Fusing Geometric Features for Skeleton-Based Action Recognition Using Multilayer LSTM Networks

Songyang Zhang, Yang Yang, Jun Xiao, Xiaoming Liu, Yi Yang, Di Xie, and Yueting Zhuang

Abstract—**Recent skeleton-based action recognition approaches achieve great improvement by using recurrent neural network (RNN) models. Currently, these approaches build an end-to-end network from coordinates of joints to class categories and improve accuracy by extending RNN to spatial domains. First, while such well-designed models and optimization strategies explore relations between different parts directly from joint coordinates, we provide a simple universal spatial modeling method perpendicular to the RNN model enhancement. Specifically, according to the evolution of previous work, we select a set of simple geometric features, and then seperately feed each type of features to a three-layer LSTM framework. Second, we propose a multistream LSTM architecture with a new smoothed score fusion technique to learn classification from different geometric feature streams. Furthermore, we observe that the geometric relational features based on distances between joints and selected lines outperform other features and the fusion results achieve the state-of-the-art performance on four datasets. We also show the sparsity of input gate weights in the first LSTM layer trained by geometric features and demonstrate that utilizing joint-line distances as input require less data for training.**

Index Terms—**Action recognition, skeleton, geometric feature, LSTM, score fusion.**

I. Introduction

Action recognition is an intensively researched topic, aiming to identify human actions from input sensor streams. Three common types of input in this task are RGB [1], [2], depth [3], [4] and skeleton [5], [6]. Specifically, RGB videos are the most popular input and have been widely studied due to its convience of data capturing. Nontheless, capturing information in the 3D space, where human actions happens, still has advantages in some aspects since its representation is much richer. For instance, motion capture systems extract accurate 3D joint positions using markers and high precision camera arrays. Admittedly, it is not designed for recognizing actions in daily life. In that, Kinect sensor provides a cost-effective daily living solution, which generates relatively reliable skeletons from depth maps. In this paper, we focus on recognizing actions from skeleton inputs rather than RGB or depth for three reasons. First, skeletons are invariant to viewpoint or appearance, thus they suffer less intra-class variances compared to RGB or depth. Second, skeletons are high-level information describing human's movement only. Without the interference of some irrelevent signals, the learning of action recognition itself can be greatly simplified. Third, it is demenstrated by Yao *et al.* [7] that skeleton-based features outperform appearance-based features by using the same classifier on the same dataset.

Recent exploration of recurrent neural network (RNN) [10]–[12] made a great influence on processing video sequences. Several works [5], [6], [8], [9] successfully built well-designed multilayer RNNs for recognizing action based on skeletons. However, while promising recognition performances are observed using these methods, they have three common limitations: (1) their inputs are limited to the coordinates of joints; (2) the RNN models are sophisticated and have a high complexity; and (3) the relations learned from these models are rarely self-explanatory and intuitive to human.

In this paper, considering that LSTM is suitable for modeling dependence in the temporal domain, we focus on feeding LSTM with rich spatial domain features by exploring geometric relations between joints. Our method is inspired by the evolution of recent skeleton-based action recognition using RNN models. Du *et al.* [5] model the relations of neighboring parts (two arms, two legs and torso) with handcrafted RNN subnets and ignore the relations between non-adjacent parts (Fig. 1(a)), which is remedied by two methods in different ways. Zhu *et al.* [6] add a mixed-norm regularization term to the fully connected LSTMs cost function which can exploit relations between non-adjacent parts (Fig. 1(b)). Another solution is introduced by Shahroudy *et al.* [8], who separate the memory cell to part-based subcells and the non-adjacent parts relations are learned over the

Manuscript received May 9, 2017; revised November 30, 2017; accepted January 16, 2018. Date of publication February 5, 2018; date of current version August 14, 2018. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0203001, in part by the National Natural Science Foundation of China under Grant 61572431, in part by the Zhejiang Natural Science Foundation under Grant LZ17F020001, in part by the Key R&D Program of Zhejiang Province under Grant 2018C01006, and in part by the Joint Research Program of ZJU & Hikvision Research Institute. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Elisa Ricci. *(Corresponding author: Yang Yang.)*

S. Zhang, Y. Yang, J. Xiao, and Y. Zhuang are with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China (e-mail: zhangsongyang@zju.edu.cn; yangya@zju.edu.cn; junx@cs.zju.edu.cn; yzhuang@cs.zju.edu.cn).

X. Liu is with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: liuxm@cse.msu.edu).

Y. Yang is with the Centre for Quantum Computation and Intelligent Systems, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: yi.yang@uts.edu.au).

D. Xie is with the Hikvision Research Institute, Hangzhou 310051, China (e-mail: xiedi@hikvision.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2018.2802648

1520-9210 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

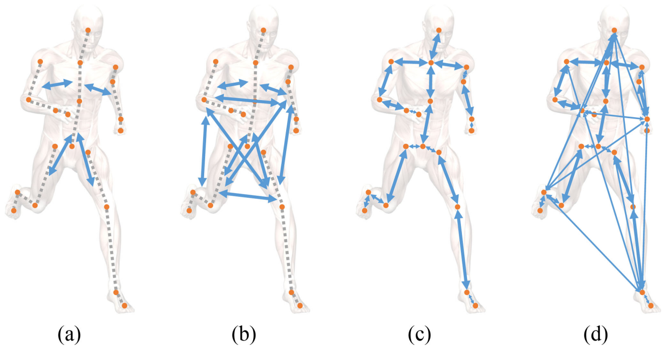


Fig. 1. Evolution of geometric relation modeling for RNN-based action recognition. Orange points are joints, gray dotted lines connect several joints represent body parts, blue bidirectional arrows represent relations between parts or joints. (a) relations between adjacent parts [5]; (b) relations among all parts [6], [8]; (c) relations between adjacent joints [9]; (d) relations among all joints (ours).

concatenated part-based memory cells (Fig. 1(b)). Admittedly, these two methods successfully explore relations between body parts, but dividing body into parts might not be well funded. A more elaborate division is proposed by Liu *et al.* [9]. They focus on adjacent joints and design a sophisticated model traversing skeleton as a tree (Fig. 1(c)). However, Liu *et al.* [9] ignore the relations between non-adjacent joints. The evolution of geometric relation modeling indicates that adding relations between non-adjacent joints may further enhance the performance.

Based on this intuition, we enumerate eight geometric features to describe relations among all joints inspired by several previous work [13]–[15] (Fig. 1(d)). This kind of feature describes geometric relations between specific joints in a single frame or a short frame sequence, which is typically used for indexing and retrieval of motion capture data. We evaluate their performances on LSTM. Experimentally, we find joint-line distances outperform others on four datasets. To further understand our deep LSTM network, we visualize the weights learned in the first LSTM layer and find the weight of the input gate is sparse, which means a small subset of joint-line distances is sufficiently representative. Our method has three advantages. First, our simple geometric feature is superior than the joint coordinates in all evaluations, which implies future work shall pay attention to this type of geometric features. Second, the fact that we achieve the state-of-the-art performance using the standard LSTM model [16] indicates that our finding is applicable to perpendicular development in RNN models. Third, the geometric features describing relations between joints, lines and planes are easy for human to comprehend.

Feeding LSTM with a single type of feature is effective in some cases. Such features only describe raw data from one aspect and the complete semantics might be carried by a set of features. Thus, the fusion of multiple feature streams may boost the classification performance. Similar research has been conducted in recognizing actions in RGB videos. Karen *et al.* [17] first adapt a two-stream convolutional network to this task and evaluate the performance on three simple score fusion methods. Ng *et al.* [18] and Christoph *et al.* [19] further explore different feature-level fusing strategies in combining two CNN architectures during training. Wu *et al.* [20] propose a new score fusing strategy utilizing the class relationships in the data after the network training. Shi *et al.* [21] adapt previous architecture to a three-stream CNN network by introducing an additional feature. Inspired by the above research, we concatenate two LSTM networks trained by different geometric features in different layers during training. We also test the performance of average score fusion after training. Experimentally, we find that fusing during training is inferior for skeleton-based action recognition while the average score fusion outperforms any single network. However the average score fusion is still not well funded, since it ignores the different score distributions in each stream. Thus we propose a new score fusion strategy where the distribution of each stream is first smoothed with a hyperparameter, and then the weights of each stream are learnt in a joint framework. Furthermore, our method alleviates the overfitting problem occured in the single stream.

It should be mentioned that this paper is an extension of our conference paper [22]. Compared to the previous version, we achieve better performance by integrating all individually trained models with a smoothed score fusion technique. The main contributions of our work are summarized as follows:

1) We introduce an integrated system combing the advantages of geometric features and stacked LSTM model for skeleton-based action recognition. We demonstrate the proposed JL_d requires less training samples than using joint coordinates.
2) We propose a novel score fusion method to effectively fuse the outputs of the individual networks. The method first smoothes the confidence scores and then learns the weights of individual network streams adaptively.
3) Our model is simpler than many well-designed LSTM architectures, and yet it achieves state-of-art action recognition accuracy in widely used four benchmark datasets.

The remainder of the paper is organized as follows. In Section II, we introduce the related work on skeleton based action recognition. In Section III, we model human spatial information via eight types of geometric relational features. Experimental results and discussions are presented in Section IV. Finally, we conclude the paper in Section V.

II. Related Work

In this section, we briefly review the existing works that closely relate to the proposed method, including two categories of approaches representing relational geometric features and skeleton-based action recognition.

A. Geometric Features

Many prior works recognize actions from direct measures of joint parameters of the human body, e.g., angles, position, orientation, velocity, acceleration [23]–[26]. Muller *et al.* [27] introduce a class of Boolean features expressing geometric relations between certain body points of a pose. Yao *et al.* [7] develop a variety of pose-based features including distance between joints, distance between joints and planes, and velocity of joints, etc. Yun *et al.* [13] extend [7]'s idea and modify pose-based features that are suitable for two persons' interaction. Chen *et al.* [14]

enumerate 9 types of geometric features and concatenate all of them as pose and motion representations. Vinagre *et al.* [28] propose a relational geometric feature called Trisarea, which describes the geometric correspondence between joints by means of the area of the defined triangle. Vemulapalli *et al.* [29] utilize rotations and translations to represent the 3D geometric relationships of body parts in Lie group. *In contrast, our work extends geometric features to action recognition via deep learning methods.*

B. RNN for Skeleton-Based Action Recognition

Recently, several RNN models have also shown promising performance in this task. Du *et al.* [5] propose an end-to-end hierarchical RNN with handcrafted subnets, where the raw positions of human joints are divided to five parts according to human structure, and then are separately fed into five bidirectional RNNs. As the number of layers increases, the representations extracted by the subnets are hierarchically fused to a higher-level representation. Zhu *et al.* [6] find such methods ignore the inherent co-occurrences of joints, and thus design a softer division method. They add a mixed-norm regularization term to fully connected LSTMs cost function, which is capable to exploit the groups of co-occurring and discriminative joints for better action recognition. An internal dropout mechanism is also introduced for stronger regularization in the network, which is applied to all the gate activations. Shahroudy *et al.* [8] separate the memory cell to part-based sub-cells and push the network towards learning the long-term context representations individually for each part. The output of the network is learned over the concatenated part-based memory cells followed by the common output gate. Liu *et al.* [9] focus on adjacent joints in a skeleton, which split body into smaller parts than prior work. They extend LSTM to spatial-temporal domains with a tree-based traversal method. *Compared to learning features with well-designed LSTM networks, we show that properly defining hand-crafted features for a basic LSTM network can be superior.*

C. Fusion Methods in RGB-Based Action Recognition

Several studies made their efforts on fusing multiple features in RGB-based action recognition tasks. Yang *et al.* [30] design a hierarchical regression model to exploit the information derived from each type of feature, which is then collaboratively fused in order to obtain a multimedia semantic concept classifier. Simonyan *et al.* [17] evaluated three fusion methods on fusing two-stream ConvNets: training a fully-connected layer on top of two streams, averaging the softmax scores and training a linear SVM using softmax scores. They found that SVM-based fusion are superior to the average fusion, while the fusion with fully-connected layers is impracticable due to the overfitting problem. Wu *et al.* [20] tested more methods, including SVM-based early fusion, SVM-based late fusion, multiple kernel learning, early fusion with neural networks, late fusion with neural networks, multimodal deep Boltzmann Machines, RDNN, and their proposed Regularized Feature Fusion Network. Shi *et al.* [21] propose a novel feature and further evaluate the late fusion performance of the combination among different features. *To the best of our knowledge, our approach is the first work evaluating fusion methods on skeleton-based action recognition task. Also, The proposed method provides another solution in adjusting the score distribution of fusion problem.*

III. Our Approach

Many traditional computer vision systems rely on hand-crafted features and well-designed optimization algorithm [31]–[35]. However, recent deep learning-based systems utilizing features learned from raw data have demonstrated great success on various vision tasks, e.g., video classification [1], [18], [36], [37] and image description [1], [38]. Such data-driven features, without the guidance of domain knowledge, may run into the overfitting problem, especially in the cases of small amount of training data, or the difference data distributions between training and testing data. To this end, we hypothesize that properly designed hand-crafted features could be valuable to deep learning-based methods, in contrast to the typical raw data input. Specifically, our skeleton-based action recognition approach utilizes a set of relational geometric features. Similar features have been used in motion retrieval applications [14].

A. Learning With A Single Type of Feature

In order to put our proposed approach into context, we first review Long-Short Term Memory neuron (LSTM). LSTM is an advanced structure which overcomes the RNN's vanishing gradient problem [39] and is able to model long-term dependencies. Different from RNN's simple neuron, a LSTM neuron contains an input gate, an output gate, a cell and a forget gate that determines how the information flow into and out of the neuron. One LSTM layer is shown in Fig. 2. In our approach, we do not use in-cell connections [16] (also called peepholes) as no improvement has shown in recent experiments [40]. In summary, components in LSTM neurons are calculated as follows:

$$\begin{aligned}
\mathbf{i_t} &= \sigma(\mathbf{W_{xi}}\mathbf{x_t} + \mathbf{W_{hi}}\mathbf{h_{t-1}} + \mathbf{b_i}),\\
\mathbf{f_t} &= \sigma(\mathbf{W_{xf}}\mathbf{x_t} + \mathbf{W_{hf}}\mathbf{h_{t-1}} + \mathbf{b_f}),\\
\mathbf{u_t} &= \tanh(\mathbf{W_{xu}}\mathbf{x_t} + \mathbf{W_{hu}}\mathbf{h_{t-1}} + \mathbf{b_u}),\\
\mathbf{c_t} &= \mathbf{i_t} \circ \mathbf{u_t} + \mathbf{f_t} \circ \mathbf{c_{t-1}},\\
\mathbf{o_t} &= \sigma(\mathbf{W_{xo}}\mathbf{x_t} + \mathbf{W_{ho}}\mathbf{h_{t-1}} + \mathbf{b_o}),\\
\mathbf{h_t} &= \mathbf{o_t} \circ \tanh(\mathbf{c_t}),
\end{aligned} \tag{1}$$

where $\mathbf{W}$ and $\mathbf{b}$ are the weight matrices and bias vectors respectively. The symbol σ is the sigmoid function. The operation $\circ$ is an element-wise multiplication.

Taking advantage of multilayer LSTM (a.k.a. the stacked or deep LSTM) architectures, we build our model shown in Fig. 3. Specifically, the first LSTM layer takes geometric features as the input $\mathbf{x_t}$, and the upper LSTM layer takes the output $\mathbf{h_t}$ from the lower LSTM layer as the input $\mathbf{x_t}$. This variation of LSTM enables the higher layers to capture longer-term dependencies of the input sequence. For the purpose of providing confidence scores at the last time step n, a softmax layer is placed on the highest LSTM layer L to estimate the probability of a sequence

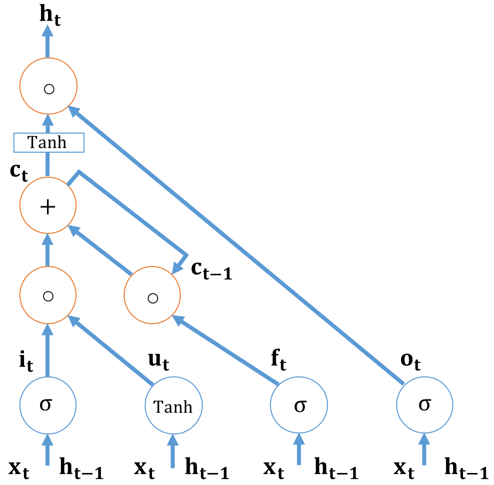

$\mathbf{h_t}$
○
Tanh
$\mathbf{c_t}$
+
○
○
$\mathbf{c_{t-1}}$
$\mathbf{i_t}$
$\mathbf{u_t}$
$\mathbf{f_t}$
$\mathbf{o_t}$
σ
Tanh
σ
σ
$\mathbf{x_t}$ $\mathbf{h_{t-1}}$ $\mathbf{x_t}$ $\mathbf{h_{t-1}}$ $\mathbf{x_t}$ $\mathbf{h_{t-1}}$ $\mathbf{x_t}$ $\mathbf{h_{t-1}}$

Fig. 2. The structure of a LSTM layer. The input $\mathbf{x_t}$ of the first layer is the geometric feature. For higher layers, the input $\mathbf{x_t}$ is the output $\mathbf{h_t}$ from the previous layer at the same time instance.

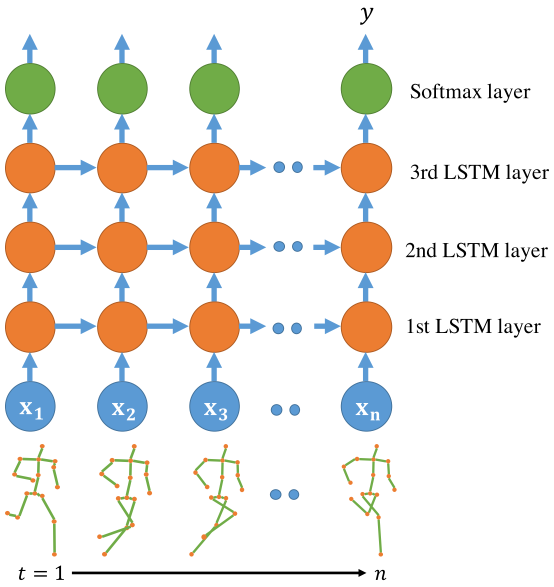


Fig. 3. The LSTM architecture in our approach, where each orange dot is one LSTM layer as Fig. 2.

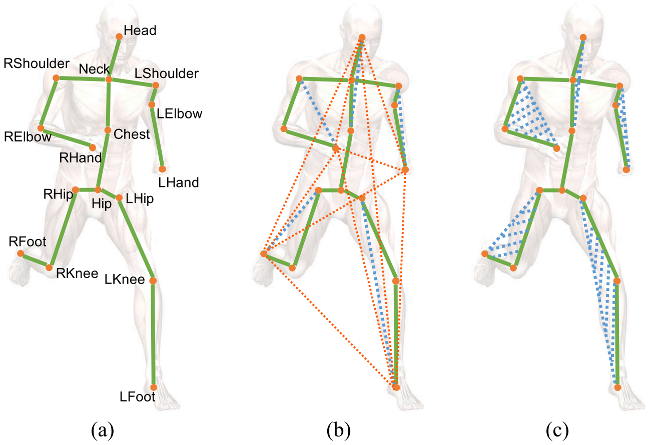
$\mathbf{X}$ belonging to the class C_k as:

$$p(C_k|\mathbf{X}) = \frac{\exp(z_k)}{\sum_{i=1}^{C}\exp(z_i)}, \tag{2}$$

$$\mathbf{z} = \mathbf{W_z}\mathbf{h}_\mathbf{n}^\mathbf{L} + \mathbf{b_z}. \tag{3}$$

B. Spatial Modeling via Geometric Feature

In this section we consider spatial modeling using geometric features in a single frame. We adopt a typical human skeleton model with 16 joints for illustration. Any two of joints form a line and any three of joints form a plane. Thus, there are $C_{16}^2 = 120$ lines and $C_{16}^3 = 560$ planes in total. The pair-wise combination of joints, lines, and planes form geometric features. Fig. 4(a) shows the skeleton model. Table I summarizes the numbers of all possible features, where duplicated features are removed when identical lines or planes are determined by the same set of joints.



(a) (b) (c)

Fig. 4. (a) A skeleton model. Orange dots represent joints and green lines represent limbs. (b) Lines. 15 Green, 5 blue and 10 red lines are three types of lines. (c) Planes.

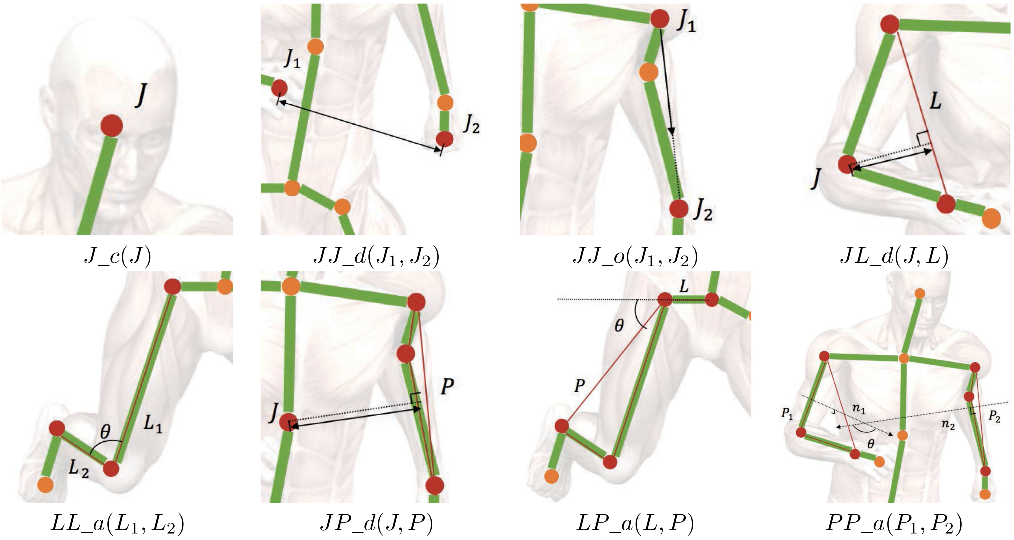
TABLE I
NUMBER OF ALL POSSIBLE FEATURES WHEN USING THE HUMAN MODEL WITH 16 JOINTS

	Joint	Line	Plane
Joint	120	1680	7280
Line		7140	65 520
Plane			156 520

Since the number of combinations is extremely large, using all of them in the learning could be very time consuming. Therefore, we need to select several important lines and planes in order to reduce the computational cost. Specifically, we select the following joints, lines and planes on the 16-joint human skeleton, as shown in Fig. 4. The reason we select these joints are explained in IV-D1.

- Joint. Each joint J is encoded with its coordinate (J_x, J_y, J_z).
- Line. $L_{J_1\to J_2}$ is the line from joint J_1 to J_2, if one of the following three constraints is satisfied:
  1. J_1 and J_2 are directly adjacent in the kinetic chain.
  2. If one of J_1 and J_2 is at the end of skeleton chain (one of Head, L(R)Hand or L(R)Foot), the other one can be two steps away in the kinetic chain (Head→Chest, RHand→RShoulder, LHand→LShoulder, RHip→RFoot, and LHip→LFoot). This produces five lines.
  3. If both J_1 and J_2 are at the end of skeleton chain, $L_{J_1\to J_2}$ is a line. This produces ten lines.
- Plane. $P_{J_1\to J_2\to J_3}$ is the plane determined by the triangle with vertices J_1, J_2, and J_3. We only consider five planes corresponding to the torso, arms and legs, namely: $P_{Chest\to Neck\to Head}$, $P_{LShoulder\to LElbow\to LHand}$, $P_{RShoulder\to RElbow\to RHand}$, $P_{LHip\to LKnee\to LFoot}$ and $P_{RHip\to RKnee\to RFoot}$.

As LSTM are designed to learn variation in time, we enumerate eight types of geometric features that are encoded in one



$J\_c(J)$ $JJ\_d(J_1, J_2)$ $JJ\_o(J_1, J_2)$ $JL\_d(J, L)$

$LL\_a(L_1, L_2)$ $JP\_d(J, P)$ $LP\_a(L, P)$ $PP\_a(P_1, P_2)$

Fig. 5. Eight feature types. Note that for each feature only the relevant joints, lines, and planes are drawn in red.

TABLE II
DEFINITIONS OF EIGHT GEOMETRIC FEATURES

Name	Symbol	Definition	Description
Joint Coordinate	J_c	$J\_c(J) = (J\_x, J\_y, J\_z)$	The 3D coordinate of the joint J.
Joint-Joint Distance	JJ_d	$JJ\_d(J_1, J_2) = \left\| \overrightarrow{J_1 J_2} \right\|$	The Euclidean distance between joint J_1 to J_2.
Joint-Joint Orientation	JJ_o	$JJ\_o(J_1, J_2) = \text{unit}(\overrightarrow{J_1 J_2})$	The orientation from joint J_1 to J_2, represented by the unit length vector $\overrightarrow{J_1 J_2}$.
Joint-Line Distance	JL_d	$JL\_d(J, L_{J_1 \to J_2}) = 2S_{\triangle J J_1 J_2} / JJ\_d(J_1, J_2)$	The distance from joint J to line $L_{J_1 \to J_2}$. The calculation is accelerated with Helen formula.
Line-Line Angle	LL_a	$LL\_a(L_{J_1 \to J_2}, L_{J_3 \to J_4})$ $= \arccos(JJ\_o(J_1, J_2)^T \quad \odot JJ\_o(J_3, J_4))$	The angle (0 to π) from line $L_{J_1 \to J_2}$ to $L_{J_3 \to J_4}$.
Joint-Plane Distance	JP_d	$JP\_d(J, P_{J_1 \to J_2 \to J_3})$ $= (J\_c(J) - J\_c(J_1)) \quad \odot JJ\_o(J_1, J_2) \otimes JJ\_o(J_3, J_4)$	The distance from joint J to plane $P_{J_1 \to J_2 \to J_3}$.
Line-Plane Angle	LP_a	$LP\_a(L_{J_1 \to J_2}, P_{J_3 \to J_4 \to J_5})$ $= \arccos(JJ\_o(J_1, J_2)) \odot JJ \quad \_o(J_3, J_4) \otimes JJ\_o(J_3, J_5)$	The angle (0 to π) between line $L_{J_1 \to J_2}$ and the normal vector of plane $P_{J_3 \to J_4 \to J_5}$.
Plane-Plane Angle	PP_a	$PP\_a \quad (P_{J_1 \to J_2 \to J_3}, P_{J_4 \to J_5 \to J_6})$ $= \arccos(JJ\_o(J_1, J_2) \otimes JJ\_o(J_1, J_3)$ $\odot JJ\_o(J_3, J_4) \otimes JJ\_o(J_3, J_5))$	The angle (0 to π) between the normal vectors of planes $P_{J_1 \to J_2 \to J_3}$ and $P_{J_4 \to J_5 \to J_6}$.

Note that Hips coordinate is excluded as it is fixed as $(0, 0, 0)$. On the other hand, the y coordinate of Hip in the world coordinate frame reflects the absolute height of body and is informative in some cases (e.g., discerning jumping in the air), and hence is included. $\odot$ is the dot product. $\otimes$ is the cross product of two vectors.

pose and are independent of time, as shown in Fig. 5. In contrast, features like joints velocity and acceleration consider spatial variations over the time. Specific definitions of the features are shown in Table II. In addition, we remove duplicated features due to symmetry or degeneration. For example, $JJ\_d(J_1, J_2)$ is symmetric to $JJ\_d(J_2, J_1)$, and $JL\_d(J, L_{J_1 \to J_2})$ degenerates to zero if J is the same as J_1 or J_2.

C. Integrating Models via Smoothed Score Fusion

In order to take advantages of multiple trained models and further improve the recognition performance, we propose a model fusion method. We begin by introducing the findings from several exploratory experiments including input fusion, fully-connected fusion and average fusion. These methods inspired us to propose our smoothed score fusion method.

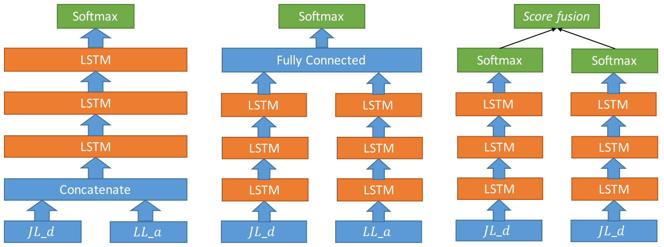


Fig. 6. The illustration of three exploration fusion methods. The input fusion concatenates two features before feeding to network. The fully-connected fusion utilizes a fully-connected layer to combine the two outputs of last LSTM layers. The average fusion computes the average scores of all streams and selects the label with the highest score as prediction.

1) Exploratory Findings: As shown in Fig. 6, a common way to fuse two streams is to concatenate their layers. In order

TABLE III
COMPARISON OF THREE FUSION METHODS

Method	JL_d	LL_a	Fusion
input fusion	70.26%	66.90%	64.79%
fully-connected fusion	–	–	68.70%
average fusion	–	–	72.44%

The networks are trained in the NTU-RGB+D cross-subject setting, by fusing the JL_d and LL_a features.

TABLE IV
THE ACCURACY BY FEATURE TYPES AND CLASSES

Label	J_c	JJ_d	JL_d	LL_a
Jumping up	94.95%	79.50%	83.03%	72.46%
Clapping hands	20.44%	33.94%	41.88%	58.76%
Taking off shoes	30.91%	39.78%	33.09%	43.22%

to find the best fusion strategy, we concatenate the layers of different depths and evaluate their performance. As shown in Table III, we analyze the results as following:

First, input fusion is worse than any single stream. Previous work that combines multiple kinds of geometric features as input also shows no improvement compared to a single type of feature [13]. This may be caused by the weak ability of LSTM in distinguishing useful information from many different types of, and somewhat less discriminative, features.

Second, the fully-connected fusion has almost the same performance as the single stream. In our opinion, the low accuracy is due to the over-fitting problem of the single network, which will be shown in the experiment section. Since two overfitted networks generate very low loss values, adding an extra fully connected layer may not improve its overall performance. The similar phenomena has also be resported in recognizing RGB videos in [17], [20].

Third, the average fusion of softmax scores produces much better results than the input fusion and fully-connected fusion, and it is the only method outperforms any single stream. It is reasonable since this method does not require a training process and thus avoids over-fitting.

2) Our Smoothed Score Fusion: According to the exploration above, we find that when fusing multiple streams, we should limit the information exchanging among streams. This means we shall avoid fusion of final features or predictions during training. To fulfill this, each stream should have their own softmax and loss layers. A simple and widely used score fusion strategy is to assign average weights to confidence scores of each feature [17], [18]. However, since different features may not contribute equally to the final decision, they should have different weights. Taking the cross-subject task in NTU-RGB+D dataset as an example,Table IV shows the accuracy of models trained from some geometric features, respectively. For the action “jumping”, the feature J_c achieves the best prediction performance, and is much better than other relational features, because“jumping” is highly relevant to the height between floor and the center of body. On the other hand, for the action “clapping hands”, relations corresponding to two hands are more discriminative, and therefore J_c has worse performance than others. Similarly, other features achieve good performance for some actions while performs worse for others. This example shows that different features do not contribute equally to the final predictions and their weights should not be identical.

TABLE V
THE MAX SCORE $\max_i t_{ij}$ BY FEATURE TYPES AND CLASSES

Label	J_c	JJ_d	JL_d	LL_a
Jumping up	0.9091	0.8622	0.9078	0.8711
Clapping hands	0.5707	0.7687	0.8509	0.8052
Taking off shoes	0.6453	0.8201	0.8715	0.8515

Another issue in the score fusion is the difference of smoothness among the score distributions from different models. t_{ij} is the confidence score of each of the 16 506 testing samples according to feature i and class j, as shown in (4),

$$t_{ij} = \frac{1}{M_{ij}} \sum_{\mathbf{X} \in C_{ij}} (p_i(C_k|\mathbf{X})), \tag{4}$$

where C_{ij} is the set of samples that are predicted as class j by model i, M_{ij} is the number of samples in the set C_{ij}, $p_i(C_k|\mathbf{X})$ is the softmax value of class k when input the sample $\mathbf{X}$ to model i, which is defined in (2).

Table V demonstrates that $\max_i t_{ij}$ vary widely among different classes and features. Though JL_d has higher confidence scores than others in general, it does not show better performance in some actions, such as “Clapping hands” and “Taking off shoes”. In order to alleviate the consequence in such conditions, we consider to use a smoothed score distribution in the final fusion algorithm.

To solve the problem, a weighted fusion with a special factor T is given below. The outputs of log-softmax from different models are denoted as $(z_1, z_2, ...z_N)$. Here N is the number of models and each z_i equals to $(z_{i1}, z_{i2}, ..., z_{iC})$ where C is the number of classes. q_{ij} in (5) is a processed probability distribution over classes using a scalar T, where T is normally set to 1. Using a larger value of T produces a softer probability distribution over classes. $(\alpha_1, \alpha_2, ...\alpha_N)$ is a set of weights which average the softmax score from different streams by their value. o is the final probability weighted by $\alpha_1, \alpha_2, ...\alpha_N$.

$$q_{ij} = \frac{\exp\,(z_{ij}/T)}{\sum_{k=1}^{C} \exp(z_{ik}/T)}, \tag{5}$$

$$o = \sum_{i=1}^{N} \alpha_i q_i. \tag{6}$$

The objective of network training is to minimize the cross entropy shown below, where i is the ground-truth index:

$$\min_{\alpha_1, \alpha_2, ...\alpha_N} -\log \left(\frac{\exp o_i}{\sum_{j=1}^{N} \exp o_j} \right). \tag{7}$$

A general framework is shown in Fig. 7. It should be mentioned that the weights of each LSTM network are static and

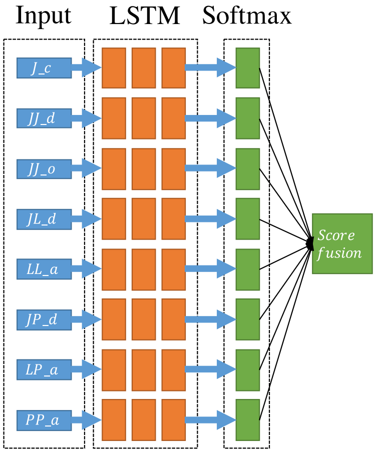


Fig. 7. Illustration of the proposed framework, where each stream represent a 3-layer LSTM as Fig. 3.

they are not adjusted in this part of training. This can be seen as a simple way to alleviate the overfitting problem.

D. Implementation Details

Joint coordinates are preprocessed in a way similar to the scheme in Shahroudy *et al.* [8], which transforms all joint coordinates from the camera coordinate system to the body coordinate system. The original point of the body coordinate system is translated to the “center of hips”, and then rotates the X axis parallel to the 3D vector from “right shoulder” to “left shoulder” and Y axis towards the 3D vector from “center of shoulders” to “center of hips”. The Z axis is fixed as the new $X \times Y$. After that, we normalize all 3D points based on the summation of skeletal chains distances. Since other features such as distances and angles are invariant to the coordinate system, they are calculated in the camera coordinate system in order to reduce the deviation introduced by the coordinate transformation.

In our system, we use a 3-layer LSTM implemented by torch7 bindings for NVIDIA CuDNN. The learning rate is set to 0.01 with a classic momentum of 0.9 [41]. We set an upper bound on the $L2$ norm of the incoming weight vector for each individual neuron [42]. We also adopt common techniques such as adding weight noises and early stopping. Though there are variations in terms of the sequence length, joint number, and data acquisition equipment for different datasets, we use the same parameter settings above. This demonstrates the robustness of our method to the parameter settings, as it achieves promising results on all the datasets with the same configuration.

IV. Experimental Results

In this section, we conduct extensive experiments to demonstrate our action recognition algorithm based on geometric relational features. In order to conduct a convincing evaluation, the algorithm should be tested from different perspectives. Following this view, four datasets are selected including NTU-RGB+D dataset [8], SBU-Kinect dataset [13], UT-Kinect dataset [23], and Berkeley MHAD dataset [43]. A preview of each dataset is shown in Table VI. Then, we visualize and analyze the learned weights of the first LSTM layer. We also demenstrate the relation between the performance and the amount of training samples in using different geometric features. Furthermore, we discuss about the overfitting issue of the proposed model.

TABLE VI
Comparison of Different Dataset Characteris

	SBU-Kinect	NTU-RGB+D	UT-Kinect	Berkeley MHAD
(1)	-	✓	-	-
(2)	✓	-	-	-
(3)	-	-	-	✓
(4)	-	✓	✓	-
(5)	-	✓	-	-

We choose these datasets based on the following considerations, including: (1) single (-) or multi-view action dataset (✓), (2) single (-) or multi-person action dataset (✓), (3) captured by Kinect (-) or motion capture system (✓), (4) the model is tested using different subjects (✓) or not (-), (5) the model is tested in a different view point (✓) or not (-). Though NTU-RGB+D has both single and multi-person actions, prior work [2], [8] treat them as single-person actions, and we follow this designation.

A. Dataset Description

SBU-Kinect dataset [13]. The SBU Kinect dataset is a Kinect captured human action recognition dataset depicting two-person interaction. In most interactions, one person is acting and the other person is reacting. The entire dataset has a total of 282 sequences belonging to 8 classes of interactions performed by 7 participants. Each person has 15 joints. The smoothed positions of joints are used during the experiment. The dataset provides a standard experimental protocol with 5-fold cross validation.

NTU-RGB+D dataset [8]. To the best of our knowledge, NTU-RGB+D dataset is the largest RGBD database for action recognition, which is captured by Kinect v2 in various views containing 4 different data modalities per sample. It consists of 56 880 action samples of 60 different classes including daily activities, interactions and medical conditions performed by 40 subjects aged between 10 and 35. The large intra-class and view point variations make it very challenging to distinguish its actions. Due to the large amount of samples, this dataset is suitable for applying deep learning based methods. In order to evaluate the effectiveness of scale-invariant and view-invariant features, it provides two evaluation protocols, cross-subject and cross-view. A 25-joint human model is provided.

UT-Kinect Dataset [23]. The UT-Kinect dataset is captured by a single stationary Kinect containing 200 sequences of 10 classes performed by 10 subjects in varied views. Each action is recorded twice for every subject and each frame in a sequence contains 20 skeleton joints. We follow the half-vs-half protocol proposed in [44], where half of the subjects are used for training and the remaining for testing.

Berkeley MHAD [43]. Berkeley MHAD is captured by a motion capture system. Skeleton joint coordinates provided by this type of equipment are of high precision and can accurately represent the performer’s movements. It contains 659 sequences of 11 classes. Actions are performed by 7 male and 5 female subjects in the 23-30 years of age except for one elderly subject. All the subjects performed 5 repetitions of each action, which

TABLE VII
DIMENSIONS OF GEOMETRIC FEATURES IN FOUR DATASETS

	J_c	JJ_d	JJ_o	JL_d	LL_a	JP_d	LP_a	PP_a
SBU-Kinect	86	435	1305	1624	1653	270	550	45
NTU-RGB+D	73	300	900	897	741	110	180	10
UT-Kinect	58	190	570	612	561	85	155	10
Berkeley MHAD	103	595	1785	1551	1081	160	230	10

correspond to about 82 minutes of total recording time. There are 35 joints accurately extracted according to the 3D marker trajectory. We follow the protocol in [5], in which 384 sequences corresponding to the first 7 subjects are used for training and 275 sequences of the remaining 5 subjects are for testing.

B. Dataset Related Parameters

1) Feature Dimension: Since the number of joints are not the same among different datasets, we list the dimension of each feature in Table VII. Also, we do not follow the definition of J_c in SBU-Kinect. Because two persons' skeletons are recorded simultaneously, we transform the camera coordinates to the person with the largest joint's variance in location.

2) Smooth Filtering: Many previous works [5], [6] use specific designed filters to smooth the joint coordinates since they are estimated by primitive pose estimation algorithms with noise. However, during our experiments, filtering does not appear to improve performance. Thus, for simplicity, all reported results of our algorithm do not involve smooth filtering.

3) Sequential Downsampling: The frame rate of Microsoft Kinect is about 30 FPS, which is adequate for daily usage. The frame rate of motion capture systems is even higher, which achieves 480 FPS, in order to capture accurate human movement for specialized usage, e.g., animation. Considering that actions can be recognized by human in a much lower frequency, according to the Nyquist sampling theorem, we can downsample the sample rate with a fixed interval to reduce the computational cost as well as avoid losing effective motion information. The downsampling rates of different datasets vary according to their original sampling rates. For instance, one frame is sampled in every 16 frames in Berkeley MHAD and every 8 frames in NTU-RGB+D. Since UT-Kinect and SBU-Kinect limit their original frame rates to 15 FPS, downsampling is not necessary. For a fair comparison, our proposed models are trained with the same downsampling rate as the respective previous works.

4) Hidden Layer Size: We evaluate how the number of neurons in LSTM influences the performance. We find that the neuron size has little influence on the final results, as long as the number of neurons is roughly proportional to the input feature dimension. For example, the relation between JL_d-based performance and the number of neurons is shown in Fig. 8, where the action recognition rate changes very little despite the large change of the number of neurons. The numbers of neurons used in the experiment are listed in Table VIII. All three layers of LSTM contain the same number of neurons.

Another noted difference is that the lines between wrist and hand are ignored for simplicity in Berkeley MHAD, since the markers of these two joints may be sticked so close that they always appear to have the same position.

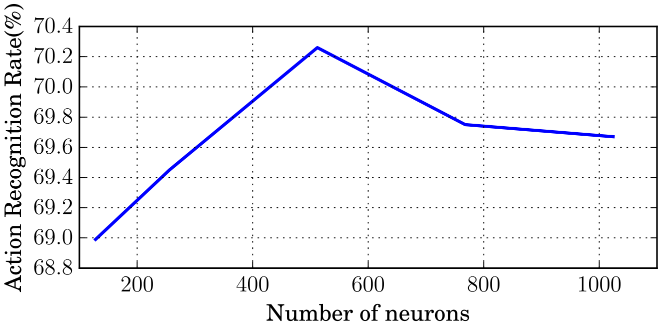


Fig. 8. The action recognition rates of the JL_d feature on NTU-RGB+D, with different numbers of neurons.

C. Performance Comparison

We summarize the comparison of the action recognition rates on all four benchmark datasets in Table IX. We choose the baseline algorithms that are widely reported in prior work, such as [6], [8], [9]. ST-LSTM [9] achieves the highest accuracy in four datasets among all previous works. Each ST-LSTM neuron contains two hidden units, one for the previous joint and the other for the previous frame. Each ST-LSTM neuron corresponds to one of the skeletal joints. During training, neurons' states are transformed in a tree structure based on skeletal connections. A new gating mechanism within LSTM is developed to handle the noise in raw skeleton data. Contrast to the comprehensive design in ST-LSTM, our approach further improves the performance on all datasets, except the already saturated Berkeley MHAD. This improvement is especially remarkable in the context that we simply use geometric features on top of the conventional LSTM architecture. Meanwhile, we make further improvement by fusing the softmax scores of eight individually trained networks, which means our proposed fusion method provides an effective solution for integrating different models in action sequences.

1) SBU-Kinect: We follow the experimental protocol proposed in [13] and perform 5-fold cross validation on this dataset. We compare our proposed method with Yun *et al.* [13], Ji *et al.* [45], Li *et al.* [46], Du *et al.* [5], Zhu *et al.* [6], Liu *et al.* [9] as well as eight proposed features and their fusion results. As shown in Table IX, we achieve superior performance over other methods by a large margin in this dataset. Since there are more relations in two persons' interaction than action performed by a single person, using hand-crafted geometric feature is easier to discover relations than using joint coordinates. A

TABLE VIII
THE NUMBER OF NEURONS IN FOUR DATASETS

	J_c	JJ_d	JJ_o	JL_d	LL_a	JP_d	LP_a	PP_a
SBU-Kinect	128	512	512	512	512	256	512	64
NTU-RGB+D	73	300	512	512	512	110	180	10
UT-Kinect	58	190	570	612	561	85	155	10
Berkeley MHAD	128	512	1024	1024	1024	256	256	32

TABLE IX
PERFORMANCE COMPARISON

Method	SBU-Kinect	NTU-RGB+D		UT-Kinect	Berkeley MHAD
		cross-subject	cross-view		
Yun *et al.* [13]	80.3%	–	–	–	–
Ji *et al.* [45]	86.9%	–	–	–	–
CHARM [46]	83.9%	–	–	–	–
Li *et al.* [15]	94.12%	–	–	–	–
HOG^2 [24]	–	32.24%	22.27%	–	–
Super Normal Vector [3]	–	31.82%	13.61%	–	–
Skeleton Joint Features [44]	–	–	–	87.9%	–
HON4D [47]	–	30.56%	7.26%	–	–
Skeletal Quads [48]	–	38.62%	41.36%	–	–
FTP Dynamic Skeletons [49]	–	60.23%	65.22%	–	–
Elastic functional coding [50]	–	–	–	94.9%	–
Kapsouras *et al.* [51]	–	–	–	–	98.18%
Chaudhry *et al.* [25]	–	–	–	–	100%
Ofli *et al.* [26]	–	–	–	–	95.37%
Lie Group [29]	–	50.08%	52.76%	93.6%	97.58%
HBRNN-L [5]	80.35%	59.07%	63.97%	–	100%
P-LSTM [8]	–	62.93%	70.27%	–	–
Co-occurrence LSTM [6]	90.41%	–	–	–	100%
ST-LSTM [9]	93.3%	69.2%	77.7%	95.0%	100%
JTM [52]	–	73.4%	75.2%	–	–
Liu *et al.* [53]	–	75.97%	82.56%	–	–
J_c	77.55%	59.32%	70.01%	90.91%	98.18%
JJ_d	97.54%	67.41%	80.39%	87.88%	97.45%
JJ_o	95.13%	**73.23**%	80.18%	84.85%	96.00%
$\boldsymbol{JL\_d}$	**99.02%**	71.88%	**85.09%**	**95.96**%	**100**%
LL_a	84.74%	66.77%	81.00%	94.95%	98.18%
JP_d	71.92%	58.05%	69.29%	74.75%	67.64%
LP_a	64.43%	58.64%	61.82%	78.79%	34.18%
PP_a	21.52%	28.57%	30.21%	27.27%	31.64%
Max	95.28%	74.46%	85.28%	93.94%	98.91%
Average	98.30%	76.13%	87.37%	94.95%	**100**%
Linear SVM	97.50%	74.87%	86.37%	94.95%	**100**%
RBF-kernel SVM	97.50%	75.86%	87.48%	95.96%	**100**%
weighted average($T = 1$)	98.30%	76.20%	87.46%	94.95%	**100**%
weighted average($T = 4$)	**99.33**%	**76.43**%	**87.69**%	**95.96**%	**100**%
weighted average($T = 9$)	99.33%	76.42%	87.66%	95.96%	**100**%
weighted average($T = 16$)	99.33%	76.27%	87.58%	95.96%	**100**%
weighted average($T = 25$)	99.33%	76.20%	87.48%	95.96%	**100**%
weighted average($T = 36$)	99.33%	76.18%	87.47%	95.96%	**100**%

The performances of baseline skeleton-based methods are obtained from[8], [9].

slight improvement is achieved after applying our score fusion methods.

2) NTU-RGB+D: We follow the experimental protocol proposed in [8] on this dataset. Different from our previous paper, since NTU-RGB+D involves both single person actions and two person actions, we adopt a two-person model same as SBU-Kinect. For single person performing actions, the first person joints are copied to the second person. Lines and planes are selected within each individual, no additional lines or planes are added. We compare our proposed method with Ohn-Bar *et al.* [24], Yang *et al.* [3], Oreifej *et al.* [47], Evangelidis *et al.* [48], Hu *et al.* [49], Vemulapalli *et al.* [29], Du *et al.* [5], Shahroudy *et al.* [8], Liu *et al.* [9] as well as eight proposed features and their fusion results. As shown in Table IX, some network trained by a single feature achieves better performance than previous work, for example, the new JJ_o shows best performance in the cross-subject setting, while the new JL_d shows best in the cross-view setting. After smoothing the softmax scores of each trained model and averge them with weights, our method significantly surpasses the state-of-the-art precision

TABLE X
TOP 10 MOST ACCURATELY RECOGNIZED ACTIONS FOR EACH FEATURE

Rank	J_c	JJ_d	JL_d	LL_a
1	falling	wear jacket	take off jacket	take off jacket
2	jump up	stand up	wear jacket	wear jacket
3	stand up	take off jacket	stand up	stand up
4	wear jacket	cross hands	hugging	pickup
5	take off jacket	cheer up	falling	falling
6	hopping	falling	pickup	sitting down
7	pickup	pickup	cross hands	cheer up
8	sitting down	hugging	sitting down	hugging
9	cross hands	salute	salute	cross hands
10	nod head/bow	sitting down	fold hands	handshaking

by 0.46% in cross-subject setting and 5.13% in the cross-view setting. Given the scale of this large benchmark dataset, we like to point out that these are substantial margins and they demonstrate that our proposed method provides an effective scale-invariant and view-invariant model.

In order to evaluate each kind of feature, Table X lists the top 10 accuracy per class of four typical features. We find that J_c outperforms other features in some actions such as "jump up" and "hopping". Because the height from the floor to the center of body is a discriminative factor in this case and only J_c contains such information. We also find that the ranks of features like JJ_d, JL_d and LL_a are quite similar, since they all reflect relations among joints. Each kind of feature describes human's movement from different aspects, thus it explains why integrating multiple streams can make further improvement.

As shown in Table IX, some common score fusion methods as well as our proposed smoothed fusion are evaluated. Average fusion and RBF-kernel SVM achieve competitive results in both settings. Weighted average fusion gains little benefits compared to the average fusion. We observe a noticeable improvement when changing the smooth factor T from 1 to 4 and further increase of T has little influence on the final results. When T equals 4, we achieve the best performance in both the cross-subject setting and the cross-view setting.

The confusion matrix of JL_d in the cross-subject setting is shown in Fig. 9. We can see that our model performs well on most of the actions. the misclassification is not avoidable in two kinds of situations: inaccurate skeleton estimation and human-object interaction. First, inaccurate skeleton estimation may cause some tiny body movement indistinguishable. For instance, the action "rub two hands together" (Action 34) is often misclassified to "clapping" (Action 10) , since Kinect provides rough hand coordinate estimation. Second, some human-object interactions are innately indistinguishable. For example, the action "wear a shoe" (Action 16) is often misclassified to "take off a shoe" (Action 17) while the action "wear on glasses" (Action 18) is misclassified to "take off glasses" (Action 19). Distinguishing such actions is very difficult without using RGB or other informations.

3) UT-Kinect: We follow the experimental protocol proposed by Zhu *et al.* [44] on this dataset. We compare our proposed method with Zhu *et al.* [44], Anirudh *et al.* [50], Vemulapalli *et al.* [29], Liu *et al.* [9] as well as eight proposed features and their fusion results. Similar to NTU-RGB+D, this dataset is recorded in a variety of view angles and is evaluated in a cross-subject setting, but with less classes. As shown in Table IX, JL_d outperforms all other methods. It is worth noting that the averge fusion may not be benefital in all situations. According to our observation, in this dataset, JJ_d and LL_a usually generate peaky distribution. On the contrary, J_c and JL_d provide much smoother distribution. A typical score distribution is shown in Fig. 10. Due to the peaky distribution of LL_a, the average fusion performs the same as LL_a. After smoothing all features' scores, the performance is improved.

4) Berkeley MHAD: We follow the experimental protocol proposed in [43] on this dataset. We compare our proposed method with Kapsouras *et al.* [51], Chaudhry *et al.* [25], Ofli *et al.* [26], Vemulapalli *et al.* [29], Du *et al.* [5], Zhu *et al.* [6], Liu *et al.* [9] as well as eight proposed features and their fusion results. As shown in Table IX, using JL_d can achieve the 100% accuracy, the same as several previous works [5], [6], [9], [25]. Other features such as J_c, JJ_d, JJ_o and LL_a also achieve competitive results. Since this dataset is already saturated, we

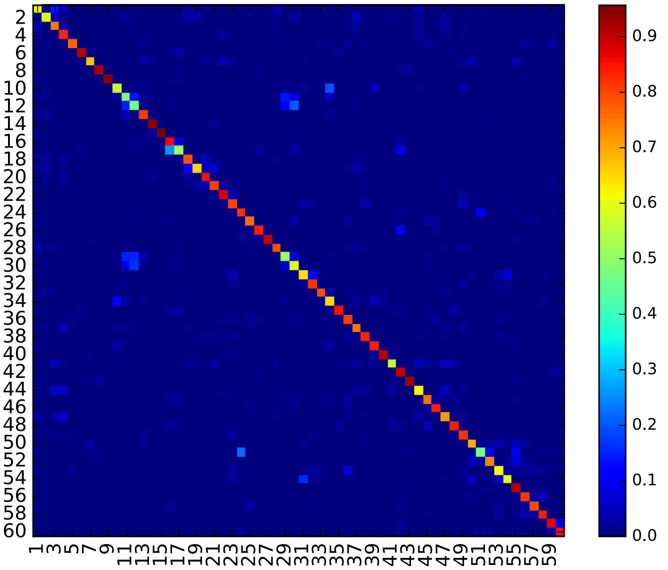


Fig. 9. The confusion matrix of JL_d in the NTU-RGB+D cross-subject setting.

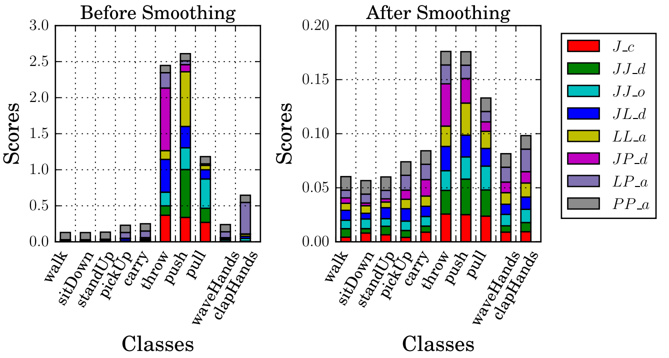


Fig. 10. A typical score distribution in UT-Kinect dataset where the true label is "throw". Average fusion will misclassify the label "throw" as the label "push".

TABLE XI
SORTED JOINTS IN A DESCENDING ORDER, WHERE THE NUMBER IN PARENTHESES IS ITS RANK

	Group 1	Group 2	Group 3
Right Arm	RHand(1)	RElbow(5)	RShoulder(10)
Left Arm	LHand(2)	LElbow(8)	LShoulder(9)
Right Leg	RFoot(3)	RKnee(7)	RHip(12)
Left Leg	LFoot(4)	LKnee(6)	LHip(13)
Torso	Head(11)	Neck(14)	Chest(15)

Hip is ignored since it is the original point.

TABLE XII
THE ACCURACY OF EACH VARIANCE-BASED GROUP

	Group 1	Group 2	Group 3
Accuracy	82.39%	81.50%	80.42%

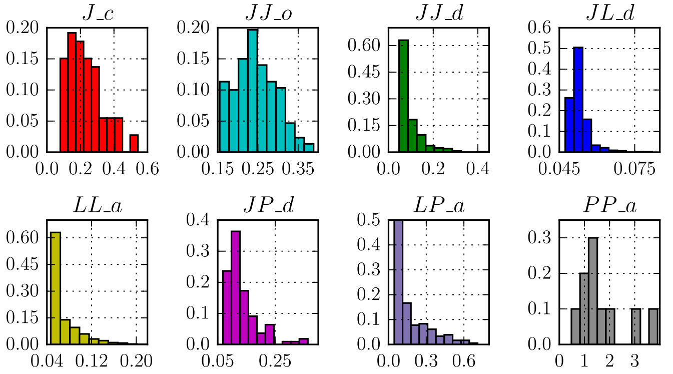


Fig. 11. The histogram of s_i calculated by (8). x axis represents the value of s_i and y axis represents the percentage of s_i with the same value.

use it for the completeness of the experiments, rather than demonstrating the advantages of the proposed fusion method.

D. Discussion

1) Joint Selection Analysis: The selection is primarily based on the assumption that the body joint's variance of locations is indicative to its representativeness or discriminativeness. For example, the end sites(head, hands, foots) are more variant than the nonterminal sites. The average variance of each joint in Fig. 4 is calculated in NTU-RGB+D in a body coordinate system. In order to validate our assumption, we partition all joints into three groups by ranking the location variance of each joint and test their performance in a JL_d model with a cross-view setting. The ranking and grouping results are shown in Table XI. The performance results are shown in Table XII. This shows that indeed the more variance in locations (e.g., group 1), the better the recognition performance is.

2) Feature Discriminative Analysis: To further understand the effect of different features on the deep LSTM network, we visualize the weights learned in the first LSTM layer using histograms. All experiments in this section are conducted on NTU-RGB+D dataset with the cross-subject setting. As shown in Fig. 11, each element represents the average weight among LSTM neurons calculated by (8),

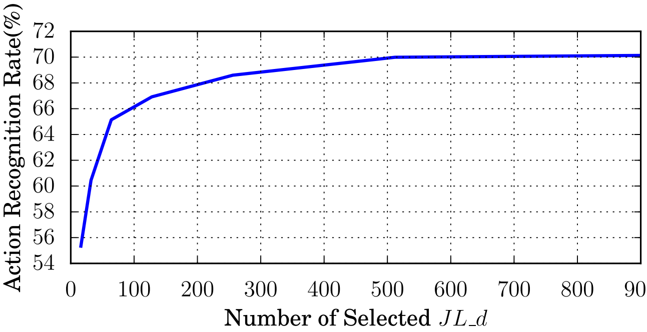


Fig. 12. Action recognition rates with different top JL_d feature numbers on NTU-RGB+D.

$$s_i = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{W_{xi}}(i,j)\| \; (j = 1, 2...M), \tag{8}$$

where $\mathbf{W_{xi}}(i,j)$ is essentially $\mathbf{W_{xi}}$ corresponding to the ith neuron and jth input in the first LSTM layer shown in (1), N is the number of neurons in the first LSTM layer, and M is the dimension of the input feature.

From Fig. 11, we observe that the weight distributions of JJ_d, JL_d, LL_a, JP_d and LP_a are relatively sparse. In contrast, J_c, JJ_o, and PP_a do not show such a sparsity because they have a lower level of abstraction and more intra-dependencies among feature elements compared to features such as JL_d. Given the sparsity, we hypothesize that only a small set of geometric features is sufficiently discriminative.

To verify our hypothesis, we rank all feature elements in JL_d based on s_i and test their recognition rates on the selected top 16, 32, 64, 128, 256 and 512 elements with the highest average weight, respectively. We find that the recognition rate increases rapidly when the feature number is small (<64), and after that the increasing is slowed down. The results are shown in Fig. 12. When the feature number is above 500, the performance does not show notable improvement with the increasing feature number. Therefore, this shows that a small set of features is effective. In practice, if there is a validation set, we could learn the feature subset and only use it for testing. In addition, four JL_d feature elements with the highest weights are: J_{head} to $L_{\text{base of spine}\rightarrow\text{middle of spine}}$, $J_{\text{left wrist}}$ to $L_{\text{left hand}\rightarrow\text{left thumb}}$, $J_{\text{right wrist}}$ to $L_{\text{left wrist}\rightarrow\text{left ankle}}$, and $J_{\text{middle of spine}}$ to $L_{\text{head}\rightarrow\text{neck}}$. This is reasonable since most of actions in the NTU-RGB+D dataset correspond to hands and head. Taking an example of "drinking water", the distance from the hand to spine and the distance from the head to spine change simultaneously.

3) Feature Robustness Analysis: Kinect often presents some miss detected or wrong posture estimated skeletons. In order to investigate whether the proposed method is robust also in the case where the skeleton is not given for granted, we randomly select N joints as missing joints. Each missing joint is completed by the average coordinate value of its connected joints. Experimentally, we find that as the number of joints increase, our smoothed fusion method can still outperforms the average fusion, which is shown in Fig. 13.

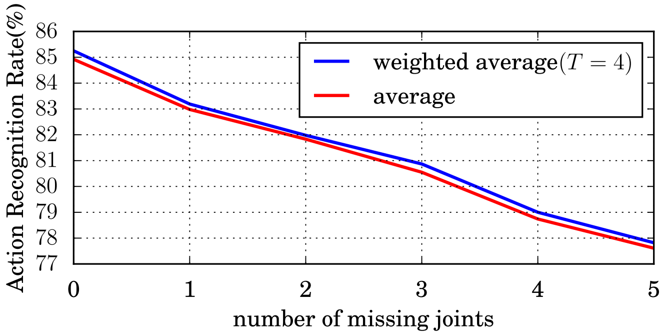


Fig. 13. performance under different number of missing joints. The model is trained in NTU-RGB+D cross-view setting.

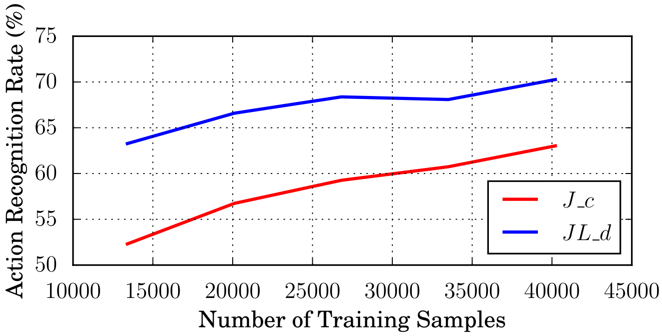


Fig. 15. Influence of training data samples. The performance of the LSTM model using JL_d decreases slower than using J_c, with decreasing training samples.

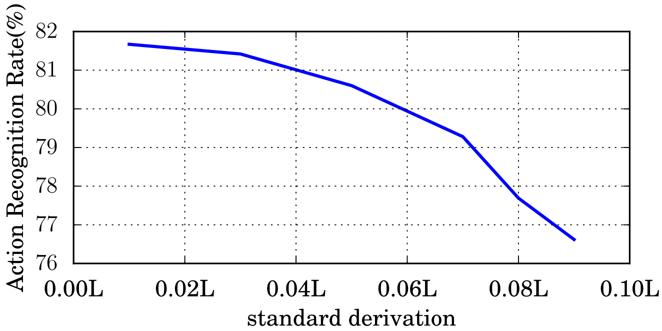


Fig. 14. performance under different noise level. The model is trained on JL_d in the NTU-RGB+D cross-view setting.

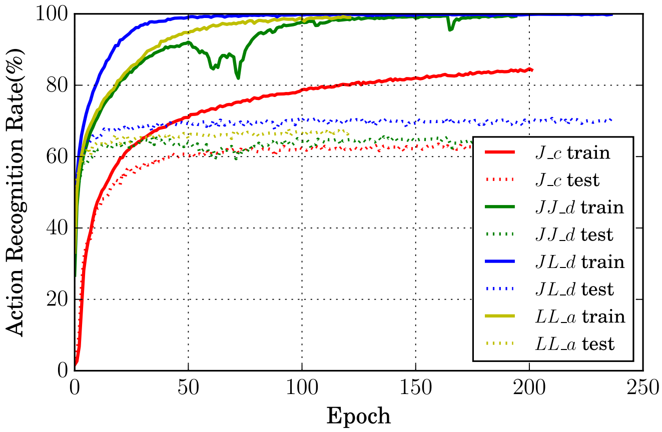


Fig. 16. Performance of four features in the NTU-RGB+D cross-subject setting. Solid and dashed lines represent the training and testing accuracies respectively.

we also add white gaussian noise to each joint in the testing set with different standard deviation, std(std is the same for all directions) from $0.01L$ to $0.09L$, where L is the average length of bones connecting to the joint. As shown in Fig. 14, while there is performance drop as the noise increases, the amount of drop is small and this shows the noise-resistance of our algorithm.

4) Smoothing Factor: The smoothing factor(T) is not computed, but a constant number. By testing several different T values, we found that $T = 4$ achieves the best performance in most cases. Futhermore, for $T > 4$, the performance are quite similar, so designing an algorithm to optimize T seems not necessary.

5) Data Sample Size: Most hand-crafted features demand fewer data samples for training than the raw data input. This is also true when we use LSTM as a learning model. We observe that using JL_d requires fewer samples for training compared to J_c as shown in Fig. 15.

6) Overfitting Problem: Experimentally we observe that our hand-crafted features suffer from the overfitting problem in large datasets such as NTU-RGB+D, despite achieving the state-of-the-art performance. We compare three overfitted features (JJ_d, JL_d and LL_a) with J_c and show their training and testing accuracies in Fig. 16. As we can see, these features achieve higher accuracies than J_c in both the training set and testing set, which confirms that geometric features are more discriminative than J_c. Due to J_c's weak discriminative ability, optimization is rather difficult, which is the potential reason why J_c is less overfitted than others.

TABLE XIII
THE RESULTANT WEIGHTS OF ALL STREAMS AFTER TRAINING, ON EACH DATASET

Method	SBU-Kinect	NTU-RGB+D		UT-Kinect	Berkeley MHAD
		cross-subject	cross-view		
J_c	0.99	7.31	6.96	0.54	1.62
JJ_d	1.02	7.23	7.17	0.54	1.65
JJ_o	1.02	**7.67**	7.16	0.53	1.65
JL_d	1.02	7.66	7.20	0.54	1.65
LL_a	1.02	7.65	**7.24**	**0.55**	1.65
JP_d	1.01	7.23	6.78	0.52	1.65
LP_a	1.00	5.79	5.78	0.52	1.54
PP_a	0.99	5.87	5.79	0.52	1.62

7) Score Fusion: Table XIII shows the weight α of different streams trained by different datasets. We find that the higher weight mostly comes from streams with better performance. Noted that the little difference of weights in SBU-Kinect, UT-Kinect and Berkeley MHAD are caused by their small amount of training samples. Because only 8 parameters are updated during training, we observe that the training accuracy stop increasing after 2 or 3 epochs, thus we simply stop training after 10 epochs. Since the learning rate is also fixed, the magnitude of weights only depends on the number of updates and the changing value in each update, thus it explains the different magnitudes among different datasets. Take an example of NTU-RGB+D dataset, which has a large number of samples and low accuracy compared to other datasets, the weights are updated many times with large changing value in one epoch. It explains the large average weight and the large variance of weights in NTU-RGB+D dataset.

V. Conclusions

In this paper, we summarize the evolution of previous work on RNN-based 3D action recognition using skeletons and hypothesize that exploring relations among all joints may lead to better performance. Following the intuition, we design eight geometric relational features and evaluate them in a 3-layer LSTM network. Extensive experiments show the distance between joints and selected lines outperforms other features.

In order to integrate all individual networks, we explore several fusion methods and find that information exchanging between streams during training has a negative impact to the performance. On the other hand, simply averaging scores ignore the score distribution and contribution of different streams. Based on previous two reasons, we propose a new smoothed score fusion. The state-of-art performance on four selected publicly available datasets demonstrate the effectiveness of the proposed methods. Moreover, we show that using a subset of joint-line distances can achieve comparative results and using joint-line distances as input requires fewer samples for training compared to joint coordinate input.

References

[1] J. Donahue *et al.*, “Long-term recurrent convolutional networks for visual recognition and description,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 2625–2634.

[2] Y. Zhang, X. Liu, M.-C. Chang, W. Ge, and T. Chen, “Spatio-temporal phrases for activity recognition,” in *Proc. Eur. Conf.Comput. Vision*, 2012, pp. 707–721.

[3] X. Yang and Y. Tian, “Super normal vector for activity recognition using depth sequences,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2014, pp. 804–811.

[4] P. Wang, W. Li, Z. Gao, J. Zhang, C. Tang, and P. O. Ogunbona, “Action recognition from depth maps using deep convolutional neural networks,” *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 4, pp. 498–509, Aug. 2016.

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

[6] W. Zhu *et al.*, “Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks.” in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 3697–3703.

[7] A. Yao, J. Gall, G. Fanelli, and L. J. Van Gool, “Does human action recognition benefit from pose estimation?” in *Proc. Brit. Mach. Vision Conf.*, 2011, pp. 67.1–67.11.

[8] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, “NTU RGB+D: A large scale dataset for 3D human activity analysis,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 1010–1019.

[9] 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. Vision*, 2016, pp. 816–833.

[10] Z. Zhao *et al.*, “Social-aware movie recommendation via multimodal network learning,” *IEEE Trans. Multimedia*, vol. 20, no. 2, pp. 430–440, Feb. 2018.

[11] D. Xu *et al.*, “Video question answering via gradually refined attention over appearance and motion,” in *Proc. ACM Multimedia*, 2017, pp. 1645–1653.

[12] H. Zhang, M. Wang, R. Hong, and T.-S. Chua, “Play and rewind: Optimizing binary representations of videos by self-supervised temporal hashing,” in *Proc. ACM Multimedia*, 2016, pp. 781–790.

[13] 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. Vision Pattern Recognit. Workshops*, 2012, pp. 28–35.

[14] C. Chen *et al.*, “Learning a 3D human pose distance metric from geometric pose descriptor,” *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 11, pp. 1676–1689, Nov. 2011.

[15] M. Li and H. Leung, “Multiview skeletal interaction recognition using active joint interaction graph,” *IEEE Trans. Multimedia*, vol. 18, no. 11, pp. 2293–2302, Nov. 2016.

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

[17] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in *Proc. Conf. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[18] J. Yue-Hei Ng *et al.*, “Beyond short snippets: Deep networks for video classification,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 4694–4702.

[19] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Convolutional two-stream network fusion for video action recognition,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 1933–1941.

[20] Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, and X. Xue, “Multi-stream multi-class fusion of deep networks for video classification,” in *Proc. ACM Multimedia*, 2016, pp. 791–800.

[21] Y. Shi, Y. Tian, Y. Wang, and T. Huang, “Sequential deep trajectory descriptor for action recognition with three-stream CNN,” *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1510–1520, Jul. 2017.

[22] S. Zhang, X. Liu, and J. Xiao, “On geometric features for skeleton-based action recognition using multilayer LSTM networks,” in *Proc. IEEE Winter Conf. Appl. Comput. Vision*, 2017, pp. 148–157.

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

[24] E. Ohn-Bar and M. Trivedi, “Joint angles similarities and HOG2 for action recognition,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops*, 2013, pp. 465–470.

[25] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal, “Bio-inspired dynamic 3D discriminative skeletal features for human action recognition,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops*, 2013, pp. 471–478.

[26] 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. Visual Commun. Image Representation*, vol. 25, no. 1, pp. 24–38, Jan. 2014.

[27] M. Müller, T. Röder, and M. Clausen, “Efficient content-based retrieval of motion capture data,” *ACM Tran. Graphics*, vol. 24, no. 3, pp. 677–685, Jul. 2005.

[28] M. Vinagre, J. Aranda, and A. Casals, “A new relational geometric feature for human action recognition,” in *Proc. Int. Conf. Inform. Control Autom. Robot.*, 2015, pp. 263–278.

[29] 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. Vision Pattern Recognit.*, 2014, pp. 588–595.

[30] Y. Yang *et al.*, “Multi-feature fusion via hierarchical regression for multimedia analysis,” *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 572–581, Apr. 2013.

[31] S. Wang *et al.*, “Semi-supervised multiple feature analysis for action recognition,” *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 289–298, Feb. 2014.

[32] X. Cai, W. Zhou, L. Wu, J. Luo, and H. Li, “Effective active skeleton representation for low latency human action recognition,” *IEEE Trans. Multimedia*, vol. 18, no. 2, pp. 141–154, Feb. 2016.

[33] Y. Yang *et al.*, “Discriminative multi-instance multitask learning for 3d action recognition,” *IEEE Trans. Multimedia*, vol. 19, no. 3m, pp. 519–529, Mar. 2017.

[34] Y. Yan *et al.*, “Image classification by cross-media active learning with privileged information,” *IEEE Trans. Multimedia*, vol. 18, no. 12, pp. 2494–2502, Dec. 2016.

[35] Y. Zhang, W. Ge, M.-C. Chang, and X. Liu, “Group context learning for event recognition,” in *Proc. IEEE Winter Conf. Appl. Comput. Vision*, 2012, pp. 249–255.

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

[37] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," in *Proc. Int. Conf. Learn. Representations Workshops*, 2016.

[38] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[39] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[40] T. M. Breuel, "Benchmarking of LSTM networks," arXiv:1508.02774, to be published.

[41] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1139–1147.

[42] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv:1207.0580, to be published.

[43] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Berkeley MHAD: A comprehensive multimodal human action database," in *Proc. IEEE Winter Conf. Appl. Comput. Vision*, 2013, pp. 53–60.

[44] Y. Zhu, W. Chen, and G. Guo, "Fusing spatiotemporal features and joints for 3D action recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops*, 2013, pp. 486–491.

[45] Y. Ji, G. Ye, and H. Cheng, "Interactive body part contrast mining for human interaction recognition," in *Proc. Int. Conf. Multimedia Expo Workshops*, 2014, pp. 1–6.

[46] W. Li, L. Wen, M. Choo Chuah, and S. Lyu, "Category-blind human action recognition: A practical recognition system," in *Proc. Int. Conf. Connected Veh.*, 2015, pp. 4444–4452.

[47] O. Oreifej and Z. Liu, "Hon4D: Histogram of oriented 4D normals for activity recognition from depth sequences," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2013, pp. 716–723.

[48] G. Evangelidis, G. Singh, and R. Horaud, "Skeletal quads: Human action recognition using joint quadruples," in *Proc. Int. Conf. Pattern Recognit.*, 2014, pp. 4513–4518.

[49] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang, "Jointly learning heterogeneous features for RGB-D activity recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 5344–5352.

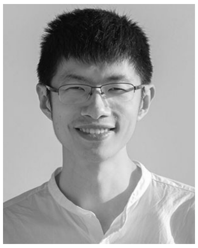
[50] R. Anirudh, P. Turaga, J. Su, and A. Srivastava, "Elastic functional coding of human actions: From vector-fields to latent variables," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 3147–3155.

[51] I. Kapsouras and N. Nikolaidis, "Action recognition on motion capture data using a dynemes and forward differences representation," *J. Visual Commun. Image Representation*, vol. 25, no. 6, pp. 1432–1445, Aug. 2014.

[52] C. Li, Y. Hou, P. Wang, and W. Li, "Joint distance maps based action recognition with convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 24, no. 5, pp. 624–628, May 2017.

[53] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognit.*, vol. 68, pp. 346–362, 2017.

Songyang Zhang received the B.S. degree from Southeast University, Nanjing, China, in 2015. He is currently working toward the M.S. degree at the School of Computer Science, Zhejiang University, Hangzhou, China. His research interests include computer vision, deep learning, and action recognition.

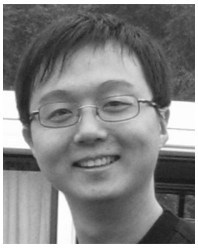


Yang Yang received the Ph.D. degree from Tsinghua University, Beijing, China in 2016. He is currently an Assistant Professor with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China. He visited Cornell University in 2012, and University of Leuven. He has authored and coauthored more than 20 papers in top conference and journals, such as KDD, WWW, AAAI, TKDD, ICDM, etc. His research interests include mining deep knowledge from large-scale social and information networks.

Jun Xiao received the Ph.D. degree in computer science and technology from the College of Computer Science, Zhejiang University, Hangzhou, China, in 2007. He is currently an Professor with the College of Computer Science, Zhejiang University. His current research interests include computer animation, multimedia retrieval, and machine learning.

Xiaoming Liu received the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2004. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Michigan State University (MSU), East Lansing, MI, USA. Before joining MSU in Fall 2012, he was a Research Scientist with General Electric Global Research. His research interests include computer vision, machine learning, and biometrics. As a coauthor, he was a recipient of Best Industry Related Paper Award runner-up at ICPR 2014, the Best Student Paper Award at WACV 2012 and 2014, and the Best Poster Award at BMVC 2015. He has been the Area Chair for numerous conferences, including FG, ICPR, WACV, ICIP, and CVPR. He is the Program Chair of WACV 2018. He is an Associate Editor for the *Neurocomputing* journal. He has authored more than 100 scientific publications, and has filed 22 U.S. patents.

Yi Yang received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2010. He is currently an Associate Professor with the University of Technology Sydney, Ultimo, NSW, Australia. He was previously a Postdoctoral Researcher with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. His current research interest includes machine learning and its applications to multimedia content analysis and computer vision, such as multimedia indexing and retrieval, surveillance video analysis, and video content understanding.



Di Xie received the B.S. degree in software engineering and the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2007 and 2012, respectively. He is currently a principle Research Manager with Hikvision Research Institute, Hangzhou, China. His research interests include computer vision, video understanding, and deep neural network optimization.

Yueting Zhuang received the B.Sc., M.Sc., and Ph.D. degrees in computer science from Zhejiang University, Hangzhou, China, in 1986, 1989, and 1998, respectively. From February 1997 to August 1998, he was a visiting scholar at Prof. Thomas Huangs group, University of Illinois at Urbana-Champaign. He is currently a Full Professor with the College of Computer Science, Zhejiang University. His research interests mainly include artificial intelligence, multimedia retrieval, computer animation, and digital library.