Vision, Modeling, and Visualization (2020), pp. 1–9
J. Krüger, M. Niessner, and J. Stückler (Eds.)

Interactive Generation of 1D Embeddings from 2D Multi-dimensional Data Projections

Quynh Quang Ngo and Lars Linsen

Westfälische Wilhelms-Universität Münster, Germany

Abstract
Visual analysis of multi-dimensional data is commonly supported by mapping the data to a 2D embedding. When analyzing a sequence of multi-dimensional data, e.g., in case of temporal data, the usage of 1D embeddings allows for plotting the entire sequence in a 2D layout. Despite the good performance in generating 2D embeddings, 1D embeddings often exhibit a much lower quality for pattern recognition tasks. We propose to overcome the issue by involving the user to generate 1D embeddings of multi-dimensional data in a two-step procedure: We first generate a 2D embedding and then leave the task of reducing the 2D to a 1D embedding to the user. We demonstrate that an interactive generation of 1D embeddings from 2D projected views can be performed efficiently, effectively, and targeted towards an analysis task. We compare the performance of our approach against automatically generated 1D and 2D embeddings involving a user study for our interactive approach. We test the 1D approaches when being applied to time-varying multi-dimensional data.

1. Introduction

Multi-dimensional data are commonly visualized using embeddings into a lower-dimensional space. The main goal for establishing the embedding is to preserve properties and characteristics of the multi-dimensional data in the embedding. Given that analysis tasks may vary, different approaches for generating mappings have been proposed that target different objective functions [SNAA19, NA19b]. Often multiple criteria shall be fulfilled, where the emphasis may shift depending on the task. Pattern recognition tasks such as detecting and analyzing clusters and outliers are among the most common tasks. The neighborhood-preservation property of the t-distributed stochastic neighbor embedding (t-SNE) approach [vdMH08] facilitates such visual pattern recognition tasks in the embedding [KB19, LCYH17, BM16, YMXC13].

For visualization purposes, 1D, 2D, or 3D embeddings may be used. While several studies argue that 2D embeddings should be preferred over 3D embeddings due to depth perception issues in 3D and the necessity to interact with 3D embeddings [SMT13], 1D embeddings have rarely been used. When looking into sequence of multi-dimensional data such as temporal data or ensemble data, 1D embeddings are extremely useful, as they allow for plotting the entire sequence in a 2D layout, e.g., by using time as a second axis [FL19, FML16, LHFL19, NHL19]. When using a 2D embedding with time as a third axis, one would introduce again the issues of working in a 3D space.

While approaches such as t-SNE (with proper parameter settings) often produce excellent results for 2D embeddings, the mapping to 1D embeddings sometimes have some issues. As t-SNE (like many other methods) perform an iterative optimization process that starts with a random initial configuration, the low number of degrees of freedom in a 1D space may lead to the process to be stuck in a local optimum. We acknowledge this issue and present examples, where 1D embeddings using t-SNE have a much lower quality than the respective 2D embeddings, see Section 3.

We propose to overcome the issue by involving the user. The main idea is to create an automatic mapping to a 2D embedding using an existing dimensionality reduction algorithm (as a proof of concept we use t-SNE) and then let the user define the last step of mapping from 2D to a 1D embedding. There are multiple arguments for our strategy to be useful and successful: (1) Interacting in a 2D visual space is intuitive for humans and they may outperform automatic solutions. (2) Interactively defining a 1D embedding in a 2D space is as simple as drawing a curve that fits the data. (3) The user can define the mapping according to the analysis tasks in mind and can bring in own experiences and expertise in pattern recognition tasks. Often the desired embedding has primary and secondary design goals, where the weighting is hard to determine automatically, while the user is able to find the best trade-off.

We present our interactive mapping approach from 2D to 1D embeddings by drawing a curve and mapping to the curve in Section 4. We perform a user study to evaluate the robustness of the interactive mapping approach and quantitatively compare the quality of the interactively generated 1D embeddings to 1D and 2D embeddings obtained by t-SNE, see Section 5.

In a second step, we extend our approach to time-varying multi-dimensional data. Recently, dynamic t-SNE has been proposed, which tries to keep t-SNE layouts consistent over time [RFaT16]. This dynamic approach can be combined with our interactive 1D embedding by applying our approach to the first time step. We further present how the interactively defined curve used for mapping 2D to 1D embeddings can be automatically adjusted when the 2D embeddings change over time. We present results and compare them to a dynamic 1D t-SNE approach, see Section 6.

Our main contributions can be summarized as follows:

- A quantitative comparison of 1D and 2D t-SNE for cluster preservation.
- A semi-automatic approach to generate 1D embeddings, where a 2D embedding is generated automatically and a user-centric approach is proposed to perform the 2D-to-1D mapping.
- A user study to evaluate the robustness of the user-centric component to our approach.
- A quantitative comparison of the quality of the 1D embeddings achieved with our approach when compared to 1D t-SNE.
- An extension to embeddings for time-varying multi-dimensional data with an automatic adjustment of the 2D-to-1D mapping.

2. Related Work

The multi-dimensional data embedding technique t-SNE was introduced by van der Maaten and Hinton [vdMH08]. The objective was to find a low-dimensional embedding that preserves the neighborhoods, which is essential for many pattern recognition tasks, as it facilitates preserving clusters. Extensions of the original work try to increase the efficiency of the method by proposing approximate solution, i.e., effectively trading off speed for accuracy [VDM14], and include the user to decide the granularity of the outcome projection [PHL*16]. Since the original approach is supposed to produce results of higher quality, we use the original approach for our comparisons. Wattenberg et al. [WVJ16] indicate that the quality of the t-SNE outcome depends on properly chosen hyper-parameters (in particular, perplexity) and that bad choices can lead to misinterpretations. However, they did not discuss possible issues with 1D projection (even for proper hyper-parameter selection), which we discuss in Section 3.

For time-varying multi-dimensional data, Rauber et al. [RFaT16] proposed dynamic t-SNE, which aims at creating smoothly varying temporal evolutions of 2D embeddings. The approach tries to minimize differences between embeddings of subsequent time steps. The visualization uses animations of 2D embeddings. We, instead, show the temporal evolution by drawing 1D embeddings over time, which supports easy analysis of the evolution of clusters over time using a static visualization. This concept of showing the evolution of 1D embeddings over a temporal axis has already been sucessfully applied to multiple applications using Multi-dimensional Scaling (MDS) [FL19, FML16, LHFL19] as well as dynamic t-SNE [NHL19]. The latter approach deploys 1D t-SNE for the initial time step, which may suffer from the issue mentioned above. 1D t-SNE has also been used in other contexts such as building t-SNE heatmaps [LRH*17]. The authors claim that 1D t-SNE would preserve structures as well as 2D t-SNE without further proof of the claim. We show in Section 3 that this is not necessarily true. 1D projections plotted over an attribute vector have been proposed using the first principle component to interpret latent spaces [LJLH19]. Jäckle et al. [JFSK16] presented a technique named temporal MDS plots. They propose applying MDS separately for certain sliding time windows with overlapping offsets to have a 1D embedding of data in ascending temporal order. The technique can be applied in our case, but extracting similarities without using the whole entries of matrices would cause losing information. Bernard et al. [BWS*12] apply dimensionality reduction single multivariate time series for analyzing time-oriented multi-dimensional data, while we are looking into comparing multiple time series. An approach to project multiple multivariate time series into a 2D plot was presented by Wilhelm et al. [WVZ*15] to study horse motion capture data. However, their method requires given similarities of entities between different time steps, which we do not assume.

3. Comparing 1D to 2D Embeddings Using t-SNE

In this section, we want to demonstrate that 1D embeddings obtained by t-SNE may have significantly lower quality that respective 2D embeddings. First, we need to discuss what quality measure shall be used, before we apply it to 1D and 2D t-SNE for concrete examples.

Quality Measures. The t-SNE approach is motivated by the objective to preserve neighborhoods, which is crucial for performing pattern recognition tasks such as cluster analysis in the embedded view. Consequently, we want to test and quantify how much clusters were preserved in the embeddings. Many measures are known to judge dimensionality reduction methods [EMK*19, NA19a]. Among them, the silhouette coefficient [Rou87] is a widely applied measure for evaluating the quality of a cluster and can be used to evaluate the performance of embedding techniques (such as t-SNE) in preserving cluster structures.

In order to validate the cluster preservation quality of an embedding, we can look into the silhouette coefficients of all points in the embedding. A common approach for combining the silhouette coefficients of all points in a layout is to compute the *average silhouette coefficient*. However, one may argue that we are not interested in the average case, but in the worst case, i.e., one would compute the *minimum silhouette coefficient*. Using the average or minimum silhouette coefficient we can quantify the quality of an embedding and obtain values in the range $[-1,1]$, where larger values are better. However, one may argue that the exact value of the silhouette coefficient for a point does not matter, as long as it is positive. Hence, we argue that a better measure for the quality of cluster preservation by an embedding is to compute how often the silhouette coefficient is positive. Hence, we propose to define an *outlier index* as the percentage how often the silhouette coefficient is positive, i.e., $O(D) = \frac{p}{|D|}$, where p is the number of points with a positive silhouette coefficient and $|D|$ is the overall number of points in the dataset. The outlier index $O(D)$ is always in $[0,1]$, where again higher values are better. The perfect outlier index of $O(D) = 1$ means that 100% of the silhouette coefficients are positive, i.e., no mixing of clusters occurs.

In the remainder of the paper, we will base our argumentation mainly on the outlier index, but will also report average and minimum silhouette coefficients.

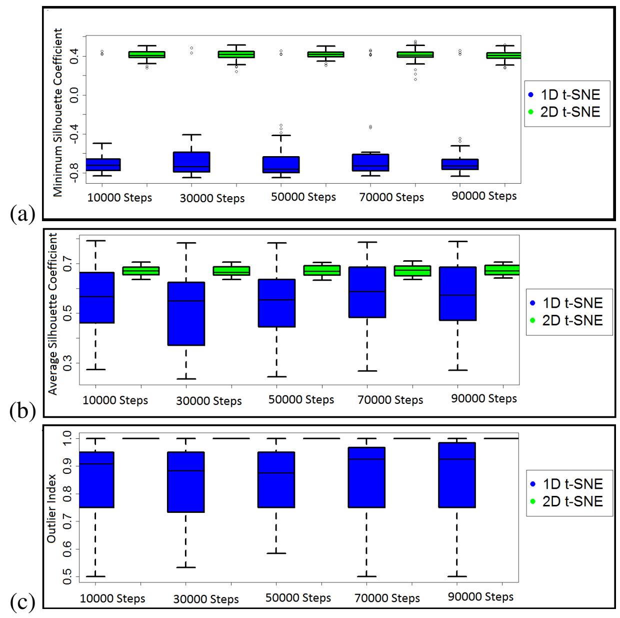


Figure 1: *Comparing 1D and 2D t-SNE results when applied to Four Cluster data set by showing boxplots of (a) the minimum silhouette index, (b) the average silhouette coefficient, and (c) the outlier index for different number of t-SNE iterations. 2D t-SNE clearly outperforms 1D t-SNE.*

Comparison. For testing 1D and 2D embeddings using t-SNE, we use a simple dataset with four clusters. The points of the dataset represent nodes in a network, on which dynamical simulations are executed. The similarity of the multi-dimensional data points is computed as the similarity of the time series of the dynamics, while the labels (or classes) represent the topological clusters in the network. For the given example, the dynamics match topology, i.e., the four classes actually form four non-mixing clusters in the multi-dimensional space.

We can visually observe that the 2D embedding using t-SNE preserves the four clusters well, see Figure 2(a). The 1D embedding using t-SNE, however, visually exhibits a mixing/interleaving of the clusters, see Figure 2(b)(top). To back up this finding with quantitative measures, we compute the average silhouette coefficient, the minimum silhouette coefficient, and the outlier index (see Figure 1). As the outcome of the embedding depends on the initial random configuration and hyper-parameters (perplexity and number of iterations), we compute ensembles of embeddings and present the quality measures for the ensembles in the form of boxplots. To demonstrate that the iterative optimization had converged, we show the boxplots for different number of iterations (no major changes occur when plotting for 10k, 30k, 50k, 70k, or 90k iterations). The outcome of the quantitative quality assessment is that 2D t-SNE performs drastically better no matter what quality measure is used. We performed statistical analyses using a paired t-test and obtained that all differences shown in the figures are statistically significant with p-values in the order of 10^{-5} to 10^{-16}. Hence, we have shown that the quality of t-SNE may deteriorate substantially when going from 2D to 1D embeddings. Of course, we cannot and do not want to say that the 1D embeddings are generally worse, but we provided evidence that they can be worse. In the remainder of the paper, we will provide further examples and tests.

4. Interactive 1D Embeddings

Dimensionality reduction from a multi-dimensional space to a 2D embedding inevitably produces loss, in general. However, projection methods like t-SNE have proven to produce good 2D embeddings for many applications. The previous section showed that this goodness of the embedding may get lost when reducing the dimensionality further to a 1D embedding. However, producing a 1D embedding from a 2D embedding while maintaining certain properties is something that humans are quite capable of and where humans may outperform automatic solutions. Hence, we propose a user-centric approach for mapping a 2D distribution to a 1D embedding.

Given a 2D data set, we propose to generate a 1D embedding by having the user interactively drawing a curve into a 2D scatterplot. The user performs the drawing with a certain task in mind such as keeping all clusters separated. Then, all data points of the 2D scatterplot get projected onto the curve and the curve is flattened to generate a 1D embedding of the 2D data set. The individual steps of our *interactive* 1*D embedding* are detailed below. The accompanying video shows examples of the interactions.

Interactive Curve Drawing. Given a 2D scatterplot, the task is to draw a curve that follows the structure of the data and captures all its components. The curve needs to be one connected component. This task is given to the user and as such shall be easy and fast to perform. The easiest and arguably fastest way to define a curve that fits a certain shape is to define a piecewise linear curve in the form of a polyline. A polyline can interactively be defined by clicking at the respective positions in the desired order. A piecewise linear curve comes with the additional advantage that the projection onto the curve is easy and fast to compute. Figure 2(a) shows an example of a interactively defined, piece-weise linear curve that traverses all relevant components (the four clusters) of the given scatterplot.

Projection to Curve. Projecting 2D points onto a curve is a well-known problem with a variety of solutions depending on the type of curves. Most work is concerning with piece-wise polynomial curves of higher order, e.g., [OKL*10, MH03]. We decided to use piece-wise linear curves (i.e., polynomials of degree 1), which did not only make the definition of curve easy and controllable, but also allows for a simple and efficient projection to that curve: We project a given point $\mathbf{p}$ to each line segment separately and then take the mapped point $\mathbf{p}'$ with minimum distance to the given point $\mathbf{p}$ as its projection.

Flattening the Curve. Given the piece-wise linear curve L consisting of line segments $\mathbf{a}_0\mathbf{a}_1, \ldots, \mathbf{a}_{n-1}\mathbf{a}_n$, we derive a bijective mapping from the curve domain to $[0,1]$ to flatten the curve. Let l_i denote the length of line segment $\mathbf{a}_i\mathbf{a}_{i+1}$, then the length of the curve L is given by $l = \sum_{i=0}^{n-1} l_i$. We define the flattening by mapping $\mathbf{a}_i\mathbf{a}_{i+1}$ linearly to $[b_i, b_{i+1}] = \left[\sum_{k=0}^{i-1} l_k/l, \sum_{k=0}^{i} l_k/l\right]$ for $i = 0, \ldots n-1$. Given a point $\mathbf{p}$ with its projection $\mathbf{p}'$ onto the piece-wise linear curve

L being $\mathbf{p}' = \mathbf{a}_j + t_0(\mathbf{a}_{j+1} - \mathbf{a}_j)$. Then, the 1D projection of $\mathbf{p}$ is defined by $\overline{p} = b_j + t_0(b_{j+1} - b_j)$.

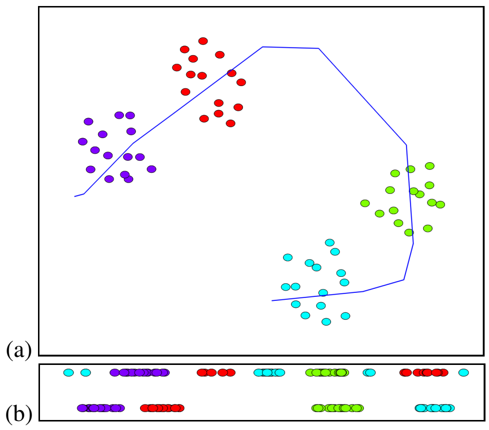


Figure 2: *(a) 2D t-SNE of dataset with four clusters and interactively drawn curve that traverses the clusters. (b) 1D t-SNE embedding of the same dataset (top) and interactive 1D embedding when projecting onto the drawn curve (bottom). The interactive 1D embedding preserves the clusters in contrast to 1D t-SNE.*

5. Evaluation

Since our interactive 1D embedding approach is mainly determined by the ability of users to reliably perform a suitable 2D-to-1D mapping by drawing a curve, we performed a user study to test the robustness of the projection quality via the interactive 1D embedding approach. Giving the ensemble of projections obtained by users drawing curves, we can compare the quality of the ensemble against automatic approaches. We compare the outcome of the interactive 1D embedding against 1D and 2D t-SNE using the quality measures introduced in Section 3.

Data Sets. For the evaluation, we identified a number of data sets with different numbers of clusters. Apart from using the data set with 4 clusters/classes (cf. Figure 2(a)) introduced in Section 3, we apply the defined methods to the following three data sets (from [FS18]): The Flame data set [FM07] that contains two labeled classes as shown in Figure 3(a), the $R15$ data set [VRB02] that contains 15 labeled classes, see Figure 3(b), the Aggregation data set [GMT07] that consists of 7 labeled classes, see Figure 3(c). Since our objective is to project from 2D to 1D, the chosen data sets have a dimensionality of 2. One should note that the classes are mostly also forming clusters, but some of the classes are partially overlapping.

2D t-SNE. Since we want to compare the outcome of our interactive 1D embeddings to both 1D and 2D t-SNE results, the input to our user study shall not be the scatterplots shown in Figure 3, but the scatterplots that are obtained after applying 2D t-SNE. Since t-SNE is not reproducing 2D scatterplots when applied to a 2D input, the outcome of the 2D t-SNE approach may vary quite a bit from its input. This can be observed in Figure 3, where the 2D t-SNE results in (d) and (e) are quite close to its input in (a) and (b), while the 2D t-SNE results in (f) is rather different to its input in (c).

User Study. We conducted a user study to test how reliably users can generate a “good" 1D embedding from a 2D scatterplot. As input we used the 2D t-SNE projections shown in Figures 2(a) and 3(d-f). However, the class labels are not revealed to the study’s subjects. Instead, all points in the scatterplot were drawn using black color.

The body of participants in our study consists of 38 subjects, of which 7 have profound visualization knowledge (working in a visualization research group), while 31 do not. The subject body consisted of 29 males and 9 females with 29 being of age 20-29, 8 of age 30-39, and 1 of age 60. The majority of the subject body were graduate students. All subjects reported to have no visual impairment and normal or corrected to normal vision.

Participants were instructed on how to draw the curve and were explained how 2D points will be projected on the curve. The were also told that the task is to maintain clusters when projecting from 2D to 1D. Each participant was then asked to produce the best outcome for each of the four data sets. They were also given feedback of how the 1D embedding would look like and were given the opportunity to adjust their curve drawings to improve the embedding. Each subject performed the same task on the four data sets in the order Four Clusters, Flame, R15, Aggregation.

Examples of the outcome of the user study is shown in Figure 3. Figure 3(g-i) show examples of curves that were drawn by the user given the 2D t-SNE input of Figure 3(d-f). We observed that the subjects consistently manage to draw curves that pass through all clusters. Interestingly, some subjects tried to maximize separation of clusters by adding detours to the curves as shown in Figure 3(h). The interactively obtained 1D embeddings are shown in Figures 3(j-l) bottom. Visual inspection shows that the classes are quite well separated with some issues on the Aggregation data set, where the 2D t-SNE had produced some overlaps.

All subjects reported that they felt quite confident with their task of drawing the curve, which indicates that the tasks given to the subjects was indeed somewhat intuitive,

Comparison and Statistical Analysis. To compare the interactive 1D embeddings against 1D and 2D t-SNE, we perform statistical analyses of the set of results obtained by our interactive 1D embedding against the t-SNE results. Since t-SNE outcome depends on the initial random configuration, we computed for both 1D and 2D t-SNE an ensemble of embedding results (but with the same perplexity and maximum number of iterations, where perplexity was optimized for each data set independently). Then, we compute for each of the four tested data sets and each of the three methods (interactive 1D embedding, 1D t-SNE, 2D t-SNE) the quality of the embedding. We argued that the outlier index would be the most appropriate measure, but also report average and minimum silhouette coefficient. Results are reported in the form of box plot, which are shown for each of the data sets and the three quality measures. To statistically compare the outcome against each other, we perform a pairwise comparison of the three methods using a paired t-test. The respective p-values are shown in Table 1.

Results. In Table 1, the outcome of the pairwise comparison is color-coded by the winner: red for our 1D embedding, green for 2D t-SNE, and blue for 1D t-SNE, black if there was no statisti-

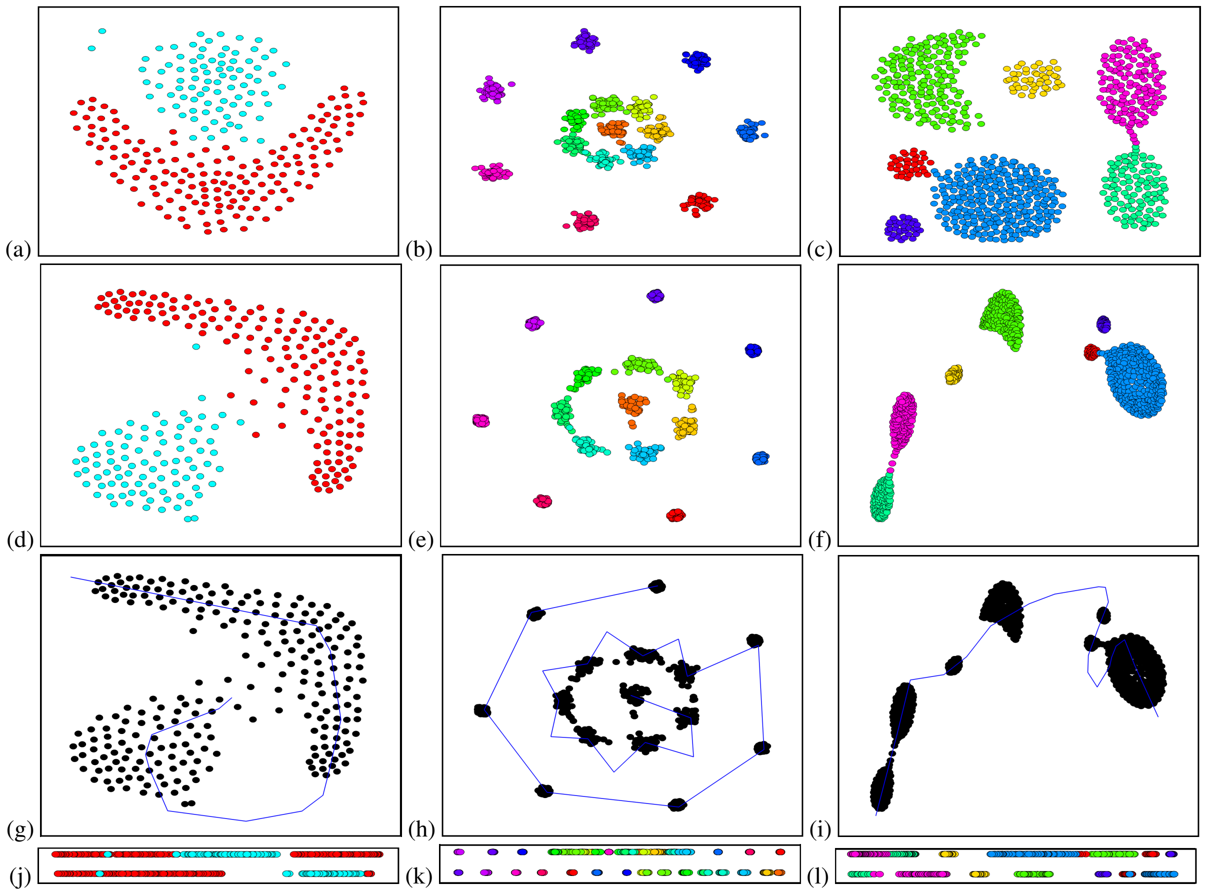


Figure 3: *(a-c) Scatterplots of 2D labeled data sets referred to as Flame, R*15*, and Aggregation. (d-f) Respective 2D t-SNE plots of the data sets, which are the plots used in our user study. (g-i) Examples of interactively generated curves within our user study, where labels were not color-coded. (j-l) Respective 1D embeddings using 1D t-SNE (top) and interactive 1D embedding (bottom) when projecting to the drawn curves.*

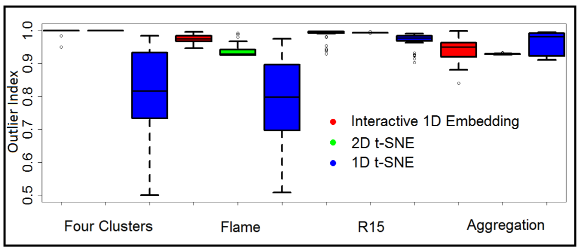


Figure 4: *Box plots of outlier index for comparing the three techniques for the four data sets.*

cal significance. Looking at the outlier index (also see Figure 4), our 1D embedding generally performs best and the 2D t-SNE outperforms the 1D t-SNE. However, for one of the data sets (Aggregation), things are a bit different. We already had seen that 2D t-SNE had issues and mixed clusters. As this was given as input to our method, the performance of our method also dropped. Our method actually outperformed 2D t-SNE, but 1D t-SNE did even better. Looking at the average and minimum silhouette coefficient, things get a bit more mixed, but at least for the average silhouette coefficient our method seems to perform best, while in terms of the minimum silhouette coefficient, 2D t-SNE seems to perform best.

6. Time-varying Interactive 1D embedding

One of the main motivations in introducing 1D embeddings was to show the temporal evolution of time-varying multi-dimensionel data in an intuitive visual encoding. Using the 1D interactive embedding introduced above, we can initialize a 1D t-SNE layout and then apply the dynamic t-SNE [RFaT16] approach to it, whose primary purpose is to maintain a smooth transition between consecutive time steps of the t-SNE projection layouts for time-varying multi-dimensional data. However, during the execution of the dynamic t-SNE approach the 1D embedding may need to be updated depending on the quality of the dynamic t-SNE result. We propose an automatic scheme to update the interactively defined 1D curve during the evolution of time.

Timeline View. Given a collection of 1D projections for different

Dataset	Min. Silh. Coeff.	Avg. Silh. Coeff.	Outlier Index
Four Clusters	$p_{12} = 0.08$	$p_{12} < 10^{-11}$	$p_{12} = 0.2$
	$p_{13} < 10^{-15}$	$p_{13} < 10^{-14}$	$p_{13} < 10^{-19}$
	$p_{23} < 10^{-15}$	$p_{23} < 10^{-9}$	$p_{23} < 10^{-9}$
Flame	$p_{12} < 10^{-15}$	$p_{12} < 10^{-15}$	$p_{12} < 10^{-12}$
	$p_{13} < 10^{-15}$	$p_{13} < 10^{-9}$	$p_{13} < 10^{-9}$
	$p_{23} < 10^{-15}$	$p_{23} = 0.7$	$p_{23} < 10^{-7}$
R15	$p_{12} < 10^{-14}$	$p_{12} < 10^{-6}$	$p_{12} = 0.03$
	$p_{13} = 0.01$	$p_{13} = 0.02$	$p_{13} = 0.003$
	$p_{23} < 10^{-15}$	$p_{23} = 0.02$	$p_{23} < 10^{-6}$
Aggregation	$p_{12} < 10^{-15}$	$p_{12} < 10^{-15}$	$p_{12} = 0.004$
	$p_{13} < 10^{-9}$	$p_{13} < 10^{-5}$	$p_{13} = 0.01$
	$p_{23} = 0.0004$	$p_{23} < 10^{-13}$	$p_{23} < 10^{-6}$

Table 1: *p-values of paired t-test, where p_{12} stands for the p-value when comparing interactive 1D embedding to 2D t-SNE, p_{13} when comparing interactive 1D embedding to 1D t-SNE, and p_{23} when comparing 2D t-SNE to 1D t-SNE. p-values are highlighted by color, when the difference is statistically significant at level 0.05. The colors indicate the winning method, where red stands for interactive 1D embedding, green 2D t-SNE, and blue for 1D t-SNE. Black indicates no significant difference.*

time steps of a time-varying 1D multi-dimensional data, one could observe the development of time-related patterns such as merging or splitting of clusters in a timeline view as shown in Figure 5(b). The x-axis of the view represents time, while the y-axis represents the 1D embedding. Given one point in the data set, the embeddings of the point at two consecutive time steps are connected by a line segment. All the line segments that are connected to each other will provide a means to track the development of the points over time. The color of the lines can be used to encode the class labels.

In order to produce high-quality time-varying time lines, the user would need to draw a 2D curve for each time step. Given that data sets may consist of hundreds or thousands of time steps, this is not a viable solution. Since the dynamic t-SNE approach was designed to produce time-coherent 2D projections, we can restrict ourselves to only draw the curve for the first time step and use that curve for all 2D layouts produced by the dynamic t-SNE approach. Figure 5(a) shows a curve drawn into the 2D t-SNE layout for the first time step of the Four Cluster data set.Figure 5(b)(top) shows the timeline view when using that curve for 100 time steps of the dynamic 2D t-SNE layouts to obtain our 1D embeddings. Figure 5(b)(bottom) shows in comparison the timeline view when using dynamic 1D t-SNE. The clusters in the time-varying multi-dimensional data never mix, which is not obvious when looking at the dynamic 1D t-SNE timeline. Our approach produces a much more stable result. Still, even with our approach the timelines of the clusters start to cross each other after many time steps. The reason for this behavior is that the dynamic 2D t-SNE is not robust enough and introduces rotations of cluster pairs. Therefore, we next propose an automatic adjustment of our interactively drawn curve over the time steps.

Tracking of Curve. To compensate for the motions that the dynamic 2D t-SNE approach allows, we propose to use an automatic tracking of the curve that was drawn by the user for the initial time step. More precisely, we monitor the changes of the neighborhoods of our curve over time and adjust the shape of the curve accordingly. To account for large local deformations, the piece-wise curve gets refined by introducing additional vertices when necessary.

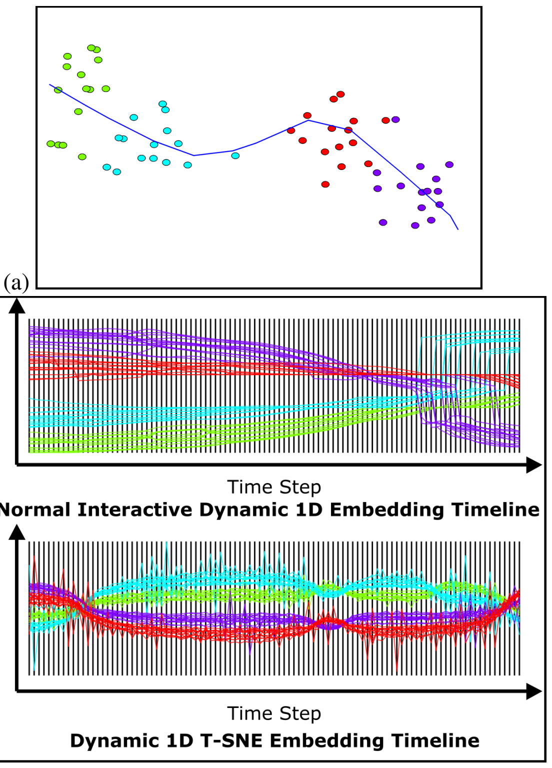
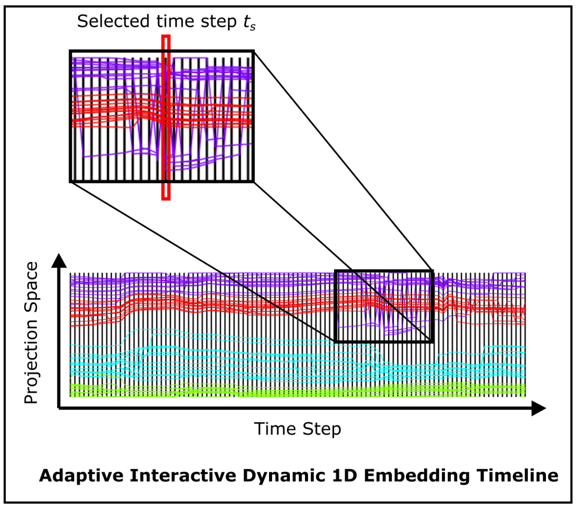


Figure 5: *Time-varying multi-dimensional data visualization: (a) First time step of dynamic 2D t-SNE output and interactively generated curve. (b) Timeline view showing 1D embeddings over time for 100 time steps: While dynamic 1D t-SNE (bottom) generates many crossings of time lines and mixing of clusters, the projection to the interactively generated curve (top) preserves the clusters well for the first half of the time steps.*

To account for the temporal changes of the neighborhood of a vertex on the curve, we make local considerations to update its position (and thus the curve's position) over the time steps. For each vertex $\mathbf{p}$ of the piece-wise linear (possibly refined) curve that was drawn by the user, we consider its k-nearest neighbor points from the 2D scatter plot. From those k points, we extract the barycenter $\mathbf{o}$ and the respective two principal directions $\mathbf{e}_1$ and $\mathbf{e}_2$. We call ($\mathbf{o}$, $\mathbf{e}_1$, $\mathbf{e}_2$) the local frame of point $\mathbf{p}$, which forms a 2D Cartesian coordinate system. We store for point $\mathbf{p}$ its local frame as well as its position with respect to the local frame. In the next time step, we re-compute the local frame based on the updated positions of the k-nearest neighbors and re-position vertex $\mathbf{p}$ to a new position that has the same coordinates with respect to the local frame as before, but now using the updated local frame. As such the vertex $\mathbf{p}$ always adjusts its position when the positions of the k-nearest neighbors change. During this process, we may need to flip the orientation of principal directions as well as swap first and second principal direction to have matching local frames of consecutive time steps. This procedure is executed for each time step of the timeline.

(a)


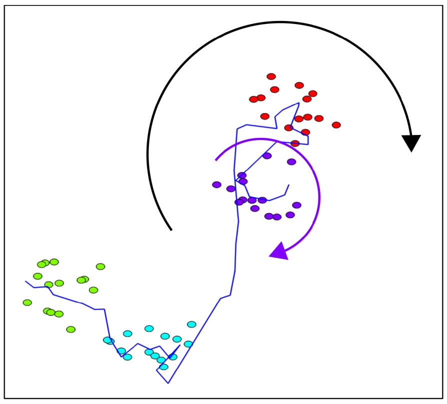
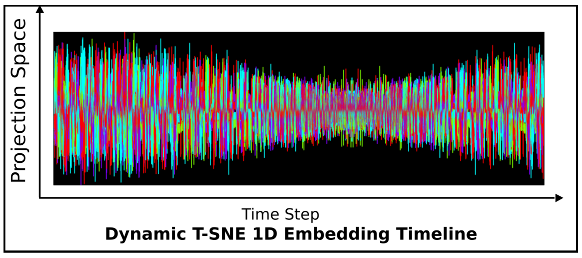
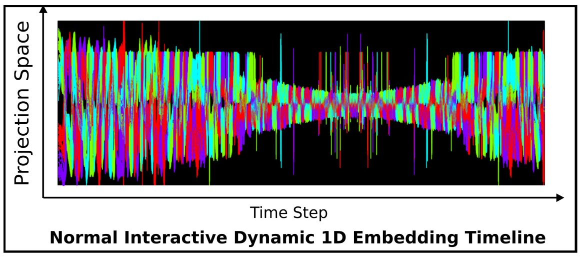
(b)


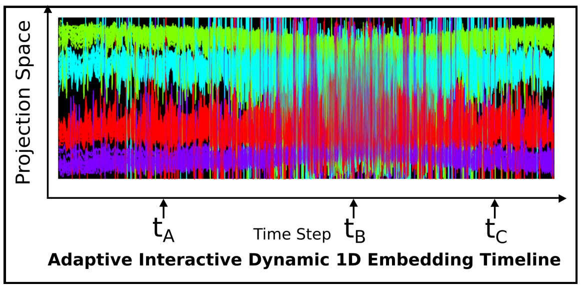
Figure 6: *Automatic curve tracking: (a) Timeline view generated by automatically tracking the initial interactively generated curve. The clusters are generally well-preserved except for the intersections of the red and purple time lines around time step $t_s = 69$ shown in the inset. Selecting time step $t_s = 69$ allows for its investigation in the 2D embedding (b). We observe a self-intersection of the tracked curve for time step $t_s = 69$ due to swirling motions of the purple and red clusters (black arrow).*

Given the example we showed in Figure 5(a), we can process the same example using the automatic tracking of our initial user-defined curve. The respective result is shown in Figure 6(a). We observe that the mixing of clusters is now (mostly) prevented, i.e., the clusters remain separate during the evolution. The edge crossing of the timeline is effectively prevented. **Results.** Figure 7 shows the same example of the Four Clusters data set, but now the simulation was run for much longer (thousands of time steps), such that the clusters actually merge. We synthetically extended the time series by adding a splitting event after the merging, i.e., the clusters start separating again.

Figure 7 shows a comparison of the timeline views. The dynamic 1D t-SNE approach (a) does not allow for a time-coherent visualization. Our approach without curve tracking (b) is maybe somewhat better, but the clusters are still not tractable over time. Our approach with curve tracking (c) though shows clearly separated clusters at the beginning (around time step t_A), some mixing in the middle (around time step t_B), and again clearly separated clusters at the end (around time step t_C). When looking at the 2D t-SNE plots and the tracked curves at those three phases (d), this behavior can be verified.

(a)


(b)


(c)


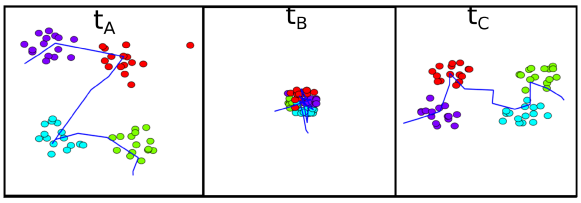
(d)


Figure 7: *Timeline views for extended Four Clusters data set with* 11,500 *time steps including merging and splitting events using dynamic 1D t-SNE (a) and our interactive dynamic 1D embeddings without (b) and with curve tracking (c). With curve tracking (c) three phases of separated clusters (around time step $t_A = 2,362$), mixed cluster (around $t_B = 6,854$), and again separated clusters (around $t_C = 10,210$) can be observed and confirmed via coordinated views to the respective 2D t-SNE plots (d).*

Discussion. When comparing Figures 7(b) and (c), one may argue that the timeline view in (b) exhibits a contracting and expanding phase, which is lost in (c). This is due to the tracking of the curve, which also makes the curve contract and expand, while the length of the curve is always mapped to interval $[0,1]$ for the timeline view. We could also scale by the length of the curve to show the contracting behavior, but the goal here was to observe cluster mixing and splitting, which may be less obvious when scaling.

Our approach introduces one parameter, namely the number of neighbors k used for tracking the curve. We observed that generally good results were obtained when choosing k to be in the range of the cluster sizes.

While the timeline views with our tracked curves generally keep separated clusters separate, we observe that for some time steps that may still be some line crossings, e.g., at time step $t_s = 69$ in Figure 6(a), see inset. Animating the respective 2D embeddings overlaid with the automatically adjusted curve explains the swapping: The purple and red clusters exhibit a swirling pattern over time, i.e., the clusters rotate around each other as indicated by the black arrow inserted into Figure 6(b). In addition, the purple clusters rotates roughly around its barycenter as indicated by the purple arrow in the figure. Due to the swirling of the two clusters, the curve has self-intersections around time step $t_s = 69$, which prevent a meaningful projection onto the curve. In Figure 6(b), it becomes apparent that the points of the purple cluster will be assigned to different parts of the curve.

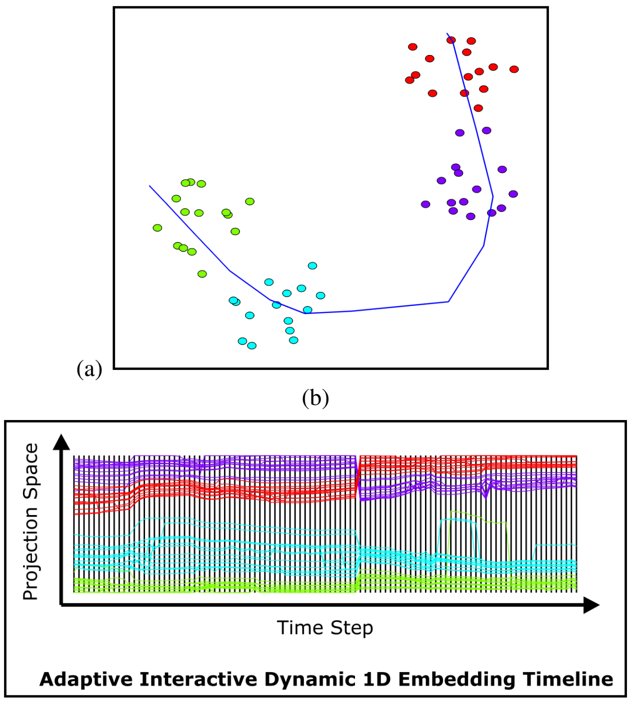


Figure 8: *Interactive re-initialization: (a) Selecting time step $t_s = 58$ the curve can interactively be re-initialized by drawing a new curve. (b) Tracking the new curve from time step $t_s = 58$ onward generates a new timeline view.*

Interactive Analysis. The timeline view and the visualization of the 2D embedding are linked via coordinated interaction. The workflow of an interactive analysis process is supported as follows: Initially, the first time step is shown in the 2D embedding and the user can draw the curve that best follows the data structure. In the case of labeled data, the labels are automatically assigned colors, which can be customized interactively. In the case of unlabeled data, the user can identify clusters in the 2D embedding and interactively assign colors to them. The timelines are then generated with respect to the defined (and tracked) curve and colored accordingly. In the timeline view, on the other hand, each time step can be selected by mouse click and the respective 2D embedding of the selected time step is shown using the same color scheme and overlaid with the automatically tracked curve. The interactive operations are best conveyed in the accompanying video.

Given this interactive set-up of the two views, the entire time series can be analyzed. Merging and splitting events can be analyzed as in Figure 7 and suspicious time points can be investigated as in Figure 6. If the user at some point feels that the curve tracking is suboptimal, an interactive re-initialization of the curve is supported. For example, if a projection artifact is observed, the user can interactively define a new curve for the selected time step, which re-initializes the curve tracking from that time step on. This re-initialization leads to an update of the timeline view, where the timelines change from the selected time step onward. Figure 8 shows an example. Starting with the timelines of the automatically tracked curve in Figure 6, time step $t_s = 58$ is selected for a re-initialization and a new curve is drawn for $t_s = 58$ as shown in Figure 8(a). Then, the timeline view is updated as in Figure 8(b). Here, the user decided for a swapping of the cluster order at time step $t_s = 58$. Since the user initiates teh re-initialization, the discontinuity at time step $t_s = 58$ cannot be misinterpreted as a cluster mixing.

One may consider replacing the interactive curve generation by an automatic curve generation. However, humans are typically good at detecting structures in 2D data distributions and based on different data distributions may follow different strategies to generate the curves. As mentioned above, different design goals for the projection also induce different strategies to define the curve. Still, if the design goal has been defined a priority, e.g., cluster preservation, then automatic approaches may be developed to find curves that are optimal with respect to the design goal. We would want to consider such ideas for future work.

7. Conclusion and Discussion

We presented case studies that provide evidence that the dimensionality of 1D t-SNE is too low to faithfully preserve desired structures, while 2D t-SNE still produced desired results. This observation motivated our approach to interactively generate 1D embeddings from 2D embeddings based on a simple curve drawing. We documented that our strategy leads to desirable results when the 2D embeddings had preserved the structures well. We showed how our approach can be coupled with a dynamic t-SNE approach for time-varying multi-dimensional data visualization, where the 1D embeddings are arranged over a time axis. Automatic curve tracking and interactive re-initialization allowed for improved visualizations.

As t-SNE aims at preserving neighborhoods, the goal of the examples shown in this paper was also neighborhood preservation. Consequently, other possibly desirable properties of projections such as preserving distances are not fulfilled. However, our approach could be coupled with other projection methods. For example, we could use MDS in case distance preservation is the ultimate goal. Of course, the curve drawing strategy would also be different for different design goals of the projections.

The quality of the projection also depends on the ability of the user to generate suitable curves. However, in our user study we observed that users quite quickly developed respective skills. Humans are typically good at detecting structures in 2D data distributions and based on different data distributions may follow different strategies to generate the curves. As mentioned above, different design goals for the projection also induce different strategies to define the curve. Still, if the design goal has been defined a priori, automatic approaches may be developed to find curves that are optimal with respect to the design goal, which we would want to consider for future work.

References

[BM16] BALAMURALI M., MELKUMYAN A.: t-sne based visualisation and clustering of geological domain. In *Neural Information Processing* (2016), Hirose A., Ozawa S., Doya K., Ikeda K., Lee M., Liu D., (Eds.), Springer International Publishing. 1

[BWS*12] BERNARD J., WILHELM N., SCHERER M., MAY T., SCHRECK T.: Timeseriespaths: Projection-based explorative analysis of multivarate time series data. vol. 20. 2

[EMK*19] ESPADOTO M., MARTINS R. M., KERREN A., HIRATA N. S. T., TELEA A. C.: Towards a quantitative survey of dimension reduction techniques. *IEEE Transactions on Visualization and Computer Graphics* (2019), 1–1. 2

[FL19] FOFONOV A., LINSEN L.: Projected field similarity for comparative visualization of multi-run multi-field time-varying spatial data. *Computer Graphics Forum 38*, 1 (2019), 286–299. 1, 2

[FM07] FU L., MEDICO E.: Flame, a novel fuzzy clustering method for the analysis of dna microarray data. *BMC Bioinformatics 8*, 1 (Jan 2007), 3. 4

[FML16] FOFONOV A., MOLCHANOV V., LINSEN L.: Visual analysis of multi-run spatio-temporal simulations using isocontour similarity for projected views. *IEEE Transactions on Visualization and Computer Graphics 22*, 8 (Aug 2016), 2037–2050. 1, 2

[FS18] FRÄNTI P., SIERANOJA S.: K-means properties on six clustering benchmark datasets, 2018. URL: http://cs.uef.fi/sipu/datasets/. 4

[GMT07] GIONIS A., MANNILA H., TSAPARAS P.: Clustering aggregation. *ACM Trans. Knowl. Discov. Data 1*, 1 (Mar. 2007). 4

[JFSK16] JÄCKLE D., FISCHER F., SCHRECK T., KEIM D. A.: Temporal mds plots for analysis of multivariate data. *IEEE Transactions on Visualization and Computer Graphics 22* (2016), 141–150. doi:10.1109/TVCG.2015.2467553. 2

[KB19] KOBAK D., BERENS P.: The art of using t-sne for single-cell transcriptomics. *Nature Communications 10*, 1 (2019), 5416. 1

[LCYH17] LI W., CERISE J. E., YANG Y., HAN H.: Application of t-sne to human genetic data. *Journal of Bioinformatics and Computational Biology 15*, 04 (2017), 1750017. 1

[LHFL19] LEISTIKOW S., HUESMANN K., FOFONOV A., LINSEN L.: Aggregated ensemble views for deep water asteroid impacts simulations. *IEEE Computer Graphics and Applications* (2019), 1–1. 1, 2

[LJLH19] LIU Y., JUN E., LI Q., HEER J.: Latent Space Cartography: Visual Analysis of Vector Space Embeddings. *Computer Graphics Forum* (2019). 2

[LRH*17] LINDERMAN G. C., RACHH M., HOSKINS J. G., STEINERBERGER S., KLUGER Y.: Efficient algorithms for t-distributed stochastic neighborhood embedding. *ArXiv abs/1712.09005* (2017). 2

[MH03] MA Y., HEWITT W.: Point inversion and projection for nurbs curve and surface: Control polygon approach. *Computer Aided Geometric Design 20* (05 2003), 79–99. doi:10.1016/S0167-8396(03)00021-9. 3

[NA19a] NONATO L., AUPETIT M.: Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics 25* (2019), 2650–2673. 2

[NA19b] NONATO L. G., AUPETIT M.: Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics 25*, 8 (Aug 2019), 2650–2673. 1

[NHL19] NGO Q. Q., HÜTT M.-T., LINSEN L.: Visual Analytics of Simulation Ensembles for Network Dynamics. In *Vision, Modeling and Visualization* (2019), Schulz H.-J., Teschner M., Wimmer M., (Eds.), The Eurographics Association. 1, 2

[OKL*10] OH Y.-T., KIM Y.-J., LEE J., KIM M.-S., ELBER G.: Efficient point projection to freeform curves and surfaces. In *Advances in Geometric Modeling and Processing* (Berlin, Heidelberg, 2010), Mourrain B., Schaefer S., Xu G., (Eds.), Springer Berlin Heidelberg, pp. 192–205. 3

[PHL*16] PEZZOTTI N., HÖLLT T., LELIEVELDT B. P., EISEMANN E., VILANOVA A.: Hierarchical stochastic neighbor embedding. *Computer Graphics Forum (Proc. of EuroVis) 35*, 3 (June 2016), 21–30. 2

[RFaT16] RAUBER P. E., FALCÃO A. X., TELEA A. C.: Visualizing time-dependent data using dynamic t-sne. In *Proceedings of the Eurographics / IEEE VGTC Conference on Visualization: Short Papers* (2016), EuroVis '16, Eurographics Association, pp. 73–77. 2, 5

[Rou87] ROUSSEEUW P. J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics 20* (1987), 53 – 65. 2

[SMT13] SEDLMAIR M., MUNZNER T., TORY M.: Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (Dec 2013), 2634–2643. 1

[SNAA19] SAEED N., NAM H., AL-NAFFOURI T. Y., ALOUINI M.: A state-of-the-art survey on multidimensional scaling-based localization techniques. *IEEE Communications Surveys Tutorials 21*, 4 (2019), 3565–3583. 1

[VDM14] VAN DER MAATEN L.: Accelerating t-sne using tree-based algorithms. *J. Mach. Learn. Res. 15*, 1 (Jan. 2014), 3221–3245. 2

[vdMH08] VAN DER MAATEN L., HINTON G.: Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research 9* (2008), 2579–2605. 1, 2

[VRB02] VEENMAN C., REINDERS M., BACKER E.: A maximum variance cluster algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 24* (10 2002), 1273– 1280. 4

[WVJ16] WATTENBERG M., VIÉGAS F., JOHNSON I.: How to use t-sne effectively. *Distill* (2016). URL: http://distill.pub/2016/misread-tsne, doi:10.23915/distill.00002. 2

[WVZ*15] WILHELM N., VÖGELE A., ZSOLDOS R., LICKA T., KRÜGER B., BERNARD J.: Furyexplorer: Visual-interactive exploration of horse motion capture data. vol. 9397. 2

[YMXC13] YI J., MAO X., XUE Y., COMPARE A.: Facial expression recognition based on t-sne and adaboostm2. In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing* (Aug 2013), pp. 1744–1749. 1