ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10446553

HUMAN MOTION CAPTURE DATA SEGMENTATION BASED ON ST-GCN

Xiuyun Ma[†] *Na Lv*[⋆†]

[†]School of Information Science and Engineering,
University of Jinan, Jinan 250022, China

ABSTRACT

Human motion sequence segmentation plays a crucial role in understanding and applying human motion capture(MoCap) sequences. However, most of the traditional segmentation methods are designed to find the locations where the motion features have changed significantly. When dealing with complex motion scenes, such methods often lead to inefficiency, inaccuracy, and limitations. To address these challenges, we propose an end-to-end sequence segmentation method based on the Spatial Temporal Graph Convolutional Networks(ST-GCN). Our network effectively extracts motion features from MoCap sequences, reduces dimensions through convolutional operations, and identifies segmentation points between different motions. Under the constraints of excessive segmentation and clip length, the optimal segmentation is achieved by combining three carefully designed loss functions. The proposed framework was evaluated on two benchmark datasets, CMU MoCap database and HDM05 dataset, and achieved better accuracy and robustness compared with existing methods.

Index Terms— Human motion state segmentation, Spatial temporal graph convolutional network, Motion capture database

1. INTRODUCTION

Human motion analysis[1] is a challenging area in computer vision that includes tasks such as motion recognition and object detection. It has a wide range of applications in many fields, such as medical diagnosis, movie animation, and security inspection. Long motion sequences are usually segmented before motion analysis[2], since the segmented motion fragments contain only a single type of motion, such as running, jumping, or clapping. The segmented motion segments are further used for action synthesis[3], recognition[4], retrieval and prediction[5]. As a result, the computer vision community has devoted considerable effort to the research and development of action segmentation techniques.

However, current research in human motion segmentation mainly relies on traditional segmentation methods[6][7][8]. These methods typically involve manual feature extraction and segmentation based on feature changes, which can result in low efficiency and accuracy. In recent years, the availability of low-cost depth cameras such as Kinect, along with advances in human pose estimation from depth images, has made it easier to extract 3D skeletal sequences[9]. This has opened up new avenues for the study of human motion. The development of deep learning has also brought new breakthroughs in automating and improving the performance of motion data processing. However, the accuracy of current segmentation methods is not high, and the problem lies in excessive segmentation.

In this paper, we propose an end-to-end human motion capture(MoCap) sequence segmentation method based on the Spatial Temporal Graph Convolutional Networks(ST-GCN)[10].Our method uses a novel segmentation strategy that is more closely related to human cognition. It has been validated on the CMU MoCap[11] database and the HDM05 database[12], both of which showed promising results and provided strong support for subsequent analysis and application.

The main contributions of this paper can be summarized as follows:

- We propose a new motion sequence segmentation method that is more in line with human cognition.
- A novel end-to-end deep learning network framework for the segmentation of MoCap sequences, which achieves both feature extraction and segmentation point estimation.
- Two novel loss functions, the segment intersection loss function and the intra-segment coupling loss function, are proposed to alleviate over-segmentation and enhance segmentation quality. The use of these loss functions demonstrates a clear improvement in the performance of the segmentation algorithm.

2. RELATED WORK

Sequence segmentation work first arose within the realm of video processing, and since then, many segmentation methods have been developed in this field. More recently, with the advancements in motion capture technology, some segmentation methods for MoCap data have also emerged. Considering the distinct data characteristics and application scenarios of video and MoCap data, the segmentation techniques for these two types of sequence data significantly differ. In the following section, we will provide a brief overview of the segmentation methods employed in both of these fields.

2.1. Video Segmentation

Video segmentation primarily employs motion recognition algorithms. The basic idea is to make fine-grained predictions and identify the action type for each frame. These predictions subsequently group adjacent frames with the same action type to achieve segmentation. This domain has explored several models, such as TCN[13], MS-TCN[14], GTRM [15], GAN [16], etc. Specifically, the TCN performs time modeling to predict labels for each frame, the MS-TCN improves segmentation accuracy by utilizing stacked TCN modules to extract high-dimensional features, the GTRM network employs graph convolution to predict video frames, and the GAN network achieves frame-wise prediction using coupled adversarial generation networks. These approaches have demonstrated promising outcomes, making them effective in this research field.

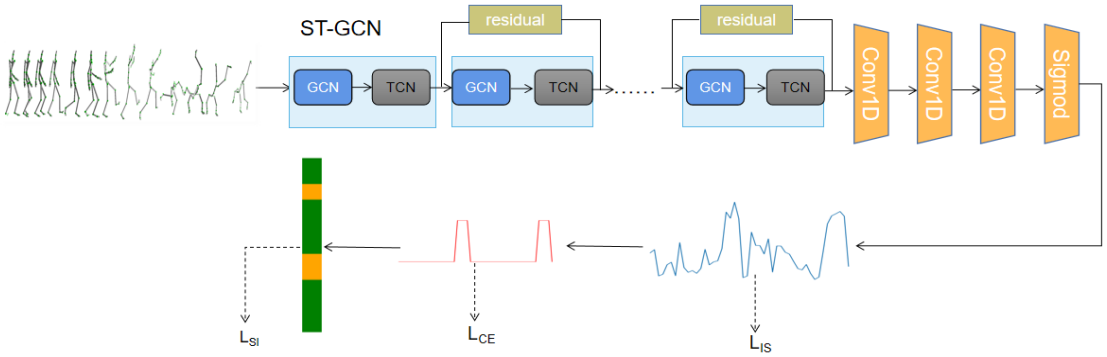

Fig. 1. Overview of proposed framework

2.2. MoCap Data Segmentation

MoCap data segmentation methods can be roughly divided into three categories.

The first category of methods is based on clustering techniques. For example, after reducing the data dimensions using PCA[17] and PPCA[18], clustering techniques such as K-means[19] and DBSCAN[20] were applied to split the long sequence into separate motion subsequences. The ACA (Aligned Cluster Analysis) algorithm [6] directly clusters the raw data. Based on the ACA method, the HACA (Hierarchical Aligned Cluster Analysis) [8] techniques utilized a bottom-up strategy [21] to merge clusters progressively, thereby creating a hierarchical structure which then generates segmentation results through termination conditions [8]. The DCNN [22] pre-processes motion capture data and subsequently optimizes segmentation results by using a clustering-based post-processing method.

The second category of method is based on dynamic change probability model. These methods include GMM-HMM (Gaussian Mixture model Hidden Markov model) [7], HBSVS (Hash Coding Based On State Variable Selection) [23], DSDM(Deep Switching Dynamical model)[24], etc. The GMM-HMM combines the ideas of Gaussian Mixture model and Hidden Markov Model, and regards human motion data as a stochastic process. The HBSVS uses hash coding technology to encode human motion data. The DSDM models each action segment as a state to achieve action segmentation.

The third category of methods is based on kinematic features such as velocity, acceleration, angular changes, and so on. These methods assume that similar kinematic features correspond to similar actions. For example, the TS-WMCS[25] algorithm uses class curvature descriptors to segment human motion sequences. The ASSL[26] uses 3D skeletal data for spatial modeling by LSTM. The UTSSMD[27] uses joint distance to represent bone motion data and uses hierarchical clustering algorithm for unsupervised time segmentation of bone motion data.

3. METHOD

Our data set is represented as $f = X_1, X_2, X_3...X_n$, where $X_i(1 <= i <= n)$ represents a long sequence with T frames and n is the number of MoCap sequences. And we have completed the completion work for sequences with different frame numbers. Unlike other segmentation methods, our method does not recognize the motion category of each frame. Our goal is to find the segmentation points, in other words, to classify each frame whether it is a segmentation point or not. For each sequence X_i, we predict $\hat{Y}_i = \{\hat{y}_t | \hat{y}_t \in (0,1)\}_{t=1}^{T}$, then obtain the prediction results of the whole dataset $\hat{Y} = \hat{Y}_1, \hat{Y}_2, ...\hat{Y}_n$.

3.1. Data Preprocessing

In a long motion sequence consisting of multiple behaviors, there is a natural transition between any two adjacent behaviors. This transition may span dozens or more frames. Based on this observation, we manually annotated the MoCap data using visual tools. For example, a sequence containing running and jumping is labeled as { 0,0,0,0,1,1,1,0,0,0,0}, where the first few '0's represent running, the fifth to seventh frames whose value is '1' represent the transition, and '0's in the last few frames represent jumping. In this paper, we refer to a continuous '1' as a segmented segment.

Regarding the data format, each human MoCap sequence stores the skeletal structure and the rotational motions of its joints. We use forward kinematics to convert these into trajectory information for each joint. To adapt to our baseline network, a MoCap sequence is converted into a matrix $X \in R^{T*V*C*M}$, where T is the number of frames, V is the number of joints, C is the number of feature channels for each joint, and M is the number of people in the sequence.

3.2. Motion Segmentation Network

Our framework consists of 10 stacked ST-GCN modules followed by a 3-layer CNN network, as shown in Fig.1. Each ST-GCN module consists of a GCN, a TCN and a residual connection, except for the first ST-GCN module. Since there are significant differences between the initial input and output of the first ST-GCN module, directly applying the residual connection may lead to information distortion or instability. Starting from the second ST-GCN module, a residual connection is added after each convolutional layer of the spatio-temporal map to promote information flow and information transmission.

The MoCap sequence X_i is fed into the stacked ST-GCN module to extract spatial and temporal features of the data through TCN and GCN, respectively. The 3-layer CNN after stacking ST-GCN modules performs 1x1 convolution operations. It will reduce the dimensionality of the data and capture local patterns and contextual information of features. Finally, the output layer uses the softmax function to convert the features of each frame into the probability of segmentation points. If the probability corresponding to the frame is

greater than a certain threshold, the frame is judged to be a segmentation point.

3.3. Constraints

To improve the segmentation accuracy of our model, we introduced two constraints.

3.3.1. Over-segmentation Constraint

The transition between adjacent behaviors typically takes several seconds. Once the number of consecutive predicted labels with '1' is less than a certain threshold, we consider there to be over-segmentation and reset these labels to '0'.

3.3.2. Segment Length Constraint

We observed that the duration of a single motion is usually more than 2 seconds. If the number of frames between the adjacent segmented segments is less than a certain threshold, it can be determined that the segmentation is incorrect. We merge the two short sequences into one long motion sequence.

3.4. Loss Function

We train the model by combining three loss functions. The overall loss function can be defined as

$$L = \alpha \cdot L_{CE} + \beta \cdot L_{SI} + \gamma \cdot L_{IS} \tag{1}$$

where α, β and γ are hyper-parameters.

The Cross Entropy Loss function(L_{CE}) [14][28] is designed at the frame level and used to measure the difference between the prediction results and the ground truth. The Cross Entropy Loss is defined as follows:

$$\begin{aligned} L_{CE} = -\sum_{i=1}^{N} (y_i \log(\hat{y}_i + 0.00001) \\ +(1 - y_i) \log(1 - \hat{y}_i * 0.99999)) \end{aligned} \tag{2}$$

where 0.00001 and 0.99999 are used to prevent the occurrence of $log0$.

The segment intersection loss function(L_{SI}) is designed at the segment level and is used to increase the amount of overlap between the predicted results and the ground truth, and to reduce erroneous segmentation. Specifically, we convert the predicted results and the ground truth into the segment intervals, denoted as $S = \{S_1, S_2, S_3, \ldots, S_q\}$ and $P = \{P_1, P_2, P_3, \ldots, P_p\}$, respectively. Here, q and p represent the number of segments in the predicted results and the ground truth. For each segment interval $S_i = [t_{i_s}, t_{i_e}]$ in the predicted results, we need to find a matching segment interval $P_j = [t_{j_s}, t_{j_e}]$ in the ground truth. The segment intersection loss function is designed as:

$$L_{SI} = \sum_{i=1}^{q} [(t_{i_e} - t_{i_s} - overlaylength)/(t_{i_e} - t_{i_s})] \tag{3}$$

where $overlaplength$ is the intersection of the predicted result segment S_i and the ground truth segment P_j, and $t_{i_e} - t_{i_s}$ is the length of the predicted result segment.

The intra-segment coupling loss function(L_{IS}) is also designed at segment level and used to measure the gap within and between segmentation segments in the predicted results, aiming to encourage the model to generate more accurate segmentation results. According to the ground truth, the predicted probability results of the sequence are divided into segments as follows: $G = \{G_1, G_2, G_3, \ldots, G_p\}$, where $G_j = [\hat{y}_s, \hat{y}_e]$. L_{IS} designed as:

$$L_{IS} = within\_ss/(between\_ss + 0.00001) \tag{4}$$

$between\_ss$ represents the discrepancy between different action segments, $within\_ss$ is the difference results between frames within the same segment. They are all obtained by calculating the covariance of the predicted probability $\hat{Y}$. The calculation formula is as follows:

$$between\_ss = \sum_{j=1}^{p-1} \left(\bar{h}_{j+1} - \bar{h_j}\right)^2 \tag{5}$$

$$within\_ss = \sum_{j=1}^{p} \sum_{i=s}^{e} \left(\hat{y}_i - \bar{h}_j\right)^2 \tag{6}$$

where $\bar{h_j}$ is the average of the predicted probabilities of all frames in the segment G_j.

4. EXPERIMENT

4.1. Datasets

We conducted experiments on two well-known MoCap datasets, namely CMU[11] and HDM05[12] MoCap databases. CMU MoCap database contains a large amount of MoCap data from 144 subjects. HDM05 database contains a total of more than three hours of systematically recorded MoCap data from more than 70 motion classes.

To accomplish the segmentation task, we selected long motion sequences containing multiple motion classes from two datasets. For the CMU MoCap database, we selected 121 sequences from subjects 1, 5, 6, 13, 14, 16, 40, 41, and so on. For the HDM05 dataset, we used 87 sequences from scenes 3, 4 and 5. The entire dataset is partitioned into training, validation, and test sets in a 8:1:1 ratio.

4.2. Evaluation Indicators

To evaluate our model, we used three common evaluation indicators in action segmentation[29]: Precision, Recall and F1 score(F1).

We made some modifications to the calculation of True Positive (TP), False Positive (FP) and False Negative (FN) according to the specific situation of the segmentation task. In Fig.2, the red region represents a ground truth segment interval, while the black region represents a predicted segment interval. Fig.2(a), (b) and (c) show the intersection between these two intervals in three cases. In Fig.2(a), the formula for calculating the TP and FP can be as follows: $TP = (t_3 - t_2)/(t_4 - t_2)$, $FP = (t_4 - t_3)/(t_4 - t_2)$. The calculation method of figures (b) and (c) is similar to that of (a). For the

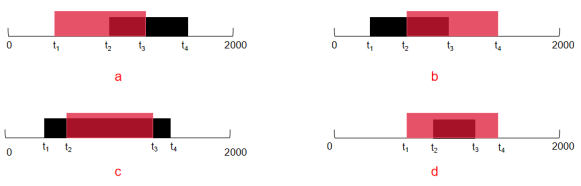

Fig. 2. Calculation of evaluation indicators

Table 1. Comparing with other methods in the CMU MoCap dataset

CMU Mocap	Precision	Recall	F1
TS-WMCS[25]	0.667	0.225	0.336
PCA[17]	0.778	0.123	0.212
PPCA[18]	**0.889**	0.235	0.378
HBSVS[23]	0.738	0.704	0.721
DSDM[24]	0.752	0.729	0.740
ASSL[26]	0.791	0.700	0.743
DCNN[22]	0.805	0.782	0.793
UTSSMD[27]	0.727	0.698	0.712
Our Approach	0.836	**0.981**	**0.903**

Table 2. Comparing with other methods in the HDM05 dataset.

HDM05	Precision	Recall	F1
TS-WMCS[25]	0.626	0.287	0.394
PCA[17]	0.714	0.327	0.449
PPCA[18]	**0.872**	0.249	0.387
HBSVS[23]	0.693	0.780	0.734
DSDM[24]	0.612	0.579	0.595
ASSL[26]	0.738	0.711	0.736
DCNN[22]	0.796	0.703	0.747
UTSSMD[27]	0.701	0.635	0.666
Our Approach	0.811	**0.896**	**0.851**

case in Fig.2(d), the segmentation is considered completely correct, so TP is 1 and FP is 0. During the traversal of the predicted results, if there are excessive segmentations, FP is incremented. Similarly, during the traversal of the ground truth labels, FN is incremented if it does not match in the predicted results.

Then the three evaluation indicators can be calculated by $Precision = TP/(TP + FP)$, $Recall = TP/(TP + FN)$ and $F1 = Precision * Recall * 2/(Precision + Recall)$.

4.3. Comparison with Other Methods

We compared our method with several state-of-the-art MoCap sequence segmentation methods on the CMU MoCap database (Table1) and the HDM05 database (Table2). From Table1, we see that our method performs well on all three evaluation metrics, outperforming state-of-the-art models on recall and F1-score with improvements of 0.199 and 0.110, respectively. We attribute this performance to the design of our constraints and loss functions, which effectively mitigate over-segmentation and miss-segmentation. Although the PPCA method shows higher precision compared to our method, its recall rate is very low, which infers that this method has missed many segmentation points. In addition, when we apply it to complex long sequences, it exhibits over segmentation.In summary, the segmentation efficiency of the PPCA method is not high.

From Table 2, it was observed that PCA, PPCA and TS-WMCS methods suffer from significant problems of over-segmentation and segmentation errors. While compared to them, other methods are slightly better, but their accuracy is not high. Our proposed method effectively addresses the limitations observed in the aforementioned approaches. As shown in Fig.3 , our model can correctly segment the majority of actions.

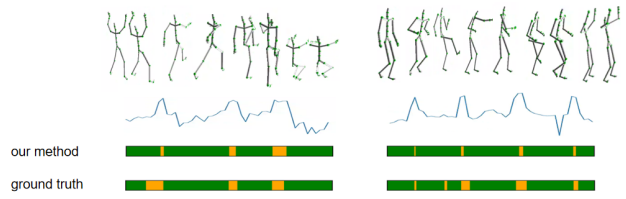

Fig. 3. We propose an example of the model segmentation process. The first row shows the predicted probability of the model. The second row shows the predicted segmentation results. The third row displays the ground truth.

Table 3. Study on the threshold of CMU Mocap Dataset

methods	Precision	Recall	F1
$+L_{CE}$	0.743	0.859	0.797
$+L_{CE} + L_{SPI}$	0.760	0.932	0.837
Our Approach	**0.836**	**0.981**	**0.903**

4.4. Ablation Experiment

To fully understand the impact of different components of the model, we performed ablation experiments on the CMU MoCap database by changing or deleting some components of the model and comparing their performance. In particular, the threshold mentioned in section 3.2 and the loss function were studied.

We designed the loss function as a combination of cross-entropy loss, segment intersection loss, and fragment intra-coupling loss. The experimental results (Table 3) prove that three components are essential to achieve good segmentation results. Our loss formulation leads to fewer over-segmentation errors by enforcing similar class probabilities between successive frames, resulting in smoother outputs. Furthermore, we observed the effect of different thresholds on the segmentation accuracy. As shown in Table 4, our method achieved the best result when the threshold was set to 0.547.

5. CONCLUSION

In this paper, we propose an end-to-end MoCap sequence segmentation method based on ST-GCN. We introduced two novel loss functions and constraints, which can greatly reduce the problem of over-segmentation and miss-segmentation. Our method outperforms the state-of-the-art methods in terms of overall segmentation effectiveness on both the CMU and the HDM05 MoCap databases. In the future, we plan to combine additional contextual information and apply it to motion video segmentation tasks.

Table 4. Research on the loss function of CMU MoCap Dataset

Threshold	Precision	Recall	F1
0.500	0.516	0.720	0.601
0.540	0.791	0.927	0.856
0.547	**0.836**	**0.981**	**0.903**
0.550	0.743	0.689	0.715
0.600	0.597	0.604	0.554

6. REFERENCES

[1] Huseyin Coskun, David Joseph Tan, Sailesh Conjeti, Nassir Navab, and Federico Tombari, "Human motion analysis with deep metric learning," 2018.

[2] Rongyi Lan and Huaijiang Sun, "Automated human motion segmentation via motion regularities," *The Visual Computer*, vol. 31, pp. 35–53, 2015.

[3] Hyun Joon Shin and Jehee Lee, "Motion synthesis and editing in low-dimensional spaces," *Computer Animation and Virtual Worlds*, vol. 17, no. 3-4, pp. 219–227, 2006.

[4] Samitha Herath, Mehrtash Harandi, and Fatih Porikli, "Going deeper into action recognition: A survey," *Image and vision computing*, vol. 60, pp. 4–21, 2017.

[5] Yu Kong, Dmitry Kit, and Yun Fu, "A discriminative model with multiple temporal scales for action prediction," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 596–611.

[6] Zhou Feng, JK Hodgins, et al., "Aligned cluster analysis for temporal segmentation of human motion," in *Automatic Face and Gesture Recognition. FG. 8th IEEE International Conference on*, 2008, pp. 1–7.

[7] Hilde Kuehne, Juergen Gall, and Thomas Serre, "An end-to-end generative framework for video segmentation and recognition," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016, pp. 1–8.

[8] Feng Zhou, Fernando De la Torre, and Jessica K Hodgins, "Hierarchical aligned cluster analysis for temporal clustering of human motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 582–596, 2012.

[9] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann, "Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2272–2281.

[10] Sijie Yan, Yuanjun Xiong, and Dahua Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proceedings of the AAAI conference on artificial intelligence*, 2018, vol. 32.

[11] Ralph Gross, *The cmu motion of body (mobo) database*, Carnegie Mellon University, The Robotics Institute, 2001.

[12] Meinard Müller, Tido Röder, and Michael Clausen, "Efficient content-based retrieval of motion capture data," in *ACM SIGGRAPH 2005 Papers*, pp. 677–685. 2005.

[13] Shaojie Bai, J Zico Kolter, and Vladlen Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[14] Yazan Abu Farha and Jurgen Gall, "Ms-tcn: Multi-stage temporal convolutional network for action segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3575–3584.

[15] Yifei Huang, Yusuke Sugano, and Yoichi Sato, "Improving action segmentation via graph-based temporal reasoning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 14024–14034.

[16] Harshala Gammulle, Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes, "Coupled generative adversarial network for continuous fine-grained action segmentation," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 200–209.

[17] Yan Ke and Rahul Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* IEEE, 2004, vol. 2, pp. II–II.

[18] Li Qu, Li Li, Yi Zhang, and Jianming Hu, "Ppca-based missing data imputation for traffic flow volume: A systematical approach," *IEEE Transactions on intelligent transportation systems*, vol. 10, no. 3, pp. 512–522, 2009.

[19] J MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symposium on Math., Stat., and Prob*, 1965, p. 281.

[20] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu, "Dbscan revisited, revisited: why and how you should (still) use dbscan," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.

[21] Eamonn Keogh, "Fast similarity search in the presence of longitudinal scaling in time series databases," in *Proceedings Ninth IEEE International Conference on Tools with Artificial Intelligence*. IEEE, 1997, pp. 578–584.

[22] Hashim Yasin and Saqib Hayat, "Deepsegment: Segmentation of motion capture data using deep convolutional neural network," *Image and Vision Computing*, vol. 109, no. 8, pp. 104147, 2021.

[23] Yang Liu, Lin Feng, Muxin Sun, and Shenglan Liu, "Hashing based state variation for human motion segmentation," in *Computer Vision: Second CCF Chinese Conference, CCCV 2017, Tianjin, China, October 11–14, 2017, Proceedings, Part III*. Springer, 2017, pp. 627–638.

[24] Amirreza Farnoosh and Sarah Ostadabbas, "Dynamical deep generative latent modeling of 3d skeletal motion," 2021.

[25] Shenglan Liu, Lin Feng, Yang Liu, Hong Qiao, Jun Wu, and Wei Wang, "Manifold warp segmentation of human action," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1414–1426, 2017.

[26] David Ada Adama, Ahmad Lotfi, and Robert Ranson, "Adaptive segmentation and sequence learning of human activities from skeleton data," *Expert Systems with Applications*, vol. 164.

[27] Christian Lins, Sebastian M Müller, Max Pfingsthorn, Marco Eichelberg, and Andreas Hein, "Unsupervised temporal segmentation of skeletal motion data using joint distance representation," 2018.

[28] Yuchi Ishikawa, Seito Kasai, Yoshimitsu Aoki, and Hirokatsu Kataoka, "Alleviating over-segmentation errors by detecting action boundaries," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 2322–2331.

[29] Li Ding and Chenliang Xu, "Weakly-supervised action segmentation with iterative soft boundary assignment," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6508–6516.