Automatic Human Motion Segmentation and Identification using Feature Guided HMM for Physical Rehabilitation Exercises

Jonathan Feng-Shun Lin and Dana Kulić

***Abstract*— Fast and accurate motion segmentation and identification methods are required to enable real-time assessment and feedback for physical rehabilitation. Exercise motions exhibit cyclic patterns that can be characterized by simple features, such as zero-velocity crossings or velocity peaks. In this paper, these features are used as framing windows for simultaneous motion segmentation and identification via Hidden Markov models. Comparisons to other segmentation methods show that feature guiding increases the segmentation accuracy and greatly reduces the runtime needed to perform the segmentation.**

I. INTRODUCTION

Rehabilitation is a component of modern health care that is concerned with the development, maintenance and restoration of body movement and function. Range of motion is restored, to the greatest possible extent, though a regime of physiotherapist-prescribed exercises. Typically, the physiotherapist will observe the patient while they perform these exercises, which can be a time consuming task. Instead, this supervision can be facilitated by automated means.

Computer-assisted rehabilitation is a large and growing field. A patient's exercise motions can be measured via motion capture or accelerometers and gyroscopes [1], [2]. Relevant motion segments can then be identified, extracted and assessed [3], [4]. These systems enable physiotherapists to assess the patient's performance quantitatively, as well as opening doors to home rehabilitation [5], [6] and robot-assisted rehabilitation [7], [8].

A key problem that needs to be addressed in order to implement such automated rehabilitation systems is the automatic extraction of motion segments from continuous time series data. Segmentation is the process of locating logical points to break up this sequence into smaller components, termed *motion primitives*, to facilitate motion identification.

Early work in motion segmentation looked at zero-velocity crossings (ZVC) as points where joint segments changed directions [9], [10]. For recognition, temporal variations in human motion make direct comparison between motions difficult. This led to the development of alignment techniques such as Dynamic Time Warping (DTW), where the observation data is selectively warped to be time-aligned to template motions [11], [12], [13]. Later works considered the use of stochastic methods such as Hidden Markov models (HMMs) for both segmentation and recognition [14], [15], [16].

This paper proposes a 2-tier system for simultaneous segmentation and recognition. During the training phase, the exemplar motions are used to form feature and HMM templates. During the segmentation phase, the observed data is first scanned for features matching the trained templates. When potential segmentation points are found, HMM template matching is applied to determine the best windowing for the segment and to filter out similar looking motions.

II. PROPOSED APPROACH

In order to compare the continuous time series data against a template, candidate segments are required. A simple approach is to generate candidates by sliding a fixed-size window over the data. However, this approach can be time consuming, as it requires many comparisons against the template and assumes that the motion duration is constant across trials, which may not be the case during rehabilitation exercises. Instead, this paper proposes a feature searching method to quickly estimate possible windowing edges prior to template matching. The exemplar motion is scanned for key features, such as velocity peak positions or ZVCs, as a way to approximate a potential interest region in the observed data. Rehabilitation exercise motions are well suited to such an approach, as they are characterized by regular patterns of flexion and extension cycles. HMM template matching is then used to identify the exact boundary and motion.

A. Review of Hidden Markov Models

HMM [17] is a stochastic dynamic model where the dynamic process is represented as the evolution of a hidden (unobservable) state, with observable state-dependent outputs. The state has the Markov property, where the next state depends solely on the current state. HMM stores its model information as a set of three variables: the initial state distribution, π, the state transition probability matrix, A, and the observation distribution, B. For continuous observation variables such as human motion data, Gaussian or mixture of Gaussians distributions are used.

Given a set of exemplar motions, the Baum-Welch algorithm [17] is utilized to train the model. To assess the similarity between a trained model and a new observation sequence, the forward algorithm [17] is utilized. The forward algorithm calculates the likelihood that the observation data could have been generated by the model.

B. Template training

Template training is carried out off-line, prior to the start of on-line data processing. We assume that n (typically 3-6) examples of the motion to be segmented are available as exemplars for training. The template training consists of two

J. F. Lin and D. Kulić are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (email: jf2lin@uwaterloo.ca; dkulic@ece.uwaterloo.ca)

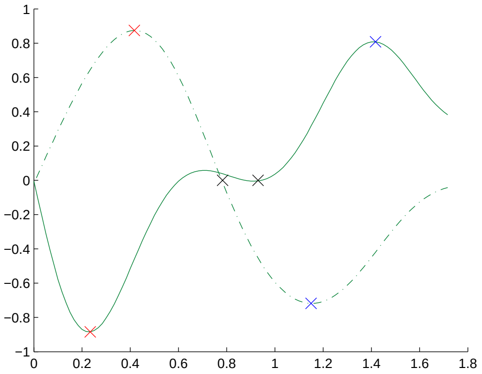

Fig. 1. An exemplar template with two significant DoFs, with its features marked. All exemplars are expected to start and end with a ZVC, and have a peak, ZVC and a peak in between. In this example, a ZVC point is not found at the end of the exemplar. The feature model automatically adds a terminating ZVC in this case.

parts: feature template extraction and HMM training. In this paper, the velocity peak position is used as the indicating feature. It is also assumed that ZVCs start and terminate any given primitive [10], and are added to the feature model if not found in the exemplars.

As the motion can consist of multiple degrees of freedom (DoFs) moving simultaneously, during feature extraction, the DoFs contributing to the motion, termed *significant DoFs*, are first selected by calculating the standard deviations of the exemplar joint angles and grouping them via 2-means clustering. The DoFs that are in the cluster with the higher centroid are assumed to be significant.

Each exemplar is expected to have a ZVC near its midpoint (Black points in Figure 1). Between the start of the exemplar motion and the ZVC, there is a peak point (red points in Figure 1). Similarly, between the ZVC and the end of the motion, there will be another peak point (blue points in Figure 1). The locations and magnitudes of these feature points are stored as the feature template.

The exemplars are also used to train HMM models for each motion type. An 8-state left-right HMM is used, trained using the Baum-Welch algorithm. The Gaussian observation functions are initialized by *k*-means clustering. As part of the training procedure, the optimal threshold for recognition T_R is also determined, via leave-one-out cross-validation.

C. Feature Guided Segmentation

During the on-line processing phase, a small sliding window is first passed over the observation data, noting the local peak values and ZVCs. At each time step, the current recorded peak magnitude and ZVCs are compared against the known templates. If there is a match, then a potential segment point has been located. Several ZVC points before and after the peaks are used as possible windowing bounds. To reduce the computational cost, the velocity magnitudes are checked between all pairs of ZVCs. If the maximum velocity is small (i. e. when the subject is stationary), it suggests that no motion is observed and the ZVC further from the detected feature peaks is removed. The remaining ZVCs are used for HMM template matching. The template and window edge combination that results in the highest log-likelihood value over the threshold T_R is declared a segment. Following HMM template matching, the recorded peak magnitude and ZVCs are reset, and the feature search resumes at the next time step.

III. EXPERIMENTS

This section outlines the data collection process and the algorithm performance in terms of segmentation accuracy and computational cost.

In order to compare the proposed approach to existing work, a ZVC method, described in [9] was implemented. Segmentation points are declared when the squared sum of the angular velocities over a small sliding window is below a threshold. To reduce the impact of noise-induced stray ZVCs, an empirically determined minimum length of time must pass before the next segmentation point can be declared.

A DTW algorithm was also implemented. DTW mapping matrices were constructed between exemplar data of a given primitive, to time-align them, using dynamic programming and Euclidean distances [13], [18]. Once aligned, the exemplar motions were all averaged together to create a template. The observation data was then time-aligned to the templates. A segment was considered a primitive candidate as long as the Euclidean distance between a given template and the observation data was maintained below some empirically-derived threshold, even if only part of the observation data was passed in. If the observation data began to deviate from the template (that is, the Euclidean distance does not change or increases for a few timesteps), a new window was declared as an attempt to find a new starting window.

A fixed-sliding window HMM was also implemented, to assess the impact to the feature guiding. HMM construction for the fixed sliding window was identical as the feature guided HMM. Window length used was approximated from exemplar templates. Segmentation points were declared on local maximas of the likelihood.

All processing and algorithmic implementation were done in MATLAB 7.11.0. The HMM functions were implemented with Kevin Murphy's HMM MATLAB Toolbox [19]. In the MATLAB figures, segmentation boxes are coloured and offset solely for viewing clarity.

A. Data Collection

The proposed algorithm was applied to a set of motion capture data of a single human male right leg, performing various motions:

- Single leg extensions. These are used as exemplar motions for template training. These motions are 2 seconds long. Three motions were used for training purposes.
- Three sequential leg extensions. This sequence is used for segmentation, and can be characterized as well-defined motions. This sequence is 7 seconds long.

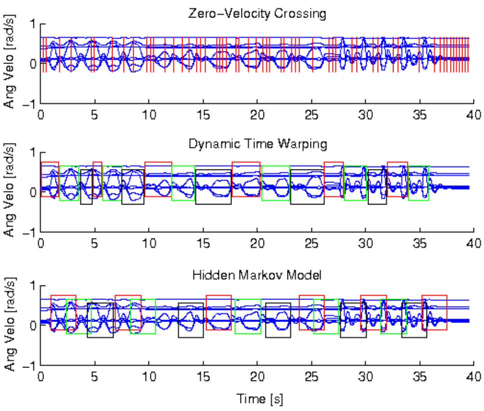

Fig. 2. Segmentation results of ZVC, DTW and fixed-window HMM methods compared, on a 9 DoF system with random leg motion. Knee joint velocity shown.

- Random leg motion sequence. This sequence is also used for segmentation, and can be characterized as poorly-defined motions. This sequence consists of three parts: a set of leg extensions, then a set of leg extensions, with the leg held out when fully extended, then a series of random leg kicking motions. This sequence is 40 seconds long.

The data was collected on a MotionAnalysis motion capture system, with a sampling frequency of 62 Hz. The data was filtered with a 5^{th} order single-pass Butterworth filter. The Cartesian coordinates of the hip, knee and ankle were used, for a total of 9 DoFs.

B. Comparison of Segmentation Performance

Four methods were compared: ZVC, DTW, fixed-window HMM, and the feature guided HMM method proposed here. ZVC performance was the worst of all the algorithms tested. Since ZVC does not use a template to guide its segmentation, extra segmentation points were declared in all the data examined. As shown in Figure 2, this makes it difficult to determine which region is the motion of interest.

As can be seen in Figure 2, DTW performs very well. However, in many instances, the centre of the algorithmic segment and the actual primitive do not align. Often, DTW was not able to discard the static periods between motions, such as at 15 seconds in Figure 2. A false positive was also identified at the end of the observation data.

Similar to DTW, the HMM determined segmentation windows were usually not centred properly on each curve, and the fixed window was sometimes also not appropriately sized for the motion. Overlapping segmentation windows were declared, sometimes centred between motions (7 seconds, on Figure 2). It also misidentifies multiple curves as a single leg extension past the 30 second mark. Overall, the segmentation results were noticeably worse than DTW.

Using the features to guide the segmentation improved the HMM results significantly, as can be seen in Figures 3 and 4. There were still some segments where misidentification occured, such as at 15 seconds, but this may be fixable with some adjustment to the likelihood thresholds. This method also segments the motions past 30 seconds in a consistent fashion, which was an improvement over the fixed-window HMM algorithm. It should be noted that, although the motions shown in Figure 3 and 4 are similar, the reported LLs are different, and the LL thresholds were manually tuned to obtain the shown results. This may be due to changes in leg motion speed and movement pattern between the two motion sets.

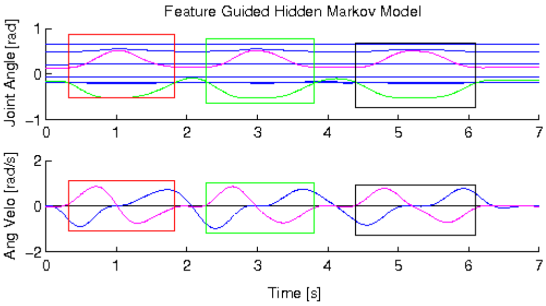

Fig. 3. Segmentation results of feature guided HMM, on 9 DoF data, with joint angle and angular velocity profiles. This figure shows the short dataset.

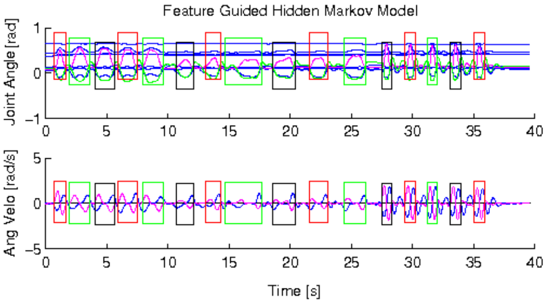

Fig. 4. Segmentation results of feature guided HMM, on 9 DoF data, with joint angle and angular velocity profiles. This figure shows the long dataset.

C. Quantitative Segmentation Results

A segmentation accuracy assessment metric was developed in order to systemically compare the segmentation results. Manual segmentation was used as 'ground truth' for segmentation. The algorithmic segmentation points were assigned one of the following status:

- Correct. An algorithmic segment point falls within $\pm t_{error}$ of a manual segment point
- False positive. An algorithmic segment was declared when there is not one

TABLE I
SEGMENTATION ERROR ANALYSIS ON 9 DOF DATA FOR LONG SEQUENCE (MAXIMUM SCORE = 32)

Algorithm	Correct	False Positive	False Negative	Out-of-Bounds
ZVC	13	82	19	2
DTW	19	2	0	13
HMM (Fixed)	8	4	12	12
HMM (Feature)	30	0	0	2

TABLE II

TIMING ANALYSIS ON 9 DOF DATA

Algorithm	Training (s)	Short (s)	Long (s)
Observation data		7.25	39.60
ZVC		0.10	0.41
DTW	0.25	10.8	68.5
HMM (Fixed)	0.67	2.75	19.94
HMM (Feature)	0.86	0.49	2.50

- False negative. An algorithmic segment was not declared when there should be one
- Out of bound. When one of the algorithmic segment points is just outside its corresponding manual segment point's $\pm t_{error}$, but the other is marked correct

Each segment's two segment points are awarded points separately. That is, a completely correct segment could receive +2 correct, whereas a partially correct segment could receive +1 correct and +1 out-of-bound. t_{error} was set to 10 timesteps (0.16 seconds).

Table II shows that ZVC does very poorly in providing accurate segmentation, performing the worst of all the algorithms tested. Fixed window HMM also performs very poorly, implying that local likelihood maxima is an insufficient condition for good segmentation. DTW is generally more accurate than fixed-window HMM. HMM with feature searching segments the most accurately, out of the algorithms examined.

D. Timing Results

Since real-time functionality is an end-goal, timing analysis was also performed. The timing results are summarized in Table III. This table outlines the average length of time needed to complete the template training ('Training'), and perform segmentation on the three leg extension ('Short') and random sequences ('Long').

ZVC requires the least time, since it consists only of a numerical differentiation, and computation of the sum of squares of angular velocities. Comparatively, DTW requires significantly more calculations, as it needs to build a DTW grid at each time step. With larger datasets and more templates, DTW performance is expected to deteriorate further. Fixed-window HMM performs much better than DTW, suggesting that the forward algorithm is less computationally costly than the DTW grid. Augmenting the HMM routine with feature searching, to decrease the number of forward algorithm executions, improves the computation time significantly.

IV. CONCLUSIONS AND FUTURE WORKS

Feature guided HMM segments data with greatly improved accuracy over existing algorithms like DTW and fixed-window HMM. By utilizing feature points such as ZVCs and peak positions, a quick estimation of possible segmentation points can be found, and the closeness of match can be checked with HMM. This method is particularly well suited for rehabilitation applications, where motions are known apriori and are characterized by well defined features. The proposed approach also has less computational overhead than other algorithms considered, and is not significantly impacted by high dimensionality data.

We are currently working on applying the algorithm to a larger dataset to confirm performance against interpersonal variations and a larger number of templates. Another area of future research is to consider template scaffolding. The target users of this algorithm are patients undergoing physiotherapy. Such patients may not necessarily move in a similar fashion as healthy subjects, so a method to dynamically update the template could improve segmentation results.

REFERENCES

[1] K. Lorincz, B.-R. Chen, G. W. Challen, A. R. Chowdhury, S. P., P. Bonato, and M. Welsh, "Mercury: A wearable sensor network platform for high-fidelity motion analysis," in *Proc. ACM Conf. on Embedded Networked Sensor Systems*, 2009.

[2] H. Zhou and H. Hu, "Human motion tracking for rehabilitation - a survey," *Biomedical Signal Processing and Control*, vol. 3, no. 1, pp. 1–18, 2008.

[3] D. Kulić, G. Venture, and Y. Nakamura, "Detecting changes in motion characteristics during sports training," in *Int. Conf. IEEE Engineering in Medicine and Biology Society*, 2009, pp. 4011–4014.

[4] F. Zhou, F. De la Torre Frade, and J. K. Hodgins, "Aligned cluster analysis for temporal segmentation of human motion," in *IEEE Conf. Automatic Face and Gestures Recognition*, 2008.

[5] J. J. Baskett, J. B. Broad, G. Reekie, C. Hocking, and G. Green, "Shared responsibility for ongoing rehabilitation: a new approach to home-based therapy after stroke," *Clinical Rehabilitation*, vol. 13, pp. 23–33, 1999.

[6] H. Zhou, H. Hu, and Y. Tao, "Inertial measurements of upper limb motion," *Medical and Biological Engineering and Computing*, vol. 44, no. 6, pp. 479–487, 2006.

[7] T. Mouri, H. Kawasaki, Y. Nishimoto, T. Aoki, and Y. Ishigure, "Development of robot hand for therapist education/training on rehabilitation," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2007, pp. 2295–2300.

[8] A. M. Okamura, M. J. Mataric, and H. I. Christensen, "Medical and health-care robotics," *IEEE Robotics Automation Magazine*, vol. 17, no. 3, pp. 26–37, 2010.

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

[10] M. Pomplun and M. Matarić, "Evaluation metrics and results of human arm movement imitation," in *Proc. RAS Conf. Humanoid Robotics*, 2000.

[11] W. Ilg, G. H. Bakir, J. Mezger, and M. A. Giese, "On the representation, learning and transfer of spatio-temporal movement characteristics," *Int. J. Humanoid Robotics*, vol. 1, no. 4, pp. 613–636, 2004.

[12] E. J. Keogh and M. J. Pazzani, "Derivative dynamic time warping," in *SIAM Int. Conf. Data Mining*, 2001.

[13] E. J. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: A survey and empirical demonstration," *Data Mining and Knowledge Discovery*, vol. 7, pp. 349–371, 2003.

[14] J. Kohlmorgen and S. Lemm, "A dynamic hmm for on-line segmentation of sequential data," in *Proc. NIPS 2001: Adv. Neural Inform. Process. Syst.*, vol. 14, 2002, pp. 793–800.

[15] D. Kulić, W. Takano, and Y. Nakamura, "Online segmentation and clustering from continuous observation of whole body motions," *IEEE Trans. Robotics*, vol. 25, no. 5, pp. 1158–1166, 2009.

[16] F. Lv and R. Nevatia, "Recognition and segmentation of 3-d human action using hmm and multi-class adaboost," in *European Conf. on Computer Vision*, vol. 4, 2006, pp. 359–372.

[17] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[18] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Speech and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[19] K. Murphy, "Hidden markov model toolbox for matlab," University of British Columbia, 2005.