Aligning Manifolds of Double Pendulum Dynamics Under the Influence of Noise *

Fayeem Aziz,[†] Aaron S.W. Wong, James S. Welsh, and Stephan K. Chalup[‡]

School of Electrical Engineering and Computing
The University of Newcastle,
Callaghan NSW 2308, Australia

September 21, 2018

Abstract

This study presents the results of a series of simulation experiments that evaluate and compare four different manifold alignment methods under the influence of noise. The data was created by simulating the dynamics of two slightly different double pendulums in three-dimensional space. The method of semi-supervised feature-level manifold alignment using global distance resulted in the most convincing visualisations. However, the semi-supervised feature-level local alignment methods resulted in smaller alignment errors. These local alignment methods were also more robust to noise and faster than the other methods.

1 Introduction

Manifold alignment of two data sets assumes that they have similar underlying manifolds. The aim is to find a mapping between the two data sets so that the underlying manifold structure can be better recognised. Due to its generality manifold alignment has great potential to be useful in various domains. In the past it has been applied to facial expression analysis by image sequence alignment [6, 10, 17, 24, 26], graph matching [7], image classification [9, 25], and bioinformatics [21]. However, due to its high computational demands and the complexity of manifold data, manifold alignment still requires substantial further research and development to increase its impact in practical applications.

Real-world data is often sampled from a high-dimensional space and can be modelled as a set of points that lie on a low-dimensional non-linear manifold [12]. This also applies to data collected from robot kinematics or human motion which could be represented in form of dynamical system trajectories

---

[†]MdFayeemBin.Aziz@uon.edu.au

[‡]Stephan.Chalup@newcastle.edu.au

on low-dimensional manifolds [4]. It has been shown that motion data can be transferred from a human to a robot or between robots by aligning the corresponding manifolds [3, 5]. In this context, the ability of manifold learning and manifold alignment to represent non-linear data is critical [18]. An example that can illustrate this is a double pendulum and its non-linear dynamics [13].

The contribution of this study is an experimental comparison of several existing manifold alignment algorithms using data sampled from the motions of two simulated double pendulums in three-dimensions. Previous restricted pilot experiments, had focused on a pendulum in two dimensions [1]. The present paper also it investigates the stability of the methods under the influence of noise. It introduces the concepts of normalised distances along with pairwise correspondence measures [8]. The experimental evaluation focuses on visualisations and measuring the proximity of alignments and the execution time of the algorithms.
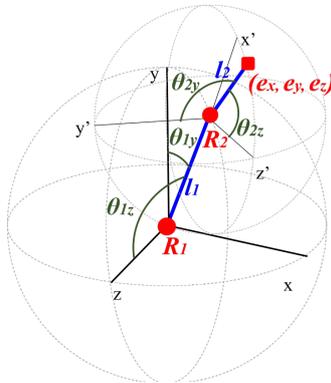
Figure 1: Double pendulum motion in 3D. x, y and z are the local axes of limb l_1 and x', y' and z' are the local axes of limb l_2. (e_x, e_y, e_z) is the end-effector.

2   Double Pendulum Data

The simulated data used in our experiments represents the motion of a three-dimensional double pendulum similar to a two-limb robot arm as shown in Figure 1. The two limbs are denoted l_1 and l_2 and the joints are denoted R_1 and R_2. One end of l_1 is fixed at R_1 and can rotate around this joint. The other end of l_1 is attached to l_2 at joint R_2. l_2 can rotate freely around joint R_2. The end point of the arm, that is, the end of l_2 not attached to R_2, is also known as the end-effector and has coordinates (e_x, e_y, e_z). The coordinate system follows the right-hand rule and has its origin at joint R_1. θ_{1y} and θ_{1z} are the angles of l_1 with the y and z axes, respectively, and $\theta_{2y'}$ and $\theta_{2z'}$ are the angles of l_2 with the y' and z' axes, respectively. The feature vector for each sample point was calculated from the kinematics at the joints and the end-effector coordinates were calculated using forward kinematics:

$$(e_x, e_y, e_z, \cos\theta_{1y}, \cos\theta_{1z}, \cos\theta_{2y'}, \cos\theta_{2z'}, \sin\theta_{1z}, \sin\theta_{1y}, \sin\theta_{2y'}, \sin\theta_{2z'}) \quad (1)$$

The data sets were generated from two similar double pendulums, which had different limb lengths and slightly different limb length ratios, where we restricted

the experiments to the case $l_2 < l_1$ (Figure 1):

Pendulum 1: $(l_2/l_1) = 0.75/1.25 = 0.60$

Pendulum 2: $(l_2/l_1) = 1.25/1.56 = 0.80$,

The data was acquired using increments of $30°$ on all four axes y, z, y' and z'. As a result, the number of instances was $(360/30)^4 = 20736$ and the size of each of these data sets was 20736×11. If the following sections refer to data sets X and Y it is assumed that the data is arranged in the form of matrices of dimension 20736×11. The correspondence subsets of X and Y comprise corresponding points in both sets that have the same or similar joint angles. They were selected by using $90°$ joint angle steps between two instances in the same data set. Therefore each of the two correspondence subsets had $(360/90)^4 = 256$ instances which was about 10% of the data set.

In the experiments, uniformly distributed white noise was added in two different ways to the data. First, noise was added to the joint angles and the noise range was incremented from $0°$ to $\pm10°$ in steps of $1°$. The second type of noise was added to the end effector coordinates. The range of this coordinate noise was increased from 0 to ±1.0 in steps of 0.1.

(a) Approach 1   (b) Approach 2
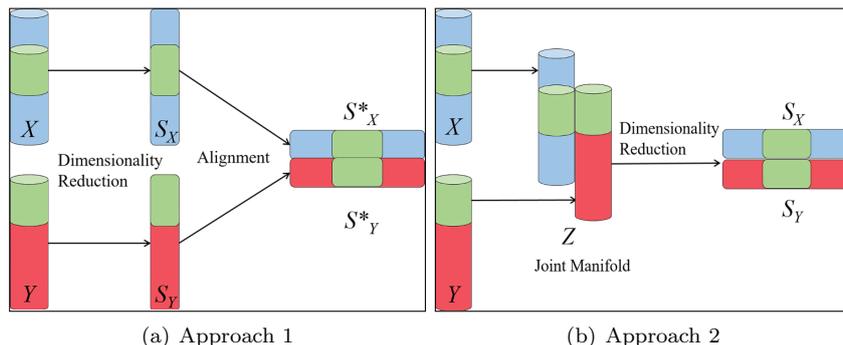
Figure 2: (a) Approach 1 : Dimensionality reduction is applied to X and Y, which yields low-dimensional manifolds S_X and S_Y, respectively. The manifolds are then mapped into a joint space as aligned manifolds S_X^* and S_Y^*. (b) Approach 2: A joint matrix Z is generated from X and Y by regularisation of correspondence information. Then, dimensionality reduction is applied to obtain the low-dimensional aligned manifolds S_X and S_Y. The green areas in both approaches indicate the correspondence subsets.

3   Manifold Alignment Methods

We address two general approaches to align manifolds. Each of them has two stages (Figure 2). In Approach 1 the data sets are first mapped into a low-dimensional space and then alignment is performed as described by Wang et al. [21]. In Approach 2, first a joint manifold is created to represent the union of the given manifolds and then the joint manifold is mapped to a lower dimensional latent space as described by Ham et al. [10]. Approach 2 was also described

as a generalised semi-supervised manifold alignment framework by Wang and Mahadevan [22], where the joint manifolds can be constructed using different characteristics of the data based on the application.

Wang and Mahadevan [22] further distinguished between two levels of manifold alignment: instance-level alignment (Inst) and feature-level alignment (Feat). For instance-level alignment, their method computes a non-linear low-dimensional embedding based on an alignment of the instances in the data. The embedding results in a direct matching of corresponding instances in alignment space. In contrast, feature-level alignment builds on mapping functions of features which map any associated instance or set of instances into the newly aligned domain.

The present study applied the following four methods to the double pendulum data sets. Method 1 followed Approach 1 ([21] and Figure 2 (a).) and methods 2-3 followed the semi-supervised Approach 2 ([22] and Figure 2 (b)). Further, each of the four methods had an instance-level and a feature-level version that was addressed in separate experiments (see overview in Figure 3):

- **Method 1:** Locality Preserving Projection (LPP) [11] is used for feature-level dimensionality reduction, and Laplacian eigenmaps (eigenmaps) [2] is used for instance-level dimensionality reduction, as the first stage, in two different experiments. In the second stage (Figure 2 (a)) we followed [21] and employed Procrustes analysis [14] to align the two data sets in low-dimensional space.

- **Method 2:** For semi-supervised manifold alignment preserving local geometry [22], first, the joint manifold Z was calculated using the graph Laplacians for X and Y. Then, eigenvalue decomposition of the joint manifold provided instance-level alignment and generalised eigenvalue decomposition provided feature-level alignment.

- **Method 3:** For semi-supervised manifold alignment using local weights [20] the intermediary joint manifold Z was formed by weights for X and Y, which were calculated for each set using a k-nearest neighbour graph and the heat kernel. Similar to Method 2 the low-dimensional manifolds were mapped by eigenvalue decomposition of joint matrices in two experimental components.

- **Method 4:** For semi-supervised manifold alignment preserving global geometry [23] the joint manifold Z was generated using the global distances of corresponding pairs in $X \cup Y$. The eigenvalue decomposition of Z provided the dimensionality reduction to obtain the aligned low-dimensional manifolds in the case of instance-level alignment. Generalised eigenvalue decomposition was used to reduce the dimensionality in the case of feature-level alignment.

4   Performance Evaluation

To measure the performance of the alignment methods, we assumed that all instances of one data set can be mapped one-to-one to instances in the other data set. The widths of the resulting manifolds originating from the same data
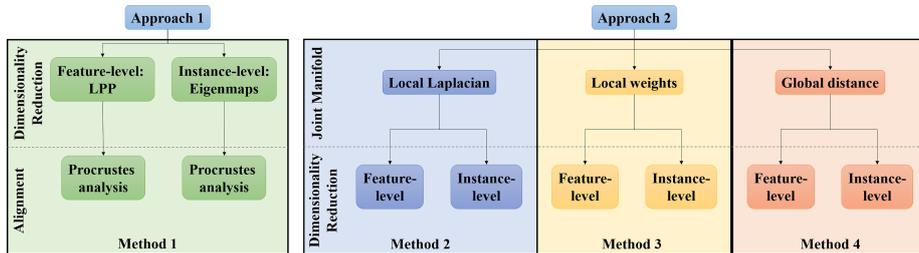
Figure 3: Overview of methods used in the experiments where the general structure of Approach 1 and 2 is detailed in Figure 2.

set could vary for the different methods. As a result of scaling, the distances between corresponding points could be larger between "wider" manifolds and smaller between "narrower" manifolds. To obtain comparable distance measurements between the two aligned low-dimensional manifolds, the distances were normalised by the maximum width of the two manifolds as follows:

$$D_i = \frac{\|S_X(i) - S_Y(i)\|}{\max_{\substack{j=1,\dots,n \\ k=1,\dots,n}}(\|S_X(j) - S_X(k)\|, \|S_Y(j) - S_Y(k)\|)} \tag{2}$$

where $S_X(i)$ and $S_Y(i)$ are corresponding points for $i = 1, ..., n$ and n is the number of points that represent each of the manifolds. The matching errors of the alignments were measured using the average of the D_i, denoted by Δ. That is, Δ indicates the closeness of the aligned manifolds in low-dimensional space. The standard deviation of the D_i, denoted by σ, represents the consistency or smoothness of the alignment where a smaller σ indicates a smoother alignment. Δ and σ could be used to measure the quality of an alignment if it was successful.

5 Results

Both limbs of the double pendulum rotated freely in three dimensions. The resulting manifolds were too complex to be visualised in full in a three-dimensional graph. Therefore, we used snapshots of $90°$ steps for the motion of limb l_1 and of $10°$ steps for limb l_2. Figure 4 shows that feature-level manifold alignment using global distance resulted in six small spheres that were distributed regularly on a bigger sphere. The small spheres represent the motion of limb l_2 and the bigger sphere represents the motion of limb l_1. Ergo this figure can be interpreted as sections in time of the expected shape of the aligned manifolds of pendulum motion. The results from the other three methods for feature level alignment show complex shapes that include partially collapsed submanifolds. For the instance-level cases the manifolds collapsed completely into line segments.

Table 1 shows results for the feature-level cases where Δ and σ were lowest for Method 3. The execution time of each method for same datasets is shown in the last column of 1. Due to the involvement of high dimensional matrix multiplication, the manifold alignment using global distance required a significantly higher execution time than the others. Figures 5 and 6 show the aligned manifolds in the feature level cases after application of joint angle and coordinate noise, respectively. Both limbs of the double pendulums were rotating

Procrustes analysis		Local Laplacian		Local weight		Global distance	
Feat	Inst	Feat	Inst	Feat	Inst	Feat	Inst

Figure 4: 3D motion manifold alignments using snapshots in 90° steps for one limb and 10° steps for the other limb. The output of the feature-level manifold alignment using global distance (column 7) displays six small spheres representing the motion of l_2 which are distributed on a larger sphere representing the motion of l_1.

Table 1: Summary of the performance of manifold alignment methods: Feature-level manifold alignment using local weights achieved the lowest Δ and σ (3rd row).

	Method	Level	Δ	σ	Time(s)
1	Procrustes analysis	Feature	1.37×10^{-05}	6.20×10^{-06}	393
2	Local Laplacian	Feature	1.22×10^{-07}	5.08×10^{-08}	382
3	**Local weights**	**Feature**	$\mathbf{1.22 \times 10^{-07}}$	$\mathbf{4.92 \times 10^{-08}}$	**343**
4	Global distance	Feature	2.76×10^{-06}	7.38×10^{-07}	6.3×10^4

in three-dimensional spherical motion. Ergo the overall motion manifold can be described as the cross product of two spheres (as indicated in column 7 of Figure 4). The visualisations of the outcome of manifold alignment preserving global geometry are shown in column 4 of Figures 5 and 6. The results seemed to be robust to noise and the visualisation were as expected. On the other hand, manifold alignments using the local Laplacian and local weights resulted in lower Δ and σ (Figure 7) but visually more complex outcomes that also significantly varied under the influence of noise (Figures 5 and 6).

6   Discussion and Conclusion

This study compared four different manifold alignment methods (Figure 3). The data sets were generated by simulating the motion of two three-dimensional double pendulums that differed in the lengths of their arms. Where the joint angles of the two pendulums were equal, they were recorded as corresponding pairs, and this was exploited in the manifold alignment and performance evaluation.

We also investigated the effects of two different types of noise on the alignment of these data sets. First, random noise was added to the joint angles to observe the stability of the alignment with respect to actuator irregularities. Then, random noise was added to the end effector coordinates to observe the stability of the alignment methods with respect to jittery motions of the arm or noise in the data recordings. The performance of these methods was evaluated quantitatively by measuring the proximity of the corresponding points and

Noise level	Method 1: Procrustes analysis	Method 2: Local Laplacian	Method 3: Local weights	Method 4: Global distance
$\pm 0°$				
$\pm 1°$				
$\pm 2°$				
$\pm 3°$				
$\pm 4°$				
$\pm 5°$				
$\pm 6°$				
$\pm 7°$				
$\pm 8°$				
$\pm 9°$				
$\pm 10°$				
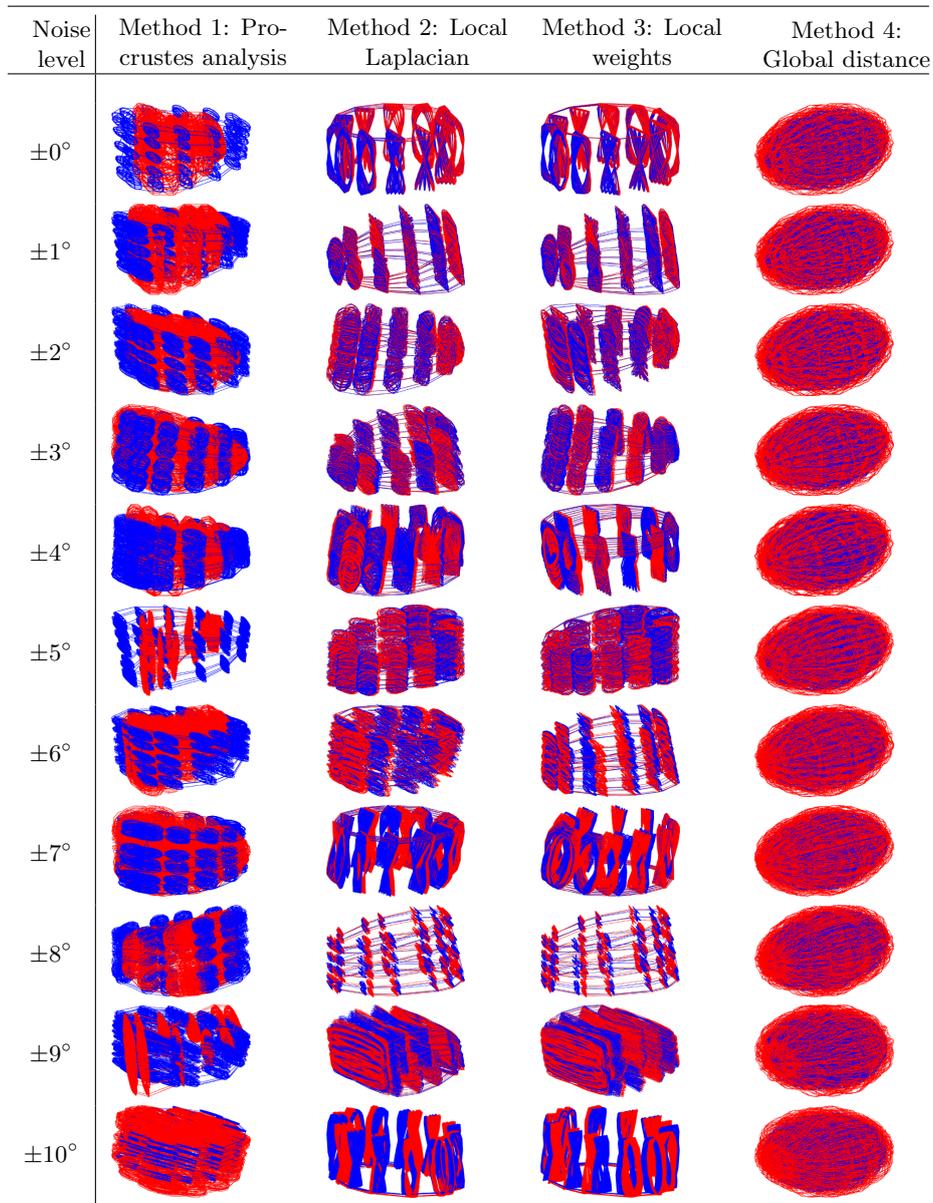
Figure 5: Feature-level alignment of manifolds under the influence of different levels of joint angle noise: Each graph visualises a different way of aligning the two pendulum manifolds. Each row shows the results for a different level of joint angle noise.

qualitatively using visualisations. The experiments were conducted in Matlab 2016b on a high-performance computer cluster equipped with Xenon CPUs and 500GB RAM where all experiments for this paper could be executed in a total running time of about eighteen days. Individual running times are reported in Table 1.

The approach of alignment preserving global geometry produced the most
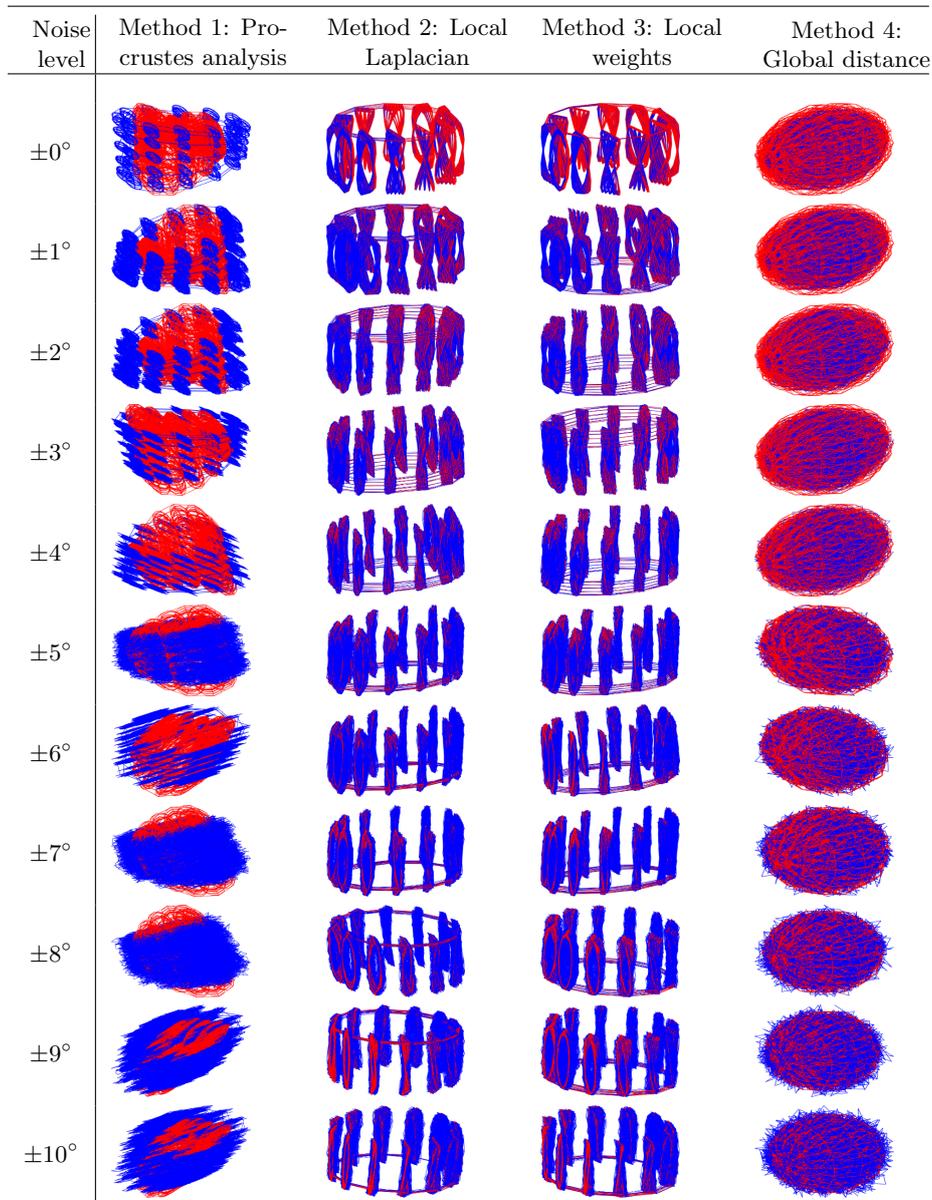
Figure 6: Feature-level alignment of manifolds under the influence of different levels of coordinate noise: Each graph visualises a different way of aligning the two pendulum manifolds. Each row shows the results for a different level of coordinate noise.

convincing visualisations but was also much slower than the other methods. The methods using local weights resulted in the numerically smallest alignment errors (Table 1). One possible interpretation is that local methods had an advantage because the experiments were limited to data from two similarly configured pendulums. Another interpretation is that parts of the manifolds collapsed, i.e., they were projected onto line segments during the manifold lear-
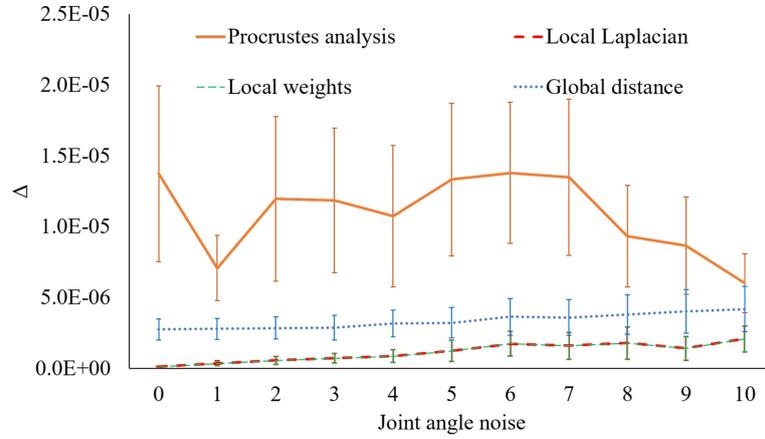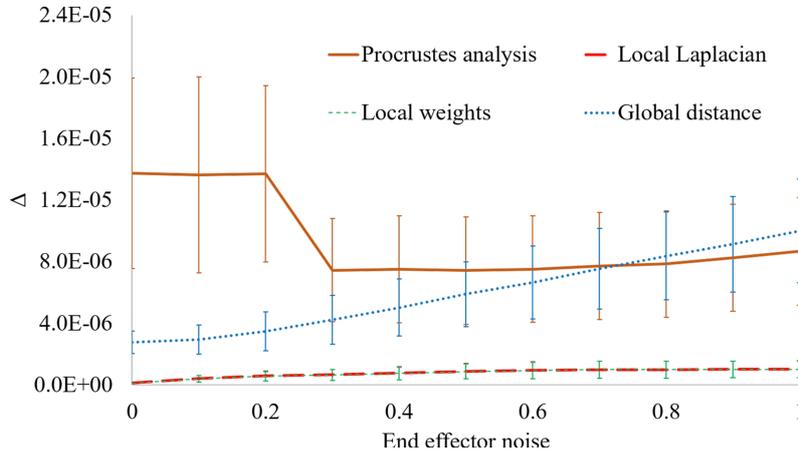
Figure 7: Performance graphs of manifold alignment methods that align the pendulum data with additional (a) joint angle noise and (b) end effector noise. The x-axes show the noise range. The y-axes show the averages Δ where the error bars are the standard deviations σ of the correspondence distances of the aligned manifolds. Δ and σ for manifold alignment using local weights and local Laplacian are lower than for the other methods and this result is not much affected by any of the added noise.

ning process. This is supported by the visualisations in Figures 4, 5 and 6. The observed instabilities of the local and instance level methods require further investigation.

Traditional robotic motion control engines often rely on inverse kinematics and can be difficult to adapt when the robot configuration changes [15]. Chalodhorn and Rao found that direct use of kinematics data from a human motion capture system to replicate human movement in a robot can result in

dynamically unstable motion [5]. This instability is the result of differences in the degrees-of-freedom and inadequate physical parameters for a robot to match human motion. The outcomes of the present study support the hypothesis that manifold alignment could provide a mapping between motion trajectories of similar manifolds. Moreover, a feature-level alignment approach could align trajectories on manifolds that may include out-of-sample extensions that may be generated during changes of configurations.

One of the purposes of dimensionality reduction is to process data faster while requiring less memory than the calculations in high dimensions. Feature-level alignment provides a function that can map instances from high-dimensional feature space to low-dimensional alignment space. This function can map any new sample, whether it was included in the training data set or not, to the aligned manifolds. In contrast, the instance-level methods require a complete recalculation of the alignment mapping for each new sample. Therefore, feature-level alignment can perform better than instance-level alignment when out-of-sample extension processing is necessary [19]. The latter could be the case, for example, when the robot configuration changes.

In future research, the findings of this study could possibly become useful for the task of motion imitation [16]. Using manifold alignment transfer learning between robot arms could be achieved by copying previously trained stable motions from one robot to another where the robots' arms could have different limb length proportions.

Acknowledgements

References

[1] Fayeem Aziz, Aaron S. W. Wong, James Welsh, and Stephan K. Chalup. Performance comparison of manifold alignment methods applied to pendulum dynamics. In *Proceedings of the Applied Informatics and Technology Innovation Conference.* (in press) Springer, 2016.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[3] Botond Bocsi, Lehel Csato, and Jan Peters. Alignment-based transfer learning for robot models. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2013.

[4] Rawichote Chalodhorn, David B. Grimes, Keith Grochow, and Rajesh P. N. Rao. Learning to walk through imitation. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2084–2090, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[5] Rawichote Chalodhorn and RajeshP N. Rao. *Learning to imitate human actions through eigenposes*, volume 264 of *Studies in Computational Intelligence*, book section 15, pages 357–381. Springer Berlin Heidelberg, 2010.

[6] Z. Cui, S. Shan, H. Zhang, S. Lao, and X. Chen. Image sets alignment for video-based face recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2626–2633, June 2012.

[7] F. Escolano, E. Hancock, and M. Lozano. Graph matching through entropic manifold alignment. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2417–2424, 2011.

[8] Ke Fan, Ajmal Mian, Wanquan Liu, and Ling Li. Unsupervised manifold alignment using soft-assign technique. *Machine Vision and Applications*, 27(6):929–942, 2016.

[9] Ricardo Guerrero, Christian Ledig, and Daniel Rueckert. *Manifold alignment and transfer learning for classification of Alzheimer's disease*, pages 77–84. Springer International Publishing, Cham, 2014.

[10] Jihun Ham, Daniel Lee, and Lawrence Saul. Semisupervised alignment of manifolds. In *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence*, volume 10, pages 120–127. AISTATS, 2005.

[11] Xiaofei He and Partha Niyogi. Locality preserving projections. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, pages 153–160. MIT Press, 2004.

[12] D. Huang, Z. Yi, and X. Pu. Manifold-based learning and synthesis. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(3):592–606, June 2009.

[13] Jakob Søndergaard Jensen. Non-linear dynamics of the follower-loaded double pendulum with added support-excitation. *Journal of Sound and Vibration*, 215(1):125–142, 1998.

[14] Bin Luo and E. R. Hancock. Feature matching with Procrustes alignment and graph editing. In *Image Processing And Its Applications, 1999. Seventh International Conference on (Conf. Publ. No. 465)*, volume 1, pages 72–76, July 1999.

[15] Amir Mosavi and Annamaria Varkonyi. Learning in robotics. *International Journal of Computer Applications*, 157(1):0975–8887, 2017.

[16] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, October 2010.

[17] Yuru Pei, Fengchun Huang, Fuhao Shi, and Hongbin Zha. Unsupervised image matching based on manifold alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1658–1664, August 2012.

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

[19] Chang Wang. *A geometric framework for transfer learning using manifold alignment*. Ph.d. thesis, Department of Computer Science, University of Massachusetts Amherst, UMass Amherst, September 2010.

[20] Chang Wang, Peter Krafft, and Sridhar Mahadevan. *Manifold alignment*, chapter Manifold alignment, pages 95–120. CRC Press, December 2011.

[21] Chang Wang and Sridhar Mahadevan. Manifold alignment using procrustes analysis. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1120–1127, New York, NY, USA, 2008. ACM.

[22] Chang Wang and Sridhar Mahadevan. A general framework for manifold alignment. In *AAAI Fall Symposium: Manifold Learning and Its Applications*, pages 79–86. AAAI Press, 2009.

[23] Chang Wang and Sridhar Mahadevan. Manifold alignment preserving global geometry. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1743–1749. AAAI Press, 2013.

[24] Xianwang Wang and Ruigang Yang. Learning 3D shape from a single facial image via non-linear manifold embedding and alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010*, pages 414–421, June 2010.

[25] Hsiuhan L. Yang and Melba M. Crawford. Manifold alignment for multi-temporal hyperspectral image classification. In *2011 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 4332–4335, July 2011.

[26] Deming Zhai, Bo Li, Hong Chang, Shiguang Shan, Xilin Chen, and Wen Gao. Manifold alignment via corresponding projections. In *Proceedings of the British Machine Vision Conference*, pages 1–11. BMVA Press, 2010.