

Dimensionality Reduction for Visualization of Time Series and Trajectories

Pattreeya Tanisaro and Gunther Heidemann

Institute of Cognitive Science, University of Osnabrück

Abstract. Visualization is essential for data analysis and it is particularly challenging for data in high-dimensional space, especially for temporal information. Many techniques have been employed in an attempt to transform multivariate time series data to one-dimensional data by reducing the number of features in order to visualize their time-dependent behaviors. However, the applicability of these approaches is restricted to a limited number of data instances that can be visualized simultaneously.

We present a technique to visualize time series and trajectories that overcomes these limitations by transforming these data into subspaces which allows data analysts to easily select the instance of interest from a bunch of data. The benefits of our proposed method are threefold: it provides i) a visual representation of time-dependent data in a massive amount simultaneously, ii) a very concise feature representation and iii) an easy identification of anomalies. The results are demonstrated by employing this technique to various data traits from public archives, they are i) univariate time series data from the UCR archive, ii) multivariate time series data from several sources, and iii) human motion trajectories from two motion capture (MoCap) datasets.

Keywords: Dimensionality Reduction · Visualization · Time Series

1 Introduction

Visualization techniques help us to understand data by observing its patterns. A common approach to meet this challenge is to transform the high-dimensional data representation into lower dimensions. High-dimensional datasets typically have certain traits that can be captured after a transformation to a different coordinate system, and become directly visible to the human eye when mapped to two or three-dimensional space. Although there are numerous dimensionality reduction techniques available; nevertheless, there are much fewer dimensionality reduction approaches that are practical enough for the general application of which take the temporal information into account. Some interesting techniques for visualization of time-dependent data, for example [4, 7, 9] are tied to a domain-specific application and their data are not publicly accessible; therefore, it raises a question of whether to adopt those techniques to a new domain application even though the data may be categorized in the same time characteristics.

Generally, a known method for visualizing time series data is by using a line graph which is widely used in the monitoring of vital signs in order to detect the

abnormal exceeding of a specified threshold. However, the line graph does not work well for multivariate time series, and it is especially difficult to visualize a large collection of signals. On top of that, for the case of a multivariate time series where the interdependencies occur among many variables, detecting the outliers happens to be much more difficult and cannot be solved by visualizing each data feature on the timeline. A conventional approach to visualize high-dimensional time series data is to examine their distance matrices. One benefit of using a distance matrix such as Euclidean distance is that we can further analyze the matrix using a recurrence plot (RP) by applying a threshold distance and the Heaviside function. The RP is well-known for the visualization of time series because it allows any high-dimensional phase space trajectories to be visualized in subspaces through a two-dimensional representation of its recurrences.

Unlike other traditional dimensionality reduction approaches for temporal data which allow a user to analyze signals over time, **our technique allows the user to inspect the structure of a bunch of time series data or trajectories simultaneously and to detect the outliers**. Particularly for multivariate time series which have many practical usages in real-world applications, there is no general solution to compare a bunch of information at the same time. We demonstrate that it is possible to apply conventional dimensionality reduction approaches from a non-time-series to transform high-dimensional time series into low-dimensional subspaces **by neglecting time information in the display**. Although the dimensionality reduction techniques themselves are not new, to the best of our knowledge, such an attempt to visualize the time-dependent data by neglecting the time axis has never been investigated before. The advantages of our approach are threefold. They allow the data analyst: *i) to visualize the large-scale time series data clusters simultaneously, ii) to get a very concise feature representation of the time series regardless of its length and the number of features, and iii) to detect an anomaly in the data.*

2 Data Transformation To Subspaces

Let p be the total number of instances in the dataset. For any given instance i , each individual instance is specified by $\{X^i\}$ where $i \in \{1, \dots, p\}$. For any high-dimensional data sequence X^i with a fixed number of features m and arbitrary length T_i , we can be interpreted X^i as a real-valued matrix X with a dimension $m \times T_i$ as illustrated in Figure 1a. Pick the number of selected components c_n for any transformation F to the matrix X^i where n is the number of dimensionality reduction technique used. For the chosen first principal components c_1 at $n=1$, we obtain $F_1(X^i)$ as illustrated in Figure 1b where $T_i \geq c_1$. Hence, to apply n -times of dimensionality reduction of F to X^i for c_n components, namely $F_n(F_{n-1}(\dots F_1(X^i)\dots))$ as shown in Figure 1c, requires:

$$T_i \geq c_1 \quad \forall i \in \{1, \dots, p\} \quad \text{and} \quad (m \cdot c_1) \geq c_2 \dots \geq c_n \quad (1)$$

Usually, the sequence length of any signal instance is much larger than the selected number of principal components, that is $T_i \gg c_1 \forall i \in \{1, \dots, p\}$. Before

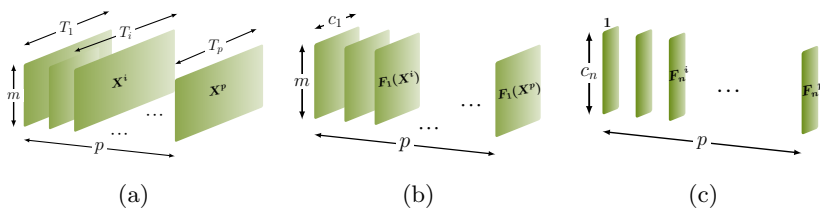


Fig. 1: Transformation of time-dependent data into subspaces. a) p instances of time-dependent data of m features with arbitrary sequence lengths T_i . b) Results after the first transformation of F_1 . From this point onward, the arbitrary size of “time dimension” T_i has become all equal with the selected principal components c_1 . c) The data after an arbitrary n^{th} transformation giving each signal instance of size c_n which can portray a small feature representation of the data.

applying the first order transformation, $n = 1$, we may build a feature vector by normalizing each $X_{j,k}^i$ where $j \in \{1, \dots, m\}$ and $k \in \{1, \dots, T_i\}$ as:

$$X_{j,k}^i \Leftarrow X_{j,k}^i - \bar{X}_j^i \quad (2)$$

where \bar{X}_j^i is the average over the sequence length T_i of feature j . Likewise, for the case of the trajectories of MoCap dataset, we first normalize the stick-figure’s joint positions which were computed by the marker positions following [12] by subtracting the position of the center of the torso from each joint position. The normalization by subtracting the mean is optional but is proved to enhance the visualization in many cases. For the case of different scaling of features, the rescaling prior to applying the dimensionality reduction transformation can be beneficial. However, normalizing time series data by dividing it by its standard deviation does not improve our visualization in general. Similar evidence was reported in [11] for human motion classification. After applying the first transformation of F_1 on each normalized X^i , the data sequence X^i can be newly represented as $F_1(X^i) \in \mathbb{R}^{m \times c_1}$ as depicted in Figure 1b. The time axis now has been replaced with the number of principal components of the first transformation. The feature vector for the second order transformation may be arranged using a concatenation of an average vector to $F_1(X^i)$ as $[\bar{X}_j^i; F_1(X^i)]$. After a second order transformation, $F_2(F_1(X^i))$, the new matrix can be shortly written as $F_2^i \in \mathbb{R}^{1 \times c_2}$ which is depicted in Figure 1c.

3 Datasets

The datasets in this paper were selected by considering the number of data classes up to twelve classes which can be easily identified by different colors. There is no restriction on the number of features nor sequence lengths.

3.1 Univariate Time Series

The UCR archive [3] contains 85 datasets of univariate time series. It is target for benchmarking time series classification. Each dataset consists of a separate training and test set of a fixed sequence length which was already normalized to have zero mean and a standard deviation of one. We picked five datasets from this archive and merged the training and test data together because we only focus on the visualization of data and not on classification. These datasets are *Plane*, *ItalyPowerDemand*, *Wafer*, *CBF* and *ECG5000*.

3.2 Multivariate Time Series

We selected three datasets which are frequently used in benchmarking classification tasks as found in [1, 5, 6]. These three datasets were: *JapaneseVowel*, *NetworkFlow* and *Wafer* (It has the same dataset name but different set from the UCR). The *JapaneseVowel* dataset was a collection of nine male speakers for a total of 640 sequences. Each utterance forms a sequence with lengths in the range of 7-29 and consists of 12 features each. The *Networkflow* dataset represents a network traffic protocol of a total of 1337 sequences with the sequence length of 50-997 where a series of network packets define a sequence. Each packet consists of four attributes which are used to identify the applications that generated the traffic flow. These attributes are a packet size, transfer direction, payload, and the duration. The *Wafer* stands for silicon wafer in semiconductor manufacture. The dataset contains 1194 sequences with the sequence length of 104-198. Each sequence consists of six measurement variables recorded during the etch process. Each wafer is marked as normal or defective.

3.3 Motion Capture

We chose two different MoCap datasets, the UTD-MHAD [2] and the HDM05 [8] to demonstrate the effectiveness of our proposed technique.

The **UTD-MHAD** consists of 27 different actions performed by eight subjects. Each action was recorded using 20 markers in 3D coordinates, resulting in the total number of 60 features. Each subject repeated the same actions four times (trials) for only one cycle, and each action trial has different sequence length. Hence, we have only 32 sequences for each action in total. This small amount of data is statistically not interesting. Based on the results of a quasi-view independent of human movement in 3D described in [12], an eigenvector of the largest eigenvalue still maintains its projection size even when a subject performs the same action facing in a different direction. Therefore, we rotated the actor’s default view by 10, 20 and 30 degrees in order to obtain more samples.

The **HDM05** was originally made up of 130 classes consisting of five subjects, called “bd”, “bk”, “dg”, “mm” and “tr”, performing actions with and without repeating the same cycles separately. Following [10], we grouped the non-repetitive and repetitive motions together yielding 65 actions, resulting in a various number of trials in each action. Some actions e.g., *walk* consists of four types of

walk, they are *walk2StepsL*, *walk2StepsR*, *walk4StepsL*, and *walk4StepsR*. In this dataset, each user had more freedom to perform the action, for example, the numbers of repetitions and the directions of movement were not fixed.

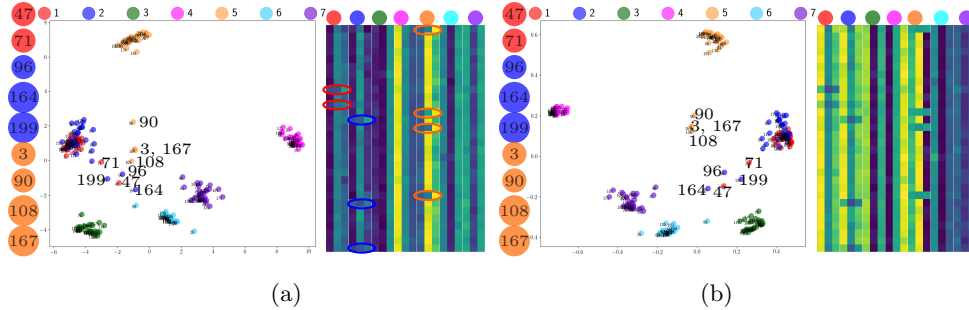


Fig. 2: Results of the dimensionality reduction of 210 sequences of “Plane” using (a) PCA and (b) Kernel PCA with RBF kernel. Nine outliers listed on the left side of each image (a) and (b) can be easily identified in the two-dimensional views (That is only the first two principal components are drawn). The right side of each image shows the feature representation map of the corresponding algorithm from three principal components of a matrix size $30 \times (3 \cdot 7)$, where 30 signals of each data class are in the rows and three components of seven classes are in the columns. The irregular patterns in each class found in the feature representation map highlighted in ellipses at (a)-right can be found in the same positions at the (b)-right. The feature vectors of the data class ● and ● as seen in the feature maps are very similar.

4 Visualization Results

In this section, we will examine the outputs from projecting data on 2D and 3D space after applying the transformations discussed in section 2. Without prior knowledge of the characteristics of the data, the dimensionality reduction techniques were randomly chosen from two characteristics, linear projections i.e., PCA and nonlinear projections such as kernel PCA with nonlinear kernels and t-SNE. We selected some interesting outputs to be demonstrated here.

4.1 Visualization Results of The Univariate Time Series

The results of applying two dimensionality reduction techniques to *Plane* in the UCR archive are displayed in Figure 2a and Figure 2b. Not only do our outputs exhibit to the viewer the intrinsic properties of each data cluster, but they also

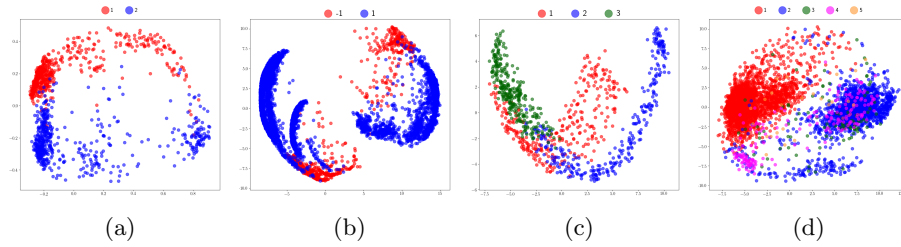


Fig. 3: Results of the dimensionality reduction of selected datasets in the UCR archive. a) 1097 sequences of “ItalyPowerDemand” using Kernel PCA. b) 7174 sequences of “Wafer” using PCA. c) 930 sequences of “CBF” using PCA. d) 5000 sequences of “ECG5000” using PCA. This dataset has much of unbalancing in the data in each class. The number of signals in each class are: (●) 2919, (●) 1767, (●) 96, (●) 194 and (●) 24.

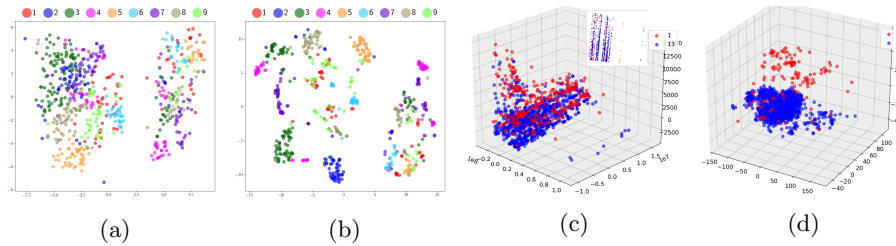


Fig. 4: Results of the dimensionality reduction of multivariate time series. a) “JapaneseVowel” with PCA followed by t-SNE ($n = 2$, the perplexity of t-SNE 40). b) “JapaneseVowel” with PCA followed by double t-SNE ($n = 3$, the perplexity of t-SNE 40 and 30, respectively). c) “NetworkFlow” using PCA and t-SNE. d) “Wafer” using PCA and t-SNE.

reveal the outliers that laid afar from their groups. It is obvious that using just three principal components as illustrated in Figure 2a and Figure 2b can make a time series much easier to interpret. Even employing different dimensionality reduction techniques, we can easily spot the same outliers from the feature representation maps on Figure 2b-right (no highlight) at the same locations on Figure 2a-right (highlighted with small ellipses). Furthermore, the outputs of four other datasets in the same archive, namely, *ItalyPowerDemand*, *Wafer*, *CBF* and *ECG5000* were illustrated in Figure 3a-3d. As the figures show, several thousands of time series can be displayed simultaneously in the same plane. The unbalanced data, the data clusters, and anomalies can be easily identified.

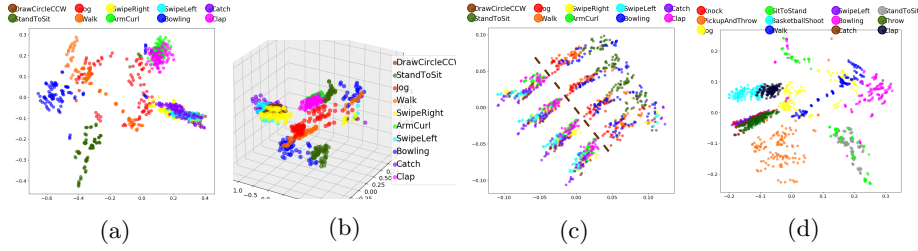


Fig. 5: Results of the dimensionality reduction of human movements in the UTD-MHAD. a) Ten actions using PCA followed by Kernel PCA. b) 3D projection of the same action set as (a) using Kernel PCA and PCA. c) The same action set as (a) using Kernel PCA with two kernels, RBF and polynomial, accordingly. The dashed line was drawn to separate another view. d) Twelve actions using PCA and Kernel PCA.

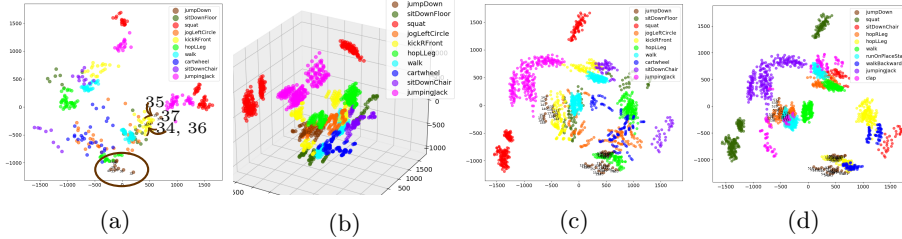


Fig. 6: Results of the dimensionality reduction of human movements in the HDM05. a) 446 number of actions from a default view using Kernel PCA and PCA. Three movements of “jumpDown” (●) of user “dg” (34, 35, 36) and one movement of user “mm” (37) are laid far away from the others (in the brown ellipse). b) Three-dimensional projection of 1784 trials from ten actions as (a) with additional movements of rotating subjects by 10, 20 and 30 degrees. c) Two-dimensional projection of (b). d) A similar effect in a different subset of motions of four viewpoints comparing to c) depicted in different colors from (d) → (c) as the following: “jumpDown” (●), “squat” (● → ●), “jumpingJack” (● → ●), “hopLLeg” (● → ●), “sitDownChair” (● → ●) and “walk” (● → ●).

4.2 Visualization Results of The Multivariate Time Series

For multivariate time series, the result of applying PCA followed by t-SNE, namely using $n = 2$ to *Japanese Vowel* is displayed in Figure 4a. Figure 4b is the output of adding a second t-SNE ($n = 3$) with smaller perplexity to Figure 4a. Generally, applying just two appropriate transformations should be sufficient to capture an important structure of the data, however in this case we prefer to have more compact clusters. Figure 4c displays two patterns of traffic data flow from the *NetworkFlow*. An inset in the figure shows a two-dimensional projection. The characteristics of the data are very different in large scale, that is the direction

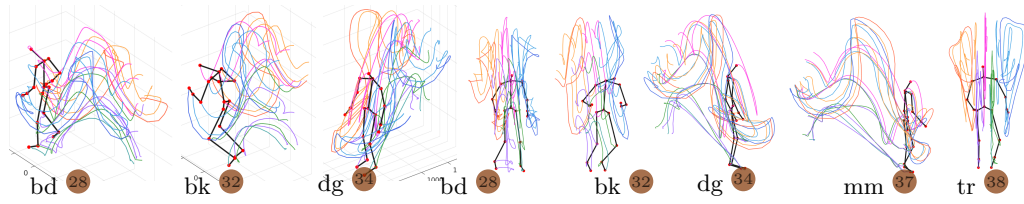


Fig. 7: Five subjects performs “JumpDown” (●) from the default view as seen in Figure 6a. From left to right: the first three images are the 3D projection of three subjects. The next five images are the simplified 2D projections in which the trajectories are much easier to be observed. The subjects “dg” and “mm” face 90° opposing to the direction of “bd”, “bk”, and “tr” causing different patterns of the trajectories and are considered as another group as seen in Figure 6a.

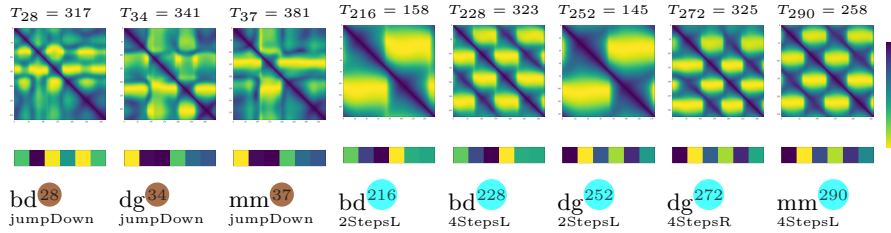


Fig. 8: A comparison of eight trials in the HDM05 between applying the unthresholded RP and our proposed algorithm with six principal components lying below each corresponding RP. The first three images on the left are from the action “jumpDown” (●) as depicted in Figure 7. The next five images are obtained from four types of “walking” (●). Notice the time length (T_i) where the repetitions of the walking cycle occur.

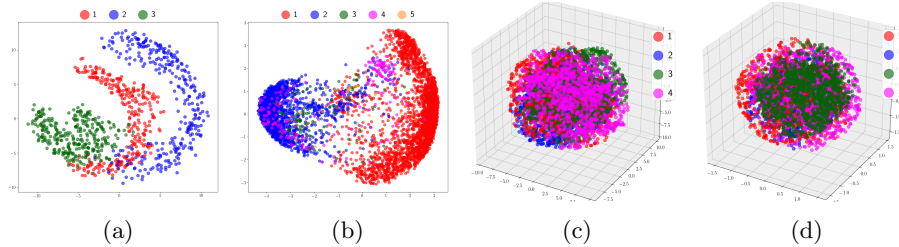


Fig. 9: Variations of transformation. a) “CBF” using MDS. b) “ECG5000” using Kernel PCA with polynomial kernel. c) “TwoPatterns” in the UCR archive using PCA. d) “TwoPatterns” using Kernel PCA with third-degree polynomial.



of the traffic is just either 1 or 0, whereas the payload sizes are about a few thousand of units. Nevertheless, our approach allows a user to easily spot an

outlier. An output of *Wafer* where the wafers were marked either defective (●) or not defect (●) from 1194 signals has been revealed in Figure 4d.

4.3 Visualization Results of The Motion Capture

The UTD-MHAD dataset. The outputs of applying the proposed method to the UTD-MHAD can be seen in Figure 5a–5d. Ten actions from Figure 5a are *DrawCircleCCW*, *StandToSit*, *Jog*, *Walk*, *SwipeRight*, *ArmCurl*, *SwipeLeft*, *Bowling*, *Catch*, and *Clap*. Four actions which involve only one arm movement, *SwipeLeft*, *SwipeRight*, *Catch*, and *DrawCircleCCW* are drawn very close to each other in two-dimensional space; while two-hand movements, *ArmCurl* and *Clap* plots are also close to one another. The actions concerning moving hands and feet such as *Jog* and *Walk* can be related to each other in the plot. The rest of the actions, *StandToSit* and *Bowling* are obviously distinct from other actions. When the transformation algorithms have been changed, this yields different patterns as found in Figure 5b-5c. It is not easy to interpret the data by simply inspecting Figure 5c without understanding the underlying patterns. There are two viewpoints to be used to interpret this figure: i) two groups of actions are separated by the dashed line and ii) four groups of actions which are perpendicular to the dashed line. The first view divides actions into two groups, one group involving actions with just “hands” moving lies on the left side of the dashed line and another group on the right involves actions in which both, “hands” and “legs” move. From the second viewpoint, the data can be seen as four clusters, as data lies perpendicular to the dashed line which is the consequence of rotating subjects around an axis by viewing the same action from four angles (0, 10, 20, and 30 degrees). As a result, these four groups look quite similar to each other but just shifted along the same axis. Moreover, the results of the transformation of twelve actions are displayed in Figure 5d. We take out three actions, *DrawCircleCCW*, *SwipeRight* and *ArmCurl* and add five actions, *Knock*, *PickupAndThrow*, *SitToStand*, *BasketballShoot*, and *Throw* to the plot. The action *StandToSit* now gets closer to the new action *SitToStand*, whereas the actions about one arm movement *Knock*, *Throw*, *SwipeLeft* and *Catch* lie close to each other. Some data of *Jog* and *Walk* lay close to each other, and the rest of actions such as *BasketballShoot*, *Clap*, *Bowling*, *PickupAndThrow* are properly clustered.

The HDM05 dataset. The results of applying similar techniques as to the UTD-MHAD to the **HDM05** can be found in Figure 6a-6d. Figure 6a shows 446 trials from ten actions using the default view (0°) of MoCap. The action *jumpDown* (●) has the minimum number of trials because it consists of only 13 trials, of which four trials are from the subject “bd”, three trials from the subject “dg” and another three trials from the subject “tr”, two trials from the subject “bk” and one trial from the subject “mm”. Notice that four trials of the *jumpDown* from the subject “dg” and “mm”, 34, 35, 36 and 37 lay far away from the others in the brown ellipse. When we looked closely to these particular sequences as depicted in Figure 7, we found that the subjects “dg” and “mm” have completely different trajectories compared to the subjects “bd”, “bk”, and “tr”. This is because these two subjects turn completely by 90 degrees difference

to the camera, or in perpendicular to the other three subjects “bd”, “bk”, and “tr”. This is the reason why those four trials of *jumpDown* are laid afar from the group. Next, when we slightly rotated these five subjects in all trials by 10, 20, and 30 degrees, the patterns of each action in these four viewpoints can be observed in Figure 6b- 6d. Figure 6b shows the first three principal components of these four views from the rotations (0° , 10° , 20° , 30°). After exchanging four of ten actions, while six actions are kept as depicted in Figure 6c and Figure 6d, **the shapes and distributions of these actions still remain the same in the subspaces**. Notice the distributions of *jumpDown*, *squat*, *jumpingJack*, *hopLLeg*, *sitDownChair* and *walk* in Figure 6c comparing to Figure 6d. Additionally, eight images in Figure 8 show the comparison of the visualization of *jumpDown* and *walk* between the unthresholded RPs and our feature representations. The RPs are symmetric matrices of size $T_i \times T_i$ but we fit them to the same image size for a comparison. By comparing our extracted features with six components which are laid below each corresponding RP, we can easily spot two distinct groups of action *jumpDown* and *walk*. The RP of action *jumpDown* from the subject “dg”  has a distinctive pattern other than the subject “mm” , whereas the extracted features from our method for both actions show about the same features which are in accordance to the trajectories illustrated in Figure 7. The action *walk* consists of four types of walking, “2StepsL”, “2StepsR”, “4StepsL”, and “4StepsR”. The number “2” and “4” indicate the walking steps, while the “L” and “R” indicate whether left or right leg starts. By comparing the RPs of *walk* in Figure 8, the repetitive actions such as two steps (2Steps) or four steps (4Steps) by our method yield similar fixed small features. The “4Steps” *walk* in the RP reveals two harmonic oscillations or two complete cycles, while “2Steps” shows one cycle of action. Furthermore, an output from the subject “bk” have a distinct feature but quite close to “bd”, while the features of “dg” and “mm” are about the same. These results comply with Figure 6, Figure 7 and Figure 8. The trivial changes e.g., either left leg (L) or right leg (R) starting first have no significance considering from the involved markers on one leg versus the markers on the whole body. The concept of principal components makes the results robust to noise and the trivial changes.

5 Discussion

5.1 Stability and Robustness

The current trends of time series and trajectory classification are to use deep networks by fine-tuning millions of parameters to achieve the best output performance. However, it lacks an explanation of why a particular signal fails. Furthermore, some outliers may lead to overfitting of the training data. Yet, our approach can complement this deficiency by offering a concise feature representation which can give data analyst an understanding of the underlining patterns. By projecting the extracted features onto a two- or three-dimensional space, it provides a visual representation for a data analyst to have an overview of a bunch of data simultaneously. The results are robust to small changes as seen in

Figure 6c, Figure 6d and Figure 8. Five images in Figure 8 show similar feature representations for different types of *walk* (and three images of *jumpDown*) from various subjects leading to the same direction. In addition, even though we employed different transformation techniques, we can still spot the same outliers in different visual representations as the results shown from “Plane” dataset.

5.2 Limitations and Variations of Transformation

Our proposed technique has a few limitations. First and foremost, the dimensionality reduction methods employed in our experiment were selected from several known techniques based on the assumption of linear subspace embedding and nonlinear manifold learning of the data. Therefore, we are not able to tell which technique is the best choice. For instance, the output of employing Multidimensional Scaling (MDS) on *CBF* (previously PCA was employed) now can be seen in Figure 9a. This new figure shows better separation of three data clusters comparing to Figure 3c. Changing the transformation of *ECG5000* from PCA (depicted in Figure 3d) to Kernel PCA in Figure 9b gives us an interesting alternative viewpoint. Nevertheless, we have had several cases of failure, for instance, *TwoPatterns* in the UCR archive which composes of 5000 sequences. The data clusters cannot be clustered in the way we had expected. The results are shown in Figure 9c-9d. Another failure was also found in case of visualizing the UTD-MHAD dataset when all the actions in a subset involved only hand gestures. This may be because the changes in the movement due to one arm gestures with three corresponding markers (of dimension 3×3) were considered insignificant when compared with the movement of a whole body consisting of 20 markers (of dimension 20×3). Considering PCA which we employed Singular Value Decomposition (SVD) for a matrix decomposition, for any given element i , X^i has an arbitrary size of $m \times T_i$. We may assume that the best rank r of the matrix X^i is the number of crucial time points c_1 . The matrix X^i is just a product of two matrices U and V where U is an $m \times c_1$ matrix expressing the weighted factor, and V is a $c_1 \times T_i$ matrix. We keep the most highest weighted factors in U and repeat the process for n times. Hence, the principal components c_n are to be in the final results.

6 Conclusion

In this paper, we have presented a new approach for time series and trajectory visualization by employing existing well-known non-time-series dimensionality reduction techniques. Our proposed methodology does not seek to make an interpretation of an individual signal nor to inspect the changes of data over time. Nonetheless, we can reveal some meaningful information such as the overview of data clusters. Moreover, outliers of each data class can be easily identified. By integrating this technique into a visual analytic pipeline in visualization tools, it can take the load off a data analyst in order to investigate any anomalies presented in the large data size. The datasets applied in the experiments were

selected from diverse sources to demonstrate and enforce the robustness of our proposed technique. This technique is not tailored to any particular data type, hence it can be integrated into any application domain. In addition, our approach is very simple to implement and lightweight as well as reproducible across different runs. Finally, it is to be noted that good clustering depends on the inter-relationship of the data structure and the correctly applied manifold learning method to achieve the optimum results.

References

1. Baydogan, M.G., Runger, G.: Time series representation and similarity based on local autopatterns. *Data Mining and Knowledge Discovery* **30**(2), 476–509 (Mar 2016). <https://doi.org/10.1007/s10618-015-0425-y>, <https://doi.org/10.1007/s10618-015-0425-y>
2. Chen, C., Jafari, R., Kehtarnavaz, N.: UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor, vol. 2015-December, pp. 168–172. IEEE Computer Society (12 2015). <https://doi.org/10.1109/ICIP.2015.7350781>, <http://www.utdallas.edu/~kehtar/UTD-MHAD.html>
3. Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G.: The UCR time series classification archive (2015), http://www.cs.ucr.edu/~eamonn/time_series_data/
4. 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**(1), 141–150 (Jan 2016). <https://doi.org/10.1109/TVCG.2015.2467553>
5. Karim, F., Majumdar, S., Darabi, H., Harford, S.: Multivariate lstm-fcns for time series classification (2018)
6. Luczak, M.: Combining raw and normalized data in multivariate time series classification with dynamic time warping. *Journal of Intelligent and Fuzzy Systems* **34**(1), 373–380 (2018). <https://doi.org/10.3233/JIFS-171393>
7. Martin, S., Quach, T.T.: Interactive visualization of multivariate time series data. In: Proceedings, Part II, of the 10th International Conference on Foundations of Augmented Cognition: Neuroergonomics and Operational Neuroscience - Volume 9744. pp. 322–332. Springer-Verlag New York, Inc., New York, NY, USA (2016)
8. Müller, M., Röder, T., Clausen, M., Eberhardt, B., Krüger, B., Weber, A.: Documentation mocap database hdm05. Tech. Rep. CG-2007-2, Universität Bonn (June 2007)
9. Nguyen, M., Purushotham, S., To, H., Shahabi, C.: m-tsne: A framework for visualizing high-dimensional multivariate time series. *CoRR* **abs/1708.07942** (2017)
10. Tanisaro, P., Heidemann, G.: An empirical study on bidirectional recurrent neural networks for human motion recognition. In: The 25th International Symposium on Temporal Representation and Reasoning (TIME) (2018)
11. Tanisaro, P., Lehman, C., Sütfeld, L., Pripa, G., Heidemann, G.: Classifying bio-inspired model in point-light human motion using echo state network. In: The 26th International Conference on Artificial Neural Networks (ICANN), 2017 (2017)
12. Tanisaro, P., Mahner, F., Heidemann, G.: Quasi view-independent human motion recognition in subspaces. In: Proceedings of 9th International Conference on Machine Learning and Computing (ICMLC) (2017), feb 24-26, 2017, Singapore