MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Hand Graph Representations for Unsupervised Segmentation of Complex Activities

Das, P.; Kao, J.-Y.; Ortega, A.; Mansour, H.; Vetro, A.; Sawada, T.; Minezawa, A.

TR2019-009 March 29, 2019

Abstract

Analysis of hand skeleton data can be used to understand patterns in manipulation and assembly tasks. This paper introduces a graphbased representation of hand skeleton data and proposes a method to perform unsupervised temporal segmentation of a sequence of subtasks in order to evaluate the efficiency of an assembly task. We explore the properties of different choices of hand graphs and their spectral decomposition. A comparative performance of these graphs is presented in the context of complex activity segmentation. We show that the spectral graph features extracted from 2D hand motion data outperform the direct use of motion vectors as features. We also make the collected hand position data available to the research community to facilitate further development in this direction

IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)

HAND GRAPH REPRESENTATIONS FOR UNSUPERVISED SEGMENTATION OF COMPLEX ACTIVITIES

Pratyusha Das, Jiun-Yu Kao,
Antonio Ortega[*]

Dept. of Electrical and Computer Engineering
University of Southern California
Los Angeles, CA, USA

Tomoya Sawada[a]*, Hassan Mansour*[b]*,*
Anthony Vetro[b]*, Akira Minezawa* [a]

[a]Mitsubishi Electric Corporation,
Kamakura, Kanagawa, Japan
[b]Mitsubishi Electric Research Labs,
Cambridge, MA, USA

ABSTRACT

Analysis of hand skeleton data can be used to understand patterns in manipulation and assembly tasks. This paper introduces a graph-based representation of hand skeleton data and proposes a method to perform unsupervised temporal segmentation of a sequence of sub-tasks in order to evaluate the efficiency of an assembly task. We explore the properties of different choices of hand graphs and their spectral decomposition. A comparative performance of these graphs is presented in the context of complex activity segmentation. We show that the spectral graph features extracted from 2D hand motion data outperform the direct use of motion vectors as features. We also make the collected hand position data available to the research community to facilitate further development in this direction.

***Index Terms*—** Hand graph, Graph based representation, Complex activity, Unsupervised online segmentation.

1. INTRODUCTION

Activity monitoring is an important topic in computer vision, and can be applied to various tasks, from surveillance to work-flow monitoring or quality control inspection. For example, in an industrial environment, it may be important to monitor workers for quality control purposes or for accident prevention. Activity segmentation continues to be a challenging task, especially for very fine motor activities. Complex activities such as assembly tasks [1], food preparation [2], surgical procedures [3], etc can often be broken down into a sequence of smaller sub-tasks. However, action segmentation often needs to be performed without prior knowledge of the task or sub-tasks involved, including the number of action classes, so that an unsupervised segmentation method is desired.

Temporal action segmentation is often tackled by designing a complete processing pipeline where video is captured and analyzed in order to provide segments corresponding to individual actions [4], [5]. These systems are trained with video data representing the actual tasks that have to be detected. In practice good performance can only be achieved if there is enough task-specific data for training. Since there is a large variety of tasks across application domains, this may not be always feasible. In particular, there is a significant overhead in generating data for training. For example, an industrial activity segmentation system may have to be retrained every time tasks performed by workers change.

In this paper, we are motivated by the observation that there has been significant progress in the development of *generic* video-based human motion trackers, with OpenPose [6] an excellent representative example. These systems are highly optimized and trained for generic tracking tasks, for which sufficient amounts of training data exist. Based on this, we propose to decouple tracking from activity segmentation, and develop systems that use a standard tracker, such as OpenPose, as an off-the-shelf first stage in the action segmentation process. OpenPose directly extracts human skeleton data, along with face and hands key points providing their position in the 2D space of the video frame. Thus, we propose to develop unsupervised action segmentation techniques that use *only* the OpenPose output, i.e., key points associated to human motion. For such a system, training data is no longer required. This has further advantages in terms of privacy, e.g., video used by OpenPose could be discarded after processing since it is not needed for activity segmentation.

Low-level representation of skeleton data using graphs [7] has become increasingly popular for human motion understanding. Since graphs can efficiently model the skeleton structure, improved representations can be obtained relative to standard features based on distance and skeleton joint angles [8]. It has been shown that transforms such as the Graph Fourier Transforms (GFT) [9] and Graph Wavelet Transforms (GWT) [10] can be used to extract meaningful features, for applications such as gait recognition [11].

Though there are various graph models available for human skeleton data [12] to analyze *MoCap* data, to the best of our knowledge there is no such representation available for hand-based activity analysis [13]. In this paper, we introduce a graph representation of hand skeleton data and propose three different topologies for hand graph construction, which can efficiently capture the hand motion, coordination of the fingers and also the intra-hand motion. We also analyze interesting spectral properties of these graphs. These graphs are used to extract features from hand motion, where feature extraction is completely data independent and unsupervised.

It is important to note that most of the online segmentation approaches found in the literature are supervised. In [14], Bargi et al. proposed an online HDP-HMM scheme for joint segmentation and classification of actions while exploring new classes as they occur and tuning the parameters using a feedback loop. Mumtaz et al. constructed a vocabulary of primitive actions in [15] during training and performed the online segmentation, matching the input sequence with the existing vocabulary. In [16], Liu et al. proposed a martingale-based method to select the characteristic frames and used supervised method for segmentation. In this paper, we exploit

[*]This work has been funded in part by Mitsubishi Electric Corporation

the idea of Bayesian Information Criteria (BIC) [17] for online unsupervised segmentation using graph features.

In this paper we use a new fine motor activity dataset to evaluate the segmentation performance of proposed hand graphs. This data set consists of videos of 11 subjects performing a robot toy assembling task which has a fixed number of sequential sub-tasks. 2D key points of the hand are extracted using OpenPose from the captured videos. The online action segmentation system only uses the 2D position data of the hands and then computes 2D motion vectors, which are later used as the graph signal for feature computation. The key point dataset [18] is publicly available to the research community. To the best of our knowledge there is a limited number of a hand datasets available. The best known one, EgoHands [19] cannot be used in our context because it is captured by a moving google glass and the motion of the camera is not available.

2. PROPOSED APPROACH

We start with a detailed description of the graph based representation of hands and its application to activity segmentation. The proposed system uses no video information, and relies completely on the 2D hand key points extracted by OpenPose. Frames where OpenPose fails to extract hand key points because of occlusion are ignored.

2.1. Feature extraction

2.1.1. Graph construction

OpenPose provides 2D coordinates of the hand key-points, but we have a choice of how to create a graph to analyze these data. Inspired by the structure of human hands, we consider three alternative hand graphs. First, we construct graph $\mathcal{G}_{H_1}$ (21 nodes, 20 edges) as shown in Fig. 1(a). Second, in order to account for relative motion of the tips of the fingers, we also propose $\mathcal{G}_{H_2}$ (21 nodes, 24 edges), which adds a set of new edges to $\mathcal{G}_{H_1}$ so that the fingertips are linked, as shown in Fig. 1(a). Finally, we note that both hands are involved in an assembling task, so that the relative motion between hands is also an important feature for activity understanding. Consequently, $\mathcal{G}_{H_3}$ (42 nodes, 46 edges) is constructed as shown in Fig. 2 adding a new set of edges between the two hands. $\mathcal{G}_{H_3}$ can capture the relative motion between two hands along with the intra-hand motion. All these graphs are undirected and unweighted.

Each graph is defined as $\mathcal{G} = [\mathcal{V}, \mathcal{E}]$, where $\mathcal{V}$ and $\mathcal{E}$ denote the set of vertices and edges respectively, with respective cardinalities N_v and N_e. We use the symmetric normalized graph Laplacian, defined as $\mathcal{L} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$, where $\boldsymbol{A}$ is the adjacency matrix, $\boldsymbol{D}$ is the degree matrix . The graph Fourier transform (GFT) is used to analyze the frequency content of graph signals. The spectral basis of the graph are the eigenvectors of $\mathcal{L}$ leading to a matrix $\boldsymbol{U}$ with columns $\{\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_{N_v}\}$. The corresponding spectral frequencies are the eigenvalues of $\mathcal{L}$ associated with $\boldsymbol{U}$ denoted by $\sigma(\mathcal{G}) = \lambda_1, \lambda_2, ..., \lambda_{N_v}$ where $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_{N_v}$.

We use the approach proposed in [9] to compute the GFT based features for our graph in each frame. The spectral basis $\boldsymbol{u}_k, k = 1, ..., N_v$ forms a basis for any graph signal residing on $\mathcal{G}$. That implies any graph signal can be represented as a unique linear combination of $\boldsymbol{u}_k$ as:

$$\boldsymbol{c_i} = \sum_{k=1}^{N_v} \alpha_{k,i} \boldsymbol{u}_k \tag{1}$$

$$\alpha_{k,i} = \boldsymbol{c}_i^{\top} \boldsymbol{u}_k \tag{2}$$

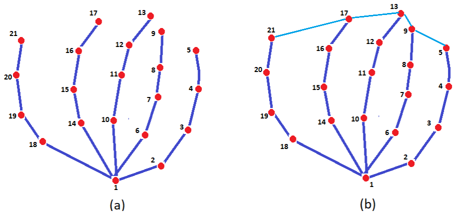

Fig. 1: Graph constructed for hand (a) $\mathcal{G}_{H_1}$ and (b) $\mathcal{G}_{H_2}$

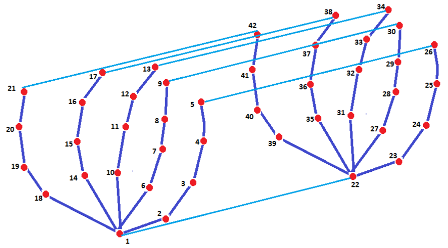

Fig. 2: Graph $\mathcal{G}_{H_3}$ constructed for both the hands

where, $\boldsymbol{c_i}$ is the motion vector present in each node (joint) of the graph (hand) and $\boldsymbol{\alpha_k}$ is a vector with length 2. Thus, $\alpha_{1,i}$, $\alpha_{2,i}$, ..., $\alpha_{N_v,i}$ can act as a unique representation for a given frame fr_j. We use these α's as features for activity segmentation. At any given time, we form a graph signal where to each node of the hand graph we associate a motion vector with the corresponding motion of that joint estimated by OpenPose.

2.1.2. Analysis of graph frequencies

Clearly, from Fig. 1(a), $\mathcal{G}_{H_1}$ is a tree structured graph, and therefore it is also bipartite. Thus, the eigenvectors of the normalized graph Laplacian are in the interval $[0, 2]$ with $\lambda_N = 2$ [20]. For $\mathcal{G}_{H_1}$, each alternative eigenvalue has multiplicity greater than 1. We observe that $\mathcal{G}_{H_1}$ is an extended version of a star graph, where each finger of the star has more than one node. A hand graph with N_1 fingers, and N_2 nodes per finger (here, $N_v = N_1 \times N_2 + 1$), has a special eigenstructure with the following properties.

- Every second eigenvalue has multiplicity $N_1 - 1$.
- The number of other eigenvalues which has multiplicity 1, is $N_2 + 1$.
- The minimum and maximum eigenvalues are 0 and 2 respectively with multiplicity 1.

Likewise, the N-star graph [21] has eigenvalue 1 with multiplicity $N - 2$, and other two eigenvalues are 0 and 2 respectively (with multiplicity 1). It is also similar to the above mentioned properties. N-star graph means it has $N - 1$ fingers, so $N_1 = N - 1$ here. So,

- The only eigenvalue (here, $\lambda = 1$) with higher multiplicity has multiplicity $N_1 - 1 = (N - 1) - 1 = N - 2$.
- The no. of eigenvalues with multiplicity 1 ($\lambda = 0$ and 2) is $N_2 + 1 = 2$, (here $N_2 = 1$ as each finger has only 1 node).

Note that $\mathcal{G}_{H_2}$ has a unique eigenstructure meaning distinct eigenvalues, but $\mathcal{G}_{H_3}$ has an eigenstructure that is similar $\mathcal{G}_{H_1}$. The spectral properties of these graphs are important, given that we are using the GFT as our feature vector. In particular, for eigenvalues with multiplicity greater than one there are multiple ways to project the input graph signal onto the corresponding subspace. In this paper we do not exploit this to improve performance and leave further optimization of this choice for future work. Moreover, all the three graphs follow the graph symmetric property which can potentially lead to fast algorithms for GFT computations, similar to those proposed in [22].

2.2. Segmentation

For online segmentation of time-series data we need to define a measure of similarity between two consecutive windows. For our segmentation task, we rely on the motion pattern changes from one action to another without using any prior knowledge about the activity.

2.2.1. Segmentation using BIC

In the proposed method, the distance between two consecutive windows is measured by Generalized likelihood ratio (GLR) [17]. At time t_i, let $\boldsymbol{W}_l$ and $\boldsymbol{W}_r$ be the feature matrix of the left and right window. Each column of W is constructed from the features computed using (2).

Determining whether a boundary exists at frame i is dependent on the relative performance of two competing models. The first model assumes that $\boldsymbol{w}_1, ..., \boldsymbol{w}_N \in \boldsymbol{W}_l \cup \boldsymbol{W}_r$ is more appropriately modeled by a single distribution ($\boldsymbol{W}_l \cup \boldsymbol{W}_r \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\boldsymbol{w}_i \in \mathbb{R}^d$, d is the dimension of the feature vector space. The second model assumes that $\boldsymbol{w}_1, ..., \boldsymbol{w}_N$ is more appropriately modeled by two separate distributions where $\boldsymbol{w}_1, ..., \boldsymbol{w}_i \in \boldsymbol{W}_l$ and $\boldsymbol{W}_l \sim \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)$; $\boldsymbol{w}_{i+1}, ..., \boldsymbol{w}_N \in \boldsymbol{W}_r$ and $\boldsymbol{W}_r \sim \mathcal{N}(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$. Then, ΔBIC_i is computed using (3).

$$\Delta BIC_i = \log(\frac{|\boldsymbol{\Sigma}_{\boldsymbol{W}_l \cup \boldsymbol{W}_r}|^{\frac{N}{2}}}{|\boldsymbol{\Sigma}_{\boldsymbol{W}_l}|^{\frac{N_l}{2}}|\boldsymbol{\Sigma}_{\boldsymbol{W}_r}|^{\frac{N_r}{2}}}) - \frac{\lambda}{2}(d + \frac{d(d+1)}{2}) \log N \tag{3}$$

where, $|.|$ is the determinant of a matrix, and (d, N_l), (d, N_r), (d, N) are the dimension of $\boldsymbol{W}_l, \boldsymbol{W}_r, \boldsymbol{W}_l \cup \boldsymbol{W}_r$, and $N = N_l + N_r$.

Now, if $\Delta BIC_i > 0$, then frame i is a good segmentation boundary, otherwise we merge $\boldsymbol{W}_l$ and $\boldsymbol{W}_r$ and compare the next window with this merged window. The first term in (3) is GLR when the model is Gaussian and the second term, $\frac{\lambda}{2}(d + \frac{d(d+1)}{2}) \log N$, is responsible for penalizing the candidate models according to their complexities. λ controls the number of segments.

3. EXPERIMENTAL SET UP AND DATASET

This section presents a detailed description of the experimental setup and the toy assembling task. Each subject is asked to assemble a gopigo3 [23] robot base kit according to a specific set of instructions. Fig. 3 shows a pictorial representation of the sequential subtasks of the toy assembling task. This task has three main sub-actions.

- **Action1. Assembling**: Attach the front wheel; Set the red board; Tighten the screws.
- **Action2. Combining**: Attach the power cable with the red board; Connect the sonic sensor cable to the red board; Combine the green board, use the pins; Attach the side wheels.
- **Action3. Checking**: Check for all the parts attached/ assembled properly or not.

Before starting the task, one instructor demonstrates all the steps clearly to each subject. Moreover, a pictorial representation of the sequential steps is available in front of them during the task. The parts of the toy car are kept on a table with a height of $105cm$. 11 subjects (9 Male, 2 Female) are asked to perform the task three times. All subjects are in their 20s and early 30s and they all are from an engineering background, thus accustomed to the type of tasks involved in robot car assembling. Each subject performs the assembling task three times, hence, we get 33 data sequences. A HD 1080p logitech camera is used to capture the scene. After capturing all the videos, we use OpenPose to extract the 2D hand skeleton key points for each subject. OpenPose estimates 2×21 - hand key-points in each frame at fps 30. The key point hand dataset is available online [18].

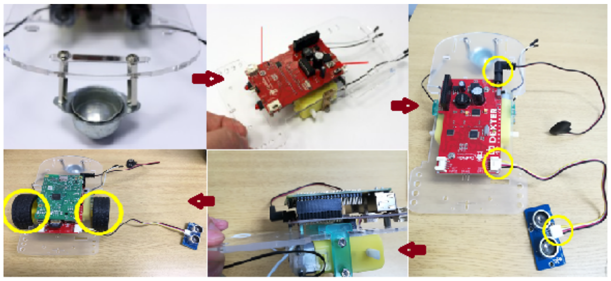

Fig. 3: Steps for the toy assembling task

4. RESULTS

Metric proposed by Gensler et al. in [24] is used for performance evaluation. In this assembling task in an industrial setting, we can tolerate a little early or late segmentation while performing the online unsupervised segmentation. Moreover, for online segmentation, in order to decide a segmentation point, we need to wait for the information from the current window and then process it. As a result, we can only have segmentation point at the start or end of the window. To account for such a scenario, a segmentation zone (SZ) is defined around each ground truth segmentation instance. The definition of True positive (TP), True negative (TN), False positive (FP), False negative (FN) is given below.

- TP: If segmentation zone only has one segmentation instance from the algorithmic output.
- TN: For any time point, it is not a true segmentation instance and algorithm also detected as same.
- FP: If a SZ has more than 1 segmentation instance from the algorithm, or if a time step which is not a segmentation point in the ground truth, but algorithm detected it as a segmentation instance.
- FN: If there is no segmentation point from the algorithm in a SZ.

Let $\mathcal{S}_g$ and $\mathcal{S}_a$ be the set of segmentation time points given by ground truth and algorithm, respectively. $\hat{\mathcal{S}_a}$ contains segmentation instances corresponding to TP, hence, $\hat{\mathcal{S}_a} \subset \mathcal{S}_a$. Let f_g and $\hat{f}_a$ be the action labels of frames segmented by $\mathcal{S}_g$ and $\hat{\mathcal{S}_a}$ respectively. Then, $f_{tp} = f_g \cap \hat{f}_a$ and cardinality of f_{tp} is L_{tp}. Letting the length of the sequence be L we have

$$SegAcc = \frac{L_{tp}}{L} \times 100\% \tag{4}$$

where a higher value for $SegAcc$ corresponds to better performance. To take into consideration the early and late segmentation, the distance between $\mathcal{S}_g$ and $\hat{\mathcal{S}_a}$ is measured using (5). Letting the cardinality of $\mathcal{S}_g$, $\hat{\mathcal{S}_a}$ and $\mathcal{S}_a$ be L_g , $\hat{L_a}$ and L_a respectively, we define

$$\begin{aligned} Score_1 &= (1 - \sum_{i=1}^{L_g} \beta_i \frac{|\mathcal{S}_{g_i} - \hat{\mathcal{S}}_{a_i}|}{L}) \times 100\% \\ Score_2 &= 100 - \delta \times |\hat{L_a} - L_a|\% \end{aligned} \tag{5}$$

where $\sum_{i=1}^{L_g} \beta_i = 1$ is weight vector and δ is a penalty factor. Higher value of $Score_1$ stands for segmentation instances closer to ground truth. Number of unwanted segmentation instances is also counted for qualitative analysis using $Score_2$.

An experiment is conducted for varying SZ from 5s to 10s. The minimum value of SZ is set to $5s$ as in our online segmentation system $WindowLength$ is also set to $5s$. Note that for increasing value of SZ, accuracy is increased, but with the increase in SZ, we are allowing more tolerance of early and late segmentation. As a compromise, for the rest of the paper we report results for $SZ = 7s$ and $WindowLength = 5s$. Due to lack of space a detailed study is not presented. For comparison, the same online segmentation is performed with the motion vector computed from the hand position data, instead of using the GFT coefficients as features, and this is considered as the baseline. The set of TPs corresponding to proposed method and baseline are saved in $\hat{\mathcal{S}_a^p}$ and $\hat{\mathcal{S}_a^b}$ respectively.

Fig. 4 and 5 show the segmentation achieved by the proposed method ($\hat{\mathcal{S}_a^p}$) and the baseline method ($\hat{\mathcal{S}_a^b}$), respectively, with $SegAcc$ for each participant. Color transitions represent action changes and the cross marks represent the ground truth segmentation points. Clearly, our proposed method detects segmentation points within SZ for most of the data sequences, while the baseline method mostly fails to do that. The average evaluation metrics for proposed method are $Precision = 54.3\%, Recall = 85.7\%, F1-Score = 64.1\%, Score_1 = 59.6\%$ and the average evaluation metrics for baseline method are $Precision = 25.1\%, Recall = 33.3\%, F1-Score = 22.2\%, Score_1 = 16.4\%$. The proposed method outperforms the baseline in terms of all the metrics .

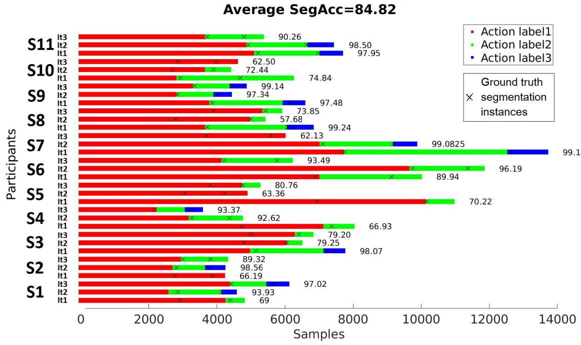

Fig. 4: Segmentation outcome($\hat{\mathcal{S}_a^p}$) using features from $\mathcal{G}_{H_3}$ ($\lambda = 1$) for the proposed method with $SegAcc = 84.8\%$. S_i and It_i represent subject ID and iteration number respectively.

Table 1 compares the three proposed hand graphs in terms of segmentation performance for different λ. For lower value of λ, $Score_2$ decreases but $Score_1$ increases which implies over-segmentation but segmentation closer to $\mathcal{S}_g$, and for higher λ, the opposite behavior is observed. At the same time, $Precision$ and $Recall$ value decrease with the increase of λ. So, $\lambda = 1$ can be chosen for better performance. It is evident in the table that the features extracted from $\mathcal{G}_{H_3}$ outperform $\mathcal{G}_{H_1}$ and $\mathcal{G}_{H_2}$ in terms of all metrics. This justifies our assumption that information of relative motion between hands is important and efficiently captured by $\mathcal{G}_{H_3}$. If there is a task where only one hand is involved, one can use $\mathcal{G}_{H_1}$ or $\mathcal{G}_{H_2}$ instead. The segmentation takes 0.0034s to process a window of $5s$ using Matlab 2017b running on a 8-Core Intel Xeon processor with 64GB RAM.

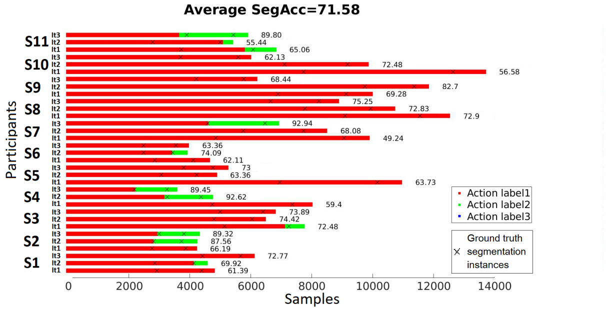

Fig. 5: Segmentation outcome($\hat{\mathcal{S}_a^b}$) using features from baseline method ($\lambda = 1$) with $SegAcc = 71.6\%$. S_i and It_i represent subject ID and iteration number respectively.

Table 1: Comparison between different graphs

in %	$\lambda = 0.8$			$\lambda = 1$			$\lambda = 1.5$		
$\mathcal{G}$	$\mathcal{G}_{H_1}$	$\mathcal{G}_{H_2}$	$\mathcal{G}_{H_3}$	$\mathcal{G}_{H_1}$	$\mathcal{G}_{H_2}$	$\mathcal{G}_{H_3}$	$\mathcal{G}_{H_1}$	$\mathcal{G}_{H_2}$	$\mathcal{G}_{H_3}$
M_1	45.3	42.1	**58.1**	37.3	39.8	**54.3**	10.6	16.7	**26.2**
M_2	78.7	72.7	**100**	66.7	66.7	**85.7**	21.2	33.3	**47.2**
M_3	54.8	51.1	**71.2**	46.7	48.3	**64.1**	14.1	22.2	**33.2**
M_4	86.6	84.3	**93.1**	78.9	79.3	**84.8**	69.2	71.58	**75.5**
M_5	61	56.7	**66.1**	41.6	43.1	**59.6**	10.4	16.3	**18.9**
M_6	66.3	69.7	**72.5**	71.2	73.1	**75.4**	91.5	92.1	**92.3**

$M_1, M_2, M_3, M_4, M_5, M_6$ represents $Precision$, $Recall$, $F1-Score$, $SegAcc$, $Score_1$ and $Score_2$, respectively (measured as %).

5. CONCLUSION

In this paper, we propose novel graph-based representations of hand *MoCap* data. These representations can efficiently capture the motion of a single hand, coordination between hands and can be used in understanding complex activities. A study of the spectral properties of these proposed graph structures is also presented. The efficiency of the proposed graphs is evaluated on a segmentation problem of an assembly task. The goal is to segment the videos of subjects, performing this task, according to the given sequence of the action primitives. The collected data is made available to research community to facilitate further development in this direction. The motion vector computed from the 2D hand skeleton data given by OpenPose is then used as graph signal in the proposed graphs to compute the GFT coefficients. A BIC based online unsupervised segmentation is performed using these GFT features. It is shown that these graph based features outperform the features based on the motion vector in this context, with one of the three proposed graphs performing consistently better in this action segmentation problem. Our proposed graph-based representations can be used in other hand *MoCap* tasks for which video is not available.

6. REFERENCES

[1] Tengda Han, Jue Wang, Anoop Cherian, and Stephen Gould, “Human action forecasting by learning task grammars,” *arXiv:1709.06391*, 2017.

[2] S. Stein and S. J. McKenna, “Combining embedded accelerometers with computer vision for recognizing food preparation activities,” in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2013), Zurich, Switzerland*. September 2013, ACM.

[3] N. Ahmidi, L. Tao, S. Sefati, Y. Gao, C. Lea, B. B. Haro, L. Zappella, S. Khudanpur, R. Vidal, and G. D. Hager, “A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery,” *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 9, pp. 2025–2041, Sept 2017.

[4] Ronald Poppe, “A survey on vision-based human action recognition,” *Image Vision Comput.*, vol. 28, no. 6, pp. 976–990, June 2010.

[5] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, “Machine recognition of human activities: A survey,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1473–1488, Nov 2008.

[6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh, “Realtime multi-person 2d pose estimation using part affinity fields,” in *CVPR*, 2017.

[7] Pei Wang, Chunfeng Yuan, Weiming Hu, Bing Li, and Yanning Zhang, “Graph based skeleton motion representation and similarity measurement for action recognition,” in *European Conference on Computer Vision*. Springer, 2016, pp. 370–385.

[8] V. O. Andersson and R. M. Araujo, “Full body person identification using the kinect sensor,” in *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, Nov 2014, pp. 627–633.

[9] Jiun-Yu Kao, Antonio Ortega, and Shrikanth Narayanan, “Graph-based approach for motion capture data representation and analysis,” in *Proceedings of IEEE International Conference on Image Processing*, oct 2014.

[10] Sunil K. Narang and Antonio Ortega, “Perfect reconstruction two-channel wavelet filter-banks for graph structured data,” *CoRR*, vol. abs/1106.3693, 2011.

[11] S. A. More and P. J. Deore, “Gait recognition by cross wavelet transform and graph model,” *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 3, pp. 718–726, May 2018.

[12] Tommi Kerola, Nakamasa Inoue, and Koichi Shinoda, “Spectral graph skeletons for 3d action recognition,” in *Asian Conference on Computer Vision*. Springer, 2014, pp. 417–432.

[13] Fadime Sener and Angela Yao, “Unsupervised learning and segmentation of complex activities from video,” *CoRR*, vol. abs/1803.09490, 2018.

[14] Ava Bargi, Richard Y. D. Xu, and Massimo Piccardi, “An online hdp-hmm for joint action segmentation and classification in motion capture data,” *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–7, 2012.

[15] I. Mumtaz, J. Lv, and J. Wei, “A novel method for online action segmentation and classification,” in *2015 5th International Conference on Information Science and Technology (ICIST)*, April 2015, pp. 569–573.

[16] Xueping Liu, Yibo Li, and Qing Shen, “Real-time action detection and temporal segmentation in continuous video,” *The Imaging Science Journal*, vol. 65, no. 7, pp. 418–427, 2017.

[17] Kyu J. Han, Panayiotis G. Georgiou, and Shrikanth Narayanan, “The SAIL Speaker Diarization System for Analysis of Spontaneous Meetings,” in *Proceedings of IEEE International Workshop on Multimedia Signal Processing (MMSP)*, Cairns, Australia, Oct. 2008.

[18] Pratyusha Das, Jiun-Yu Kao, Tomoya Sawada, Antonio Ortega, Hassan Mansour, Anthony Vetro, and Akira Minezawa, “Hand skeleton dataset for a toy assembling task extracted using openpose,” `https://drive.google.com/file/d/1DDmkSL-_CqZxdFgg83NOufXY-LFtDI0s/view?usp=sharing`, 2018.

[19] Sven Bambach, Stefan Lee, David J. Crandall, and Chen Yu, “Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions,” in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[20] Sunil K Narang and Antonio Ortega, “Perfect reconstruction two-channel wavelet filter banks for graph structured data,” *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2786–2799, 2012.

[21] Gregory Berkolaiko, EB Bogomolny, and JP Keating, “Star graphs and eba billiards,” *Journal of Physics A: Mathematical and General*, vol. 34, no. 3, pp. 335, 2001.

[22] Keng-Shih Lu and Antonio Ortega, “Fast implementation for symmetric non-separable transforms based on grids,” in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4109–4113.

[23] Dexter Industries, “Gopigo3 robot base kit,” `https://www.amazon.com/Dexter-Industries-GoPiGo3-Robot-Base/dp/B071WPZ2GF`.

[24] André Gensler and Bernhard Sick, “Novel criteria to measure performance of time series segmentation techniques.,” in *LWA*, 2014, pp. 193–204.