Movement Primitive Segmentation for Human Motion Modelling: A Framework for Analysis

Jonathan Feng-Shun Lin, Michelle Karg and Dana Kulić

***Abstract*— Movement primitive segmentation enables long sequences of human movement observation data to be segmented into smaller components, termed *movement primitives*, to facilitate movement identification, modelling, and learning. It has been applied to exercise monitoring, gesture recognition, human-machine interaction, and robot imitation learning. This paper proposes a segmentation framework to categorize and compare different segmentation algorithms considering segment definitions, data sources, application specific requirements, algorithm mechanics, and validation techniques. The framework is applied to human motion segmentation methods by grouping them into online, semi-online and offline approaches. Among the online approaches, distance based methods provide the best performance, while stochastic dynamic models work best in the semi-online and offline settings. However, most algorithms to date are tested with small datasets, and algorithm generalization across participants and to movement changes remains largely untested.**

I. Introduction

Motion segmentation is the process of identifying the temporal extents of movements of interest, breaking a continuous sequence of movement data into smaller components, termed *movement primitives* [1], and identifying the *segment points*, the starting and ending time instants of each movement primitive. If more than one type of movement is performed, *identification*, or *labelling*, of each segment with the appropriate movement type may also be required.

Movement primitive segmentation is used in many applications. In imitation learning, segmentation is used to isolate movement primitives from demonstrations provided to robots in order to teach them to perform complex tasks [2], [3], [4]. In exercise and physical rehabilitation, movement segments are used to isolate exercise repetitions to provide feedback on the quality of the exercise [5], [6]. For activity tracking, accurately identifying the type and amount of activity performed is of interest [7], [8].

Time-series segmentation is a difficult task. One difficulty in performing accurate segmentation is the large number of degrees of freedom (DOF) in the movement data. In addition, segmentation is made more difficult due to the variability observed in human movement. Motions can vary between individuals due to differing kinematic or dynamic characteristics and within a single individual over time [9], due to short term factors such as fatigue, or long term factors such as physiotherapy recovery or disease progression [10]. The participant may also start a subsequent movement before fully completing the previous one (*co-articulation*) [11], leading to hybrid or blended motion primitives. These factors introduce both spatial and temporal variability, which a robust segmentation algorithm must be able to handle. Lastly, some segmentation algorithms require labelled data for training, which can be time-consuming to generate.

Depending on the target application, different types of segmentation may be required. *Gesture* [12] or *activity recognition* [13], [14] refers to segmentation where multiple repetitions of the same motion type are considered to be one activity (label), and thus segments are declared when the label transitions from one activity to another. The focus of this paper is on *primitive segmentation*. Motion primitive segmentation is differentiated from activity recognition, as the algorithms are designed to provide more granular temporal detail by segmenting both repetitions of the same motion, as well as transitions between different motion types. Previous surveys have focused on activity recognition [12], [13], [14], primitive modelling [15], or action recognition and modelling applied to robotics [1]. Others elaborate on computer vision approaches for human movement recognition [16], [1], [17], [18], [19], [20].

This paper proposes a novel framework for segmentation algorithm analysis. The proposed framework provides a structure for considering the factors that must be incorporated when constructing a segmentation and identification algorithm. The lack of such a framework makes it difficult to compare different segmentation algorithms systematically, since different algorithms tend to be assessed against different criteria and with different procedures. The proposed framework provides the means to examine the impact of each algorithm component and allows for a systematic approach to determine the best algorithm for a given situation.

II. Framework

The proposed framework identifies five components that comprise any complete segmentation algorithm (Figure 1): (1) Segment definition (Section III): There is no common agreed-upon definition of a segment, and thus the segment definition must be clarified for each application; (2) Data collection (Section IV): Factors such as how the data are collected, as well as the availability of exemplar data or ground

J. Lin, M. Karg, and D. Kulić are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada. Contact: (jf2lin, mekarg, dana.kulic)@uwaterloo.ca.

DOI: 10.1109/THMS.2015.2493536

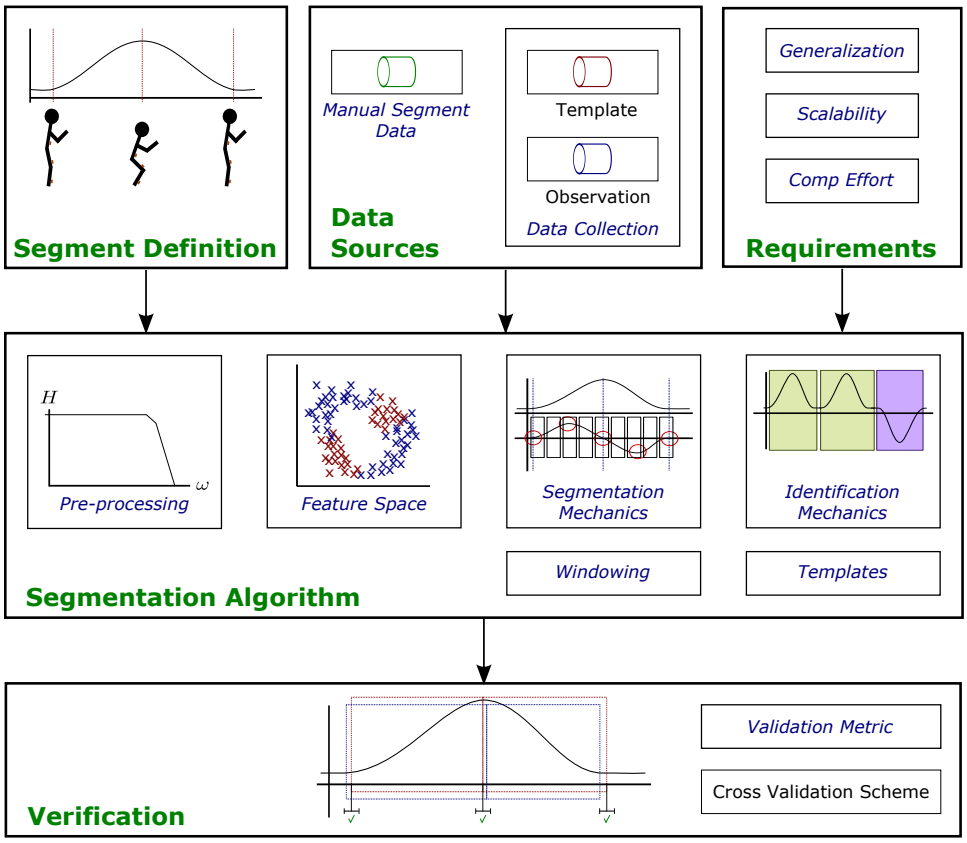

Fig. 1. Overview of the segmentation framework. The bold labels denote major sections, while italicised labels denote subsections.

truth data for algorithm training and verification, should be determined; (3) Application specific requirements (Section V): Once the source of data and the segment definition have been established, any application specific requirements on inter-subject or inter-primitive generalization, computational cost constraints and scalability requirements need to be defined; (4) Algorithm design (Section VI): Once the above factors are determined, the specific segmentation mechanism can be designed. The segmentation algorithm can be divided into four components: filtering and outlier rejection, feature space transformation, segmentation mechanism, and identification mechanism; (5) Verification (Section VII): Different validation schemes can serve to emphasize or obscure the performance of a given algorithm, and must be selected carefully.

This proposed framework supports a systematic way to construct a segmentation algorithm by ensuring critical details, such as the segment definitions and data sources, are determined early in the design process. The proposed framework also facilitates algorithm comparison and verification.

III. Segment Definition

Segment definitions refer to how the segment boundaries are characterized, to allow a human or algorithmic observer to identify the segment boundaries from the measured data. These definitions are often subjective, algorithm-dependent, application-driven, and tend to fall into three categories:

Physical boundaries: For movement applications, the definition of a segment typically refers to physical changes that occur when the movement starts or ends. These natural physical boundaries can be defined by joint movement direction changes [21], or contact changes such as at heel strike [22] or during object pickup [23]. These domain knowledge characteristics may be specific to a particular movement (*e.g.*, heel strike during gait) or may generalize to multiple motions (*e.g.*, joint movement direction changes). Relying only on contact changes limits the segmentation approach to movements that involve these types of physical interactions. For joint direction changes, it can be unclear which joint should be used, or how to segment if multiple joints are changing directions. Segments defined using joint angle direction changes can separate flexion and extension, or include both in a single primitive (Figure 2).

Derived metrics boundaries: The segment can also be defined by a change in a metric or derived signal. Segment boundaries can be signalled by changes in variance

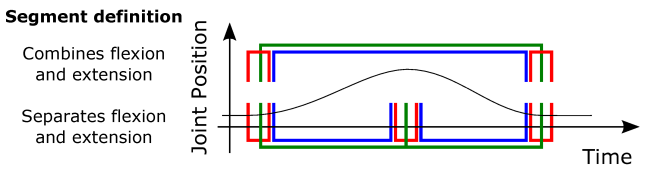

Fig. 2. The location of the segment edge region (red) and the within-primitive region (blue), based on the segment primitive definition used (green), illustrated on a joint angle time series. This image shows two alternate segment definitions: a segment definition with extension and flexion combined (top), and a segment definition where extension and flexion are considered separately (bottom).

[24], at metric thresholds [25], [26], or at hidden Markov model (HMM) state transitions, as determined by the Viterbi algorithm [27] (see Section VI-B.1 for details). Segments can then be determined by either unsupervised [24], [25] or supervised [27] algorithms. Unsupervised algorithms reduce the need for manual data labelling, and can identify segments similar to those denoted by domain experts [21], [28]. However, unsupervised segment identification may be less suitable for rehabilitation applications, where the segments of interest may be clinically defined and may not coincide with the derived metric.

Template boundaries: Segments can be defined based on a user-provided template. Template-based algorithms include template matching [29], dynamic time warping (DTW) [30], or classifiers [31], [32]. Defining the primitive by a template allows for maximum flexibility, allowing the user to define the template to suit requirements. This approach requires preparation time to generate the templates and generally requires more computationally intensive algorithms to handle the segmentation process.

IV. Data Sources

This section summarizes the main approaches for data collection and discusses feature space dimensionality.

A. Data Collection

Common sensory systems for human motion analysis utilize motion capture systems [33], [34], ambulatory sensors such as inertial measurement units (IMUs) [35], [36], or cameras [16], [37]. Other modalities such as electroencephalography (EEG) [38], [39] or electromyogram (EMG) have been used to study motion in kinesiology and biomechanics studies [40] but have only rarely been utilized for segmentation purposes.

Motion capture: Motion capture systems are considered the gold standard in biomechanics research; they rely on infrared cameras to determine the absolute positions of reflective markers placed on the body. These multi-camera systems are accurate for collecting gross movement but can suffer from marker swapping and marker occlusions [41], [42]. They tend to be expensive, require large amounts of space, and are time-consuming to set up, limiting their use to the laboratory. Algorithms applied to motion capture data appear in [25], [30], [43], [44], [45].

Cameras: Video and depth cameras have found widespread usage due to their price and size. Pose detection [16] and skeleton tracking algorithms [46] can be used to estimate joint angle data. Similar to the motion capture system, cameras require clear line-of-sight which limits applications to environments where occlusions are not a concern. Algorithms that rely on cameras include [47], [48], [49], [8].

Inertial measurement units: IMUs, consisting of accelerometers, gyroscopes, and magnetometers, are lightweight and inexpensive, and measure linear acceleration, angular velocity and orientation, respectively. The measured data can be used directly for segmentation, or converted to joint angles via the Composite [50] or Kalman filter [51], [52]. An IMU-based measurement system makes minimal assumptions about the deployment environment and does not require line-of-sight. However, IMUs suffer from integrational drift, leading to accuracy issues over time. Magnetometers are also ineffective indoors, where metallic objects, such as steel framing in walls, interfere with the sensors. Techniques that have been applied to raw IMU data include [31], [53], [8], [54], while techniques utilizing post-processed IMU data include [29], [32].

B. Manual Segment Data

Manually segmented data is required for training labels or as ground truth for algorithm verification. Various techniques have been employed to obtain *ground truth*, *manual segment*, or *labelled* data.

Video playback: In this approach, the ground truth is generated by having a human observer generate labels by observing video playback of the recorded data. The video can be collected simultaneously with data collection [44], [25] or by playback via regeneration from measurement data, such as animating motion capture markers [29]. An analyst observing the video indicates when segments begin and end, thus introducing subjectivity. Disadvantages of this approach are inaccuracies in the segment points caused by the expert's reaction speed, limitations in the viewing angle while displaying the movement, and effort.

Annotation: The manual segments can be generated by reviewing the collected data as a time-series graph and annotating directly the regions that contain motions of interest [53], [8]. It takes less time to generate the manual segments than with video playback, but this approach relies on an expert rater that can interpret the time-series data to extract the motions of interest [55].

Proxy sensors: Ground truth can be generated by a secondary data source. For example, gait cycles can be segmented by locating the impact acceleration [22] or when an accelerometer determines that the lower leg posture is parallel to gravity [56].

Counting primitive occurrences: It is possible to only identify the number of primitives that occur, either by collecting this information during data collection, or by video playback [54]. This is the fastest method to provide ground truth but also provides no information about the temporal segmentation accuracy of the algorithm.

C. Public Databases

Several movement databases exist for algorithm training and verification. The use of public databases reduces the effort needed for data collection and allows different algorithms to be compared using the same data. The following databases contain both temporal identification labels, and movement data from multiple participants performing multiple actions: (1) The Carnegie Mellon University (CMU) Multimodal Activity Database [57] contains temporally synchronized video, audio, motion capture and IMU data from 26 subjects cooking 5 different recipes in a kitchen. This dataset consists of common activities of daily living (ADL), with

significant variations in how each primitive performed. However, the participants were heavily instrumented, which may have impeded natural motion; (2) The Technische Universität München (TUM) Kitchen Data Set [58] contains video and motion capture data of 4 subjects performing kitchen tasks. The TUM dataset aimed to provide data that contained less intra-primitive variabilities than the CMU database [58]; (3) The University of Tokyo Full-body Movement Data Set [44] provides video and motion capture data of 1 subject performing 49 different types of full body motions for a total of 751 segments. This dataset provides well-defined full-body movements, but only contains data from 1 subject; and (4) the Yale Human Grasping Dataset [59] contains video data of 4 subjects performing housekeeping and machining tasks over 27 hours of hand movement tasks.

Other movement databases do not provide temporal segmentation data: the University of California [60], [61], the CMU Graphics Lab [62], the Hochschule der Medien [63], and the Kungliga Tekniska Högskolan [64]. A review of databases for computer vision can be found in [65].

V. Application Specific Requirements

The algorithm requirements for the specific application, such as template generalizability, scalability, and computational effort constraints, are considered in this section.

A. Generalizability

Generalizability refers to an algorithm's ability perform on data different from the training set. Two types of generalizability can be considered: (1) subject variability, subdivided into intra-[1] and inter-subject variability[2], and (2) inter-primitive variability.

Subject variability: [27], [29] and [32] have examined intra-subject and inter-subject variability. Stochastic techniques, such as the HMM [29], model the variability between exemplar motions inherently. Deterministic techniques, such as the support vector machine (SVM) [32], can be constructed with the exemplar data from multiple participants to improve generalizability. Aggregator techniques have been used for activity recognitions applications [7], [67], which can reduce the impact of overfitting from training data, but have only seen limited success in segmentation applications [32]. To date, few techniques successfully generalize between subjects that have very different motion characteristics while performing the same type of motion, such as training on healthy subjects and segmenting on rehabilitation subjects. This is a difficult task due to the differences between the two populations [29].

Inter-primitive variability: Inter-primitive variability considers when the training data are obtained from one set of movement primitives, while the test-set consists of a second set of unseen primitives. Algorithms built to segment based on domain knowledge features, such as segments that are defined via velocity characteristics [21], [22], or contact condition changes [23], can be robust against inter-primitive variability, as they define segment edge points share common characteristics across all primitives of interest. To reduce the reliance on domain expertise, learning approaches can be utilized to determine these common characteristics [32].

[1]Data from the same participant but at different instances. Variability is due to effects such as random muscle recruitment and fatigue [66].

[2]Data from a different set of participants not observed during training. In addition to the intra-personal effects, variability is due to effects such as stature and physiological differences [66].

B. Computational Effort and Causality

Another important consideration is whether the algorithm is capable of running online. This will be influenced by the computational complexity and runtime of the algorithm.

Runtime constraints come from two sources: (1) computational effort for online operation, or (2) the algorithm is non-causal and requires the full observation sequence to be available. The training component of many algorithms tends to be computationally expensive, such as the Baum-Welch algorithm for the HMM [68], [29], or the backpropagation method for the artificial neural network (ANN) [69]. For the observation component, DTW [48], [30] and the Viterbi algorithm (see Section VI-B.1 for details) [27], [64], [70] are two common techniques that require a large computational effort. However, it is difficult to assess computational effort accurately if it is not explicitly reported, since it is a factor of algorithm design and implementation methodology.

Non-causal algorithms can only run offline as they require the full observation sequence before algorithmic processing can begin. Examples of non-causal algorithms applied to motion segmentation include the Viterbi algorithm [27], [64], [70], regression modelling techniques [53], or dimensionality reduction tools [71], [72]. Although the Viterbi algorithm is a non-causal technique, some applications run the algorithm against shorter segments of the observation data instead of requiring the full dataset, or run a truncated version [73], [74], allowing the Viterbi algorithm to be operated online.

Considering computational effort and causality, segmentation algorithms can be separated into three broad categories:

Fully online approaches: Algorithms that fall into this category may or may not consist of a training phase. If training and template generation is required, it is computationally fast enough to be performed online. The segmentation component is also performed online.

Semi-online approaches: Algorithms that fall into this category are trained offline, either because the training computational effort is expensive, or full sequences of the training data are required. Once trained, the segmentation algorithm can be performed online.

Offline approaches: Algorithms that fall into this category may or may not consist of a training phase. If training is required, then the training is performed offline. Once trained, the segmentation is also performed offline, due to computational runtime requirements, or causality requirements.

A related concern to computational effort is scalability. For many algorithms, the computation time does not scale well with increasing feature set dimensionality or the number of templates in the motion library. Data streams used for full-body human modelling can consist of 20-30 DOFs [75]. Al-

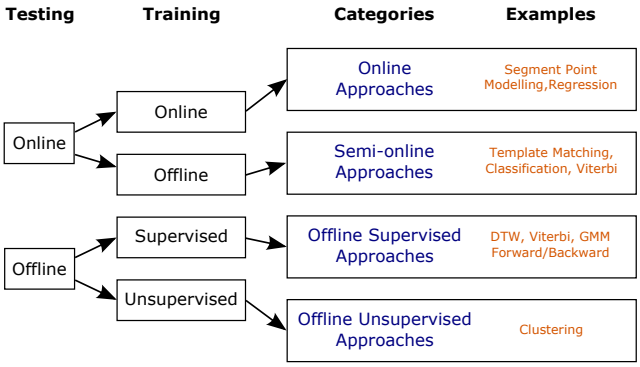

Fig. 3. Overview of the segmentation mechanics in Section VI.

gorithms that iterate or rely on dynamic programming, such as HMM or ANN, can become computationally intractable when the feature space is too large, or if there are too many motion types.

VI. Algorithms

The method for performing the segmentation can be designed after the various constraints in the previous sections have been considered. This section examines different components of the segmentation algorithm, such as any pre-processing or feature space transformations, as well as key algorithm design criteria, such as windowing and algorithm supervision (Figure 3, Table I). The subsequent sections also review the various segmentation algorithms, separating online, semi-online, and offline approaches.

A. Pre-processing

1) Filtering and Outlier Rejection: Pre-processing of the data is often required to remove additive sensor noise from the raw data, due to varying sensor characteristics and limitations in the digitization process [40]. A common procedure is to pre-process the observation data with a properly tuned low-pass [49], [64], [29], [8], [67], or median [64], [67] filter to remove high-frequency noise. High-pass [67] filters have also been employed to reduce drift impact. Low-pass filters with a cutoff frequency (f_c) of 0.1-4.0 Hz [67], [54], [29], [22] have been applied to IMU and joint angle data. Fast moving signals may require low-pass filters up to the 4^{th} order and a f_c of 6.0 Hz to sufficiently remove noise [40]. For high-pass filters, filters with a f_c of 0.1 Hz have also been applied to accelerometer data [67]. Normalization of the data [82], [49] may also be necessary.

2) Feature Space: The feature space to be used by the segmentation algorithm is dependent on the type of data available from the data collection phase. The measurement data can be processed to extract proxy features or statistics, or projected into some latent space, to allow for easier processing when compared to the original observation space.

Different feature space manipulation techniques have been used for these purposes, but generally fall into four categories: (1) no transformation, (2) transformation without dimensionality reduction, (3) transformation with dimensionality reduction, and (4) kernel methods. These methods can be characterized by the resultant DOFs versus the complexity of the mapping algorithm (Figure 4).

No transformation: The segmentation is performed directly on the input space. Techniques that use this approach typically rely on data directly from sensors, such as accelerometer signals [54]. This technique requires no pre-processing but can suffer from scaling issues, as it is difficult to perform segmentation on high dimensional data due to high computational cost and the existence of correlated and uninformative dimensions.

Transformation without dimensionality reduction: A transformation of the original features is used as the new representative feature space. The representation retains approximately the same dimensionality as the original feature space. Common techniques include differentiation [21], or the calculation of joint angle data from motion capture [44] and IMUs [52]. Statistical and spectral features, such as mean, or entropy, can be extracted from the data.

Transformation with dimensionality reduction: The observation data are mapped to a lower dimensional space where segmentation may be easier to perform. This can be achieved using dimensionality reduction tools like principal component analysis (PCA) [49], feature selection [29], coefficients of frequency transforms [83], [54], or distance metrics [39].

Transformation with dimensionality increase: Kernel methods map data to an implicit higher-dimensional feature space [73], [74], [84] via the *kernel trick*, where the higher dimension is obtained by taking the inner product between all pairs of data in the feature space. This is computationally cheaper than explicitly determining the higher dimensional coordinate space. This is common in algorithms that employ SVM [85], to generate a higher dimensional space where the data are better separable. Higher-dimensional embedding [86], [44] is also a way to increase the dimensionality.

3) Windowing: Rather than considering raw measured data, some algorithms instead use summary statistics computed over windows of measured data. The two main variables for windowing are the size of window, and the amount of overlap between adjacent windows. *Fixed window* algorithms use a sliding fixed-length window, while *variable length window* algorithms employ windows that change their length dynamically to fit the incoming data. Fixed window

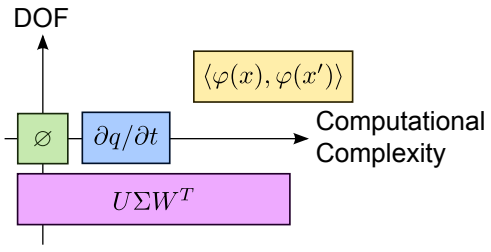

Fig. 4. Feature space mapping can be conceptualized as a two dimensional space of dimensionality and computational complexity. The origin denotes the original number of DOFs with no transformation (green, $\varnothing$). Transformation methods without dimensionality reduction (blue, $\delta q/\delta t$), methods with dimensionality reduction (purple, $U\Sigma W^T$), and kernel methods (yellow, $\langle\varphi(x),\varphi(x')\rangle$) require increasing computational complexity.

TABLE I

COMPARISON OF SEGMENTATION ALGORITHMS WITH VALIDATION RESULTS. IF BOTH PRECISION AND RECALL SCORES ARE REPORTED, THEY ARE COMBINED INTO THE F_1 SCORE IN THIS TABLE. THE SPECIFICATIONS (SPEC) COLUMN DENOTES THE ALGORITHM'S COMPUTATIONAL EFFORT (ONLINE (O), SEMI-ONLINE (M) OR OFFLINE (F)), TEMPLATE (SUPERVISED (S), UNSUPERVISED (U) OR ADAPTIVE (A)), AND SEGMENT DEFINITION (PHYSICAL (P), DERIVED (D) OR TEMPLATE (T)). FOR THE DATA USED, JOINT ANGLE (q), CARTESIAN (x), GYROSCOPE (w), ACCELEROMETER (a), AND FORCE (f) DATA ARE USED. THE NUMBER OF SUBJECTS (n_s) AND PRIMITIVES (n_p) USED FOR VERIFICATION ARE ALSO DENOTED. INFORMATION NOT REPORTED IN THE PAPER IS DENOTED BY DASHES.

	Spec	**Segmentation Algorithm**	n_s	n_p	**Data**	**GT**	**Verification**	**Accuracy**	**Percentage**
[44]	O/A/D	Segment on high PDF distances between windows.	1	45	q; exercise [44].	VP	Temporal, $t_{err} = 0.12s$.	F_1	75-76%
[76]	O/A/D	Threshold on neighbouring topical similarity.	6	51	x; exercise [62], [63].	OP	Temporal, gradient.	F_1	87-90%
[77]	O/U/D	Piece-wise regression.	2	4	q; eating.	-	Temporal.	F_1	48-92%
[25]	O/U/D	Segment on distance threshold between windows.	-	7	q; ADL.	VP	Temporal inv.	F_1	93%
[49]	O/U/D	GLR.	5	3	x; hand [61].	-	Labels.	$1 - \frac{FP}{TP}$	91-96%
[78]	O/U/D	Region grow from moving window distance matrix.	6	51	x; exercise [62], [63].	OP	Labels.	Precision.	88-97%
[22]	O/U/P	Gyroscope turning points.	20	5	w; gait.	a spike.	Labels.	$\frac{TP}{total}$	80-99%
[31]	M/S/T	Piecewise regression slopes discretization. Sequences labelled by bag-of-word.	6	1	a; exercise.	-	-	F_1	60-95%
[73]	M/S/T	SVM hyperplane used in HMM emissions. Segment by online Viterbi.	-	4	f; teleop.	-	Labels.	Class.	84%
[68]	M/S/T	Online Viterbi.	4	3	q; tracing.	AP	Labels.	Class.	94%
[29]	M/S/T	Velocity profiles matched to observation, then verified on HMM.	20	5	q; exercise.	VP	Temporal, $t_{err} = 0.12s$.	F_1	85-90%
[32]	M/S/T	Used classifiers to recognize segment points vs. non-segment points.	20	5	q; exercise.	VP	Temporal, $t_{err} = 0.12s$.	F_1	80-92%
[11]	M/S/T	DMP in Kalman filter parameter estimation.	1	22	x; writing.	-	Labels.	Class.	90-100%
[79]	M/S/T	Threshold on similarity between observation and time aligned low-DOF templates.	30	12	x; exercise [80].	OP	Labels.	F_1	55-73%
[67]	F/S/T	Statistical features used in classifiers, then HMM. Segment by Viterbi.	48	9	a; ADL.	-	Labels.	Class.	76-99%
[53]	F/S/T	Viterbi fitted data to regression model library.	6	12	a; ADL.	OP	Labels.	Class.	90%
[43]	F/S/T	AdaBoost HMM.	-	22	x; various.	-	Labels.	Class.	84-100%
[48]	F/S/T	DTW with dynamic band.	2	2	x; hand.	-	Labels.	Class.	100%
[27]	F/S/T	Viterbi algorithm.	8	20	q; surgical.	-	Labels.	Class.	70-87%
[64]	F/S/T	SVM Labels. used as HMM feature vectors. Segment by Viterbi.	10	12	x; reaching.	-	Labels.	Class.	48-95%
[81]	F/U/D	Minimize segment-cluster distances.	36	25	q, x; ADL [62]. Video.	OP	Labels.	Class.	77-83%

algorithm performance tends to be sensitive to window length and the amount of overlap between subsequent windows [24], [86]. Overlap between the current window and the subsequent window ranges from 0-50% [77], [29], [67], [32]. A special case of the fixed window approach is a window with the length of one data point [87], [21]. Variable length window algorithms tend to be more computationally expensive as additional computation is required to determine window size. However, the window size is more targeted to the underlying movement which potentially improves the segmentation quality. Typically, variable length windows do not overlap with each other [29], [31].

4) A Priori Knowledge Requirements: The need for labelled data, and the amount of effort required to generate such labels, separates segmentation algorithms into three distinct groupings:

Unsupervised: No labelled data are provided to the algorithm and no pre-trained models are generated *a priori* [44], [8]. Unsupervised approaches tend to identify the segment edge points directly by relying on domain knowledge [21] or changes in features [24]. They tend to be computationally faster than supervised algorithms.

Supervised: Labelled data are provided *a priori* to the algorithm, leading to supervised model training [29], [53]. The key characteristics that describe a segment are determined automatically by the algorithm based on the training data.

Adaptive: Adaptive solutions update the model online as new data are observed, which allows existing models to be contextualized to the observation data, and potentially increasing segmentation accuracy. Two possible methods to achieve adaptive modelling are: (1) template modification, where an existing primitive model is modified by new observation data [44], and (2) template insertion, where new templates are created from the observations and added to the primitive library [44].

B. Online Segmentation Approaches

Online algorithms refer to methods that train and segment in an online fashion. These approaches tend to be computationally light, and do not require the full observation sequence to be available before performing segmentation.

1) Segment Point Modelling: Algorithms are designed to identify segment points, as opposed to recognizing the motion being performed. They tend to be simple, but some prior knowledge of the nature of the motion is required. These algorithms do not require template training, and thus are considered unsupervised.

Thresholding on Feature Vectors: Many algorithms declare a segment point when observed features cross a threshold. This method of segment point declaration is simple when only small feature sets are examined. It becomes more difficult with larger feature sets as it becomes more difficult to determine which feature set to threshold on. A common method of segmentation by thresholding is segmentation by zero crossings [87], [21], [88]. Local maxima and minima can also be thought of as an instance of zero-crossings, since local maxima in one dimension are crossings in the derivative of that dimension.

Many motion-based segmentation algorithms rely on velocity-based features. These methods assume that a change in body link direction denotes a natural segment point. These direction changes are accompanied by local turning points in the joint angle space, resulting in zero-velocity crossings (ZVC). Pomplun and Matarić [87] apply the ZVC concept to motion segmentation to study the effects of rehearsal in motion imitation in humans. The algorithm assumes that motions have clear velocity-crossings to denote the start/end points. To reduce over-segmentation, minimum segment time lengths are enforced. Fod *et al.* [21] segment when the sum of all velocities is lower than some threshold. Lieberman and Breazeal [88] replace the static threshold for ZVCs with a dynamic one, which allows the algorithm to adjust to mean velocity changes and to movements that do not cross zero in joint space.

The velocity crossings concept has been applied in other motion-based applications [28], [89]. Other feature crossings examined are joint acceleration [89], [90], linear acceleration [54], and angular jerk [91]. For movement data, these crossings represent a pause in the movement, and thus serve as a logical segment point.

Despite their popularity, thresholds and zero-crossing approaches suffer from various shortfalls. They have a tendency to over-segment, particularly with noisy data or with an increasing number of DOFs [21]. It is difficult to determine which crossing points are actual segment points, or spurious crossings. The threshold value requires tuning, which becomes more difficult as the number of DOFs increases. Additional algorithms such as HMMs [28], [49] or stochastic context-free grammar (SCFG) [91], [89] must be used to provide movement labels following segmentation. Lastly, using thresholds assumes that primitives have well-defined thresholds and crossings in the input space. This is not true in all cases. For example, a circle being traced repeatedly would not exhibit crossings [21].

Thresholding on Distance Metrics: Derived metrics can be used to denote the degree of separation or difference between two sets of data. When the derived metric reports a value above some threshold between two sequential windows of data, or between a window of data and standard templates, a segment is declared. Common metrics are the Euclidean distance [39], [92], Mahalanobis distance [25], calculated by normalizing the Euclidean distance by the signal variance, and the Kullback-Leibler (KL) divergence [86], [44], a measure of the difference between two probability distributions. Other distance metrics can be employed, including the generalized likelihood ratio (GLR) [49], and signal variance [24], [86].

Vögele *et al.* [78] segment by creating a self-similarity matrix, where the distance between the current frame and all future frames is calculated. The main diagonal of this matrix is removed, since a given frame will always be the most similar to itself. Segments are declared by finding isolated regions, suggesting that the movement in one region is very

different from the next. Similar activities can be clustered for identification.

Bashir *et al.* [49] apply the GLR [93] as a distance metric for curvature data segmentation. The GLR is a statistical test used for hypothesis testing. The null hypothesis is that two different segments were generated from the same model. If the ratio is above some threshold, then the null hypothesis is rejected, and thus a segment should be declared. This method has been applied to ADL accelerometer data [83].

Using signal variance as a distance metric is also a common technique. When a feature set suddenly exhibits a large change and the variance becomes very high, it indicates that the underlying motion may have changed to another motion; thus a segment point should be declared. Using variance for movement data segmentation was first proposed by Koenig and Matarić [24].

Instead of calculating the variance directly, different representations of the signal variance can be used. Barbič *et al.* [25] apply PCA to a window of the observation data and retain the top *r* PCs. They apply this truncated PCA transformation matrix to the subsequent frames of the observation data. The reconstruction error between the pre-transformed data and the post-transformed data is calculated. If the underlying motion changed at time t, the PCA-projected data will differ from the pre-transformed data, causing a spike in the reconstruction error at time t when compared to the error at previous timesteps, and a segment is declared. However, this method is sensitive to the *r* and error threshold used.

Another alternative to calculating the variance directly is to segment based on changes exhibited by a signal's probability density function (PDF), typically via the HMM and the Viterbi algorithm. The HMM [94] is a modelling technique where the movement data is represented by a stochastic dynamic model. The motion is represented by an evolving unobservable state that obeys the Markov property. Given an HMM, the Viterbi algorithm finds the best state sequence for a given observation sequence. It does so by iteratively calculating the highest probability $\delta_{t+1}(j)$ at time t, which accounts for the first t observations, ending in the j^{th} state.

Kohlmorgen and Lemm [86] use the HMM and an online version of the Viterbi algorithm [94] to perform segmentation. A sliding window is used to calculate the PDF of the windowed data. The PDFs are used to train a HMM. The online Viterbi algorithm is applied to determine the state transitions of the HMM, thus producing segment bounds for a given observation set. This algorithm has been applied to human joint angle data [95]. Kulić *et al.* [44] extended Kohlmorgen and Lemm's algorithm [86] by clustering previously segmented sequences to generate new templates in real-time. Once a new primitive has been identified, the primitive is modelled as a HMM. The novelty of the primitive is determined by its KL distance from existing HMMs. Novel primitives are added into the movement library, whereas non-novel observations are clustered into existing HMMs. A hierarchical HMM can also be employed, where a higher-level HMM contains each primitive as a state to determine the transition between primitive types [45]. This method has also been applied to IMU data [96].

Distance metrics allow segmentation to be performed on a wider range of feature vectors and do not require *a priori* knowledge. Using variance as a distance metric allows these particular algorithms to scale to higher DOFs. Distance-based segmentation shares many of the same weaknesses as the direct feature thresholding approaches. They do not have mechanisms to reject false segments and do not provide segment labels. Tuning is required to determine threshold values [97], [98], [99], [25], [24], [100], [8], or the algorithm is sensitive to the width of the sliding window, or cost function.

2) Regression: The algorithms examined in Section VI-B.1 model the segment point explicitly. Online regression approaches consider a different conceptual model by modelling the primitive itself.

Keogh *et al.* [26] assume that the observation data can be described by piecewise linear fitting. A large sliding window, sufficient to fit 5-6 segments, is used to window the observation data. This sliding window is divided into small sub-windows and separate linear regression models are calculated from the start of the sliding window to the end of each of the sub-windows. The error for each sub-window is defined as the error between the regression line and the underlying data with a cost function that penalizes long sub-windows, to keep each segment small. The window edge that results in a model with the smallest error that exceeds a defined error threshold is declared a segment. The sliding window is advanced to the end of this segment and this process is repeated. Keogh's algorithm [26] has been applied to segment for gesture recognition, with an HMM for motion identification [77].

Lu and Ferrier [82] use an auto-regressive (AR) model to represent the data over a two-timestep sliding window. When the model from the previous window to the current one is sufficiently different, which is determined by the Frobenius norm, a segment is declared. To reduce over-segmentation, segment points that are close together are removed.

The regression algorithms described above are suitable for segmenting both repetitions of the same primitive, as well as separating different motion primitives. However, these algorithms are very sensitive to parameter tuning, such as the cost function or window size [26]. The resultant regression functions can overfit and do not generate high quality segments. Similar to variance-based approaches, these algorithms do not include methods to reject trivial motions such as tremors and other noise.

C. Semi-online Segmentation Approaches

Semi-online approaches encompass methods that require an offline training component but can segment online following the training phase.

1) Template Matching: Performing segmentation on a single feature's zero-crossings or threshold levels as proposed in Fod *et al.* [21] is often too simplistic and leads to over-segmentation. A sequence of such features can be used as patterns to identify in the observation data. Requiring a

sequence of feature thresholds to be matched in a specific pattern reduces over-segmentation in comparison to single threshold approaches.

Kang and Ikeuchi [101] use curve fitting to the volume swept out by a polygon formed by fingertip Cartesian coordinates and hand velocity thresholds to segment grasping tasks. Lin and Kulić [29] propose using sequences of velocity and acceleration crossing points to coarsely locate segment points. HMMs are employed in a fine-tuning step to further reduce over-segmentation. Feature sequence templates and the HMM templates are trained *a priori* from labelled data. Zhang *et al.* [22] use sequences of velocity features to denote heel strikes during gait.

Ormoneit *et al.* [102] examine the signal-to-noise (SNR) ratio of the observation data where the signal is denoted as data amplitude and the noise is the variance. The SNR is used to determine the optimal window for segment searching, and this window length is used to perform curve fitting to premade templates.

Feature matching may be robust to small movement variability but may not generalize sufficiently to large interpersonal variability. For example, when healthy templates are used to segment patient templates, the performance is noticeably worse, suggesting generalizability issues [29].

2) Segmentation by Classification: In many of the algorithms examined so far, a segment bound is declared when the signal passes a threshold, either considering the signal directly, or some distance metric computed from the signal. A separate labelling algorithm is required to classify the motion segments found between each pair of identified segment points. An alternative approach is to label the observation data based on the patterns in the observation vectors, thus transforming the time-series segmentation problem to a multi-class classification problem. This method can be performed by employing sliding windows and pre-trained classifiers [103] and is commonly used in activity recognition contexts [67], [7], [104], [105], [106] where each data point is labelled as a primitive type by the classifier. It has also been applied for primitive segmentation [32], [31], [79].

Instead of assigning each data point a primitive label, Lin and Kulić [32] use classifiers to label each data point as either a segment point or a non-segment point, in order to automatically learn segment point features of multiple primitives without explicit domain knowledge. Joint angle and velocity data are used in order to incorporate temporal information into the classifiers.

Berlin and van Laerhoven [31] monitor psychiatric patients using accelerometers on a wrist watch and apply piecewise linear approximation [26]. The slope of the linear segments is converted to angles and binned. The degree of discretization was determined *a priori*, via tuning. Symbols are assigned to sequential pairs of bins, creating motifs. The motifs are generated by inserting training data into a suffix tree, and common chains are used as the motif templates. Segmentation and identification are performed simultaneously by using a bag-of-words classifier.

Zhao *et al.* [79] cluster n-DOF manually segmented data together to create a template dictionary. Each dictionary entry only contains one DOF from the data, resulting in n times more models but less overall computational cost. A time-warping distance feature set is calculated between the training data and each entry in the template dictionary that corresponds to the correct DOF. This feature set is used to train a linear classifier. Observed data are segmented by determining the optimal window via DTW, converted into the distance features, and labelled via the linear classifier.

Classifier approaches reduce the need for domain expertise but are poor at handling temporal variability. Velocity information [32] can be incorporated to account for this issue.

3) Online Supervised Viterbi: The algorithms described here modify the traditional offline Viterbi algorithm so that it can be operated online, assuming that the model has been trained *a priori*. This approach has been applied to segment human joint angle data as the human guides a robotic arm through a pre-determined trajectory [68], [107].

Castellani *et al.* [73] use pre-trained SVMs to classify sub-tasks during robotic tele-operation tasks. A one-vs-all SVM is used to classify each subtask, which form the HMM states. The SVM hyperplane is translated into a sigmoid function and used as the HMM state emission probability. An online Viterbi algorithm is used to segment the whole data sequence. A 'peg in hole' telerobotic task was used to verify the segmentation accuracy, which consists of several smaller sub-tasks. The state transitions, denoting the change from one sub-task to another, are defined as the segment points. Ekvall *et al.* [74] apply this method to other telerobotic tasks.

Hong *et al.* [47] use the finite state machine (FSM), a deterministic version of the HMM, for video data segmentation. The training data are represented by spatial Gaussians. The number of Gaussians is calculated by dynamic *k*-means without the temporal data and is done offline. Once the spatial information is segmented, the temporal data are included in the training of the FSM. One FSM is trained for each gesture. When a new observation vector arrives, each FSM decides if the state should be advanced, based on spatial and temporal distances between the observation and the FSM state model. When an FSM reaches the final state, a segment is declared. The approach is verified using a 'Simon Says' system, where the program requests the user to perform a given a task, and the program verifies that the task is performed correctly.

The online Viterbi approach enables online application, but can provide different results than the standard Viterbi algorithm. Ignoring the back propagation component means that the online Viterbi algorithm does not have the full data set to calculate its likelihood values, only the data up to the current time step, and can result in the algorithm suggesting an incorrect segment, when the standard Viterbi performs optimally, as demonstrated in [73].

4) Other Stochastic Methods: The Kalman filter has been applied for segmentation purposes. Meier *et al.* [11] constructed multiple dynamical movement primitive (DMP) templates for segmentation. The algorithm assumes that the start of the observation is the start of the first segment, so

the segmentation task is to find the end of the movement primitive. It does so using the pre-trained DMPs and uses the Kalman filter to estimate the segment length (τ_i) and posture of the observed segment. The observed segment is identified by an expectation-maximization (EM) procedure that is used in the Kalman filter. Once the elapsed time (t_{curr}) exceeds τ_i, the end of the segment is assumed to be found. The segment is declared, and the algorithm restarts again at $t = t_{curr}$.

D. Offline Supervised Segmentation Approaches

Offline methods refer to techniques that perform both training and testing offline. These algorithms, such as the Viterbi algorithm, require the full observation sequence to be available before segmentation. Other algorithms, such as boosted HMMs [43], are too computationally expensive to run online.

1) Dynamic Time Warping: A major challenge of segmentation is the temporal and spatial variations between the template and the observation. DTW [108] overcomes the temporal variations between motion data sequences by selectively warping the time vector to minimize the spatial error between the observation data and the movement template. DTW has been applied to segment Cartesian gesture data [48], full-body exercise data [109], as well as EEG data [38].

Ilg *et al.* [30] employ DTW in a multi-tier fashion. The observation signal is downsampled by removing all data points that are not at a ZVC. The downsampled points are used to warp to downsampled versions of templates. The sum of spatial error is minimized between the observation and the template to ensure alignment, allowing for segmentation and identification to be performed simultaneously.

DTW-based algorithms are computationally expensive at higher dimensionality, preventing them from being utilized online. Poor warping can also lead to singularity issues, where large portions of the motion are warped to small portions of the template. The severity of the singularity issue can be mitigated by constraining the warping path [108], or using the derivative of the data instead of the Euclidean distance [38].

2) Viterbi Algorithm: The Viterbi algorithm overcomes temporal and spatial variations by using an HMM to model each motion template, thus explicitly modelling these variances by the HMM observation variances, and the state transition matrix. The Viterbi algorithm has been used to segment movement data in different applications, such as joint angle data for tele-operative surgeries [27], hand gesture Cartesian data [110], and joint angle and tactile data for hand grips [111].

Ganapathiraju *et al.* [112] and Vicente *et al.* [64] both use SVM and HMM hybrids in a similar fashion. The training data are used to train SVMs, and the SVM is used to label windows of data. The SVM label sequences are used as the feature vectors for the HMM, and the primitive sequences are represented by the HMM state evolution. The Viterbi algorithm is then employed to determine the segment points.

In computational linguistics, a commonly used technique is the *n*-gram model, a *n-1* ordered Markov model [113]. Ivanov and Bobick [114] combine the *n*-gram model with the stochastic context free grammar (SCFG), starting from the current position of the observation data and hypothesizing the possible continuations of the input by tracing down branches on the SCFG tree. The observation input is then compared against expected states and a likelihood of state advancement is generated. When a grammar branch is exhausted, a primitive may have been completed, and the Viterbi algorithm denotes the state path taken and thus the segmentation result.

Baby and Krüger [70] apply HMM merging to improve Viterbi performance. Given an existing HMM template λ_M, new observation data are formulated into a new HMM $\lambda_{\mathbf{c}}$, and merged into $\lambda_{\mathbf{M}}$ by merging similar states between $\lambda_{\mathbf{c}}$ and $\lambda_{\mathbf{M}}$. If there are states in $\lambda_{\mathbf{c}}$ that do not have a corresponding state in $\lambda_{\mathbf{M}}$, these states are inserted as new states into $\lambda_{\mathbf{M}}$. Once all the observation data are merged into the $\lambda_{\mathbf{M}}$, the Viterbi algorithm is used to trace through the motions, and common paths are removed from the Viterbi paths via the longest common substrings method until no common paths exist between any components. Each of these components becomes a primitive; segments are detected on switches between components. This approach was applied to human interactions with objects from the object's point of view [115].

Chamroukhi *et al.* [53] segment movement data employing multiple regression models and segmenting on model switch. The observation data are represented by a regression model, $y_i = \beta_{z_i} t_i + \epsilon_i$, where the regression coefficient β_{z_i} is a function of the logistic hidden state z_i. z_i controls the switching from one activity to another, for k different activities. That is, the regression model describes a different motion according to the state of z_i, while the logistic model captures the higher level stochastic dynamics of the transitions between motions. When the state of z_i changes, a segment point is declared. The parameters of the regression models and $\mathbf{z}$ are trained by the EM algorithm. The segments are produced by estimating z_i at each y_i in a similar fashion to the Viterbi algorithm.

Although the usage of the Viterbi algorithm is widespread, it suffers from several key issues. It is expensive to use, and requires the full observation sequence to be available. The Baum-Welch and the Viterbi algorithm are also local optimizations, so the solutions provided may not be globally optimal. For the HMM, the modelled data are assumed to be Gaussian, which does not hold for human movement in general [9]. Tuning is required to find the suitable number of states to represent the model and to prevent overfitting.

3) Gaussian Mixture Models: Gaussian mixture models (GMMs) are parametric PDFs represented as the weighted sum of Gaussians. For segmentation, the boundaries between each Gaussian are used to denote the segments. The number of Gaussians needed for the GMM is typically determined *a priori* [25]. GMMs have been used to segment exercise data [25], and for imitation learning [116].

Like the HMM, GMM modelling assumes that the modelled data are Gaussian, or near Gaussian in nature. The number of Gaussians needed to model the data requires some degree of tuning; the Bayesian Information Criterion (BIC)

can be used to assist the tuning effort [116].

4) Forward/Backward Algorithm: This algorithm is a technique for determining the likelihood that a given sequence of observation data is generated from a given HMM [94] and is typically used for primitive identification. However, it has also been applied to primitive segmentation.

Wilson and Bobick [117] utilize pre-trained parametric HMMs (PHMMs) of hand pointing gestures. A fixed-length sliding window is used on the observation data, and the EM algorithm is used to estimate the parameters of the PHMM over the windowed data. The corresponding likelihood value is determined by the forward/backward algorithm. Windows that result in a high likelihood are declared segments.

Lv and Nevatia [43] use HMM templates as classifiers in an AdaBoost algorithm. The observation data are split into two windows, with the first window starting at some minimum length l_{min}, and increasing at each iteration. The two windows are run through the AdaBoost classifiers, and the window length that results in the highest likelihood is selected, forming a segment at l_{maxLL}. The algorithm is run multiple times, with the starting point of the first window advancing to the end of the previous segment at each run.

Both of these methods incur a large computational cost for both training and segmenting and cannot be used online.

E. Offline Unsupervised Segmentation Approaches

Another approach is to assume that the observed data evolves according to an underlying deterministic model that has been contaminated with time warping and additive noise. Probabilistic methods can be used to approximate both the parameters of the underlying model and find the segmentation locations. Chiappa and Peters [118] estimate the underlying model, warping terms and noise model via EM. This approach requires the full sequence for action fitting, making it unsuitable for online applications.

Zhou *et al.* [81] use an extension of DTW to produce similarity measures between two temporally aligned segments. Given an initial set of segments, this measure is used in a kernel *k*-means framework to determine segment cluster centres and to assign each segment to a cluster. For each segment, a search is used to determine the segment boundaries that would result in minimal distance between the segment and its cluster centre. These two steps are repeated to iteratively converge to a solution.

Lan and Sun [76] model motion data as a written document with unknown topics (the motion), composed from a vocabulary of words (key poses). Hierarchical clustering is used to extract key poses, then all data frames are discretized to these key poses. Latent Dirichlet allocation, a topic discovery generative modelling technique, is used to group the key poses into motion primitives. A sliding window is used to calculate between-window topical similarity, and a segment is declared using a threshold. Newly observed key poses and primitives can be incorporated to update the model.

Fox *et al.* [119] examine multiple sets of time series data simultaneously to extract global movement characteristics over all movements. Individual time series are assumed to exhibit only a subset of these characteristics, over certain lengths of time. These characteristics and behaviours are modelled as autoregressive HMMs (AR-HMMs), trained by a Markov Chain Monte Carlo process, and can be thought of as features of the movement. Features that describe a given time-series data are selected via a beta process model. Segments are declared when the time series shift from one AR-HMM to another, signifying a shift in the underlying movement.

Although most distance and threshold-based algorithms are computationally light enough to be computed online, some approaches employing computationally expensive derived features can require offline implementation. Isomap, a dimensionality reduction technique, has been combined with thresholding on Cartesian maxima [71], [120] and joint angle crossing points [72] to segment.

VII. Verification

Verification techniques are used to determine how well an algorithm performs on a given dataset compared to ground truth. The selection of the verification technique can highlight or obscure the strengths and weaknesses of an algorithm. In the following, it is assumed that ground truth data consists of *manual segment edge points* provided by the expert rater (Section IV-B), while the algorithm being evaluated generates *algorithmic segment edge points*.

An algorithmic segment edge point is labelled as a true positive (TP) if it corresponds to a manual segment edge point or false positive (FP) otherwise, whereas the absence of an algorithmic segment edge point can be labelled as a true negative (TN) or false negative (FN), if a manual segment edge point is present or absent, respectively, at the corresponding algorithmic segment edge point. This general scheme conforms to common statistical measures of performance, but in some of the assessment schemes, for example, when comparing point pairs, the TN set would result in an empty set. Alternative assessment metrics, such as ones based on shape similarity between templates and observations, may exist, but have not been used for segmentation. In the following, Acc is used to denote the different scores used to represent accuracy. Ver is used to denote the verification methods used to calculate the algorithm accuracy.

Two common accuracy metrics, precision ($Acc_{precision}$) and recall (Acc_{recall}) can be computed from the comparison of the algorithmic and manual segmentation labels. Of the labels declared by the algorithm, *precision* reports the percentage of points that received the correct label. *Recall* computes the ratio of correct labels to the total number of labels. These two scores can be aggregated together into the F_1 (Acc_{F_1}) score. These metrics are formulated as follows:

$$Acc_{precision} = \frac{TP}{TP + FP} \tag{1}$$

$$Acc_{recall} = \frac{TP}{TP + FN} \tag{2}$$

$$Acc_{F_1} = \frac{2 \cdot TP}{2 \cdot TP + FN + FP} \tag{3}$$

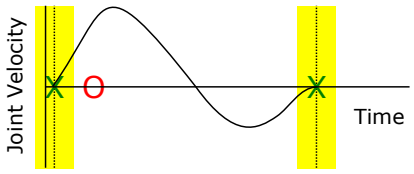

Fig. 5. An example of the temporal tolerance approach, where the dotted line denotes a manual segment edge point and the yellow region is error tolerance ($\pm t_{err}$). The green (X) algorithmic segment edge points are declared correct and the red (O) is declared incorrect. This algorithm is declared to have an $Acc_{precision}$ of 50%.

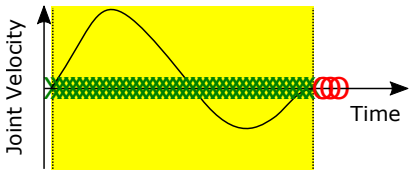

Fig. 6. An example of the classification by data point labels approach, where the dotted line denotes a manual segment edge point and the yellow region is the correct region for the primitive under examination. Of the 43 labels, 40 are declared correct (green X) and 3 are declared incorrect (red O). This algorithm is declared to have an $Acc_{precision}$ of 93%.

In addition to these metrics, the allocation of data between training and testing is important. The process of training on one set of data and testing on another set of data is called *k*-fold cross-validation, where the dataset is divided into *k* folds, and one fold is used to validate, while *k-1* folds are used to train. In cases where the dataset cannot be easily divided, or there are insufficient samples, leave-one-out cross-validation (LOOCV) is employed, where only one sample is used for validation, and the rest are used for training. This technique can be applied to validate the inter-subject variability and the inter-primitive variability by placing all of the test subject's data, or all of the test motion data into the testing fold.

The accuracy of an algorithm can be measured by temporal tolerance or by classification by data point labels.

A. Temporal Tolerance

One approach to assess algorithmic segment edge points is to declare TPs when they are close to existing manual segments. An algorithmic segmentation point is declared correct if it falls within $\pm t_{err}$ of a manual segment edge point. A FP error is declared if an algorithmic segment edge point was identified when there is not a corresponding manual segmentation point within the $\pm t_{err}$ region. A FN error is declared if a segment edge point was not found algorithmically for a manually identified segment edge point within the $\pm t_{err}$ region. Alternatively, a distance-based metric can be utilized, where a score of 1 is assigned if a manual segment coincides perfectly with an algorithmic segment and decreases towards 0 as the algorithmic segment approaches $\pm t_{err}$ [76]. An inverted version can also be defined where the algorithmic segment edge point is deemed correct if a manual segment edge point falls within $\pm t_{err}$ of it [25].

This verification method is typically used by segmentation algorithms that search for the entire primitive at once or otherwise determine the segment boundaries in a direct fashion and want to assess the accuracy of the segment boundaries [44], [29]. See Figure 5 for an illustration.

This method is sensitive to the selection of t_{err}. t_{err} ranges widely and should be considered in relation to the length of expected primitives. Authors have used t_{err} from 0.2-1.0 s [44], [29], [98]. This method will be denoted as $Ver_{temporal}$.

B. Classification by Data Point Labels

Instead of only declaring algorithmic segment edge points when they occur, all data points can be assigned a label and compared against ground truth.

This method is less stringent than the temporal tolerance approach. Incorrect segment edge points do not heavily influence the results since there are many more within-segment data points available to smooth out poor segment boundaries. See Figure 6 for illustration. Note that, even though this result looks similar to Figure 5 in that the resultant segment bounds are in the same place, the accuracy is higher using this verification method, because of how the accuracy is being reported. Although FPs are penalized more lightly, FNs are penalized more heavily. If a segment is completely missed, 3 FNs would be counted in this example, as compared to only 1 FN for the temporal tolerance scheme.

This verification method has been used by segmentation algorithms that label each time step as a given label and segment when the label changes from one primitive to another [121]. Usually, this is done by assessing windowed data rather than individual points. This method treats time-series segmentation as a classification problem as each data window is assigned a label. Acc_{Class} is defined as the number of data windows with the correct label when compared to the ground truth. In these cases, segmentation and label accuracy are the same. This method will be denoted as Ver_{labels}.

C. Validated Algorithms

A majority of the algorithms examined do not perform any form of verification, or they do not report their verification methods or scores, or report only the identification accuracy instead of the segmentation accuracy, making it difficult to compare between methods. Methods that reported some form of validation results are summarized in Table I.

For online approaches, the highest temporal accuracy was reported by Barbič *et al.* [25], with an Acc_{F_1} of 93%. This paper segments by calculating the Mahalanobis distance between a window of data against subsequent windows, and thresholds on distance peaks. The distance metrics scale well to higher dimensions as the segmentation focuses on a single feature and is lightweight to use. This method successfully separated action sequences consisting of 7 different primitives, but is sensitive to the tuning parameters, such as the window length and the threshold. The primitives examined consist of highly different full-body movements, such as

jumping, walking, kicking, or punching, which proved to be dissimilar enough for the distance metric, but may be difficult to generalize to primitives that are very similar to each other, or primitives that are not very correlated [25].

For semi-online approaches, the highest temporal accuracy was reported by Lin and Kulić [29], with an Acc_{F_1} of 85%. They applied a two-tier algorithm where pre-trained velocity peak and crossing templates provide a set of segment candidates from the observation data. The candidates are passed into HMMs where the forward algorithm is used to verify the segments. They tested with both individualized velocity/HMM templates, as well as generalized templates. HMM-based methods scale well to higher dimensions but require long training times, which need to be completed *a priori*. However, velocity features rely on relatively simple primitive types, and may not scale well to complex or unsteady movements [29].

For offline approaches, none of the examined algorithms reported temporal segment accuracy. The highest reported label identification accuracy was reported by Chamroukhi *et al.* [53], with an Acc_{class} of 90%. They modelled the observation data as a set of regression models and switched between the models via a Viterbi-like algorithm. This method was applied to various postures and ADL. The Viterbi algorithm shows high accuracy, scales well to higher dimensions, but requires both training and testing to be carried out offline, which needs to be completed *a priori*.

VIII. Outlook

A. Outstanding Problems

While many approaches have been proposed for motion primitive segmentation, active areas of research remain.

1) Inter-subject Generalizability: Inter-subject generalizability remains an outstanding problem. Although techniques such as HMM can be applied to generate multiple-subject templates and account for spatial and temporal variabilities, only a few algorithms generate such templates and test against multiple subjects [29], [27]. Multiple-subject templates often lead to a drop in segmentation accuracy [32]. Applications such as physical rehabilitation often do not have access to patient movement data *a priori*, and thus must rely on template generalization.

If the segmentation algorithm is to be applied to subjects of different demographics or capabilities, large variations can be expected, and pose a significant challenge for algorithm generalizability. This problem is especially significant in cases where few training samples are available.

2) Inter-primitive Generalizability: Inter-primitive generalizability also remains an outstanding problem. A few algorithms, such as those reliant on domain-knowledge [21], classifiers [32], or parametrized models [117] provide potential solutions, but do not tend to explicitly explore or report inter-primitive generalizability.

Generalizability is an important issue in rehabilitation since the exercises are typically modified slightly in order to suit the patient's capabilities, and an algorithm that can provide some degree of generalizability would increase the utility of any given template. In exercise applications, movements that vary in only direction should be considered the same movement, but may pose a challenge for the algorithm [117], and thus require techniques that are robust to these types of variability.

Generalizability is also an important concern for applications where the motions to be performed are not known *a priori*, such as in online human machine interaction. The existing segmentation work can be divided into techniques that model the primitive explicitly, or techniques that model the segment point directly. Techniques that model the segment point directly using domain-knowledge [21] or learned automatically via classifiers [32] provide the means to segment based on common characteristics over all the primitives of interest, and warrant further investigation.

3) Adaptive Techniques: Some of the generalizability issues above may be alleviated with adaptive techniques, where existing models are retargetted or augmented with new observations. Adaptive techniques are useful in situations where the primitive of interest can vary widely or training from scratch is computationally prohibitive. Segmentation techniques that examine model modification tend to be computationally expensive [70], but online algorithms [44] have been developed.

4) Algorithm Verification and Public Databases: The majority of the algorithms examined do not explicitly report segmentation accuracy, in part due to the difficulties of providing labelled data. Algorithm testing against comprehensive publicly available datasets with labelled data is recommended, as they would provide both a common ground to compare different algorithms, as well as reduce the amount of post-processing work that researchers must do to carry out algorithm verification.

Currently available databases (Section IV-C) tend to focus on healthy populations, omitting populations which may have significantly different movements. These alternative populations would provide a wider spectrum of data for inter-subject testing, and are particularly important for rehabilitation applications.

IX. Conclusion

Considering applications such as human movement analysis for rehabilitation or imitation learning for robotics, algorithms that detect the start and end points of movement primitives with high temporal accuracy are required. Movement primitive segmentation enables learning detailed motion models for analysis of motion performance and robotic motion imitation.

The proposed framework provides a structure and a systematic approach for designing and comparing different segmentation and identification algorithms. This framework outlines key points of consideration for the segment definition, data collection, application specific requirements, segmentation and identification mechanisms, and verification schemes. The framework can guide any designer though the various components of solving the segmentation problem. The framework has also been applied to a review of the

literature. The analyzed algorithms can be separated into online, semi-online and offline approaches, as computational cost constraints and the availability of exemplar data often serves as the major limiting factor in any given application, and allows the algorithm designer to narrow down the possible techniques quickly.

Online approaches refer to techniques where template training and observation segmentation are performed online. Typically, template training is not required and the segment point is modelled explicitly. These techniques include segmenting when specific features [87], [21] or distance metrics [25], [86] exceed some threshold, or at the junction points of piecewise linear regression fits of the data [26]. These techniques are computationally light, and do not have large numbers of tunable variables to increase algorithm complexity. However, without a model, they are sensitive to false positives and tend to oversegment.

Semi-online approaches refer to techniques where the template training occurs offline, but the segmentation is performed online. These techniques include using sequences of velocity features to segment and identify the observation motion [29], and converting piecewise linear regression lines to motion motifs and using bag-of-words classifier [31], or applying classifiers to learn the segment characteristics of motions [32]. Lastly, online variants of the Viterbi algorithm have also been used [68], [73]. These techniques are computationally light, but require templates to be constructed *a priori*, which can help in reducing oversegmentation.

Offline approaches are techniques where both template training and observation segmentation are performed offline, due to the segmentation process being non-causal, or the segmentation process is too computationally expensive to be performed online. Algorithms that fit into this category include the DTW [48], [30], the Viterbi algorithm [27], [70], GMM methods [25], and Isomap [120]. These techniques are very computationally expensive and cannot be operated online, but may yield more accurate results.

The proposed framework also helps to identify potential directions for future research. These include algorithms designed for inter-primitive generalization, generalization to large variabilities in movement performance, adaptive movement templates, and the creation of public datasets with temporal ground truth segments with a wide range of population types.

REFERENCES

[1] V. Krüger, D. Kragic, A. Ude, and C. Geib, "The meaning of action: A review on action recognition and mapping," *Adv Robotics*, vol. 21, pp. 1473–501, 2007.

[2] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robot Auton Syst*, vol. 57, pp. 469–83, 2009.

[3] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and reproduction of gestures by imitation," *IEEE Robot Autom Mag*, vol. 17, pp. 44–54, 2010.

[4] V. Krüger, D. L. Herzog, S. Baby, A. Ude, and D. Kragic, "Learning actions from observations," *IEEE Robot Autom Mag*, vol. 17, pp. 30–43, 2010.

[5] R. Houmanfar, M. E. Karg, and D. Kulić, "Movement analysis of rehabilitation exercises: Distance metrics for measuring patient progress," *IEEE Syst J*, 2014, in Print.

[6] M. Karg, W. Seiberl, F. Kreuzpointner, J.-P. Haas, and D. Kulić, "Clinical gait analysis: Comparing explicit state duration HMMs using a reference-based index," *IEEE Trans Neural Syst Rehabil Eng*, vol. 23, pp. 319–31, 2015.

[7] N. C. Krishnan, D. Colbry, C. Juillard, and S. Panchanathan, "Real-time human activity recognition using tri-axial accelerometers," in *Sens Signal Inform Proc Wkshp*, 2008.

[8] Z. Li, Z. Wei, W. Jai, and M. Sun, "Daily life event segmentation for lifestyle evaluation based on multi-sensor data recorded by a wearable device," in *IEEE Conf Eng Med Biol Soc*, 2013, pp. 2858–61.

[9] K. M. Newell and A. B. Slifkin, *Motor Behavior and Human Skill: A Multidisciplinary Approach*. Human Kinetics, 1998, ch. The Nature of Movement Variability, pp. 143–60.

[10] D. A. Winter, *Biomechanics and Motor Control of Human Gait: Normal, Elderly and Pathological*. Waterloo, 1991.

[11] F. Meier, E. Theodorou, F. Stulp, and S. Schaal, "Movement segmentation using a primitive library," in *IEEE Conf Intell Robot Syst*, 2011, pp. 3407–12.

[12] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Trans Syst Man Cybern C, Appl Rev*, vol. 37, pp. 311–24, 2007.

[13] M. A. R. Ahad, J. K. Tan, H. S. Kim, and S. Ishikawa, "Human activity recognition: Various paradigms," in *Conf Contr Automat Syst*, 2008, pp. 1896–901.

[14] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Commun Surveys Tutorial*, vol. 15, pp. 1192–209, 2013.

[15] D. Kulić, D. Kragic, and V. Krüger, "Learning action primitives," in *Vis Anal Humans*. Springer, 2011, pp. 333–53.

[16] T. B. Moeslund, A. Hilton, and V. Krger, "A survey of advances in vision-based human motion capture and analysis," *Comput Vis Image Und*, vol. 104, pp. 90–126, 2006.

[17] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *IEEE Trans Circuits Syst Video Technol*, vol. 18, pp. 1473–88, 2008.

[18] R. Poppe, "A survey on vision-based human action recognition," *Image Vision Comput*, vol. 28, pp. 976–90, 2010.

[19] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Comput Vis Image Und*, vol. 115, pp. 224–41, 2011.

[20] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artif Intell Rev*, vol. 43, pp. 1–54, 2015.

[21] A. Fod, M. J. Matarić, and O. C. Jenkins, "Automated derivation of primitives for movement classification," *Autonomous Robots*, vol. 12, pp. 39–54, 2002.

[22] T. Zhang, M. E. Karg, J. F.-S. Lin, D. Kulić, and G. Venture, "IMU based single stride identification of humans," in *IEEE Symp Robot Human Interactive Commun*, 2013, pp. 220–5.

[23] F. Worgotter, E. Aksoy, N. Kruger, J. Piater, A. Ude, and M. Tamosiunaite, "A simple ontology of manipulation actions based on hand-object relations," *IEEE Trans Auton Mental Develop*, vol. 5, pp. 117–34, 2013.

[24] N. Koenig and M. J. Matarić, "Behaviour-based segmentation of demonstrated tasks," in *Conf Devel Learn*, 2006.

[25] J. Barbič, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard, "Segmenting motion capture data into distinct behaviors," in *Graph Interface*, 2004, pp. 185–94.

[26] E. J. Keogh, S. Chu, D. Hart, and M. Pazzani, *Data Mining in Time Series Databases*. World Scientific, 2004, vol. 57, ch. Segmenting Time Series: A Survey and Novel Approach, pp. 1–22.

[27] B. Varadarajan, C. Reiley, H. Lin, S. Khudanpur, and G. Hager, "Data-derived models for segmentation with application to surgical assessment and training," in *Med Image Comput Comput Assist Interv*. Springer, 2009, pp. 426–34.

[28] S. Calinon and A. Billard, "Stochastic gesture production and recognition model for a humanoid robot," in *IEEE/RSJ Conf Intell Robot Syst*, vol. 3, 2004, pp. 2769–74.

[29] J. F.-S. Lin and D. Kulić, "On-line segmentation of human motion for automated rehabilitation exercise analysis," *IEEE Trans Neural Syst Rehabil Eng*, vol. 22, no. 1, pp. 168–80, 2014.

[30] W. Ilg, G. H. Bakir, J. Mezger, and M. A. Giese, "On the representation, learning and transfer of spatio-temporal movement characteristics," *Int J Hum Robot*, vol. 1, pp. 613–36, 2004.

[31] E. Berlin and K. Van Laerhoven, "Detecting leisure activities with dense motif discovery," in *ACM Conf Ubiquitous Comput*, 2012, pp. 250–9.

[32] J. F.-S. Lin, V. Joukov, and D. Kulić, "Human motion segmentation by data point classification," in *IEEE Conf Eng Med Biol Soc*, 2014, pp. 9–13.

[33] Motion Analysis Corp, "Cortex," www.motionanalysis.com, 2015.

[34] Vicon Motion Systems, "Tracker," www.vicon.com, 2015.

[35] D. Roetenberg, H. Luinge, and P. J. Slycke, "Xsens MVN: Full 6DOF human motion tracking using miniature inertial sensors," Xsens Tech, Tech. Rep., 2009.

[36] A. Burns, B. R. Greene, M. J. McGrath, T. J. O'Shea, B. Kuris, S. M. Ayer, F. Stroiescu, and V. Cionca, "SHIMMER: A wireless sensor platform for noninvasive biomedical research," *IEEE Sensors J*, vol. 10, pp. 1527–34, 2010.

[37] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE Multimedia*, vol. 19, pp. 4–10, 2012.

[38] E. J. Keogh and M. J. Pazzani, "Derivative dynamic time warping," in *SIAM Conf Data Mining*, 2001, pp. 1–11.

[39] A. Mueen, E. Keogh, Q. Zhu, S. Cash, and B. Westover, "Exact discovery of time series motifs," in *SIAM Conf Data Mining*, 2009, pp. 473–84.

[40] D. A. Winter, *Biomechanics and Motor Control of Human Movement*. Wiley, 2005.

[41] A. Aristidou, J. Cameron, and J. Lasenby, "Real-time estimation of missing markers in human motion capture," in *Conf Bioinform Biomed Eng*, 2008, pp. 1343–6.

[42] L. Herda, P. Fua, R. Plänkers, R. Boulic, and D. Thalmann, "Using skeleton-based tracking to increase the reliability of optical motion capture," *Human Movement Science*, vol. 20, pp. 313–41, 2001.

[43] F. Lv and R. Nevatia, "Recognition and segmentation of 3-D human action using HMM and multi-class AdaBoost," in *European Conf Comput Vis*, 2006, pp. 359–72.

[44] D. Kulić, W. Takano, and Y. Nakamura, "Online segmentation and clustering from continuous observation of whole body motions," *IEEE Trans Robot*, vol. 25, pp. 1158–66, 2009.

[45] D. Kulić and Y. Nakamura, "Incremental learning of human behaviors using hierarchical hidden Markov models," in *IEEE/RSJ Conf Intell Robot Syst*, 2010, pp. 4649–55.

[46] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from a single depth image," in *IEEE Conf Comput Vis Pattern Recog*, 2011.

[47] P. Hong, M. Turk, and T. S. Huang, "Gesture modeling and recognition using finite state machines," in *Conf Autom Face Gesture Recognit*, 2000, pp. 410–5.

[48] C. A. Ratanamahatana and E. Keogh, "Making time-series classification more accurate using learned constraints," in *SIAM Conf Data Mining*, 2004, pp. 11–22.

[49] F. Bashir, W. Qu, A. Khokhar, and D. Schonfeld, "HMM-based motion recognition system using segmented PCA," in *IEEE Conf Image Process*, 2005, pp. 1288–91.

[50] M. C. Boonstra, R. M. A. van der Slikke, N. L. W. Keijsers, R. C. van Lummel, M. C. de Waal Malefijt, and N. Verdonschot, "The accuracy of measuring the kinematics of rising from a chair with accelerometers and gyroscopes," *J Biomech*, vol. 39, pp. 354–8, 2006.

[51] H. Luinge and P. Veltink, "Measuring orientation of human body segments using miniature gyroscopes and accelerometers," *Med Biol Eng Comput*, vol. 43, pp. 273–82, 2005.

[52] J. F.-S. Lin and D. Kulić, "Human pose recovery using wireless inertial measurement units," *Physiol Meas*, vol. 33, pp. 2099–115, 2012.

[53] F. Chamroukhi, S. Mohammed, D. Trabelsi, L. Oukhellou, and Y. Amirat, "Joint segmentation of multivariate time series with hidden process regression for human activity recognition," *Neurocomput*, vol. 120, pp. 633–44, 2013.

[54] M. Yuwono, S. W. Su, B. D. Moulton, and H. T. Nguyen, "Unsupervised segmentation of heel-strike IMU data using rapid cluster estimation of wavelet features," in *IEEE Conf Eng Med Biol Soc*, 2013.

[55] P. Brossier, J. P. Bello, and M. D. Plumbley, "Real-time temporal segmentation of note objects in music signals," in *Comput Music Assoc*, 2004.

[56] A. Laudanski, S. Yang, and Q. Li, "A concurrent comparison of inertia sensor-based walking speed estimation methods," in *IEEE Conf Eng Med Biol Soc*, 2011, pp. 3484–7.

[57] F. De la Torre, J. Hodgins, A. Bargteil, X. Martin, J. Macey, A. Collado, and P. Beltran, "Guide to the carnegie mellon university multimodal activity (cmu-mmac) database," Carnegie Mellon Univ, Tech. Rep. 22, 2008.

[58] M. Tenorth, J. Bandouch, and M. Beetz, "The TUM kitchen data set of everyday manipulation activities for motion tracking and action recognition," in *IEEE Conf Comput Vis Wkshp*, 2009, pp. 1089–96.

[59] I. M. Bullock, T. Feix, and A. M. Dollar, "The yale human grasping dataset: Grasp, object, and task data in household and machine shop environments," *Int J Robot Res*, vol. 34, pp. 251–5, 2014.

[60] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Berkeley MHAD: A comprehensive multimodal human action database," in *IEEE Wkshp Applic Comput Vis*, 2013, pp. 53–60.

[61] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: archive.ics.uci.edu/ml

[62] Carnegie Mellon Univ, "CMU graphics lab motion capture database," mocap.cs.cmu.edu.

[63] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, "Documentation mocap database HDM05," Universität Bonn, Tech. Rep. CG-2007-2, 2007.

[64] I. S. Vicente, V. Kyrki, D. Kragic, and M. Larsson, "Action recognition and understanding through motor primitives," *Adv Robotics*, vol. 21, pp. 1687–707, 2007.

[65] J. M. Chaquet, E. J. Carmona, and A. Fernández-Caballero, "A survey of video datasets for human action and activity recognition," *Comput Vis Image Und*, vol. 117, pp. 633–59, 2013.

[66] D. G. E. Robertson, G. E. Caldwell, J. Hamill, G. Kamen, and S. N. Whittlesey, *Research Methods in Biomechanics*. Human Kinetics, 2014.

[67] A. Ataya, P. Jallon, P. Bianchi, and M. Doron, "Improving activity recognition using temporal coherence," in *IEEE Conf Eng Med Biol Soc*, 2013, pp. 4215–8.

[68] M. Li and A. M. Okamura, "Recognition of operator motions for real-time assistance using virtual fixtures," in *Symp Haptic Interfaces Virtual Environment Teleoperator Syst*, 2003, pp. 125–31.

[69] Z. Liu, Y. Wang, and T. Chen, "Audio feature extraction and analysis for scene segmentation and classification," *J VLSI Signal Process Syst Signal Image Video Technol*, vol. 20, pp. 61–79, 1998.

[70] S. Baby and V. Krüger, "Primitive based action representation and recognition," in *Image Analysis*, ser. Lecture Notes in Computer Science. Springer, 2009, vol. 5575, pp. 31–40.

[71] O. C. Jenkins and M. Mataric, "Deriving action and behavior primitives from human motion data," in *IEEE/RSJ Conf Intell Robot Syst*, vol. 3, 2002, pp. 2551–6.

[72] A. Valtazanos, D. K. Arvind, and S. Ramamoorthy, "Comparative study of segmentation of periodic motion data for mobile gait analysis," in *Wireless Health*, 2010, pp. 145–54.

[73] A. Castellani, D. Botturi, M. Bicego, and P. Fiorini, "Hybrid HMM/SVM model for the analysis and segmentation of teleoperation tasks," in *IEEE Conf Robotics Automat*, vol. 3, 2004, pp. 2918–23.

[74] S. Ekvall, D. Aarno, and D. Kragic, "Online task recognition and real-time adaptive assistance for computer-aided machine control," *IEEE Trans Robot*, vol. 22, pp. 1029–33, 2006.

[75] M. G. Pandy, "Computer modeling and simulation of human movement," *Annu Rev Biomed Eng*, vol. 3, pp. 245–73, 2001.

[76] R. Lan and H. Sun, "Automated human motion segmentation via motion regularities," *Vis Comput*, vol. 31, pp. 35–53, 2015.

[77] O. Amft, H. Junker, and G. Tröster, "Detection of eating and drinking arm gestures using inertial body-worn sensors," in *IEEE Symp Wearable Comp*, 2005, pp. 160–3.

[78] A. Vögele, B. Krüger, and R. Klein, "Efficient unsupervised temporal segmentation of human motion," *ACM SIGGRAPH Symp Comput Animat*, 2014.

[79] X. Zhao, X. Li, C. Pang, X. Zhu, and Q. Z. Sheng, "Online human gesture recognition from motion data streams," in *ACM Conf Multimedia*, 2013, pp. 23–32.

[80] S. Fothergill, H. M. Mentis, P. Kohli, and S. Nowozin, "Instructing people for training gestural interactive systems," in *ACM Conf Human Factors Comput Syst*, 2012, pp. 1737–46.

[81] F. Zhou, F. de la Torre, and J. K. Hodgins, "Hierarchical aligned cluster analysis for temporal clustering of human motion," *IEEE Trans Pattern Anal Mach Intell*, vol. 35, pp. 582–96, 2013.

[82] C. Lu and N. J. Ferrier, "Repetitive motion analysis: Segmentation and event classification," *IEEE Trans Pattern Anal Mach Intell*, vol. 26, pp. 258–63, 2004.

[83] J. Boyd and H. Sundaram, "A framework to detect and classify activity transitions in low-power applications," in *IEEE Conf Multimedia Expo*, 2009, pp. 1712–5.

[84] J. F.-S. Lin, V. Joukov, and D. Kulić, "Full-body multi-primitive segmentation using classifiers," in *IEEE Conf Humanoid Robot*, 2014, pp. 874–80.

[85] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min Knowl Discov*, vol. 2, pp. 121–67, 1998.

[86] J. Kohlmorgen and S. Lemm, "A dynamic HMM for on-line segmentation of sequential data," in *Adv Neural Inf Process Syst*, vol. 14, 2002, pp. 793–800.

[87] M. Pomplun and M. Matarić, "Evaluation metrics and results of human arm movement imitation," in *IEEE Conf Humanoid Robot*, 2000.

[88] J. Lieberman and C. Breazeal, "Improvements on action parsing and action interpolation for learning through demonstration," in *IEEE Conf Humanoid Robot*, 2004, pp. 342–65.

[89] G. Guerra-Filho and Y. Aloimonos, "A language for human action," *Comput*, vol. 40, pp. 42–51, 2007.

[90] L. Ricci, D. Formica, E. Tamilia, F. Taffoni, L. Sparaci, O. Capirci, and E. Guglielmelli, "An experimental protocol for the definition of upper limb anatomical frames on children using magnetoinertial sensors," in *IEEE Conf Eng Med Biol Soc*, 2013.

[91] M. Yamamoto, H. Mitomi, F. Fujiwara, and T. Sato, "Bayesian classification of task-oriented actions based on stochastic context-free grammar," in *Conf Autom Face Gesture Recogn*, 2006, pp. 317–22.

[92] Y. Hao, Y. Chen, J. Zakaria, B. Hu, T. Rakthanmanon, and E. Keogh, "Towards never-ending learning from time series streams," in *ACM Conf Knowl Discov Data Min*, 2013, pp. 874–82.

[93] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*. Wiley, 1958, vol. 2.

[94] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc IEEE*, vol. 77, pp. 257–86, 1989.

[95] B. Janus and Y. Nakamura, "Unsupervised probabilistic segmentation of motion data for mimesis modeling," in *IEEE Conf Adv Robot*, 2005, pp. 411–7.

[96] T. Aoki, G. Venture, and D. Kulic, "Segmentation of human body movement using inertial measurement unit," in *IEEE Conf Syst Man Cybern*, 2013, pp. 1181–6.

[97] M. A. Siegler, U. Jain, B. Raj, and R. M. Stern, "Automatic segmentation, classification and clustering of broadcast news audio," in *DARPA Broadcast News Wkshp*, 1997, pp. 97–99.

[98] T. Zhang and C. C. J. Kuo, "Audio content analysis for online audiovisual data segmentation and classification," *IEEE Speech Audio Process*, vol. 9, pp. 441–57, 2001.

[99] N. Peyrard and P. Bouthemy, "Content-based video segmentation using statistical motion models," in *British Mach Vis Conf*, 2002, pp. 1–10.

[100] O. Lartillot and P. Toiviainen, "A matlab toolbox for musical feature extraction from audio," in *Conf Digital Audio Effects*, 2007, pp. 237–44.

[101] S. B. Kang and K. Ikeuchi, "Toward automatic robot instruction from perception-temporal segmentation of tasks from human hand motion," *IEEE Trans Robot Autom*, vol. 11, pp. 670–81, 1995.

[102] D. Ormoneit, H. Sidenbladh, M. J. Black, and T. Hastie, "Learning and tracking cyclic human motion," *Adv Neural Inf Process Syst*, pp. 894–900, 2001.

[103] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans Pattern Anal Mach Intell*, vol. 22, pp. 4–37, 2000.

[104] U. Maurer, A. Smailagic, D. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *Wkshp Wearable Implantable Body Sensor Netw*, 2006, pp. 113–6.

[105] O. D. Lara, A. J. Pérez, M. A. Labrador, and J. D. Posada, "Centinela: A human activity recognition system based on acceleration and vital sign data," *Pervasive Mob Comput*, vol. 8, pp. 717–29, 2012.

[106] Z.-Y. He and L.-W. Jin, "Activity recognition from acceleration data using AR model representation and SVM," in *Conf Mach Learning Cybernetics*, vol. 4, 2008, pp. 2245–50.

[107] D. Kragic, P. Marayong, M. Li, A. M. Okamura, and G. D. Hager, "Human-machine collaborative systems for microsurgical applications," *Int J Robot Res*, vol. 24, pp. 731–41, 2005.

[108] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans Speech Signal Process*, vol. 26, pp. 43–9, 1978.

[109] A. Veeraraghavan, R. Chellappa, and A. Roy-Chowdhury, "The function space of an activity," in *IEEE Conf Comput Vis Pattern Recognit*, vol. 1, 2006, pp. 959–68.

[110] H.-K. Lee and J. H. Kim, "An hmm-based threshold model approach for gesture recognition," *IEEE Trans Pattern Anal Mach Intell*, vol. 21, pp. 961–73, 1999.

[111] K. Bernardin, K. Ogawara, K. Ikeuchi, and R. Dillmann, "A sensor fusion approach for recognizing continuous human grasping sequences using hidden markov models," *IEEE Trans Robot*, vol. 21, pp. 47–57, 2005.

[112] A. Ganapathiraju, J. E. Hamaker, and J. Picone, "Applications of support vector machines to speech recognition," *IEEE Trans Signal Process*, vol. 52, pp. 2348–55, 2004.

[113] A. Stolcke and E. Shriberg, "Automatic linguistic segmentation of conversational speech," in *Conf Spoken Lang*, vol. 2, 1996, pp. 1005–8.

[114] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Trans Pattern Anal Mach Intell*, vol. 22, pp. 852–72, 2000.

[115] S. Baby, V. Krüger, and D. Kragic, "Unsupervised learning of action primitives," in *IEEE/RAS Conf Humanoid Robots*, 2010, pp. 554–9.

[116] S. H. Lee, I. H. Suh, S. Calinon, and R. Johansson, "Learning basis skills by autonomous segmentation of humanoid motion trajectories," in *IEEE/RAS Conf Humanoid Robot*, 2012.

[117] A. D. Wilson and A. F. Bobick, "Parametric hidden markov models for gesture recognition," *IEEE Trans Pattern Anal Mach Intell*, vol. 21, pp. 884–900, 1999.

[118] S. Chiappa and J. Peters, "Movement extraction by detecting dynamics switches and repetitions," in *Adv Neural Inf Process Syst*, 2010, pp. 388–96.

[119] E. B. Fox, M. C. Hughes, E. B. Sudderth, and M. I. Jordan, "Joint modeling of multiple time series via the beta process with application to motion capture segmentation," *Ann Appl Stat*, vol. 8, pp. 1281–313, 2014.

[120] O. C. Jenkins and M. J. Mataric, "Automated derivation of behavior vocabularies for autonomous humanoid motion," in *Joint Conf Autonom Agents Multiagent Syst*, 2003, pp. 225–32.

[121] L. Lu, H. J. Zhang, and H. Jiang, "Content analysis for audio classification and segmentation," *IEEE Speech Audio Process*, vol. 10, pp. 504–16, 2002.