Body Motion Segmentation via Multilayer Graph Processing for Wearable Sensor Signals

Qinwen Deng, *Student Member, IEEE*, Songyang Zhang, *Member, IEEE*, and Zhi Ding, *Fellow, IEEE*

***Abstract*—Human body motion segmentation plays a major role in many applications, ranging from computer vision to robotics. Among a variety of algorithms, graph-based approaches have demonstrated exciting potential in motion analysis owing to their power to capture the underlying correlations among joints. However, most existing works focus on the simpler single-layer geometric structures, whereas multi-layer spatial-temporal graph structure can provide more informative results. To provide an interpretable analysis on multilayer spatial-temporal structures, we revisit the emerging field of multilayer graph signal processing (M-GSP), and propose novel approaches based on M-GSP to human motion segmentation. Specifically, we model the spatial-temporal relationships via multilayer graphs (MLG) and introduce M-GSP spectrum analysis for feature extraction. We present two different M-GSP based algorithms for unsupervised segmentation in the MLG spectrum and vertex domains, respectively. Our experimental results demonstrate the robustness and effectiveness of our proposed methods.**

***Index Terms*—multilayer graph signal processing, motion segmentation, unsupervised learning.**

I. Introduction

HUMAN motion analysis has been an important tool and an active research field, stemming from its broad applications in many areas, ranging from human-robot interaction to autonomous driving [1]–[3]. Among a variety of tasks, human motion segmentation serves as an important analytical step, benefiting a wide range of motion/action-related tasks, such as gesture recognition, human activity recognition, and human gait analysis [4]. Generally, human motion segmentation aims to divide a long sequence of motion frames into several short, non-overlapping temporal sections [5], each of which has its distinct physical meaning, as shown in Fig. 1(a). Specifically, we aim to process motion skeleton data extracted from video or synthesized from multiple sensors, instead of the raw video footage. Thus, the question of how to cluster the motion video/sequence into meaningful clips has a vital role in human motion analysis.

Despite many works focusing on temporally segmenting videos about motions [6]–[8], existing motion segmentation methods within the scope of processing motion sequences can be categorized into either unsupervised segmentation or supervised classification. Unsupervised motion segmentation usually utilizes the temporal dependency in the video sequence, including that in accelerometer and gyroscope data, to segment different motions. Typical clustering methods include low-rank transfer clustering [5], hierarchical aligned cluster analysis (HACA) [9], and subspace clustering [10]. However, these clustering-based approaches usually stress temporal information and require some external information, such as the exact number of different actions, which may not be available within the datasets themselves. On the other hand, supervised approaches usually assume pre-labeled datasets to train the deep learning networks. For example, in [11], a deep neural network is proposed for human action classification. Other methods also include long short-term memory networks (LSTM) [12] and few-shot learning [17]. Despite some notable successes, most learning based methods need training sets that are often labeled by human supervisors, which may be inaccurate and unavailable in real applications, thereby limiting the practicability of supervised learning. The development of a more efficient method for motion segmentation remains an open direction of endeavor.

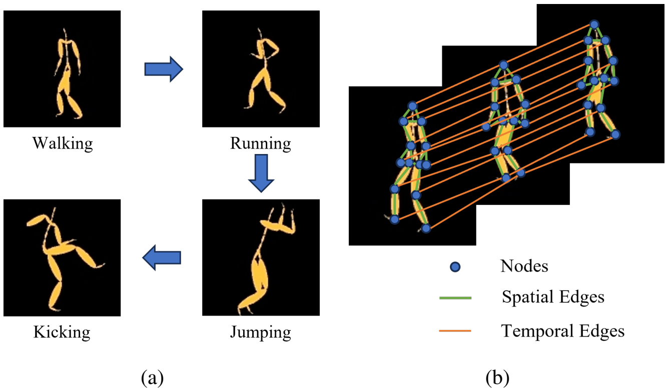


Fig. 1. Illustration of Human Motion: (a) Example of Motion Segmentation; (b) Spatial-Temporal Relationships Modeled By Multilayer Graph.

Recent development of geometric approaches, including graph signal processing (GSP) [18] and graph neural networks (GNN) [19], [20], has provided another promising alternative in human motion segmentation, for both supervised and unsupervised scenarios. In [21], a spatial temporal graph convolutional networks (ST-GCN) is introduced for skeleton-based action recognition. Extended from ST-GCN, the authors of [22] proposed a multi-stage spatial-temporal graph convolutional neural network (MS-GCN). Another work [28] introduced spatio-temporal graph cuts for event-based motion segmentation. However, most existing works assume

This material is based upon work supported by the 2029848.

Q. Deng and Z. Ding are with Department of Electrical and Computer Engineering, University of California, Davis, CA, 95616. (E-mail: mrdeng@ucdavis.edu, and zding@ucdavis.edu).

S. Zhang is with Department of Electrical and Computer Engineering, University of Louisiana at Lafayette, Lafayette, LA, 70504 (E-mail: songyang.zhang@louisiana.edu).

that human motions would fit a homogeneous spatial graph structure whereas, in fact, an alternative multilayer heterogeneous structure could be more informative. As shown in Fig. 1(b), joints in each temporal framework might have different underlying geometric structures due to the motion dynamics, suitable for a multilayer graph (MLG) structure. Moreover, limited by the homogeneous spatial structure, existing works are inefficient in processing the inter-layer (temporal) and intra-layer (spatial) correlations jointly, by conducting separate spatial and temporal analysis. How to jointly extract spatial-temporal geometric features remains a critical challenge. Fortunately, within the context of GSP, a multilayer graph signal processing (M-GSP) framework has been introduced for MLG based on tensor representation [30]. Different from the more traditional multiway GSP (MWGSP) [31], M-GSP allows different spatial layers to represent heterogeneous geometric structures, and defines a joint MLG spectral space for data analysis. M-GSP has exhibited some strong potentials in image compression [32] and hyperspectral image segmentation [33].

To capture the heterogeneous underlying geometry and address the spatial-temporal relationships jointly, we apply M-GSP and propose two novel MLG-based methods for unsupervised human motion segmentation. Compared with deep-learning based methods, our proposed MLG-based unsupervised algorithm is capable of addressing limited labeled data without requiring fine-tuning, leading to substantial reductions in labeling efforts for practical applications. More specifically, we first introduce the MLG modeling for human motion datasets and define a MLG singular space for motion analysis. We then investigate the M-GSP spectral properties and design a M-GSP Haar-like highpass filter for feature extraction, based on which spectral segmentation is implemented. To reduce complexity and enhance efficiency, we present another M-GSP based approach according to the tensor representation of MLG. Our experimental results demonstrate the power of M-GSP in extracting spatial-temporal features, as well as the efficacy of the proposed methods. We summarize our contributions as follows:

- To characterize spatial-temporal geometric correlations in human motion sequences, we introduce an MLG model, together with its tensor representation, for motion segmentation. To our best knowledge, we are the first to propose M-GSP/GSP for human motion analysis.
- To reveal geometric features of human motions, we propose an M-GSP spectral method for unsupervised motion segmentation, and to investigate the properties in the MLG singular space.
- Beyond spectral analysis, we develop an M-GSP based motion segmentation method in the vertex domain by exploring tensorial and structural features of MLG.
- Using our suggested guidelines for parameter selection, our experimental results demonstrate the efficacy of the proposed methods in both unsupervised and supervised testing setups.

We organize the rest of the paper as follows. In Section II, we first review related works on motion segmentation of motion capture data. We then provide fundamentals of M-GSP with respect to the preliminaries and notations in Section III. Next, we present the MLG models for human motion datasets in Section IV, with which we propose two novel unsupervised motion segmentation algorithms in spectrum domain and vertex domain, respectively, in Section V. We present experimental results of the proposed methods in both supervised and unsupervised setup in Section VI, before summarizing our work in Section VII.

II. Related Works

In this section, we first briefly review prior works on motion segmentation. Generally, existing motion segmentation solutions belong to either unsupervised motion clustering or supervised motion recognition.

A. Unsupervised motion clustering

The unsupervised motion clustering usually exploits global information of a motion sequence, and divides the sequence into several meaningful sections [36]. In traditional unsupervised setup, conventional clustering algorithms, such as K-means clustering [34] and spectral clustering (SC) [35], can be applied for human motion segmentation. However, these traditional clustering algorithms are often inefficient to capture geometric information in human motions. For example, the efficacy of some conventional clustering algorithms, such as K-means clustering, are constrained by the fact that they are only optimal for spherical clusters, making them unsuitable for capturing distinctive distances of sections in motion sequences [9]. Furthermore, even for the same motion, the lengths of segments in human motion vary due to inconsistency in movement speeds, leading to difficulties for these basic clustering algorithms. Extending traditional clustering, an aligned cluster analysis (ACA) together with its extension hierarchical ACA (HACA) is introduced in [9] as a generalization of kernel k-means (KKM) and SC for time series clustering and embedding. The ACA algorithm combines dynamic time alignment kernel (DTAK) with KKM and SC to better capture features of segments with different lengths. Leveraging ACA, the HACA provides a hierarchical structure at different temporal scales to refine temporal segmentation results while reducing computational complexity. Other typical algorithms also include low-rank transfer clustering [5], transfer subspace clustering, kernel subspace clustering [10] and auto-encoder [37].

In addition, most existing clustering algorithms focus on temporal dynamics while ignoring joint spatial-temporal information that can be more informative in realistic scenarios. These algorithms simply treat all data within the same time frame as a vector of features. However, in realistic scenarios, changes of spatial connections within a single frame, along with the consideration of joint spatial-temporal connections can provide more informative insights. As shown in Fig. 1, a multilayer graph (MLG) built on the motion sequence can naturally capture spatial-temporal connections. Also, the M-GSP framework proposed in [30] has shown its ability to capture and process joint spatial-temporal information. Therefore, in this work, we investigate the application of

M-GSP on motion clustering by introducing an MLG-based clustering algorithm. This algorithm characterizes both spatial and temporal correlations and is capable of efficient joint spatial-temporal processing.

B. Supervised Motion Recognition

Supervised motion recognition usually assumes a given prior-labeled dataset to train the neural networks. For example, a deep neural network called SE3-NETS was proposed in [11] to segment point clouds into distinct objects and jointly predict their rigid body motion. Another typical type of learning framework is temporal convolutional networks (TCN). Based on the framework of TCN, a multi-stage temporal convolutional networks (MS-TCN) was proposed in [50]. By stacking multiple stages sequentially, MS-TCN can process all temporal resolutions of videos to achieve better results. However, the predictions of each stage in MS-TCN tend to have over-segmentation errors. To address this issue, MS-TCN++ was proposed in [13] by introducing a dual dilated layer, which combines both large and small receptive fields. Additionally, the authors in [14] proposed a new cascading paradigm and a smoothing operation to enhance the adaptability and improve prediction confidence of the model for ambiguous frames. Another approach called efficient two-step network (ETSN) was introduced in [15] by using local burr suppression (LBS) to significantly reduce the over-segmentation errors. To further improve the performance, [16] presented a hierarchical action segmentation refiner (HASR), which can be plugged into MS-TCN model to refine the segment labels by referring to the entire video.

Recently, graph neural networks have attracted significant attention in motion segmentation. A spatial temporal graph convolutional network (ST-GCN) has been introduced in [21] for skeleton-based action recognition. Later, the authors in [23] extended ST-GCN by introducing the stacked hourglass architecture to improve the accuracy. Meanwhile, a decoupling GCN model was proposed in [24]. Similar to the decoupling aggregation mechanism in CNNs, this decoupling GCN model can improve the graph modeling ability without additional cost. Another graph convolutional network called central difference graph convolution (CDGC) was proposed in [25] by considering aggregating both node and gradient information in the learning model. Despite the successes, their graph modeling are normally limited by physical adjacency of the elements. To address this issue, the authors in [26] introduced two separate GCN models for spatial and temporal information modeling. To further improve the performance, the authors of [22] combined temporal convolutional neural network (TCN) with ST-GCN blocks to build a multi-stage spatial-temporal graph convolutional neural network (MS-GCN), which can lead to better segmentation. In addition, the authors of [17] added the connectionist temporal classification (CTC) into MS-GCN to improve temporal alignment between network predictions and ground truth. In [48], a local self-expression subspace learning network was proposed, where local self-expression layers maintain the representation relations between temporally adjacent motion frames. Besides, an end-to-end involving distinguished temporal graph convolutional networks called IDT-GCN was introduced in [27], where an involving distinction graph convolutional model and temporal segment regression module could enhance the spatial and temporal modeling capacity, respectively. However, most learning-based methods require training datasets that are often labeled by human supervisors, which are often unavailable and/or inaccurate in many practical applications. Our proposed MLG-based unsupervised algorithm is designed with easy-tuning with a minimal amount of labeled data, offering significant savings in labeling efforts for real-world applications.

III. Preliminaries

In this section, we provide the preliminaries and notations of M-GSP.

A. Multilayer Graph Signal Processing

1) Overview of GSP: Graph signal processing (GSP) has recently emerged as an important tool for structural data analysis [18] due to its power in capturing underlying data correlations. Representing the geometric relationships by graph models, a spectral space called graph Fourier space can be defined for data analysis [38], which shows great potentials in massive applications, including point cloud analysis [39], signal denoising [40] and data resampling [41]. To leverage the power of GSP in higher order correlations, high-dimensional GSP, such as hypergraph signal processing [42], [43] and topological signal processing [44], are developed. Among these high-dimensional GSP, multilayer graph signal processing (M-GSP) represents an efficient low complexity tool for spatial-temporal signal analysis [30], [33].

2) Preliminaries of M-GSP: M-GSP is a tensor-based framework for MLG analysis [30]. A multilayer graph with M layers and N nodes in each layer can be viewed as projecting N virtual entities $\{u_1, \cdots, u_N\}$ into M layers $\{\ell_1, \cdots, \ell_M\}$, such as spectrum band frames for hyperspectral images and color frames for RGB images. In skeleton-based human motion dataset, each joint can be viewed as an entity u_i and each temporal frame serves layer ℓ_α. Then, a motion sequence as shown in Fig. 1(b) can be modeled as an MLG with the same number of nodes in each layer. In M-GSP, such an MLG structure can be represented by a 4-th order adjacency tensor defined as follows:

$$\mathbf{A} = (A_{\alpha i \beta j}) \in \mathbb{R}^{M \times N \times M \times N}, \tag{1}$$

where $1 \leq \alpha, \beta \leq M, 1 \leq i, j \leq N$.

Here, each entry $A_{\alpha i \beta j}$ of adjacency tensor $\mathbf{A}$ indicates the edge strength between entity j's projected node in layer β and entity i's corresponding node in layer α. Note that tensor representation requires each layer to have the same number of nodes. We can generate such an MLG by:

- Adding isolated nodes to layers with fewer nodes to reach N nodes and set the interpolated signals as zeros; Since isolated node does not connect to any other nodes, they would not affect the message passing in MLG. This method is suitable for physical networks, such as smart grid and cyber-physical systems [45].

- Aggregating similar nodes as supernodes to reduce the node number to N: This can be an intuitive method for image processing, where several pixels can be grouped into superpixels.

Similar to traditional GSP, we define the MLG Fourier space via tensor decomposition. In an undirected MLG, the adjacency $\mathbf{A}$ is partially symmetric between orders one and three, and between orders two and four, respectively. Then, it can be approximated via orthogonal CANDECOMP/PARAFAC (CP) decomposition [30] as

$$\mathbf{A} \approx \sum_{\alpha=1}^{M}\sum_{i=1}^{N} \lambda_{\alpha i} \cdot \mathbf{f}_\alpha \circ \mathbf{e}_i \circ \mathbf{f}_\alpha \circ \mathbf{e}_i, \tag{2}$$

where $\circ$ is the tensor outer product [30], $\mathbf{f}_\alpha \in \mathbb{R}^M$ and $\mathbf{e}_i \in \mathbb{R}^N$ are orthonormal bases characterizing the properties of layers and entities, respectively.

Besides MLG Fourier space, the singular space is defined from HOSVD [46] as an alternative subspace of MLG, i.e.,

$$\mathbf{A} = \mathbf{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \times_4 \mathbf{U}^{(4)}. \tag{3}$$

Here, $\times_n$ denotes the n-mode product introduced in [30], which can be used to modify the dimension of the n-th order. $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times I_n}$ is a unitary matrix with $I_1 = I_3 = M$ and $I_2 = I_4 = N$ [33]. Similar to MLG Fourier space, there are two modes of singular spectra, i.e., $(\gamma_\alpha, \mathbf{f}_\alpha)$ for mode $1, 3$, and $(\sigma_i, \mathbf{e}_i)$ for mode $2, 4$. More specifically, $\mathbf{U}^{(1)} = \mathbf{U}^{(3)} = (\mathbf{f}_\alpha)$ and $\mathbf{U}^{(2)} = \mathbf{U}^{(4)} = (\mathbf{e}_i)$. Both singular tensor analysis and spectral analysis are efficient tools for image processing depending on specific tasks. In this work, we explore MLG singular analysis in human motion.

We now introduce the M-GSP singular transform (M-GST). Suppose that the singular vectors form $\mathbf{W}_f = [\mathbf{f}_1 \cdots \mathbf{f}_M] \in \mathbb{R}^{M \times M}$ and $\mathbf{W}_e = [\mathbf{e}_1 \cdots \mathbf{e}_N] \in \mathbb{R}^{N \times N}$. Given an MLG signal $\mathbf{s} = (s_{\alpha i}) \in \mathbb{R}^{M \times N}$, the layer-wise M-GST can be defined as

$$\check{\mathbf{s}}_L = \mathbf{W}_f^{\mathrm{T}} \mathbf{s} \in \mathbb{R}^{M \times N}, \tag{4}$$

and the entity-wise M-GST can be defined as

$$\check{\mathbf{s}}_N = \mathbf{s} \mathbf{W}_e \in \mathbb{R}^{M \times N}. \tag{5}$$

The joint M-GST can be calculated by

$$\check{\mathbf{s}} = \mathbf{W}_f^{\mathrm{T}} \mathbf{s} \mathbf{W}_e \in \mathbb{R}^{M \times N}. \tag{6}$$

For brevity, here we only present the basic concepts of M-GSP. Interested readers could refer to [30] for more details, including M-GSP spectral transform, filter design and spectral analysis.

IV. Problem Description and Modeling

We now introduce our problem description, together with the MLG models of the human motion sequence.

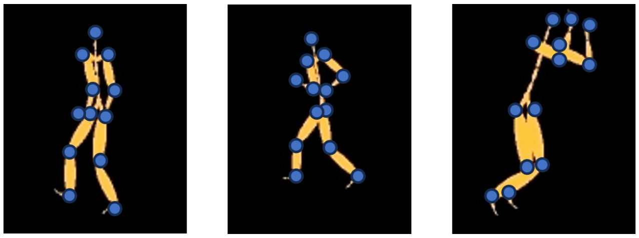


Fig. 2. Example of Skeleton-based Human Motion Dataset in CMU graphics lab motion capture database.

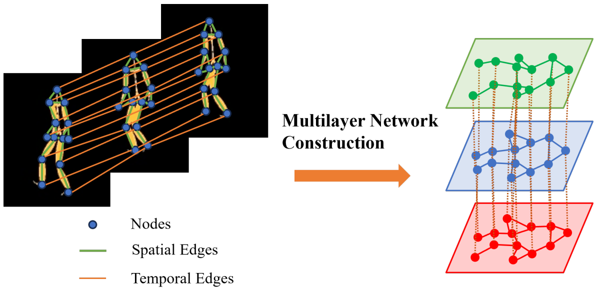


Fig. 3. Example of MLG model for one motion sequence.

A. Problem Description

Similar to [21], we focus on the skeleton-based human motion segmentation for wearable sensors. Such skeleton-based motion dataset can be collected by body sensors or reconstructed from videos [47]. As shown in Fig. 2, a human body in motion within the skeleton dataset is abstracted into N joints, which can be annotated with additional information, including three-dimensional (3D) coordinates, accelerometer data and gyroscope data. Suppose that the human motion sequence contains M frames. Then, the annotated signals of the joint i in layer α is defined by a feature vector $\mathbf{x}_{\alpha i}$. Our goal in this work is to segment the N temporal frames into several successive sections, which could capture the features of human behavior and match the realistic human motion. Instead of training on prior-labeled data samples, we focus on the unsupervised human motion segmentation.

B. Multilayer Graph Construction

Next, we introduce the MLG construction for human motion sequence. For most skeleton-based motion data captured by sensors, the number of data points within one frame is constant for one motion sequence. Thus, we can apply an MLG with the same number of nodes on each layer to model one such motion sequence shown as Fig. 3. Suppose that the motion sequence contain M temporal frames and N joints in each frame. Intuitively, such multilayer spatial-temporal structure can be viewed as projecting N entities into M layers, where the entity is defined by the spatial joints and the layer is defined by temporal frames. Note that, as we mentioned in Section III-A2, if the motion data sequence contains unequal numbers of joints across layers, we can add some dummy nodes in the MLG to keep the same number of nodes across different layers. These dummy nodes are isolated to all other nodes, and would not change the topological structure of the original multilayer architecture.

With such definition, any skeleton-based motion sequence can be intuitively modeled by an MLG, which can be represented by the forth-order adjacency tensor

$$\mathbf{A} = (A_{\alpha i \beta j}) \in \mathbb{R}^{M \times N \times M \times N}, \tag{7}$$

where $1 \leq \alpha, \beta \leq M, 1 \leq i, j \leq N$.

Here, $\alpha, \beta \in [1, M]$ are the indices of layer in MLG, while $i, j \in [1, N]$ are the indices of joints in each layer. Each entry in the adjacency tensor, i.e., $A_{\alpha i \beta j}$, represents the relationship between the i-th joint in α-th temporal layer and the j-th joint in β-th temporal layer. Now, we need to define the weights of $A_{\alpha i \beta j}$ to capture the geometric similarity among different joints. One common choice is to set the value of the element $A_{\alpha i \beta j}$ by Gaussian kernel [18], i.e.,

$$A_{\alpha i \beta j} = \exp\left(-\frac{\|\mathbf{x}_{\alpha i} - \mathbf{x}_{\beta j}\|^2}{\sigma^2}\right), \tag{8}$$

where $\mathbf{x_{\alpha i}}$ and $\mathbf{x_{\beta j}}$ represent data vectors, such as the coordinates, of the i-th joint in α-th temporal layer and the j-th node in β-th temporal layer, respectively. Also, the standard deviation σ controls the support of the kernel function.

Considering the different natures in the interlayer (inter-temporal) and intralayer (spatial) connections, we apply different σ. i.e., $\sigma = \sigma_s$ when calculating the (spatial) similarity between two nodes within the same layer with $\alpha = \beta$, while using $\sigma = \sigma_t$ when calculating the (temporal) similarity between two nodes in different layers with $\alpha \neq \beta$. More specially, the value of σ_s and σ_t should be related to the statistics of all spatial and temporal distances. Thus, we apply the average of all distances as the value in this work, i.e.,

$$\sigma_s = \frac{1}{N_s} \sum_{i,j \in [1,N], \alpha = \beta} \|\mathbf{x_{\alpha i}} - \mathbf{x_{\beta j}}\|, \tag{9}$$

and

$$\sigma_t = \frac{1}{N_t} \sum_{\alpha \in [1,M-1], \beta = \alpha+1, i=j} \|\mathbf{x_{\alpha i}} - \mathbf{x_{\beta j}}\|, \tag{10}$$

where N_s is the total number of point pairs that are on the same frame, N_t is the total number of point pairs that are on the successive frames with the same index.

To highlight the interlayer correlations from the same joint, we utilize the multiplex structure in which each node only connects to its counterparts in its successive layers with $i = j$ and $|\alpha - \beta| = 1$. This structure further simplifies the interlayer geometric models. Then, the final weight of each entry $A_{\alpha i \beta j}$ shall be calculated as

$$A_{\alpha i \beta j} = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_{\alpha i} - \mathbf{x}_{\beta j}\|^2}{\sigma_s^2}\right) & \text{if } \alpha = \beta \\ \exp\left(-\frac{\|\mathbf{x}_{\alpha i} - \mathbf{x}_{\beta j}\|^2}{\sigma_t^2}\right) & \text{if } i = j \text{ and } |\alpha - \beta| = 1 \cdot \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

With the calculated adjacency tensor $\mathbf{A}$, tensor decomposition can be applied via Eq. (2) or Eq. (3) to obtain the MLG Fourier space or singular space for data analysis. Since the HOSVD is faster and more robust in comparison with CP decomposition, we calculate the singular space of the undirected multilayer graph via HOSVD for spectral feature extraction. More details regarding M-GSP singular analysis shall be discussed later in Section V-A.

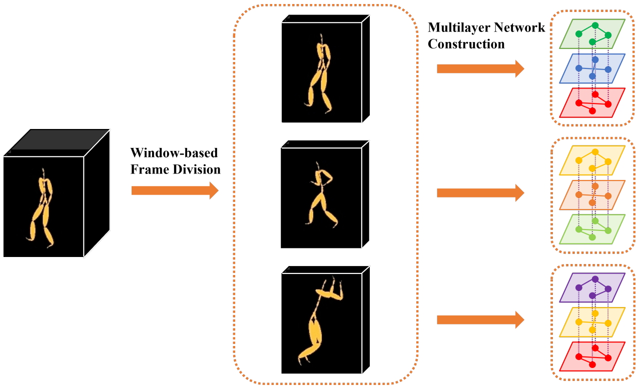


Fig. 4. Example of window cuts in motion sequence and the multilayer network construction.

V. M-GSP Body Motion Segmentation

We now introduce our MLG-based body motion segmentation. Since many body motion sequences in most dataset have thousands of frames with dozens of data points in each frame, it is unpractical, in terms of memory and computational time, to build the MLG and decompose the MLG for the entire sequence. To solve this problem, we focus on a short-time processing method. As shown in Fig. 4, we first cut the entire motion sequence into N_{seg} shorter segments with window length W_d for temporal frames. Successive segments may overlap with one other to give a smooth representation of the current motion. For each segment, we build an MLG using the 3D coordinates or other annotated information by using the model introduced in Section IV-B. We then extract features from the MLG as representation of the corresponding motion. In this work, we have two different MLG segmentation approaches: 1) spectrum-based MLG motion segmentation based on spectral signals; and 2) vertex-based MLG motion segmentation based on structure signals.

A. Spectrum-based MLG Motion Segmentation

As shown in Fig. 1(a), the body motion sequence consists of some major movements, which can be captured by the low-frequency components, and detailed joint actions, which can be captured by the high-frequency components. To highlight the detailed behavior of body motions, such as leg movement and hand waving, we apply an M-GSP filter for feature extraction. To locate high-frequency components in M-GSP singular domain, we first estimate the singular space of each segment using HOSVD in Eq. (3). We then transform the signal into the singular space using these spectrum bases. Given the features of motion sequences, we denote the features of the joint i in layer α as $\mathbf{x}_{\alpha i} \in \mathbb{R}^K$, where K is the feature dimension of each joint in a given temporal frame, including angles, coordinates and other available features. Suppose that

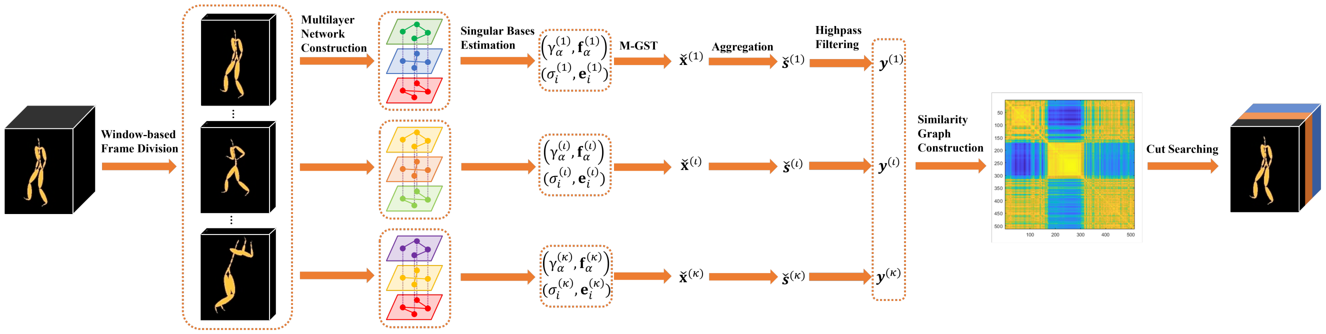

Window-based Frame Division
Multilayer Network Construction
Singular Bases Estimation
$(\gamma_\alpha^{(1)}, \mathbf{f}_\alpha^{(1)})$
$(\sigma_i^{(1)}, \mathbf{e}_i^{(1)})$
$(\gamma_\alpha^{(\iota)}, \mathbf{f}_\alpha^{(\iota)})$
$(\sigma_i^{(\iota)}, \mathbf{e}_i^{(\iota)})$
$(\gamma_\alpha^{(\kappa)}, \mathbf{f}_\alpha^{(\kappa)})$
$(\sigma_i^{(\kappa)}, \mathbf{e}_i^{(\kappa)})$
M-GST
$\check{\mathbf{x}}^{(1)}$
$\check{\mathbf{x}}^{(\iota)}$
$\check{\mathbf{x}}^{(\kappa)}$
Aggregation
$\check{\mathbf{s}}^{(1)}$
$\check{\mathbf{s}}^{(\iota)}$
$\check{\mathbf{s}}^{(\kappa)}$
Highpass Filtering
$\mathbf{y}^{(1)}$
$\mathbf{y}^{(\iota)}$
$\mathbf{y}^{(\kappa)}$
Similarity Graph Construction
Cut Searching

Fig. 5. Block Diagram of Spectrum-based MLG Motion Segmentation.

$\mathbf{x}_{\alpha i}[a]$ is the a-th annotated feature of the joint i in layer α. The whole signals for the a-th feature is represented by

$$\mathbf{x}[a] = (\mathbf{x}_{\alpha i}[a]) \in \mathbb{R}^{M\times N} \tag{12}$$

According to Eq. (6), the joint M-GST of the a-th feature signal can be calculated by

$$\check{\mathbf{x}}[a] = \mathbf{W}_f^{\mathrm{T}}\mathbf{x}[a]\mathbf{W}_e \in \mathbb{R}^{M\times N}. \tag{13}$$

Next, we aggregate all features into one signal $\check{\mathbf{s}} = (\check{s}_{\alpha i}) \in \mathbb{R}^{M\times N}$, where each entry is calculated as

$$\check{s}_{\alpha i} = ||[\check{\mathbf{x}}_{\alpha i}[1]], \cdots, \check{\mathbf{x}}_{\alpha i}[K]||_2^2, \tag{14}$$

where K is the dimension of features.

For the aggregated signals, an M-GSP highpass filter can be designed to extract the details in body motions. In this work, we combine two different kinds of highpass filters: ideal highpass filter and Haar-like highpass filter. Given the singular domain signal $\check{\mathbf{s}} \in \mathbb{R}^{M\times N}$, we first flatten it into a vector, and keep top k elements in the high frequency part, i.e., $\check{\mathbf{s}}' = [\check{s}_1, \cdots, \check{s}_k, 0, \cdots, 0] \in \mathbb{R}^{MN}$.

Thereafter, we use a Haar-like highpass filter V_{Haar} to process $\check{\mathbf{s}}'$. Suppose that $\lambda_{\alpha i} = \gamma_\alpha \sigma_i$. The filter V_{Haar} is defined as

$$\begin{aligned} V_{Haar} &= \mathbf{I} - diag(\boldsymbol{\lambda}) \qquad (15)\\ &= \begin{bmatrix} 1-\lambda_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1-\lambda_{MN} \end{bmatrix}. \qquad (16) \end{aligned}$$

The filtered signal is calculated as

$$\check{\mathbf{y}} = V_{Haar}\check{\mathbf{s}}' \in \mathbb{R}^{MN}. \tag{17}$$

Finally, we reshape $\check{\mathbf{y}}$ into $\check{\mathbf{s}}'' \in \mathbb{R}^{M\times N}$ and implement the inverse M-GST to get the vertex domain signal $\mathbf{y}$, which can be expressed as

$$\mathbf{y} = \mathbf{W}_f \check{\mathbf{s}} \mathbf{W}_e^{\mathrm{T}} \in \mathbb{R}^{M\times N}. \tag{18}$$

Upon obtaining the vertex domain signal for all N_{seg} segments in one body motion sequence, we calculate a similarity graph $\mathbf{A}_{sim} \in \mathbb{R}^{N_{seg}\times N_{seg}}$, whose elements are given by

$$\mathbf{A}_{sim}[m, n] = e^{-||\mathbf{y}^{(m)} - \mathbf{y}^{(n)}||_2^2}, \tag{19}$$

where $\mathbf{y}^{(m)}$ and $\mathbf{y}^{(n)}$ are the vertex domain signal for m-th and n-th segment in the motion sequence, respectively. We then convert the similarity graph $\mathbf{A}_{sim}$ into a sparse self similarity matrix (SSSM) $\mathcal{M} \in \mathbb{R}^{N_{seg}\times N_{seg}}$ by finding the peak values in $\mathbf{A}_{sim}$ in each row. The threshold for finding the peaks is set to top 3% largest value among all elements in $\mathbf{A}_{sim}$. To keep the $\mathcal{M}$ symmetric, once the element $\mathbf{A}_{sim}[m, n]$ is selected in each row as the peak values, the element $\mathbf{A}_{sim}[n, m]$ will also be set to the same value. All these peaks and its symmetrical elements are considered as the nearest neighbors.

To find the cutting frame based on the SSSM $\mathcal{M}$, we use a similar region growing search technique introduced in [36]. The search technique contains two steps: forward step and backward step. The forward step starts from the upper left corner of $\mathcal{M}$ and attempts to extend the connected region to the next row, while the backward step starts from the lower right corner of $\mathcal{M}$ and tries to extend the connected region to the previous row. In the forward step, a connected region starts as a seed denoted by $\mathcal{M}[1, 1]$. The region is extended to the next rows as long as the nearest neighbors in the updated region increases. Similarly, in the backward step, a connected region starts as a seed i.e., $\mathcal{M}[N_{seg}, N_{seg}]$, and the region is extended to the previous rows as long as the nearest neighbors in the updated region increases. If no new neighbors are found between segment i and $i + \omega$ in the larger region, except for the neighbors from the main diagonal of $\mathcal{M}$, then the current region search is considered complete. The parameter ω is set to 8 in our experiments. The major steps of spectrum-based MLG Motion Segmentation (SMLGS) is presented in Algorithm 1.

B. Vertex-based MLG Segmentation

To reduce complexity, another set of features to consider is the multilayer graph structure signal. To extract structural features in the vertex domain, we first reshape the adjacency tensor $\mathbf{A} \in \mathbb{R}^{M\times N\times M\times N}$ into a feature vector after constructing the MLG for each overlapping shorter segment with length W_d. We denote the feature vector for the m-th segment as $\mathbf{z}^{(m)} \in \mathbb{R}^{M^2N^2}$. Once we get $\mathbf{z}^{(m)}$ for all N_{seg} segments, we concatenate all $\mathbf{z}^{(m)}$ into one matrix

$$\mathbf{z} = [\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \cdots, \mathbf{z}^{(N_{seg})}] \in \mathbb{R}^{M^2N^2\times N_{seg}}. \tag{20}$$

We subsequently reduce the size of $\mathbf{z}$ by keeping top k elements with highest variance across all segments as follows:

- We first calculate the variance for each row in $\mathbf{z}$, and concatenate them into $\sigma^2(\mathbf{z}) \in \mathbb{R}^{M^2N^2}$

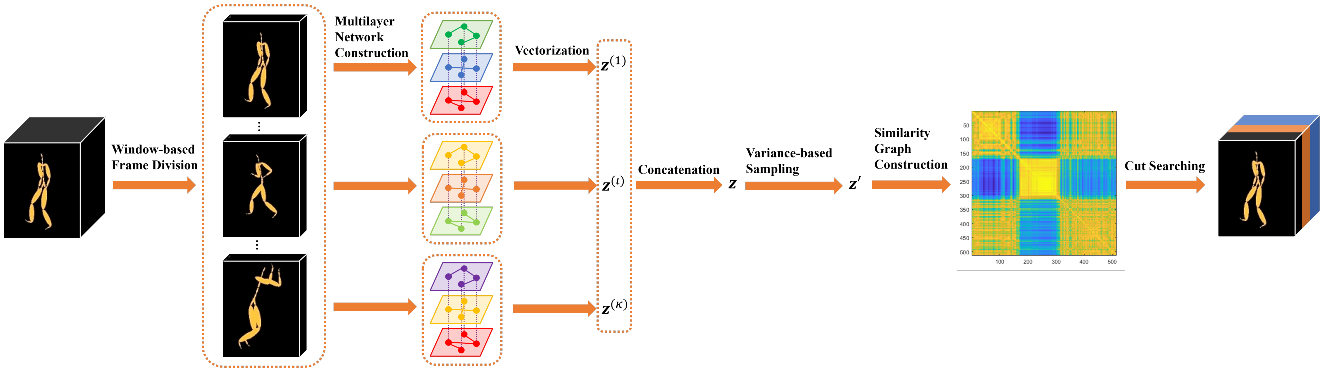


Fig. 6. Block Diagram of Vertex-based MLG Motion Segmentation.

Algorithm 1 Spectrum-based MLG Segmentation (SMLGS)

```
Input: Features of motion sequences with $M$ temporal frames and $N$ joints in each frame, where each joint in a given frame is annotated by $K$-dimensional features.
1. Calculate the spatial and temporal intrinsic resolutions, $\sigma_s$ and $\sigma_t$, of motion sequences in Eq. (9) and Eq. (10);
2. Cut the motion sequences into overlapping segments of length $W_d$ in frames;
for each segment do
    3. Construct the adjacency tensor based on Eq. (11);
    4. Estimate the singular bases using HOSVD in Eq. (3);
    5. Transform the signals/features to the singular space using these spectrum bases;
    6. Use a Haar-like highpass filter to extract features as Eq. (15) and Eq. (17);
end for
7. Calculate the similarity graph using the extracted features as Eq. (19);
8. Find the cutting frame based on the similarity graph using the region growing search technique in [36].
```

Algorithm 2 Vertex-based MLG Segmentation (VMLGS)

```
Input: Features of motion sequences with $M$ temporal frames and $N$ joints in each frame, where each joint in a given frame is annotated by $K$-dimensional features.
1. Calculate the spatial and temporal intrinsic resolutions, $\sigma_s$ and $\sigma_t$, of motion sequences in Eq. (9) and Eq. (10);
2. Cut the motion sequences into overlapping windows of length $W_d$ in frames;
for each segment do
    3. Construct the adjacency tensor based on Eq. (11) and reshape it into a vector $\mathbf{z}^{(m)}$;
end for
4. Concatenate all $\mathbf{z}^{(m)}$ into feature matrix $\mathbf{z}$ and calculate the variance for each row of $\mathbf{z}$;
5. Select the rows in $\mathbf{z}$ with $k$ highest variance to form the sampled feature matrix $\mathbf{z}'$;
6. Calculate the similarity graph using the extracted features as Eq. (21);
7. Find the cutting frame based on the similarity graph using the region growing search technique in [36].
```

- Then we find indices of the elements in $\sigma^2(\mathbf{z})$ with k largest variances and denote the indices by $\mathcal{I} = \{\mathcal{I}_1, \cdots, \mathcal{I}_k\} \in \mathbb{R}^k$;
- Finally, we construct the sampled feature matrix $\mathbf{z}' \in \mathbb{R}^{k \times N_{seg}}$ by keeping the rows in $\mathbf{z}$ with same indices in $\mathcal{I}$, i.e., the pth row in $\mathbf{z}'$ is the $\mathcal{I}_p$-th row of $\mathbf{z}$.

In this way we can remove the low frequency elements in the feature matrix.

With the extracted high-frequency structure signals, we use the column vector of $\mathbf{z}'$ to calculate the similarity graph $\mathbf{A}_{sim}$. Let $\mathbf{a}'^{(m)}$ be the m-th column vector of $\mathbf{a}'$. The similarity graph $\mathbf{A}_{sim}$ is calculated by

$$\mathbf{A}_{sim}[m, n] = \exp\left(-||\mathbf{z}'^{(m)} - \mathbf{z}'^{(n)}||_2^2\right). \tag{21}$$

We then convert the similarity graph A_{sim} into an SSSM as SMLGS, and find the cutting frames using the region growing search technique introduced in Section V-A. The major steps of vertex-based MLG Motion Segmentation (VMLGS) is presented in Algorithm 2.

VI. Experiments

We now present the experimental results of the proposed algorithms in both unsupervised and supervised setup compared to the existing clustering and recognition approaches.

A. Dataset

In our experiment, we test over two different datasets: 1) the CMU Graphics Lab Motion Capture Database; and 2) the Human Gait Database.

1) CMU Graphics Lab Motion Capture Database: The CMU graphics lab motion capture database[1] have 2605 trials in 6 categories and 23 subcategories. They are captured at 120 Hz with images of 4 megapixel resolution. An example motion trail in the database is shown in Fig. 2. We test all motion segmentation methods on trails 01 to 14 of subject 86, which have the human (supervisor)-labeled motion segmentation. In this work, we use the optimal cutting frame as the ground truth. We do not account for transitions between distinct actions in a manner similar to that of [9].

[1] http://mocap.cs.cmu.edu/

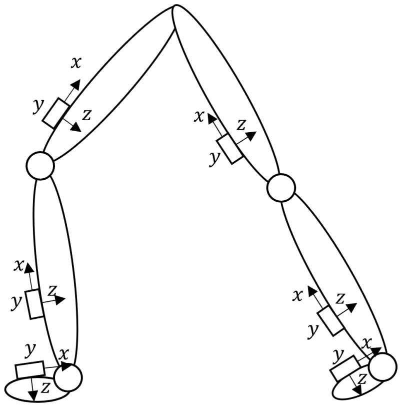


Fig. 7. Example of the location of inertial measurement units in HuGaDB.

TABLE I
ACCURACY OF MOTION SEGMENTATION ON CMU 86 DATASET

Dataset	Method				
	SMLGS	VMLGS	SC	ACA	EUTS
86_01	0.9067	0.9558	0.7372	0.9212	0.9505
86_02	0.9387	0.9396	0.8916	0.8891	0.9469
86_03	0.9277	0.9488	0.8406	0.8953	0.9262
86_04	0.9138	0.9294	0.7539	0.8735	0.9267
86_05	0.8928	0.9228	0.6523	0.8963	0.9252
86_06	0.9058	0.9631	0.7325	0.9237	0.9068
86_07	0.9377	0.9537	0.8920	0.9214	0.9449
86_08	0.9527	0.9317	0.7808	0.9384	0.9682
86_09	0.9471	0.9235	0.8334	0.8761	0.9058
86_10	0.9709	0.9666	0.9626	0.8928	0.9238
86_11	0.9380	0.9603	0.9228	0.9147	0.9672
86_12	0.9150	0.9712	0.8730	0.8498	0.9275
86_13	0.7671	0.8612	0.8075	0.8206	0.6073
86_14	0.9136	0.9131	0.6049	0.7308	0.9216
Average	0.9163	**0.9386**	0.8061	0.8817	0.9106

2) Human Gait Database (HuGaDB): HuGaDB is an action segmentation dataset, where the subjects record typical lower limb activities, e.g. walking, running, and cycling [4]. 18 subjects are included in this dataset. MoCap was performed with 6 inertial measurement units (IMUs) at a sampling frequency of 60 Hz in HuGaDB. Each IMU contains one accelerometer and one gyroscope. The IMUs were placed on the right and left thighs, shins and feet, as shown in Fig. 7. This dataset contains 364 IMU trials in 12 action categories. Since the accelerometer data and gyroscope data are different captures of the same motion, they should be processed in different ways. In our test, we further divide each trial into three datasets: accelerometer only (ACC) dataset, gyroscope only (GYRO) dataset, and one dataset containing both accelerometer and gyroscope data. We test all motion segmentation methods on all these three datasets to further investigate the robustness of motion segmentation methods on different kinds of data.

B. Unsupervised Motion Segmentation

We first evaluate the performance of proposed methods in unsupervised setup. Here, we compare our proposed method with four unsupervised motion methods: spectral clustering (SC), Aligned Cluster Analysis (ACA), Hierarchical Aligned Cluster Analysis (HACA) [9], and Efficient Unsupervised Temporal Segmentation (EUTS) [36] on CMU and HuGaDB datasets. For the CMU dataset, we convert data into 3D coordinates of all joints as input for our proposed algorithm, i.e., $\mathbf{x}_{\alpha i} \in \mathbb{R}^3$. For the HuGaDB dataset, we directly use the accelerometer data $\mathbf{a}_{\alpha i} \in \mathbb{R}^3$ and gyroscope data $\mathbf{g}_{\alpha i} \in \mathbb{R}^3$ as the input of out proposed algorithm, where $\mathbf{x}_{\alpha i} \in \mathbb{R}^6$. The parameters of the proposed method are fixed for all trials in one dataset. We vary the window size W_d from 2 to 10 in our experiment. We chose the parameters of all the existing methods based on the set of parameters provided in the original paper and codes with the best performance. To evaluate the clustering accuracy, we calculate the mean intersection over union (mIoU) on each trial.

TABLE II
AVERAGE ACCURACY OF MOTION SEGMENTATION ON HUGADB DATASET

Dataset	Method				
	SC	ACA	HACA	SMLGS	VMLGS
ACC	0.6162	0.7167	0.7210	**0.8171**	0.7667
GYRO	0.4738	0.5436	0.5276	0.7000	**0.8495**
Both	0.4540	0.5341	0.5314	0.8249	**0.8551**

We summarize the average accuracy and segmentation results on CMU datasets in Table. I and Fig. 8, respectively. As shown in Fig. 8, our proposed spectrum-based (SMLGS) algorithm tends to segment each motion sequence into larger sections, whereas the vertex-based (VMLGS) algorithm cuts the motion sequence into finer sections. This distinction highlights the focus of SMLGS algorithm in capturing significant differences between distinct motions, whereas the VMLGS algorithm excels at detecting subtle variations among elements, even within the same motion. The average accuracy results presented in Table I indicate that VMLGS algorithm achieves superior overall accuracy. However, the increased number of segments generated by VMLGS may pose a greater challenge to the classifier in real-world applications. On the other hand, the number of segments generated by SMLGS closely aligns with the ground truth in the majority of CMU 86 datasets. The average accuracy results on HuGaDB dataset are shown in Table II, and an example of segmentation on the first dataset in HuGaDB is shown in Fig. 10. The best performance is marked in bold font and the second best result uses underlined font. From these test results, VMLGS provides the best performance in most of the dataset while SMLGS often ranks as the second best. The results demonstrate the strength of M-GSP in body motion analysis and the robustness of our proposed algorithms. From the visualization results, our proposed algorithms provide fewer segments and a clearer segment boundary which is closer to ground truth. This further demonstrates the benefits of applying M-GSP in body motion analysis.

C. Supervised Motion Recognition

Although our algorithms are designed under unsupervised setup, we also test and provide comparison with supervised

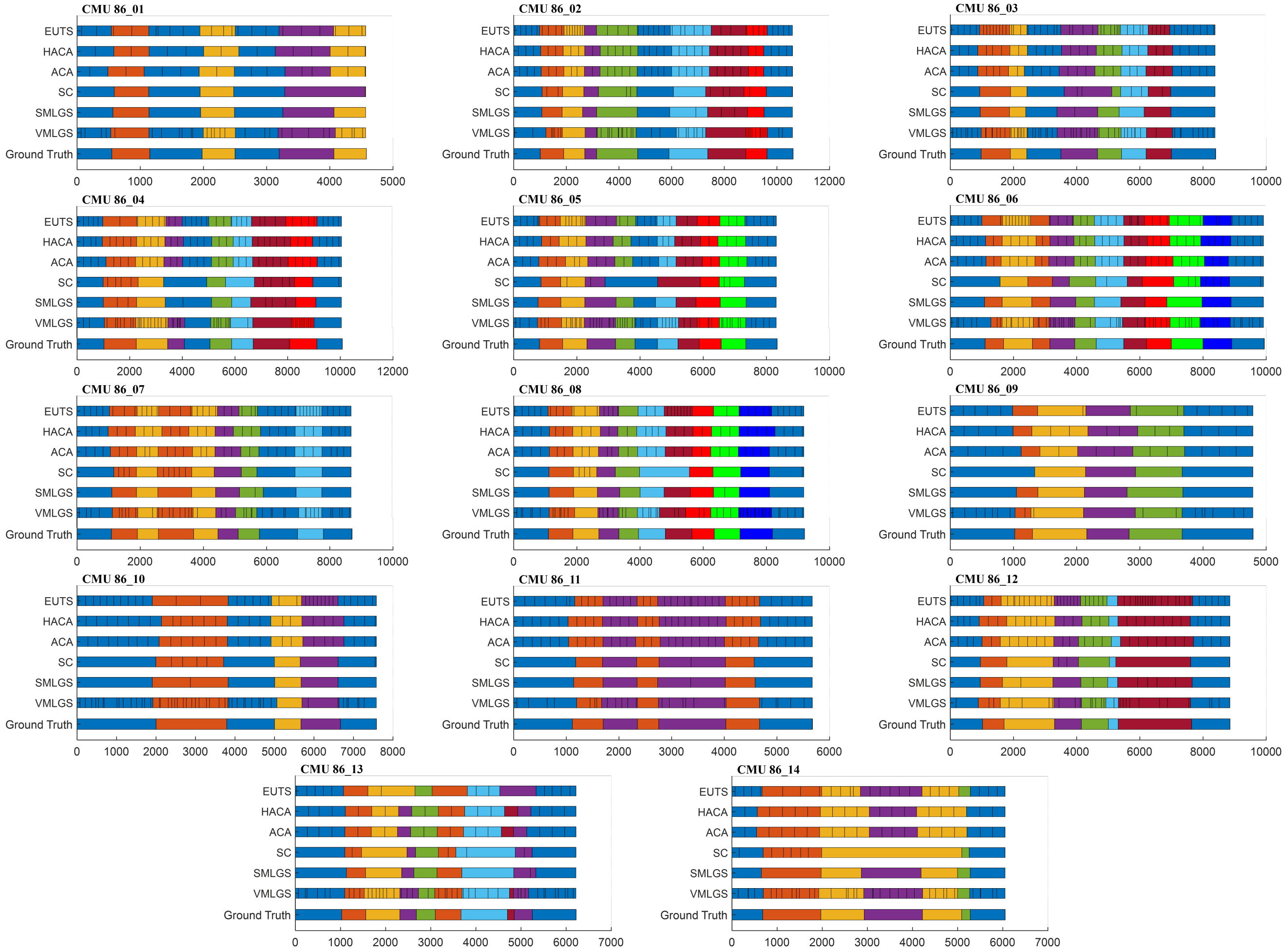


Fig. 8. Segmentation result on CMU trial 86.

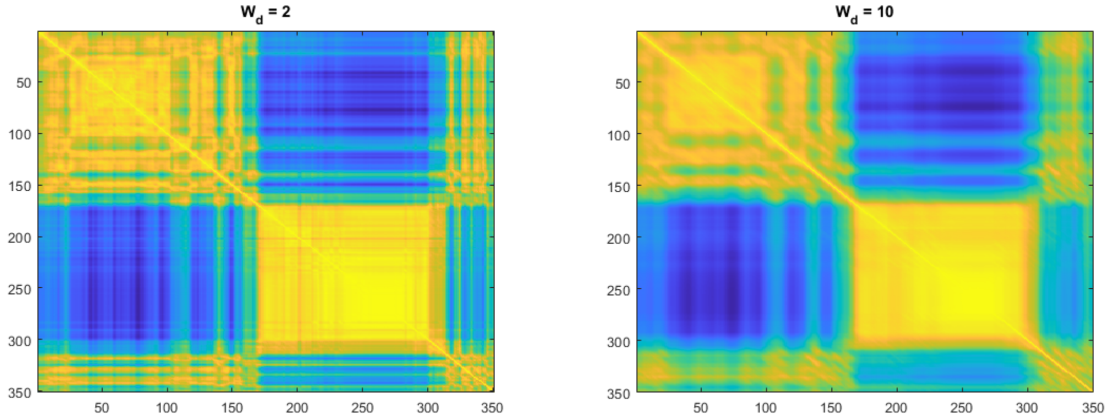


Fig. 9. Example of the similarity matrix for different window sizes on HuGaDB dataset with GYRO data.

motion segmentation. To ensure a fair comparison, we tune the parameters of our algorithms based on the best performance from the training dataset. Here, we compare our proposed method with several existing supervised motion methods: (1) bidirectional long short term memory-based network (Bi-LSTM) [12], (2) temporal convolutional neural networks (TCN) [49], (3) spatial-temporal graph convolutional neural network (ST-GCN) [21], (4) multi-stage temporal convolutional neural networks (MS-TCN) [50], and (5) multi-stage spatial-temporal graph convolutional neural networks (MS-GCN) [22] on HuGaDB datasets.

We present the results in Table III. From the results, our proposed algorithms have competitive performance against supervised learning machines, even without ever utilizing the label

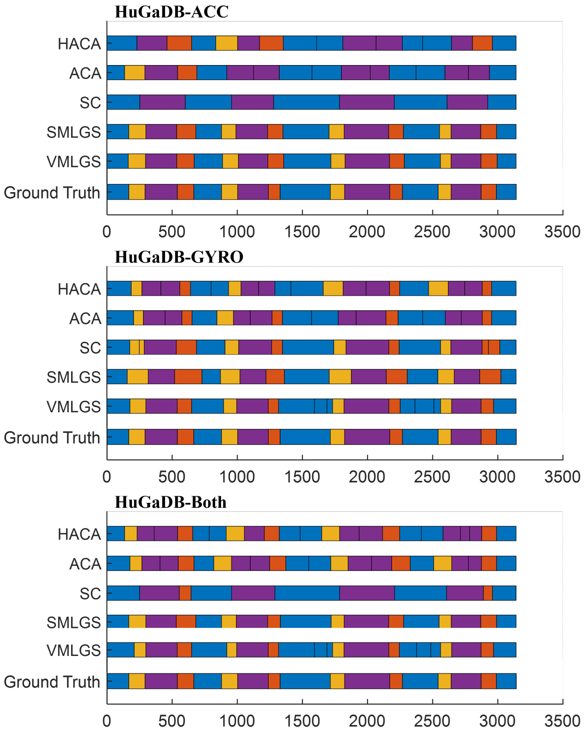


Fig. 10. Segmentation result on HuGaDB trial 01 data 00.

TABLE III
AVERAGE ACCURACY ON HUGADB DATASET WITH BOTH ACCELEROMETER AND GYROSCOPE DATA.

Method	Accuracy (%)
Bi-LSTM	86.1
TCN	88.3
ST-GCN	88.7
MS-TCN	86.8
MS-GCN	90.4
SMLGS (w. optimal parameters)	87.9
VMLGS (w. optimal parameters)	90.2

information for clustering. This demonstrates the effectiveness of our proposed method in terms of feature extraction from body motion sequences. Note that our clustering algorithms can be easily integrated with the supervised learning machines to further improve the performance. Furthermore, our proposed MLG-based algorithms do not require large amount of labeled data, leading to significant saving in labeling efforts for practical applications. We shall investigate application of M-GSP in deep learning in our future works.

D. Ablation study on window size

In this section we test our proposed method under different window sizes to show its characteristics as a short-time processing method. We test the average accuracy on CMU dataset, ACC dataset in HuGaDB and GYRO dataset in HuGaDB. We only change the window size W_d of each segment where we extract the features while keeping all other parameters fixed. From Fig. 11, our result indicates that the optimal window size would vary for different datasets. While a larger window size tends to decrease the average accuracy performance once it exceeds the optimal value (as shown in Fig. 11), both SMLGS and VMLGS exhibit a performance loss under 5% when compared the result obtained with a window size of $W_d = 10$ and the optimal configuration. In practice, such consistency and stability in performance make it easier in selecting parameters for our proposed algorithms. As shown in Fig. 9, the similarity matrix of window size $W_d = 10$ is smoother than the similarity matrix of $W_d = 2$, which makes it more difficult to accurately locate the cut frame between two body motions. In practice, the optimal window size may be estimated from a small subset within the entire dataset.

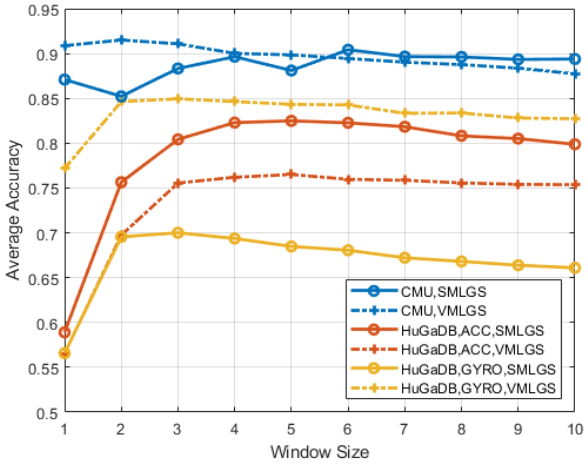
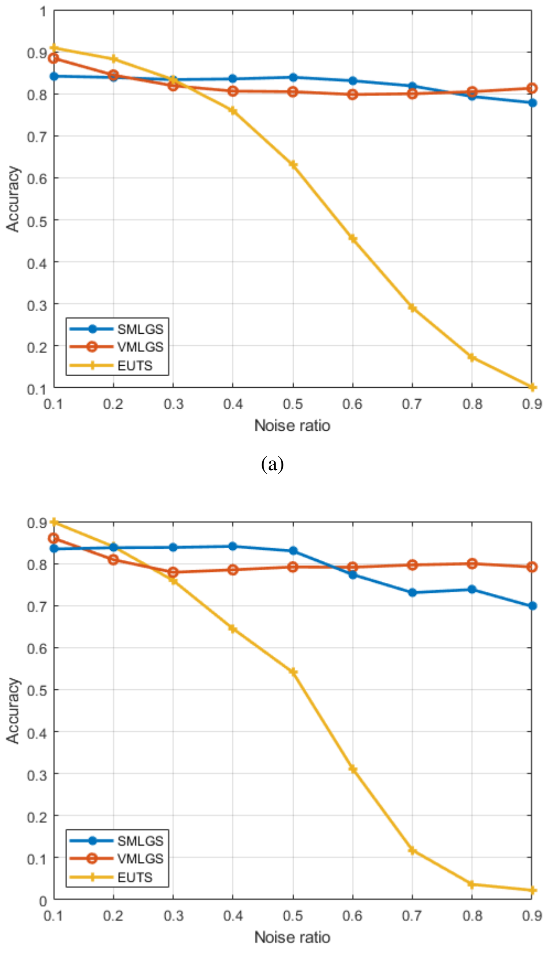


Fig. 11. Average accuracy for different window sizes.

Note also that SMLGS performs better in the ACC dataset in HuGaDB while the VMLGS is more robust across different datasets. The reason is that SMLGS implements the spectrum decomposition to extract additional features while introducing uncertainty during HOSVD. Thus, for simpler datasets, such as CMU data with only coordinate information, it could lead to superior performance with extracted spectral features. On the other hand, VMLGS evaluates the MLG via structural features, which is more stable and robust in complicated dataset, such as HuGaDB.

E. Robustness

In this part of test, we compare our proposed method with Efficient Unsupervised Temporal Segmentation (EUTS) [36] on CMU dataset with Gaussian noise. We add the Gaussian noise directly to the original dataset with varying standard deviation σ between 0.1 and 0.2. To test robustness against different level of noise, we add the Gaussian noise to 10% to 90% of the frames in the sequence. These noisy frames are randomly selected with equal probability and we repeat the process 10 times for each trial for each noisy frame ratio. In order to minimize the effect of randomness, we average the accuracy over these 10 noisy samples to arrive at the final result. As shown in Fig. 12, our proposed method exhibits stronger robustness against additive Gaussian noise.



(a)

(b)

Fig. 12. Average accuracy on CMU trial 86 with different noisy frame ratio. (a) $\sigma = 0.1$; (b) $\sigma = 0.2$.

VII. Conclusions

This work studies the use of M-GSP for body motion analysis. More specially, we introduce the MLG models for body motion sequence, with which we propose two different M-GSP filter-based algorithms for motion segmentation. Our proposed methods are easier to implement, and show robustness across multiple datasets. Our experimental results demonstrated the efficacy of the proposed methods and the potentials of M-GSP in motion analysis. This work establishes M-GSP as an efficient tool to model multilateral relationship and to extract features in motion sequence applications. In our future works, we shall investigate new ways to integrate machine learning methods and M-GSP for better feature extraction. The interpretation of body motion from the perspective of graph Fourier space is another interesting direction for exploration.

References

[1] T. B. Moeslund and E. Granum, “A survey of computer vision-based human motion capture,” *Comput. Vis. Image Underst.*, vol. 81, no. 3, pp. 231-268, Mar. 2001.

[2] Y. Desmarais, D. Mottet, P. Slangen and P. Montesinos, “A review of 3d human pose estimation algorithms for markerless motion capture,” *Comput. Vis. Image Underst.*, vol. 212, Nov. 2021.

[3] J. Sedmidubsky, P. Elias, P. Budikova and P. Zezula, “Content-Based Management of Human Motion Data: Survey and Challenges,” *IEEE Access*, vol. 9, pp. 64241-64255, Apr. 2021.

[4] R. Chereshnev and A. Kertész-Farkas, “Hugadb: Human gait database for activity recognition from wearable inertial sensor networks,” in *Proc. Int. Conf. Anal. Images, Soc. Netw. Texts*, 2017, pp. 131-141.

[5] L. Wang, Z. Ding and Y. Fu, “Low-Rank transfer human motion segmentation,” *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 1023-1034, Feb. 2019.

[6] L. Xi, W. Chen, X. Wu, Z. Liu and Z. Li, “Online Unsupervised Video Object Segmentation via Contrastive Motion Clustering,” *IEEE Trans. Circuits Syst. Video Technol.*, doi: 10.1109/TCSVT.2023.3288878.

[7] X. Xu, L. Zhang, L. -F. Cheong, Z. Li and C. Zhu, “Learning Clustering for Motion Segmentation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 908-919, March 2022.

[8] S. Lin, A. Yang, T. Lai, J. Weng and H. Wang, “Multi-motion Segmentation via Co-attention-induced Heterogeneous Model Fitting,” *IEEE Trans. Circuits Syst. Video Technol.*, doi: 10.1109/TCSVT.2023.3298319.

[9] F. Zhou, F. De la Torre and J. K. Hodgins, “Hierarchical Aligned Cluster Analysis for Temporal Clustering of Human Motion,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 582-596, Mar. 2013.

[10] G. Xia, H. Sun, L. Feng, G. Zhang and Y. Liu, “Human Motion Segmentation via Robust Kernel Sparse Subspace Clustering,” *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 135-150, Jan. 2018.

[11] A. Byravan and D. Fox, “SE3-nets: Learning rigid body motion using deep neural networks,” in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017, pp. 173-180.

[12] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM networks,” in *Proceedings of 2005 IEEE International Joint Conference on Neural Networks*, vol. 4, pp. 2047-2052, Aug. 2005.

[13] S. Li, Y. A. Farha, Y. Liu, M. -M. Cheng and J. Gall, “MS-TCN++: Multi-Stage Temporal Convolutional Network for Action Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 6647-6658, Jun. 2023.

[14] Z. Wang, Z. Gao, L. Wang, Z. Li, and G. Wu, “Boundary-Aware Cascade Networks for Temporal Action Segmentation,” in *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[15] Y. Li, Z. Dong, K. Liu, L. Feng, L. Hu, J. Zhu, L. Xu, S. Liu et al., “Efficient two-step networks for temporal action segmentation,” *Neurocomputing*, vol. 454, pp. 373-381, 2021.

[16] H. Ahn and D. Lee, “Refining action segmentation with hierarchical video representations,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16302-16310.

[17] L. Xu, Q. Wang, X. Lin and L. Yuan, “An efficient framework for few-shot skeleton-based temporal action segmentation,” *Computer Vision and Image Understanding*, vol. 232, pp. 103707, 2023.

[18] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura and P. Vandergheynst, “Graph Signal Processing: Overview, Challenges, and Applications,” *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808-828, May 2018.

[19] T.N. Kipf, and M. Welling, “Semi-supervised classification with graph convolutional networks,” 2016, arXiv:1609.02907.

[20] S.Zhang, M. Wang, S. Liu,P. Y. Chen, and J. Xiong, “Fast learning of graph neural networks with guaranteed generalizability: one-hidden-layer case,” in *International Conference on Machine Learning*, Nov. 2020, pp. 11268-11277.

[21] S. Yan, Y. Xiong and D. Lin, “Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, pp. 1-9, Apr. 2018.

[22] B. Filtjens, B. Vanrumste and P. Slaets, “Skeleton-Based Action Segmentation with Multi-Stage Spatial-Temporal Graph Convolutional Neural Networks,” *IEEE Transactions on Emerging Topics in Computing*, pp. 1-11, Dec. 2022.

[23] P. Ghosh, Y. Yao, L. Davis, and A. Divakaran, “Stacked spatio-temporal graph convolutional networks for action segmentation,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[24] K. Cheng, Y. Zhang, C. Cao, L. Shi, J. Cheng, and H. Lu, “Decoupling GCN with dropgraph module for skeleton-based action recognition,” in *Proceedings of European Conference on Computer Vision (ECCV)*, Aug. 2020, pp. 536-553.

[25] S. Miao, Y. Hou, Z. Gao, M. Xu, and W. Li, “A central difference graph convolutional operator for skeleton-based action recognition,”

IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 7, pp. 4893-4899, 2021.

[26] C. Wu, X.-J. Wu, and J. Kittler, “Graph2net: Perceptually-enriched graph learning for skeleton-based action recognition,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 2120-2132, 2021.

[27] Y. -H. Li, K. -Y. Liu, S. -L. Liu, L. Feng and H. Qiao, “Involving Distinguished Temporal Graph Convolutional Networks for Skeleton-Based Temporal Action Segmentation,” *IEEE Transactions on Circuits and Systems for Video Technology*, doi: 10.1109/TCSVT.2023.3285416.

[28] Y. Zhou, G. Gallego, X. Lu, S. Liu and S. Shen, “Event-Based Motion Segmentation With Spatio-Temporal Graph Cuts,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 4868-4880, Aug. 2023.

[29] F. Grassi, A. Loukas, N. Perraudin and B. Ricaud, “A time-vertex signal processing framework: scalable processing and meaningful representations for time-series on graphs,” *IEEE Transactions on Signal Processing*, vol. 66, no. 3, pp. 817-829, Feb. 2018.

[30] S. Zhang, Q. Deng and Z. Ding, “Signal Processing over Multilayer Graphs: Theoretical Foundations and Practical Applications,” *IEEE Internet of Things Journal*, Jul. 2023.

[31] J. S. Stanley, E. C. Chi and G. Mishne, “Multiway Graph Signal Processing on Tensors: Integrative Analysis of Irregular Geometries,” *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 160-173, Nov. 2020

[32] S. Zhang, Q. Deng and Z. Ding, “Image Processing via Multilayer Graph Spectra,” 2021, arXiv :2108.13639.

[33] S. Zhang, Q. Deng and Z. Ding, “Multilayer graph spectral analysis for hyperspectral images,” *EURASIP Journal on Advances in Signal Processing*, vol. 1, no. 92, pp. 1-25, Oct. 2022.

[34] J. B. Kim, H. S. Park, M. H. Park, and H. J. Kim, “A real-time region-based motion segmentation using adaptive thresholding and K-means clustering,” in *AI 2001: Advances in Artificial Intelligence: 14th Australian Joint Conference on Artificial Intelligence Adelaide*, Australia, Dec. 2001, pp. 213-224.

[35] F. Lauer and C. Schnorr, “Spectral clustering of linear subspaces for motion segmentation,” in *2009 IEEE 12th International Conference on Computer Vision*, Kyoto, Japan, 2009, pp. 678-685.

[36] B. Krüger, A. Vögele, T. Willig, A. Yao, R. Klein and A. Weber, “Efficient Unsupervised Temporal Segmentation of Motion Data,” *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 797-812, Apr. 2017.

[37] Y. Bai, L. Wang, Y. Liu, Y. Yin, H. Di and Y. Fu, “Human Motion Segmentation via Velocity-Sensitive Dual-Side Auto-Encoder,” *IEEE Transactions on Image Processing*, vol. 32, pp. 524-536, 2023.

[38] X. Dong, D. Thanou, L. Toni, M. Bronstein and P. Frossard, “Graph Signal Processing for Machine Learning: A Review and New Perspectives,” *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 117-127, Nov. 2020.

[39] S. Zhang, Z. Ding and S. Cui, “Hypergraph Spectral Analysis and Processing in 3D Point Cloud,” *IEEE Transactions on Image Processing*, vol. 30, pp. 1193-1206, 2021.

[40] M. Onuki, S. Ono, M. Yamagishi and Y. Tanaka, “Graph Signal Denoising via Trilateral Filter on Graph Spectral Domain,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 137-148, Jun. 2016.

[41] Q. Deng, S. Zhang and Z. Ding, “An Efficient Hypergraph Approach to Robust Point Cloud Resampling,” *IEEE Transactions on Image Processing*, vol. 31, pp. 1924-1937, Feb. 2022.

[42] S. Zhang, Z. Ding and S. Cui, “Introducing Hypergraph Signal Processing: Theoretical Foundation and Practical Applications,” *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 639-660, Jan. 2020.

[43] K. Pena-Pena, D. L. Lau and G. R. Arce, “t-HGSP: Hypergraph Signal Processing Using t-Product Tensor Decompositions,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 9, pp. 329-345, 2023.

[44] S. Barbarossa and S. Sardellitti, “Topological Signal Processing Over Simplicial Complexes,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 2992-3007, 2020.

[45] S. Zhang, H. Zhang, H. Li and S. Cui, “Tensor-based Spectral Analysis of Cascading Failures over Multilayer Complex Systems,” in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, USA, 2018, pp. 997-1004

[46] L. De Lathauwer, B. De Moor and J. Vandewalle, “A multilinear singular value decomposition,” *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253-1278, 2000.

[47] L. Herda, P. Fua, R. Plankers, R. Boulic and D. Thalmann, “Skeleton-based motion capture for robust reconstruction of human motion,” in *Proceedings Computer Animation 2000*, Philadelphia, PA, USA, 2000, pp. 77-83

[48] G. Xia, P. Xue, H. Sun, Y. Sun, D. Zhang and Q. Liu, “Local Self-Expression Subspace Learning Network for Motion Capture Data,” *IEEE Transactions on Image Processing*, vol. 31, pp. 4869-4883, Jul. 2022.

[49] C. Lea, M. D. Flynn, R. Vidal, A. Reiter and G. D. Hager, “Temporal Convolutional Networks for Action Segmentation and Detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1003-1012, Jul. 2017.

[50] Y. Abu Farha and J. Gall, “MS-TCN: Multi-stage temporal convolutional network for action segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3570-3579, Jun. 2019.