Segmentation of Exercise Repetitions Enabling Real-Time Patient Analysis and Feedback Using a Single Exemplar

SARSFIELD, Joe, BROWN, David, SHERKAT, Nasser, LANGENSIEPEN, Caroline, LEWIS, James, TAHERI, Mohammad, SELWOOD, Louise, STANDEN, Penny and LOGAN, Pip

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2019.2907483, IEEE Transactions on Neural Systems and Rehabilitation Engineering

Segmentation of Exercise Repetitions Enabling Real-Time Patient Analysis and Feedback Using a Single Exemplar

Joe Sarsfield, David Brown, Nasser Sherkat, Caroline Langensiepen, James Lewis, Mohammad Taheri, Louise Selwood, Penny Standen and Pip Logan

***Abstract*—We present a segmentation algorithm capable of segmenting exercise repetitions in real-time. This approach uses subsequence dynamic time warping and requires only a single exemplar repetition of an exercise to correctly segment repetitions from other subjects, including those with limited mobility. This approach is invariant to low range of motion, instability in movements and sensor noise while remaining selective to different exercises. This algorithm enables responsive feedback for technology-assisted physical rehabilitation systems. We evaluated the algorithm against a publicly available dataset (CMU) and against a healthy population and stroke patient population performing rehabilitation exercises captured on a consumer-level depth sensor. We show the algorithm can consistently achieve correct segmentation in real-time.**

***Index Terms*—technology-assisted rehabilitation, real-time, segmentation, rehabilitation, depth sensors, dynamic time warping**

I. Introduction

TECHNOLOGY-ASSISTED physical rehabilitation systems (TAPRS) are an active area of research and development [1]–[5]. These systems provide a platform for the patient to perform their rehabilitation exercises, and bring many potential benefits to the patient and healthcare provider. They improve adherence to the prescribed exercise regime through motivation and engagement, reduce the time burden on the physiotherapist, decrease costs and improve health through corrective feedback.

Enabling real-time analysis and feedback within such systems requires the ability to correctly segment exercise repetitions. Algorithmically, the main difficulties in automatic segmentation of exercise repetitions, especially of users with physical disabilities, are low range of motion, sensor noise in the data, instability in movements and variance in the movements [6]. Finding an invariant representation of an exercise repetition that overcomes these difficulties while remaining selective between different exercises is a key aim of exercise segmentation algorithms. In order to provide optimal feedback to the patient it is important that the algorithm can segment in real-time.

This paper proposes a practical real-time algorithm for segmenting exercise repetitions that is robust to the aforementioned difficulties. Our approach requires only a single exemplar repetition of an exercise from a single subject to effectively segment exercise repetitions from other subjects, including those with abnormal movement patterns and limited mobility.

For evaluation, this study has focused on stroke as it is the third most common cause of disability worldwide in 2010 [7] with 16.9 million cases of stroke [8] occurring globally in the same year. Stroke patients commonly have a low range of motion, instability and variance in the way they perform an exercise. Segmenting stroke patient exercise data is challenging and therefore it provides a good baseline for evaluating the robustness of the algorithm.

This segmentation algorithm is intended for TAPRS. It is assumed these systems will present the user with an exercise to perform; therefore we are not dealing with the identification of the exercise being performed, just correct segmentation when a repetition of the exercise is performed.

We hypothesise that one can take a sample exemplar repetition from any healthy subject for a single repetition of a rehabilitation exercise and use it subsequently to segment the repetitions of other subjects, including those with limited mobility.

The contribution of this paper consists of several components inspired from literature to segment repetitions of exercises from subjects with impaired mobility. This approach uses subsequence dynamic time warping (SDTW) for subsequence matching of a human motion descriptor that is robust to subjects with impaired mobility. Feature ranking is used for automatic selection of features for sequence matching and segmentation confirmation/rejection, explained in more detail in section III.

The paper is organised as follows. Section II introduces existing segmentation algorithms, section III introduces the proposed segmentation algorithm starting with an overview and then breaking down the stages: human motion descriptor; pre-processing of exemplar repetition; feature extraction; and segmentation. Section IV evaluates the segmentation accuracy and execution performance on a public dataset and our own stroke rehabilitation dataset. Finally, section V examines limitations and draws conclusions.

II. Related Work

This section introduces published research relating to the temporal segmentation of actions. Gong, et al. [9] have proposed an approach that segments multivariate time series data of temporal joint positions (or angles). The algorithm was trained and tested on a subset of data of 15 people performing 10 actions once. However there was no evaluation using clinical data of patients with physical disability

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2019.2907483, IEEE Transactions on Neural Systems and Rehabilitation Engineering

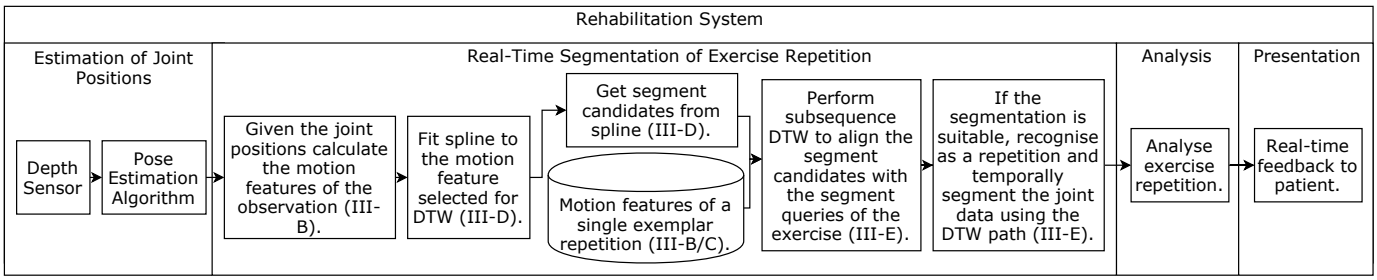


Fig. 1. Flowchart of the segmentation algorithm and how it fits into a proposed rehabilitation system.

that commonly show low range of motion and instability. Lin and Kulic [6] proposed using zero-velocity crossings (ZVC) as segment candidates with a Hidden Markov Model (HMM) based template matching method to segment actions, which requires training data. A multimodal approach was proposed by Wu, et al. [10] that segments gestures from joint positions, depth and RGB data. This approach requires training data for different modalities. Chaun-Jun, et al [11] have proposed a Dynamic Time Warping (DTW) approach for segmenting rehabilitation exercises. The system was tested on three shoulder exercises performed by four healthy people. They use DTW to align the joint data but no dimensionality reduction methods are used. De Souza Vicente, et al. [12] have proposed using latent-dynamic conditional random fields. A filtering technique based on key poses is used to reduce the number of frames prior to segmentation. They tested their algorithm on Taekwondo moves which are relatively fast compared to general gestures. The approach requires training data. Krüger, et al. [13] proposed a kd-tree-based nearest-neighbor-search that is said to be a fast alternative to subsequence DTW alignment. This approach was used by Baumann, et al. [14] and extended to enable its use for action recognition. This approach was not tested on clinical subjects with limited mobility. Krüger, et al. [15] have proposed an unsupervised approach that identifies action repetitions and further decomposes the actions into atomic motion primitives. However, the segmentation algorithm was tested on non-clinical data from ”fairly constrained settings”. Wang, et al. [16] have proposed an unsupervised approach to segmentation. Joint trajectories are converted to a kinematic model using an unscented Kalman filter and the most representative kinematic parameters for the segmentation of an action repetition are selected. ZVC are detected to produce a list of segmentation candidates. Finally, k-means clustering is used to determine the boundaries of each repetition. The algorithm was tested on joint data from healthy subjects performing non-clinical actions. Lin, et al. [17] have proposed a two-class classifier to classify each data point as either a segment or non-segment point. Dimensionality reduction is performed prior to data point classification. However, the classification stage requires training data. The top performing classifiers used Principal Component Analysis (PCA) prior to classification. Support Vector Machine (SVM), Artificial Neural Network (ANN) and k-Nearest Neighbour (k-NN) provided the highest accuracy in segmentation. They mention high processing costs of k-NN

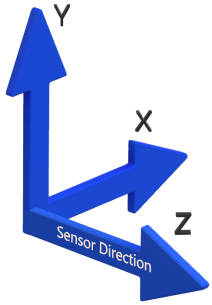


Fig. 2. Depiction of the global coordinate system used in this paper, this follows the same coordinate system used by Kinect.

making it unsuitable for real-time applications. Devanne, et al. [18] have proposed a segmentation algorithm that jointly analyses the shape of the human pose and motion in a Riemannian manifold. The approach was tested on temporal skeleton data of healthy participants performing actions. Shan, et al. [19] identify key poses by analysing the joint data for minimal changes in kinetic energy, a parameter that must be tuned. Then atomic action templates (AAT) are produced from the key poses and temporal midway points. Multiple AATs can form an action template. Finally, a classifier is used to classify the AATs and determine a label for the action. Four classification models were tested; HMM, k-NN, SVM and Random Forest. The classifiers obtain similar recognition results suggesting the feature representations of an action are sufficiently discriminative, although this was tested on healthy datasets of non-clinical actions. This approach also requires training data.

Many of these approaches are limited in that they either require multiple samples for training and/or were evaluated using healthy participants, or are too computationally expensive to run in real-time.

III. Real-Time Segmentation Algorithm

A. Algorithm Overview

Figure 1 depicts a high-level flowchart of a proposed rehabilitation system to incorporate the segmentation algorithm.

The challenge is to find the start and end of an exercise repetition in real-time, thus enabling analysis of the repetition and responsive feedback to the user. This approach requires only a single exemplar repetition from a single subject for each exercise. This approach assumes that the exercise being

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2019.2907483, IEEE Transactions on Neural Systems and Rehabilitation Engineering

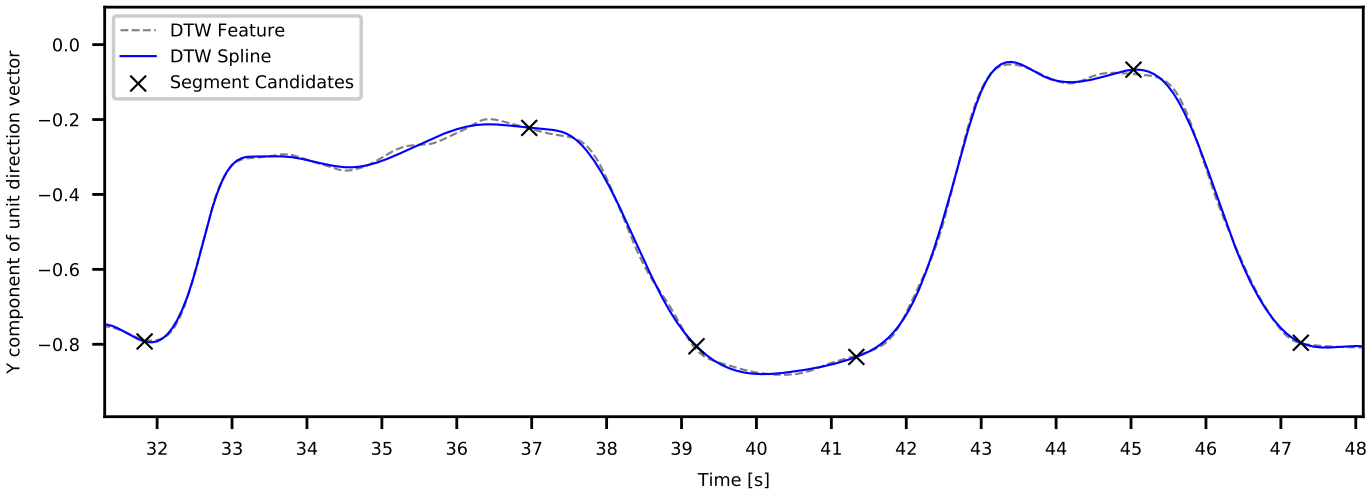


Fig. 3. Plot showing a cubic spline fitted to the DTW feature and the segment candidates selected for DTW alignment.

performed is known; this follows current rehabilitation exercise regimes where the exercises are presented to the patient in a specified order. The term exemplar refers to the motion data of a single repetition of an exercise performed correctly and the term observation refers to the motion data collected in real-time of a patient or other subject.

Our proposed algorithm is the combination of several components, inspired from literature, to solve the challenge of real-time segmentation of repetitions from subjects with impaired mobility. This approach evaluates a subset of sequences in the observation using subsequence dynamic time warping (SDTW) on a 1-D human motion feature ranked as the most informative, as described in Section III-C. If the alignment is similar, the warping path is used to measure the similarity of the other 1-D motion features in the human motion descriptor to determine if a repetition of an exercise was performed, meaning SDTW is only performed on the most informative 1-D feature.

The following list provides a brief overview of the key aspects of the proposed algorithm:

- Exercise repetitions represented as a temporal array of unit direction vectors between joints rotated to a local coordinate system. These features make the system invariant to skeleton size, position and plane while remaining selective to the exercise. Detailed in section III-B.
- Pre-processing of the exemplar repetition to rank features by importance and find the total cost threshold for each motion feature. Detailed in section III-C.
- Fitting a spline to the DTW feature using cubic spline interpolation, as proposed by [20]. This improves invariance to noise and instability in movements by capturing the general trend of the movement. Detailed in section III-D.
- Feature extraction, achieved by extracting key features from the spline, such as zero-velocity crossings. This ensures the algorithm can run in real-time. Detailed in section III-D.
- Normalisation of the DTW motion feature to zero-mean and unit variance, as proposed in [21]. This scales the DTW features to comparable ranges before performing DTW alignment, as shown in Figure 4. Detailed in section III-E.
- Alignment of the exemplar to the observation using SDTW. This ensures the segmentation is invariant to speed as rehabilitation subjects tend to perform repetitions slowly. Detailed in section III-E.
- Segmentation confirmation/rejection is performed using the DTW warping path on other motion features to quickly calculate their total cost and compare against the total cost threshold. This reduces false positive segmentations. Detailed in section III-E.

B. Human Motion Descriptor

The human motion descriptor consists of a set of 1-D human motion features. A single motion feature is a temporal array of a single component of a unit direction vector between joints e.g. the X axis of a unit direction vector between the shoulder joint and the elbow joint. This motion feature is more selective, when compared to angles, as they describe the direction in which a bone is moving, as well as its speed along the given axis. The motion feature is invariant to body size and selective to direction. A set of these motion features are used to perform a segmentation but only one of these motion features is used for finding the warping path using SDTW.

To achieve viewpoint invariance, a local coordinate system is used as proposed by [22]. For each frame in the features, the direction vector between the hip joints is aligned parallel to the X axis of the global coordinate system, shown in Figure 2. Then, given this rotation, each of the features is rotated around the Y (vertical) axis by the same rotation.

C. Pre-processing of Exemplar Repetition

Given an exemplar repetition of an exercise and a set of joints, suitable sets of joints are selected to become candidates for the automatic generation of motion features e.g. the squat exercise may use the sets {{HipLeft, KneeLeft}, {HipRight,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2019.2907483, IEEE Transactions on Neural Systems and Rehabilitation Engineering

KneeRight}}. Using an approach similar to [23], the motion features are ranked based on the most informative i.e. most change over time. The motion features taken from the exemplar exercise are ranked by their change over time, as shown in equation 1, where FR is the feature rank and F is the motion feature. The FR of each F is stored in a set FRS ordered by FR. The motion feature with the most change over time is used as the DTW feature for alignment.

$$FR = \sum_{i=2}^{len(F)} |F_i - F_{i-1}| \tag{1}$$

Total cost thresholds for each motion feature are defined by equation 2, where TCT is the total cost threshold that the same motion feature in the observation must be lower than to be considered a segmentation, DM is the distance multiplier and DB is the distance base. DM and DB are parameters that need setting for each exercise. This ensures motion features taken from joints with more movement have a higher TCT. This approach reduces the number of parameters that require setting for each exercise. The FR and TCT are calculated once from the exemplar repetition of the exercise.

$$TCT = ((FR/max(FRS)) * DM) + DB \tag{2}$$

Given that the exemplar represents a repetition of the exercise with complete range of motion, a minimum scale parameter is calculated to reject subsequences of the observation with a small range. As shown in equation 3, where $MinScalePerc$ is the percentage of the range, DF is the DTW feature of the exemplar and $MinScale$ is the minimum scale attained by a subsequence to be considered for segmentation.

$$MinScale = (max(DF) - min(DF)) * MinScalePerc \tag{3}$$

D. Feature Extraction

The motion features will likely contain superfluous information for the task of segmentation as joint data is captured many times a second, e.g. Microsoft Kinect captures joint data at 30Hz. Feature extraction enables a real-time implementation and improves accuracy of segmentation by selecting the most informative motion features which we define in this section. The feature extraction occurs on the DTW feature, chosen using the feature ranking method in section III-C, of both the exemplar and observation. This results in a subset of values from the DTW feature consisting of only the key motion features, these features will be referred to as segment queries and segment candidates respectively and collectively referred to as segment points; these concepts are explained in detail in this section.

Before extracting the segment points, the general trend of the DTW feature is calculated by fitting a cubic spline to produce a DTW spline, as shown in Figure 3. The first derivative of the DTW spline is also calculated to retrieve the velocity.

Segment queries are then extracted using the first two methods while segment candidates are extracted using all of the following methods:

Algorithm 1 Pseudocode to find the optimal warp path for a single time step. One-based indexing.

Input: C {Segment candidates}
Input: Q {Segment queries}
$m \leftarrow len(Q)$
$l \leftarrow len(C)$
$Q \leftarrow Normalise(Q)$ {Normalise query to zero-mean and unit variance. This step can be performed once and stored as the query does not change}
$OW \leftarrow []$ {Optimal warp path i.e. path with lowest total cost}
$OTC \leftarrow \infty$ {Total cost of OW}
for $i \leftarrow l; i > 0; i \leftarrow i - 1$ **do**
  $S \leftarrow C[c_i, ..., c_l]$ {Subsequence of candidate}
  if $max(S) - min(S) < MinScale$ **then**
    continue
  end if
  $S \leftarrow Normalise(S)$
  $W, A \leftarrow GetWarpPath(Q, S)$ {A is costs matrix}
  $TC \leftarrow GetTotalCost(W, A, m)$
  if $TC < OTC$ **then**
    $OTC \leftarrow TC$
    $OW \leftarrow W$
  end if
end for
return OW, OTC

1) Zero-velocity crossings. Given the spline representing the velocity, extract the DTW spline values where the velocity crosses zero, as proposed in [24].
2) The first and last values in the DTW spline. As we are dealing with real-time segmentation, when new joint data arrives, the very latest value is a potential repetition end segment candidate. The oldest value is a potential repetition start segment candidate. The first and last values of the exemplar will of course be segment queries that represent the start and end of a repetition.
3) Values in the observations DTW spline that cross through the first value in the exemplars DTW spline. This is required because if movement occurs before the repetition starts then method 1 will miss a potential segment candidate or will segment early.

The segment points are further reduced by removing segment points that have similar values to surrounding segment points with the latest segment point kept. Explicitly, a rolling absolute difference of the segment points is calculated, and if this difference is below a threshold then the oldest segment points are removed. This can be seen in Figure 3 between frames ≈1300 to ≈1350 where multiple segment points would have been proposed at the peak but only the latest segment point was kept. These segment points become the DTW features for alignment.

E. Segmentation

Algorithms 1 and 2 present the pseudocode for finding the DTW warping path and performing a segmentation respectively. These are described in more detail in this section.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2019.2907483, IEEE Transactions on Neural Systems and Rehabilitation Engineering

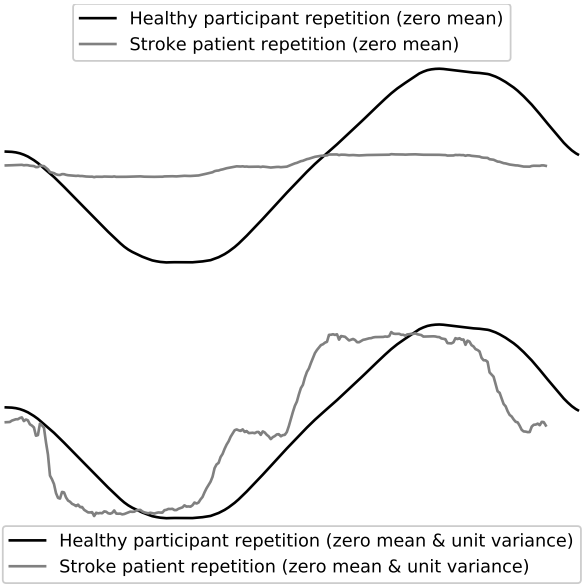


Fig. 4. The top graph shows a repetition of a healthy and stroke patient DTW motion feature scaled to zero mean. The bottom graph shows the result of normalising the same features to zero mean and unit variance.

The problem formulation is as follows. Given two sequences, a query sequence Q with length m and a candidate sequence C with length l, find an alignment between Q and a subsequence S with length n, of C that minimises the DTW cost between Q and S. Q consists of the segment queries and C consists of the segment candidates described in section III-D. Note that the segment points in C and Q will likely have differing distances temporally, e.g. C_{i-1} and C_i may be temporally closer together than C_i and C_{i+1}, when mapped back to the original motion feature. Further definitions are used in this section. W refers to the warping path, of length k, consisting of the indices of the segment points from Q and S that are aligned. A and D refer to the costs and cumulative costs matrices between Q and S respectively. i and j are indices referring to an element in a one or two dimensional array, e.g. $D_{i,j}$ would refer to the ith row and jth column of D. Note that the indices are stored in W instead of the values of the segment points, e.g. $W_1 = (1, 1)$ instead of $W_1 = (Q_1, S_1)$, this allows us to map the warping path indices to other motion features in order to calculate their total cost. These variables are defined in table I.

This approach uses SDTW for the sequence matching between Q and a suitable subsequence of C that meets the conditions described in this section.

Equation 4 [21] is used to normalise sequences Q and S to zero-mean and unit variance. Given input sequence $T = \{t_1, ..., t_{end}\}$ where μ and σ are the mean and standard deviation of T respectively. This step is shown on lines 3 and 11 in algorithm 1. Sart, et al. [21] note that some researchers suggest normalising between a range of [0,1] or [-1, 1] but state that this approach is sensitive to noise. Figure 4 shows the result of normalising a Twist repetition from a healthy subject and a repetition from a stroke patient to zero mean and unit variance. It can be seen that this normalisation step brings the motion feature of the healthy subject and stroke patient subject into a comparable range.

$$\hat{T} = \{\hat{t}_1, ..., \hat{t}_{end}\} \text{ where } \hat{t}_i = \frac{(t_i - \mu)}{\sigma} \tag{4}$$

Before performing DTW on Q and S, the scale of S must meet a minimum scale threshold. Specifically, the distance between the minimum and maximum value in S is checked against the minimum scale value calculated in equation 3. This ensures the motion was significant enough to be considered as a repetition and is performed before normalising S, see line 8 in algorithm 1.

For each iteration in the *for* loop on line 6 algorithm 1; a new subsequence is created on line 7, DTW alignment between Q and S is performed on line 12 and the total cost of the warp path is calculated on line 13. It can be seen that for each iteration, the length of S increases by one, starting with the latest segment candidate in C, this approach finds the latest repetition.

The DTW presented here follows the original implementation [25] with modifications and conditions as follows:

- Start and end boundary condition [25]: The first and last indices of Q and S are respectively placed in the first and last element in the warping path e.g. $W_1 = (1, 1)$ and $W_k = (m, n)$. Note that DTW aligns Q to S instead of C.
- Continuity condition [25]: The indices in the warp path advance at most one index, i.e. step size is 1, this ensures all indices of the segment points in Q and S are in W; e.g. $W_k - W_{k-1} \in \{(1, 0), (0, 1), (1, 1)\}$
- Normalisation to zero-mean and unit variance, as proposed in [26]. Q and S are normalised to zero-mean and unit variance before alignment. This step ensures segment candidates from a patient with a low range of motion are correctly aligned to the segment queries from a healthy subject by normalising the peaks and troughs.
- Early abandoning of subsequence alignment [26]. By storing the total cost of the best subsequence so far, we can abandon the calculation of the new subsequence if the total cost has gone higher than the best so far.
- The total cost of the warp path between Q and S differs from the original DTW algorithm [25], as shown in equation 8.

As proposed in [25], to find the optimal warp path W, the costs, i.e. distances, between each segment point in Q and S needs to be calculated. The Manhattan distance is used to calculate the difference. A and D are $m \times n$ matrices that store the costs and cumulative costs respectively. A, D and W are formally described in equations 5, 6 and 7, these calculations are performed on line 12 in algorithm 1. Note, W is calculated backwards starting with (m, n) until $(1, 1)$ is appended to W.

$$A_{i,j} = |Q_i - S_j| \tag{5}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2019.2907483, IEEE Transactions on Neural Systems and Rehabilitation Engineering

TABLE I
GLOSSARY OF THE MAIN VARIABLES PRESENTED IN THIS SECTION.

Variable/s	Definition
Q, m	Query sequence with length m
C, l	Candidate sequence with length l
S, n	Subsequence of C with length n
A	DTW costs matrix with length $m \times n$
D	DTW cumulative costs matrix with length $m \times n$
i, j	Indices referring to an element in a one or two dimensional array e.g. $D_{i,j}$ refers to the ith row and jth column of D
W, k	DTW warping path with length k. Each element consists of a pair of indices relating to elements in Q and S that are aligned, e.g. W_k containing a pair of indices (i, j) = $W_{k_{i,j}}$ = the alignment between Q_i and S_j

$$D_{i,j} = \begin{cases} A_{i,j} \text{ if } i = 1 \text{ and } j = 1, \\ A_{i,j} + D_{i,j-1} \text{ if } i = 1 \text{ and } j > 1, \\ A_{i,j} + D_{i-1,j} \text{ if } i > 1 \text{ and } j = 1, \\ A_{i,j} + \min(D_{i-1,j-1}, \\ \qquad D_{i-1,j}, \\ \qquad D_{i,j-1}) \text{ otherwise} \end{cases} \tag{6}$$

$$W_{index} = (i, j) = \begin{cases} (1, 1) \text{ if } i = 1 \text{ and } j = 1, \\ (m, n) \text{ if } i = m \text{ and } j = n, \\ (i, j-1) \text{ if } i = 1 \text{ and } j > 1, \\ (i-1, j) \text{ if } i > 1 \text{ and } j = 1, \\ indices \text{ otherwise} \end{cases} \tag{7}$$

$$indices = \begin{cases} (i-1, j) \text{ if } D_{i-1,j} \leq D_{i,j-1} \wedge \\ \qquad D_{i-1,j} < D_{i-1,j-1} \\ (i, j-1) \text{ if } D_{i,j-1} < D_{i-1,j} \wedge \\ \qquad D_{i,j-1} < D_{i-1,j-1} \\ (i-1, j-1) \text{ otherwise} \end{cases}$$

The total cost TC, i.e. DTW distance, between Q and S given W, is calculated differently from the original DTW proposal. To calculate TC the average Manhattan distance of all segment candidates in S that align to a segment query in Q is calculated, this occurs for each Q_i and the averages summed. Finally, the result is normalised by m. Equation 8 formally describes the calculation for finding TC which occurs on line 13 in algorithm 1.

$$TC = \frac{\sum_{p=1}^{m} \overline{\{A_{i,j} | (i, j) \in W \wedge i = p\}}}{m} \tag{8}$$

Window constraints, such as the Sakoe-Chiba band [25] and Itakura parallelogram [27], should be used with caution as there can be large and varying differences in time between segment points.

When the optimal warp path has been found for a single time step, its total cost is checked against a minimum total cost parameter. If it is below this parameter then the subsequence can be considered for segmentation.

Finally, as DTW alignment has only been performed on one motion feature, confirmation of the segmentation is performed as follows. Several motion features with the highest feature ranks, as described in equation 1, have their total cost calculated using the indices in the optimal warp path. If any of these values are above their total cost threshold, calculated

Algorithm 2 Pseudocode to confirm a segmentation of a repetition after performing subsequence DTW.

Input: OW {Optimal Warp Path for this time step, see Algorithm 1}
Input: OTC {Optimal Warp Path Total Cost for this time step, see Algorithm 1}
Input: $Paths$ {Optimal Warp Paths for the last NT time steps. Global variable initialised to empty set}
Input: $Costs$ {Optimal Warp Path Total Cost for the last NT time steps. Global variable initialised to empty set}
Input: WRE {Wait for Repetition End boolean. Global variable initialised to FALSE}

```
1:  if WRE = TRUE or OTC < SegmentThreshold then
2:      if WRE = FALSE then
3:          for all F ∈ Features do
4:              TC ← GetFeatureTotalCost(OW, F, Q) {Subsequence of F is retrieved using indices in OW}
5:              if TC > TCT then
6:                  return
7:              end if
8:          end for
9:          WRE ← TRUE
10:     end if
11:     Paths.append(OW)
12:     Costs.append(OTC)
13:     if len(Costs) − argmin(Costs) > MaxLookAhead then {MaxLookAhead is the number of future time steps to check for a lower DTW cost}
14:         RWP = Paths[argmin(Costs)] {Repetition Warp Path}
15:         WRE ← FALSE
16:         Paths.clear()
17:         Costs.clear()
18:         return RWP {Segmentation confirmed, RWP can retrieve joint positions for the repetition.}
19:     end if
20: end if
21: return
```

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2019.2907483, IEEE Transactions on Neural Systems and Rehabilitation Engineering

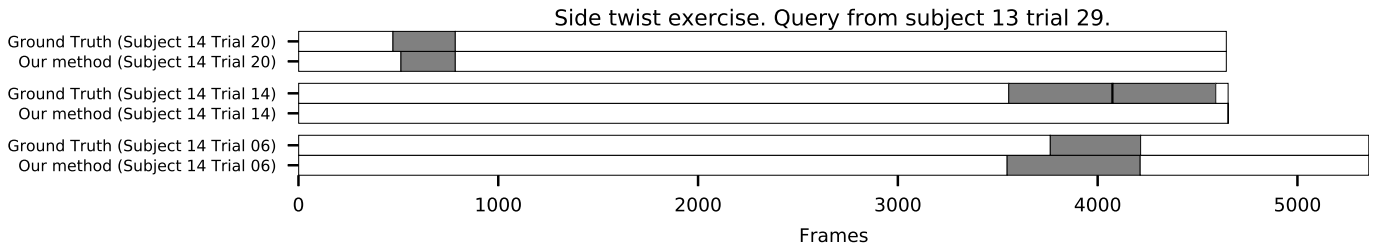


Fig. 5. Segmentation performance of side twist exercise using query from subject 13.

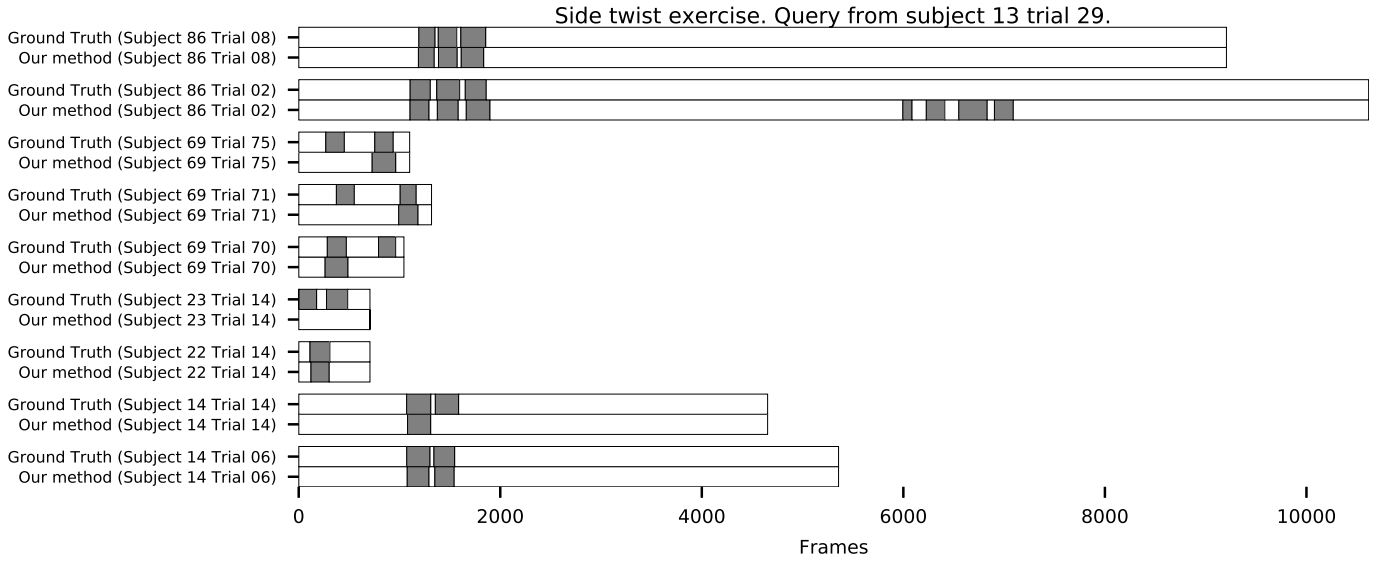


Fig. 6. Segmentation performance of squat exercise using query from subject 13.

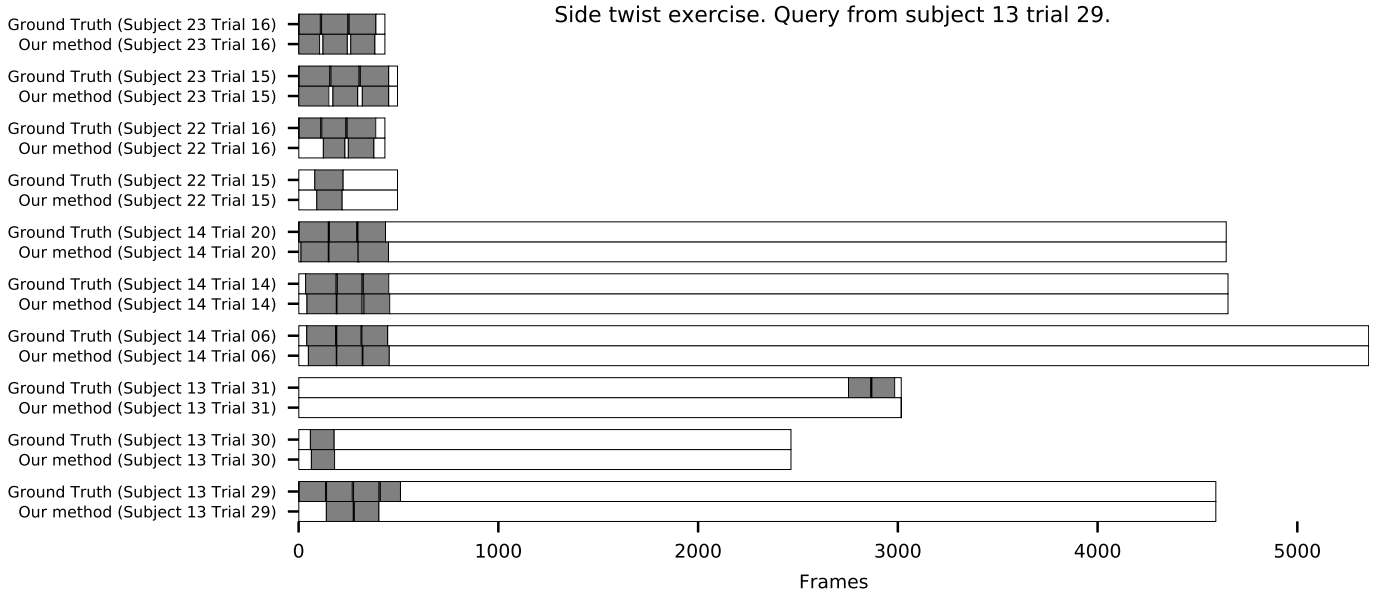


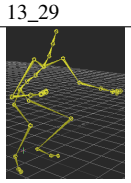
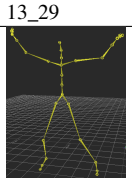
Fig. 7. Segmentation performance of jumping jack exercise using query from subject 86.

in equation 2, then the segmentation is rejected. Otherwise the segmentation is confirmed for this time step. The pseudocode for confirming a segmentation is presented in Algorithm 2.

Future joint information is required to determine if this is in fact the very end of the exercise repetition. Thus, the trend of the DTW motion feature's total cost is measured over several time steps to check if it is still decreasing. Once the total cost begins to rise the segmentation of the observation occurs on the time step with the lowest total cost.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2019.2907483, IEEE Transactions on Neural Systems and Rehabilitation Engineering

TABLE II
SUMMARY OF CMU SEGMENTATION RESULTS USING A SINGLE EXEMPLAR REPETITION FROM A SINGLE SUBJECT.

Exercise	Number of Candidate Subjects	Number of Ground Truth Segments	True Positive Segments	False Negative Segments	False Positive Segments	Exemplar Subject_Trial
Side Twist	1	4	2	2	0	 13_29
Squat	5	19	13	6	4	 13_29
Jumping Jacks	4	26	21	5	0	 86_05

F. Parameters

The following list describes the key parameters of the segmentation algorithm:

- Minimum Scale Percentage: When performing DTW, the scale of the subsequence must be at least this percentage of the query scale to be considered for segmentation, see equation 3.
- Segmentation Threshold (ST): The optimal warp path's total cost must be less than this threshold to be considered for segmentation.
- DTW Distance Multiplier (DM): A multiplier to give motion features with more change over time a higher total cost threshold, see equation 2.
- DTW Distance Base (DB): A base value each motion feature is given when calculating the total cost threshold, see equation 2.

IV. EVALUATION AND RESULTS

A. General Evaluation Against Public Dataset

Three exercises were selected from the CMU Graphics Lab Motion Capture Database (CMU) [28] based on their similarity to rehabilitation exercises which tend to have repetitive motions. Thus, actions that were more activity based were ignored as they are out of scope of the proposed algorithm. During rehabilitation, patients follow an exercise regime presenting them with exercises to perform. Therefore, unlike the following papers [6], [9], [10], [12], [15], [18], [19] that deal with the recognition and segmentation of actions, our proposed algorithm is aimed at segmenting repetitions of a known exercise given a correct repetition of the exercise.

Figures 5, 6 and 7 show the segmentation performance on the CMU database. The grey bars represent the segmentation of a repetition and white spaces represent no repetition. The bars are in pairs with the ground truth segmentations at the top and the algorithmic segmentations on the bottom. Most of the trials contained more than one exercise/action which occurred during the large white spaces in the bars. The algorithm was tested on all frames of the trials to test the algorithm on its ability to avoid false positives. Where the ground truth grey bars do not have an algorithmic grey bar below, this is a false negative. Likewise, algorithmic grey bars without a ground truth grey bar above is a false positive.

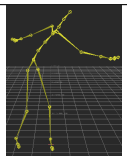
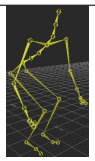
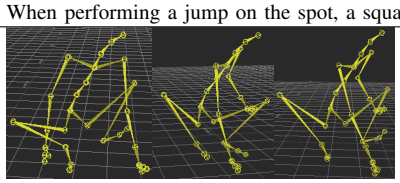
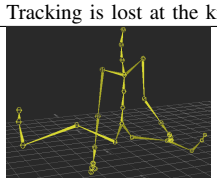
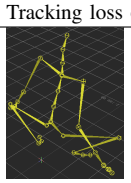
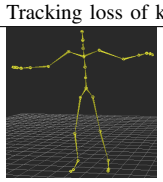
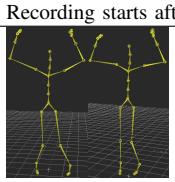
Table II summarises the results of the segmentations. The jumping jacks, squat and side twist exercises achieved 21/26, 13/19 and 2/4 correct segmentations from 4, 5 and 1 candidate subjects respectively. It should be noted that the segmentation algorithm was not the cause of most of the failures, as shown in table III. The reason for the false negative segments on the squat exercise was poor joint tracking e.g. squat 14_14, 23_14 and 69_70/71/75. Jumping jacks and side twist exercises failed due to the exercise being incorrectly performed e.g. side twist 14_14 and jumping jacks 13_29/31, and one case of the recording starting after the repetition had begun e.g. jumping jacks 22_16. One set of false positive segments occurred on the squat exercise, 86_02, where the subject performed a squatting action before a jump. This is the only case of false positive segments although many other exercises and actions were performed during the trials. This demonstrates that the segmentation algorithm can achieve suitable results from a single exemplar repetition.

B. Execution Performance Evaluation

Figure 8 shows the execution performance of the whole segmentation algorithm segmenting the latest repetition of an exercise as the number of frames increases. The exercises squat, jumping jacks and side twists were evaluated. The DTW

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2019.2907483, IEEE Transactions on Neural Systems and Rehabilitation Engineering

TABLE III
REASONS FOR FAILURES OF CMU SEGMENTATIONS USING A SINGLE EXEMPLAR REPETITION FROM A SINGLE SUBJECT.

Exercise Subject_Trial	**Failure**	**Comment**
Side Twist 14_14	False Negative	 Side twists are performed with arm extension and leaning.
Squat 86_02	False Positive	 When performing a jump on the spot, a squat action is performed before jumping.
Squat 69_70, 69_71, 69_75	False Negative	 Tracking is lost at the knees and feet joints, causing them to rise up and align along the X axis.
Squat 23_14	False Negative	 Tracking loss of the leg joints.
Squat 14_14	False Negative	 Tracking loss of knee joints.
Jumping Jacks 22_16	False Negative	 Recording starts after the first repetition has begun, subsequent repetitions are correctly segmented.
Jumping Jacks 13_29, 13_31	False Negative	 The repetitions are performed incorrectly, with the legs moving together as the arms are raised.

query of each exercise consisted of 3, 3 and 6 segment queries respectively.

The time complexity for the worst case scenario of the whole segmentation algorithm is $O(ml^2)$ as multiple subsequences are evaluated using SDTW. The function GetWarpPath in algorithm 1 on line 12 performs DTW which has a time complexity of $O(mn)$ and GetTotalCost on line 13 has a time complexity of $O(m)$.

The runtime performance on real-world examples of CMU exercises is depicted in Figure 8. The processing time taken does not increase as much as the $O(ml^2)$ term would suggest, this is because a subsequence must meet a minimum scale to be considered for segmentation, and the DTW alignment is abandoned when the total cost goes above the lowest total cost so far.

Considering a worst case scenario of segmenting stroke patient repetitions, where the movements are often slow. The longest repetition performed by a stroke patient was ≈13 seconds. Thus, segmenting the latest repetition given a 20 second window consisting of 600 frames, from a device with a 30Hz capture rate, would take ≈5ms given Figure 8. Although the algorithm can be run more infrequently, the processing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2019.2907483, IEEE Transactions on Neural Systems and Rehabilitation Engineering

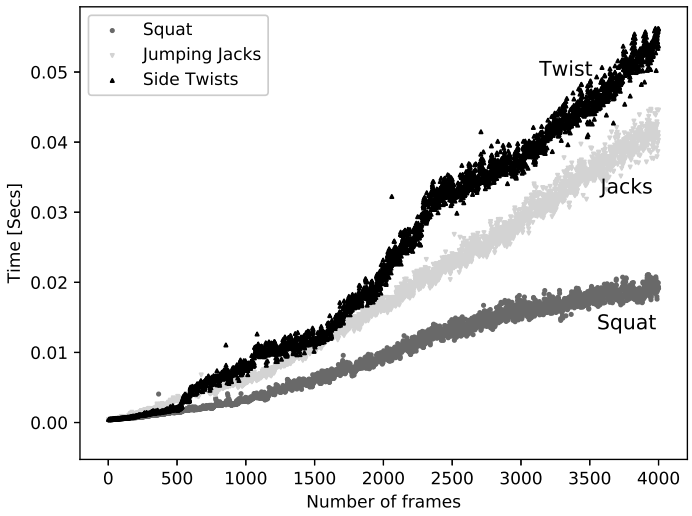


Fig. 8. Execution performance of the segmentation algorithm when finding the latest repetition. This includes: pre-processing the motion features, feature extraction, finding the optimal warp path and segmentation rejection/confirmation using the highly ranked motion features.

time ≈5ms is well within the real-time capture rate of 33ms. If an exercise repetition was to be performed beyond the window size then the repetition will likely be missed or segment the start of the exercise late. To resolve this every n^{th} frame could be dropped until it is able to process the repetition in real-time at the expense of segmentation accuracy, however, the evaluation shows the algorithm maintained real-time processing well beyond the sequence length of real exercise repetitions.

The performance evaluation took place on a Intel Core i7 4790K at 4.00GHz using a single threaded Python implementation. Note that this execution performance evaluation is an example of the worst-case performance as certain optimisations have been left out such as: DTW window constraints; online normalisation [26]; compilation to native machine code; and clearing the buffer when a repetition is detected. But for the purposes of a real-world implementation of a system designed to segment exercises in real-time, it can be seen that even an unoptimised implementation is sufficiently fast.

C. Detailed Evaluation on Rehabilitation Exercises

1) Evaluation Technique: The dataset used for evaluation contains three exercises performed by four stroke patients undergoing rehabilitation at home. Ethical approval for this study was obtained via the University's ethical approval board. The three exercises within the dataset are arm to side, arm to front and twist exercise. The exercises were taken from the Graded Repetitive Arm Supplementary Program (GRASP) manual [29], an exercise program developed for stroke patients. Each subject performed three repetitions of each exercise. For evaluation, the data passed to the segmentation algorithm simulates a real-time implementation, i.e. the algorithm receives data frame by frame and cannot see future data. For each exercise being evaluated, joint data of all exercises were included in the evaluation to ensure the algorithm was robust to false positives. We follow the methodology presented in [6] as they evaluated a similar clinical population as ours, thus allowing an almost direct comparison of the effectiveness of both algorithms. The evaluation methodology is as follows:

1) Simulate a real-time implementation by sending observation motion data frame by frame to the segmentation algorithm. Once the full observation has been processed, a list of segments is returned. This approach tests the segmentation algorithm for early segmentation.
2) The following definitions are used in the evaluation of the segmentation algorithm's accuracy:
    a) Ground truth segment (GT): Ground truth segment envelopes (e.g. hatched rectangles in Figure 9) are added to the observations to represent the ground truth of the start and end of an exercise repetition. A GT has an envelope of acceptability with varying sizes as the start and end of an exercise repetition is often ambiguous, as mentioned in [6]. For this evaluation, video data was used to determine the start and end of an exercise, timestamps were used to temporally align the video data and joint data.
    b) Time Error (TE): Time error is a variable that increases the temporal width of the GT envelopes by X time. This is to allow algorithmic segmentations that were close to a GT without a TE to be considered a true positive segment.
    c) True positive segment (TP): If an algorithmic segment is within the TE of a GT then it is classed as a TP; e.g. if TE is set to 1 second and an algorithmic segment is within 1 second of a GT, then the number of TPs is incremented by 1.
    d) False positive segment (FP): If an algorithmic segment exists where there should not be a segment; e.g. an algorithmic segment is not within a TE of a GT, then the number of false positive segments is incremented by 1.
    e) False negative segment (FN): If no algorithmic segment exists where there should be one; e.g. no algorithmic segment is within the TE of a GT, then the number of false negative segments is incremented by 1.

Note that if an algorithmic segment is just outside of the TE envelope to a TP segment then the number of FP and FN segments are incremented by one. Algorithmic segments that represent the start of a repetition cannot be classed as TP if they fall within the TE of a GT representing the end of a repetition. Similarly, algorithmic segments that represent the end of a repetition cannot be classed as TP if they fall within the TE of a GT representing the beginning of a repetition.

2) Segmentation Performance on Rehabilitation Exercises: Tables IV, V, VI, VII, VIII, IX show the performance of the segmentation algorithm at different Time Errors (TE) starting at 0 seconds and incrementing by 0.1 up to 0.5 seconds. The results of the segments are represented as a percentage of the total GT. The mean average error (MAE) is displayed, where error is the minimum difference in the algorithmic segment direction to the GT segment direction.

The patient dataset we have evaluated is challenging due

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2019.2907483, IEEE Transactions on Neural Systems and Rehabilitation Engineering

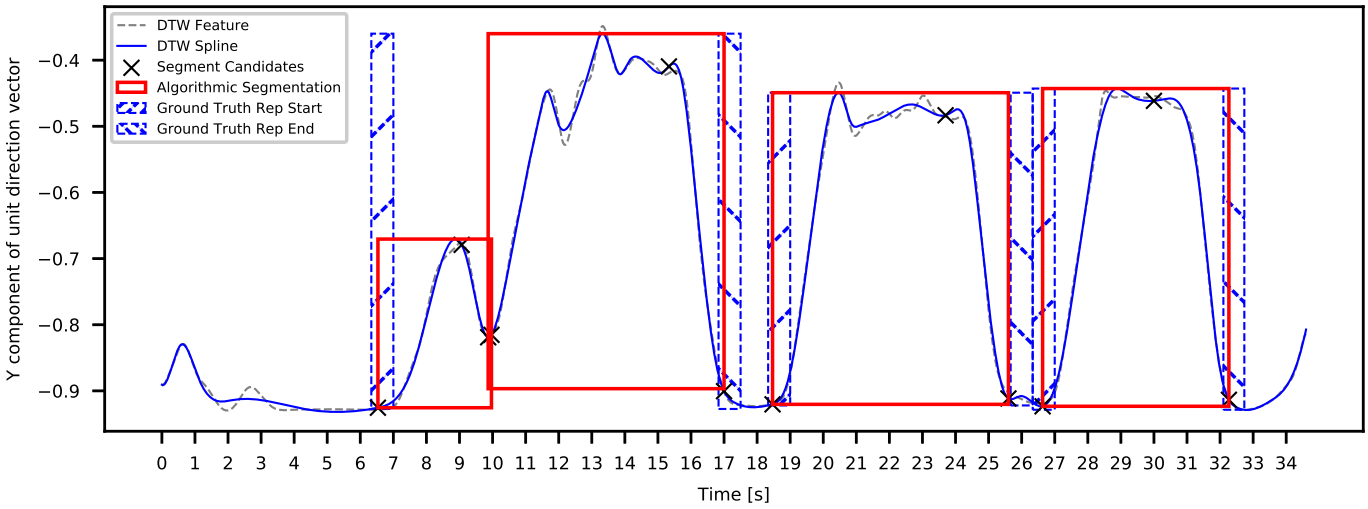


Fig. 9. Plot showing the algorithmic segmentations of a stroke patient performing the Arm to Side exercise taken from GRASP, a stroke rehabilitation manual.

TABLE IV
KINECT EXERCISE 1

TE (s)	TP (%)	FP (%)	FN (%)	ESS		LSS		EES		LES	
				%	MAE (m)	%	MAE (m)	%	MAE (m)	%	MAE (m)
0.0	54.2	54.2	45.8	0.0	0.0000	0.0	0.0	0.0	0.000	0.0	0.0000
0.1	75.0	33.3	25.0	11.1	0.0016	0.0	0.0	11.1	0.012	5.6	0.0009
0.2	83.3	25.0	16.7	10.0	0.0016	0.0	0.0	15.0	0.022	10.0	0.0029
0.3	87.5	20.8	12.5	14.3	0.0021	0.0	0.0	14.3	0.022	9.5	0.0029
0.4	91.7	16.7	8.3	13.6	0.0021	0.0	0.0	18.2	0.042	9.1	0.0029
0.5	95.8	12.5	4.2	13.0	0.0021	0.0	0.0	21.7	0.048	8.7	0.0029

TABLE V
KINECT EXERCISE 2

TE (s)	TP (%)	FP (%)	FN (%)	ESS		LSS		EES		LES	
				%	MAE (m)	%	MAE (m)	%	MAE (m)	%	MAE (m)
0.0	75.0	25.0	17.9	0.0	0.0000	0.0	0.0	0.0	0.0	0.0	0.0000
0.1	82.1	17.9	10.7	4.3	0.0021	0.0	0.0	0.0	0.0	4.3	0.0017
0.2	85.7	14.3	7.1	4.2	0.0021	0.0	0.0	0.0	0.0	8.3	0.0020
0.3	89.3	10.7	3.6	4.0	0.0021	0.0	0.0	0.0	0.0	12.0	0.0180
0.4	89.3	10.7	3.6	4.0	0.0021	0.0	0.0	0.0	0.0	12.0	0.0180
0.5	89.3	10.7	3.6	4.0	0.0021	0.0	0.0	0.0	0.0	12.0	0.0180

TABLE VI
KINECT EXERCISE 3

TE (s)	TP (%)	FP (%)	FN (%)	ESS		LSS		EES		LES	
				%	MAE (m)	%	MAE (m)	%	MAE (m)	%	MAE (m)
0.0	45.5	45.5	45.5	0.0	0.0	0.0	0.000	0.0	0.000	0.0	0.000
0.1	59.1	31.8	31.8	0.0	0.0	15.4	0.083	0.0	0.000	7.7	0.037
0.2	81.8	9.1	9.1	0.0	0.0	22.2	0.130	5.6	0.018	16.7	0.120
0.3	86.4	4.5	4.5	0.0	0.0	26.3	0.160	5.3	0.018	15.8	0.120
0.4	90.9	0.0	0.0	0.0	0.0	25.0	0.160	10.0	0.046	15.0	0.120
0.5	90.9	0.0	0.0	0.0	0.0	25.0	0.160	10.0	0.046	15.0	0.120

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2019.2907483, IEEE Transactions on Neural Systems and Rehabilitation Engineering

TABLE VII
QUALISYS EXERCISE 1

TE (s)	TP (%)	FP (%)	FN (%)	ESS		LSS		EES		LES	
				%	MAE (m)	%	MAE (m)	%	MAE (m)	%	MAE (m)
0.0	81.0	19.0	19.0	0.0	0.0	0.0	0.0000	0.0	0.0	0.0	0.000
0.1	88.1	11.9	11.9	0.0	0.0	2.7	0.0037	0.0	0.0	5.4	0.044
0.2	95.2	4.8	4.8	0.0	0.0	2.5	0.0037	0.0	0.0	12.5	0.043
0.3	97.6	2.4	2.4	0.0	0.0	2.4	0.0037	0.0	0.0	14.6	0.045
0.4	97.6	2.4	2.4	0.0	0.0	2.4	0.0037	0.0	0.0	14.6	0.045
0.5	100.0	0.0	0.0	0.0	0.0	2.4	0.0037	0.0	0.0	16.7	0.048

TABLE VIII
QUALISYS EXERCISE 2

TE (s)	TP (%)	FP (%)	FN (%)	ESS		LSS		EES		LES	
				%	MAE (m)	%	MAE (m)	%	MAE (m)	%	MAE (m)
0.0	45.0	55.0	55.0	0.0	0.000	0.0	0.000	0.0	0.000	0.0	0.000
0.1	80.0	20.0	20.0	12.5	0.027	12.5	0.024	6.2	0.023	12.5	0.044
0.2	100.0	0.0	0.0	10.0	0.027	10.0	0.024	5.0	0.023	30.0	0.054
0.3	100.0	0.0	0.0	10.0	0.027	10.0	0.024	5.0	0.023	30.0	0.054
0.4	100.0	0.0	0.0	10.0	0.027	10.0	0.024	5.0	0.023	30.0	0.054
0.5	100.0	0.0	0.0	10.0	0.027	10.0	0.024	5.0	0.023	30.0	0.054

TABLE IX
QUALISYS EXERCISE 3

TE (s)	TP (%)	FP (%)	FN (%)	ESS		LSS		EES		LES	
				%	MAE (m)	%	MAE (m)	%	MAE (m)	%	MAE (m)
0.0	65.0	35.0	35.0	0.0	0.0	0.0	0.00	0.0	0.00	0.0	0.00
0.1	95.0	5.0	5.0	0.0	0.0	21.1	0.11	10.5	0.06	0.0	0.00
0.2	100.0	0.0	0.0	0.0	0.0	20.0	0.11	10.0	0.06	5.0	0.03
0.3	100.0	0.0	0.0	0.0	0.0	20.0	0.11	10.0	0.06	5.0	0.03
0.4	100.0	0.0	0.0	0.0	0.0	20.0	0.11	10.0	0.06	5.0	0.03
0.5	100.0	0.0	0.0	0.0	0.0	20.0	0.11	10.0	0.06	5.0	0.03

to the common low range of motion, instability and variance of the patient movements, as shown in Figures 10 and 11. The accuracy of the joint positions can be unreliable due to occlusion and contain a lot of jitter.

Tables IV, V, VI shows the performance of the segmentation algorithm on the Kinect patient dataset, where it manages to achieve correct segmentation for 90% of the data with a time error of ≈0.3 seconds. There are more false positives in the arm to side and arm to front exercises than the twist exercise. This is due to the similar arm movements performed during these exercises. The MAE of all exercises are negligible which suggests a time error of at least 0.5 seconds and likely larger are acceptable.

Tables VII, VIII, IX show the performance of the segmentation algorithm segmenting joint exercise data of healthy participants performing the same exercises captured by Qualisys, a marker-based motion capture system with a reported sub-millimetre accuracy [30]. The algorithm achieves 100% correct segmentation on all exercises within 0.5 seconds. The MAE for all exercises at different time errors is negligible.

Figure 11 depicts a patient performing the arm to front exercise with the accompanying plot depicted in Figure 12. The left image in Figure 11 shows the patient's starting position; this is captured at ≈30 seconds in Figure 12. The right image in Figure 11 shows the patient's arm extended to the highest achievable height, captured at ≈34 seconds in Figure 12. This demonstrates the robustness of the algorithm as the exercise query expects the arm to be extended to 90 degrees in the sagittal plane. Each of these repetitions has a low range of motion but still achieves good segmentation. Note that at ≈25 seconds the algorithmic segment is early to segment, due to a segment candidate being produced as the arm's velocity slows down. The ground truth repetition start and end rectangles have different widths because the start and end of a repetition is often ambiguous [6]; faster and smoother movements usually result in a smaller ground truth rectangle.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2019.2907483, IEEE Transactions on Neural Systems and Rehabilitation Engineering

Fig. 10. A stroke patient performing the twist exercise.

Fig. 11. A stroke patient performing the arm to front exercise.

Figure 10 shows a patient performing the twist exericse. Smoother motions and a larger range of motion are generally found in this exercise as the paretic limb is grasped with the healthy limb.

Figure 9 plots the direction of the arm in the Y axis for the arm to side exercise, at ≈10 seconds the algorithm incorrectly segments the repetition as the patient's arm dips down during the exercise.

Figure 13 shows the alignment between the candidate and query sequences for exercises arm to front (top) and twist (bottom). In the top image the range of motion of the candidate is lower than the query but DTW still finds a suitable alignment.

The results show that even for people with limited movement, we have achieved good segmentation accuracy of exercise repetitions. All but one of the segmentation algorithms highlighted in the literature review have tested their algorithms only on healthy participants [9]–[12], [15]–[19]. Lin and Kulic [6] tested their algorithm on 4 total joint replacement patients undergoing lower-body rehabilitation and they achieved 79% segmentation accuracy whereas our approach achieved 86.4-89.3% accuracy within the same 0.3 seconds time error threshold. We also found the MAE of late and early segmentations to be negligible even with a 0.5 second TE, which gave a correct segmentation percentage of 89.3%-95.8%. They mention that "DTW provides an accurate method of segmentation that is robust against temporal variations, but is too computationally expensive to be employed on-line.", however we have presented a real-time DTW segmentation algorithm. Chaun-Jun, et al. [11] tested their algorithm on rehabilitation exercises but with healthy participants. Due to the variability of stroke patient movements, it is important that segmentation algorithms are tested on real clinical data of rehabilitation exercises.

3) Parameter Evaluation: The parameters described in section III-F were set via the following process. The range of motion of stroke patient repetitions were measured and compared with healthy subjects to establish an appropriate baseline. It was found that setting the minimum scale percentage to 0.08, i.e. 8%, was required to ensure that stroke patient repetitions with the lowest range of motion could be adequately segmented. The segmentation threshold (ST) is the maximum value the total cost of a normalised subsequence can be from the normalised query to be considered for segmentation. This total cost is essentially a measure of the abnormality in the movements between the subsequence and query; therefore a total cost below the segmentation threshold indicates that the patient's movements are an attempted repetition of the exemplar exercise. The distance multiplier (DM) and distance base (DB) adjust the total cost thresholds, as described in equation 8, for each of the motion features based on their rate of change, i.e. motion features that exhibit the most movement. The total cost of each motion feature in a subsequence must be lower than their total cost threshold to be considered for segmentation.

A parameter evaluation was performed on parameters ST, DM and DB (described in Section III-F) using grid search with a step size of 0.1 within the range 0 to 1. By evaluating the parameter space using a linear step size we can see how the algorithm's accuracy changes with respect to the changes in the parameter values. Each parameter combination was cross-validated over CMU exercises: jumping jacks, squat and side twist. To determine accuracy we use F_1 (Acc_{F_1}) score presented in [31] and shown in equation 9.

$$\frac{2 \cdot TP}{2 \cdot TP + FN + FP} \tag{9}$$

The parameter combinations with the highest F_1 accuracy are presented in Table X. TP and FP segments are presented as percentages of the total GT segments. Given the top five combinations, the ST values range from 0.1 to 0.3, DM range from 0 to 0.1 and DB range from 0.3 to 0.4. To further understand how the F_1 accuracy changes with respect to these parameters, we have graphed three plots (Figures 14, 15 and 16) with each plot showing a representative DM and DB value combination that achieves the best, average and worst accuracy

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2019.2907483, IEEE Transactions on Neural Systems and Rehabilitation Engineering

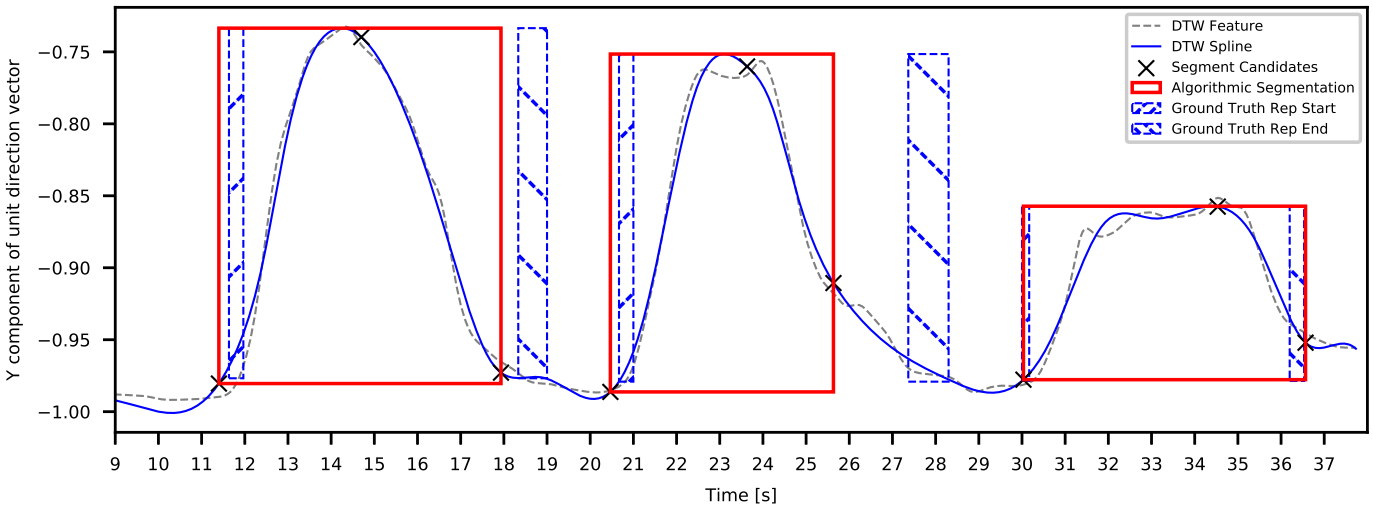


Fig. 12. Plot showing the algorithmic segmentations of a stroke patient performing the Arm to Front exercise taken from GRASP, a stroke rehabilitation manual.

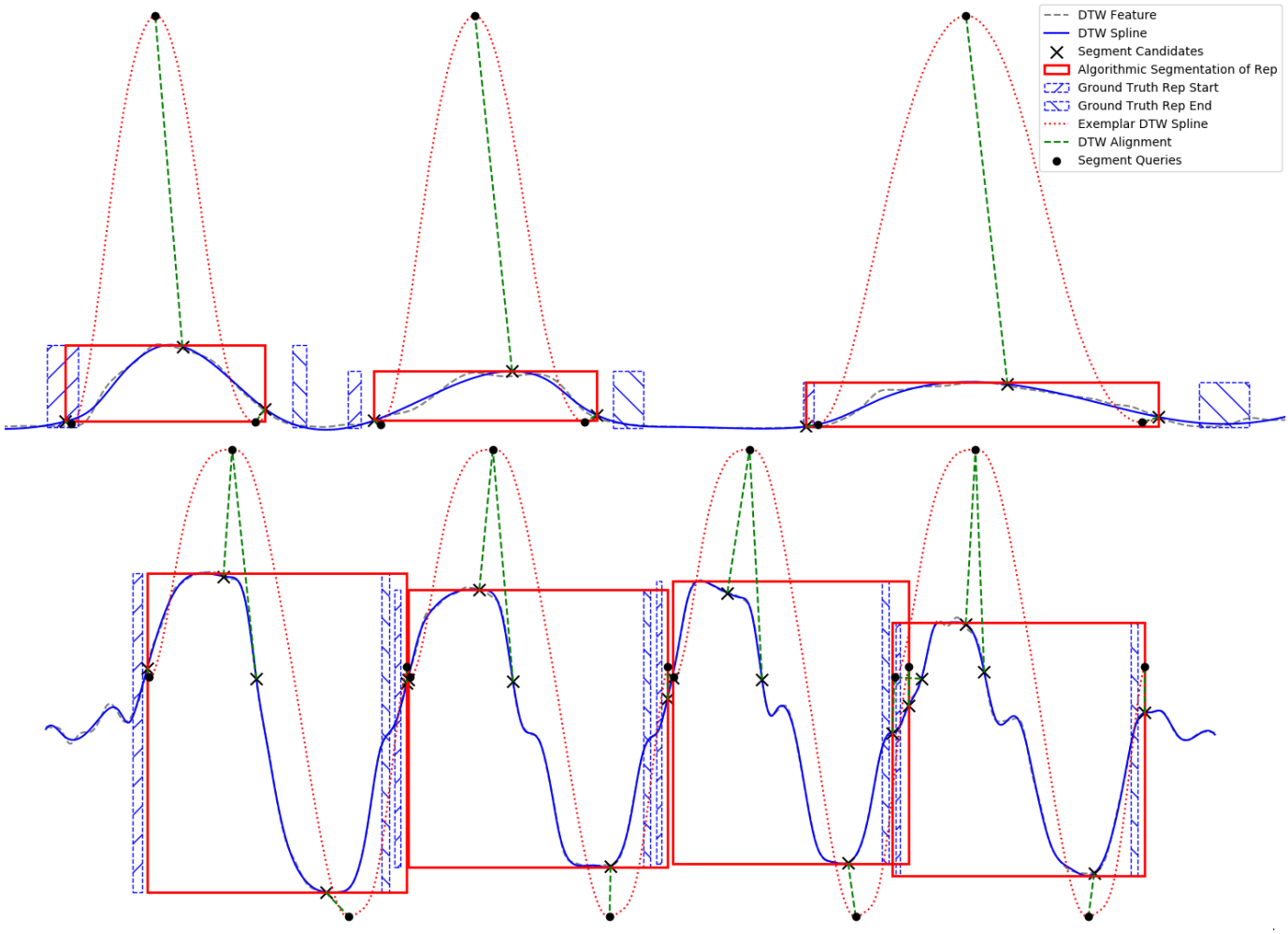


Fig. 13. Plots showing the algorithmic segmentations and DTW alignments of a stroke patient performing the Arm to Side and Twist exercise taken from GRASP, a stroke rehabilitation manual. The exemplar repetition was performed by a healthy subject.

respectively, and plotted the F_1 accuracy with respect to the ST parameter value.

A greater value of ST results in candidate sequences that are less similar to the query sequence being treated as a potential segmentation, but with suitable DM and DB values potential segmentations that would be considered as a FP are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2019.2907483, IEEE Transactions on Neural Systems and Rehabilitation Engineering

TABLE X
PARAMETER EVALUATION RESULTS PRESENTING THE TOP 5 PARAMETER COMBINATIONS.

Row	F_1	TP	FP	ST	DM	DB
1	0.77	0.85	0.36	0.1	0.1	0.3
2	0.77	0.85	0.36	0.2	0.1	0.3
3	0.75	0.85	0.4	0.3	0.1	0.3
4	0.75	0.85	0.4	0.1	0	0.4
5	0.75	0.85	0.4	0.2	0	0.4

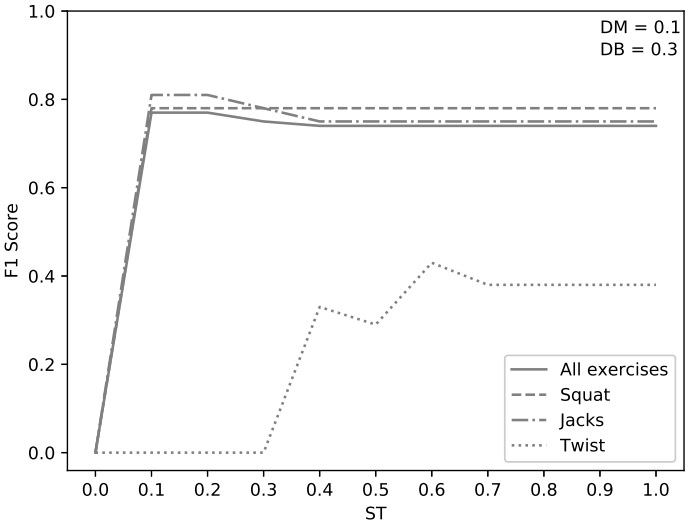


Fig. 14. Plot showing the change in the F_1 score with respect to the ST parameter and given the best (highest F_1 score) DM and DB parameter combination.

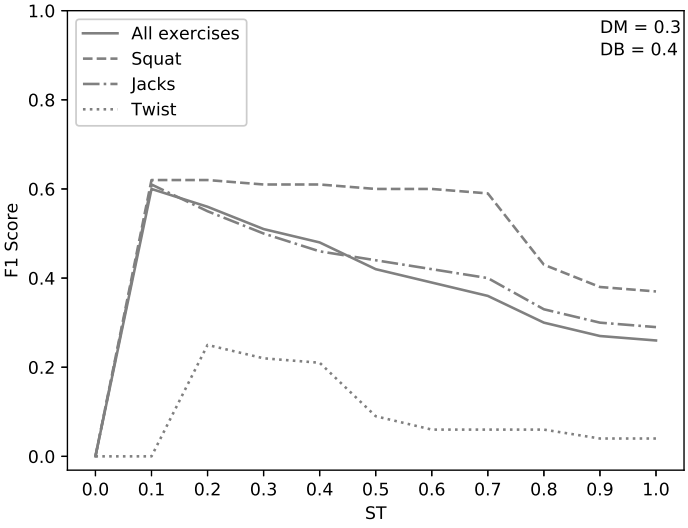


Fig. 15. Plot showing the change in the F_1 score with respect to the ST parameter and given an average (mean F_1 score) DM and DB parameter combination.

mostly rejected. This can be seen in Figure 14 whereby the F_1 accuracy does not decrease much as ST increases to a value of 1. As greater values are used for DM and DB the number of FP segments increases resulting in a lower F_1 accuracy, as can be seen in Figure 15. Using very low values for DM and DB results in fewer FP segments but also fewer TP segments

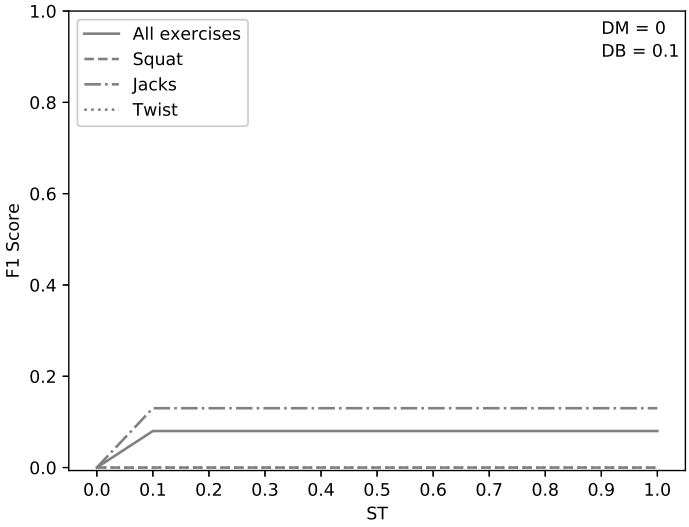


Fig. 16. Plot showing the change in the F_1 score with respect to the ST parameter and given the worst (lowest F_1 score) DM and DB parameter combination (excluding combinations with an F_1 score of zero).

and thus a low F_1 accuracy, as can be seen in Figure 16.

V. CONCLUSION

We have presented an algorithm for segmenting exercise repetitions in real-time. This approach addresses the limitations of previous approaches in that it requires only a single exemplar and has shown robustness to repetitions with low range of motion, instability in the movements and noise in the sensor data. The algorithm was evaluated on 10 subjects performing 3 exercises from a publicly available dataset (CMU) and we showed that it was capable of segmenting the repetitions. Further evaluation was performed on our own datasets of a healthy population and a stroke population performing stroke rehabilitation exercises. We showed that the algorithm correctly segments all the healthy population exercise repetitions within 0.5 seconds and the stroke patient exercise repetitions to 90% TP segments within 0.3-0.4 seconds. Our next step is to develop a pose estimation algorithm designed for clinical use to combine with this new segmentation algorithm, in order to enable accurate real-time feedback for stroke patients undergoing rehabilitation.

REFERENCES

[1] A. Jaume-I-Capó, P. Martinez-Bueso, B. Moya-Alcover, and J. Varona, "Interactive rehabilitation system for improvement of balance therapies in people with cerebral palsy," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 2, pp. 419–427, 2014.

[2] B. Lange, S. Koenig, E. McConnell, C.-Y. Y. Chang, R. Juang, E. Suma, M. Bolas, A. Rizzo, E. McConnell, E. Suma, M. Bolas, A. Rizzo, R. Juang, C.-Y. Y. Chang, R. Juang, E. Suma, M. Bolas, and A. Rizzo, "Interactive game-based rehabilitation using the Microsoft Kinect," *Ieee Virtual Reality Conference 2012 Proceedings*, vol. 34, pp. 170–171, 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6180935%5Cn%3CGo to ISI%3E://WOS:000317006300073%5Cnhttp://www.ncbi.nlm.nih.gov/pubmed/22494437%5Cnhttp://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6180935%5Cnhttp://informahealthc

[3] Y. J. Fan, Y. H. Yin, L. D. Xu, Y. Zeng, and F. Wu, "IoT-based smart rehabilitation system," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1568–1577, 2014.

[4] Y. J. Chang, W. Y. Han, and Y. C. Tsai, "A Kinect-based upper limb rehabilitation system to assist people with cerebral palsy," *Research in Developmental Disabilities*, vol. 34, no. 11, pp. 3654–3659, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.ridd.2013.08.021

[5] "MindMotion Pro." [Online]. Available: https://www.mindmaze.com/mindmotion/

[6] J. F. S. Lin and D. Kulic, "Online segmentation of human motion for automated rehabilitation exercise analysis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 1, pp. 168–180, 2014.

[7] C. J. L. Murray and e. al, "Disability-adjusted life years (DALYs) for 291 diseases and injuries in 21 regions, 1990-2010: A systematic analysis for the Global Burden of Disease Study 2010," *The Lancet*, vol. 380, no. 9859, pp. 2197–2223, 2012.

[8] V. L. Feigin, M. H. Forouzanfar, R. Krishnamurthi, G. A. Mensah, D. A. Bennett, A. E. Moran, R. L. Sacco, L. Anderson, M. O. Donnell, N. Venketasubramanian, S. Barker-collo, C. M. M. Lawes, W. Wang, Y. Shinohara, E. Witt, and M. Ezzati, "Global Burden of Disease Study 2010," *The Lancet*, vol. 383, no. 9913, pp. 245–254, 2010.

[9] D. Gong, G. Medioni, and X. Zhao, "Structured Time Series Analysis for Human Action Segmentation and Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1414–1427, 2014.

[10] D. Wu, L. Pigou, P. J. Kindermans, N. D. H. Le, L. Shao, J. Dambre, and J. M. Odobez, "Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1583–1597, 2016.

[11] C.-J. Su, C.-Y. Chiang, and J.-Y. Huang, "Kinect-enabled home-based rehabilitation system using Dynamic Time Warping and fuzzy logic," *Applied Soft Computing*, vol. 22, no. November 2010, pp. 652–666, 2014. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1568494614001859

[12] C. M. De Souza Vicente, E. R. Nascimento, L. E. C. Emery, C. A. G. Flor, T. Vieira, and L. B. Oliveira, "High performance moves recognition and sequence segmentation based on key poses filtering," *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016*, 2016.

[13] B. Krüger, J. Tautges, A. Weber, and A. Zinke, "Fast Local and Global Similarity Searches in Large Motion Capture Databases," *Eurographics / ACM SIGGRAPH Symposium on Computer Animation*, pp. 1–10, 2010. [Online]. Available: http://portal.acm.org/citation.cfm?id=1921427.1921429

[14] J. Baumann, R. Wessel, B. Krüger, and A. Weber, "Action Graph: A Versatile Data Structure for Action Recognition," *International Conference on Computer Graphics Theory and Applications (GRAPP)*, 2014.

[15] B. Krüger, A. Vögele, T. Willig, A. Yao, R. Klein, and A. Weber, "Efficient Unsupervised Temporal Segmentation of Motion Data," *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 797–812, 2017.

[16] Q. Wang, G. Kurillo, F. Ofli, and R. Bajcsy, "Unsupervised Temporal Segmentation of Repetitive Human Actions Based on Kinematic Modeling and Frequency Analysis," *Proceedings - 2015 International Conference on 3D Vision, 3DV 2015*, no. 1111965, pp. 562–570, 2015.

[17] J. F. e. S. Lin, V. Joukov, and D. Kulic, "Human motion segmentation by data point classification," *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, vol. 2014, pp. 9–13, 2014.

[18] M. Devanne, H. Wannous, P. Pala, S. Berretti, M. Daoudi, and A. Del Bimbo, "Combined shape analysis of human poses and motion units for action segmentation and recognition," *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pp. 1–6, 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7284880/

[19] J. Shan and S. Akella, "3D Human Action Segmentation and Recognition using Pose Kinetic Energy," *Coitweb.Uncc.Edu*, 2014. [Online]. Available: http://coitweb.uncc.edu/ sakella/papers/ShanAkellaARSO2014.pdf

[20] O. Arikan, "Compression of motion capture databases," *Tog*, vol. 25, p. 890, 2006.

[21] D. Sart, A. Mueen, W. Najjar, E. Keogh, V. Niennattrakul, and E. Keogh, "Accelerating dynamic time warping subsequence search with GPUs and FPGAs," *Proceedings - IEEE International Conference on Data Mining, ICDM*, no. December, pp. 1001–1006, 2010.

[22] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a lie group," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 588–595, 2014.

[23] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition," *Journal of Visual Communication and Image Representation*, vol. 25, pp. 24–38, 2014.

[24] A. Fod, M. J. Matari, and O. C. Jenkins, "Automated derivation of primitives for movement classification," *Autonomous Robots*, vol. 12, no. 1, pp. 39–54, 2002.

[25] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[26] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, "Searching and mining trillions of time series subsequences under dynamic time warping," *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 262–270, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2339576%5Cnhttp://practicalquant.blogspot.com/2012/10/mining-time-series-with-trillions-of.html

[27] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, no. 1, pp. 67–72, feb 1975. [Online]. Available: http://ieeexplore.ieee.org/document/1162641/

[28] "CARNEGIE MELLON UNIVERSITY GRAPHICS LAB: CMU Motion Capture Database." [Online]. Available: mocap.cs.cmu.edu

[29] Janice Eng and Jocelyn Harris, "GRASP (Graded Repetitive Arm Supplementary Program)." [Online]. Available: http://neurorehab.med.ubc.ca/grasp/

[30] Qualisys, "Qualisys Engineering." [Online]. Available: https://www.qualisys.com/applications/engineering/

[31] J. F. S. Lin, M. Karg, and D. Kulić, "Movement Primitive Segmentation for Human Motion Modeling: A Framework for Analysis," *IEEE Transactions on Human-Machine Systems*, 2016.