

# Accurate Online Alignment of Human Motor Performances

Felix Hülsmann

Computer Graphics Group & Social Cognitive Systems  
Bielefeld University

Stefan Kopp

Social Cognitive Systems  
Bielefeld University

Andreas Richter

Computer Graphics Group  
Bielefeld University

Mario Botsch

Computer Graphics Group  
Bielefeld University

## ABSTRACT

Many approaches for motion processing or motion analysis employ Dynamic Time Warping (DTW) for temporally aligning an input movement with a reference movement. DTW, however, does not work online since it requires the *complete* input trajectory. Its online extension Open-End DTW can lead to poor alignments. In this paper we propose Weight-Optimized Open-End DTW, which combines path-length weighting and joint weights optimized from training data. We demonstrate our method to work online and to outperform Open-End DTW in terms of alignment quality.

## CCS CONCEPTS

• **Computing methodologies** → *Motion processing*;

## KEYWORDS

motion alignment, Dynamic Time Warping, Open-End DTW

### ACM Reference format:

Felix Hülsmann, Andreas Richter, Stefan Kopp, and Mario Botsch. 2017. Accurate Online Alignment of Human Motor Performances. In *Proceedings of MiG '17, Barcelona, Spain, November 8–10, 2017*, 6 pages. <https://doi.org/10.1145/3136457.3136470>

## 1 INTRODUCTION

Online algorithms for motion analysis and synthesis become highly important, as applications, such as virtual coaching environments, gain more and more popularity. However, numerous state-of-the-art algorithms for movement analysis and synthesis were originally developed to work offline. Many of them require a temporal alignment of an input motion with a reference trajectory as preliminary step [Giese and Poggio 2000; Krüger et al. 2017; Min and Chai 2012]. This alignment is frequently achieved via Dynamic Time Warping (DTW). Even though DTW can be prone to outliers and noise [Vlachos et al. 2002], DTW and its extensions provide compelling results in various applications. Unfortunately, DTW needs a whole trajectory to be completed before it can start calculating the optimal alignment. Thus algorithms that rely on DTW can only provide results as soon as the input motion has been completed.

---

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

MiG '17, November 8–10, 2017, Barcelona, Spain

© 2017 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5541-4/17/11.

<https://doi.org/10.1145/3136457.3136470>

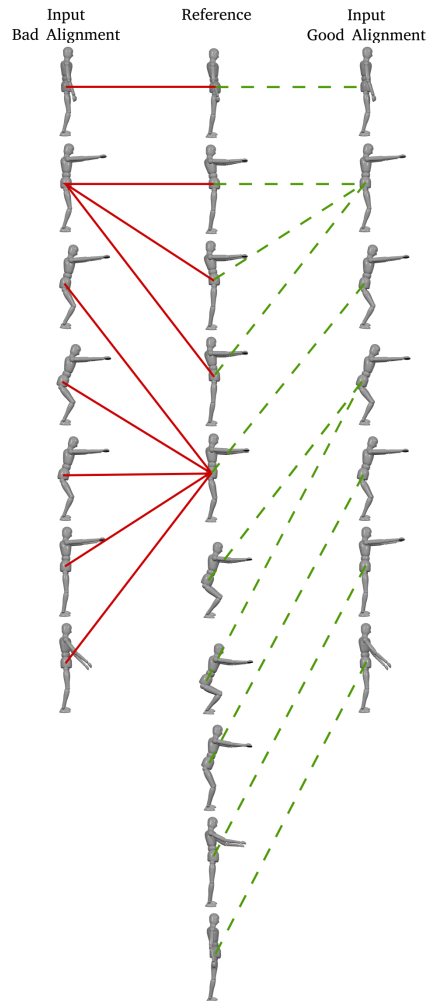
If the alignment could be estimated earlier, ideally directly after an input frame is observed, such algorithms could already provide results online during the performance.

An extension of DTW, Open-End DTW (OE-DTW) has been shown to work for the alignment of trajectories in online scenarios [Tormene et al. 2009], but unfortunately it can perform much worse than its offline counterpart. This is crucial: If the alignment fails, e.g., the algorithm decides that the final frame of an incomplete input motion still matches an early frame of the reference motion (cf. Figure 1), the whole alignment can become useless. In this case, all further steps that build on the aligned trajectories, such as motion classification, might fail.

In this work, we extend OE-DTW by path-length weighting together with joint weights based on evolutionary optimization to improve the alignment. We call the resulting algorithm Weight-Optimized Open-End DTW (WOOE-DTW). Despite the fact that there is a large number of related work on DTW and Open-End DTW, we are the first who combine path-length weighting with evolutionary optimized joint weights in favor of an improved alignment performance of Open-End DTW. Joint weights already appear in the literature, however, these are often engineered based on prior knowledge of the movement of interest or are based on heuristics. We propose a data-driven optimization-based approach that can, additionally to the improved online DTW performance, even uncover insights on the movement of interest via the estimated joint weights. We demonstrate that for our test scenario, which contains recordings of 95 body-weight squats by 49 subjects, our algorithm WOOE-DTW clearly enhances the alignment performance compared to OE-DTW.

## 2 RELATED WORK

One way to improve the performance of OE-DTW is to weight individual features in the error function. This weighting can reduce noise induced by unimportant features: If, for instance, two squat movements have to be aligned, any impact of, e.g., the rotation of the wrist should be minimized, since the rotation of the wrist has nothing to do with the performance of the squat. If this joint influences the alignment, it can only induce noise. Jeong et al. propose an approach to prevent DTW from aligning frames that belong to different phases of a repetitive movement. They introduce a penalty based on the temporal location of a frame. For the alignment of motor performances we mainly focus on the online analysis of the most recent single motor action. Thus we might not suffer from repetitive movement, for which penalties as described in [Jeong et al. 2011] would be wise to use. Other approaches introduce joint



**Figure 1: Bad and good alignment of a squat performance (left and right) with a reference movement (middle). Note that the bad alignment estimates an early frame of the reference as the end frame. Nearly the whole input movement is mapped on one single reference frame. Using this alignment to warp the input into the timing of the reference would result in an incomplete movement. If the good correspondence would be used, the result would be a similar movement than the reference squat in time, but with the spatial properties of the input trajectory.**

weights based on inter- as well as intra-class variability of the gesture of interest [Reyes et al. 2011]. In their evaluation, the weighted DTW improved the classification quality for gesture recognition. Arici et al. [2014] and Celebi et al. [2013] follow a similar approach. Weights for each joint are calculated for each gesture class of interest. The basic idea is to capture the contribution of each joint to the specific gesture. This contribution is quantified via the total displacement of the joint during the performance of a trained user. Furthermore, an additional meta parameter inside the weighting term is calculated based on maximization of a discriminant ratio

with respect to different gesture classes. The authors’ extensions of DTW increase the performance of gesture classification for their test cases. This approach is extended in [Choi et al. 2015]: The authors introduce additional dynamic weights which are able to change over time. To summarize, related approaches to feature weighting mainly focus on the overall movement or the variance of a feature. This prevents to account for important joints whose movement is rather small. Additionally, unimportant joints that move mostly non-functional could be higher ranked than the important ones. Instead of variance-based weights, we propose an approach that uses a suitable optimization of DTW weights. To this end, we introduce an error measure for DTW alignments and use it to optimize weights.

Another problem of DTW is its bias against temporal shifts in the warping function [Anguera and Ferrarons 2013; Dixon 2005; Sakoe and Chiba 1978]. This bias is normally unwanted. As a solution, some related approaches propose specific penalties that reward temporal shifts [Dixon 2005]. Unfortunately, this has a major drawback: Even when a warp without any shift would be correct, the penalty could induce the algorithm to prefer a different mapping. To make DTW independent from assumptions on the movements’ timing, we apply path-length weighting [Anguera and Ferrarons 2013; Muscariello et al. 2009]. This approach slightly increases the computational effort, but avoids the bias of DTW. In contrast to Anguera and Ferrarons [2013], we do not privilege specific directions of the warping path. Furthermore, we weight each feature on the whole temporal axis equally, whereas in [Anguera and Ferrarons 2013] later frames implicitly gain more weight for DTW. In favor of performing path-length weighting, Muscariello et al. [2009] introduce a specific weight matrix to store local path length weights as well as a matrix that stores possible warping path lengths. In our approach, we only require one additional matrix that stores the paths’ lengths.

### 3 SCENARIO AND DATASET

We use the body-weight squat as a test movement as it is frequently used in sports training [Escamilla 2001] and rehabilitation [Bailey et al. 2011; Escamilla 2001]. The squat involves nearly the whole body to keep balance and is therefore a highly complex movement. Moreover, it can be performed in different styles and thus expresses a high variety of motion.

The input data for most algorithms that deal with motion data is a sequence, called *trajectory*, of features which describe the movement of the human body. The features can be, for instance, joint angles. A trajectory of sports movements can be divided into the performance of single exercises, also called *motor actions*. Motor actions can be divided into *movement segments* (for the squat: “preparation”, “going down”, “is down”, “going up” and “wrap-up”). Our data is annotated with respect to the current movement segment. It consists of 95 annotated squat performances, recorded from 49 subjects.

We use a skeleton-based approach to represent the motion capture data: Kinematic features of 19 joints are recorded at 120 Hz with a 10 camera Prime 13W OptiTrack system. The skeleton hierarchy originates at the hips. Each single motion capture frame consists of  $k = 19$  joint rotations, encoded as unit quaternions. The orientation of the root  $\mathbf{q}_1$  – the hips – denotes the rotation of the root with respect to the beginning of the motor action. Each other

quaternion  $\mathbf{q}_2, \dots, \mathbf{q}_k$  denotes the rotation of the specific joint with respect to its parent. By using only rotations and not positions, we reduce the impact of subjects' different body proportions.

For all our optimization steps, we use cross-validation (CV) with 5 folds. Our plots contain averaged results. We ensured that no data from any recorded subject contained in a specific training set is contained in the corresponding test set. This enables us to test the generalization to new subjects, which is especially crucial as performances can vary much between subjects. A leave-one-subject-out protocol would also lead to reasonable results (similar to CV, the test error of each sample contributes to the overall test error), but would increase the training time.

#### 4 ONLINE TEMPORAL ALIGNMENT

Dynamic Time Warping (DTW) establishes a frame-to-frame correspondence between two trajectories. This is achieved via finding the optimal warping path between both, according to an integrated per-frame distance measure. The following subsection describes the standard DTW algorithm [Müller 2007, p. 69].

Let  $T_1$  and  $T_2$  be motion capture trajectories.  $|T_a|$  denotes the number of frames in the trajectory. To establish the correspondence between  $T_1$  and  $T_2$ , DTW first calculates the  $|T_1| \times |T_2|$  per-frame distance matrix  $\mathbf{M}$ . Each element  $\mathbf{M}(i, j)$  contains the distance between frame  $i$  of  $T_1$  and frame  $j$  of  $T_2$ . We define this distance as the summed distance between all quaternions  $\mathbf{q}_1, \dots, \mathbf{q}_k$  of these frames. As quaternion distance, we use the inner product as evaluated in [Huynh 2009] and construct each entry in  $\mathbf{M}$  as follows:

$$\mathbf{M}(i, j) = \sum_{d=1}^k (1 - |\mathbf{q}_{i,d} \cdot \mathbf{q}_{j,d}|). \quad (1)$$

DTW now finds the alignment path with minimal costs from start  $\mathbf{M}(1, 1)$  to end  $\mathbf{M}(|T_1|, |T_2|)$  through the matrix based on dynamic programming. First, one calculates the  $(|T_1| + 1) \times (|T_2| + 1)$  matrix  $\mathbf{D}$  which accumulates the minimal costs on possible paths.  $\mathbf{D}(i, j)$  in the accumulated cost matrix corresponds to  $\mathbf{M}(i - 1, j - 1)$  in the local cost matrix.  $\mathbf{D}$  is initialized as follows:

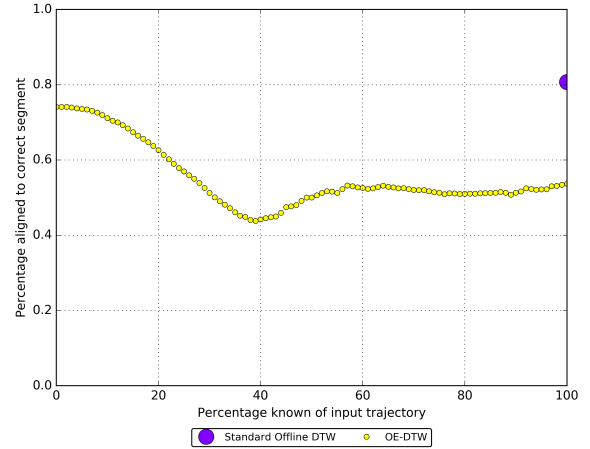
$$\mathbf{D}(i, j) = \begin{cases} 0, & \text{if } i = 1 \text{ and } j = 1 \\ \infty, & \text{otherwise} \end{cases}.$$

The entries of  $\mathbf{D}$  are iteratively updated as follows:

$$\mathbf{D}(i, j) = \mathbf{M}(i - 1, j - 1) + \min\{\mathbf{D}(i - 1, j - 1), \mathbf{D}(i - 1, j), \mathbf{D}(i, j - 1)\}. \quad (2)$$

for  $2 \leq i \leq |T_1| + 1, 2 \leq j \leq |T_2| + 1$ . The alignment path is traced back via minimizing the accumulated error in each step, starting from  $\mathbf{D}(|T_1| + 1, |T_2| + 1)$  [Müller 2007, p. 73]. To warp trajectory  $T_1$  to the timing of  $T_2$ , we select the corresponding frame in  $T_1$  according to the calculated alignment for each frame in  $T_2$ .

As DTW needs two complete trajectories to calculate an alignment it cannot work online. This can be bypassed by using Open-End DTW (OE-DTW) [Tormene et al. 2009]. OE-DTW allows to align a prefix  $T_1$  of a query trajectory with a complete reference trajectory  $T_2$ . It yields a warp as well as an estimation of which frame in the reference matches the last frame of  $T_1$ . Thus, the backtracing step in OE-DTW does not start from  $\mathbf{D}(|T_1| + 1, |T_2| + 1)$ , but from



**Figure 2: Comparison of the alignment quality of standard DTW and OE-DTW averaged over all folds. Standard DTW only provides results as soon as the whole input trajectory is known, OE-DTW can already provide alignments earlier.**

$\mathbf{D}(|T_1| + 1, \Omega)$ , with

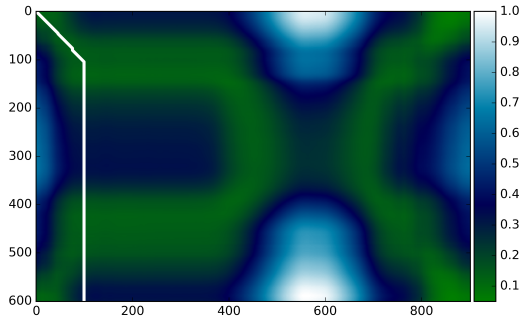
$$\Omega = \underset{j}{\operatorname{argmin}} \mathbf{D}(|T_1| + 1, j), \text{ where } 2 \leq j \leq |T_2| + 1.$$

$\Omega - 1$  is the frame in the reference trajectory that matches the last frame of the incomplete trajectory  $T_1$ . To calculate OE-DTW for a new incoming motion frame, we only have to update the last row of the cost matrix  $\mathbf{M}$  as well as the last row of the accumulated cost matrix  $\mathbf{D}$ . For implementation reasons, we start with the calculation as soon as the input motion consists of at least three frames.

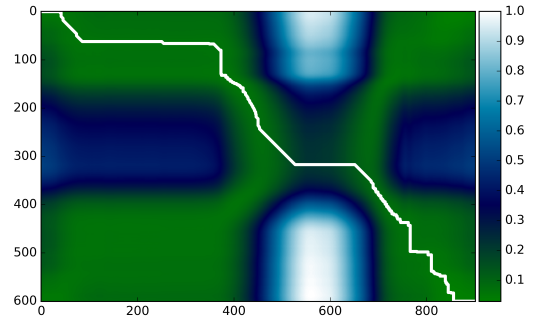
We compare the alignment quality of OE-DTW to the quality of the offline alignment. To estimate the alignment error, we make use of the annotated movement segments (“preparation”, “going down”, “is down”, “going up” and “wrap-up”): When  $T_1$  is warped to the timing of the reference  $T_2$ , we count the number of frame pairs in the warping path that belong to the same movement segment. We consequently define the alignment quality as the percentage of correctly aligned frame pairs. Figure 2 contains the results of the comparison between standard offline DTW and OE-DTW. We observe that the alignment quality of OE-DTW is inferior to the one by standard offline DTW. In some cases, the alignment even becomes completely degenerated: Starting from one specific frame of the input trajectory, all further frames are warped to the same frame of the reference trajectory. Such an alignment is visualized in Figure 3(a). In the following, we describe and evaluate our proposed extensions to improve the alignment performance.

##### 4.1 Path-length Weighting

The formulation of the accumulated cost matrix is biased towards shorter paths [Anguera and Ferrarons 2013; Dixon 2005; Sakoe and Chiba 1978]: The shorter the path, the smaller the accumulated error. This bias would make sense, if shorter paths would represent – in general – better alignments than longer ones. However, this is not the case: Let us consider the alignment of two performances of the same motor action that are similar in the spatial domain, but differ



(a) Failure of OE-DTW alignment: After a certain early input frame, all remaining input frames are matched to the same reference frame.



(b) WOOE-DTW significantly improves the alignment quality for the same two trajectories.

**Figure 3: Comparison of exemplary alignments based on OE-DTW and WOOE-DTW. The image displays the local cost matrix  $M$  together with the alignment path estimated by the two DTW variants. Both cost matrices are normalized to the same interval. The axis' labels indicate the frame numbers of the corresponding trajectories.**

in timing. One of these trajectories is performed with a specific speed. Now, the more similar the speed of the other trajectory is, the shorter is the alignment path for the standard DTW: It would stay mainly on the diagonal of  $D$ . However, if the performance is paused, e.g., because the performing subject has to think about how to continue, an optimal alignment path must leave the diagonal to account for the change in timing. If we allow the algorithm to prefer shorter paths, which means preferring less deviation in timing, it would tend to stay on the diagonal. To make DTW independent from assumptions on the movements' timing, we apply path-length weighting via adapting Equation (2) as follows:

$$D(i, j) = M(i - 1, j - 1) + D \left( \underset{(k, l)}{\operatorname{argmin}} \left( \frac{D(k, l) + M(i - 1, j - 1)}{L(k, l) + 1} \right) \right),$$

where  $(k, l) \in \{(i - 1, j - 1), (i - 1, j), (i, j - 1)\}$ .

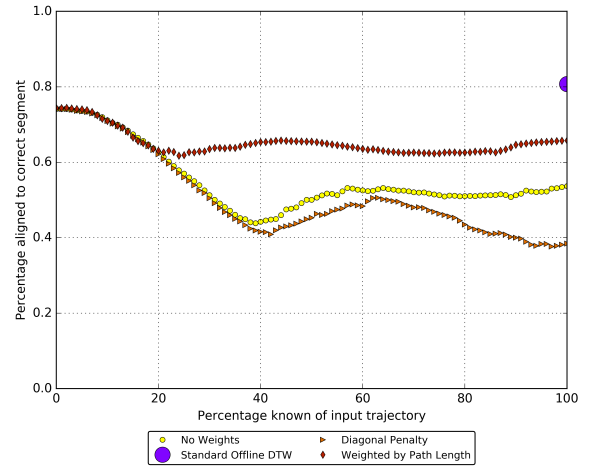
(3)

Matrix  $L$  contains the path-lengths of each optimal path. It is updated together with  $D(i, j)$  based on the just calculated values for  $k$  and  $l$ . After calculating  $D$  and  $L$ , we determine the optimal path via backtracing from  $D(|T_1| + 1, \Omega)$ . In each step, we divide all examined cells of the accumulated cost matrix  $D$  by their corresponding path-lengths from  $L$  and select the one with the smallest result.

Figure 4 displays the alignment quality of OE-DTW with path-length weighting compared to standard OE-DTW. Additionally, we calculate the alignment when extending OE-DTW with a diagonal penalty [Dixon 2005]. We observe a positive impact of path-length weighting on the alignment. When removing the weights or when using a penalty factor for diagonal steps the accuracy decreases.

## 4.2 Evolutionary-weighted DTW

DTW uses an equal weighting for all joints in Equation (1). However, for a given motor action, certain joints are more important for the alignment than others: A motor action such as the squat, for instance, mainly depends on the motion of the legs and the upper arms. For DTW, non-functional motion in other joints, such as the wrists, has the same impact on the alignment as these important joints. Thus, if the motion in the legs is only minimal, but the



**Figure 4: Impact of path-length weighting on the average alignment quality of OE-DTW.**

wrists move a lot, they would dominate the alignment, although an optimal alignment would prefer a simultaneous motion in the legs. Intuitively, one would thus increase the weight of the important joints. Thus, we incorporate the joints' importance for warping using a weight vector  $w$  with an entry  $w_d$  for each joint, where  $d \in \{1, \dots, k\}$ . Consequently the DTW cost matrix  $M$  from Equation (1) is adapted as follows:

$$M(i, j) = \sum_{d=1}^k w_d (1 - |q_{i,d} - q_{j,d}|). \quad (4)$$

In a naive approach, we would now weight the joints we consider important (e.g., the legs for the squat) more than the unimportant ones (e.g., the wrists). Depending on the type of motor action, this could require a huge effort in manually adjusting the weights. Instead, we propose a data-driven approach to find the appropriate weights. Related approaches for a better joint weighting [Arici et al.

2014; Celebi et al. 2013] often quantify the joints' importance via their overall contribution to the motor action. The contribution is quantified e.g., by calculating the variance of the joints' features in training recordings of the motor action of interest. However, these joint weights are prone to a high amount of noise in some of the involved joints. We aim at a goal-directed approach which optimizes the joint weights  $w$  by a minimization of the alignment error. In order to optimize  $w$ , we quantify the alignment error as

$$\frac{\#IA}{\#CA + \#IA}, \quad (5)$$

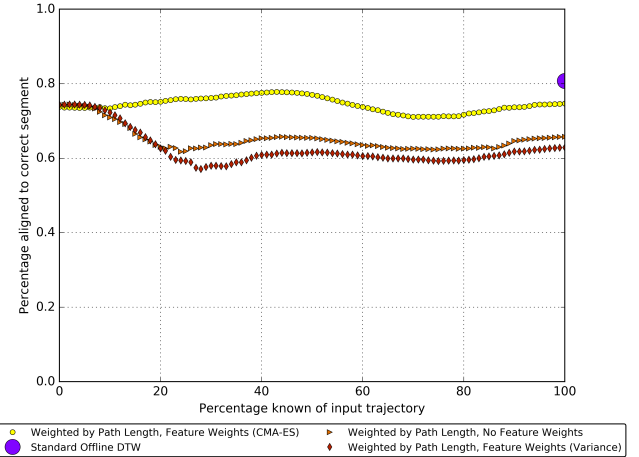
where  $\#CA$  is the number of frames on the warping path that are aligned to the correct movement segment,  $\#IA$  is the number of incorrectly aligned frames, according to the annotations.

For each CV fold, we have to optimize the weights. As we cannot directly compute derivatives of the DTW process, we use a gradient-free method: Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [Hansen and Ostermeier 1997]. An advantage of evolutionary algorithms, besides not requiring a gradient, is their low susceptibility to ending-up in local optima. We decide to use CMA-ES, as the calculation of our error term is expensive and CMA-ES only needs a comparatively low number of error evaluations during the optimization. A further advantage of CMA-ES is the small number of parameters: We only have to set initial weights  $w$  ( $w_d = 1$  for all  $d$ ), an initial step size (0.02), as well as the desired population size (parent population: 6, offspring population: 12). We use 300 iterations for optimization.

CMA-ES needs a fitness function to rate the quality of each individual. To this end, we perform our weighted OE-DTW for each training trajectory and a reference trajectory, based on the weights to be evaluated. Then, we calculate the alignment error based on Equation (5) for all training trajectories and sum-up the results. For more details on how CMA-ES works, we refer to [Hansen 2016]. We use the CMA-ES implementation from the Shark library in version 3.1.0, which is a reference implementation of [Hansen 2016].

We extend OE-DTW with optimized joint weights and path-length weighting. We call the resulting algorithm Weight-Optimized Open-End DTW (WOOE-DTW). As baseline to analyze the influence of optimized joint weights on the alignment, we use OE-DTW with path-length weighting. Additionally, we compare our results to another type of feature weighting related to approaches such as [Arici et al. 2014; Celebi et al. 2013; Reyes et al. 2011]: We quantify the influence of each joint on the motor action using its averaged variance over the whole movement: For each training trajectory, we calculate the variance of roll, pitch, and yaw of each joint over time. This variance is normalized by the maximum variance of all these features of the given trajectory. We then calculate the average of each feature over all training trajectories. For each joint, the corresponding weight is the maximum value for roll, pitch and yaw of the joint. For the OE-DTW that uses these variance-based weights, we also use path-length weighting. See Figure 5 for the results. Our optimized joint weights clearly achieve the best alignment quality.

For time measurements, we used a machine with Intel Core i7-7700K 4.2 GHz. The time needed to update the DTW matrices for a new frame only depends on the size of the reference trajectory and is constant during the whole process. The time to calculate the optimal path after the matrices are filled depends on the size of the



**Figure 5: Averaged impact of evolutionary optimized joint weights on the alignment quality.**

warping path which is always smaller than the sum of the lengths of the input trajectories. On average, we need approximately 3 ms for the alignment if WOOE-DTW sees 20 % of the motor action used as input, approximately 5 ms if 60 % are available, and less than 7 ms when WOOE-DTW knows the whole trajectory. Our results show that WOOE-DTW allows further algorithms that require an alignment of the current input, such as classifiers for error patterns in sports or rehabilitation exercises, to provide results online.

## 5 DISCUSSION AND CONCLUSION

In this paper, we propose an extension of Open-End DTW to improve the online alignment of a reference movement with an incoming motion stream. We demonstrate that the alignment quality of Open-End DTW can strongly fall behind the alignment of the offline DTW. To explain this behavior, we carve out two reasons. One is the preference of DTW for shorter paths. We demonstrate that simple penalties for diagonal paths in the accumulated weight matrix, which are suggested in related literature, do not necessarily improve the alignment quality. To circumvent the bias of DTW, we propose path-length weighting and show its positive impact on the alignment quality. The other drawback of OE-DTW is the equal weighing of all joints: In real-world scenarios, some joints are more important for certain motor actions than others, which is not considered by DTW. We exploit our annotated training data to optimize weights for each joint using evolutionary optimization. We show that our extension WOOE-DTW improves the alignment quality of OE-DTW and beats variance-based joint weights. We reach a high alignment score that nearly reaches the performance of offline DTW. This shows that WOOE-DTW can provide good alignments even for such heterogeneous data as the performance of fitness exercises. See the supplementary video for a comparison of the warps obtained by standard OE-DTW and WOOE-DTW.

As WOOE-DTW calculates an online correspondence between the input prefix and a given reference, online segmentation of the input motion comes for free. Labels of the reference trajectory can

directly be transferred to the input. When using the proposed approach for an online stream of arbitrary motion data, preceding to starting the WOOE-DTW calculation, the beginning of the motor action of interest must be detected. This can be, for instance, achieved via state-machine-based segmentation [de Kok et al. 2015] or by sliding-window-based classification [Cao et al. 2004].

Some approaches such as [Van Hanh et al. 2009] and [Caramiaux et al. 2015], which aim at the alignment of motion capture data, rely on algorithms other than DTW, partly to be less prone to noise and outliers. However, these two related approaches rely on parameters that have to be adjusted manually whereas our implementation does not have such critical parameters that directly affect alignment performance. Still, it might generally be interesting to compare the performance of these approaches to WOOE-DTW on our heterogeneous data set: This data is, as it consists of motion capture data, noisy and it contains outliers, as the squat can be performed with different styles.

### 5.1 Limitations and Future Work

Our extension of DTW requires manual labeling of training data. Indeed, the labels are as simple as “movement segment starts”. In our test scenario, we already obtain high accuracies using less than 100 annotated example movements. One limitation of our results is that we only use the squat for evaluation. Even though it is a comparatively complex motion and it is used in many related approaches, further test cases with different kinds of motor actions and synthetically generated data would be desirable to strengthen our results. We assume that different motor actions will lead to similar results: We did not use any squat-specific heuristics or tuning, but instead propose a data-driven optimization. Evaluating our approach for new motor actions only requires to annotate the movement segments in new training data and to run the evolutionary optimization to obtain the weights for the new motor action.

In the future, an integration and evaluation of WOOE-DTW in combination with other optimizations of DTW is worthwhile, such as Derivative DTW [Keogh and Pazzani 2001], Sakoe-Chuba Band [Sakoe and Chiba 1978], and Fast DTW [Salvador and Chan 2007]. Furthermore, the representation of the motion capture data itself is worth evaluating. In our work, we rely on raw data. Heloir et al. [2006] propose a PCA-based representation. Although this requires the crucial part of the movement to be covered by the PCA, such an approach can be worth using for specific types of motor actions. Another representation that can be worth evaluating is based on Self-Organizing Maps [Deng et al. 2011]. Other evaluations of the proposed WOOE-DTW could consist in using it in the context of classification of specific error patterns for a given motor action or in the discrimination of motor actions.

### ACKNOWLEDGMENTS

This work was supported by the Cluster of Excellence Cognitive Interaction Technology “CITEC” (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG).

### REFERENCES

- X. Anguera and M. Ferrarons. 2013. Memory efficient subsequence DTW for query-by-example spoken term detection. In *Multimedia and Expo (ICME), 2013 IEEE Int. Conf. On*. 1–6.
- A. Arici, S. Celebi, A. S. Aydin, and T. T. Temiz. 2014. Robust gesture recognition using feature pre-processing and weighted dynamic time warping. *Multimedia Tools and Applications* 72, 3 (2014), 3045.
- R. Bailey, J. Selfe, and J. Richards. 2011. The single leg squat test in the assessment of musculoskeletal function: a review. *Physiotherapy Practice and Research* 32, 2 (2011), 18–23.
- D. Cao, O. T. Masoud, D. Boley, and N. Papanikolopoulos. 2004. Online motion classification using support vector machines. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE Int. Conf. On*, Vol. 3. IEEE, 2291–2296.
- B. Caramiaux, N. Montecchio, A. Tanaka, and F. Bevilacqua. 2015. Adaptive gesture recognition with variation estimation for interactive systems. *ACM Transactions on Interactive Intelligent Systems (TiIS)* 4, 4 (2015), 18.
- S. Celebi, A. S. Aydin, T. T. Temiz, and T. Arici. 2013. Gesture recognition using skeleton data with weighted dynamic time warping. In *Proc. of VISAPP*. 620–625.
- H.-R. Choi, H. Y. Cho, and T. Y. Kim. 2015. Dynamically weighted DTW for dynamic full-body gesture recognition. In *Proc. of The Int. Conf. on Intelligent Systems and Image Processing*. 126–129.
- I. de Kok, J. Hough, F. Hülsmann, M. Botsch, D. Schlagen, and S. Kopp. 2015. A multimodal system for real-time action instruction in motor skill learning. In *Proc. of the Int. Conf. on Multimodal Interaction*. ACM, 355–362.
- L. Deng, H. Leung, N. Gu, and Y. Yang. 2011. Real-time mocap dance recognition for an interactive dancing game. *Computer Animation and Virtual Worlds* 22, 2-3 (2011), 229–237.
- S. Dixon. 2005. Live tracking of musical performances using on-line time warping. In *Proc. of the 8th Int. Conf. on Digital Audio Effects*. 92–97.
- R. F. Escamilla. 2001. Knee biomechanics of the dynamic squat exercise. *Medicine and science in sports and exercise* 33, 1 (2001), 127–141.
- M. A. Giese and T. Poggio. 2000. Morphable models for the analysis and synthesis of complex motion patterns. *International Journal of Computer Vision* 38, 1 (2000), 59–73.
- N. Hansen. 2016. The CMA evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772* (2016).
- N. Hansen and A. Ostermeier. 1997. Convergence properties of evolution strategies with the derandomized covariance matrix adaptation: The  $(\mu/\mu_I, \lambda)$ -CMA-ES. *Proc. of the 5th Europ. Congr. on Intelligent Techniques and Soft Computing* 97, 650–654.
- A. Heloir, N. Courty, S. Gibet, and F. Multon. 2006. Temporal alignment of communicative gesture sequences. *Computer animation and virtual worlds* 17, 3-4 (2006), 347–357.
- D. Q. Huynh. 2009. Metrics for 3D rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision* 35, 2 (2009), 155–164.
- Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu. 2011. Weighted dynamic time warping for time series classification. *Pattern Recognition* 44, 9 (2011), 2231–2240.
- E. J. Keogh and M. J. Pazzani. 2001. Derivative dynamic time warping. In *Proc. of the 2001 SIAM Int. Conf. On Data Mining*. SIAM, 1–11.
- B. Krüger, A. Vögele, T. Willig, A. Yao, R. Klein, and A. Weber. 2017. Efficient unsupervised temporal segmentation of motion data. *IEEE Transactions on Multimedia* 19, 4 (2017), 797–812.
- J. Min and J. Chai. 2012. Motion graphs++: a compact generative model for semantic motion analysis and synthesis. *ACM Transactions on Graphics* 31, 6 (2012), 1–12.
- M. Müller. 2007. *Information retrieval for music and motion*. Springer Science & Business Media.
- A. Muscariello, G. Gravier, and F. Bimbot. 2009. Variability tolerant audio motif discovery. *Advances in Multimedia Modeling (2009)*, 275–286.
- M. Reyes, G. Dominguez, and S. Escalera. 2011. Feature weighting in dynamic time warping for gesture recognition in depth data. In *Computer Vision Workshops (ICCV Workshops), IEEE Int. Conf. on*. 1182–1188.
- H. Sakoe and S. Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on acoustics, speech, and signal processing* 26, 1 (1978), 43–49.
- S. Salvador and P. Chan. 2007. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis* 11, 5 (2007), 561–580.
- P. Tormene, T. Giorgino, Silvana Quaglini, and M. Stefanelli. 2009. Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation. *Artificial intelligence in medicine* 45, 1 (2009), 11–34.
- N. Van Hanh, F. Merienne, and J. L. Martinez. 2009. Effects of virtual avatar characteristics on performance of healthy subjects’ training tasks. *Journal of CyberTherapy and Rehabilitation* 2, 3 (2009), 221–234.
- M. Vlachos, G. Kollios, and D. Gunopulos. 2002. Discovering similar multidimensional trajectories. In *Data Engineering, 2002. Proceedings. 18th Int. Conf. On*. IEEE, 673–684.