2015

Segmenting Motion Capture Data Using a Qualitative Analysis

Durell Bouchard

Norman I. Badler
University of Pennsylvania, badler@seas.upenn.edu

Recommended Citation

Bouchard, D., & Badler, N. I. (2015). Segmenting Motion Capture Data Using a Qualitative Analysis. *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games,* 23-30. http://dx.doi.org/10.1145/2822013.2822039

MIG '15 was held November 16-18, 2015, in Paris.

Segmenting Motion Capture Data Using a Qualitative Analysis

Abstract

Many interactive 3D games utilize motion capture for both character animation and user input. These applications require short, meaningful sequences of data. Manually producing these segments of motion capture data is a laborious, time-consuming process that is impractical for real-time applications. We present a method to automatically produce semantic segmentations of general motion capture data by examining the qualitative properties that are intrinsic to all motions, using Laban Movement Analysis (LMA). LMA provides a good compromise between high-level semantic features, which are difficult to extract for general motions, and lowlevel kinematic features, which often yield unsophisticated segmentations. Our method finds motion sequences which exhibit high output similarity from a collection of neural networks trained with temporal variance. We show that segmentations produced using LMA features are more similar to manual segmentations, both at the frame and the segment level, than several other automatic segmentation methods.

Keywords

human motion, motion capture, motion segmentation, Laban movement analysis

Disciplines

Computer Sciences | Engineering | Graphics and Human Computer Interfaces

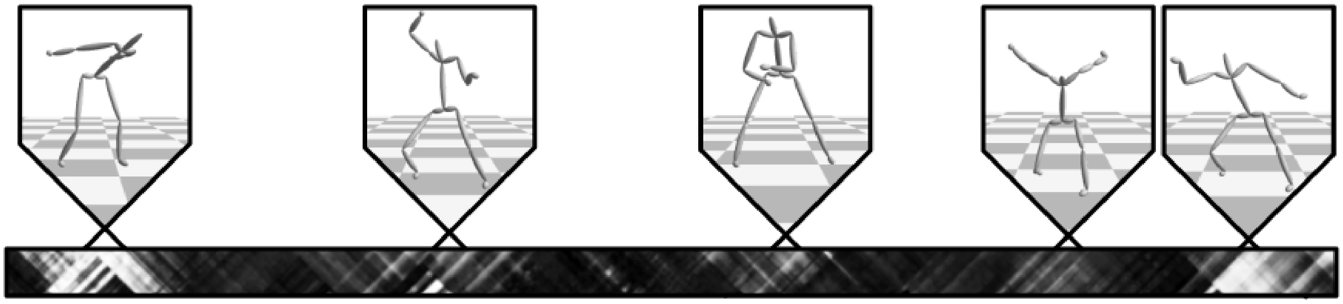
Comments

MIG '15 was held November 16-18, 2015, in Paris.

Segmenting Motion Capture Data Using a Qualitative Analysis

Durell Bouchard*
Roanoke College

Norman I. Badler†
University of Pennsylvania



Abstract

Many interactive 3D games utilize motion capture for both character animation and user input. These applications require short, meaningful sequences of data. Manually producing these segments of motion capture data is a laborious, time-consuming process that is impractical for real-time applications. We present a method to automatically produce semantic segmentations of general motion capture data by examining the qualitative properties that are intrinsic to all motions, using Laban Movement Analysis (LMA). LMA provides a good compromise between high-level semantic features, which are difficult to extract for general motions, and low-level kinematic features, which often yield unsophisticated segmentations. Our method finds motion sequences which exhibit high output similarity from a collection of neural networks trained with temporal variance. We show that segmentations produced using LMA features are more similar to manual segmentations, both at the frame and the segment level, than several other automatic segmentation methods.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: human motion, motion capture, motion segmentation, Laban movement analysis

1 Introduction

Segmentation of motion capture data is the process of partitioning a long stream of motion capture data into short, meaningful, contiguous subsequences. It is a critical step in both the production and execution of different types of games. Game designers can use motion capture to create realistic animations, but the long streams of motion capture data must be segmented before creating an animation state machine. Manually segmenting motion capture data is a laborious and time-consuming process. Furthermore, it is not feasible for applications with real-time motion capture such as motion controlled games.

Automatically segmenting motion capture data can be as trivial as detecting low-level kinematic features that correlate to the beginning and end of a motion, such as a drop in speed. However, this will not produce a semantic segmentation where segments correlate to human describable motions. Using a motion classifier to search for semantic segmentations is not feasible because the performance of a motion classifier depends on well-defined input motion data. For example, a motion classifier will fail if the input sequence contains only half of a motion or if it contains two different motions.

There are existing automatic segmentation techniques that are effective on general motion capture data but do not produce semantic segmentations. There are also techniques that produce semantic segmentations, but are not effective on general motion capture data. We present a method that uses a qualitative analysis of motion that both produces semantic segmentations of motion capture data and is effective on general motion capture data.

The proposed segmentation method is an extension of [Bouchard and Badler 2007]. Both are based on Laban Movement Analysis (LMA), but the segmentation method proposed here is entirely different. The proposed segmentation method searches for semantic motion sequence boundaries by using the confidence of an LMA classifier that is calculated by measuring the agreement of multiple LMA classifiers, each trained to be sensitive to different motion sequence boundaries. The method is evaluated by comparing it to manual segmentations and three other motion capture segmentation methods.

2 Related Work

A frequently used method of motion capture segmentation is to detect changes in simple kinematic features that correlate to segment boundaries. Some kinematic features are joint positions relative to other points or planes [Jenkins and Matarić 2003; Jenkins and Matarić 2004; Müller et al. 2005; Peng 2010; Reng et al. 2006], linear or angular velocity [Bernhardt and Robinson 2007; Fod et al. 2002; Ilg et al. 2004; Kwon and Shin 2005; Mezger et al. 2005; Mori and Uehara 2001; Osaki et al. 2000; Shiratori et al. 2003], linear or angular acceleration [Bindiganavale and Badler 1998; Guerra-Filho and Aloimonos 2006], and curvature [Zhao and Badler 2005; Wang et al. 2001]. Some changes in these kinematic features that are used to determine segment boundaries are local

[*] e-mail:bouchard@roanoke.edu
[†] email:badler@seas.upenn.edu

minima or maxima, zero-crossings, and thresholds. Kinematic segmentation methods have the advantage of being easy to implement, very fast, and effective on diverse motions. However, they are prone to over or under segmentation and frequently fail to identify semantic segments.

Other segmentation methods improve on kinematic segmentation methods by using more sophisticated analysis including change detection [Barbič et al. 2004; Barnachon et al. 2013; Endres et al. 2011; Gong et al. 2012; Nakata 2007; Ofli et al. 2014] and clustering [Barbič et al. 2004; Beaudoin et al. 2008; Lee and Elgammal 2006; López-Méndez et al. 2012; Vögele et al. 2014; Yun et al. 2008; Zhou et al. 2013]. Like the kinematic methods, these unsupervised learning methods are effective on general motion capture. In addition, these methods utilize a more global view of the motion capture data and can therefore recognize more sophisticated motions. However, these methods do not analyze the semantic content of motion and therefore do not always produce semantic segmentations.

Segmentation methods that do utilize the semantic content of motion capture use supervised learning techniques including nearest neighbor [Müller and Röder 2006], hidden Markov models [Lv and Xiao 2013; Xia et al. 2012], and conditional random fields [Heryadi et al. 2014]. These classification methods are capable of producing semantic segmentations, but they are not effective on general motions because it is difficult to train them to recognize any possible motion. Furthermore, the performance of a classifier is dependent on the training data's segment boundaries. This hinders the performance of these techniques because slightly shifted segment boundaries can produce dramatically different output.

3 Model

Kinematic segmentation methods and unsupervised learning segmentation methods are effective on a large domain of motion capture data but do not produce semantic segmentations. Supervised learning segmentation methods produce semantic segmentations but are not effective on all motion capture data and are prone to false positives. We present a method to segment motion capture data that, like other supervised learning methods, produces semantic segmentations. However, unlike other supervised learning methods, this method utilizes a classifier that is effective on a large domain of motion and is robust to input boundary shift errors.

The classifier is robust to input boundary shift errors because it consists of many classifiers, each trained to be sensitive to different input boundary shifts. The similarity of the classifiers' outputs is used to evaluate the classification confidence and to limit the number of false positives. The classifier is effective on general motion capture data because it is trained to recognize low-level, qualitative properties of Laban Movement Analysis that manifest consistently in all motions.

3.1 Laban Movement Analysis

Laban Movement Analysis (LMA) is a system for interpreting and describing human motion that focuses on the relationship between an individual's internal state and its effect on motion. LMA characterizes human motion as a combination of four components: Body, Shape, Space, and Effort. Body, Shape, and Space describe the articulation of the body, or what motion is performed. Effort describes the quality of a motion, or how it is performed, in terms of four different factors: Space, Time, Weight, and Flow.

Each of the LMA Effort factors is a continuum between an abundance of and a lack of a particular quality. The extremes of the continua are referred to as indulging and condensing elements and are summarized in Table 1 [Chi et al. 2000]. Any motion can be qualitatively described by quantifying how indulging or condensing it is for each of the four Effort factors. For example, an excited waving motion is more Light and more Sudden than a casual waving motion.

Table 1: *LMA Effort Factors and Elements*

	Indulging	Condensing
Space	**Indirect**: deviating, wandering	**Direct**: straight, focused
Time	**Sustained**: lingering, leisurely	**Sudden**: hurried, urgent
Weight	**Light**: buoyant, weightless	**Strong**: powerful, forceful
Flow	**Free**: uncontrolled, abandoned	**Bound**: restrained, rigid

Laban Movement Analysis is useful for producing a semantic segmentation of motion capture data because LMA descriptions are semantically meaningful and present in all motions. These two qualities have been demonstrated by the use of LMA for motion retrieval in large databases [Kapadia et al. 2013] and for motion style synthesis [Torresani et al. 2006]. Furthermore, it is possible to determine LMA Effort elements without identifying the shape of a motion, so creating a supervised learning classifier is tractable. LMA Effort is also a particularly useful marker for semantic segment boundaries because it depends on some motion qualities that manifest at the beginning or end of a motion. For example, an analysis of the beginning of a motion is required to characterize whether a motion is Sudden.

3.2 Classification Confidence Estimation

It is not possible to segment a motion capture sequence by using an LMA Effort classifier to search for all semantic subsequences because the classifier is sensitive to the boundaries of the input sequences. However, the sensitivity to input sequence boundaries can be leveraged to determine an output confidence estimation. If multiple classifiers, each trained to be sensitive to slightly different input boundary shifts, have the same output, then it is more likely to be a true positive. For example, one LMA classifier is trained on manually segmented data and a second LMA classifier is trained on the same data but with the first frame of every segment shifted forward by 1 frame. If an input sequence is segmented differently for the two classifiers, the first frame shifted forward 1 frame for the second classifier, and the output of the two classifiers is the same, then there is more confidence in the output. The possibility of two classifiers having the same incorrect output is mitigated by using many classifiers and computing the average similarity of all the outputs as a measure of classification confidence.

4 Method

LMA classifiers that use simple kinematic features and neural networks are used to determine the LMA Effort Elements of motion sequences. The confidence of the LMA classifier output is computed by measuring the agreement of many LMA classifiers, each of which is trained to be sensitive to different input sequence boundaries. The segmentation is computed by searching for sequences with high LMA classifier output confidence.

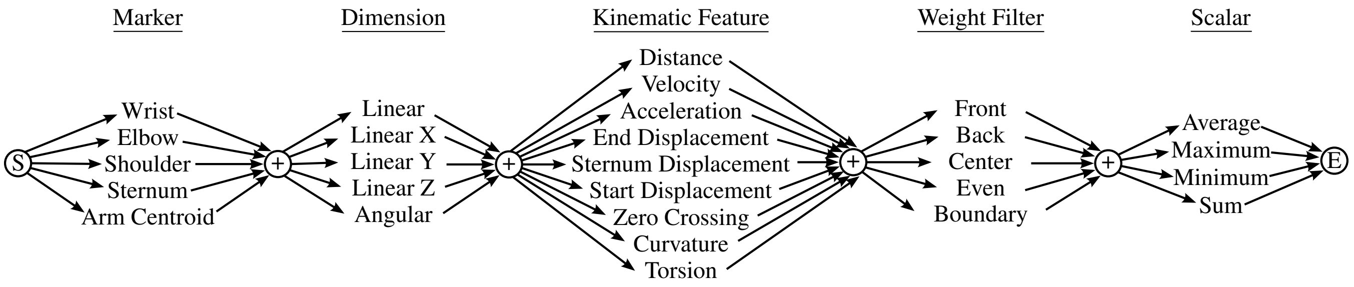


Figure 1: *Feature Generation Steps*

4.1 Laban Movement Analysis Effort Classifier

Two LMA experts assisted in creating a repertoire of primitive motions defined by Laban Movement Analysis. Each of the 12 motions was performed 12 times to isolate and emphasize the four different LMA Effort factors, Space, Time, Weight, and Flow, for each of the three different elements, condensing, neutral, and indulging. In total, 288 different motions were captured using Ascension Technology's active optical motion capture system, the ReActor, which tracks 30 markers at 30 Hz with an accuracy of 3 mm. The training data with descriptions of each of the motions is available for download.[1]

The input to an LMA Effort classifier is a sequence of motion capture data. In order to account for the differences in size and orientation of individuals, the data is normalized. The marker locations are translated such that the sternum marker is located at the origin and rotated about the vertical axis so that the sagittal planes are aligned. The marker locations are scaled by a factor estimated by averaging distances between joints that remain fairly constant, such as the elbow and the shoulder.

A single LMA Effort classifier consists of four neural networks, one for each LMA Effort factor. Four different machine learning techniques, k-nearest neighbor, naive Bayes classifier, support vector machine, and neural network, were evaluated. The neural network performed best and has a demonstrated effectiveness with LMA Effort elements [Zhao and Badler 2005]. Each of the neural networks consists of two layers of feedforward sigmoid units with three output nodes, one for each Effort element, indulging, condensing, and neutral. Multiple neural network outputs help reduce the occurrence of false positives because it is not possible for a motion to be both indulging and condensing for a particular Effort factor.

The inputs to the neural networks are features derived from the positions and orientations of the upper body's joints. Including the lower body has little effect on classifier performance as leg motion without corresponding arm and torso motion is rare in natural movements. The features are created from the descriptions of the LMA Effort elements. For example, the Space factor describes movement as Direct or Indirect, so the curvature of a hand's position over time is a useful feature. Each feature is computed for each joint at every frame of animation and is weighted to emphasize different temporal portions of a motion. For example, the Time factor describes movement as Sudden or Sustained, so the velocity of a hand at the beginning of the movement is emphasized by scaling the velocity of each frame by its normalized distance from the last frame. Each weighted feature is reduced to a single scalar value to eliminate the need for time warping by computing statistics such as the average or the sum. The feature set is a subset of all permutations of these steps, summarized in Fig. 1.

There are 4500 different permutations of the steps in the feature extraction process, which are too many features for an effective neural network. We select the most salient features using a method from Ruck et al. [1990]. In this method, the saliency of a feature is the amount of change in the output of the neural network compared to the amount of change in a feature. The assumption is that more salient features contribute to larger changes in the output of the neural network. There are too many features to compute saliency using a single network, so the saliency of each feature is computed using the average saliency of the feature over many random subsets of the potential features.

A sorted plot of feature saliency for each of the four LMA neural networks, Fig. 2, shows an exponential drop around the 100 most salient features. Figure 3 is a plot of the performance of the four neural networks versus an increasing number of the most salient features. The apexes of the curves in Fig. 3 are used to determine that the Space, Time, Weight, and Flow neural networks have 84, 72, 75, and 57 inputs, respectively. To further optimize the performance of the neural networks, Principal Component Analysis (PCA) is performed on these inputs to create a set of inputs with 95 percent variance, which reduces the number of inputs to 36, 23, 33, and 25, respectively. Finally, using a method from Park et al. [1996], the optimum number of hidden layer units is determined by performing PCA on the hidden layer weights. The number of hidden units with weights that together have a 98 percent variance are 3, 3, 5, and 8, respectively. The four neural networks together comprise an LMA Effort classifier.

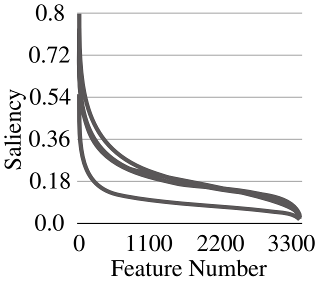


Figure 2: *Saliency of the Kinematic Features*

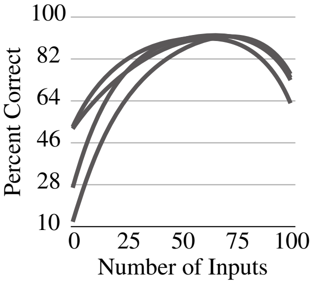


Figure 3: *Effect of Input Size on Neural Network Performance*

The training data is segmented using a hybrid manual/kinematic technique where segment boundaries are manually selected from hand linear velocity local minima. The neural networks are trained using backpropagation with an output range from 0.0 to 1.0. Training is stopped when the classification error rate for the validation set increases. The validation set is 12 segments selected from the training data so that it contains a condensing, indulging, and neutral segment for each of the LMA Effort Factors. Iyer and Rhinehart [1999] have shown that by randomizing the initial weights and

[1] http://cs.roanoke.edu/ bouchard/publications/LMASegmentation.zip

training 20 different neural networks there is a 99 percent probability that the network with the lowest validation error rate will be one of the 20 percent best for errors over the validation set. Therefore, each of the four neural networks is trained 20 times, each time with different random initial weights, and the network with the lowest validation error is used for the classifier. A classification is correct if the correct output is greater than the sum of the other two outputs. For example, if the Space neural network outputs 0.6, 0.1, and 0.1 for indulging, neutral, and condensing, and the correct output is indulging then the network is correct because $0.6 > 0.1 + 0.1$.

Table 2 shows the leave-one-out cross-validation classification rate for the neural networks. The Time network performs slightly better than the other three because it is easier to encapsulate whether a motion is Sudden or Sustained with kinematic features like velocity. The average leave-one-out cross-validation classification rate for the four neural networks is 92.4 percent.

Table 2: *LMA Classifier Leave-one-out Cross-validation Classification Rate*

Space	Time	Weight	Flow	Average
90.3%	95.8%	91.7%	91.7%	92.4%

4.2 Segmentation

To reduce false positives, the confidence of the LMA classifier's output is used to search for segment boundaries. The confidence is calculated as the mean deviance of many LMA classifiers, each trained with a different permutation of shift on input sequence boundaries. The mean deviance is calculated as:

$$d = \frac{1}{(2s+1)^2} \sum_{i=-s}^{s} \sum_{j=-s}^{s} |\mathbf{m} - \mathbf{x}_{i,j}| \qquad (1)$$

The scalar s is the maximum number of frames of boundary shift. The vector $\mathbf{x}_{i,j}$ is the 12-dimensional output of the LMA classifier trained with a shift of i frames on the start boundary and j frames on the end boundary. The vector $\mathbf{m}$ is the mean output of every permutation of the classifier, which is defined as:

$$\mathbf{m} = \frac{1}{(2s+1)^2} \sum_{i=-s}^{s} \sum_{j=-s}^{s} \mathbf{x}_{i,j} \qquad (2)$$

The maximum shift of $s = 5$ frames, 0.67 seconds, was experimentally determined to produce the best results. If the maximum shift is fewer, then there are fewer outputs in the mean deviation calculation and more false positives. If the maximum shift is greater, then data from neighboring segments alter the extracted kinematic features and lower the classification rate. With a shift between -5 and 5 frames on both the start and end boundaries there are $(2 \cdot 5 + 1)^2 = 121$ LMA classifiers used to compute the mean deviance. Each LMA classifier is four neural networks with 259 kinematic features as input and 12 LMA Effort elements as output. The process of computing the confidence for a motion sequence is illustrated in Fig. 4.

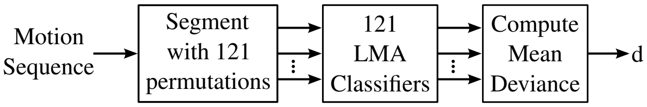


Figure 4: *LMA Classifier Confidence Estimation*

Figure 5 illustrates the average deviation of the 121 classifiers for all segments between 0.67 and 3.33 seconds in a 25-second test motion capture sequence that consists of a person repeatedly throwing a ball. Each pixel in the image represents the average deviation of the 121 LMA effort classifiers, darker pixels are less deviation, lighter pixels are more deviation. The y-axis is the start frame and the x-axis is the end frame of a segment. Only segments between 0.67 and 3.33 seconds are computed, hence the diagonal band across the whole test sequence. Dark vertical lines correspond to frames that are potential segment start boundaries and dark horizontal lines correspond to frames that are potential segment end boundaries.

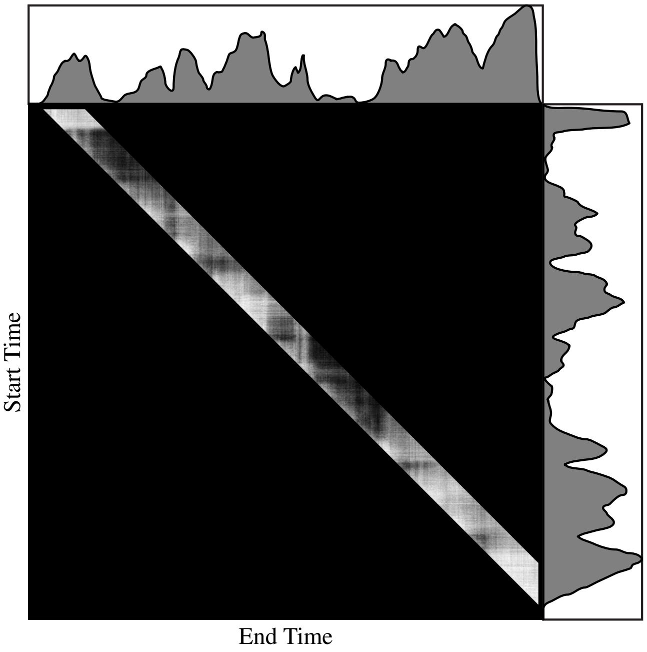


Figure 5: *Mean Deviance of Potential Segments in 25-Second Test Sequence*

The frames that correspond to dark vertical bands in Fig. 5, are found by summing all rows and columns and searching for local minima. The plots on the top and right of Fig. 5 are the sum of the columns and rows. Note that the troughs in the plots correspond to the dark bands. The locations of the troughs are found by smoothing the sums and searching for local minima.

Redundant local minima are culled by removing those that are not followed by a change in the deviation sum that exceeds an experimentally determined threshold. For the test sequence a threshold of 30 m/s yields the local minima that are illustrated as vertical lines in Fig. 6. A segmentation is produced by matching each start boundary local minimum with the closest subsequent end boundary local minimum as illustrated in Fig. 6. The resulting segmentation is represented as the horizontal box where the x-axis is time and the ovals represent the segments produced.

The test sequence in Fig. 6 consists of a throwing motion, so the segmentation is generated by only analyzing the right arm. More complex motions are segmented by producing a segmentation for each arm individually and merging similar segments. Two segmentations are merged by replacing segments that have similar boundaries with the average of the two segments. The threshold at which two segments are merged is a parameter that can be optimized to adjust the number and extent of overlapping segments.

The segmentation method as outlined is computationally expensive.

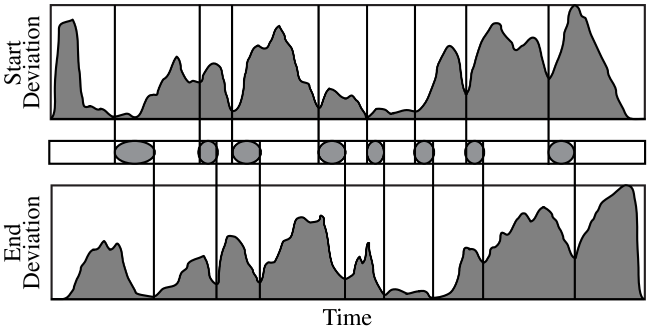


Figure 6: *Merging Boundaries to Produce Segmentation*

In the test sequence there are 74,520 segments that are between 0.67 and 3.33 seconds in length. Computing the average classifier deviation for all of those potential segments is more than 36 million neural net executions because there are 121 LMA Effort classifiers each consisting of four neural networks. The segmentation can be performed in real-time by only computing the average deviation for frames where the end effector speed is below a threshold. For example, computing all average classifier deviations of the 25-second test sequence in figure 5 takes more than 3 hours, but computing the segmentation takes 20.2 seconds if the search is reduced to 63 potential segment boundaries with low end effector speed. This modification has little effect on the segments produced and is used in the following analysis.

5 Analysis

The performance of the LMA segmentation method is measured in two different ways. First, the method's ability to identify segments that are semantic is measured by counting the number of segments that occur in the automatic LMA segmentation and manual segmentations. Second, the method's ability to select segment boundaries that are similar to a manual segmentation is measured by evaluating the performance of a simple motion classifier that is trained with manually segmented motion data. Using these measures with a diverse set of motion capture data, the LMA segmentation method is compared to other automatic segmentation methods.

5.1 Comparison to Manual Segmentation

The test sequences consist of three different motion capture sessions with three different performers. The sessions were designed to represent diverse activities. The test sequences are a 27-second monologue, a 38-second modern dance routine, and a 30-second martial arts routine. All three sequences can be viewed in the companion video.[2]The monologue sequence consists of many simple gesticulations used as emphasis that are very short, simple, and well differentiated. The modern dance routine is a fluid sequence of complex motions including jumps, turns, and poses. The martial arts sequence is a routine of actions including punching, kicking, and blocking that are a mix of simple discrete motions as well as complex fluid motions.

Manual segmentation has high intra-annotator agreement at the segment level, but low intra-annotator agreement at the frame level. That is, if a person segments a sequence of motion capture multiple times, the segments will contain the same semantic motions but the actual boundaries may be different. To compensate for the low intra-annotator agreement, individuals segmented each sequence four times and the average boundaries are used for comparison. Experiments show that four manual segment boundaries have an average deviation of 0.2 seconds. To compensate for the differences in the segmentations produced by different individuals, four individuals segmented each sequence. Four individuals was experimentally found to produce a sufficient diversity of segments. None of the individuals participated in manually segmenting the training data.

The participants used a program with an interactive, 3D view of the motion sequence and an editable time-line. Segments are specified by dragging across the time line, which allows users to create both disjoint and overlapping segments. The program includes instructions to find and mark all describable segments of motion in the three test sequences. The interface of the program can be seen in the companion video.[2]

The similarity of two segmentations is evaluated by counting the number of segments that match. Two segments match if corresponding boundaries are within one-half of a second of each other. Each segment can only match one other segment and if a segment matches multiple segments, then it is matched with the segment with the smallest difference in boundaries.

The granularity of the LMA-based segmentation method can be adjusted by modifying several parameters such as the mean deviation sum threshold, the Gaussian filter's standard deviations, and the hand speed local minima threshold. These parameters are optimized by a grid descent search on how similar the automatic segmentation is to manual segmentations. The similarity score is calculated as:

$$s = TP - (FP + FN)/2 \tag{3}$$

where TP, FP, and FN are the number of segment matches that are true positives, false positives, and false negatives, respectively. Equation 3 balances the number of matched segments, true positives, with half the total the number of unmatched segments, false positives and false negatives, which are roughly twice as common.

When the granularity of the LMA-based segmentation is optimized for the 12 manual segmentations of the three test motion capture sequences, 136 of the segments are matched. However, there are 75 unmatched segments in the LMA segmentation and 92 unmatched segments in the manual segmentations. This result is compared to three other automatic motion capture segmentation methods in Table 3.

Table 3: *Comparison of Automatic and Manual Segmentations*

	Velocity	KCS	PCA	LMA
Matched	119	151	131	136
Unmatched in Manual	109	77	97	92
Unmatched in Automatic	92	204	161	75
Similarity Score	18.5	10.5	2.0	52.5

The velocity method from Osaki et al. [2000] and Shiratori et al. [2003] is a kinematic method that defines boundaries as hand linear velocity local minima below some threshold and segment maximal velocity above another threshold. The Kinematic Centroid Segmentation (KCS) method from Jenkins and Matarić [2003; 2004] is an in-line approach that correlates segment boundaries to locally maximum hand displacement from the torso. The Principal Component Analysis (PCA) segmentation method from Barbič et al. [2004] is an in-line method that finds the largest segments with the smallest PCA projection error. It assumes that a sequence of motion capture that contains two different motions will reduce to

[2] http://cs.roanoke.edu/ bouchard/publications/LMASegmentation.mp4

a larger number of dimensions than a sequence that contains only one. The granularity of each of these segmentation methods is optimized using the same grid descent search as the LMA segmentation method.

The methods are adjusted to make its comparison to LMA Effort-based segmentation more logical. The PCA method is adjusted to only use the motion capture data from the left or right arm. The segmentations for the left and right arms are merged into one segmentation as is in the LMA Effort method. This partitioning and merging improves the performance of the PCA method. The KCS and PCA methods are in-line, so one segment ends where the next begins. They are both adjusted to limit the search for segment end boundaries to 300 frames, the same maximum segment size as the LMA method. If no segment end is found within 300 frames, the start frame is incremented and the process is restarted. This change creates segmentations where not every segment begins where the last one ended. This improves the performances of both methods.

All four methods have similar numbers of matched segments and manual unmatched segments. The KCS method matches more and misses fewer than the other methods and the Velocity method matches fewer and misses more than the other methods. This means that they find and miss roughly the same number of semantic segments. The KCS and PCA methods, however, have more automatically generated segments that are not semantic because both methods are in-line and generate far more segments than the LMA or Velocity methods. The LMA method has the highest similarity score, which takes into consideration both matched and unmatched segments.

5.2 Effects of Segmentation on Classification

Motion classification performance is affected by small changes in the segment boundaries of the input motion capture sequence. This effect is used to evaluate the segment boundaries produced by the LMA segmentation method and to demonstrate the performance of the LMA classifier in the context of a real-world application. A classifier will perform better if the boundaries of automatically segmented input sequences are similar to the boundaries of the manually segmented data on which the classifier was trained. The classifier consists of a neural network trained with 80 manually segmented motions of 12 different motion classes, four motions from each of the three test motion capture sequences from the previous section. The classes are the segments for which the four automatic segmentation methods have the most agreement.

The number of dimensions of motion capture data is too high for an effective classifier, and data dimension reduction techniques, such as PCA, are not sufficient. Instead, simple, meaningful features are extracted from the motion capture data using the spatial binary features from Müller and Röder [2006]. The neural network has 12 hidden sigmoid units and is trained using back propagation for 500 epochs with a learning rate of 0.3 and a momentum of 0.2. Each motion capture input sequence is segmented by each of the four segmentation methods from the previous section. The classification rates of the neural network for each segmentation method are summarized in Table 4.

Table 4: *Comparison of Segmentation Method Effects on Neural Network Classification Rate*

	Velocity	KCS	PCA	LMA
Dance Sequence	16.7%	0.0%	16.7%	33.3%
Monologue Sequence	100.0%	80.0%	100.0%	100.0%
Martial Arts Sequence	83.3%	83.3%	66.7%	83.3%
All Sequences	64.7%	52.9%	58.8%	70.6%

The performance of the neural network motion classifier is a measure of how similar segment boundaries are to manual segmentation boundaries. All four methods poorly classify the motions in the dance sequence and successfully classify the motions in the monologue sequence. The motions in the monologue sequence are simple and well differentiated while the motions in the dance sequence fluidly transition from one motion to the next which makes determining when one motion ends and another begins difficult. The LMA method, however, successfully classifies more dance motions than the other three methods. The LMA method also performs as well as or better than the other three methods on all three sequences and has the highest overall classification rate. This means that the LMA segmentation method's segment boundaries are more similar to the manually segmented boundaries.

6 Conclusion

The two major results from the experiments of the last section are summarized in the bar graphs of Figs. 7 and 8. Figure 7 is a plot of the similarity scores of the tested segmentation methods from Table 3 that shows LMA segmentation is the most similar to manual segmentations at the segment-level. Figure 8 is a plot of the motion classifier success rate for the tested segmentation methods from Table 4 that shows LMA segmentation is the most similar to manual segmentations at the frame-level. Improved automatic semantic segmentation for general motions can reduce the amount of time animators spend creating animation state machines and can simplify input handling for motion controlled games. Before being deployed in these applications, however, the performance of the LMA segmentation method needs to be evaluated on a larger, more diverse set of motions.

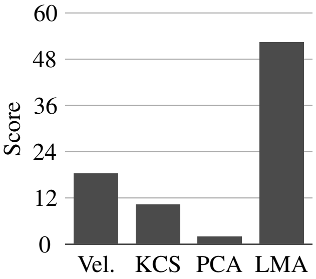


Figure 7: *Similarity Score: Segment-level Similarity to Manual Segmentation*

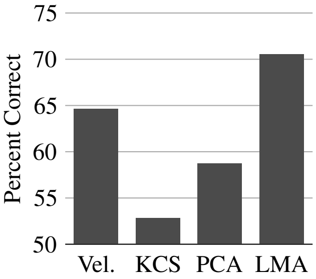


Figure 8: *Motion Classifier Performance: Frame-level Similarity to Manual Segmentation*

Despite performing better than the other tested segmentation methods, 36 percent of the LMA method's segments do not match manual segments. The 8 percent error rate of the LMA classifier causes some of these false positives, but does not fully account for the high false positive rate. It is likely that many of the false positive segments have LMA Effort elements, but do not have any high-level meaning and therefore are not considered semantic segments by the participants. The accuracy of the automatically generated LMA-based segmentation could be improved if it was combined with a more high-level, application-dependent classifier. Just like the LMA classifier uses a low-level kinematic feature as a first pass to limit potential segment boundaries, the LMA classifier could act as a first pass for a more high-level, application-dependent classifier. Using the LMA segmentation as the middle layer in a hierarchical segmentation scheme, would create better high-level, semantic segmentations.

References

BARBIČ, J., SAFONOVA, A., PAN, J.-Y., FALOUTSOS, C., HODGINS, J. K., AND POLLARD, N. S. 2004. Segmenting motion capture data into distinct behaviors. In *Proceedings of Graphics Interface*, Canadian Human-Computer Communications Society, 185–194.

BARNACHON, M., BOUAKAZ, S., BOUFAMA, B., AND GUILLOU, E. 2013. A real-time system for motion retrieval and interpretation. *Pattern Recognition Letters 34*, 15, 1789–1798.

BEAUDOIN, P., COROS, S., VAN DE PANNE, M., AND POULIN, P. 2008. Motion-motif graphs. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 117–126.

BERNHARDT, D., AND ROBINSON, P. 2007. Detecting affect from non-stylised body motions. In *Affective Computing and Intelligent Interaction*. Springer, 59–70.

BINDIGANAVALE, R., AND BADLER, N. I. 1998. Motion abstraction and mapping with spatial constraints. In *Modelling and Motion Capture Techniques for Virtual Environments*. Springer, 70–82.

BOUCHARD, D., AND BADLER, N. I. 2007. Semantic segmentation of motion capture using laban movement analysis. In *Intelligent Virtual Agents*, Springer, 37–44.

CHI, D., COSTA, M., ZHAO, L., AND BADLER, N. 2000. The emote model for effort and shape. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM, 173–182.

ENDRES, D., CHRISTENSEN, A., OMLOR, L., AND GIESE, M. A. 2011. Segmentation of action streams human observers vs. bayesian binning. In *KI 2011: Advances in Artificial Intelligence*. Springer, 75–86.

FOD, A., MATARIĆ, M. J., AND JENKINS, O. C. 2002. Automated derivation of primitives for movement classification. *Autonomous robots 12*, 1, 39–54.

GONG, D., MEDIONI, G., ZHU, S., AND ZHAO, X. 2012. Kernelized temporal cut for online temporal segmentation and recognition. In *Computer Vision–ECCV 2012*. Springer, 229–243.

GUERRA-FILHO, G., AND ALOIMONOS, Y. 2006. Understanding visuo-motor primitives for motion synthesis and analysis. *Computer Animation and Virtual Worlds 17*, 3-4, 207–217.

HERYADI, Y., FANANY, M. I., AND ARYMURTHY, A. M. 2014. A method for dance motion recognition and scoring using two-layer classifier based on conditional random field and stochastic error-correcting context-free grammar. In *Consumer Electronics (GCCE), 2014 IEEE 3rd Global Conference on*, IEEE, 771–775.

ILG, W., BAKIR, G. H., MEZGER, J., AND GIESE, M. A. 2004. On the representation, learning and transfer of spatio-temporal movement characteristics. *International Journal of Humanoid Robotics 1*, 04, 613–636.

IYER, M. S., AND RHINEHART, R. R. 1999. A method to determine the required number of neural-network training repetitions. *IEEE Transactions on Neural Networks 10*, 2, 427–432.

JENKINS, O. C., AND MATARIĆ, M. J. 2003. Automated derivation of behavior vocabularies for autonomous humanoid motion. In *Proceedings of the second international joint conference on autonomous agents and multiagent systems*, ACM, 225–232.

JENKINS, O. C., AND MATARIĆ, M. J. 2004. A spatio-temporal extension to isomap nonlinear dimension reduction. In *Proceedings of the twenty-first international conference on Machine learning*, ACM, 56.

KAPADIA, M., CHIANG, I.-K., THOMAS, T., BADLER, N. I., AND KIDER, JR, J. T. 2013. Efficient motion retrieval in large motion databases. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, 19–28.

KWON, T., AND SHIN, S. Y. 2005. Motion modeling for on-line locomotion synthesis. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, ACM, 29–38.

LEE, C.-S., AND ELGAMMAL, A. 2006. Human motion synthesis by motion manifold learning and motion primitive segmentation. In *Proceedings of the 4th international conference on Articulated Motion and Deformable Objects*, Springer, 464–473.

LÓPEZ-MÉNDEZ, A., GALL, J., CASAS, J. R., AND VAN GOOL, L. J. 2012. Metric learning from poses for temporal clustering of human motion. In *Proceedings of the British Machine Vision Conference*, British Machine Vision Association, 1–12.

LV, J., AND XIAO, S. 2013. Real-time 3d motion recognition of skeleton animation data stream. *International Journal of Machine Learning & Computing 3*, 5.

MEZGER, J., ILG, W., AND GIESE, M. A. 2005. Trajectory synthesis by hierarchical spatio-temporal correspondence: comparison of different methods. In *Proceedings of the 2nd symposium on Applied Perception in Graphics and Visualization*, ACM, 25–32.

MORI, T., AND UEHARA, K. 2001. Extraction of primitive motion and discovery of association rules from motion data. In *Proceedings of the 10th IEEE Workshop on Robot and Human Interactive Communication*, IEEE, 200–206.

MÜLLER, M., AND RÖDER, T. 2006. Motion templates for automatic classification and retrieval of motion capture data. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, 137–146.

MÜLLER, M., RÖDER, T., AND CLAUSEN, M. 2005. Efficient content-based retrieval of motion capture data. *ACM Transactions on Graphics 24*, 3, 677–685.

NAKATA, T. 2007. Temporal segmentation and recognition of body motion data based on inter-limb correlation analysis. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 1383–1388.

OFLI, F., CHAUDHRY, R., KURILLO, G., VIDAL, R., AND BAJCSY, R. 2014. Sequence of the most informative joints (smij): A new representation for human skeletal action recognition. *Journal of Visual Communication and Image Representation 25*, 1, 24–38.

OSAKI, R., SHIMADA, M., AND UEHARA, K. 2000. A motion recognition method by using primitive motions. In *Proceedings of the Fifth Working Conference on Visual Database Systems: Advances in Visual Information Management*, Kluwer, BV, 117–128.

PARK, Y. R., MURRAY, T. J., AND CHEN, C. 1996. Predicting sun spots using a layered perceptron neural network. *Neural Networks, IEEE Transactions on 7*, 2, 501–505.

PENG, S.-J. 2010. Motion segmentation using central distance features and low-pass filter. In *Proceedings of the IEEE Interna-*

tional Conference on Computational Intelligence and Security, IEEE, 223–226.

RENG, L., MOESLUND, T., AND GRANUM, E. 2006. Finding motion primitives in human body gestures. *Gesture in Human-Computer Interaction and Simulation*, 133–144.

RUCK, D. W., ROGERS, S. K., AND KABRISKY, M. 1990. Feature selection using a multilayer perceptron. *Journal of Neural Network Computing 2*, 2, 40–48.

SHIRATORI, T., NAKAZAWA, A., AND IKEUCHI, K. 2003. Rhythmic motion analysis using motion capture and musical information. In *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, IEEE, 89–94.

TORRESANI, L., HACKNEY, P., AND BREGLER, C. 2006. Learning motion style synthesis from perceptual observations. In *Advances in Neural Information Processing Systems*, NIPS Foundation, 1393–1400.

VÖGELE, A., KRÜGER, B., AND KLEIN, R. 2014. Efficient unsupervised temporal segmentation of human motion. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, The Symposium on Computer Animation, 167–176.

WANG, T.-S., SHUM, H.-Y., XU, Y.-Q., AND ZHENG, N.-N. 2001. Unsupervised analysis of human gestures. In *Second IEEE Pacific Rim Conference on Multimedia*, Springer, Beijing, China, IEEE, 174–181.

XIA, L., CHEN, C.-C., AND AGGARWAL, J. 2012. View invariant human action recognition using histograms of 3d joints. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, IEEE, 20–27.

YUN, S., PARK, A., AND JUNG, K. 2008. Graph-based high level motion segmentation using normalized cuts. *Proceedings of The World Academy of Science, Engineering and Technology 20*, 306–311.

ZHAO, L., AND BADLER, N. I. 2005. Acquiring and validating motion qualities from live limb gestures. *Graphical Models 67*, 1, 1–16.

ZHOU, F., DE LA TORRE, F., AND HODGINS, J. K. 2013. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*, 3, 582–596.