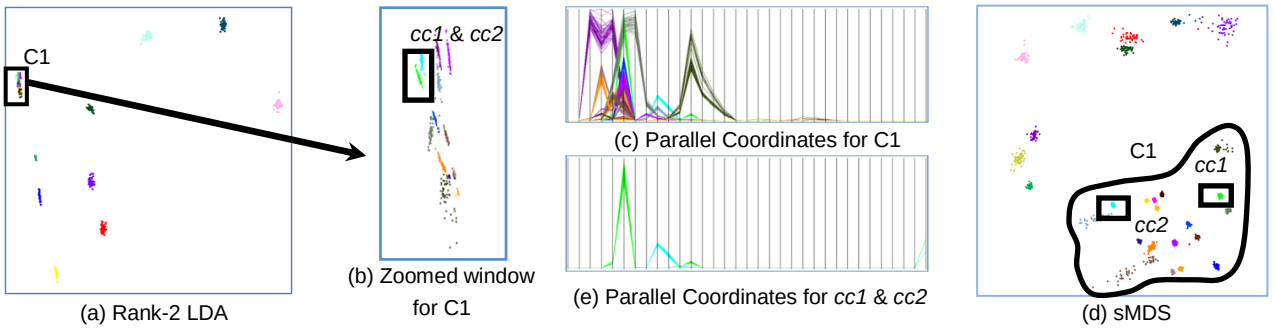


Fig 5: Comparing Rank-2 LDA and sMDS using the OS-dataset. (a) Rank-2 LDA projection – the selected region C1 includes 19 clusters. (b) Zooming into region C1 – the selected region inside the black box contains two clusters, *cc1* (green) and *cc2* (cyan). (c) Parallel coordinates display of C1. (d) sMDS layout – the two clusters *cc1* and *cc2* map to distant locations, but all clusters of C1 are contained exclusively in the outlined region. (e) Parallel coordinates display of *cc1* and *cc2* – the two clusters have very different distributions along the dimensions, but they are close in (a) and (b), but not in (d), confirming that sMDS can gauge the similarity better than Rank-2 LDA.

Fig. 5e. Clearly, these two clusters are not overly similar, yet they come to rest very closely in the Rank-2 LDA plot. Conversely, as we can observe in Fig. 5d, the sMDS locates the two clusters appropriately far apart. On the other hand, a second observation we make is that the outlined region in Fig. 5d contains no cluster that is outside the rectangular region marked in Fig. 5a but contains all clusters within it. This means that while Rank-2 LDA cannot resolve small-scale distances (which sMDS can), it is able to distinguish large-scale neighborhoods quite well.

7.2 Comparing sMDS with MDS-LDA

One might ask if Rank-2 LDA's inabilities of resolving small-scale distances are rooted in the fact that only the major two dimensions are used to perform the 2D projection. Clearly there is some degree of variation that is lost in this dimension-culling process. As a possible solution we replaced the selection of two projections by a non-linear MDS layout optimization that uses all or a representative subset of the k-1 dimensions that the LDA identifies. We call this approach *MDS-LDA* and it uses the Euclidean distance metric for layout to be in keeping with the other steps of the LDA algorithm.

We again employ the OS-dataset to demonstrate this approach and compare it with our sMDS. The OS-dataset has k=28 clusters, but we empirically found that the 12 dimensions with the highest generalized LDA eigenvalues achieved the best separability between the clusters. Figs. 6a and b show the sMDS and the MDS-LDA plot, respectively. When examining the sMDS layout (Fig. 6a), we readily notice in the bottom right corner an archipelago of clusters that is clearly separated from a crescent of clusters that extends across the top left corner. We subsequently outline and label this archipelago C2. We then outline the same clusters also in the MDS-LDA layout (Fig. 6b) where these two distinct constellations cannot be recognized at all. To gather more insight into this discrepancy, we visualize both the C2 and the non-C2 points in two parallel coordinate plots (Figs. 6c, d). We see that the C2 clusters (Fig. 6c) have their peaks mainly in the first third (and half) of the dimension spectrum, while the non-C2 clusters (Fig. 6d) have their peaks mainly in the remaining two thirds. Three clusters stick out as being part of the overlap region of these two dimension spectra – the olive cluster in C2 and the faint-cyan and salmon clusters in non-C2 (see arrows in Fig. 6a). The olive cluster has two peaks – at the 6th and at the 12th dimensions – and indeed it is at the archipelago's edge close to the crescent. On the other hand, the faint-cyan cluster has a peak at the 6th but also at the 15th and 28th dimension and rightfully so it is at the extreme of the crescent closest to C2.

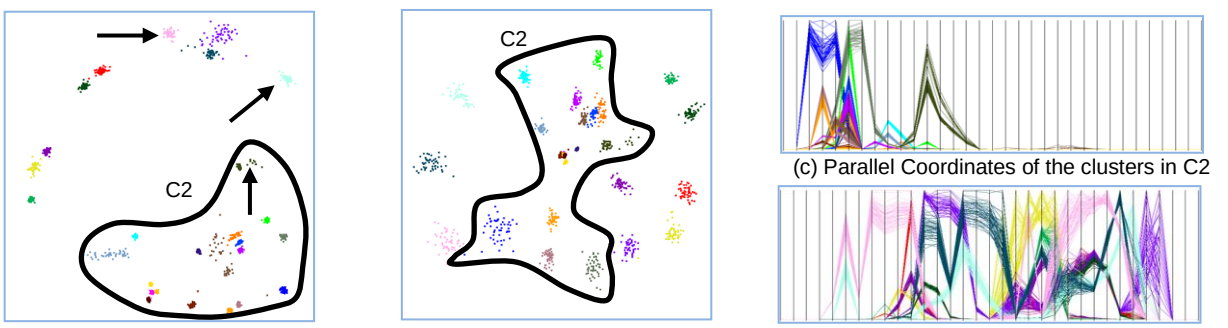
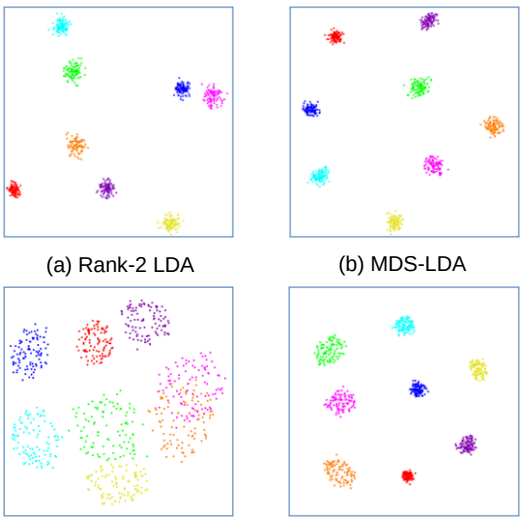


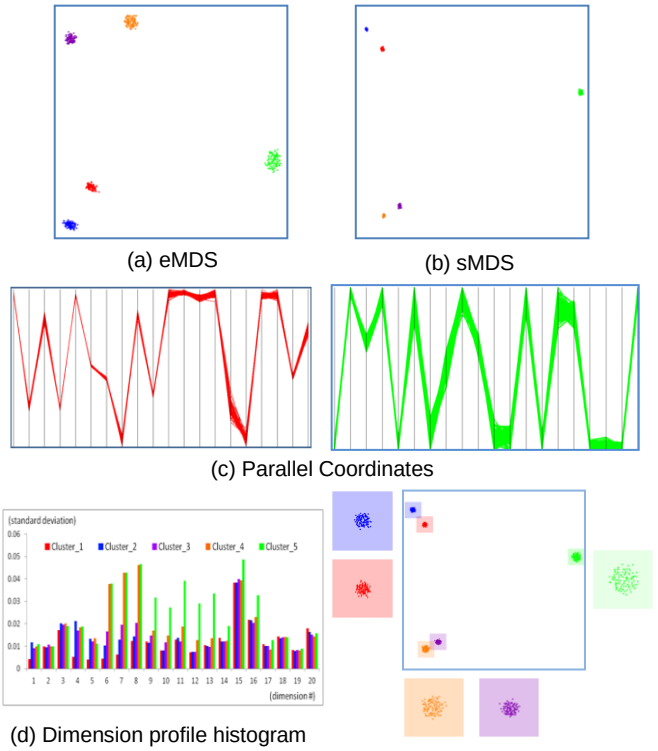
Fig 6: Comparing MDS-LDA with sMDS using the OS-dataset. (a) 2D layout obtained by sMDS – we outlined a set of clusters, C2, which are close to each other but are distant from the remaining clusters (the arrows point at some interesting clusters – see text); (b) 2D layout obtained with MDS-LDA using the first 12 dimensions from the LDA result – the outline contains the clusters of C2 previously marked in (a).There is no clear separation between the C2 clusters and those not in C2. (c) Parallel coordinates display of C2 and (d) parallel coordinates display of the clusters not in C2 – the difference between the clusters in C2 and the clusters not in C2 is much larger than the difference of clusters within C2 which confirms the superiority of the sMDS layout.



(a) Rank-2 LDA (b) MDS-LDA

(c) eMDS (d) sMDS

Fig 7: Visualizing cluster distribution using a 100-D synthetic Gaussian mixture model dataset (800 points) consisting of 8 equal-sized clusters with a wide variety of distributions (appearances): red < blue < purple < yellow < cyan < pink < orange < green. (a) and (b) 2D layouts generated by Rank-2 LDA and MDS-LDA, respectively – every cluster has a very similar type of distribution regardless of their size in high-D. (c) eMDS layout – it preserves the distribution appearance but suffers from overplotting, (d) sMDS layout – it can be appreciated that the cluster distribution in the layout respects the corresponding distribution in high-D but without incurring any overplotting.



(a) eMDS (b) sMDS

(c) Parallel Coordinates

(d) Dimension profile histogram (e) Bi-scale visualization

Fig 8: Visualizing cluster distribution using a synthetic Gaussian mixture model dataset consisting of five 20-D 100-point clusters with a variety of non-uniform extents determined by the sampling standard deviation: red < blue < purple < orange < green. (a) eMDS layout – each cluster has a distribution largely reflecting its distribution in 20-D. (b) sMDS layout – all clusters have rather similar distributions. (c) parallel coordinates display of the red and green clusters – the green cluster has a slightly wider distribution. (d) dimension profile histogram, one for each cluster – the higher the bar, the greater the standard deviation for that dimension (e) bi-scale visualization – the tiles are laid out via sMDS, but the distributions within the tiles are laid out via eMDS.

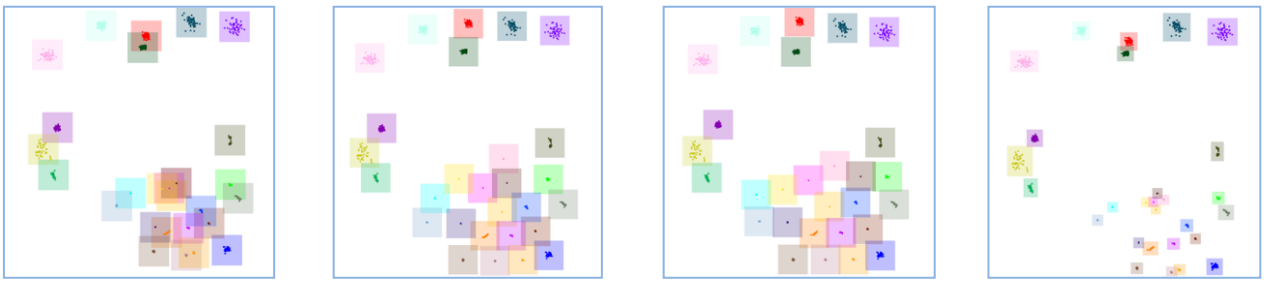
The salmon cluster, on the other hand, is far from C2, but this is justified by its many peaks in the upper two thirds of the spectrum.

Given this apparent dissimilarity of the C2 and the non-C2 populations, embedding the C2 region into the non-C2 region as is done by the MDS-LDA does not seem overly accurate. The sMDS, on the other hand, spaces these two populations appropriately far apart which further confirms the promise of this approach. So we conclude that while both methods – MDS-LDA and sMDS – achieve a good separation of the clusters, only the latter also maintains their mutual distances.

7.3 Preserving Cluster Distribution

An important quality to maintain is cluster distribution (its appearance), that is, a cluster with a wider spread of points in high-D space should also have a wider spread in the corresponding 2D layout. To explore this property we generated a Gaussian mixture dataset consisting of eight Gaussians (100 points each) with a wide variety of distributions, expressed in terms of their standard deviations σ in 100-D space. Their distribution ordering in ascending order of σ is red, blue, purple, yellow, cyan, pink, orange, and green. Fig. 7a and b explore how Rank-2 LDA and MDS-LDA perform in this regard. We observe that while the clusters are now well separated, their distributions are rather similar. This is rooted in the fundamental definition of LDA, which seeks to maximize inter-cluster distances by minimizing (or shall one say, sacrificing) intra-cluster distances, yielding fairly similar and tight distributions of points in the 2D layouts. However, eMDS can resolve the ordering, but at the price of overplotting (see Fig. 7c). The sMDS layout, on the other hand (see Fig. 7d) preserves the distribution ordering – the red cluster is smaller than the blue cluster which in turn is smaller than the purple cluster and so on – but without overplotting.

To get further insight into the ability of sMDS to preserve cluster distribution at a less extreme scale we generated yet another Gaussian mixture dataset, now consisting of five 20-D 100-point clusters with moderately different distributions, ordered in increasing distribution: red, blue, purple, orange, and green. The distribution profiles across the dimensions are visualized in Fig. 8d as dimension histograms of standard deviations for each of the five clusters. Fig. 8c shows a parallel coordinate plot for the red and green clusters, respectively. For the sMDS display (Fig. 8b), we notice that every cluster has a rather compact appearance – much more compact than would be justified from the dimension profiles. In contrast, the eMDS layout (Fig. 8a) preserves these profiles quite well. The mediocre performance of sMDS in this respect is no huge surprise since the SSIM-based distance metric only looks for statis-



(a) No overlap removal (b) p = 0.2 (DISSIM = 3.252) (c) p = 0.05 (DISSIM = 3.482) (d) p = 0.2 (DISSIM = 1.999)

Fig 9: Overlap removal using the OS dataset: *Without tile cropping* - (a) No overlap removal – initial layout, (b) 20% overlap allowed, and (c) 5% overlap allowed. *With tile cropping* - (d) 20% overlap allowed. We observe that cropping the tiles to cluster size is the best solution. If this cannot be done, then reducing the overlap just moderately can already yield acceptable displays which are also less distorted than with full overlap removal.

tical similarities and not for absolute component-wise differences. Thus, while this metric is useful to judge inter-cluster distances, it is less useful on the local, intra-cluster scale. There, the Euclidean metric seems more appropriate. This observation has motivated our final layout scheme – the bi-scale layout described next.

7.4 The Bi-Scale Layout: Fusing sMDS and eMDS

The inability of sMDS to preserve distribution motivates us to return to the Euclidean metric on the local scale (intra-cluster scale). In the bi-scale framework that results, the global scale (inter-cluster scale) is used to compare different clusters, while the local scale visualizes the distribution within each such cluster. It first lays all out data points using sMDS and then computes the midpoint of each cluster. Next, the tiles are centered on these midpoint, the individual cluster distributions are laid out via eMDS, and finally mapped onto the tiles.

However, since the distances of these two metrics are not 100% comparable, we may not want to merge the inter- and intra-cluster layouts outright. To provide a graphical distinction, we devise a two-level display that represents the clusters as an arrangement of tiles laid-out with sMDS. Making the tiles semi-transparent, similar to the value-relation display of Yang et al. [28], helps avoid problems with occlusions. An example for this layout is shown in Fig. 8e, where the distribution patterns inside the tiles now match those in the profiles and parallel coordinate plots quite well. The transparency of the tiles can be controlled by the user.

7.4.1 Controlling Tile Overlap for the Bi-scale Layout

The semi-transparent tile approach proves effective when the number of tiles involved is manageable. However, it is not sufficient in dense areas where many clusters are intermixed. In this case, the only way to cope with this problem is to reduce the overlap altogether. For this purpose we adapted an algorithm originally designed for reducing the overlap of nodes in graph drawing applications [10]. This algorithm utilizes a proximity stress model that seeks to preserve the initial layout as much as possible. To ensure smooth convergence to the solution, it iteratively adjusts the graph nodes by small increments. The quality of a layout with respect to the original layout is assessed by a dissimilarity value, DISSIM – two layouts are more dissimilar when DISSIM is greater.

To allow users to be more in control of the layout and the time it takes to achieve it, we have incorporated the algorithm into an interactive interface. The first, very basic mode allows users to stop iterations at any time, which also reduces the risk of deviating too much from the initial layout. Further, we also added support to control the amount of overlap permitted. The original scheme does not allow for any partial overlap and so often unnecessary overlap removal operations are performed. This usually occurs in empty spaces of cluster tiles. We therefore provide an interactive slider interface by which users can control how much partial overlap is allowed. It sets the *permitted overlap ratio p* which virtually scales the sizes of the tiles before passing them into the algorithm. When p>0 partial overlap will be the result, while when p=0 there will be no overlap between the tiles. Using p, the width w_i and height h_i of a tile are set to w_i*sqrt(1-p) and h_i*sqrt(1-p), respectively. We find that the resulting layouts are typically acceptable in terms of readability, but they preserve the original space relationships much better as is evident by lower DISSIM values.

Fig. 9a shows the original bi-scale layout of the OS-dataset (p=1) which is fairly cluttered in some areas. Fig. 9b and c show the layouts for p=0.2 and p=0.05, respectively. We find that p=0.2 is already quite acceptable despite some remaining overlap. Conversely, the latter has nearly no overlap but its layout is more distorted, as is evidenced by the higher DISSIM value.

Another way to remove the tile overlap is by adjusting (cropping) tile size to the cluster's bounding box. As is shown in Fig. 9d for p=0.2, this can reduce the need for overlap removal when some of the clusters are small. It typically leads to lower DISSIM values for all settings of p. Here, the layout only for p=0.2 is shown due to the space limitation. In practice, our framework uses both tile cropping and partial overlap removal.

8 More Results With Practical Datasets

We have applied our framework to visualize four practical data sets as specified in Section 6.

8.1 Concrete Compressive Strength Dataset

The concrete dataset has nine dimensions – eight quanti-

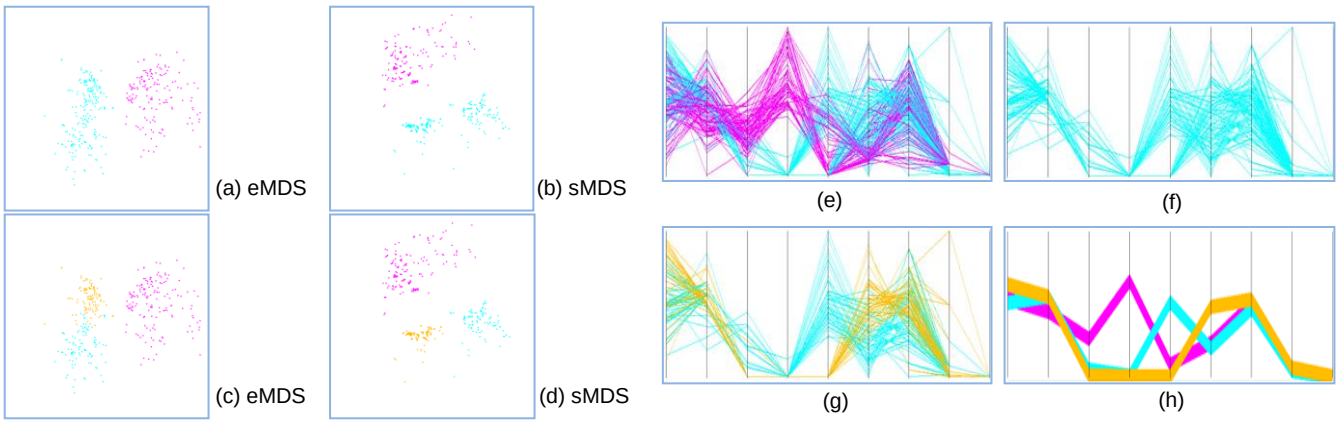


Fig 10: Concrete Compressive Strength Dataset – low strength group of data points. In the parallel coordinate display the correlation-based dimension ordering method was applied. (a) (b) eMDS and sMDS with two user-selected clusters colored in cyan and magenta – these two clusters can be easily distinguished in both plots, (c) (d) eMDS and sMDS with three user-selected clusters colored in cyan, magenta, and orange – only sMDS can distinguish the cyan and orange clusters. (e) parallel coordinate display of the cyan and magenta clusters – they give rise to clearly different patterns, (f) parallel coordinate display of the cyan cluster only – one can observe two separate patterns especially in the 5th dimension; (g) the same parallel coordinate display colored according to the two sub-clusters in (d) – the observation of (f) is confirmed. (h) abstracted parallel coordinate display only showing the centerlines of the three clusters of (d) – the similarities of their patterns reflects the cluster distances in (d).

tative input variables and one quantitative output variable (concrete compressive strength). We classified the dataset into three groups based on the output values – low, mid, high. In the following we only consider points belonging to the low strength group. After applying the correlation method to obtain a good order of dimensions, we computed the eMDS and sMDS layout (see Fig. 10a and b). There is an obvious separation of the points into two groups for both schemes – colored cyan and magenta in Fig. 10a and b. When examining these separated groups in parallel coordinates (Fig. 10e), we clearly see their different patterns, especially in the 4th dimensions. Both eMDS and sMDS preserve this difference well.

Let us now focus on the cyan sub-group in the sMDS plot of Fig. 10b where we see one further separation of points, colored orange and cyan in Fig. 10d. When we color these points also in the eMDS plot of Fig. 10c, the two groups are contained but they are not well separated. Hence, it will be difficult to recognize them in this plot. The parallel coordinate plot of Fig. 10f also indicates that there are two groups of polylines with different patterns, most pronounced in the 5th dimension. In fact, there is a strong correspondence between these groups and the ones found in Fig. 10d. This becomes readily apparent when we color the corresponding polylines in the same colors, as has been done in Fig. 10g.

Finally, Fig. 10h summarizes these findings in an abstraction parallel coordinate plot which only shows the centerlines of each of the three distributions. We observe that the structural similarity of these polylines quite closely matches the distances of the corresponding clusters in the sMDS plot of Fig. 10d.

8.2 Mass Spectra of Aerosol Particles

This case study, like the concrete data, shows that the structural similarity is preserved in the sMDS plot (see Fig. 11). In this study, we only consider points belonging to the particle type $NaNO_3$. The first row of Fig. 11a and b

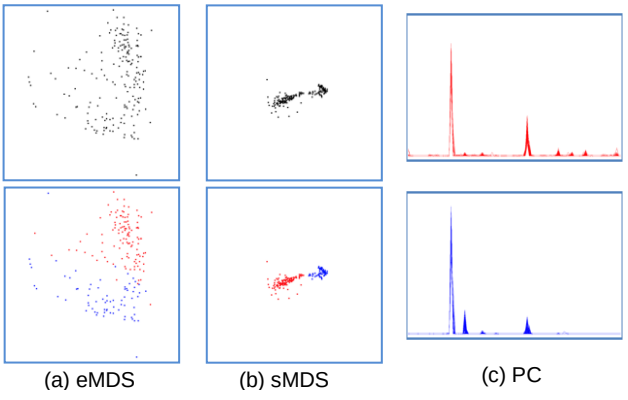


Fig 11: Mass Spectra of Aerosol Particles. (a)(b) 1st row: eMDS and sMDS with one cluster (particle type – $NaNO_3$), and 2nd row: two user-selected sub-clusters colored in red and blue – these two sub-clusters can be clearly distinguished only in sMDS, (c) parallel coordinate display of the red and blue sub-clusters – they give rise to clearly different patterns.

plots these points with eMDS and with sMDS, respectively. An obvious separation of the points into two groups is observed only in the sMDS plot – colored red and blue in the second row of Fig. 11b. The parallel coordinates plot (Fig. 11c) also indicates that the two groups have different patterns – higher first peak, lower second peak in the red group and additional peak between the two peaks in the blue group. Finally, we notice that the red, more skewed (noisier) sub-cluster in the sMDS plot is also the one that has a set of additional small peaks in the upper half of the dimension spectrum. Hence, the sMDS captures this additional variability well, while the eMDS fails to do so.

8.3 Waveform Database Generator Dataset

This dataset has 21 continuous variables and 1 class variable – each class has 33% of the points. Fig. 12a and b plot these points using eMDS and sMDS, respectively, with the three classes colored red, green and blue. We again

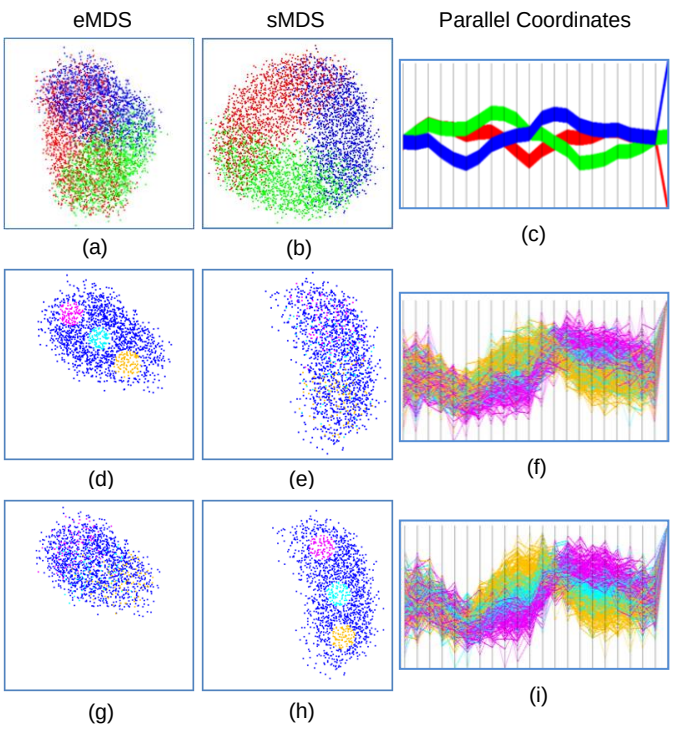


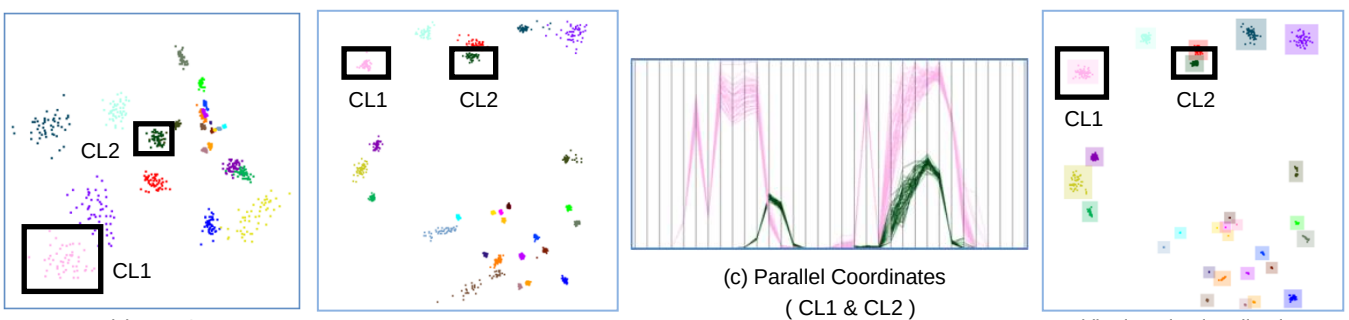
Fig 12: Waveform dataset. (a) (b) eMDS and sMDS with three clusters and (c) abstracted parallel coordinate display only showing the centerlines of the three clusters. The next plots only consider the blue cluster for two experiments, Exp1 and Exp2. Exp1: (d) eMDS plot with three filled circular regions, (e) sMDS plot with these marked points colored – the marked points of (d) appear largely at random locations, and (f) the corresponding parallel coordinate plot – there is some bundling of the marked points but also a high degree of intermixing. Exp2: (g) eMDS plot with the points marked in (h) colored – they appear largely at random locations, (h) sMDS plot with three filled circular regions marked in color, and (i) the corresponding parallel coordinate plot – there is much better bundling of the marked points than in (f). All patterns of the three circular regions are seen quite clearly in (i) while we can hardly see the pattern of the cyan group in (f).

observe that sMDS does much better than eMDS in isolating the three classes. The parallel coordinate plots of the three classes in Fig. 12c confirm this – one can clearly observe that these three classes have different patterns and so should be well separated in an MDS plot.

Interesting insight comes from an experiment in which we take the blue cluster and mark three arbitrary filled circular regions. We find these regions by marking a (center) point and locating the k nearest neighbors in 2D which results in near circular regions for both layouts – eMDS (Fig. 12d) and sMDS (Fig. 12h). We subsequently color the corresponding points in the other layout. Fig. 12e shows the marked points in Fig. 12d and Fig. 12g shows the marked points in Fig. 12h. We observe that there is no correspondence for either combination and conclude that the two layouts do not share a common mapping. But the most valuable insight is gained when examining the corresponding parallel coordinate plots. Fig. 12f colors the polylines for the three regions marked in the eMDS plot (Fig. 12d), while Fig. 12i colors the polylines for the three regions marked in the sMDS plot (Fig. 12h). We observe that for sMDS the three different polyline groups form three coherent bundles, at least for the upper 2/ 3 of the dimensions, while for the eMDS groups these polylines are more intermixed. The cyan group can be hardly seen in Fig. 12f but it is well visible in Fig. 12i. We feel that this is an impressive demonstration of the SSIM-based distance metric and sMDS overall. It essentially means that users can perform manual clustering and segmentation of high-D point clouds directly in at 2D projection display where it is most intuitive.

8.4 Operating System (OS) Dataset

This case study is illustrated in Fig. 13. We shall focus on two clusters, C1 (pink) and C2 (green), which are due to the TRUNCATE and the READPAGE operations, respectively. The eMDS plot (Fig. 13a) suggests that TRUNCATE is quite different from READPAGE, which, however, is not confirmed when looking at the corresponding parallel coordinate plot (Fig. 13c). The sMDS plot, on the other hand, puts these two operations close by, which is more appropriate. Yet, in sMDS both clusters have the same spread which is incorrect – TRUNCATE has much more diversity than READPAGE (see again Fig, 13c),



(a) eMDS (b) sMDS (c) Parallel Coordinates (CL1 & CL2) (d) Bi-scale visualization

Fig 13: OS dataset. (a) eMDS plot with two clusters marked – CL1 and CL2, (b) sMDS plot with the same clusters marked, (c) parallel coordinates display of CL1 and CL2, (d) bi-scale visualization. In (c), the parallel coordinates show that the pink cluster TRUNCATE (CL1) is much more diverse than the green cluster READPAGE (CL2), but the two clusters have a quite similar set of non-zero valued dimensions. The sMDS plot in (b) plots C1 and C2 correctly close together but has them equally compact, while eMDS in (a) plots them further apart, but maps their extents correctly. The bi-scale plot in (d) combines the best of both worlds: correct cluster location and correct cluster appearance.

which is well represented by eMDS. The bi-scale MDS plot of Fig. 13d unifies the two findings. By inspecting this plot, OS analysts can learn that while READPAGE and TRUNCATE have similar time histograms, the former is a much more stable operation.

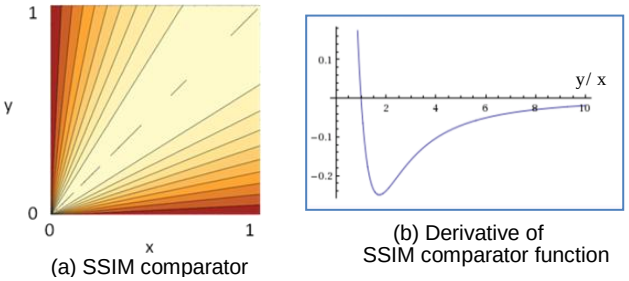
9 Discussion

The comparator function $2xy/(x^2+y^2)$ that lies at the heart of the SSIM metric has a few convenient properties which can be discerned from the plot shown in Fig. 14a. First, unlike metrics that are based on the difference $|y-x|$, as is the RMS error which underlies the Euclidean distance, its multiplicative form is naturally normalizing into an interval of [-1, 1] for any range of x and y – in practice we normalize our data into [-1, 1] or [0, 1]. Second, its sensitivity is large at low levels of x and y and also for small deviations of x and y. This is somewhat reminiscent of Weber's Law where the perception p of a stimulus s is proportional to $\ln(s/s_0)$, with s_0 being the baseline of the stimulus. The sensitivity of p is the derivative of this function, $1/s$, which decays rapidly as s deviates from s_0. Similar is true for the SSIM comparator function. To show this, let us write x and y as the ratio $r=y/x$ where, without loss of generality, we assume $y \geq x$. This yields the rationalized comparator function $2r/(1+r^2)$. We then take its derivative to arrive at the function $(2-2r^2)/(r^4+2r^2+1)$, plotted in Fig. 14b. In this plot we observe the same principal behavior than in Weber's law – small ratios get emphasized (spread apart) and large ratios get de-emphasized (compressed). This behavior is very important for visual stimuli and explains why the SSIM has been so successful in evaluating the fidelity of images.

But our focus is not images. Rather, we aim to replicate the perceptual experience users have when examining a parallel coordinate plot and transform this into the high-D distance metric used for MDS. Intuitively, when two polyline segments are close and similar, small differences will be noticed much more intensely as when they are further away and largely dissimilar. As the derivative plot of Fig. 14b shows, this impression is replicated by the SSIM comparator function. But there we also notice that very small deviations are less recognizable. This, however, is justified since these deviations could just be due to noise and should indeed be less influential.

Another feature of the comparator function has been mentioned already in the onset of this section – the fact that the comparator function is also more sensitive for low levels of x or y. This is deeply rooted in Weber's law but less motivated for the perception of parallel coordinates since here these would just be polyline segments near the zero-line. We might think of these polyline segments again as noise or as less significant – we can always shift the zero-line up or down if this is not the case.

The observations with regards to the SSIM comparator function are only relevant for the mean and variance terms – the correlation term is naturally expressed by this function. And also, while the discussion presented here may not fully substitute for a full-scale psycho-physical study, the various examples presented in this paper have clearly confirmed that the metric works exceedingly well.



(a) SSIM comparator (b) Derivative of SSIM comparator function

Fig 14: SSIM comparator function. (a) Contour plot – with value mapped to brightness. (b) derivative of the function expressed as ratio y/x.

10 Conclusions

We have described a novel high-D distance metric that leads to more accurate MDS layouts, both in terms of global and in terms of local distance relationships. Our bi-scale distance metric uses a pattern-based similarity metric inspired by vision research to assess proximity of distant clusters, but uses the conventional Euclidean distance to evaluate local distance relationships. Using this framework, we were able to create MDS layouts with point distributions that closely match the polyline data signatures observed in parallel coordinate displays. Next, we would like to incorporate multi-scale and multi-resolution analysis to compare patterns at different levels of scale, and we would like to embed our framework into interactive cluster analysis applications, such as k-means and others. We would also like to apply an adaptive-step size for controlling the overlaps between cluster tiles in order to make our bi-scale framework faster and more robust. Further, it would be interesting to confirm our currently more empirical successes with rigorous psycho-physical experiments. Finally, we would also like to study the impact of the approximate TSP solver on our layout. It might not be significant since the windowing has a smoothing effect on the small and local dimension ordering variations.

Acknowledgments

Partial support for this research was provided by NSF grants 1050477, 0959979, and 1117132. Partial support was also provided by the US Department of Energy (DOE) Office of Basic Energy Sciences, Division of Chemical Sciences, Geosciences, and Biosciences. Some of this research was performed in the Environmental Molecular Sciences Laboratory, a national scientific user facility sponsored by the DOE's OBER at Pacific Northwest National Laboratory (PNNL). PNNL is operated by the US DOE by Battelle Memorial Institute under contract No. DE-AC06-76RL0. K. Mueller was partially supported by the IT Consilience Creative Project through the Ministry of Knowledge Economy, Republic of Korea. We thank Erez Zadok for providing the OS dataset and much helpful inspiration.

REFERENCES

[1] A. Artero, M. de Olivera, H. Levkowitz, "Enhanced high-dimensional data visualization through dimension reduction and attribute arrangement," *Proc. IEEE InfoVis*, pp. 707–712, 2006.

[2] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[3] K. Bennett, U. Fayyad, D. Geiger, "Density-based indexing for approximate nearest-neighbor queries," Proc. ACM SIGKDD, pp. 233-243, 1999.

[4] J. Bertin, *Semiology of Graphics: Diagrams, Networks, Maps*, (Translation of 1967 French Original), ESRI Press, 2010.

[5] U. Brandes, C. Pich, "Eigensolver methods for progressive multidimensional scaling of large data," *Proc. Graph Drawing*, 2006.

[6] M. Bronstein, A. Bronstein, R. Kimmel, I. Yavneh, "Multigrid multidimensional scaling," *Numerical Linear Algebra with Applications (NLAA)*, 13:149–171, 2006.

[7] J. Choo, S. Bohn, H. Park. "Two-stage framework for visualization of clustered high dimensional data," *Proc. IEEE Visual Analytics Science and Technology Conference (VAST)*, pp. 67-74, 2009.

[8] J. Choo, H. Lee, J. Kim, H. Park, "iVisClassifier: An interactive visual analytics system for classification based on supervised dimension reduction," *Proc. IEEE Visual Analytics Science and Technology Conference (VAST)*, pp. 27-34, 2010.

[9] R. Coifman and S. Lafon, "Diffusion maps," Applied and Computational Harmonic Analysis, vol. 21, pp. 5-30, 2006.

[10] E. Gansner, Y. Hu. "Efficient node overlap removal using a proximity stress model," *Symposium on Graph Drawing*, pp. 206-217, 2008.

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

[12] S. Ingram, T. Munzner, M. Olano, "Glimmer: Multilevel MDS on the GPU," *IEEE Trans. Visualization and Computer Graphics*, 15(2): 249–261, 2009.

[13] A. Inselberg, B. Dimsdale, "Parallel Coordinates: A tool for visualizing multi-dimensional geometry," *Proc. IEEE Visualization*, pp. 361-378, 1990.

[14] L. J. P. van der Maaten and G. E. Hinton. Visualizing data using t-SNE. Journal of Machine Learning Research, 9:2579–2605, 2008.

[15] J. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, 29(1):1–27, 1964.

[16] P. Larranaga,; Kuijpers, C. M. H.; Murga, R. H.; Inza, I. & Dizdarevic, S. (1999): Genetic algorithms for the travelling salesman problems: A review of representations and operators, Artificial Intelligence Review, Vol. 13, 129–170

[17] J. Nam, K. Mueller, "TripAdvisorN-D: A tourism-inspired high-dimensional space exploration framework with overview and detail," *IEEE Trans. Vis. and Comp. Graphics*, 19(2):291-305 2012.

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

[19] P. Oesterling, G. Scheuermann, S. Teresniak, G. Heyer, S. Koch, T. Ertl, G.Weber. "Two-stage framework for a topology-based projection and visualization of classified document collections," *Visual Analytics Science and Technology (VAST),* pp. 91 –98, 2010.

[20] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by Locally Linear Embedding. Science, 290(5500):2323–2326, 2000.

[21] H. Siirtola, T. Laivo, T. Heimonen, and K. Raiha, "Visual perception of parallel coordinate visualizations," *Proc. International Conf. Information Visualisation*, p.3-9, July 15-17, 2009.

[22] V. de Silva, J. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," *Proc. NIPS*, pp. 705–712, 2003.

[23] J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. Science, 290(5500):2319–2323, 2000.

[24] W. Torgerson, "Multidimensional scaling: I. Theory and Method," *Psychometrika*, 17:401–419, 1952.

[25] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Processing*, 13(4): 600–612, 2004.

[26] Z. Wang, A. C. Bovik, "A universal image quality index," *IEEE Signal Processing Letters*, vol. 9, pp. 81–84, Mar. 2002.

[27] M. Williams, T. Munzner, "Steerable, progressive multidimensional scaling," *Proc. IEEE Symp. Information Visualization*, pp. 57-64, 2004.

[28] J. Yang, D. Hubball, M. Ward, E. Rundensteiner, W. Ribarsky, "Value and relation display: Interactive visual exploration of large datasets with hundreds of dimensions," *IEEE Transactions on Visualization and Computer Graphics* 13(3): 494-507, 2007.

[29] A. Zelenyuk. D. Imre, "Single particle laser ablation time-of-flight mass spectrometer: an introduction to SPLAT," *Aerosol Sci. & Techn*., 39(6):554-568, 2005.

[30] Z. Zhang, K. McDonnell, K. Mueller, "A network-based interface for the exploration of high-dimensional data spaces," *IEEE Pacific Vis*, Songdo, Korea, pp. 17-24, March, 2012.

[31] http://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength (accessed 3/2011)

[32] http://archive.ics.uci.edu/ml/datasets/Waveform+Database+Generator+%28Version+1%29 (accessed 3/2011)

Jenny Hyunjung Lee is a PhD candidate in computer science from Stony Brook University. Her research interests are information visualization, visual analytics, and data mining.

Kevin T. McDonnell earned his PhD in computer science from Stony Brook University. He is an associate professor of computer science and mathematics at Dowling College. His research interests include scientific and information visualization, visual analytics and HCI.

Alla Zelenyuk received her PhD in chemical physics from Moscow Institute of Physics and Technology, Russia and is currently a senior research scientist at Pacific Northwest National Laboratory. Her research interests include real-time multi-dimensional characterization of physical and chemical properties of individual aerosol particles.

Dan Imre earned a PhD in physical chemistry from Massachusetts Institute of Technology and is currently working as a consultant with Imre Consulting. His expertise includes single particle mass spectrometry, multidimensional single particle characterization, and data analysis. He has authored more than 110 peer-reviewed papers.

Klaus Mueller received a PhD in computer science from Ohio State University and is a professor of computer science at Stony Brook University. His research interests are visualization, visual analytics, and medical imaging. He won the NSF CAREER award in 2001 and the SUNY Chancellor's Award in 2011.