3D Skeletal Gesture Recognition via Hidden States Exploration

Xin Liu, Henglin Shi, Xiaopeng Hong, Haoyu Chen,
Dacheng Tao, *Fellow, IEEE*, and Guoying Zhao, *Senior Member, IEEE*

Abstract—**Temporal dynamics is an open issue for modeling human body gestures. A solution is resorting to the generative models, such as the hidden Markov model (HMM). Nevertheless, most of the work assumes fixed anchors for each hidden state, which make it hard to describe the explicit temporal structure of gestures. Based on the observation that a gesture is a time series with distinctly defined phases, we propose a new formulation to build temporal compositions of gestures by the low-rank matrix decomposition. The only assumption is that the gesture's "hold" phases with static poses are linearly correlated among each other. As such, a gesture sequence could be segmented into temporal states with semantically meaningful and discriminative concepts. Furthermore, different to traditional HMMs which tend to use specific distance metric for clustering and ignore the temporal contextual information when estimating the emission probability, we utilize the long short-term memory to learn probability distributions over states of HMM. The proposed method is validated on multiple challenging datasets. Experiments demonstrate that our approach can effectively work on a wide range of gestures, and achieve state-of-the-art performance.**

Index Terms—**Gesture recognition, hidden Markov model, deep neural networks, matrix decomposition**

I. Introduction

Human body gesture analysis is one of the core components in the thriving research fields of human-computer interaction, intelligent security surveillance, and video games. Recently, 3D skeletal data is gaining popularity as it simplifies the task from using monocular RGB cameras to more sophisticated sensors, such as the Kinect [1] [2]. This feature can explicitly localize gesture performers and produce the trajectories of human skeletal joints. Compared to RGB input, skeletal data is robust to background dynamics and invariant to camera view.

Over the last few years, numerous 3D skeleton-based models have been developed for human activity recognition, ranging from handcrafted-based feature representations, like histogram of 3D joints (HOJ3D) [7], EigenJoints by principal component analysis (PCA) [8], manifold representations [9]–[12], discriminative key-frames [13], histogram of oriented 4D normals (HON4D) [14], sequence of most informative joints (SMIJ) [15], rotation and relative velocity (RVV) [16]; to various forms of parametric approaches such as actionlets ensemble [17] [18], maximum entropy Markov model (MEMM) [19], latent structural SVM (pose-based) [20], hidden Markov models (HMM) [21] [22], conditional random field (CRF) [23] [24], latent Dirichlet allocation (LDA) [25], naive Bayes nearest neighbor (NBNN) [26], latent max-margin multitask learning (LM^{3}TL) [27]; and also including plenty of deep learning methods, i.e. deep belief network (DBN) [28] [29], convolutional neural network (CNN) [30] [31], and recurrent neural network (RNN) [32]–[38]. Rather than covering all works exhaustively, we refer interested readers to recent surveys [39] [40].

Despite the encouraging progresses having been made by various studies, accurate recognition of the human gestures in unconstrained settings is still challenging. Especially, one open issue of human gestures recognition lies in the temporal dynamics. For instance, even the same gesture performed by the same person can occur at different speeds and different starting/ending points, let alone for cases with different performers. Consequently, the variance of a category of human behavior can be very large, and if temporal dynamics are ignored, the accuracy of recognition would undoubtedly deteriorate [12].

Recently, researchers have been resorting to modelling human behaviors by studying temporal structures, e.g. [10] [26] [41]. However, most of these models focus on human actions rather than body gestures. Compared to actions, the structural property of gestures is more semantically meaningful and discriminative. According to the research on gesture movements [5] [6], a gesture instance can be decomposed into the following gesticular phases (see examples in Fig. 1):

1) *Resting*: see Fig. 1 (a).

2) *Stroke*: hands movement that expresses the meaning of the gesture, see Fig. 1 (a)→(b)→(c).

3) *Post-stroke hold*: brief pause at the end of a stroke, maintaining the hands' configuration and position, see Fig. 1 (c).

This work was supported in part by the Academy of Finland for project MiGA (grant 316765) and ICT 2023 project (grant 328115), the strategic Funds of the University of Oulu, Finland, the Infotech Oulu, and in part by Endeavour Research Fellowship of Australian Government Department of Education and Training, and in part by Australian Research Council Project FL-170100117. As well, the authors wish to acknowledge CSC - IT Center for Science, Finland, for computational resources. *(Corresponding author: Guoying Zhao.)*

X. Liu is with the Center for Machine Vision and Signal Analysis, University of Oulu, FI-90014, Finland, and also with the School of Information Technologies, Faculty of Engineering and Information Technologies, The University of Sydney, Australia. (E-mail: linuxsino@gmail.com)

H. Shi and H. Chen are with the Center for Machine Vision and Signal Analysis, University of Oulu, FI-90014, Finland. (E-mail: henglin.shi, chen.haoyu@oulu.fi)

X. Hong is with Xi'an Jiaotong University, Xi'an, China. (Email: hongxiaopeng@mail.xjtu.edu.cn)

D. Tao is with the School of Computer Science, in the Faculty of Engineering, at The University of Sydney, 6 Cleveland St, Darlington, NSW 2008, Australia (email: dacheng.tao@sydney.edu.au).

G. Zhao is with the School of Information and Technology, Northwest University, 710069, China, and also with the Center for Machine Vision and Signal Analysis, University of Oulu, FI-90014, Finland. (E-mail: guoying.zhao@oulu.fi)

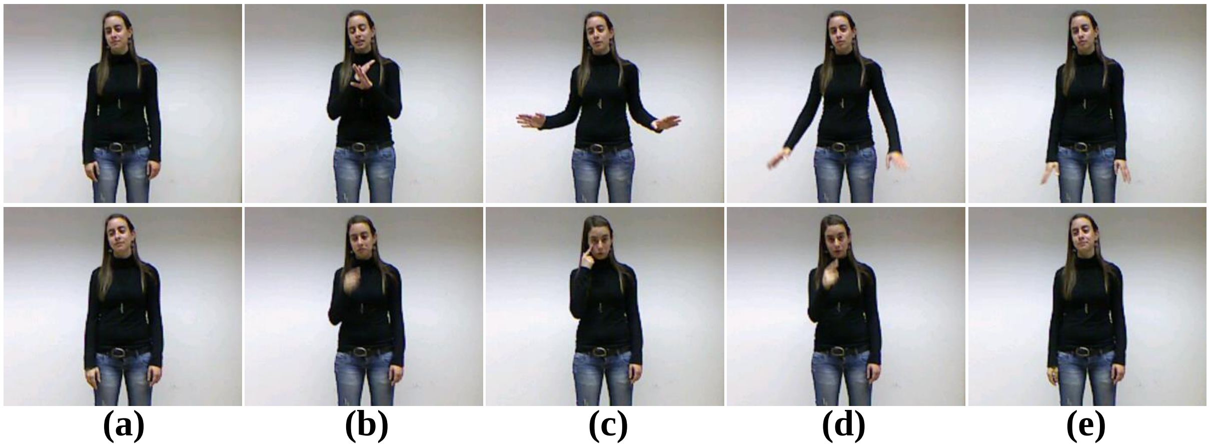


Fig. 1. Frames (cropped) selected from two gestures [3] representing the meanings of “basta (enough)” and “furbo (clever)” respectively [4]. These frames illustrate a gesture consists of a series of “gesticular phases”: *Resting*→*Stroke*→*Post-stroke hold* →*Retraction*→*Resting* [5] [6]. (a) *Resting*, (a)→(b)→(c) *Stroke*, (c) *Post-stroke hold*, (c)→(d)→(e) *Retraction*, (e) *Resting*. It is noted that two additional phases, namely the *Preparation* and *Pre-stroke hold* are defined in [5] [6], but they are optional and can be merged into the obligatory phase *Stroke* [6]. For example, the lasting time of *Pre-stroke hold* in “furbo (clever)” is too short to be determined.

4) *Retraction*: the hands move back to a rest position to conclude a gesture unit, see Fig. 1 (c)→(d)→(e).
5) *Resting*: see Fig. 1 (e).

From the above definitions, we can conclude that two phases (2) and (4) with hands movements, namely the *Stroke* and *Retraction* are partitioned by three “hold” phases (1, 3, 5) with static poses, namely *Resting* (Independent hold [6]), *Post-stroke hold*, and *Resting* again. In other words, once we can identify these “hold” phases, the temporal structure of a gesture can be obtained.

Based on this observation, in this paper, we develop a novel model for human gesture recognition aiming to address the difficulties of modeling temporal dynamics (see Fig. 2). We treat one human gesture as a series of separated phases, each of which is associated with a segment of an unfixed-length, as Fig. 3 (c) illustrates, and we propose to globally capture the temporal evolution of gestures by a generative model which is built upon a recurrent neural network to memorize contextual information for better prediction of transition and emission probabilities. We formulate the problem in a unified framework named Hidden States Learning by Long Short-Term Memory (HSL-LSTM). The main contributions are summarized as follows:

- We propose a new formulation to explore the temporal structure of human gestures based on a low-rank matrix decomposition algorithm. The only assumption is that the gesture’s “hold” phases with static poses are linearly correlated with each other, which can be captured by the low-rank matrix. We also explicitly consider the column-block prior of the outlier signals, the part of hand movements (phases) which cannot be fitted into the low-rank model. Thus, the temporal structure alignment is interpreted as a binary clustering problem (illustrated in Fig. 2 (c)). In contrast to conventional methods using fixed anchors (Fig. 3 (d)), the proposed method can segment a gesture sequence into temporal compositions (phases) with semantically meaningful and discriminative concepts (Fig. 3 (c)).

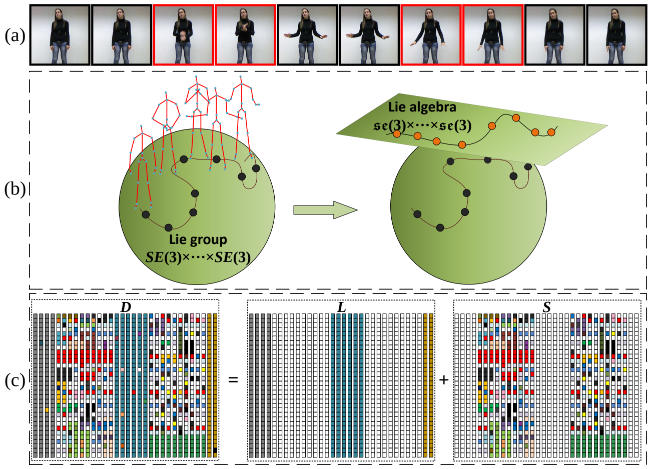

(a)
(b)
Lie algebra
$\mathfrak{se}(3)\times\cdots\times\mathfrak{se}(3)$
Lie group
$SE(3)\times\cdots\times SE(3)$
(c)
D
=
L
+
S

Fig. 2. (a) Frames (cropped) selected from gesture “basta (enough)” [3] which is composed by five distinctly phases, three “hold” phases with the black border and two “motion” phases with the red border. (b) Representation of a gesture (skeletal sequence) as a curve on the Lie group $SE(3) \times \cdots \times SE(3)$ (manifold curved space), and can be mapped into its Lie algebra (vector space). (c) Illustration of matrix decomposition for exploring hidden states.

- We propose a new hidden states learning model based on an RNN. Different temporal compositions actually correspond to the different hidden states of HMM. The usage of HMM allows us to distribute heterogeneous information of one gesture class over many states (phases), and is key to improve the capability of modeling complex patterns. Different from traditional HMMs using the Gaussian mixture model (GMM) [42] which ignores the temporal contextual information and uses specific distance metrics for clustering, the LSTM is utilized to enhance the HMM by estimating better probabilities as it provides robust classification of small temporal chunks.
- We introduce a Lie group based feature to better represent the 3D geometric relationships between various body

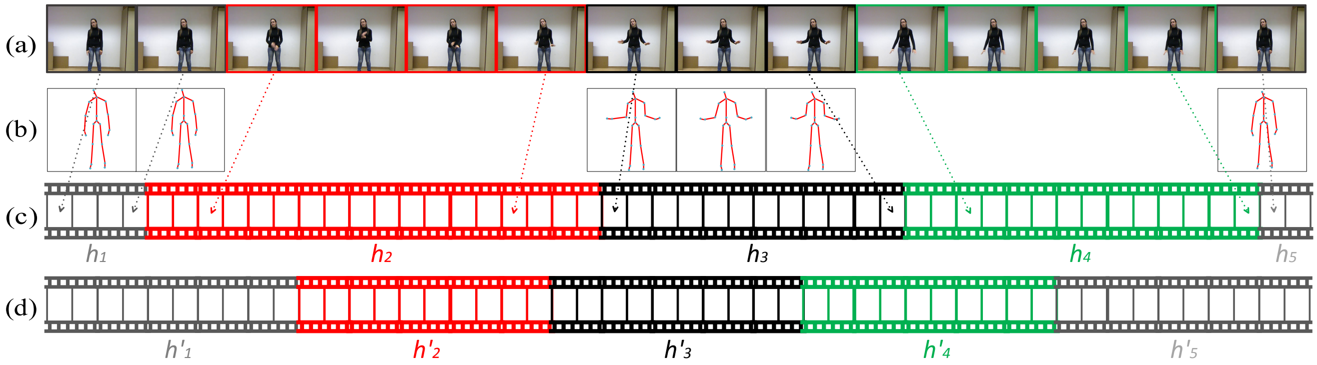


Fig. 3. Illustration of phases (hidden states) of a gesture sequence with temporal structures. (a) Frames selected from a gesture [3] representing the meaning of "basta (enough)", (b) Skeletons (corresponding to selected frames) of static poses from "hold" phases, (c) Temporal structure (phases) segmentation by proposed, resulting hidden states h_1 (*Resting*), h_2 (*Stroke*), h_3 (*Post-stroke hold*), h_4 (*Retraction*), h_5 (*Resting*), (d) Fixed anchors based methods with equal-sized segmentation, resulting in hidden states $h_1^{'}$, $h_2^{'}$, $h_3^{'}$, $h_4^{'}$, $h_5^{'}$.

parts (illustrated in Fig. 2 (b)). Moreover, we propose a new gesture recognition framework by absorbing the advantages of the HMM and LSTM. Rather than model the whole sequences (a gesture) within the LSTM as conventional RNN methods [32] [34] [36] do, we feed the network by temporal compositions (hidden states) with shorter temporal length and more training samples. Therefore, the parameter learning for LSTM with large size of training data is not needed. Experiments demonstrate that our approach achieves state-of-the-art performance on 3D skeleton based human gesture recognition benchmarks.

The remainder of this paper is organized as follows. Section II reviews related methods. In Section III, the gesture modeling problem in HMM is formulated. In Section IV, the Lie group based 3D skeleton representation is briefly described. In Section V, the low-rank and column-block sparsity decomposition for temporal structure segmentation is proposed. In Section VI, the hidden states learning via LSTM is presented. Experiments and discussions are presented in Section VII and conclusions are drawn in Section VIII.

The preliminary work has appeared in [4].

II. Related Methods

For addressing the temporal dynamics problem of human activity analysis (including actions and gestures) on 3D skeleton data, a variety of different approaches have been proposed. We provide a categorized overview of the related literature mainly on local temporal modeling, generative models, and recurrent neural networks.

A. Approaches with local temporal modeling

To account for temporal dynamics, a common treatment is the dynamic time warping (DTW), as in Lie group [9], RVV [16], and LM^{3}TL [27]. DTW resorts to finding an optimal temporal alignment (reference), then warp all sequences of the same category to that nominal reference. Finally, a classifier such as the SVM is typically utilized to perform the recognition task. Nevertheless, the performance of DTW is heavily dependent on the metric used to measure the similarity of frames (features). Furthermore, for periodic gestures, DTW tends to produce large temporal misalignments, which may encumber the accuracy of classification [17]. Gong *et al.* [10] proposed a dynamic manifold warping (DMW) method to calculate the motion similarity among video sequences, which is an adaptation of DTW methods in the manifold space. In [11], the 3D skeleton joint trajectories were modeled by curves in the Riemannian manifold space, and a dynamic programming (DP) based distance function was applied to compare them. Wang *et al.* [17] [18] introduced the local occupancy pattern (LOP) to represent 3D human activities, and proposed the Fourier temporal pyramid (FTP) to capture local temporal patterns, which is more robust to noise and temporal misalignments than DTW. On the other hand, FTP is restricted by the width of the time window and can only utilize limited contextual information [32]. In [13], Zanfir *et al.* proposed a moving pose descriptor by integrating the normalized positions of joints from discriminative key-frames, as well as their velocities and accelerations. Then, a non-parametric K-nearest-neighbors (KNN) is adopted for action classification. Leveraging key-frames can help to exclude frames that are less relevant to the underlying gestures, but in comparison to the holistic-based approaches, damaged essential information is inevitable. In these methods, the local temporal dynamics is represented within a certain time window, so they cannot globally capture the temporal evolution of gestures [32].

B. Approaches with generative models

One widely used scheme to deal with the issue of temporal dynamics is the generative models, where time series are reorganized by a sequential prototype (state). Thus, the temporal dynamics of gestures are trained as a set of transitions among these prototypes [12]. A representative work is the HMM. It can globally model the temporal evolution of gestures, which is more robust to the temporal warping of the sequence. This algorithm has been adopted in [21] [7] [22] [28] [29]. However, in HMM, the input sequences have to be previously segmented, which in itself is a challenging problem. Commonly, HMM-based methods divide each sequence into a fixed number of segments with equal-length. An example is shown in Fig. 3 (d), states $h_1^{'}, ..., h_5^{'}$ are assigned to frames from equal-sized segments correspondingly. Nevertheless, it may be hard to deal with complex gestures composed of diverse temporal durations. Another popular generative model is the conditional random field (CRF). Koppula and Saxena [23] [24] modeled the human-object interactions with a spatio-temporal CRF. However, the structure of the graphs has to be fully known, which makes this method highly dependent on the quality of annotated video data. In [25], Wu $et\ al.$ proposed a latent Dirichlet allocation (LDA) based method to model the co-occurrence and temporal relation among short actions (words) in long term videos, and the authors used a K-means based clustering to build action-words and model activities as sequences of these words. It is noted that a standard K-means segmentation approach ignored the temporal information and thus its performance is usually inferior to the traditional transition state clustering [43]. In fact, existing methods always face the same difficulty in determining the accurate states from observations without careful selection of the features, which undermines the performance of such generative models [17].

C. Approaches with recurrent neural network

Another popular technique for addressing the issue of temporal dynamics is RNN [32]–[38]. Specifically, the LSTM [44] carefully designs a suit of schemes to memorize contextual information observed from previous sequential inputs, which enables tracking the long-term temporal dependency. In [32], Du $et\ al.$ presented a bi-directional LSTMs for action recognition, where the entire skeleton was divided into five groups of joints and each group was fed into a group specific LSTM subnetwork. Then the system fused the outputs of these subnetworks hierarchically and finally fed them into another set of higher level LSTMs to represent the global body movements. Zhu $et\ al.$ [33] added a group sparse regularization term to the cost function of LSTM, which enables the network to learn the co-occurrence of discriminative skeleton joints automatically. In [36], Liu $et\ al.$ introduced a trust gate into the LSTM to learn the reliability of the inputs and accordingly adjust their confidence in updating the context information. In [34], an encoding/decoding LSTMs scheme is proposed for action recognition based on both skeletal and RGB data. The encoder is trained in an unsupervised manner on 3D skeleton sequences. Then, the manifold is used to regularize the supervised learning of decoding LSTM for RGB data based recognition. In [35], Li $et\ al.$ utilized a Gaussian-like curve to measure the confidences of the starting and ending frame of actions, and introduced a joint classification regression LSTM to solve online action detection and recognition problem. In [37], Liu $et\ al.$ proposed a global context-aware attention LSTM (GCALSTM), which aimed to handle LSTM's restriction in perceiving the global contextual information. Although LSTM is powerful in modeling sequential data, it still suffers from remembering the information of the entire sequence with many time steps (states) [31] [45]. Moreover, compared with the progress of data augmentation in RGB images, research on 3D skeleton data is still at a rather early stage. As such, it is still challenging to train the LSTM on a limited amount of data [17]. In [46], Koller $et\ al.$ embed an HMM into a deep CNN-BLSTM network for sign language recognition which is a problem closely related to temporal gesture segmentation. They firstly trained a CNN using weak frame level annotations, then used an LSTM to generate the Bayesian posteriors for HMM inferences and made use of the inferred (resulted) hidden states of each frame for CNN fine-tuning. This model is based on the hypothesis that certain boundaries can be determined by some rules in continuous sequences. Obviously, the output of temporal segment is at the "words" level but not the "phase" level with semantically meaningful and discriminative concepts. For example, this temporal boundary based segmentation may run into a stone wall when a gesture is composed of many different poses with temporal boundaries and a subject performs this gesture cyclically with different rates and orders. Besides, the number of hidden states is very hard to be determined by the "words" based model, the method in [46] utilized six hidden states empirically without clearly defined meanings.

III. HMM for Gesture Modeling

In this paper, the gesture modeling via HMM is formulated by the following definitions.

Given a set $\Theta = \{\theta_1, \theta_2, \cdots, \theta_{K-1}, \theta_K\}$ which contains K gesture sequences with arbitrary lengths. Any gesture sequence θ_k can be denoted as $\theta_k = \{f_{k,1}, f_{k,2}, \cdots, f_{k,T_k-1}, f_{k,T_k}\}$, where $f_{k,t}$ is the t^{th} frame (or representation of a frame) of θ_k, and T_k denotes its length. For any θ_k from Θ, its label δ_c satisfies $\delta_c \in \Delta$, where Δ is the set of C gesture labels which is denoted as $\Delta = \{\delta_1, \delta_2, \cdots, \delta_{C-1}, \delta_C\}$.

Specifically, given an observed gesture sequence as $X = \{x_1, x_2, \cdots x_{T-1}, x_T\}$, where $X \in \Theta$, we use the HMM to infer a hidden state sequence $H = \{h_1, h_2, \cdots h_{T-1}, h_T\}$. Any state h_t from H fulfills $h_t \in \Psi$ $(1 \leq t \leq T)$, where Ψ denotes a universal set which contains all possible Markov hidden states.

Typically, the state alignment is conducted based on a hypothesis that gestures are completed by uniformly performing Z defined hidden states in order, and hidden states from different gesture classes are not overlapping. Then, for gestures from class δ_c, given a unique hidden states set $\{\psi_{c,1}, \psi_{c,2}, \cdots, \psi_{c,Z-1}, \psi_{c,Z}\}$, we generalize this concept for all gesture classes, and define a universal set of hidden states for all gesture classes, as

$$\Psi = \left\{ \begin{array}{ccccc} \psi_{1,1} & \psi_{1,2} & \cdots & \psi_{1,Z-1} & \psi_{1,Z} \\ \psi_{2,1} & \psi_{2,2} & \cdots & \psi_{2,Z-1} & \psi_{2,Z} \\ \cdots & \cdots & \psi_{c,z} & \cdots & \cdots \\ \psi_{C-1,1} & \psi_{C-1,2} & \cdots & \psi_{C-1,Z-1} & \psi_{C-1,Z} \\ \psi_{C,1} & \psi_{C,2} & \cdots & \psi_{C,Z-1} & \psi_{C,Z} \end{array} \right\},$$

where $\psi_{c,z}$ denotes the z^{th} hidden state of gesture class δ_c, $1 \leq \delta_c \leq C$ and $1 \leq z \leq Z$. So in total there are E different states for all gestures, where $E = Z \times C$.

Thus, according to the HMM full probability model

$$P(H, X) = P(h_1)P(x_1|h_1)\prod_{t=2}^{T} P(h_t|h_{t-1})P(x_t|h_t), \quad (1)$$

where the goal of the gesture modeling problem is to find an optimal hidden state sequence $\hat{H}$ which can maximize the joint probability $P(\hat{H}, X)$, based on a given set of observations X. Because the observation X is equivalent for all hidden state combinations H, so the optimization problem for solving $\hat{H}$ could be rewritten as

$$\hat{H} = \arg\max_{H} P(H|X) \underset{X}{\propto} \arg\max_{H} P(H, X). \quad (2)$$

From the above discussion, we can conclude that HMM-based gesture recognition has two critical problems which need to be carefully solved:

- Given an observation of gesture sequence, how can a corresponding hidden state sequence be selected that is optimal in some meaningful sense to best explain the observation?
- Three sets of parameters need to be estimated to complete the specification of an HMM, namely the initial probability of the first hidden state prior $P(h_1)$, the hidden state transition probability $P(h_t|h_{t-1})$, and the emission probability $P(x_t|h_t)$ of generating an observation at time t when given the hidden state h_t. How can these parameters (distributions) be efficiently computed?

For the first problem, Wu $et\ al.$ [28] [29] employed a deep belief network (DBN) to estimate the emission probability, while the authors used a forced alignment scheme to divide video sequences temporally equal. In [7], a spherical histogram of the locations of 12 manually selected 3D skeleton joints is computed. These histograms are projected using linear discriminant analysis and then clustered into K posture words. Finally, each action is characterized as a time series of these words (hidden states). Nevertheless, this one frame one posture label (state) tactic cannot fully model the motion temporality since it ignores the contextual information. Actually, an explicit definition of hidden states of the sequences is necessary, including the number of states and the number of distinct frames per state. Although the states are hidden, for many practical applications there is often some physical significance attached to the states. For gesture recognition, the gestures themselves exhibit an internal temporal structure. As defined in Section I, gestures typically have definite gesticular phases with varying durations and starting/ending times. To illustrate this, two examples are given in Fig. 1. Based on this observation, in this paper, gestures are modeled as compositions of different gesticular phases. Once the gestures have distinct phases, models that exploit hidden states are advantageous. As such, the different gesticular phases correspond to the different hidden states of HMM and the usage of HMM allows heterogeneous information of one gesture class to be distributed over many states (phases), which is key to improve the ability to model complex patterns.

With the second issue, Gaussian mixture model (GMM) [21] [22] [42] is widely utilized as the dominant technique for estimating the emission distribution of HMM. In [28] [29], a DBN is used as a generative model to replace the traditional GMM for estimating the emission probability. However, there exists a conflict that any frame within a sequence usually has contextual information and is correlated with previous frames.

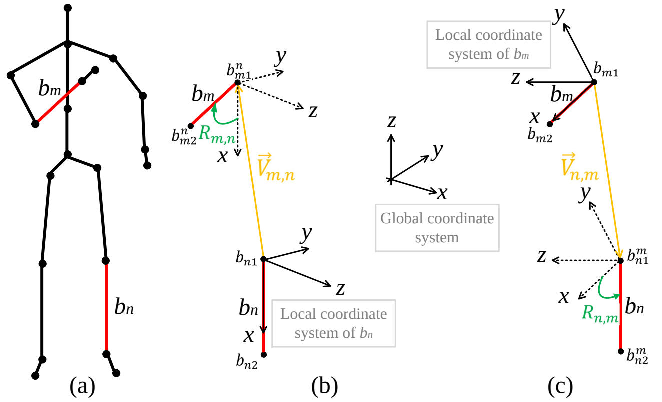


Fig. 4. (a) Illustration of a skeleton consisting of 20 joints and 19 bones, (b) Representation of bone b_m in the local coordinate system of b_n, (c) Representation of bone b_n in the local coordinate system of b_m.

Nevertheless, this is ignored in previous works. Both the DBN and GMM treat input frames at each time as independent variables so that output emission probability in the current time step only relates to the current input. To solve this issue and acquire the emission probability more appropriately, the LSTM [44] is utilized for its stronger contextual information modeling ability. As a special type of RNN, the LSTM also utilizes memory cells to store contextual information learned from previous sequential inputs and stored information can affect the output of the network.

IV. Lie Group based Skeleton Representation

One important step of modeling gesture is the choice of features to capture the variability of 3D skeletons, within and across gesture classes. In this paper, the Lie group-based representation [47] [9] is introduced. Instead of using the absolute coordinate, we utilize the relative geometry between different body parts to characterize the body configuration.

Mathematically, any rigid body displacement can be realized by a rotation about an axis combined with a translation parallel to that axis [47]. This 3D rigid body displacement forms a $SE(3)$, the special Euclidean group in three dimensions. $SE(3)$ can be identified with the space of 4×4 matrices of the form

$$P(R, \vec{v}) = \begin{bmatrix} R & \vec{v} \\ 0 & 1 \end{bmatrix}, \quad (3)$$

where $R \in SO(3)$ is a point in the special orthogonal group $SO(3)$, denotes the rotation matrix, and $\vec{v} \in \mathbb{R}^3$ denotes the translation vector.

The human skeleton can be modeled by an articulated system of rigid segments connected by joints. As such, let $S = (J, B)$ be a skeleton, where $J = \{j_1, \cdots, j_N\}$ indicates the set of body joints, and $B = \{b_1, \cdots, b_M\}$ indicates the set of body bones (oriented edges). The relative geometry between a pair of body parts (bones) can be represented as a point in $SE(3)$. More specifically, given a pair of bones b_m and b_n, their relative geometry can be represented in a local coordinate system attached to others [9]. Let $b_{i1} \in \mathbb{R}^3$, $b_{i2} \in \mathbb{R}^3$ denote the starting and end points of bones b_i respectively. The local coordinate system of bone b_n is calculated by rotating with minimum rotation and translating the global coordinate system so that b_{n1} act as the origin and b_n coincides with the $x-$axis, Fig. 4 gives an example to explain this pictorially. As such, at time t, the representation of bone b_m in the local coordinate system of b_n (Fig. 4 (b)), the starting point $b^n_{m1}(t) \in \mathbb{R}^3$ and end point $b^n_{m2}(t) \in \mathbb{R}^3$ are given by

$$\begin{bmatrix} b^n_{m1}(t) & b^n_{m2}(t) \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} R_{m,n}(t) & \vec{v}_{m,n}(t) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & l_m \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}, \tag{4}$$

where $R_{m,n}(t)$ and $\vec{v}_{m,n}(t)$ respectively denote the rotation and translation measured in the local coordinate system attached to b_n, and l_m is the length of b_m. In the same way, the representation of bone b_n in the local coordinate system of b_m can be obtained by $R_{n,m}(t)$, $\vec{v}_{n,m}(t)$, and l_n (Fig. 4 (c)). According to the theory of rigid body kinematics, the lengths of bones (body parts) do not vary with time. Therefore, the relative geometries of b_m and b_n at time t can be described by

$$\begin{aligned} P_{m,n}(t) &= \begin{bmatrix} R_{m,n}(t) & \vec{v}_{m,n}(t) \\ 0 & 1 \end{bmatrix} \in SE(3), \\ P_{n,m}(t) &= \begin{bmatrix} R_{n,m}(t) & \vec{v}_{n,m}(t) \\ 0 & 1 \end{bmatrix} \in SE(3). \end{aligned} \tag{5}$$

Let M denotes the number of bones. The resulting feature of a skeleton is interpreted by the relative geometry between all pairs of bones, as a point $C(t) = (P_{1,2}(t), P_{2,1}(t), \ldots, P_{M-1,M}(t), P_{M,M-1}(t))$ on the product space of $SE(3) \times \cdots \times SE(3)$, which is a group for the standard matrix multiplication, and that it can be endowed with a differentiable structure. It is therefore a Lie group. This relative geometry has natural stability and consistency. For example, if a pair of bones undergo the same rotation, their relative geometry matrix would not be altered. However, the Lie group is endowed with the Riemannian manifold such that standard classification and clustering algorithms are not directly applicable to this non-Euclidean space.

The tangent space of $SE(3)$ at the identity I_4 is called its Lie algebra, denoted by $\mathfrak{se}(3)$, which is isomorphic to the space of twists and therefore provides a natural setting for analyzing instantaneous motions [47]. In that way, the former classification tasks in manifold curve space are converted into the classification problems in typical vector space. The $\mathfrak{se}(3)$ can be identified with 4×4 matrices of the form

$$\widehat{\xi} = \begin{bmatrix} \widehat{\omega} & \vec{v} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 & v_1 \\ \omega_3 & 0 & -\omega_1 & v_2 \\ -\omega_2 & \omega_1 & 0 & v_3 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{6}$$

where $\widehat{\omega}$ is a 3×3 skew-symmetric matrix and can be thus identified with a vector $\omega = [\omega_1, \omega_2, \omega_3]^T \in \mathbb{R}^3$, and $\vec{v} \in \mathbb{R}^3$. In other words, each element of $\mathfrak{se}(3)$ can be identified with a vector $\xi = [\omega_1, \omega_2, \omega_3, v_1, v_2, v_3]^T \in \mathbb{R}^6$.

The logarithm map $\log P : SE(3) \to \mathfrak{se}(3)$ between the Lie group and Lie algebra [47] is given by

$$\widehat{\xi} = \log \begin{bmatrix} R & \vec{v} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \widehat{\omega} & A^{-1}\vec{v} \\ 0 & 0 \end{bmatrix}, \tag{7}$$

where $\widehat{\omega} = \log R$ and

$$A^{-1} = I - \frac{1}{2}\widehat{\omega} + \frac{2\sin\|\omega\| - \|\omega\|\left(1 + \cos\|\omega\|\right)}{2\|\omega\|^2 \sin\|\omega\|}\widehat{\omega}^2 \qquad \omega \neq 0. \tag{8}$$

If $\omega = 0$ then $A = I$. Here, since the $\log P$ is not unique, typically, the value with the smallest norm is used [9].

As a result, a skeleton can be represented by a point in the product space of the Lie group $SE(3) \times \cdots \times SE(3)$, and the number of $SE(3)$ is $2C^2_M$, where C^2_M is the combination formula. Furthermore, this $SE(3) \times \cdots \times SE(3)$ can be mapped to its Lie algebra $\mathfrak{se}(3) \times \cdots \times \mathfrak{se}(3)$ (illustrated in Fig. 2 (b)), and each $\mathfrak{se}(3)$ can be identified with a vector $[\omega_1, \omega_2, \omega_3, v_1, v_2, v_3]^T \in \mathbb{R}^6$. As such, at time t, a human skeleton G can be modeled by a $6M(M-1)$ dimensional vector, then $G \in \mathbb{R}^{6M(M-1)}$.

V. Low-rank Decomposition for Exploring Gesture Temporal Structures

In this section, we attempt to discover the temporal structures (phases) of gesture sequences and formulate a model over the temporal domain which is able to explore the hidden states of gestures.

Given an observed sequence (T frames), for a gesture performer, we can construct a matrix D by stacking (Lie algebra based) skeletal representations of every frame horizontally (column wise), then $D \in \mathbb{R}^{G \times T}$. Since the gesture's "hold" phases are with static poses, and Lie group (algebra)-based representation has properties of view-invariance and stability for dynamics, these static poses ("hold" phases) should be captured by a low-rank matrix, and hand movements (phases) means gesture changes which cannot be fitted into the low-rank model of static poses, and thus should be treated as outliers. Based on this observation, we consider the hidden states exploration from the viewpoint of a matrix decomposition and optimization problem, which can be expressed as

$$D = L + S, \tag{9}$$

where L and S denote the "hold" states (phases) and hand movement signals (phases) respectively. We assume that the static poses of "hold" states are linearly correlated with each other, forming a low-rank matrix L. Component S should be a column-block sparse matrix with non-zero columns corresponding to the outliers. In order to eliminate ambiguity, the columns of the low-rank matrix L corresponding to the outlier columns are assumed to be zeros. To formalize column-block priors on outliers, we introduce the $\ell_{2,1}$-norm and then propose a Low-rank and Column-Block sparsity matrix Decomposition (LCBD) method, as

$$\min_{L,S} \|L\|_* + \kappa\lambda\|S\|_{2,1} + \kappa\left(1 - \lambda\right)\|L\|_{2,1} \quad s.t.\ D = L + S, \tag{10}$$

where $\|L\|_*$ means the nuclear norm of matrix L, the sum of its singular values, and $\|S\|_{2,1}$ means ℓ_1-norm of the vector

Algorithm 1 Low-rank and Column-Block sparsity matrix Decomposition

```
Input: Given Matrix $D \in \mathbb{R}^{G\times T}$ and the parameters $\kappa$, $\lambda$.
Output: Estimate of $(L, S)$.
 1: Parameters initialization: $S_0 = Y_0 = 0$; $L_0 = 0$; $\mu_0 = 40/\|sign\,(D)\|_2$; $\rho > 1$; $\kappa = 0.041$; $\lambda = 0.73$; $k = 0$.
 2: While not converged do
 3:   //Line 4-11 solve $L_{k+1} = \arg\min_L f_\mu\,(L, S_k, Y_k)$, as Eq. (13).
 4:   $G^L{=}D - S_k + \mu_k^{-1}Y_k$.
 5:   $j \leftarrow 0$, $L_{k+1}^0 = G^L$.
 6:   While not converged do
 7:     $L_{k+1}^{(j+1/2)} = US_\beta\left(\sum\right)V^T$ where $L_{k+1}^{(j)} = U\sum V^T$ is the SVD of $L_{k+1}^{(j)}$.
 8:     $\Pi = \alpha_j\left(\tau_{\frac{\beta\kappa(1-\lambda)}{1+\beta\mu_k}}\left(\frac{2L_{k+1}^{(j+1/2)}-L_{k+1}^{(j)}+\beta\mu_k G^L}{1+\beta\mu_k}\right) - L_{k+1}^{(j+1/2)}\right)$
 9:     $L_{k+1}^{(j+1)} = L_{k+1}^{(j)} + \Pi$.
10:     $j \leftarrow j + 1$.
11:   end while.
12:   $L_{k+1} = L_{k+1}^{(j+1/2)}$.
13:   //Line 14-15 solve $S_{k+1} = \arg\min_S f_\mu\,(L_{k+1}, S, Y_k)$.
14:   $G^S{=}D - L_{k+1} + \mu_k^{-1}Y_k$.
15:   $S_{k+1} = \tau_{\frac{\kappa\lambda}{\mu_k}}\left(G^S\right)$.
16:   $Y_{k+1} = Y_k + \mu_k\left(D - L_{k+1} - S_{k+1}\right)$.
17:   $\mu_{k+1} = \rho\mu_k$; $k \leftarrow k + 1$.
18: end while.
19: $L \leftarrow L_k$, $S \leftarrow S_k$.
```

formed by taking the ℓ_2-norms of the columns of matrix S, as

$$\|S\|_{2,1} = \sum_{i=1}^{T}\|S_i\|_2, \tag{11}$$

where S_i denotes the i^{th} column of S.

The extra introduced term $\kappa\,(1-\lambda)\,\|L\|_{2,1}$ ensures that recovered matrix L has exact zero columns corresponding to S [48] [49] [50]. Eq. (10) is an optimization problem and we could solve it based on the augmented Lagrange multiplier (ALM) [48] [51] [52], which can be defined as

$$\begin{aligned}\mathcal{L}\,(L, S, Y; \mu) = \|L\|_* + \kappa\lambda\|S\|_{2,1} + \kappa\,(1-\lambda)\,\|L\|_{2,1} + \\ \langle Y, D - L - S\rangle + \frac{\mu}{2}\|D - L - S\|_F^2,\end{aligned} \tag{12}$$

where Y is a vector of Lagrange multipliers, μ is a positive scalar. ALM solves (12) by alternating between optimizing the prime variables L and S and updating the dual variable Y, which solves the following three sub-problems

$$\left\{\begin{array}{l} L_{k+1} = \arg\min_L \mathcal{L}_1\,(L, S_k, Y_k; \mu_k) \\ S_{k+1} = \arg\min_S \mathcal{L}_1\,(L_{k+1}, S, Y_k; \mu_k) \\ Y_{k+1} = Y_k + \mu_k\,(D - L_{k+1} - S_{k+1}) \end{array}\right. . \tag{13}$$

The first problem in (13) which solves for L at fixed S and Y, can be explicitly expressed as the following form

$$\begin{aligned}&\min_L\{\|L\|_* + \kappa\,(1-\lambda)\|L\|_{2,1} + \\ &\frac{\mu}{2}\left\|\left(D - S_k + \mu_k^{-1}Y_k\right) - L\right\|_F^2\}.\end{aligned} \tag{14}$$

In each iteration, the (14) can be rewritten as

$$\begin{aligned} L_{k+1} = \\ \arg\min_L\left\{\|L\|_* + \kappa\,(1-\lambda)\,\|L\|_{2,1} + \frac{\mu_k}{2}\left\|G^L - L\right\|_F^2\right\},\end{aligned} \tag{15}$$

where $G^L = D - S_k + \mu_k^{-1}Y_k$. We use the Douglas/ Peaceman Rachford (DR) monotone operator splitting method [53] [54] to iteratively solve (15).

Define $f_1\,(L) = \kappa\,(1-\lambda)\,\|L\|_{2,1} + \frac{\mu_k}{2}\left\|G^L - L\right\|_F^2$ and $f_2\,(L) = \|L\|_*$. For $\beta > 0$ and a sequence $\alpha_j \in (0, 2)$, the DR iteration for (15) is expressed as

$$\begin{aligned} &L^{(j+1/2)} = prox_{\beta f_2}\left(L^{(j)}\right), \\ &L^{(j+1)} = L^{(j)} + \alpha_j\left(prox_{\beta f_1}\left(2L^{(j+1/2)} - L^{(j)}\right) - L^{(j+1/2)}\right),\end{aligned} \tag{16}$$

where the two proximity operators involved in DR iteration are defined as

$$\begin{aligned} &prox_{\beta f_1}\,(L) = \tau_{\frac{\beta\kappa(1-\lambda)}{1+\beta\mu_k}}\left(\tfrac{L+\beta\mu_k G^L}{1+\beta\mu_k}\right), \\ &prox_{\beta f_2}\,(L) = US_\beta\left(\textstyle\sum\right)V^T, \\ &\tau_\eta\,(G_p) = G_p\max\left(0, 1 - \tfrac{\eta}{\|G_p\|_2}\right), p = 1, 2, ..., n \\ &S_\beta\,(x) = \max\,(0, x - \beta)\,, x \geq 0, \beta > 0.\end{aligned} \tag{17}$$

With the same idea of developing (14), the second problem in (13) can be shown as the following equivalent formula:

$$\min_S \frac{\mu_k}{2}\left\|\left(D - L_{k+1} + \mu_k^{-1}Y_k\right) - S\right\|_F^2 + \kappa\lambda\|S\|_{2,1}. \tag{18}$$

Similarly, note $G^S = D - L_k + \mu^{-1}Y_k$. Then,

$$S = \tau_{\frac{\kappa\lambda}{\mu_k}}\left(G^S\right). \tag{19}$$

The whole algorithm is summarized in Algorithm 1. In the processing of iteration, the error in outer loop is computed as $\|D - L_k - S_k\|_F/\|D\|_F$. The outer loop stops when it reaches the value lower than 10^{-7} or the maximal iteration number 500 is reached. The error in the inner loop stops when the difference between successive matrices L_k^j equals to 10^{-6} or a maximal iteration equals to 20. The tuning parameters κ and λ are set to 0.041 and 0.73, respectively. For the DR iteration, $\alpha_j \equiv 1$, and β is set to 0.57. The ALM parameter $\rho = 1.1$. The convergence is guaranteed by the ALM algorithm (we refer the interested readers to [48]).

Here we investigate the complexity of the proposed LCBD. For the optimization method shown in Algorithm 1, each outer iteration consists of three updating parts, namely the L, S, and Y. For minimizing the variable L, DR monotone operator splitting algorithm is proceeded to solve the singular value decomposition (SVD) and shrinkage operations alternately. Since the size of the matrix D is $G\times T$, according to formulae quoted by Golub and Van Loan [55], an exact SVD requires $4G^2T + 8GT^2 + 9T^3$ flops (floating-point operations), and the multiplication of the shrank singular value matrix with two singular vectors matrices which costs $(G + T)r^2$ flops [56], where r is the rank of the matrix D (commonly, r is much smaller than G and T). Also, the complexity of shrinkage operations is $O(GT)$. Therefore, to sum up, the major computations of solving L cost $O(k_1(G^2T{+}GT^2{+}T^3{+}(G{+}T)r^2{+}GT)) = O(k_1(G^2T{+}GT^2{+}T^3))$, where k_1 is the

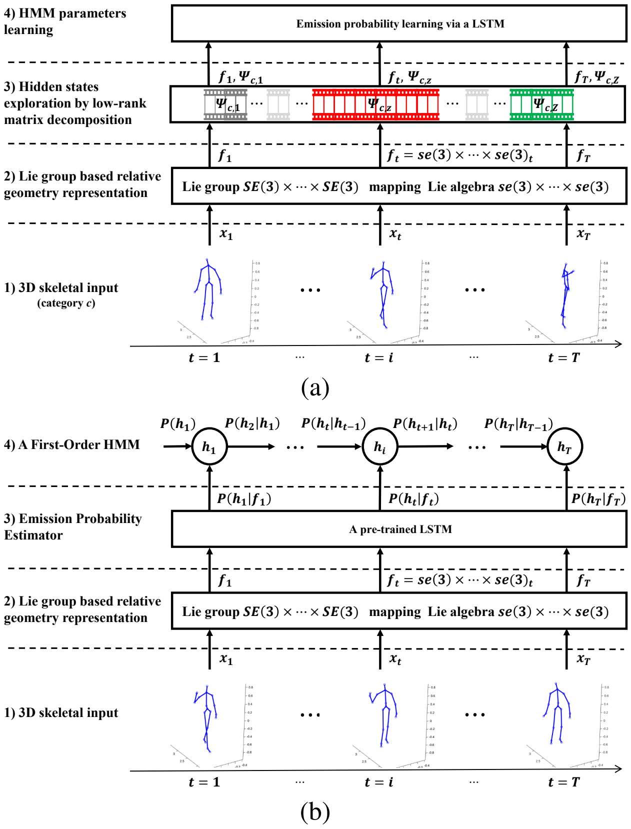

4) HMM parameters learning
Emission probability learning via a LSTM
$f_1, \Psi_{c,1}$
$f_t, \Psi_{c,z}$
$f_T, \Psi_{c,Z}$
3) Hidden states exploration by low-rank matrix decomposition
$\Psi_{c,1}$
$\Psi_{c,z}$
$\Psi_{c,Z}$
f_1
$f_t = se(3) \times \cdots \times se(3)_t$
f_T
2) Lie group based relative geometry representation
Lie group $SE(3) \times \cdots \times SE(3)$ mapping Lie algebra $se(3) \times \cdots \times se(3)$
x_1
x_t
x_T
1) 3D skeletal input (category c)
$t = 1$
$t = i$
$t = T$
(a)
4) A First-Order HMM
$P(h_1)$
$P(h_2|h_1)$
$P(h_t|h_{t-1})$
$P(h_{t+1}|h_t)$
$P(h_T|h_{T-1})$
h_1
h_i
h_T
$P(h_1|f_1)$
$P(h_t|f_t)$
$P(h_T|f_T)$
3) Emission Probability Estimator
A pre-trained LSTM
f_1
$f_t = se(3) \times \cdots \times se(3)_t$
f_T
2) Lie group based relative geometry representation
Lie group $SE(3) \times \cdots \times SE(3)$ mapping Lie algebra $se(3) \times \cdots \times se(3)$
x_1
x_t
x_T
1) 3D skeletal input
$t = 1$
$t = i$
$t = T$
(b)

Fig. 5. Illustration of the pipelines of the proposed method. (a) training pipeline, (b) testing pipeline. Please note the purpose we use f_t rather than x_t in $P(h_t|f_t)$ is to emphasize the Lie group based representation.

maximum iteration number of inner loop. Updating S involves element-wise addition and the shrinkage operations of $G \times T$ matrices, then the computational cost is $O(GT)$. Updating multiplier Y only requires element-wise addition, so its time complexity is $O(GT)$. In summary, the main computational complexity of each outer iteration of the proposed LCBD is $O(k_1(G^2T + GT^2 + T^3))$.

VI. Hidden States Learning via LSTM

As reported in the previous section, we initialize the hidden states of the temporal segments for each training sample, according to the most discriminative phases of sequences as presented in Section V. Based on these hidden states we can calculate three sets of HMM parameters in a more meaningful sense than in previous methods.

For representing the probability of the first hidden state prior, we use $\pi = (\pi_i)_{E\times 1}$, where $\pi_i = P(h_1 = \psi_i)$, and ψ_i is the i^{th} state of hidden states set Ψ. As defined in Section III, E is the number of hidden state types for all gestures. Then we can estimate π_i by calculating

$$\pi_i = \frac{\sum_{k=1}^{K}(h_{k,1} == \psi_i)}{K}, \tag{20}$$

where k denotes the index of an observation, and K is the total number of observations (gesture sequences).

Next, the hidden states transition parameter (matrix) is denoted as $A = [a_{i,j}]_{E\times E}$, where $a_{i,j} = P(h_t = \psi_j | h_{t-1} = \psi_i)$. We can calculate $a_{i,j}$ by

$$a_{i,j} = \frac{\sum_{k=1}^{K}\sum_{t=2}^{T_k}((h_{k,t-1} == \psi_i)\textbf{AND}(h_{k,t} == \psi_j))}{\sum_{k=1}^{K}\sum_{t=2}^{T_k}(h_{k,t-1} == \psi_i)}. \tag{21}$$

Then, for representing the probability of the hidden state prior $P(h_t)$, we use $\varrho = (\varrho_i)_{E\times 1}$, where $\varrho_i = P(h_t = \psi_i)$, and ψ_i is the i^{th} state of hidden states set Ψ. Then we can estimate ϱ_i by calculating

$$\varrho_i = \frac{\sum_{k=1}^{K}(h_{k,t} == \psi_i)}{\Pi}, \tag{22}$$

where k denotes the index of an observation, and Π is the total number of frames of all observations.

Another important parameter is the emission probability. Compared to DBN and GMM models are widely used in previous methods, LSTM can learn the contextual information from sequential data, which provides a powerful capability for sequential data modeling. On one hand, it receives the output from the previous one step and uses it as a part of the input in the current time step. On the other hand, it uses memory cells to store contextual information learned from the input and uses gate units to maintain the stored contextual information.

In order to ensure that the LSTM generates outputs in the form of the emission probability $P(x_t|h_t)$, we use a softmax loss function to train the network. It can instruct the LSTM network to generate a posterior distribution $P(h_t|x_t, \zeta)$, where ζ is the network parameter shared by all time steps. Thus, we can use such network to infer the emission probability by

$$P(x_t|h_t) = \frac{P(h_t|x_t)P(x_t)}{P(h_t)} \underset{x_t}{\propto} \frac{P(h_t|x_t)}{P(h_t)}. \tag{23}$$

Lastly, by combining (1), (2) and (23), we can arrive at our final objective function as follows

$$\hat{H} = \arg\max_{H} P(h_1|x_1) \prod_{t=2}^{T} P(h_t|h_{t-1}) \frac{P(h_t|x_t)}{P(h_t)}, \tag{24}$$

where $\hat{H}$ denotes the optimal hidden state sequence. The optimization problem of (24) can be easily solved by Viterbi path decoding [42]. The pipelines (training and testing) of the proposed method are shown in Fig. 5. In the training pipeline (see Fig. 5 (a)), the input of LSTM is the Lie group based representation f_t of frame x_t (an 3D skeleton), and its label is a hidden state $\psi_{c,z}$ which is obtained by the proposed LCBD method, where the subscript c is the gesture category and z is the hidden state index. The purpose of training is to force the LSTM to generate the posterior probabilities for modelling the HMM emission probabilities. More specifically, in the testing (see Fig. 5 (b)), given a new observation x_t (Lie algebra feature f_t), the pre-trained LSTM can yield posterior probabilities $P(h_t|x_t)$, which are needed for Eq. (24). Finally, a brief block diagram of the proposed system is summarized in Fig. 6.

It is noted that LSTM based methods commonly feed the network with a whole gesture or action sequence (frames with the same label). Although LSTMs are designed to learn the long-term temporal dependency, it is still challenging for

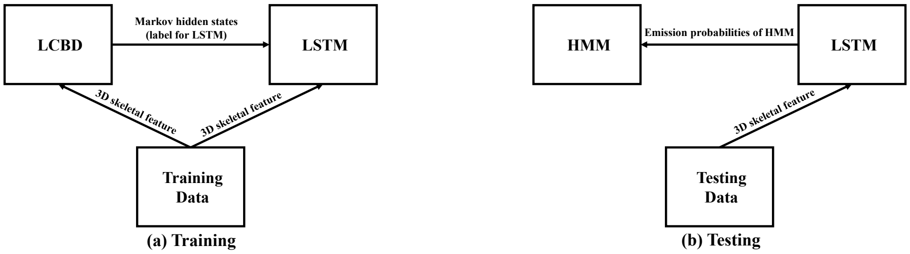


Fig. 6. Block diagram of the proposed method.

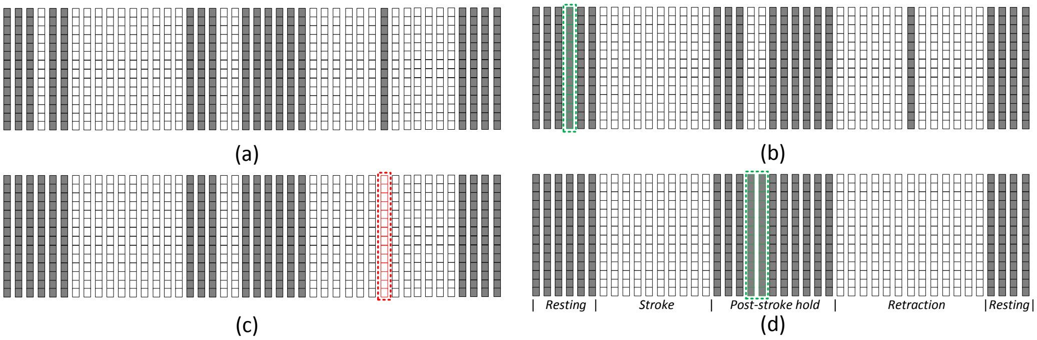

(a)
(b)
(c)
| Resting | Stroke | Post-stroke hold | Retraction |Resting|
(d)

Fig. 7. Illustration of arrangement of matrix L. (a) the original L with many chunks (continuous frames with same property), the columns in gray are the low-rank part, and the white columns denote the non-low-rank part, (b) "dilation" operation with interval threshold 2 to enlarge the boundaries of chunks, (c) "erosion" operation with length threshold 2 to erode away the isolated small chunks, (d) "dilation" operation with interval threshold 3.

LSTM to memorize the information of the entire sequence with many states [31] [45]. In addition, with a limited amount of training data, training an LSTM is prone to overfitting [18]. In our scheme, the shorter video segments (states) are fed into the network to bypass the difficulty of LSTM when modeling long-term gestures with temporal dynamics. Furthermore, this states-based feeding enlarges the number of training samples but without any data augmentation operations. Experiments demonstrate that our method outperforms LSTM with the simple mode of feeding the whole sequence.

VII. Experiments

In this section, a series of experiments are performed to evaluate the proposed approach. Four benchmark datasets, ChaLearn 2014 gesture [3], MSR Action3D [57], UTKinect-Action3D [7], and SBU Kinect interaction [58] datasets are used for evaluation purposes.

A. Implementation Details and Settings

In the proposed method, the emission probability is estimated by an RNN with four layers which are connected in the following order: one LSTM layer with 512 units, a fully connected layer with 256 neurons, a dropout layer with the dropout ratio of 50%, and a softmax loss layer to force the network to generate the likelihood $P(h_t|x_t, \zeta)$. When training the network, we set the batch size to 400. The learning rate is fixed to 0.01 for the ChaLearn 2014 gesture dataset, and 0.002 for MSR Action3D, UTKinect-Action3D and SBU Kinect interaction datasets. The network is trained until the validation accuracy and the loss are stable after a number of iterations depending on the size of training data. We set 70 as the max training epoch for the ChaLearn 2014 gesture dataset due to its large training data size. For the MSR Action3D, UTKinect-Action3D and SBU Kinect interaction datasets, the training epoch is set to 200.

An important parameter of the proposed method is Z, the number of hidden states. As mentioned in Section I, a gesture is typically composed by five phases [5] [6], as such, the Z is set to 5. The outputs (matrix L) of matrix decomposition may not always in the form of 5 chunks (continuous frames with the same property) because the low-rank matrix decomposition is an unsupervised method, another reason is the disturbances from noises and misalignments of the skeleton. Therefore, we need a post-processing step to arrange the matrix L which ensures that L can be segmented into 5 chunks (an example is illustrated in Fig. 7 (a)). Based on the phase definition of the gesture [5] [6], we can conclude that both the beginning and ending phases are *Resting*. Thus, the first and last several frames (at least the first and the last frame) of each gesture sequence should be low-rank parts (the columns in gray colors, as illustrated in Fig. 7 (a)) of the matrix L, and they could be initialized as *Resting* phases. Next, the longest low-rank chunk (except two *Resting* phases) of matrix L is selected to initialize the *Post-stroke hold* phase. The final arrangement should make matrix L to have three low-rank chunks (two *Resting* phases and a *Post-stroke hold* phase) and two non-low-rank ones (*Stroke* and *Retraction* phases), as an example shown in Fig. 7 (d). After the initialization of phases, there will be two particular situations if the number of chunks is smaller than 5, namely, there is only one chunk when all columns of matrix L are low-rank (or non-low-rank), or there will be two low-rank chunks (*Resting* phases) and a non-low-rank chunk in matrix L. For the above two cases, an equal division scheme can be utilized to obtain 5 phases (chunks). But in most cases, the number of chunks is greater than 5. In this paper, inspired by the morphology processing of image binarization, for a low-rank chunk, an operation similar to "dilation" is adopted to enlarge its boundaries through merging the adjacent low-rank chunks, if the interval among them is smaller than a given threshold. The merged chunk will repeat this operation until the above condition cannot be satisfied. The "dilation" is operated only on three chunks corresponding to the initialized low-rank phases. Also, the "erosion" operation is utilized to erode away the isolated small chunks whose length is smaller than a given threshold. In the proposed method, we employ an iterative process to perform the arrangement. More specifically, a "dilation" and an "erosion" operation are executed successively in each iteration until the number of chunks (phases) is equal to five, and we increase the thresholds of interval and chunk length after each iteration. The starting thresholds and step size are set to 2 and 1, respectively.

The effectiveness of the proposed method is compared to eighteen state-of-the-art approaches, which are simply divided into three groups.

The first group's methods are the most related to our model, including four HMM related methods, namely HMM with GMM (HMM-GMM) [42], HMM with AdaBoost (HMM-AdaBoost) [21], HMM with DBN (HMM-DBN) [28] and its extension (HMM-DBN-ext) [29].

The methods selected as the second group are based on classic feature representations, including histogram of 3D joints (HOJ3D) [7], EigenJoints [8], actionlet ensemble (Actionlet) [17] [18], histogram of oriented 4D normals (HON4D) [14], discriminative key-frames (Key-frames) [13], Lie group [9], Riemannian manifold (Manifold) [11], rotation and relative velocity with DTW (RVV+DTW) [16], latent max-margin multitask learning (LM^{3}TL) [27], and spatio-temporal naive-bayes nearest-neighbor (ST-NBNN) [26]. The last group includes four deep neural networks, namely the convolutional neural network based ModDrop (CNN) [30], LSTM [44], hierarchical recurrent neural network (HBRNN) [32], and spatio-temporal LSTM with trust gates (ST-LSTM-TG) [36]. The baseline results are reported in the original papers. Note that some of the compared methods were developed for multi-modal data such as the HMM-DBN-ext [29] which utilized RGB frames and skeletons, while the proposed method only uses 3D skeleton data.

For the sake of better understanding the performance of the proposed algorithm, we analyze the contributions of each component (ablation study) to the final performance of the system. Firstly, to clarify how much of the improvement in the results is coming from Lie group based representation, the raw data should be fed to the proposed HSL-LSTM. Here, instead of using the original skeleton data (absolute location of the performer), all 3D joint coordinates are simply transformed from the global coordinate system to a person-centric coordinate system by placing the hip center at the origin (person-centric). Next, in order to verify the effectiveness of the hidden states partitioning via proposed low-rank and column block sparsity decomposition (LCBD), we compared its performance with the equal-length division (fixed anchors).

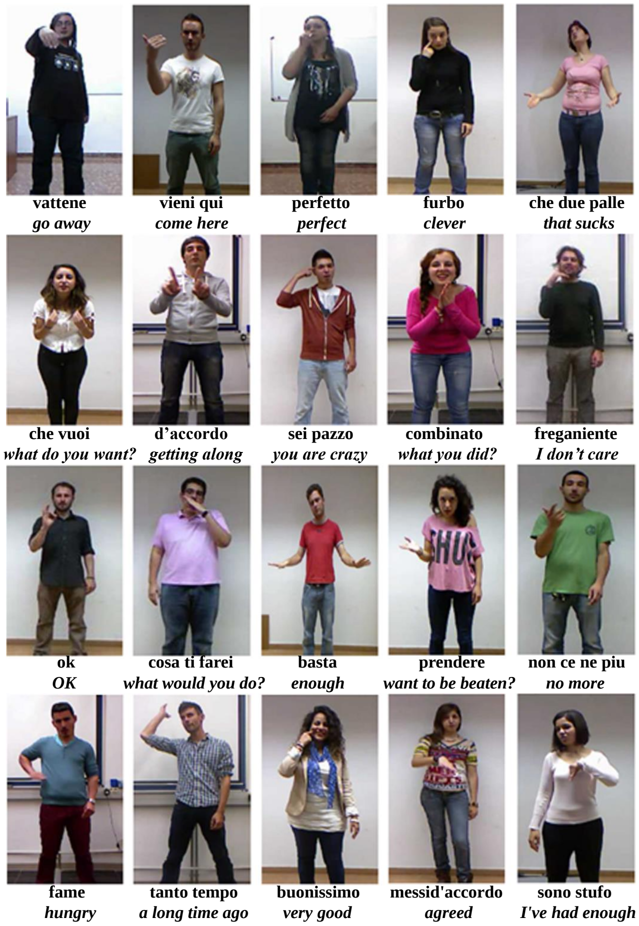
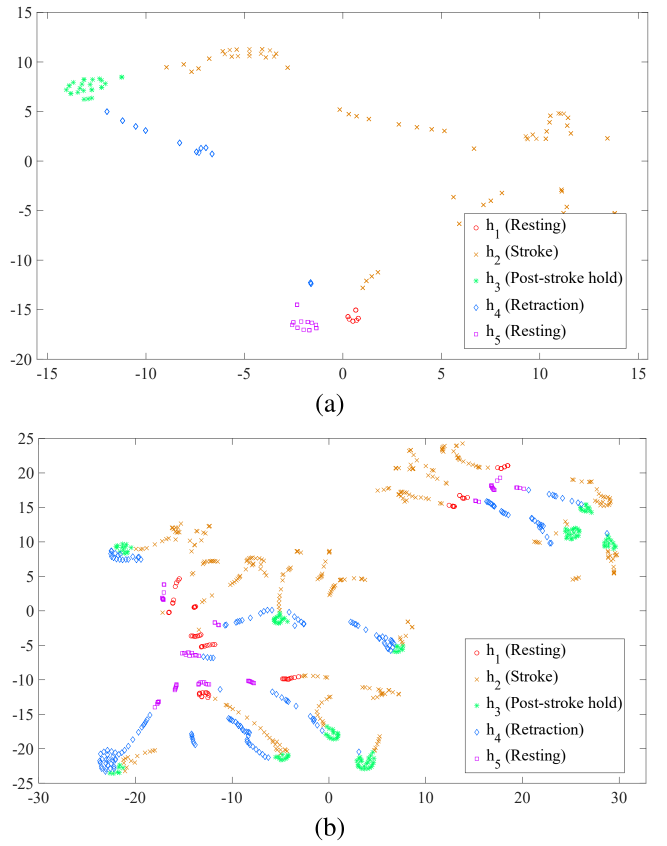


Fig. 8. Example RGB frames sampled from 20 gesture classes of the ChaLearn 2014 [3] dataset. The meanings of these gestures in Italian and English (in italics) are given.

TABLE I
COMPARISON OF RECOGNITION ACCURACY (%) WITH EXISTING SKELETON-BASED METHODS ON CHALEARN 2014 [3] DATASET (BEST: BOLD, SECOND BEST: UNDERLINE).

Methods	Accuracy
HMM-GMM [42]	49.1
HMM-DBN [28]	83.6
HMM-DBN-ext [29]*	86.4
EigenJoints [8]	59.3
Lie group [9]	79.2
ModDrop (CNN) [30]*	93.1
LSTM [44]	82.0
Ours (person-centric + fixed anchors)	87.7
Ours (Lie group + fixed anchors)	89.1
Ours (person-centric + LCBD)	93.2
Ours (Lie group + LCBD)	**93.8**

* The methods use skeleton and RGB-D data.



(a)

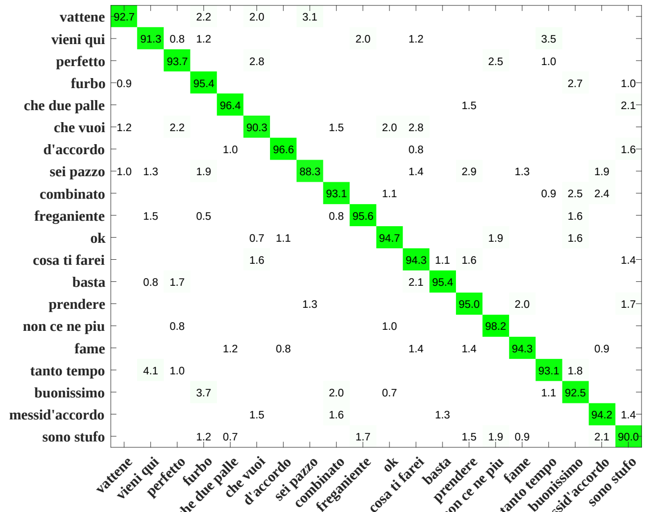
(b)

Fig. 9. Visualization results (by t-SNE) of gesture data (ChaLearn 2014 [3] dataset) with obtained hidden states. (a) a gesture sequence. (b) 10 randomly selected sequences from a category of the *basta*.

B. ChaLearn 2014 gesture Dataset

The ChaLearn 2014 is a gesture dataset of Looking At People (LAP) challenge [3] with multi-modality, including data of RGB frames, depth maps, user body masks, and 3D skeletal joint positions. This dataset collects 940 videos and each one contains 10 to 20 Italian cultural gesture instances. In total, there are 13,585 gesture instances from 20 classes. Fig. 8 gives sampled frames from each gesture class.

To illustrate the hidden states (temporal structure) of a gesture are discriminative and testify the efficient of the proposed LCBD, we visualize the gesture data (Lie algebra feature) with the obtained hidden states (by LCBD) in a 2D feature space. As reported in Section IV, the utilized Lie algebra is a high dimensional feature vector ($6M(M-1)$). The t-Distributed Stochastic Neighbor Embedding (t-SNE) [59] is employed since it is an algorithm for dimensionality reduction that is well-suited to visualizing high-dimensional data [59]. In the experiments, a gesture instance is selected randomly from ChaLearn 2014 [3] dataset, as shown in Fig. 9 (a), the visualization result of this gesture sequence has verified that the data of three "hold" phases (hidden states) h_1 (*Resting*), h_3 (*Post-stroke hold*), and h_5 (*Resting*) are clusters in the 2D feature space. It is noted that the h_1 and h_5 are close to each other, this is because both of them are *Resting* (states) performed by a subject which have similar appearances (static resting poses). In contrast to three "hold" states, h_2 (*Stroke*) and h_4 (*Retraction*) are hand movements (phases) which represent gesture changes, it can be seen from Fig. 9 (a), their data are dispersed in the 2D feature space. Furthermore, 10 sequences are selected in a random manner from a gesture category of *basta* (enough), their visualization results are illustrated in Fig. 9 (b). Obviously, all sequences have clear clusters of h_1 (*Resting*), h_3 (*Post-stroke hold*), and h_5 (*Resting*), which can prove the discrimination of the gesture again. Please note several sequences' h_3 are close to each other, as shown in the upper-right corner of Fig. 9 (b). It can be explained as a bigger cluster since they are performed in a similar pose (*Post-stroke hold*) to express the same gesture (category).



	vattene	vieni qui	perfetto	furbo	che due palle	che vuoi	d'accordo	sei pazzo	combinato	freganiente	ok	cosa ti farei	basta	prendere	non ce ne piu	fame	tanto tempo	buonissimo	messid'accordo	sono stufo
vattene	92.7			2.2		2.0		3.1												
vieni qui		91.3	0.8	1.2						2.0		1.2					3.5			
perfetto			93.7			2.8									2.5		1.0			
furbo	0.9			95.4														2.7		1.0
che due palle					96.4									1.5						2.1
che vuoi	1.2		2.2			90.3			1.5		2.0	2.8								
d'accordo					1.0		96.6					0.8								1.6
sei pazzo	1.0	1.3		1.9				88.3				1.4		2.9		1.3			1.9	
combinato									93.1		1.1						0.9	2.5	2.4	
freganiente		1.5		0.5					0.8	95.6								1.6		
ok						0.7	1.1				94.7				1.9			1.6		
cosa ti farei						1.6						94.3	1.1	1.6						1.4
basta		0.8	1.7									2.1	95.4							
prendere								1.3						95.0		2.0				1.7
non ce ne piu			0.8								1.0				98.2					
fame					1.2		0.8					1.4		1.4		94.3			0.9	
tanto tempo		4.1	1.0														93.1	1.8		
buonissimo				3.7					2.0		0.7						1.1	92.5		
messid'accordo						1.5			1.6				1.3						94.2	1.4
sono stufo				1.2	0.7					1.7				1.5	1.9	0.9			2.1	90.0

Fig. 10. Confusion matrix of the proposed method on ChaLearn 2014 [3] gesture dataset.

For quantitative evaluations, we use the protocol provided by the dataset which assigns fixed 7,754 gesture sequences for training, 3,362 sequences for validating, and 2,742 sequences for testing. It is noted that the Jaccard index score recommended by the publisher of Chalearn 2014 dataset is a frame-level metric. However, the proposed method is a sequential based model, so the Jaccard index score is not suitable to use as a metric in this study. As shown in Table I, all the results reported are in accuracy, which makes a fair comparison. To verify the effectiveness of the hidden states exploration, we compared the proposed method with three HMM-based state of the arts. It can be seen that the recognition accuracies of HMM with GMM [42] and with DBN (HMM-DBN) [28] are only 49.1% and 83.6%. This is mainly because both of the DBN and GMM treat input frames at each time step as the independent variable, thus the contextual information is ignored when learning the emission probability. The HMM-DBN-ext [29] can reach up to 86.4%, while it used skeleton, RGB, and depth information. It also can be observed that the accuracy of the LSTM [44] is 5.7 percent less than the proposed method with person-centric data input and fixed anchors division. As discussed in the introduction, LSTM is designed to explore the long-term temporal dependency, but it is still challenging for LSTM to memorize the information of the entire sequence with many states [31] [45]. Moreover, with a limited amount of training data, training an LSTM is prone to overfitting [18]. In the proposed method, shorter gesture segments (states) are fed into the network to bypass the difficulty of LSTM when modeling multi-states gestures with temporal dynamics. Furthermore, this states-based feeding enlarges the number of training samples but without any data augmentation operations. Take the Chalern 2014 dataset for example, we obtained 38,770 gesture (hidden states) segments for training, which is five times more training samples than LSTM with raw (7,754) gesture sequences. The method in [9] utilized the same Lie group to represent the 3D skeletons as ours, and it employed the DTW to deal with the temporal dynamics issue. However, DTW cannot globally capture the temporal evolution of whole sequences, so its performance is inferior to the proposed. Through the ablation study of the proposed, we can see that the results improved only 1.4 and 0.6 percent by using Lie group representation. Obviously, the main improvements in the results are coming from better partitioning of the sequences, namely the proposed LCBD. It is notable that the ModDrop [30] was the winner of the 2014 LAP Challenge (track 3). The proposed method can achieve superior performance to ModDrop even without using the RGB-D data.

Next, in order to present the accuracy of the proposed method on individual gestures, the confusion matrix is shown in Fig. 10. As can be seen, the proposed method achieves high accuracies for most of the gesture categories. There are a few confusions between similar gestures with very small values, such as the *tanto tempo* and *vieni qui*, and also in the case of *furbo* and *buonissimo*.

C. MSR Action3D Dataset

The MSR Action3D [57] is a commonly used actions recognition dataset, especially for evaluating the effectiveness of temporal dynamics modeling techniques, since this dataset is challenging where actions are highly similar to each other and have typical large temporal misalignments. MSR Action3D dataset comprises of 567 pre-segmented action instances. There are 10 subjects performing 20 classes of actions. This dataset has attracted lots of attention and many researchers have reported their results on it. For a fair comparison, the same evaluation protocol, namely the cross-subject test as described in [57] is followed, where half of the subjects are used for training (subjects number 1, 3, 5, 7, 9) and the remainder for testing (2, 4, 6, 8, 10).

The recognition accuracies are presented in Table II. It can be seen that the proposed (with LCBD) achieve better performance than DTW-based recognition approaches, such as Lie group [9] and RVV+DTW [16]. In [13], the authors emphasized the importance of discriminative key-frames for action recognition. However, the key-frames selection itself is a difficult task, which usually suffers from an issue of information losing. The HMM-DBN [28] employed a deep neural network to learn the parameters of HMM, while it utilized the fixed anchors for obtaining the hidden states. On the contrary, we formulate a model over the temporal domain that is able to capture the static poses between sub-gestures, therefore, a gesture sequence could be segmented into temporal compositions (states) with semantically meaningful and discriminative concepts. Compared with HMM-DBN, the experimental results on MSR Action3D dataset verifies the effectiveness of the proposed method again. In the ablation study of the proposed method, the person-centric inputs with fixed anchors hidden states setting can yield a better result than LSTM and HMM-based methods. As can be seen, the main improvements in the results are coming from better hidden states settings by LCBD, rather than the Lie group features. Actually, in all of the 16 methods, our models with LCBD achieve the highest recognition accuracies.

TABLE II
COMPARISON OF RECOGNITION ACCURACY (%) WITH EXISTING SKELETON-BASED METHODS ON MSR ACTION3D [57] DATASET (BEST: BOLD, SECOND BEST: UNDERLINE).

Methods	Accuracy
HMM-AdaBoost [21]	63.0
HMM-GMM [42]	81.5
HMM-DBN [28]	82.0
EigenJoints [8]	82.3
Actionlet [17] [18]*	88.2
HOJ3D [7]	78.9
HON4D [14]*	88.9
Key-frames [13]	91.7
Lie group [9]	92.5
Manifold [11]	92.1
RVV+DTW [16]	93.4
LM^{3}TL [27]	95.6
ST-NBNN [26]	94.8
LSTM [44]	88.9
HBRNN [32]	94.5
ST-LSTM-TG [36]	94.8
Ours (person-centric + fixed anchors)	92.6
Ours (Lie group + fixed anchors)	93.7
Ours (person-centric + LCBD)	95.9
Ours (Lie group + LCBD)	**96.3**

* The methods use skeleton and RGB-D data.

In Fig. 11, we report the accuracy of each action in the form of a confusion matrix. It can be found that the proposed method works very well on the MSR Action3D dataset, whereas the performances on some actions still need to improve, such as the *hand catch* and *hammer*. The classification errors occur if the way of performing these two actions varies a lot by different subjects between the training and test sets. As can be observed, the most confused actions are between *bend* and *pick up & throw*. This can be explained by the fact that the occlusion is so large that the skeleton tracker fails frequently in these actions.

D. UTKinect-Action3D and SBU Kinect Interaction Datasets

The experimental results show that the proposed method is effective and achieves state-of-the-art performance on gesture recognition. In order to better understand our performance on action recognition, we also test the proposed method on two popular skeleton-based datasets commonly used in the action recognition literature, namely the UTKinect-Action3D

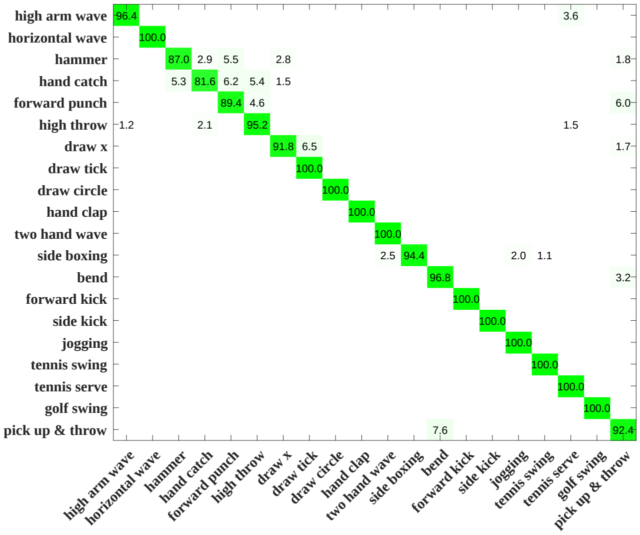


Fig. 11. Confusion matrix of the proposed method on MSR Action3D [57] dataset.

[7] and SBU Kinect interaction dataset [58]. The UTKinect-Action3D is a difficult benchmark due to its high intra-class variations. This dataset collects 10 types of actions using the Kinect. Each action is performed by 10 subjects for two times. As a result, a totally 200 action instances are collected in 20 video sequences. We follow [7] and use the *Leave-One-Sequence-Out Cross Validation* setting which selects each sequence as the testing sample in turn, regards others as training samples and calculates the average (20 rounds of testing) recognition rate. The SBU-Kinect interaction dataset contains 8 types of two-person interactions. We perform 5-fold cross validation on this dataset by using the same testing protocol as in [58]. In the process of experiments, we noticed that the low-rank assumption on video sequences from these two datasets cannot always be satisfied. This is because many videos contain person interactions, compared to regular gesture recognition, those (mutual) action sequences don't have distinct temporal structures that can be segmented by the low-rank matrix decomposition. To complete the experiments, we use fixed anchors to obtain hidden states. We summarize the classification accuracy results in Table III. It can be seen on both of two datasets, the proposed method can yield the superior results to HMM-based algorithms and LSTM [44], and achieve state-of-the-art performance.

E. Ablation study of the hidden states number

The number of hidden states Z is a critical parameter of the proposed method, which is set to 5 based on the assumption that regular gesture sequences have five distinct phases. To clarify the effect of the number of hidden states to the performance of the proposed method, an ablation study of Z is carried out. As such, the datasets of ChaLearn 2014 [3] with regular gestures and SBU Kinect interaction [58] with interactive actions are selected to provide different forms (activities) of testing. For the SBU Kinect interaction dataset, we follow to use the fixed anchors (different amounts) to obtain the hidden states. To acquire various amounts (except 5) of hidden states on ChaLearn 2014 dataset, the states of h_2 (*Stroke*), h_3 (*Post-stroke hold*), and h_4 (*Retraction*) (yielded by LCBD) are merged into a "new-state". Then, this "new-state" is divided equally to get the wanted quantity of hidden states. For example, the "new-state" could be divided into two parts (states) equally, then plus two *Resting* states h_1 and h_5 to achieve $Z = 4$. We report the recognition accuracies in Table IV with Z range from 3 to 7. It can be seen on both of two datasets the scores will drop down when the number of hidden states becomes smaller than 5. On ChaLearn 2014 dataset, although we increased Z, the recognition rates are not better than five hidden states setting. Different from the performance on ChaLearn 2014 dataset, the SBU Kinect interaction dataset with $Z = 6$ is slightly superior to $Z = 5$, this is because those complicated interactive activities may contain more temporal phases than regular gestures. It is noted that the classification accuracies are getting worse when keep increasing Z. It can be concluded that the setting of the number of hidden states should match or close to the real temporal phases of sequences.

TABLE III
COMPARISON OF RECOGNITION ACCURACY (%) WITH EXISTING SKELETON-BASED METHODS ON UTKINECT-ACTION3D [7] (UTK) [7] AND SBU KINECT INTERACTION (SBU) [58] DATASETS (BEST: BOLD, SECOND BEST: UNDERLINE).

Methods	Accuracy	
	UTK	SBU
HMM-GMM [42]	84.4	71.9
HMM-DBN [28]	93.7	89.4
EigenJoints [8]	92.4	-
HOJ3D [7]	90.9	-
HON4D [14]*	90.9	-
Lie group [9]	97.1	-
Manifold [11]	91.5	-
LM^{3}TL [27]	**98.8**	-
LSTM [44]	72.7	86.0
HBRNN [32]	-	80.4
ST-LSTM-TG [36]	97.0	$\underline{93.3}$
ST-NBNN [26]	98.0	-
Ours (person-centric + fixed anchors)	96.3	91.2
Ours (Lie group + fixed anchors)	$\underline{98.5}$	**93.5**

* The method use skeleton and RGB-D data.

TABLE IV
ABLATION STUDY OF HIDDEN STATES NUMBER (Z) ON CHALEARN 2014 (CHA) [3] AND SBU KINECT INTERACTION (SBU) [58] DATASETS (BEST: BOLD, SECOND BEST: UNDERLINE) (%).

Z	Accuracy	
	CHA	SBU
3	86.2	87.5
4	91.9	91.8
5	**93.8**	$\underline{93.5}$
6	$\underline{93.4}$	**93.7**
7	93.2	93.3

VIII. CONCLUSION

In the study of human movement, a gesture could be explained as a sequence of separated sub-gestures or phases, each of which is associated with a video segment of unfixed length. Based on that observation, this paper focuses on studying HMM-based approaches to explore more appropriate hidden states alignment of skeletal gesture data. We propose a novel skeleton based recognition framework that integrates the powers of the generative models (HMM) and deep recurrent neural network (LSTM).

As discussed in Section VII, the proposed Low-rank decomposition model is still hard to handle interactive activities, such as sequences from the SBU Kinetic dataset. However, many of them indeed have temporal structures. For example, they can be simply segmented as "two persons are approaching to each other" and "two persons have a contact". We believe information or priors like these would be beneficial to the task of action/gesture analysis. Therefore, in future research, how to separate temporal structures of those interactive actions robustly and make divisions interpretable is an interesting topic for us. Another possible directions for future work include studying the embedding problem of the Lie group for 3D human behavior analysis. Typically, the embedding is obtained by flattening the manifold via tangent spaces, such as the Lie algebra. However, in that way, only distances between points to the tangent pole are equal to true geodesic distances, which may lead to an inaccurate modeling issue. As such, a novel embedding method will be explored to keep the estimation of the distances is performed in the framework of Riemannian computing.

References

[1] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Commun. ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[2] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with Microsoft Kinect sensor: A review," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1318–1334, 2013.

[3] S. Escalera, X. Baró, J. Gonzalez, M. A. Bautista, M. Madadi, M. Reyes, V. Ponce-López, H. J. Escalante, J. Shotton, and I. Guyon, "Chalearn looking at people challenge 2014: Dataset and results," in *Proc. Eur. Conf. Comput. Vis. Workshops*. Springer, 2014, pp. 459–473.

[4] X. Liu, H. Shi, X. Hong, H. Chen, D. Tao, and G. Zhao, "Hidden states exploration for 3D skeleton-based gesture recognition," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.* IEEE, 2019, pp. 1846–1855.

[5] A. Kendon, "Gesticulation and speech: Two aspects of the process of utterance," *The relationship of verbal and nonverbal communication*, vol. 25, no. 1980, pp. 207–227, 1980.

[6] S. Kita, I. Van Gijn, and H. Van der Hulst, "Movement phases in signs and co-speech gestures, and their transcription by human coders," in *International Gesture Workshop*. Springer, 1997, pp. 23–35.

[7] L. Xia, C. C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3D joints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*. IEEE, 2012, pp. 20–27.

[8] X. Yang and Y. Tian, "Eigenjoints-based action recognition using naive-bayes-nearest-neighbor," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*. IEEE, 2012, pp. 14–19.

[9] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a Lie group," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* IEEE, 2014, pp. 588–595.

[10] D. Gong, G. Medioni, and X. Zhao, "Structured time series analysis for human action segmentation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1414–1427, 2014.

[11] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo, "3D human action recognition by shape analysis of motion trajectories on Riemannian manifold," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1340–1352, 2015.

[12] B. B. Amor, J. Su, and A. Srivastava, "Action recognition using rate-invariant analysis of skeletal shape trajectories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 1–13, 2016.

[13] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The moving pose: An efficient 3D kinematics descriptor for low-latency action recognition and detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2752–2759.

[14] O. Oreifej and Z. Liu, "HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 716–723.

[15] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition," *J Vis. Commun. Image Represent.*, vol. 25, no. 1, pp. 24–38, 2014.

[16] Y. Guo, Y. Li, and Z. Shao, "RRV: A spatiotemporal descriptor for rigid body motion recognition," *IEEE Trans. Cybern.*, 2017.

[17] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* IEEE, 2012, pp. 1290–1297.

[18] ——, "Learning actionlet ensemble for 3D human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 914–927, 2014.

[19] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from RBGD images," in *Proc. IEEE Conf. Robot. Autom.* IEEE, 2012, pp. 842–849.

[20] B. Packer, K. Saenko, and D. Koller, "A combined pose, object, and feature model for action understanding." in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1378–1385.

[21] F. Lv and R. Nevatia, "Recognition and segmentation of 3D human action using HMM and multi-class adaboost," *Proc. Eur. Conf. Comput. Vis.*, pp. 359–372, 2006.

[22] L. Piyathilaka and S. Kodagoda, "Gaussian mixture based HMM for human daily activity recognition using 3D skeleton features," in *Proc. IEEE Conf. Ind. Electron. Appl.* IEEE, 2013, pp. 567–572.

[23] H. Koppula and A. Saxena, "Learning spatio-temporal structure from RGB-D videos for human activity detection and anticipation," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 792–800.

[24] ——, "Anticipating human activities using object affordances for reactive robotic response," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 14–29, 2016.

[25] C. Wu, J. Zhang, S. Savarese, and A. Saxena, "Watch-n-patch: Unsupervised understanding of actions and relations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4362–4370.

[26] J. Weng, C. Weng, and J. Yuan, "Spatio-temporal naive-bayes nearest-neighbor (ST-NBNN) for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017.

[27] Y. Yang, C. Deng, D. Tao, S. Zhang, W. Liu, and X. Gao, "Latent max-margin multitask learning with skelets for 3-D action recognition," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 439–448, 2017.

[28] D. Wu and L. Shao, "Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* IEEE, 2014, pp. 724–731.

[29] D. Wu, L. Pigou, P. J. Kindermans, N. Le, L. Shao, J. Dambre, and J. M. Odobez, "Deep dynamic neural networks for multimodal gesture segmentation and recognition." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1583–1597, 2016.

[30] N. Neverova, C. Wolf, G. Taylor, and F. Nebout, "ModDrop: Adaptive multi-modal gesture recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1692–1706, 2016.

[31] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017.

[32] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* IEEE, 2015, pp. 1110–1118.

[33] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, X. Xie *et al.*, "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks." in *Proc. AAAI Conf. Artif. Intell.*, vol. 2, 2016, p. 8.

[34] B. Mahasseni and S. Todorovic, "Regularizing long short term memory with 3D human-skeleton sequences for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* IEEE, June 2016, pp. 3054–3062.

[35] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, and J. Liu, "Online human action detection using joint classification-regression recurrent neural networks," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 203–220.

[36] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 816–833.

[37] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention LSTM networks for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1647–1656.

[38] D. Konstantinidis, K. Dimitropoulos, and P. Daras, “Skeleton-based action recognition based on deep learning and Grassmannian pyramids,” in *26th Proc Eur Signal Process Conf EUSIPCO*. IEEE, 2018, pp. 2045–2049.

[39] L. L. Presti and M. La Cascia, “3D skeleton-based human action classification: A survey,” *Pattern Recognit.*, vol. 53, pp. 130–147, 2016.

[40] F. Han, B. Reily, W. Hoff, and H. Zhang, “Space-time representation of people based on 3D skeletal data: A review,” *Comput. Vis. Image Underst.*, vol. 158, pp. 85–105, 2017.

[41] K. Tang, L. Fei-Fei, and D. Koller, “Learning latent temporal structure for complex event detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* IEEE, 2012, pp. 1250–1257.

[42] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[43] A. Murali, A. Garg, S. Krishnan, F. T. Pokorny, P. Abbeel, T. Darrell, and K. Goldberg, “TSC-DL: Unsupervised trajectory segmentation of multi-modal surgical demonstrations with deep learning,” in *Proc. IEEE Conf. Robot. Autom.* IEEE, 2016, pp. 4150–4157.

[44] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[45] J. Weston, S. Chopra, and A. Bordes, “Memory networks,” *arXiv preprint arXiv:1410.3916*, 2014.

[46] O. Koller, S. Zargaran, and H. Ney, “Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017.

[47] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[48] Z. Lin, M. Chen, and Y. Ma, “The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices,” *arXiv preprint arXiv:1009.5055*, 2010.

[49] G. Tang and A. Nehorai, “Robust principal component analysis based on low-rank and block-sparse matrix decomposition,” in *Proc. Conf. Infor, Sci. Syst.* IEEE, 2011, pp. 1–5.

[50] J. Yao, X. Liu, and C. Qi, “Foreground detection using low rank and structured sparsity,” in *Proc. IEEE Int. Conf. Multimed. Expo.*, 2014, pp. 1–6.

[51] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, “Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 2080–2088.

[52] X. Liu, G. Zhao, J. Yao, and C. Qi, “Background subtraction based on low-rank and structured sparse decomposition,” *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2502–2514, 2015.

[53] P. L. Combettes and J.-C. Pesquet, “A Douglas–Rachford splitting approach to nonsmooth convex variational signal recovery,” *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 564–574, 2007.

[54] M.-J. Fadili and J.-L. Starck, “Monotone operator splitting for optimization problems in sparse recovery,” in *Proc. IEEE Int. Conf. Image Process.* IEEE, 2009, pp. 1461–1464.

[55] G. H. Golub and C. F. Van Loan, “Matrix Computations,” *The Johns Hopkins Univ. Press*, 1996.

[56] X. Cao, L. Yang, and X. Guo, “Total variation regularized RPCA for irregularly moving object detection under dynamic background,” *IEEE Trans. Cybern.*, vol. 46, no. 4, pp. 1014–1027, 2015.

[57] W. Li, Z. Zhang, and Z. Liu, “Action recognition based on a bag of 3D points,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*. IEEE, 2010, pp. 9–14.

[58] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras, “Two-person interaction detection using body-pose features and multiple instance learning,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*. IEEE, 2012, pp. 28–35.

[59] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” *J. Mach. Learn. Res.*, vol. 9, no. Nov, pp. 2579–2605, 2008.