Discriminative Spatio-Temporal Pattern Discovery for 3D Action Recognition

Junwu Weng, *Member, IEEE*, Chaoqun Weng, Junsong Yuan, *Senior Member, IEEE*, Zicheng Liu, *Fellow, IEEE*

***Abstract*—Despite the recent success of 3D action recognition using depth sensor, most existing works target how to improve action recognition performance, rather than understanding how different types of actions are performed. In this work we propose to discover discriminative spatio-temporal patterns for 3D action recognition. Discovering these patterns can not only help improve the action recognition performance, but also help us understand and differentiate between action categories. Our proposed method takes the spatio-temporal structure of 3D action into consideration and can discover essential spatio-temporal patterns that play key roles in action recognition. Instead of relying an end-to-end network to learn the 3D action representation and perform classification, we simply present each 3D action as a series of temporal stages composed by 3D poses. Then we rely on nearest neighbor matching and bilinear classifiers to simultaneously identify both critical temporal stages and spatial joints for each action class. Despite using raw action representation and a linear classifier, experiments on five benchmark datasets show that the proposed spatio-temporal Nave Bayes Mutual Information maximization (ST-NBMIM) can achieve competitive performance compared with the state-of-the-art methods that use sophisticated end-to-end learning, and has the advantage of finding discriminative spatio-temporal action patterns.**

***Index Terms*—NBMIM, Spatio-temporal Pattern Discovery, Discriminative Skeleton-based Action Recognition**

I. Introduction

In this decade, thanks to the availability of commodity depth cameras and the contribution of pose extraction method [1], skeleton-based action recognition has drawn considerable attention in computer vision community. Deep-learning-based methods [8], [51]–[53] in action recogntion from RGB data have made great success recently, which also inspire the works in 3D action recognition. The leading methods for 3D action recognition so far are learning-based classifiers including deep learning based methods [2]–[7], which have shown promising results on benchmark datasets.

While learning-based methods have made significant progress in 3D action recognition problem, non-parametric models, which do not involve training or learning for parameters, have not been well explored. Meanwhile, we have already witnessed the success of NN-based model Naive-Bayes Mutual Information Maximization (NBMIM) being applied to action detection problem. Motivated by the success of NBMIM in action detection problem, we explore it for 3D actions recognition.

The motivation of applying NBMIM [9] to 3D action recognition is on the basis of the following three observations. (1) Compared with RGB-based image or video analysis problem which always faces millions or billions of pixels, skeletal data only consists of tens of joints. We believe that compared with sophisticated end-to-end model, a simple NN-based model can still well handle such a lightweight problem; (2) Similar to images that are the composition of local visual primitives, actions can be represented as a set of temporal primitives, as the temporal stage-descriptor we defined in Sec. III-A. Therefore, it is possible to generalize NBNN [10] to 3D action problem by applying the *primitive-to-class* distance to recognize actions; (3) Considering that actions from different action classes may share a great number of similar temporal primitives, which are not helpful to action classification, we can borrow the idea from NBMIM [9] to introduce negative primitives into nearest neighbor matching thus boosting the discriminative ability of temporal primitives.

Our Spatio-temporal Naive-Bayes Mutual Information Maximization is an extension of NBMIM. In our framework, each 3D action is represented as a set of temporal stages which are composed of 3D poses. Each 3D pose in a stage is presented by a collection of spatial joints. Similar to NBMIM, our ST-NBMIM also applies the summation of temporal primitive mutual information with respect to action classes to distinguish action instances. Moreover, ST-NBMIM takes the spatio-temporal structure of action sequences into consideration. Even though an action instance comprises a set of temporal stages, not every temporal stage and the related spatial joints are of equal importance in action recognition. It is greatly important to discover the critical temporal stages and spatial joints that matter for recognition. As illustrated in Fig. 1, when performing right hand waving action, only the right hand and arm (key joints) are activated. Meanwhile, when observing the timing (key stage) at which the right hand and arm raise up and move horizontally towards left, we can conclude that waving right hand action is being performed. Such a spatio-temporal pattern described by key temporal stages and spatial joints is critical to identify action classes. The discovery of such patterns not only can improve recognition accuracy but also provides answers to the deeper questions of what an action instance are composed of and why it is recognized as a particular action class. We consider that the visual interpretability [47], [48], [54] of model is also an important topic in action recognition. To this end, we represent the mutual information of temporal primitives as the mutual

Junwu Weng, Chaoqun Weng are with School of Electrical & Electronics Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798. (e-mail: we0001wu@e.ntu.edu.sg, weng0018@e.ntu.edu.sg).

Junsong Yuan is with Department of Computer Science and Engineering, The State University of New York, Buffalo, NY 14260-2500, USA. (jsyuan@buffalo.edu).

Zicheng Liu is with Microsoft Research Redmond, WA 98052-6399, USA. (zliu@microsoft.com).

Copyright © 2018 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

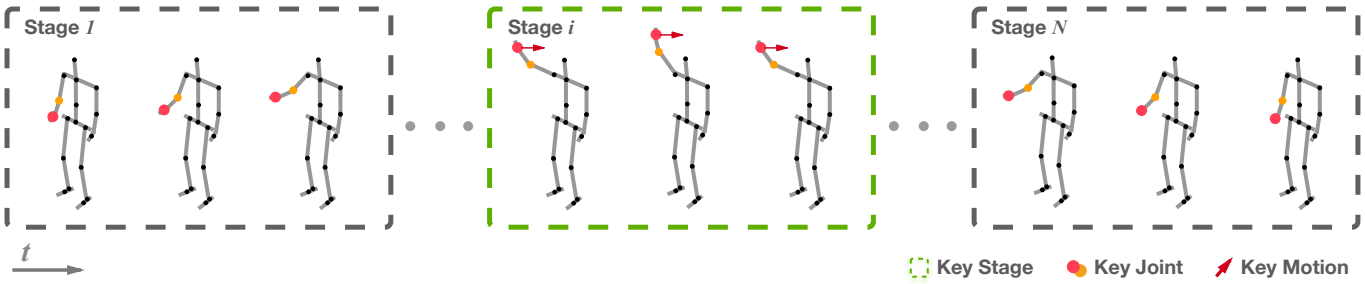

Fig. 1. Illustration of Key Stage, Joints, and Motion for the action of waving right hand action.

information matrix, which is the combination of the "mutual information" of each spatial joint. Further, ST-NBMIM adopts a bilinear classifier [11] to identify those key joints and stages with discriminative "mutual information" and utilize these key elements to classify the mutual information matrix of 3D actions. This process is implemented by iteratively learning the linear classification weight for both spatial joints and temporal stages.

ST-NBMIM combines the strengths of non-parametric model and parametric model by utilizing both the mutual information of temporal stage w.r.t action class and bilinear classifier [11]. Experiments show that with only a linear classifier, our proposed method achieves competitive performance on four benchmark datasets compared with the state-of-the-art models. We also witness the potential of ST-NBMIM on large scale dataset. Furthermore, ST-NBMIM bears the ability to capture the essential spatio-temporal patterns for each action class, which play key roles in recognizing actions and provide physical interpretations of action behavior.

II. Related Work

In skeleton-based action recognition, the input is a sequence of 3D poses that records the performing of a certain action, and the output is the corresponding action label that sequence belongs to. In these years, skeleton-based action recognition problem has attracted a lot of attention and many learning-based methods [2]–[8] have been proposed. Due to the considerable amount of work in this area, we only focus our review on the spatio-temporal modeling of skeleton-based action recognition.

The modeling of spatial domain can be divided into two categories, part-based model and sub-pose model. To date, the spatial domain modeling is mainly driven by the fact that an action is usually only characterized by the interactions or combinations of a subset of skeleton joints [4]. In the part-based model, the joints of a skeleton are partitioned into several groups, and in each group the joints are skeletal neighbors of each other. In [5] a part-aware LSTM is proposed to construct the relationship between body parts. Similarly, in HBRNN [2], skeletons are decomposed into five parts, two arms, two legs, and one torso, and a hierarchical recurrent neural network is built to model the relationship among these parts. In sub-pose model, the focus is mainly on the informative joints or their interactions. In SMIJ [12], the most informative joints are selected simply based on measures such as mean or variance of joint angle trajectories. The sequence of these informative joints is then used as the representation of actions. In Orderlet [13], interactions between joints are modeled by a few comparisons of joints' primitive feature, and in action recognition only a subset of joints is involved. On the temporal domain, temporal pyramid matching [14], [15], dynamic time warping [16], and graphical models [17], [18] are the commonly used methods for temporal modeling. While in [19], sequential pattern mining method is used to model temporal structures of a set of key poses.

Besides spatial modeling or temporal modeling, we also see efforts on spatio-temporal modeling. Wang *et al.* [55] apply data mining techniques to discover co-occurring distinctive spatial body-part structures and temporal pose evolutions. In classification, bag-of-words model based on the mined spatial-part-sets and temporal-part-sets is utilized for action representation. Compared to [55], our proposed method focuses on discriminative and class-related individual spatial joints and temporal stages discovery. For a certain action category, the classification decision is mainly determined by the data from the discovered discriminative joints and stages. In [56], a hierarchical model is proposed to recognize pose-based, composable and concurrent actions and activities. Based on the learned motions poselets and actionlets dictionaries, the hierarchical model can provide spatio-temporal annotations of complex actions. The annotation can tell when the related body parts are activated for which atomic action, but these annotations may not be discriminative in classification. While in ST-NBMIM, key joints and key stages discovered in ST-NBMIM are discriminative for classification. In [6], a LSTM model is extended to spatio-temporal domain to analyze skeletons. In *Spatio-Temporal Naive-Bayes Nearest Neighbor* ST-NBNN [20], bilinear classifier is utilized to discover the spatio-temporal structure of 3D action. Another track on spatio-temporal modeling is the CNN-based 3D action recognition model [41], [49], [50]. In these models, a 3D action sequence is first visualized as an image. The pixels of these image samples are directly or indirectly related to the joint coordinates, which means the spatio-temporal information of a 3D sequence is re-organized as the combination of pixels. A further CNN-model is applied to extract features and predict the label of actions based on the input image samples. The CNN model in these works plays the role of implicitly extracting spatio-temporal information from 3D sequences. Compared with these CNN-based methods, the spatio-temporal modeling

of ST-NBMIM is more explicit, and spatio-temporal patterns discovered by ST-NBMIM is physically interpretable.

Our ST-NBMIM is an extension of ST-NBNN [20]. In this work, we introduce the idea of mutual information into ST-NBNN. The involvement of negative samples can help boost the discriminative ability of action representation. The idea of applying mutual information calculation in NBNN framework was first proposed in [9], in which NBNN was re-designed as *Naive-Bayes based Mutual Information Maximization* (NBMIM) to solve action detection problem. One interesting property of NBMIM is that negative samples are involved in nearest neighbor matching to improve the discriminative ability of descriptors. NBMIM is a nearest-neighbor-based (NN-based) method since the calculation of mutual information in NBMIM relies on the nearest neighbor search. Even though NN-based methods are simple and non-parametric, their successes in image classification and action detection prove the effectiveness of these approaches. Recently, the combination of NBNN and CNN [21], as well as the effort to speed up NN search [22], revives the return of NN-based methods in computer vision.

III. Proposed Method

In this section, we introduce how the involvement of mutual information helps to improve the discriminative ability of descriptors, and how the proposed method, ST-NBMIM, predicts actions. The overview of our method is illustrated in Fig. 2. We first introduce how to represent a 3D sequence of actions, which includes single person action and two-person interactive action (Sec. III-A). Then NBMIM [9] is used as a basic framework to predict skeleton-based action instances (Sec. III-B). Finally, the learning of spatial and temporal weights is introduced to discover key poses and spatial joints for 3D action recognition (Sec. III-C).

A. Representation for 3D Actions

Single-Person Action

In skeleton-based action recognition, a 3D action instance is regarded as a sequence of 3D poses. Different actions performed by different subjects may have different action duration. In our proposed method, to provide a unified presentation, we partition each action into N temporal windows of equal length. Each temporal window is called *temporal stage*, and it is represented by the 3D poses in its corresponding window. Assuming each 3D pose has J joints for its skeleton, for a temporal stage descriptor $\boldsymbol{x}$, the 3D pose in its jth frame is denoted as $\boldsymbol{p}_j \in \mathbb{R}^{3J}$, and the related velocity of that pose is denoted as $\boldsymbol{v}_j \in \mathbb{R}^{3J}$. More specifically, p_j is the concatenation of 3D coordinates (x, y, z) of J joints of the pose in jth frame, and v_j is the difference between two pose features from consecutive frames, frame j and frame $j+1$, namely $v_j = p_{j+1} - p_j$. For v_j from the last frame, we assume that the pose does not move, which means that $v_j = 0$. Then the pose part $\boldsymbol{x}_p$ and velocity part $\boldsymbol{x}_v$ of $\boldsymbol{x}$ is defined as below,

$$\begin{aligned} \boldsymbol{x}_p &= [(\boldsymbol{p}_1)^\intercal, \ \dots, (\boldsymbol{p}_l)^\intercal]^\intercal \\ \boldsymbol{x}_v &= [(\boldsymbol{v}_1)^\intercal, \ \dots, (\boldsymbol{v}_l)^\intercal]^\intercal \end{aligned} \tag{1}$$

We follow the idea in [23] to also normalize $\boldsymbol{x}_p$ and $\boldsymbol{x}_v$. A temporal stage descriptor $\boldsymbol{x}$ of l frames is then represented as:

$$\boldsymbol{x} = [(\boldsymbol{x}_p)^\intercal, (\boldsymbol{x}_v)^\intercal]^\intercal \tag{2}$$

Based on the above notation, a 3D single person action video is described by its N stages-descriptors $V = \{\boldsymbol{x}^i\}_{i=1}^N$.

Two-Person Interaction

The description of two person interactive action is similar to the one of single person action. We partition each action into N temporal windows of equal length. Each stage includes the interactive action of two persons in a short temporal range. Inspired by the work in [24], we notice that the relative position between joints from two different persons is much more informative in interaction recognition than the individual position. On the basis of this idea, we use the difference between stages-descriptors of person A and person B involved in an interactive action, namely $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$, as the interaction representation. The delta descriptor is defined as,

$$\boldsymbol{x}_\delta = abs(\boldsymbol{x}_a - \boldsymbol{x}_b) \tag{3}$$

where $abs(\cdot)$ denotes the element-wise absolute operation for the input descriptor. This absolute operation can well handle the problem that we do not bear any information about which person is an "active" actor and which one is an "inactive" actor. Therefore finally, a 3D interaction video is described by its N delta stage-descriptors $V = \{\boldsymbol{x}_\delta^i\}_{i=1}^N$.

B. Naive-Bayes Mutual Information Maximization

Given a query action video $V_q = \{\boldsymbol{x}^i\}_{i=1}^N$, the goal is to find which class $c \in \{1, 2, ..., C\}$ the video V_q belongs to. *Naive-Bayes Mutual Information Maximization* (NBMIM) evaluates the mutual information I between the query video and a specific class c to identify the action label:

$$c^* = \arg\max_c I(V_q; c) = \arg\max_c log\frac{p(V_q|c)}{p(V_q)} \tag{4}$$

With the Naive-Bayes assumption (the stage-descriptors are independent of each other), Eq. 4 can be written as Eq. 5

$$\begin{aligned} c^* &= \arg\max_c log\frac{p(V_q|c)}{p(V_q)} = \arg\max_c log\prod_{i=1}^N \frac{p(\boldsymbol{x}^i|c)}{p(\boldsymbol{x}^i)} \\ &= \arg\max_c \sum_{i=1}^N log\frac{p(\boldsymbol{x}^i|c)}{p(\boldsymbol{x}^i)} = \arg\max_c \sum_{i=1}^N I(\boldsymbol{x}^i; c) \end{aligned} \tag{5}$$

where $I(\boldsymbol{x}^i; c)$ is the mutual information between the ith stage-descriptor and the class c, and with the prior $p(c) = 1/C$ it can be derived as

$$I(\boldsymbol{x}^i; c) = log\frac{C}{1 + \frac{p(\boldsymbol{x}^i|\bar{c})}{p(\boldsymbol{x}^i|c)}(C-1)} \tag{6}$$

where $\bar{c}$ denotes the negative class of c, that is, all the classes except c.

Based on the analysis of [9], the ratio between $p(\boldsymbol{x}^i|\bar{c})$ and $p(\boldsymbol{x}^i|c)$ can be estimated according to the distance between $\boldsymbol{x}$ and the nearest neighbor of $\boldsymbol{x}$ in class c and $\bar{c}$,

$$\frac{p(\boldsymbol{x}^i|\bar{c})}{p(\boldsymbol{x}^i|c)} \approx \gamma^c exp^{-\frac{1}{2\delta^2}(\|\boldsymbol{x}^i - NN_{\bar{c}}(\boldsymbol{x}^i)\|^2 - \|\boldsymbol{x}^i - NN_c(\boldsymbol{x}^i)\|^2)} \tag{7}$$

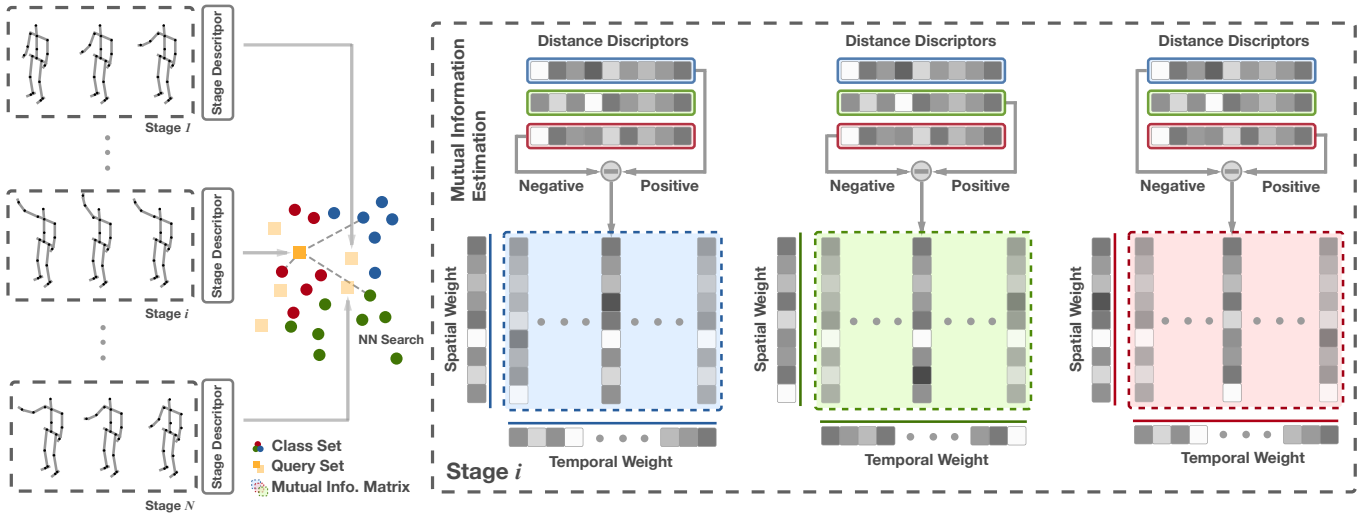

Fig. 2. Overview of ST-NBMIM. 1) A 3D action sequence is uniformly divided into N stages, which is predefined, and is represented by a set of stage-descriptors (orange query points); 2) Distances of stage-descriptors to action class sets (blue, green and red), namely distance descriptor, are calculated by NN search; 3) Mutual information are estimated by calculating the difference between positive distance descriptor and the nearest negative distance descriptor to generate mutual information descriptors; 4) Mutual information descriptors are gathered in temporal order to generate class-related mutual information matrices (marked by class-related dashed rectangular boxes); 4) Weights on the spatial (left side of the matrix) and the temporal (bottom of the matrix) domain are learned to discover key factors of actions and predict action labels

where $\gamma^c = N_c/N_{\bar{c}}$, in which N_c is the number of stage-descriptor from class c and $N_{\bar{c}}$ is the number of stage-descriptor from negative class $\bar{c}$. δ is the kernel bandwidth in density estimation. $NN_c(\boldsymbol{x})$ and $NN_{\bar{c}}(\boldsymbol{x})$ indicate the nearest neighbors of the query descriptor $\boldsymbol{x}$ in positive class c and negative class $\bar{c}$, respectively.

Based on Eq. 6 and Eq. 7, we can conclude that $I(\boldsymbol{x}^i; c) \propto \|\boldsymbol{x}^i - NN_{\bar{c}}(\boldsymbol{x}^i)\|^2 - \|\boldsymbol{x}^i - NN_c(\boldsymbol{x}^i)\|^2$, and we then simplify Eq. 5 as

$$\begin{aligned} c^* &= \arg\max_c \sum_{i=1}^{N} (\|\boldsymbol{x}^i - NN_{\bar{c}}(\boldsymbol{x}^i)\|^2 - \|\boldsymbol{x}^i - NN_c(\boldsymbol{x}^i)\|^2) \\ &= \arg\max_c \sum_{i=1}^{N} \tilde{I}(\boldsymbol{x}^i; c) \end{aligned} \tag{8}$$

where $\tilde{I}(\boldsymbol{x}^i; c)$ is the estimated mutual information between stage-descriptor $\boldsymbol{x}^i$ and class c.

For a query 3D action $V_q = \{\boldsymbol{x}^i\}_{i=1}^N$, each of its stage-descriptor will match against C classes separately by finding the best matched temporal stage, i.e., nearest neighbor, in that class. Note that different action classes may share similar temporal stages, these similar stage-descriptors $\boldsymbol{x}^i$ are not discriminative. For those discriminative ones, the differences between their distances to the positive class and negative classes are large. Eq. 8 helps suppress the similar temporal stages among C classes by applying difference operation between negative and positive nearest neighbor distances. The larger the difference between the negative and positive distances, the stronger the temporal stage will vote for that class c. As V_q has in total N temporal stages, the final decision is the summation (average) over all of the N suppressed votes towards to C classes, as described in Eq. 8.

C. Spatio-Temporal NBMIM

When performing a specific action, often only a subset of joints are activated, and for actions from different classes the activated joints are different. Meanwhile, only those joints with high mutual information, as estimated in Eq. 8, bears strong classification ability. Based on this observation, we can only select those spatial joints with high mutual information and ignore the ones that are not informative. On the temporal domain the situation is similar. Among a set of temporal stages, not every stage is of equal importance neither. Depending on the action class, a certain temporal stage can be more discriminative than others for classification. As shown in Fig. 2, the stage-descriptor (shadowed orange query square) of stage i bears higher mutual information and is more discriminative than both the beginning stage and the ending stage.

To simultaneously identify informative spatial joints and temporal stages, bilinear classifier [11] is used to mine saptio-temporal patterns in the framework of NBMIM.

Mutual Information Matrix

Although in Eq. 5 we assume that the stages are independent of each other (Naive Bayes assumption), they in fact depend on each other in a certain spatio-temporal structure. Hence, to discover the spatio-temporal structure of 3D actions, we first represent a 3D action instance from a set $V = \{\boldsymbol{x}^i\}_{i=1}^N$ as a matrix. For a given video sample with N stages, its spatio-temporal matrix is defined as

$$\boldsymbol{X} = [\boldsymbol{x}^1, \ \ldots, \boldsymbol{x}^N] \tag{9}$$

Stage-descriptors of an action instance are re-organized column by column following the temporal order. Then we define the nearest neighbor matrix of X in c as $\boldsymbol{X}_c^{NN} =$

$[NN_c(\boldsymbol{x}^1), \ldots, NN_c(\boldsymbol{x}^N)]$, and the squared distance matrix to class c is defined as

$$\boldsymbol{X}_c = (\boldsymbol{X} - \boldsymbol{X}_c^{NN}) \odot (\boldsymbol{X} - \boldsymbol{X}_c^{NN}) \tag{10}$$

where $\odot$ is an element-wise product. $\boldsymbol{X}_c$ is regarded as a representation of $\boldsymbol{X}$ for class c, and it is a combination of element-wise stage-to-class distances of the testing sample. Similarly, if we regard class c as the positive class, then the negative squared distance matrix is $\boldsymbol{X}_{\bar{c}} = (\boldsymbol{X} - \boldsymbol{X}_{\bar{c}}^{NN}) \odot (\boldsymbol{X} - \boldsymbol{X}_{\bar{c}}^{NN})$. Based on the definition above, the mutual information matrix is defined as

$$\boldsymbol{X}_c^I = \boldsymbol{X}_{\bar{c}} - \boldsymbol{X}_c \tag{11}$$

Summation of all the elements in $\boldsymbol{X}_c^I$ is equivalent to the $\sum_{i=1}^{N} \tilde{I}(\boldsymbol{x}^i; c)$ in Eq. 8. The mutual information matrix $\boldsymbol{X}_c^I$, as illustrated in Fig. 2, is the representation of the action instance V in class c.

Since the summation of elements in $\boldsymbol{X}_c^I$ determines the final classification decision, as shown in Eq. 8, each element contributes equally to the recognition task. However, only the discriminative elements have great impacts to the final decision and therefore this NBMIM framework, Eq. 8, should be parameterized to emphasize those discriminative ones. Let's simply vectorize $\boldsymbol{X}_c^I$ as $\boldsymbol{\chi}_c^I$, and the NBMIM decision function Eq. 8 is then redefined as $c^* = \arg\min_c \boldsymbol{w}^\intercal \boldsymbol{\chi}_c^I$, where the weight $\boldsymbol{w}$ can be learned by linear SVM. However, since the mutual information matrix $\boldsymbol{X}_c^I$ is a large matrix, there will be a large number of weight parameters that need to be determined. Learning weights by linear SVM is not only time-consuming but also has the risk of over-fitting. Therefore in our work, we leverage bilinear classifier [11] to solve the weight learning problem.

With the mutual information matrix $\boldsymbol{X}_c^I$ of a query matrix $\boldsymbol{X}$, the classification score is then determined by a bilinear function $f_c(\cdot)$, which is defined as

$$f_c(\boldsymbol{X}_c^I) = (\boldsymbol{u}_c^s)^\intercal \, \boldsymbol{X}_c^I \, \boldsymbol{u}_c^t \tag{12}$$

where $\boldsymbol{u}_c^s \in \mathbb{R}^M$ and $\boldsymbol{u}_c^t \in \mathbb{R}^N$ are the spatial and temporal weights of action class c. As a result, the classification becomes

$$c^* = \arg\min_c f_c(\boldsymbol{X}_c^I) \tag{13}$$

As can be seen from Eq. 12, the proposed method provides weights for both temporal stages and spatial joints. After a rearrangement, Eq. 13 can be represented as,

$$c^* = \arg\min_c \sum_{i=1}^{N} \boldsymbol{u}_c^t(i) \, \left\| \tilde{I}(\boldsymbol{x}^i; c)^\intercal \sqrt{\boldsymbol{u}_c^s} \right\|^2 \tag{14}$$

where $\sqrt{\cdot}$ is an element-wise square-root of a vector. As we can see, NBMIM is a special case of ST-NBMIM. When $\boldsymbol{u}_c^s$ and $\boldsymbol{u}_c^t$ are assigned to $\mathbf{1}$, Eq. 14 becomes NBMIM in Eq. 8. With the break of Naive-Bayes rule, ST-NBMIM becomes a generalization of NBMIM. We introduce the spatio-temporal structure of 3D action into our framework to break the assumption of stage independence. And the key joints and stages can be discovered by the learned weights $\boldsymbol{u}_c^s$ and $\boldsymbol{u}_c^t$.

Spatial and Temporal Weight Learning

For the learning of $\boldsymbol{u}_c^s$ and $\boldsymbol{u}_c^t$, we introduce the objective function that is similar to tensor SVM. Following the learning strategy of [25], we adopt the one-vs.-all strategy to classify actions. With empirical loss, the objective function of spatio-temporal weight learning is defined as

$$\begin{aligned} \min_{\boldsymbol{u}_i^s, \boldsymbol{u}_c^t} \quad & \frac{1}{2} \left\| \boldsymbol{u}_c^s (\boldsymbol{u}_c^t)^\intercal \right\|^2 + \lambda \sum_{i=1}^{K} \xi_i \\ s.t \quad & \sum\nolimits_{i=1}^{N} \boldsymbol{u}_c^t(i) = N, \qquad \boldsymbol{u}_c^t \succeq \mathbf{0} \\ & \xi_i \geqslant max(0, 1 - c_i f_c(\boldsymbol{X}_c^{I(i)}))^2 \\ & \xi_i \geqslant 0, \qquad i = 1, ..., K \end{aligned} \tag{15}$$

in which K is the number of training video samples, and $c_i \in \{-1, 1\}$ is the action label of the corresponding sample. $\boldsymbol{X}_c^{I(i)}$ is the ith training sample (mutual information matrix) in class c. λ is a parameter for classification error penalty.

Here we apply linear constraints to the temporal domain but not to the spatial domain. The reason is that for the spatial domain the number of involved key joints is uncertain. Some spatial joints like the *hip* joint usually do not have any contributions to recognition. Compared with spatial key joints, on temporal domain every stage of an action counts in classification. Experiment results also show that linear constraints on the spatial domain do not bear any contribution to performance, but the temporal constraints do.

The optimization of Eq. 15 is solved through an iterative process. There are two steps in each iteration, 1) fix $\boldsymbol{u}_c^t$ and update $\boldsymbol{u}_c^s$, 2) fix $\boldsymbol{u}_c^s$ then update $\boldsymbol{u}_c^t$. $\boldsymbol{u}_c^t$ is initialized to $\mathbf{1}$.

Fix $\boldsymbol{u}_c^t$ and Update $\boldsymbol{u}_c^s$: With $\boldsymbol{u}_c^t$ fixed, Eq. 15 is treated as a l_2-regularized l_2 loss SVM problem shown below

$$\min_{\boldsymbol{u}_c^s} \quad \frac{1}{2} \beta_1 \|\boldsymbol{u}_c^s\|^2 + \lambda \sum_{i=1}^{K} max(0, 1 - c_i f_c(\boldsymbol{X}_c^{I(i)}))^2 \tag{16}$$

where $\beta_1 = \|\boldsymbol{u}_c^t\|^2$.

Fix $\boldsymbol{u}_c^s$ and Update $\boldsymbol{u}_c^t$: With updated $\boldsymbol{u}_c^s$, Eq. 15 is regarded as a convex optimization problem with linear constraints shown below

$$\begin{aligned} \min_{\boldsymbol{u}_c^t} \quad & \frac{1}{2} \beta_2 \|\boldsymbol{u}_c^t\|^2 + \lambda \sum_{i=1}^{K} max(0, 1 - c_i f_c(\boldsymbol{X}_c^{I(i)}))^2 \\ s.t \quad & \sum\nolimits_{i=1}^{N} \boldsymbol{u}_c^t(i) = N, \qquad \boldsymbol{u}_c^t \succeq \mathbf{0} \end{aligned} \tag{17}$$

where $\beta_2 = \|\boldsymbol{u}_c^s\|^2$.

The optimization problem defined in Eq. 15 can be solved by solving Eq. 16 and Eq. 17 iteratively.

The time complexity is based on what optimization solution we use. Eq. 16 is a $l2$-regularized $l2$-loss linear SVM problem. The optimal parameters can be obtained by trust region newton method proposed in [45]. It's time complexity is not higher than $\boldsymbol{O}(L * K * M)$ for each iteration, where L is the number of conjugate gradient iterations, K is the number of training samples, and M is the dimension of stage-descriptor. Eq. 17

is the same problem with linear constraints. Here we use the interior point method to obtain the optimal parameters. It's time complexity is around $\boldsymbol{O}(\sqrt{K*N})$ when the self-concodance condition holds, as discussed in [46]. Considering that $M \gg N$, most of the time is consumed in the spatial weight learning part.

In ST-NBMIM, two steps, one for spatial weight update and one for temporal weight update, are regarded as one iteration. Let $\boldsymbol{u}_{c0}^s$ be the initial value of spatial weight. When $\boldsymbol{u}_{c0}^t$ is fixed, we obtain $\boldsymbol{u}_{c0}^s$ by solving the optimization problem in Eq. 16. Likewise, fixing $\boldsymbol{u}_{c0}^s$, we can obtain $\boldsymbol{u}_{c1}^t$ by solving Eq. 17. Notice that each sperate optimization problem, as defined in Eq. 16 and Eq. 17, is convex, so the solutions of them are globally optimums. Therefore we have,

$$h(\boldsymbol{u}_{c0}^s, \boldsymbol{u}_{c0}^t) \geq h(\boldsymbol{u}_{c0}^s, \boldsymbol{u}_{c1}^t) \geq h(\boldsymbol{u}_{c1}^s, \boldsymbol{u}_{c1}^t) \geq \dots \tag{18}$$

where $h(\cdot)$ is the objective function defined in Eq. 15. Considering that $h(\cdot)$ is larger than zeros, the optimization process converges.

IV. Experiment

In this section, we evaluate the proposed method on five 3D action datasets and compare its performance to existing methods. Implementation details are provided in Sec. IV-A. The description of the five benchmark datasets, the MSR-Action3D dataset [26], the UT-Kinect dataset [18], the Berkeley MHAD dataset [27], the SBU-Interaction dataset [24], and the NTU RGB+D dataset [5], is provided in Sec. IV-B. Among the five datasets, SBU-Interaction and NTU RGB+D contain interactive actions of two people. Comparison results on these datasets are provided and discussed in Sec. IV-C. The experiment results show that the introduction of mutual information helps to improve the action recognition accuracy over ST-NBNN. The discriminative matching helps boosting the discriminative ability of action representation. Although ST-NBMIM is simple, it is able to achieve comparable performance with state-of-the-arts and also effectively discover the key factors of an action class.

A. Implementations

3D Action Representation. The one-vs.-all strategy is utilized in this method. To ensure the responses of linear functions $f_c(\cdot)$ are comparable with each other, each sample $\boldsymbol{X}_c^{I(i)}$ is mean-centralized by $\mu^i = \sum_{c=1}^{C} sum(\boldsymbol{X}_c^{I(i)})/(C \times M \times N)$, where $sum(\cdot)$ sums up entries of the input matrix.

In Sec. IV-B, the setting of N is indicated. Due to the variation of action sequences' duration, stages defined in Sec. III-A may have overlaps when a sequence is too short.

To ensure that the representation introduced in Sec. III-A is location-invariant, for actions of single person each joint of the skeleton is centralized by subtracting coordinates of the hip joint. For interactive actions of two persons, two skeletal poses in each frame are centralized by subtracting the average coordinates of the two hip joints.

Nearest Neighbor Search. In our experiment, KD-tree implementation [28] and FLANN library [29] are used to boost the nearest neighbor searching process.

Spatio-Temporal Weight Learning. The training matrices $\boldsymbol{X}_c^I$ are generated by a leave-one-video-out strategy, which means all the stage-descriptors of a query training video are excluded from the nearest neighbor search. In our optimization, $\boldsymbol{u}_c^s$ and $\boldsymbol{u}_c^t$ are learned iteratively. To solve the SVM problem of Eq. 16, we use a SVM toolbox [30] implemented by Chang *et al.*, and to update $\boldsymbol{u}_c^t$, a convex optimization toolbox [31] is used.

B. Datasets

MSR-Action3D

There are 557 skeletal action sequences included in this dataset, and 20 human actions are involved. The actions recorded are common indoor daily actions. Each action is performed by 10 subjects twice or three times. The evaluation protocol we use is described in [26]. In this protocol, the 20 actions are grouped into three subsets AS1, AS2, and AS3, where each subset consists of eight actions. In this dataset, the number of poses in each local window is 10, and the number of stages N is set to 15.

Method	AS1	AS2	AS3	Ave.
Lie Group [32]	**95.4**	83.9	**98.2**	92.5
SCK+DCK [33]	–	–	–	94.0
HBRNN [2]	93.3	94.6	95.5	94.5
ST-LSTM [6]	–	–	–	94.8
Graph-Based [34]	93.6	95.5	95.1	94.8
ST-NBNN	91.5	**95.6**	97.3	94.8
ST-NBMIM	92.5	**95.6**	**98.2**	**95.3**

TABLE I
Comparison with State-of-the-arts on MSR-Action3D (%)

UT-Kinect

This dataset contains 10 action classes performed by 10 subjects. Each action are performed by each subject twice. We use the leave-one-out validation protocol described in [18] to evaluate our proposed method. Based on the description, there are 20 rounds of testing in our experiment. The parameters chosen for spatio-temporal weight learning are the same in each round. The number of local poses l is set to 3, and the number of stages N is 15.

Method	Accuracy
Key-Motif [19]	93.5
Simplices [23]	96.5
ST-LSTM [6]	97.0
Lie Group [32]	97.1
Graph-Based [34]	97.4
SCK+DCK [33]	**98.2**
ST-NBNN	98.0
ST-NBMIM	98.0

TABLE II
Comparison with State-of-the-arts on UT-Kinect (%)

Berkeley MHAD

The actions in this dataset are captured by a motion capture system. 11 action classes are included, and each action is performed by 12 subjects. We follow the experimental protocol described in [27] on this dataset. The sequences performed by the first seven subjects are for training while the ones performed by the rest subjects are for testing. Due to the high sampling rate, most of the data is redundant. We down-sample each sequence by selecting one frame for every ten frames. Under this setting, the number of local poses l is 20, and the number of stages N is 20.

Method	Accuracy
SMIJ [12]	95.4
Meta-cognitive [36]	97.6
Kapsouras *et al.* [35]	98.2
HBRNN [2]	**100.0**
ST-LSTM [6]	**100.0**
ST-NBNN	**100.0**
ST-NBMIM	**100.0**

TABLE III
COMPARISON WITH STATE-OF-THE-ARTS ON MHAD (%)

SBU-Interaction

This dataset contains eight classes of two-person interactive actions. 282 skeleton sequences are captured by Kinect depth sensor. For each skeleton, there are 15 joints in total. We follow the protocol proposed in [24] to evaluate our method. There is a five-fold cross validation. The evaluation is based on the average accuracy on these five folds. The number of stages N is 17, and there are three poses in each stage. Considering that there are pair actions in this dataset, when performing the nearest neighbor search, only the related stages are involved.

Method	Accuracy
HBRNN [2]	80.4
CHARM [37]	83.9
Deep LSTM [4]	86.0
Co-occurrence [4]	90.4
ST-LSTM [6]	**93.3**
ST-NBNN	89.3
ST-NBMIM	**93.3**

TABLE IV
COMPARISON WITH STATE-OF-THE-ARTS ON SBU (%)

NTU RGB+D

NTU-RGBD dataset is currently the most challenging dataset in 3D action recognition. It is collected with Kinect V2 depth camera. There are around 56 thousand sequences in total. 60 different action classes are performed by 40 subjects aged from 10 to 35. 25 joints are included in each skeletal pose. We follow the protocol introduced in [5] to conduct the experiment. In the nearest neighbor search, we only search for the related stage since there are 10 pair actions in this dataset. This dataset has two standard evaluation settings, the cross-subject (CS) evaluation and the cross-view (CV) evaluation. In cross-subject setting, half of the subjects are used for training and the remaining are for testing. In cross-view setting, two of the three views are used for training and the left one is for testing. The number of local poses l is set to 5, and the number of stages N is 20.

Method	CS	CV
Skeleton Quads [38]	38.6	41.4
Lie Group [32]	50.1	52.8
HBRNN [2]	59.1	64.0
Part-Aware LSTM [5]	62.9	70.3
ST-LSTM [6]	69.2	77.7
Two Streams [39]	71.3	79.5
GCA-LSTM [40]	74.4	82.8
New Representation [44]	79.6	84.8
Enhanced Vis. [41]	**80.0**	**87.2**
ST-NBNN	56.6	56.0
ST-NBMIM	64.5	64.1
ST-NBMIM + CNN Feature	**80.0**	84.2

TABLE V
COMPARISON WITH STATE-OF-THE-ARTS ON NTU RGB+D (%)

C. Results and Analysis

Comparison with Baselines

We compare the proposed method with spatio-temporal naive-bayes nearest neighbor (ST-NBNN) and four related baselines on five benchmark datasets, MSR-Action3D (M.), UT-Kinect (U.), SBU-Interaction (S.), Berkeley MHAD (B.), and NTU RGBD (N.CS for cross-subject setting, N.CV for cross-view setting). The baselines are (1) NBNN with N stages (NBNN-N); (2) NBNN with weight learning by linear SVM (NBNN+SVM); (3) Spatio-temporal NBNN (ST-NBNN); (4) NBMIM with N stages (NBMIM-N); and (5) NBMIM with weight learning by linear SVM (NBMIM+SVM). The results are shown in Table. VI

Method	M.	U.	S.	B.	N.CS	N.CV
NBNN-N	91.7	95.5	88.1	88.0	59.9	59.6
NBNN+SVM	92.4	94.0	83.0	100.0	44.4	43.0
ST-NBNN	94.8	**98.0**	89.3	**100.0**	56.6	56.0
NBMIM-N	93.3	96.0	89.2	88.0	60.8	60.3
NBMIM+SVM	92.7	95.0	89.3	99.3	55.1	54.1
ST-NBMIM	**95.3**	**98.0**	**93.3**	**100.0**	**64.5**	**64.1**

TABLE VI
COMPARISON WITH BASELINES ON FIVE DATASETS (%)

As we can see from Table. VI, ST-NBMIM maintains or improves the performance of ST-NBNN. On SBU interaction dataset, ST-NBMIM improves by 4% over ST-NBNN. We can also see the accuracy improvement or maintenance from NBNN to NBMIM on all five benchmark datasets. As we discussed in Sec. III-C, if we only use linear SVM as the weight learning method, there will be a large number of parameters to be determined, and this strategy may cause over-fitting. From Table. VI, in most of the cases, there are drops from NBMIM to NBMIM+SVM, which indicates the over-fitting caused by SVM, and we can see that the performance of ST-NBMIM is better than NBMIM+SVM.

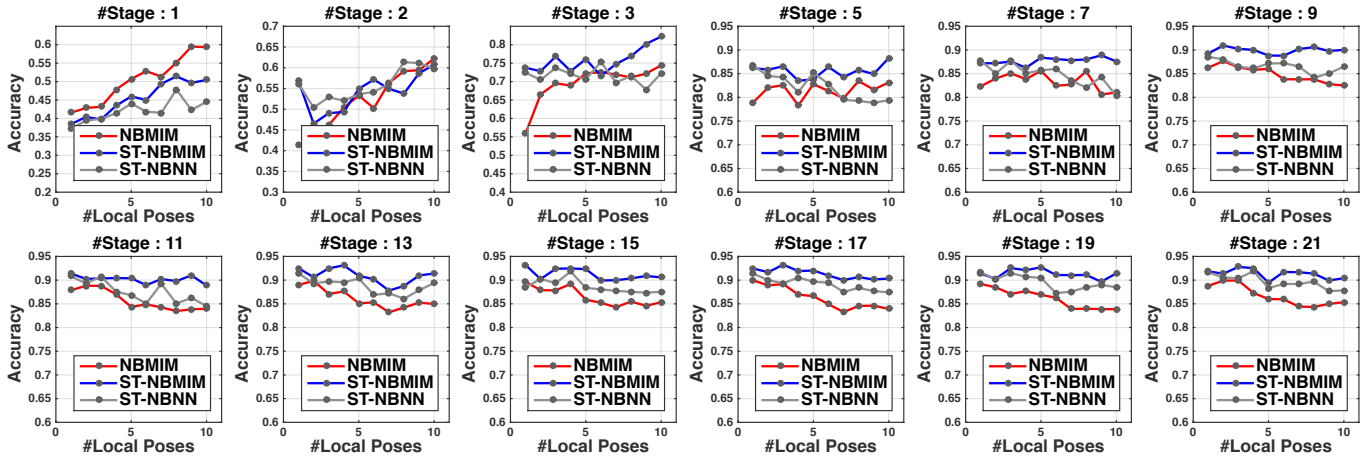

Fig. 3. Parameter Sensitivity Analysis on SBU Interaction Dataset. The x-axis indicates the chosen number of local poses. The sub-title indicates the chosen number of temporal stages.

The results from NTU-RGBD dataset show that the weight learning does not work for NBNN in this dataset. There are 3.3% and 3.6% drops from NBNN to ST-NBNN under cross-subject setting and cross-view setting respectively. However for NBMIM, the improvement is significant (3.7% improvement under cross-subject setting, 3.8% improvement under cross-view setting). The reason is that there are many actions that are very similar to each other in terms of the skeleton motion in NTU-RGBD dataset, e.g., *drinking water* and *eating snack*. The involvement of mutual information can help suppress elements that are similar among stage-descriptors and emphasize the elements that are discriminative. A further weight learning by our proposed method can help pick out those discriminative elements and improve the performance.

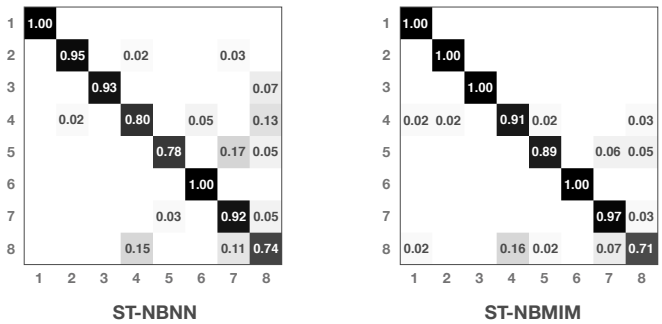

Fig. 4. Comparison of Confusion Matrix between ST-NBNN and ST-NBMIM. 1. Approaching, 2. Departing, 3. Kicking, 4. Pushing, 5. Shaking Hands, 6. Hugging, 7. Exchanging, 8. Punching

The motivation of extending ST-NBNN to ST-NBMIM via involving mutual information is to boost the discriminative ability of action representation. Fig. 4 shows the comparison of confusion matrix between ST-NBNN and ST-NBMIM on SBU Interaction dataset. As can be seen from the figure, there are great improvements of classification accuracy on action "Pushing" and "Shaking Hand". The actions "Shaking Hand" and "Exchanging" are very similar with each other. After involving mutual information into ST-NBNN, the confusion between these two action becomes less than before, and the overall performance of ST-NBMIM is better than ST-NBNN.

Combination with Convolutional Neural Network

In this section, we combine the proposed method ST-NBMIM with CNN model, ResNet18 [42]. Since our method mainly focus on skeleton-based action recognition, we can not apply CNN model on the data we use directly. Therefore we first transform the pose data of each video sample to an image by using the visualization method proposed in [43]. Then a ResNet [42] with 18 layers is used to extract CNN features for representing each video sample. In order to learn the spatio-temporal weights, we randomly pick 20% of the training samples as the validation set, and 80% of the training samples for fine-tuning ResNet18. Then the fine-tuned ResNet is used to extract CNN feature for both the validation set and testing set. We use the proposed ST-NBMIM as the classifier to predict the label of each sample based on the extracted CNN feature. We evaluate the combination version on NTU-RGBD dataset [5], and the results are shown in Table. VII. As can be seen from the table, CNN feature with ST-NBMIM can perform better than ResNet18, which means that ST-NBMIM can benefits CNN model. Besides, as can be seen from the Table. V, the performance of ST-NBMIM with CNN feature is comparable with the state-of-the-arts.

Method	CS	CV
NBMIM-N	60.8	60.3
ST-NBMIM	64.5	64.1
Res CNN Feature + FC (ResNet18)	78.9	83.2
Res CNN Feature + ST-NBMIM	**80.0**	**84.2**

TABLE VII
COMBINATION OF CNN FEATURE AND ST-NBMIM (%)

Comparison with the State-of-the-arts

In this section we compare the proposed method ST-NBMIM with the existing methods on five benchmark datasets. The

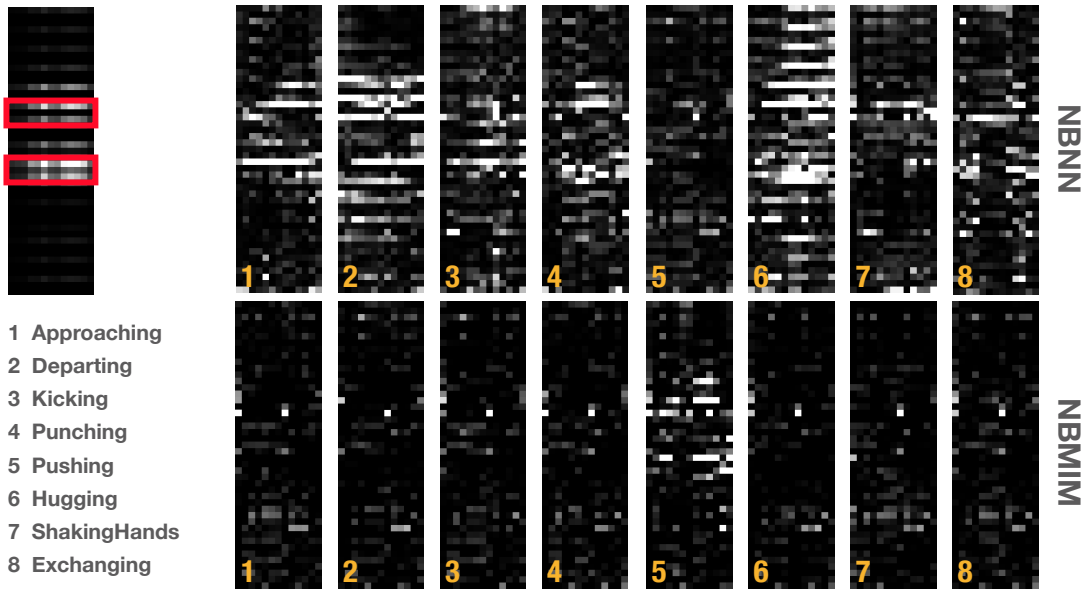

Fig. 5. An Example of Spatio-Temporal Weight Matrix and Comparison between Squared Distance Matrix from NBNN (first row) and Mutual Information Matrix from NBMIM (second row). The matrices are from the *Pushing* action. ST-Weight Matrix is on the top-left corner, and squared distance matrices as well as mutual information matrices are on the right side. Each matrix is 45 by 13. The related feature of discovered joints are marked by red box.

results are shown in Table. I, Table. II, Table. III, Table. IV, and Table. V. We can see that ST-NBMIM achieves the best accuracy on the MSR-Action3D dataset, Berkeley MHAD dataset and SBU-Interaction dataset. On the UT-Kinect dataset, the result is comparable with the state-of-the-arts.

On the NTU RGBD dataset, the proposed method ST-NBMIM can still perform better than the deep-learning-based method HBRNN [2] and Part-Aware LSTM [5]. However, we can also see that ST-NBMIM is not as good as other deep-learning-based model like [6], [39], [44]. The reason why ST-NBMIM is not better than the these models is that, compared with them, our proposed method only uses raw features directly from skeletal data, and only a linear method is utilized as the classifier, which does not have such large model capacity as the deep learning models. We also try to combine CNN feature from ResNet18 [42] with our ST-NBMIM classifier, and we witness great improvement from ST-NBMIM with raw data feature. From Table. V we can see that our combination version is comparable with the state-of-the-arts.

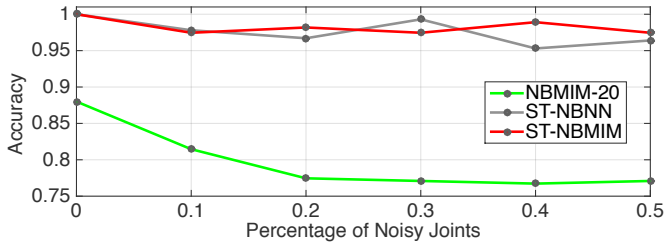

Fig. 6. Influence of Noisy Joints on Accuracy of Berkeley MHAD Dataset

Parameter Sensitivity Analysis

There are two main parameters in ST-NBMIM, the number of temporal stages N and the number of poses in each local stage l. The evaluation of parameter sensitivity is conducted on the SBU interaction dataset in this section. We change l from 1 to 10 and change N from 1 to 21. As Fig. 3 shows, ST-NBMIM needs a sufficient number of stages to learn the spatio-temporal weights and obtain good performance on action recognition. When the number of stages is larger than three, ST-NBMIM can still help improve the performance with only one pose in each stage. However, when the number of stages is larger than nine, further increasing N and l will not improve the performance explicitly. Fig. 3 also shows that when the stage number N is sufficient, ST-NBMIM achieves better accuracy than ST-NBNN, which indicates the effectiveness and robustness of the proposed method.

Robustness to Noise

In this section, we evaluate the tolerance and robustness of ST-NBMIM to random noise of skeleton data on Berkeley MHAD dataset. We randomly choose 10%, 20%, 30%, 40% and 50% of the 35 joints. For the randomly selected joints, we add noises ranging from -5 to 5 to each dimension of joints coordinates. This setting will result in mismatches of nearest neighbor search. The influence of noisy joint on accuracy is shown in Fig. 6. As we can see, as the percentage of noisy joints increases, the performance of NBMIM drops dramatically. Compared with NBMIM, ST-NBMIM can still pick out the informative elements and maintain the accuracy at high level. The average accuracy of ST-NBMIM in these six situations is 98.24%. We can also see that in most of the cases, ST-NBNN can not perform better than ST-NBMIM. The average accuracy of ST-NBNN in these six situations is

97.58%, which is not as good as ST-NBMIM.

Time Cost Analysis

In this section, we also experimented on SBU Interaction dataset to test the time cost of each step of the proposed method. In the training phase, the nearest neighbor search is conducted first to obtain the mutual information matrices for each training sample. Then the spatio-temporal weights learning part is conducted. In the testing phase, each stage-descriptor of a testing sample will be a query to search its nearest neighbors. After that class-related mutual information matrices will be generated. The classification part is then conducted to apply learned spatio-temporal weights on mutual information matrices. The test results are shown in Table. VIII and Table. IX respectively. There

Split	# Sample	NN Search	Training
1	227	19.8 s	9.1 ms
2	230	19.7 s	13.7 ms
3	226	18.8 s	12.2 ms
4	228	20.3 s	10.8 ms
5	217	18.5 s	16.9 ms
Ave.	225.6	19.4 s	12.5 ms

TABLE VIII
TIME COST IN TRAINING PHASE

Split	# Sample	NN Search	Testing
1	55	1.3 s	0.9 ms
2	52	1.2 s	0.8 ms
3	56	1.4 s	0.9 ms
4	54	1.3 s	1.6 ms
5	65	1.4 s	1.0 ms
Ave.	56.4	1.3 s	1.0 ms

TABLE IX
TIME COST IN TESTING PHASE

are five data splits in SBU Interaction dataset. We record the time cost of these five data splits on training phase and testing phase. The recorded time in Training part is from just one round (including one spatial weight update and one temporal weight update). In all the dataset we test, two rounds are already enough for spatio-temporal weight learning. This experiment is conducted on a Intel Xeon E5-2609 CPU with 2.50GHz clock frequency. As we can see, the proposed method does not take much time on training and testing. However, the most time-consuming part is the nearest neighbor search. As the number of stage-descriptors increases in the search area, the searching time will increase relatively. Therefore, our method is not suitable for real-time application. However, as we witness works like [22] focusing on boosting nearest neighbor search speed, we believe that the situation caused by low searching speed will be alleviated.

Convergence Analysis

In this section, we record the objective function value of each iteration in training on SBU Interation dataset. The result is shown in Fig. IV-C. The convergence curve shown

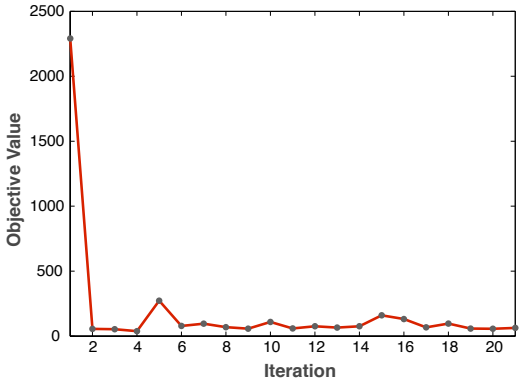

Fig. 7. Convergence Curve of ST-NBMIM

in Fig. IV-C is the average curve of eight binary classifiers of ST-NBMIM. In most of the case, only two or three iterations is enough for the training process to converge.

Visualization

In this section, we visualize the mutual information matrix and learned spatio-temporal weight matrix in Fig. 5 to help better understand the proposed method. Besides, the discovered key joints and key temporal stages are shown in Fig. 8. In Fig. 5, we provide an example of the learned spatio-temporal weight matrix and the estimated mutual information matrices $\boldsymbol{X}_c^I$ from *Pushing* action in SBU Interaction dataset. Due to space limitation, we only provide the first position feature of each stage and their related weights. Elements a_{ij} of the spatio-temporal weight matrix are determined by $a_{ij} = \boldsymbol{u}_c^s(i) \times \boldsymbol{u}_c^t(j), i = 1, ..., M, j = 1, ..., N$. The brighter the elements of the matrix, the larger the value of the elements. We can see from Fig. 5 that the *Pushing matrix* is the darkest one in NBNN and the brightest one in NBMIM. For NBNN described in ST-NBNN [20] and NBMIM described in Eq. 8, the classification is based on the summation of elements of each class-related squared-distance matrices of NBNN and mutual information matrices of NBMIM respectively. As we discussed in Sec. III-C, each elements of the matrices in NBNN and NBMIM bears the same contribution to the classification, and the motivation of the proposed weight learning method is to pick out those discriminative elements shared in each class-related matrices. Let's take the *Pushing* action for example. The ideal situation is that the elements that picked out by our weights learning method bear very high value in *matrix 5* , shown in Fig. 5, and very low value in other class-related matrices. In this case, the learned weight can help the summation of positive matrix, *matrix 5 (Pushing)* in this case, be the maximum one, and therefore help the classifier predict the true label. If the elements selected by our proposed weight learning method bear hight values in both positive matrix and negative matrices, the selected ones are not discriminative enough for the classification task. From Fig. 5 we can see that the discriminative elements as we described above in the mutual information matrices of NBMIM have more sparse pattern

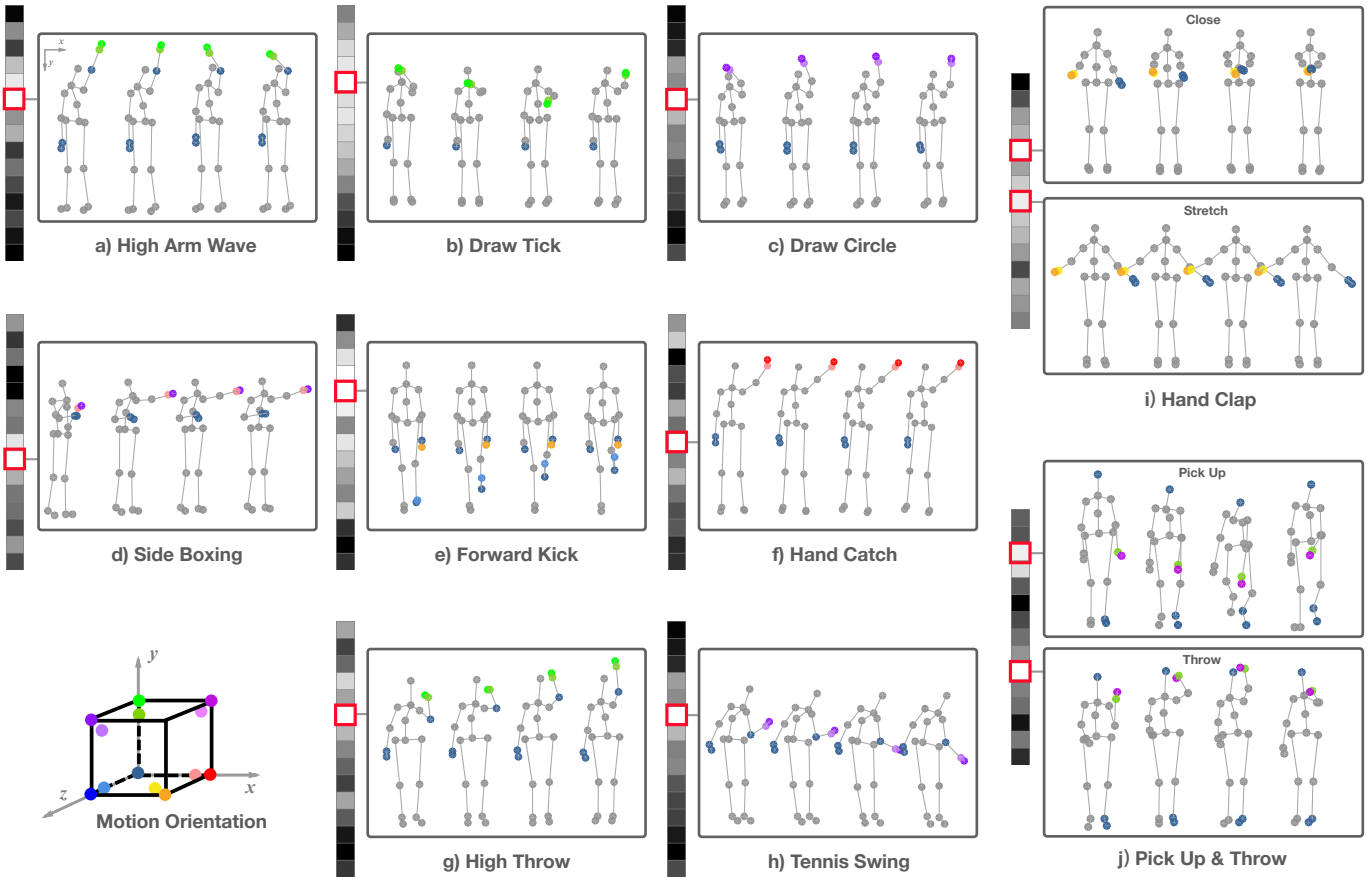

Fig. 8. Key Stages and Key Joints with Related Key Motions from MSR-Action3D. Colored joints are with weights larger than average weights. The most informative joints are marked by bright color, and the second most informative joints are marked by light color. Other key joints are marked by dark blue. The global key motions are indicated by different colors. For example, the key motion directing in the x direction is colored by bright red for the 1st most informative joint, and by light red for the 2nd most informative joint. Only the motion of 1st and 2nd key joints are marked. The temporal weights for each action are shown as gray images. Each square in this image represents a temporal stage. The whiter the square, the higher the temporal weight. The key stage is highlighted by a red box. We illustrate each key stage using its 4 representative 3D poses. The bottom two actions have two key stages each.

than those in NBNN, which is easier for the spatio-temporal weight learning method to discover, and therefore the mutual information representation is more suitable than squared distance representation in NBNN for the weights learning method. Meanwhile, compared with the ones in NBNN, the "negative" matrices in NBMIM have similar "sparse pattern" making it easy for the proposed method to discover discriminative elements and help distinguishing positive matrices from the negative ones. We can also see that the learned weights shown on the upper-left corner of Fig. 5 is able to correctly discover those discriminative elements. The red-square-marked region of the spatio-temporal weight matrix is related to the x, y, z coordinates of right and left hands (joint *6* and *9*), which are the most active joints in "Pushing" action.

Fig. 8 shows the key spatial joints and temporal stages discovered by the proposed method. For the "Forward Kick" action, ST-NBMIM selects the right hand as the most discriminative joint and right toe as the second most discriminative joint. When performing "Forward Kick", the dominant direction of joints' motion is z, and we can see that the proposed method discovers the direction. For the "Tennis Swing" action, the motion direction that is discovered by the proposed method is y-z, since when performing this action, the right hand mainly moves "down" (y) and "forward" (z). In the MSR-Action 3D dataset, "Side Boxing" and "Hand Catch" are similar to each other. ST-NBMIM selects the x direction of the second most discriminative joints for both of these two actions. In order to differentiate these two actions, the difference is on the 1st most informative joint (right hand). ST-NBMIM focuses on the y-z motion direction for the "Side Boxing" action, but on the x motion direction for the "Hand Catch" action. Interestingly, as shown in Fig. 8 i) and j), the proposed method can also indicate different phases of actions. The two peaks of the temporal weight of "Hand Clap" are related to the stages when two hands are close to each other and when the two hands are far apart from each other. For "Pick Up and Throw", the two peaks of the temporal weight are related to the "Pick Up" and "Throw" two phases respectively.

V. Conclusion

In this work, we combine the idea of spatio-temporal pattern discovery with the non-parametric model NBMIM to recognize 3D action. The spatio-temporal pattern mining in

the proposed method ST-NBMIM is capable of discovering critical spatial joints and temporal stages of action instances simultaneously, which help not only increase the action recognition performance, but also physically explain each action recognized. We introduce the idea of mutual information into our framework. The involvement of negative stage-descriptors in mutual information calculation helps to improve the discriminative ability of action representation. Experiments show that ST-NBMIM can achieve better performance than baseline like ST-NBNN. Despite using only a linear classifier, the proposed method works surprisingly well on four benchmark datasets and beats some sophisticated end-to-end models on large scale dataset NTU RGB+D. Our results demonstrate the efficiency of the proposed spatio-temporal pattern discovery method for skeleton-based action recognition.

ACKNOWLEDGEMENT

This work is supported in part by Singapore Ministry of Education Academic Research Fund Tier 2 MOE2015- T2-2-114.

REFERENCES

[1] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. "Real-time human pose recognition in parts from single depth images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[2] Y. Du, W. Wang, and L. Wang "Hierarchical recurrent neural network for skeleton based action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[3] V. Veeriah, N. Zhuang, and G.-J. Qi "Differential recurrent neural networks for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[4] 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 *AAAI*, 2016.

[5] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang "NTU RGB+ D: A large scale dataset for 3D human activity analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[6] J. Liu, A. Shahroudy, D. Xu, and G. Wang "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *European Conference on Computer Vision*, 2016.

[7] 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 *European Conference on Computer Vision*, 2016.

[8] Z.Tu, J.Cao, Y.Li, and B.Li "MSR-CNN: applying motion salient region based descriptors for action recognition," in *International Conference on Pattern Recognition*, 2016.

[9] J. Yuan, Z. Liu, and Y. Wu "Discriminative subvolume search for efficient action detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[10] O. Boiman, E. Shechtman, and M. Irani "In defense of nearest-neighbor based image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[11] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes "Bilinear classifiers for visual recognition," in *Advances in Neural Information Processing Systems*, 2009.

[12] 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," in *Journal of Visual Communication and Image Representation*, 2014.

[13] G. Yu, Z. Liu, and J. Yuan "Discriminative orderlet mining for real-time recognition of human-object interaction," in *Asian Conference on Computer Vision*, 2014.

[14] J. Wang, Z. Liu, Y. Wu, and J. Yuan "Mining actionlet ensemble for action recognition with depth cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[15] X.Yang and Y.Tian "Super normal vector for activity recognition using depth sequences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[16] S. Sempena, N. U. Maulidevi, and P. R. Aryan "Human action recognition using dynamic time warping," in *International Conference on Electrical Engineering and Informatics*, 2011.

[17] W. Li, Z. Zhang, and Z. Liu "Expandable data-driven graphical modeling of human actions based on salient postures," in *IEEE Transactions on Circuits and Systems for Video Technology*, 2008.

[18] L. Xia, C.-C. Chen, and J. Aggarwal "View invariant human action recognition using histograms of 3d joints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2012.

[19] C. Wang, Y. Wang, and A. L. Yuille "Mining 3D Key-Pose-Motifs for Action Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[20] J. Weng, C. Weng, and J. Yuan "Spatio-Temporal Naive-Bayes Nearest-Neighbor (ST-NBNN) for Skeleton-Based Action Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[21] I. Kuzborskij, F. Maria Carlucci, and B. Caputo "When naive bayes nearest neighbors meet convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[22] M. Kusner, S. Tyree, K. Q. Weinberger, and K. Agrawal "Stochastic neighbor compression," in *Proceedings of the International Conference on machine learning*, 2014.

[23] C. Wang, J. Flynn, Y. Wang "Recognizing Actions in 3D Using Action-Snippets and Activated Simplices," in *AAAI*, 2016.

[24] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2012.

[25] D. Cai, X. He, J.-R. Wen, J. Han, and W.-Y. Ma "Support tensor machines for text categorization." *Department of Computer Science Technical Report No.2714, University of Illinois at Urbana-Champaign (UIUCDCS-R-2006-2714)*, 2006.

[26] W. Li, Z. Zhang, and Z. Liu "Action recognition based on a bag of 3d points," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2010.

[27] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy "Berkeley MHAD: A comprehensive multimodal human action database," in *IEEE Workshop on Applications of Computer Vision*, 2013.

[28] D. M. Mount and S. Arya "ANN: library for approximate nearest neighbour searching," in *Proc. Center for Geometric Computing Second Ann. Workshop Computational Geometry*, 1997, 33-40.

[29] M. Muja and D. G. Lowe "Fast approximate nearest neighbors with automatic algorithm configuration," in *VISSAPP*, 2009.

[30] C.-C. Chang and C.-J. Lin "LIBSVM: a library for support vector machines," in *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2011.

[31] M. Grant and S. Boyd "CVX: Matlab software for disciplined convex programming," 2008.

[32] R. Vemulapalli, F. Arrate, and R. Chellappa "Human action recognition by representing 3d skeletons as points in a lie group," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[33] P.Koniusz, A.Cherian, and F.Porikli "Tensor representations via kernel linearization for action recognition from 3D skeletons," in *European Conference on Computer Vision*, 2016.

[34] P. Wang, C. Yuan, W. Hu, B. Li, and Y. Zhang "Graph Based Skeleton Motion Representation and Similarity Measurement for Action Recognition," in *European Conference on Computer Vision*, 2016.

[35] I. Kapsouras and N. Nikolaidis "Action recognition on motion capture data using a dynemes and forward differences representation," in *Journal of Visual Communication and Image Representation*, 2014.

[36] S. Vantigodi and V. B. Radhakrishnan "Action recognition from motion capture data using meta-cognitive rbf network classifier," in *Journal of Visual Communication and Image Representation*, 2014.

[37] W. Li, L. Wen, M. Choo Chuah, and S. Lyu "Category-blind Human Action Recognition: A Practical Recognition System," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[38] G. Evangelidis, G. Singh, and R. Horaud "Skeletal quads: Human action recognition using joint quadruples," in *International Conference on Pattern Recognition*, 2014.

[39] H. Wang and L. Wang "Modeling Temporal Dynamics and Spatial Configurations of Actions Using Two-Stream Recurrent Neural Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[40] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot "Global context-aware attention lstm networks for 3d action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
[41] M. Liu, H. Liu, and C. Chen "Enhanced skeleton visualization for view invariant human action recognition," in *Pattern Recognition*, 2017.
[42] K. He, X. Zhang, S. Ren, and J. Sun "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
[43] Y. Du, Y. Fu, and L. Wang "Skeleton based action recognition with convolutional neural network," in *IAPR Asian Conference on Pattern Recognition*, 2015.
[44] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid "A New Representation of Skeleton Sequences for 3D Action Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
[45] C.-J. Lin, R. C. Weng, and S. S. Keerthi "Trust region newton method for logistic regression," in *Journal of Machine Learning Research*, 2008.
[46] S. Boyd and L. Vandenberghe "Convex optimization," *Cambridge university press*, 2004.
[47] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras "Interactive phrases: Semantic descriptionsfor human interaction recognition," in *IEEE transactions on pattern analysis and machine intelligence*, 2014.
[48] L. Zhou, W. Li, P. Ogunbona, and Z. Zhang "Semantic action recognition by learning a pose lexicon," in *Pattern Recognition*, 2017.
[49] P. Wang, Z. Li, Y. Hou, and W. Li "Action recognition based on joint trajectory maps using convolutional neural networks," in *Proceedings of the 2016 ACM on Multimedia Conference*, 2016.
[50] C. Li, Y. Hou, P. Wang, and W. Li "Joint Distance Maps Based Action Recognition With Convolutional Neural Networks," in *IEEE Signal Processing Letters*, 2017.
[51] K. Simonyan and A. Zisserman "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 2014.
[52] M.Ma, H.Fan, and K.M.Kitani "Going deeper into first-person activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
[53] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.
[54] J. Liu, B. Kuipers, and S. Savarese "Recognizing human actions by attributes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
[55] C. Wang, Y. Wang, and A. L. Yuille "An approach to pose-based action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
[56] I. Lillo, J. Carlos Niebles, and A. Soto "A Hierarchical Pose-Based Approach to Complex Action Understanding Using Dictionaries of Actionlets and Motion Poselets," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Junwu Weng (S'17) received his M.Eng. degree from South China University of Technology (SCUT), Guangdong, China, in 2015. Before that he graduated from the Talented Student Program of School of Electronics & Information, SCUT. He is currently pursuing the Ph.D. degree at the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore. His current research interests include computer vision, machine learning, as well as action and gesture analysis.

Chaoqun Weng (S'13) received the B.E. degree in computer science and technology from Nankai University, Tianjin, China, in 2010 and Ph.D. degree from the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore, in 2017. His current research interests include computer vision and machine learning.

Junsong Yuan (M'08-SM'14) received Ph.D. from Northwestern University in 2009 and M.Eng. from National University of Singapore in 2005. Before that, he graduated from the Special Class for the Gifted Young of Huazhong University of Science and Technology (HUST), Wuhan, China, in 2002. He joined Nanyang Technological University (NTU) as a Nanyang Assistant Professor (NAP) in 2009, and was an associate professor at School of Electrical and Electronics Engineering (EEE), NTU. He is now an associate professor at Department of Computer Science and Engineering (CSE), the State University of New York, Buffalo, USA.

His research interests include computer vision, video analytics, gesture and action analysis, large-scale visual search and mining, etc. Since 2009, he has been PI and Joint-PI of over 8 million SGD grants, and has published over 100 papers in top conferences and journals, with research work licensed by industry and government agency. He received 2016 Best Paper Award from IEEE Trans. on Multimedia, Doctoral Spotlight Award from IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'09), Nanyang Assistant Professorship from NTU, and Outstanding EECS Ph.D. Thesis award from Northwestern University. He is currently Senior Area Editor of Journal of Visual Communication and Image Representation (JVCI), Associate Editor of IEEE Trans. on Image Processing (T-IP), IEEE Trans. on Circuits and Systems for Video Technology (T-CSVT), and served as Guest Editor of International Journal of Computer Vision (IJCV). He is Program Co-chair of ICME18 and VCIP15, and Area Chair of ACM MM18, ACCV1814, ICPR1816, CVPR17, ICIP17 etc.

Zicheng Liu (SM'05-F'15) received his Ph.D. in computer science from Princeton University in 1996. He got his B.S. degree in mathematics from HuaZhong Normal University, Wuhan, China, in 1984, and his M.S. in Operations Research from the Institute of Applied Mathematics, Chinese Academy of Sciences, in 1989. Before joining Microsoft Research, he worked at Silicon Graphics, Inc. as a member of technical staff for two years, where he developed the trimmed NURBS tessellator shipped in both OpenGL and the OpenGL Optimizer.

Current research interests include human activity recognition, 3D face modeling and animation, and multimedia signal processing. He has worked on a variety of topics including Steiner trees, average case complexity, linked figure animation, and trimmed NURBS tessellation for large CAD model visualization.

Liu has served in the technical committee for many international conferences. He was a member of the Audio and Electroacoustics Committee of IEEE Signal Processing Society. He is the chair of the Multimedia Systems and Applications Technical Committee of IEEE CAS society. He is a steering committee member of IEEE Transactions on Multimedia. He is the Editor-in-Chief of Journal of Visual Communications and Image Representation, and an associate editor of Machine Vision and Applications. He served as a guest editor of IEEE Transactions on Multimedia, and a guest editor of IEEE Multimedia Magazine. He is an affiliate professor in the department of Electrical Engineering, University of Washington. He was an IEEE distinguished lecturer from 2015-2016. He is a fellow of IEEE.