One-Shot SADI-EPE: A Visual Framework of Event Progress Estimation

Jianqin Yin, Xiaoli Liu, Fuchun Sun, Huaping Liu, Zhiqiang Liu, Bin Wang, Jun Liu, and Yilong Yin

***Abstract*— In many practical engineering applications, the number of actions that have been finished should be known, particularly for an untrimmed video sequence that includes an event with a series of actions, it is important to know the number of actions that have been finished. In this paper, we termed this process as visual event progress estimation (EPE). However, the research related to this problem is few in the research community. To solve this problem, a visual human action analysis-based framework, namely one-shot simultaneously action detection and identification (SADI)-EPE, is presented in this paper. The visual EPE is modeled as an online one-shot learning-based problem; sliding window and attention-based bag of key poses formulate our framework. Unlike most of the action analysis methods relying on a number of training data of some predefined classes, our method can realize SADI for any event if one sample of the event is given, which makes it feasible for practical applications. At the same time, not only SADI but also the progress estimation of the event can be realized by our algorithm. In terms of methodology, the key pose is defined by an invariant pose descriptor from skeletal data and silhouette data. Moreover, in order to extract representative and discriminative poses from one training sample, we present a new bidirectional *k*NN-based attention weighted key pose selection method, which can filter the unrelated actions and model different importance of various key poses. In addition, an attention-based multi-modal fusion scheme, which addresses the difficulty of high-dimensional features and few training samples, is proposed to augment the performance of our algorithm. Finally, we propose an evaluation criterion for the estimation problem. Extensive results demonstrated the efficacy of our proposed framework.**

***Index Terms*— Action detection and identification, one-shot, *k*NN.**

Manuscript received January 18, 2018; revised May 13, 2018; accepted June 2, 2018. Date of publication June 14, 2018; date of current version June 4, 2019. This work was supported in part by the National Natural Science Foundation of China under Grants 61673192, U1613212, and 61573219, in part by the Fund for Outstanding Youth of Shandong Provincial High School under Grant ZR2016JL023, in part by the Shandong Provincial Key Research and Development Plan under Grant 2017CXGC1504, in part by the Dominant Discipline and Talent Team of Shandong Province Higher Education Institutions, and in part by the Basic Scientific Research Project of Beijing University of Posts and Telecommunications under Grant 2018RC31. This paper was recommended by Associate Editor W. Lin. *(Corresponding author: Yilong Yin.)*

J. Yin and X. Liu are with the Automation School, Beijing University of Posts and Telecommunications, Beijing 100876, China.

F. Sun, H. Liu, Z. Liu, and B. Wang are with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China.

J. Liu is with Weill Cornell Medical College, Cornell University, New York, NY 10021 USA.

Y. Yin is with the School of Software Engineering, Shandong University, Jinan 250101, China (e-mail: ylyin@sdu.edu.cn).

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/TCSVT.2018.2847305

I. Introduction

ACTION analysis plays an important role in many applications [1]–[4] and therefore has attracted extensive research focus in recent years. Previous research has made great contributions in traditional action recognition [1], [5], [6] and action detection [7], [8]. However, event progress estimation (EPE) is also critical in action analysis to improve the quality, effectiveness and synchronization of related actions in real-world applications. For example, in a factory production line, the operator is required to complete a certain number of actions for assembly of several parts. The event progress of a worker analyzed can improve the efficiency and facilitate the control of the job quality by discovering the problems where sub-jobs are neglected or delayed. Moreover, it can also synchronize the work flows where multiple workers are involved in large projects. Similar circumstances also exist in other applications such as nursing care and so on.

At present, the PDA or speech recognition schemes are used to report event progress [9], [10]. However, these methods are passive and thus requires the cooperation from performers. As a result, these methods cannot handle cases if the performers are unable or forget to report the event progress. With the development of motion recording sensors (e.g., video camera or Kinect), human actions can be recorded for online or offline analysis to obtain EPE information. An automated process estimation scheme can be possibly developed by analyzing visual data.

A. Problem Description

In order to build automated schemes for progress estimation with visual data, we start with some concepts related to EPE, events, activities and actions. There are various definitions of events, activities and actions [1]–[4], [11]. In these works, action, activity and event are commonly used to refer to the human movements with no obvious differences. In this work, we follow the similar definitions in [11] and provide related notations to discuss the problem clearly.

Term 1 (Action): Action is a low-level semantic concept in a relatively short time defined for one specified task. In the examples of medical and physical exercises, specified tasks include arm stretching and chest exercises.

Term 2 (Event): Event is a high-level semantic concept in a relatively long time defined for one specified task. It can be defined as a set of actions that an actor needs to perform for completion of one specified task or job.

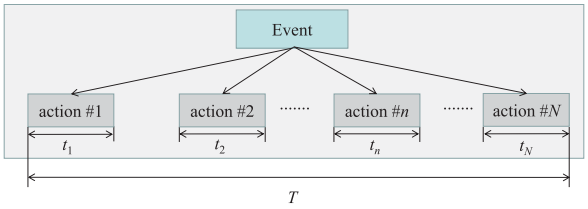


Fig. 1. Event consisting of N actions.

To estimate the event progress, we regard the entire event as a series of actions [see Fig. 1]. Suppose the event consists of N actions. The duration of the action and event is denoted by t and T, respectively; and the duration of the nth action is represented by t_n. If n_a actions have been finished, the progress of the event can be estimated by the following equation:

$$r = \frac{\sum_{i=1}^{n_a} t_{a(i)}}{T} \tag{1}$$

where, $T = \sum_{i=1}^{N} t_i$, r is the event progress rate, and n_a is the number of the finished actions, and $a(i)$ is the index of the ith finished action.

With equation (1), we can estimate the event progress based on the action identification results. However, the visual data that we obtained in practical applications is continuous or untrimmed. Accordingly, we must identify the temporal location and the class label of the action to estimate the event progress. Therefore, the EPE is a simultaneous action detection and identification (SADI) problem.

B. Problem Analysis

As discussed above, the EPE is an SADI problem. Compared to the classical SADI problem, the EPE problem has four additional special characteristics.

First, the categories of actions involved in SADI-EPE problem are unknown and dependent on specific tasks. Therefore, different from traditional SADI problem defined in which the actions are specified and known, the detection and recognition for the EPE is defined on an open set. Accordingly, the algorithm must be easily adaptable to new action classes.

Second, recording training data for multiple times is often very difficult or unacceptable for the workers in many applications. Thus, the solutions for EPE should work with limited training samples. In this study, we developed a one-shot learning solution by using only one training sample.

Third, the EPE must be conducted online. Therefore, compared to the methods that only work on segmented or trimmed video sequences, the new solutions should be able to obtain the estimation results using continuous untrimmed sequences.

Finally, the EPE requires to complete the action detection and recognition at high frequency in order to achieve a high efficiency for the practical applications.

Besides the aforementioned special challenges, the EPE also shares the common challenges in the traditional SADI problem including obstruction, large intra-variance, and interruption or action disorders. When the actor is performing actions, obstruction may occur when other passengers, self-occlusion appear in the captured video. In the SADI-EPE problem, the large intra-variance of the action is another challenge, because different performers do the same action according to their own style and the actor may also be located at different locations in relative to the sensor, resulting in different action appearances. When performing a series of actions, the actor may use different orders and the actions may be interrupted by other events, making SADI problem difficult.

In the past several years, vision-based action analysis has been a highly active area of research. However, previous efforts have mainly focused on the analysis of short video clips with a large number of training samples [12]. Action analysis from untrimmed videos with limited samples, which is a problem pertinent to real-world applications and close to human perception, is drawing increasing attention and is suitable to our application. The challenges listed above have motivated us to seek robust action representations and efficient detection algorithms.

In recent years, deep learning-based frameworks for action analysis, such as 3D convolutional networks (C3D) [13], two-stream convolutional networks [14], [15], temporal segment networks (TSN) [16], deep features [17], [18], recurrent neural networks [19] have pushed forward the state of the art. However, the gains over bag-of-words approaches have not been impressive [20], [21]. Moreover, the performance of the deep learning-based method heavily relies on a large number of training samples [16]. When not enough training samples are available, extra training data are needed to solve action recognition. At the same time, the complexity of a highly deep structure makes the training of the framework time-consuming. To solve these limitations, we use handcrafted features to form bag-of-words framework to solve one-shot EPE in this study.

EPE concerns online and real-time event progress; hence, sliding window is used in segmenting an untrimmed video to guarantee the report of the result according to regular frequency. To handle the intra-variance of the actions, we use Kinect to capture motion data for its robustness to illumination and projection loss. The invariant pose descriptor, which is robust to different persons and environmental changes, should be used based on the motion data. To select discriminative information for one-shot SADI, the representative and discriminative poses should be selected to represent the action. Finally, different module information should be used to augment the performance of the algorithm.

Compared to previous research, this work presents a new solution to the newly defined SADI-EPE problem. For the first time, this paper analyzes the characteristics of EPE, proposes a new visual framework and evaluates its performance with corresponding criteria. Additionally, the visual framework is built on an SADI scheme by defining the visual EPE as a one-shot learning based online detection and identification problem. Under this new visual frame, we use sliding window and bag-of-key-poses to determine the event process. A bidirectional kNN-based attention (BikNN) weighting

scheme is used to mine discriminative key poses. With the new BikNN scheme, different attention weights for different poses can be obtained. Attention-based fusion is also used to augment the final estimation performance.

The remainder of this paper is organized as follows. The next section investigates the related works. Section III proposes our estimation framework. Section IV presents and discusses the experimental results. Finally, Section V concludes our paper.

II. Related Work

Although human action detection and recognition have been extensively researched, EPE has not been well studied. We model EPE as a special SADI problem based on RGB-D data in one-shot setting. Accordingly, we first review the related works of EPE and the related works of human action recognition using RGB-D data. After that, we summarize the works related to one-shot action recognition and detection.

A. EPE

EPE is an important problem for improving the quality and the effectiveness of related operations. Previous solutions to this problem are available in [9], [22], and [23]. These studies focused on the estimation in service-related applications and used speech technique as the input for determination of the event progress. However, in many other applications (e.g., in shop floors), the most discriminative information is the visual data. In contrast to the previous studies, for the first time, we use a visual action analysis scheme to estimate the progress of the events.

B. RGB-D Data-Based Human Action Recognition

RGB-D data-based human action recognition has recently attracted extensive research interests [3], [24]–[27]. RGB-D data carry substantial information for modeling actions, such as the depth map, skeletal and RGB data. RGB data-based action recognition has been extensively researched [1], [15], [16], [28]–[31]. However, given the aforementioned difficulties and the convenience of visual capturing devices, we focus on the methods using depth map data and skeletal data.

Xia *et al.* [32] extracted the key skeletal data using the position of 3D skeletal points. Then, the histogram of the key points was used to model the action. Li *et al.* [33] modeled the dynamics of the action as an action graph, whose nodes are the salient postures modeled by a bag of 3D point method. They showed the promising performance of their method; however, they also indicated that their method is view dependent. Vieira *et al.* [34] described an action by space-time occupancy patterns. They segmented depth maps into different 4D grids according to the space and time axes; then the number of the points in different 4D grids was used to model the action. Wang *et al.* [35] also used space-time occupancy pattern to describe an action and used sparse coding to model the action. Yang *et al.* [36] combined motion energy images and histogram of oriented gradients to describe an action. Oreifej and Liu [4] used the histogram of oriented 4D norms to model an action. Cheng *et al.* [37] proposed the comparative coding descriptor to represent an action, and fused color information and depth information. Wang *et al.* [38] proposed weighted hierarchical depth motion maps and three-channel deep convolutional neural networks for human action recognition from depth maps. Chen *et al.* [39] and Liu *et al.* [40] proposed multi-temporal depth motion maps and Fisher vector for modeling shape discrimination and speed variations. Ding *et al.* [41] proposed to encode 5 spatial skeleton features into images with different coding methods, and CNNs were used to realize action recognition. Liu *et al.* [27] introduced a new large scale benchmark(PKUMMD) for continuous multi-modality 3D human action understanding.

In general, there are two popular pipelines for solving action detection and recognition in the research community. One is based on the bag-of-words framework, while the other is based on deep learning. In recent years, the performance of the pipeline based on deep learning was considered to outperform that of the bag-of-words. However, a large amount of training data is needed for the deep learning pipeline in order to achieve high performance. In practical applications when substantial training data is not available, one-shot learning approach is necessary.

C. One-Shot Action Detection and Identification

One-shot learning approach is advantageous because it requires minimal amount of data [42]. Fanello *et al.* [43] used 3D histograms of scene flow (HOF) and global histograms of oriented gradient (HOG) to capture low-level features, adopted sparse coding to capture high-level patterns, and used a sliding window and linear support vector machines to segment and recognize gestures simultaneously. Malgireddy *et al.* [44] proposed a hierarchical Bayesian model for one-shot gesture detection and identification. Each frame was represented by the visual words in it, and the visual words were formulated by the HOG and HOF over the entire space of gestures. In addition, a multiple channel HMM was proposed to locate and identify the gesture. Unlike the classic HMM, this model had multiple channels, where each channel was represented as a distribution over the visual words corresponding to that channel. Wan *et al.* [42] proposed some mixed features around sparse key points. To address the insufficiency of one-shot training samples, they augmented the training samples by artificially synthesizing versions of various temporal scales, which was beneficial for coping with gestures performed at varying speed. However, the synthesis process makes it time-consuming for training.

Although Fanello *et al.* [43] focused on the closest problem that we refer to, some differences still exist. They focused on SADI of the gestures; however, our scheme aims to determine high-level information and the progress of the event. Moreover, their work aimed to recognize some predefined gestures; meanwhile, our work aims to recognize uncontrolled actions.

III. Visual EPE Framework

A. Data Preparation

Obstruction is an important problem in the event progress. To make our algorithm robust to obstruction, we use multiple Kinects to capture data. Theoretically, more Kinects are able

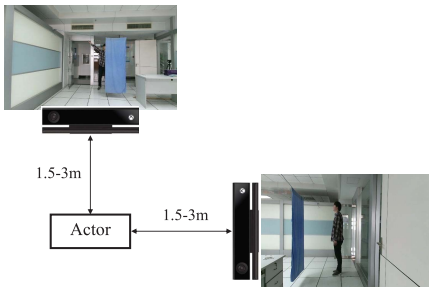


Fig. 2. Kinect distribution.

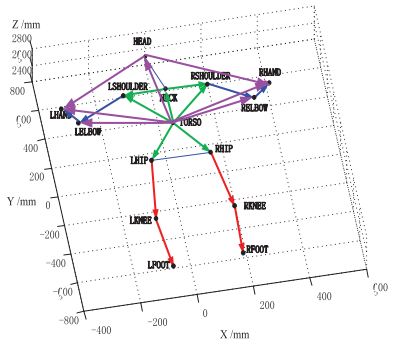


Fig. 3. Interested body structure vectors.

to help improve the system performance. However, adding more video sensors will definitely increase the computational complexity and further add the hardware cost. To simplify the system structure, we use only two Kinects positioned at orthogonal angles in our experiments to evaluate the effectiveness of the proposed methods. The setting of the two cameras is shown in Fig .2. By using this setting, obstruction can be avoided to some extent. For example, if the actor is obstructed in the frontal Kinect, then the information from the side Kinect can be used to recognize actions.

B. Action Feature Representation

The skeletal data comprise a large amount of information about actions [45]. However, the skeletal data are often corrupted or noisy because of occlusion or other problems [46]. Silhouette or binary data can also be used to represent the poses [25]. With the supplementary of binary data, the performance is better than the performance when only the skeletal data used. Thus, we use skeletal and silhouette data to describe actions. To handle the variances caused by different locations of the Kinect and different persons, we use relative angle-based pose descriptor from skeletal data. Moreover, we present vertical features to describe the pose from binary data.

1) Feature Representation of Skeletal Data: We use the features used in [47] to represent the skeletal pose. To illustrate the skeletal features, we have introduced the concept of the body structure vectors. And based on this body structure vector, we show the feature vectors we constructed. We named the vector from one joint to the other joint as the body structure vector. We select some body structure vectors to reflect the human actions, in which different parts of the human body in action are represented by different colors, as shown in Fig. 3. The blue lines represent the actions of the upper limb, the red lines represent the actions related to the lower limb, the green lines represent the torso actions and the purple lines represent the actions of the upper limb and the torso. We call these vectors as interested body structure vectors. We can define the relative angles between the pair-wise interested body structure vectors conveniently based on this concept. Unlike the method in [48], which only used the angles between the connected limbs, the virtual angles between the interested structure vectors are also used in this study. We denote these angles as S_s.

Aside from the static pairwise relative angles constructed by the interested body structure vectors, we present a new

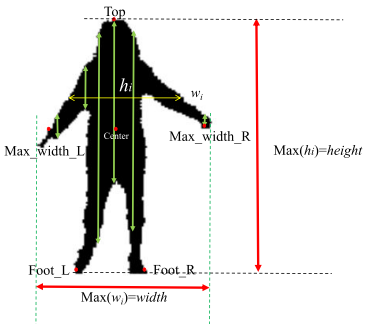


Fig. 4. Silhouette image and corresponding features.

dynamic pose descriptor. The dynamic descriptor encodes the dynamic information of the action by velocity angles. The velocity angles are constructed by the interested body structure vector and the velocity vector, which are formed by two points (One point is the joint at the previous frame, and the other point is the same joint at the current frame).

We use five joints to construct the velocity vector considering the particularity of the action. The five joints are LELBOW, RELBOW, LHAND, RHAND, TORSO separately. We use S_V to represent the velocity angles.

The combination of the static and dynamic information of the action shows that the skeletal pose can be represented by the concatenated vector (S_S, S_V). That is, the final skeletal pose can be represented by the following formula.

$$S_f = [S_S, S_V] \tag{2}$$

2) Feature Representation of Binary Data: To get the shape features of the binary image, we compute the features of the binary or silhouette data according to Fig. 4. For simplicity, we refer to these features as binary features. We first compute the center point of the silhouette of the human body. Then, we compute the maximum height and maximum width of the silhouette. Subsequently, the binary pose is obtained as follows.

Step 1: Locate six key points (i.e. Top, Center, Max_width_L, Max_width_R, Foot_L and Foot_R), as shown in Fig. 4. Compute the width and the height of the silhouette, and denote them as $width$ and $height$ respectively.

Step 2: Compute the height of the silhouette line by line, and obtain the vertical features as follows:

$$h = [h_1, h_2, \cdots, h_{width}] \tag{3}$$

To provide a uniform representation for varying image sizes and shapes, set a constant L (L is an empirical value, and here we set $L = 120$). Sample the vertical feature to the same length of L using Formula (4).

$$D[i] = h[\left\lceil \frac{i * width}{L} \right\rceil], \quad \forall i \in [1, 2, \cdots, L] \tag{4}$$

where $\lceil \bullet \rceil$ is the ceiling function. Finally, normalize the lateral feature D to obtain a unit sum using Formula (5).

$$\overline{D[i]} = \frac{D[i]}{\sum_{i=1}^{L} D[i]} \tag{5}$$

Step 3: According to the key points (i.e. Top, Center, Max_width_L, Max_width_R, Foot_L and Foot_R, marked as p_1, p_2, p_3, p_4, p_5, p_6, respectively), obtain the following structure vector related to each point.

$$v = p_i - p_j, \quad \forall i, j \in [1, 2, \cdots, 6],\ i < j, \tag{6}$$

Then, compute the angles between different structure vectors v and denote them as g.

Step 4: Obtain the height and width of the silhouette of the human body and compute the feature as follows.

$$\frac{width}{height} \tag{7}$$

Therefore, the final representation of the binary pose can be obtained as

$$b_f = \left[\overline{D}, g, \frac{width}{height}\right] \tag{8}$$

C. BikNN Based Attention Weighting Key Poses

As previously discussed, EPE can be modeled as a one-shot learning-based SADI problem. When we extract key poses for actions, we should use an algorithm that is adaptive to small-size problems. As discussed in [26], kNN is suitable to the one-shot learning case. Zanfir *et al.* [26] considered that the pose most of whose kNNs belong to one class as the key pose. Its justification lies in the fact that the pose whose kNNs belong to the same class is the pose that may be located in the center or near the center of the data set of the same class. This finding seems practical for ideal data that the inter-variances between different classes are large. However, different actions may often share some similar poses. In addition, their method did not consider unrelated actions. However, unrelated actions often happen in real-world applications. According to the attention scheme of the human being, the unique key pose of one action should be given more attention, the similar key poses between different actions should be given less attention, and the unrelated key poses should be ignored. Thus, we can evaluate the key poses according to the attention weighting scheme. Another important problem is that Zanfir et al.'s method will produce numerous redundant key poses that will influence the classification result. These redundant key poses should be filtered. To address these problems, we present a new method based on Zanfir et al.'s method.

A two-step method is presented to solve the above problem. In the first step, we learn the candidate discriminative key poses by forward kNN. In the second step, we filter the candidate key poses and evaluate the final key poses by the backward kNN.

We use $P = p_1, p_2, ..., p_n$ (n is the number of poses of all actions) to denote all the poses of all the actions, where p_i is a pose (skeletal pose or binary pose), which can be computed according to the method described in Section 3.2.

1) Learning Candidate Discriminative Key Poses by Forward kNN: In this section, we learn the candidate discriminative key poses by forward kNN.

We use kNN to group poses by computing the Euclidean distance between two poses. This process is similar to Zanfir *et al.*'s method [26], and it is presented as follows.

Input: An event sequence that contains all poses $P = p_1, p_2, ..., p_n$ of all actions and all labels $L = l_1, l_2, ..., l_n$ of all poses, which are determined by the corresponding action label of the poses; the filter ratio parameter θ; and the parameter k of kNN.

Output: Candidate key poses.

Forward Algorithm:

Step 1: Compute the Euclidean distances between the current pose p_i and the other poses p_j, $j = 1, 2, \cdots n$, $j \neq i$.

Step 2: Sort the distances. Select k poses whose distances are the first k minimum as the kNN poses of pose p_i.

Step 3: Compute the number of the kNN poses of pose p_i whose class label is the same as the label of pose p_i, and denote it by m.

Step 4: If the ratio of $\frac{m}{k} > \theta$, then pose p_i is the candidate discriminative key pose.

2) Attention Weighting and Filtering Scheme for Key Poses: By using the forward algorithm, we can obtain a number of candidate key poses. The analysis reveals that some redundant key poses exist among the candidate key poses because when we extract key poses, some different key poses are the nearest neighbors to one other. Meanwhile, different key poses may share the same neighbors but with different distances. That is, different key poses have different discriminative powers. Another important problem is that unrelated actions or interruption may occur in recognizing the actions in real-world applications. Therefore, a scheme should be provided to evaluate the discriminative power of different key poses and filter the unrelated actions. Different attention weights should be assigned to different key poses according to different discriminative powers.

We use the same training dataset to evaluate the key poses using an inverse process of the above learning candidate key poses process. We assign the poses of the training set to the candidate key poses with the minimum distance with the current pose. Thus, we can model the competing mechanism between different candidate key poses. The result of the assignment is presented in the following three cases.

The first case is that the assignment poses of the key pose comes from the same class with the key pose. In this

case, the key pose has acceptable discriminative power for its corresponding class.

The second case is that the assignment results can be obtained not only from the same class but also from the different classes with the key poses. In this case, the key pose not only has discriminative power for its corresponding class but also for other classes. We should evaluate the different discriminative powers for different classes.

The final case is that all the assignment results are obtained from different classes or that nothing is assigned. In this case, the key pose has no discriminative power and should be deleted.

We can learn the weight of different key poses and a threshold filtering the unrelated actions based on the assignment results. We call this process as backward process and present the process as follows.

Input: All poses $P = p_1, p_2, \cdots, p_n$ of all actions and the candidate key poses $KP = kp_1, kp_2, \cdots, kp_M$; the number of actions;

Output: Key poses $KP = kp_1, kp_2, \cdots, kp_L$; weights of different key poses to different classes $W = {w_1}^1, {w_1}^2, \cdots {w_i}^j, \cdots {w_L}^C$; effective radius of the key poses $R = r_1, r_2, \cdots, r_L$.

Backward Algorithm:

Step 1: For any pose $p_i, i = 1, 2, \cdots, n$, compute its distance with the candidate key poses $KP = kp_1, kp_2, \cdots, kp_M$.

Step 2: For every pose, assign it to the candidate key pose with the minimum distance. Thus, any pose can be assigned to one candidate key pose. At the same time, if the assigned pose has the same label with that of the candidate key pose, then record the distance between them as an attribute of the candidate key pose and denote the distance as candidate radius of the candidate key pose.

Step 3: For the candidate key pose $KP = kp_1, kp_2, \cdots, kp_M$, compute the number of the labels of the poses assigned to it. Denote the number as ${n_i}^j$, where i is the index of the candidate key pose, and j is the number of poses belonging to action j.

Step 4: Compute the effective radius for the candidate key poses. For the candidate key pose i, if no candidate radius exists, then delete it because this candidate key pose has no discriminative power or has extremely limited power. If candidate radii exist, then record the maximum distance as its effective radius r_i.

Step 5: Compute the weight using the following formula:

$$ {w_i}^j = \frac{{n_i}^j}{\sum_{k=1}^{L} n_i^k} \tag{9} $$

The poses belonging to the unrelated actions can be filtered using the effective radius. The discriminative power of different key poses can be evaluated using the weight. Moreover, we can pay different attention to different key poses, that is, the weights are attention-based weights.

3) Analysis of BikNN Algorithm: We illustrate the process of BikNN algorithm in Fig. 5 by using an event consisting of eight actions as an example. In Fig. 5(a), the horizontal axis represents the frames of the event. If the frame is selected as the key pose, then we draw a red vertical line. Therefore, the number of the red vertical lines is the number of key poses, as marked on the top of the figure. From Fig. 5(a), 1841 key poses are generated using forward kNN. However, the key poses are too dense; therefore, they are competing to each other. Competing results are shown in Fig. 5(b), which also presents the number of poses that match with the

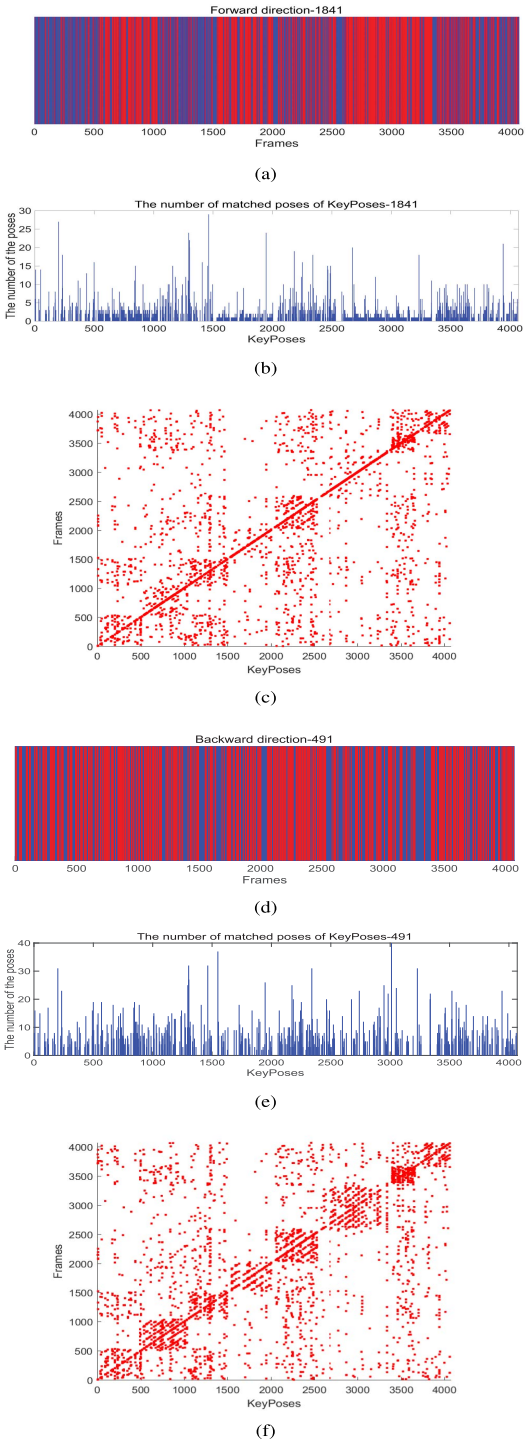


Fig. 5. The illustration of BikNN. (a) Key poses generated by the forward process of BikNN. (b) Number of poses matched to key poses of forward kNN. (c) Key poses generated by the backward process of BikNN. (d) Number of poses matched to key poses of BikNN. (e) Match relationship between poses and key poses of BikNN.

corresponding key poses. To show the relationship between the key poses and the original frames, the horizontal axis uses the number of the frames of the entire event. These dense key poses may lead two problems. First, excessive key poses increase the computation complexity. Moreover, some key poses possess less discriminative power or ambiguity. The ambiguity of key poses is shown in Fig. 5(c). More ambiguous key poses indicate that more poses will be scattered far away from diagonal blocks. By contrast, more discriminative power key poses imply denser diagonal blocks. However, as shown in Fig. 5(c), for actions 4 and 6, the distribution of the poses is relatively sparse near the diagonal elements. Some key poses with less discriminative power will affect the recognition results. Thus, the backward process is used to mine the most discriminative key poses.

The related results in Figs. 5(d), 5(e) and 5(f) show the utility of the backward kNN. From Fig. 5(d), only 491 key poses from the 1841 key poses are left and the other key poses are filtered by the backward direction kNN. For every key pose, the number of the matched poses is shown in Fig. 5(e). The match relationship of poses and key poses is shown in Fig. 5(f). The comparison between Figs. 5(c) and 5(f) shows that the resulted key poses are more discriminative in Fig. 5(f). Denser blocks are distributed around the diagonal axis.

We also use a toy example to show this process in Fig. 6. This toy example uses the first 650 frames of the video shown in Fig .5. In Fig. 6(a), in the first row, the figure illustrates the key poses of the segment using forward kNN. In order to explain our algorithm better. We take the frames circled by the green rectangles as the example, the second row shows the detailed frame index of the key poses of the video, and the bottom row shows the key poses extracted by the forward kNN. The numbers below the images are the frame indexes of the video. From Fig. 6(a), we can see that from the 199th frame to 231th frame, 12 key poses can be extracted by the forward kNN. The extracted key poses are illustrated in the bottom of Fig. 6(a). For the key poses of 200th frame and 201th frame, they are very similar. Using BikNN, the extracted key poses are illustrated in Fig. 6(b). Only 7 key poses are remained. From Fig. 6(b), we can see the final 7 key poses are more representative than the 12 key poses from Fig. 6(a).

One key pose can relate to more than one actions, as shown in Fig. 5(f). Thus, to model this problem, we compute the weights for the key poses. The contribution of some key poses to different actions is shown in Fig. 7. Each key pose contributes to different belief degrees for different actions. The attention mechanism can be explained using this figure. For example, key pose 1 generates different belief degrees to different actions. The weights are used to model this belief degree. Moreover, from the viewpoint of the pose, different weights are connected to the importance of different poses. When the key pose is more important, it is provided with more attention resulting in larger weight. Thus, it can be used to model our attention mechanism.

From the above discussion, the proposed BikNN algorithm is based on kNN, the time complexity of which is $O(n)$, where n is the number of the training samples. Because kd-tree is used to optimize our algorithm, and the time complexity of

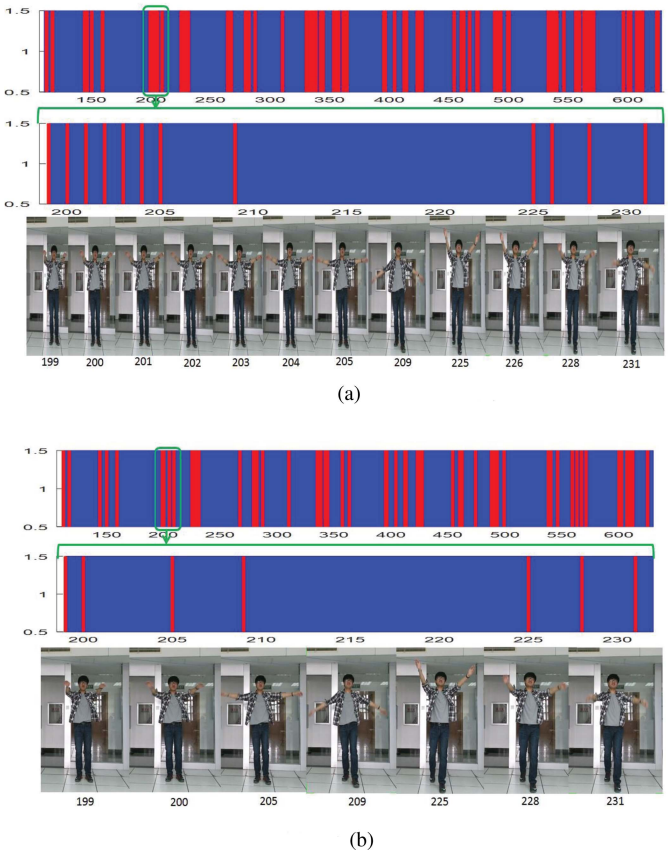


Fig. 6. The toy example of BikNN. (a) Extracted key poses using forward kNN. (b) Extracted key poses using BikNN.

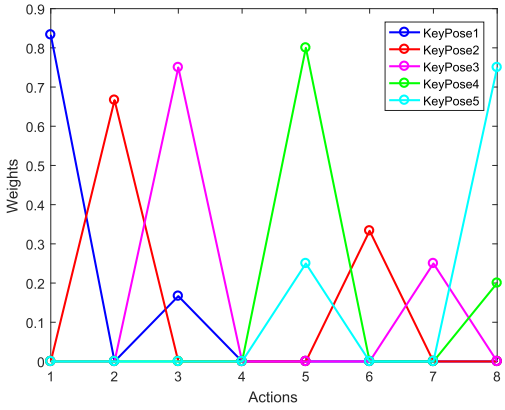


Fig. 7. Relationship between key poses and actions.

kd-tree is $O(log(n))$. Therefore, the time complexity of BikNN can be regarded as $O(log(n))$.

D. Attention-Based Fusion

We use the data from two Kinects for recognizing actions to decrease the influence of occlusion. Meanwhile, we use the skeletal and silhouette data to augment the robustness of the action recognition algorithm. Feature based fusion often concatenates different features without considering the difference of the features. The high-dimensionality of the features causes difficulty in obtaining acceptable results using small-sized training samples. Classifier based fusion often requires

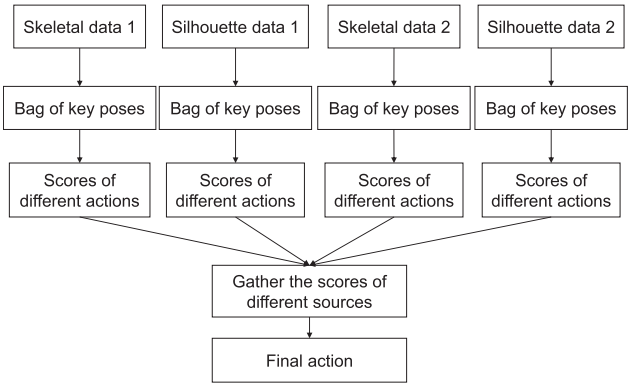


Fig. 8. Fusion framework.

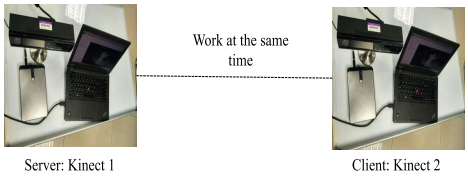


Fig. 9. Data extraction module.

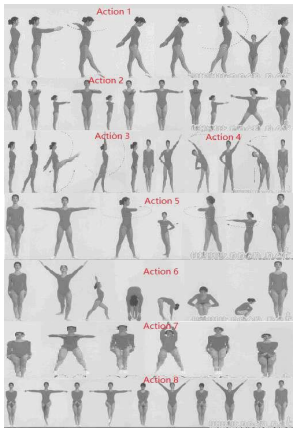


Fig. 10. Gymnastics event.

TABLE I
DESCRIPTION OF OUR DATASET

Data	Speed	Occlusion	Disorder
Data 1	Normal	No	No
Data 2	Normal	Yes	Yes
Data 3	Normal	No	Yes
Data 4	Normal	Yes	No
Data 5	Slow	No	No
Data 6	Slow	Yes	No
Data 7	Slow	Yes	Yes
Data 8	Slow	No	Yes

many training samples. Given that our problem is a one-shot learning problem, we present a fusion method that fuses the information on the level between the feature and classifier level. The attention weights from Section III-C.2 are used for weighing key poses to obtain the score of every action. Actions are classified as the action with the highest score. The illustration is presented in Fig. 8.

The fusion algorithm mainly contains four steps.

Step 1: Construct the feature vectors for the skeletal and silhouette data according to the discussion in Section III-B.

Step 2: For every pose of the sequence, compute its matched key pose for different data using the result in Section III-C.2.

Step 3: Compute the score for different actions using different data sources by bidirectional kNN using the result in Section III-C.2.

Step 4: Compute the final score for different actions.

Step 5: Recognize the sequence as the action with the highest score.

In the above presented fusion algorithm, attention weights are used; thus, the fusion method can pay different attention to different poses from different data sources.

IV. EXPERIMENTAL RESULTS

We extensively experimented on the proposed idea using our dataset and OAD dataset [49].

A. Experimental Setting and Dataset

We use two Kinects to capture the data. We develop our data extraction program to extract data from two different viewpoints, thereby obtaining data from two Kinects simultaneously. The illustration is presented in Fig. 9, and the working setup is shown in Fig. 2. One of the computers is the server computer and the other is the client one. We use the transmission control protocol to allow the two Kinects to communicate with each other. The two computers connected to the two Kinects should be in the same LAN. The data from these two Kinects can be captured simultaneously by the data extraction module. Moreover, the module can support the virtual Kinect, that is, the .XEF file. This setting can augment the flexibility of our method. We construct our dataset based on this module.

Our dataset has three actors. Every actor performs the specified event illustrated in Fig .10 eight times. The figure shows the eight actions in our interested event. For the three actors, only one data, which includes the normal data, is available. The other data of the actor may contain the data of occlusion, disorder and different speeds. The detailed information of the data is presented in Table. I. Every actor performs the event using his/her own style and speed eight times according to Table I. Therefore, the frames of every video are different, 48 videos (24 frontal videos and 24 side videos) are captured, and at least 8 actions are captured in every video because unrelated actions may occur. In order to show how the person is occluded, we also give some example images in Fig. 11.

The other dataset we used is OAD data [49]. Only one viewpoint data can be obtained, and there is no time stamp for the video. In OAD, there are 10 actions and 59 videos.

B. Evaluation Criterion

We define our criteria to evaluate our algorithm in two cases. The first case is that the training and test data are performed

Fig. 11. The occluded examples.

by the same person. The second case is that the training and test data are performed by different persons. Given an event including n actions labeled as $1, 2, \cdots, n$, we use two criteria, namely, classification criterion, cA , and regression criterion, rA , to evaluate our algorithm.

The definition of the classification criterion is as follows:

$$cA = 1 - \frac{\sum_{m=1}^{M} (\delta(R(m), C(m)))}{M} \tag{10}$$

where, $R(m)$ and $C(m)$ are the actual and estimated action labels, respectively; and M is the number of time intervals. This criterion is essentially the criterion of action recognition. Thus, we use this criterion to evaluate the performance of our one-shot SADI algorithm.

Then, we define the regression criterion as follows:

$$rA = 1 - \frac{\sum_{m=1}^{M} |RS(m) - CS(m)|}{M} \tag{11}$$

where, $RS(m)$ and $CS(m)$ are the number of the actual actions completed and the number of the estimated completed actions, respectively. The other parameters are the same as those in the classification criterion. Evidently, $RS(m)$ and $CS(m)$ can be computed by summarizing $R(m)$ and $C(m)$ respectively.

From the above equation, the regression criterion is closely connected to the classification criterion. Thus, when we evaluate our algorithm, we mainly focus on the results of the classification criterion. For the regression criterion, we only provide some simple evaluation results.

C. Some Examples of Experimental Results

The following results are provided by the following experimental setting: Only one training sample performed by one person can be obtained. Sliding window is used to segment the continuous videos and the event can be segmented into a series of segments. Then the action classification is used on every segment to detect the action. There are 30 frames in each sliding window and the decision is given in each sliding widow. One action instance may often cover multiple time intervals. For this case, the recognition results can be merged. Also, multiple actions may be completed within one time-interval. In our proposed method, data from different views are aligned according to the time stamp. When aligned, there are 15 frames in 1 second. 30 frames are used to make decision in each time interval. That is to say, 2 seconds are used in each time interval. In this short time interval, 1 action is considered. If multiple action instances occur in one time-interval,

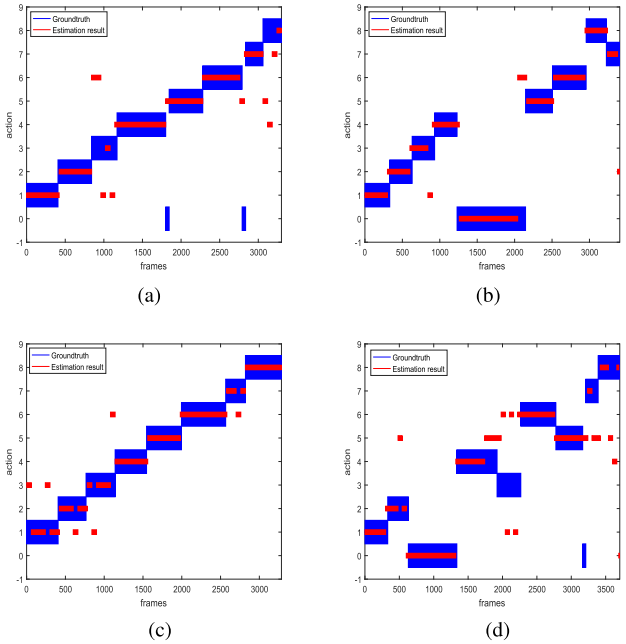


Fig. 12. Some classification results. (a) Result A. (b) Result B. (c) Result C. (d) Result D.

according to our recognition scheme, the decision will be the action with the highest score. The results in Fig. 12 are based on the classification criterion. In this figure, the ground truth and estimation result are represented by the blue rectangle red line, respectively.

In Fig. 12(a), although interruption exists in the event, our algorithm can distinguish different actions effectively. This result can be attributed to the filtering scheme of the key pose of our algorithm. Most recognition errors occur at the start point or end point of different actions because of the incomplete action information in the corresponding sliding windows. Aside from this error, some errors occur because of the confusion between the Actions 1 and 3. Several shared poses exist in Actions 1 and 3, as shown in Fig. 10, there are lots of shared poses. When we make a decision, the sliding window may cover the shared poses between Actions 1 and 3. Thus, they cannot be easily distinguished when only the incomplete sequences are obtained. In addition, Actions 1 and 3 are extremely similar to discriminate. Moreover, Fig. 12(b) shows the disorder in the action. The figure shows that the main error occurs because of the confusion between Actions 1 and 3. Similar to Fig. 12(a), this finding is caused by the use of the sliding window scheme to obtain online decision. At the same time, the test sample is extremely slow. The sliding window can only cover some shared poses of Actions 1 and 3. Some other results are shown in Figs. 12(c) and 12(d).

D. Evaluation of Our Algorithm on Our Dataset

To test our algorithm. We compare it with state-of-the-art methods, including C3D [13] and TSN [16]. Considering our problem, there is only one video can be used as training data, therefore we use the C3D network and TSN network trained by other datasets. For C3D network, it is trained on

TABLE II
PERFORMANCE COMPARISONS FOR SAME-SUBJECT

TD	Method	N	OD	D	O	S	SO	SOD	SD
P1	C3D+SVM	92.4	13.4	12.4	15.8	16.6	13.4	12.2	16.4
P1	C3D+BikNN	93.4	24.1	30.3	12.8	1.1	22.3	24.5	0.4
P1	TSN+SVM	94.2	50.0	53.9	57.1	65.5	44.3	48.6	66.1
P1	TSN+BikNN	95.4	68.0	56.8	62.8	76.9	58.1	53.8	81.9
P1	OurF+SVM	**98.8**	55.3	57.2	63.7	81.1	53.1	54.2	77.2
P1	Our method	95.4	**80.6**	**90.3**	**73.2**	**94.5**	**83.2**	**84.5**	**92.7**
P2	C3D+SVM	88.0	13.0	10.4	11.4	12.2	9.8	11.7	13.4
P2	C3D+BikNN	92.3	32.4	22.6	22.3	2.7	24.2	24.0	2.4
P2	TSN+SVM	88.3	52.7	59.0	49.6	70.9	42.0	38.9	69.1
P2	TSN+BikNN	**95.8**	65.1	71.1	**62.4**	78.1	69.1	63.8	80.3
P2	OurF+SVM	95.3	50.1	62.5	54.7	77.7	55.8	53.1	79.8
P2	Our method	**95.8**	**74.8**	**92.2**	60.9	**82.2**	**71.4**	**68.4**	**84.4**
P3	C3D+SVM	88.4	9.6	10.7	13.7	15.8	10.7	11.7	11.7
P3	C3D+BikNN	91.0	36.0	31.4	21.2	34.2	24.6	35.8	12.8
P3	TSN+SVM	88.8	57.2	64.6	67.6	72.1	54.5	57.1	76.7
P3	TSN+BikNN	96.3	64.8	78.6	**85.7**	93.6	71.1	72.9	90.2
P3	OurF+SVM	96.7	54.9	67.8	75.5	80.1	57.3	59.5	82.8
P3	Our method	**96.8**	**84.4**	**93.5**	83.8	**94.1**	**84.9**	**87.4**	**93.1**
Ave	C3D+SVM	89.6	12.0	11.2	13.7	14.8	11.3	11.9	13.8
Ave	C3D+BikNN	92.2	30.8	28.1	18.8	12.7	23.7	28.2	5.2
Ave	TSN+SVM	90.4	53.3	59.2	58.1	69.5	46.9	48.2	70.6
Ave	TSN+BikNN	95.8	66.0	68.8	70.3	82.9	66.1	63.5	84.1
Ave	OurF+SVM	**96.9**	53.4	62.5	64.6	79.6	55.4	55.6	79.9
Ave	Our method	96.0	**79.9**	**92.0**	**72.6**	**90.3**	**79.8**	**80.1**	**90.1**

TABLE III
PERFORMANCE COMPARISONS FOR CROSS-SUBJECT

TD	Method	N	OD	D	O	S	SO	SOD	SD
P1	C3D+SVM	15.2	11.6	12.6	12.6	13.0	23.7	11.4	15.5
P1	C3D+BikNN	6.9	30.5	20.2	17.0	9.6	21.6	26.1	8.0
P1	TSN+SVM	52.8	38.5	43.6	39.8	50.9	38.2	34.3	51.7
P1	TSN+BikNN	60.6	46.7	58.6	57.6	72.7	56.0	53.8	58.3
P1	OurF+SVM	58.7	36.0	41.9	44.0	55.4	38.5	39.2	61.1
P1	Our method	**84.6**	**67.6**	**81.4**	**72.8**	**82.2**	**70.1**	**77.1**	**81.8**
P2	C3D+SVM	12.4	10.0	10.2	11.3	13.0	12.1	10.0	13.6
P2	C3D+BikNN	1.4	21.3	25.0	12.8	1.1	22.3	24.8	0.4
P2	TSN+SVM	54.6	41.6	41.0	39.4	47.7	37.4	39.1	40.3
P2	TSN+BikNN	43.0	63.4	64.2	57.5	55.0	54.2	52.8	48.7
P2	OurF+SVM	60.2	38.4	43.4	48.5	**60.2**	37.7	39.6	57.1
P2	Our method	**63.8**	**72.9**	**72.6**	**51.5**	59.2	**61.8**	**66.4**	**62.0**
P3	C3D+SVM	12.0	14.4	8.8	10.4	13.0	9.7	9.4	13.2
P3	C3D+BikNN	1.0	22.8	25.8	16.5	1.5	22.2	25.3	5.9
P3	TSN+SVM	54.4	40.6	42.3	39.9	51.5	36.7	36.3	52.7
P3	TSN+BikNN	78.6	75.3	64.8	59.6	**89.5**	61.0	58.7	**89.2**
P3	OurF+SVM	63.5	43.2	46.5	49.0	66.9	43.9	44.2	60.9
P3	Our method	**83.9**	**82.2**	**89.0**	**65.0**	78.2	**83.2**	**82.6**	80.9
Ave	C3D+SVM	13.2	12.0	10.5	11.4	13.0	15.1	10.3	14.1
Ave	C3D+BikNN	3.1	24.9	23.6	15.4	4.0	22.0	25.4	4.8
Ave	TSN+SVM	53.9	40.2	42.3	39.7	50.1	37.4	36.5	48.3
Ave	TSN+BikNN	60.7	61.8	62.5	58.2	72.4	57.1	55.1	65.4
Ave	OurF+SVM	60.8	39.2	43.9	47.2	60.9	40.0	41.0	59.7
Ave	Our method	**77.4**	**74.2**	**81.0**	**63.1**	**73.2**	**71.7**	**75.4**	**74.9**

Sports-1M, ResNet18 is used as its base network, the features from pool5 layer are used as the C3D features. TSN network is trained on UCF101. BNInception Net is used as its base network. The features from global_pool layer after average pooling are used as TSN features. First, we test the performance of the recognition algorithm, BikNN. Then, we compare our algorithm (our features and BikNN) with the other combination method. Moreover, we test our method by using the same-subject and cross-subject test. In the same-subject test, the data of one person in the normal case are used as training data, and those in other cases are used as test data. In the cross-subject test, the data of one person in the normal case are used as training data, and those in other cases of different persons are used as test data.

1) Same-Subject Evaluation: We test the performance of our algorithm with the same person. The sample of the normal data of the person is used as the training sample, and the other samples of the same person are used as the testing samples. The results are shown in Table II. In the table, TD, N, O, D, and S denote training data, normal, occlusion, disorder, and slow data, respectively.

Table II shows that our algorithm performs effectively when the speed or the temporal order changes. When occlusion occurs, the accuracy is comparatively low. However, contributing to the two Kinects, the accuracy is also not too bad. Comparing our method with state-of-the-art methods, for one-shot circumstance, we can see that our features and our recognition method are both better than the traditional methods. Using the same features, C3D or TSN, our BikNN can achieve better performance than SVM. Using the same recognition method, BikNN, our features achieve the best performance. Therefore, our method can do well in processing the problem. Besides, from this table, we can see that TSN is better than C3D, we think this is because that the optical flow is robust to the original data.

2) Cross-Subject Evaluation: In order to evaluate the performance of the algorithm to different persons, we perform the cross-subject experiments: Use one person as the training sample, then the other data are used to test the performance of the algorithm. The performance under different conditions are given in Table III.

Table III shows that although approximately 10% decay is observed in the average accuracy compared with the results from the same person, the accuracy is still above 70%. Moreover, we can obtain similar conclusions, as shown in Table II. Our method can achieve the state-of-the-art performance.

3) Evaluation of Different Parts of Our Scheme: We test the performance of the fusion algorithm. To test our algorithm, we first test it using the classification criterion and then the regression criterion. For the classification criterion, we first test the performance of our algorithm using the frontal skeletal (FS), frontal binary (FB), side skeletal (SS), and side binary (SB) data. When obstruction occurs, the frontal data

TABLE IV
PERFORMANCE COMPARISONS FOR DIFFERENT MODULES

Data	FS	FB	SS	SB	FSSB	Fusion
N	**85.99**	71.47	56.52	61.06	84.17	83.47
OD	67.09	55.97	48.94	61.68	71.97	**76.12**
D	**88.04**	69.64	50.34	59.54	82.53	84.7
O	54.75	45.89	47.77	53.96	60.43	**66.3**
S	**86.65**	65.81	48.04	58.15	80.39	78.88
SO	66.10	57.74	50.08	57.00	68.12	**74.42**
SOD	67.61	57.40	62.69	67.90	74.40	**76.95**
SD	**85.46**	60.26	49.81	59.89	79.19	79.95
Ave	75.21	60.52	51.77	59.90	75.15	**77.60**

TABLE V
PERFORMANCE COMPARISONS FOR WEIGHTING AND FILTERING

Method	N	OD	D	O	S	SO	SOD	SD
Weighting	77.5	64.2	76.9	60.7	67.9	65.7	64.8	67.7
Weighting+filtering	**83.5**	**76.1**	**84.7**	**66.3**	**78.9**	**74.4**	**77.0**	**80.0**

becomes corrupted or heavily noisy. The option is to use the silhouette data of the side data. Thus, we test the performance of the result of fusing the FS and SB (FSSB) data. Finally, we fuse the frontal and side data and named it fusion method. Table IV shows the following observations.

(1) The result of the FS is often better than that of the SS, especially without occlusion case. The quality of the skeletal data of the frontal data is better than that of the side data because our skeletal pose descriptor is view-invariant.

(2) The accuracy of the FB and SB is comparable when no occlusion exists. At the same time, the accuracy is often larger than 50%. This finding shows that the binary data has discriminative power for action recognition.

(3) The accuracy of the SS is closer to 50%. This finding shows that the SS data have some discriminative power for action recognition.

(4) The fusion accuracy of FSSB is better than the accuracy of the FS data alone. Thus, the side data can augment the accuracy of the algorithm. Moreover, the fusion accuracy of FS, SB, SS, and SB is better than the fusion of FSSB. Thus, the fusion of FB and SS data can augment the discriminative power of the FS and SB data.

We also test the performance of the weighting and filtering scheme. The results are shown in Table V. From Table V, we can see that using the adaptive effective radius, filtering can increase the performance of our algorithm.

The final fusion results are presented in Fig. 13. The figure shows that our final regression criterion is good in different cases.

Some examples for the regression criterion is shown in Fig. 14, considering that it is highly related to the classification criterion. The complementary degree is shown in Fig. 14.

4) Evaluation of the Efficiency: In addition, we test the efficiency of our algorithm. The running time of the program to report once is approximately 0.8 s on an HP personal computer with a CPU time of 3.2 GHz and a memory of 4 G.

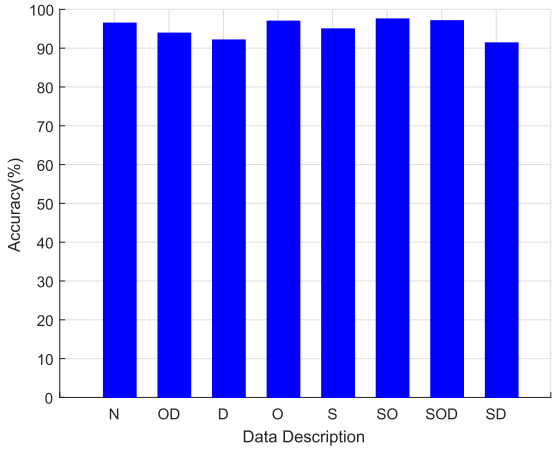
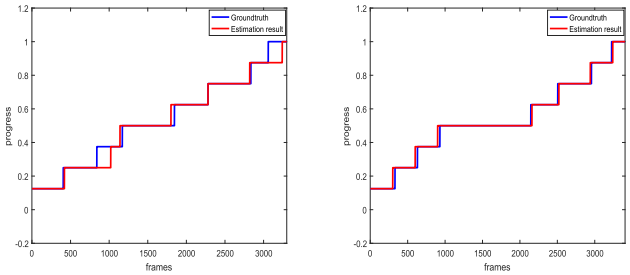


Fig. 13. Accuracy of the regression criterion.



(a) (b)

Fig. 14. Some regression results. (a) Result 1. (b) Result 2.

TABLE VI
DETAILED INFORMATION OF OAD

Actions	FN	SkeFN	TSNFN	C3DFN
Drinking	18	18	17	2
Eating	14	14	13	2
Writing	45	45	44	5
Opening cupboard	36	36	35	5
Opening microwave oven	22	22	21	2
Washing hands	31	31	30	4
Sweeping	66	66	65	8
Gargling	17	17	16	2
Throwing trash	36	36	35	4
Wiping	23	23	22	3

E. Evaluation on OAD Dataset

On OAD dataset, we use the same setting for our dataset. We first analyze the detailed information of OAD dataset. Then we test our algorithm on it. OAD contains some short videos. The detailed information of OAD is shown in Table VI, where FN, SkeFN, TSNFN, C3DFN represent the number of the frames of the video, the number of the features of skeleton, TSN and C3D respectively. Every video lasts in a short time and the number of the features extracted by C3D is small, therefore, it is of small significance to compare the action recognition results using C3D features. And we don't test the

TABLE VII
TEST RESULTS ON OAD

Method	Average accuracy
TSN+SVM	6.49
TSN+BikNN	60.13
Skeleton+SVM	5.71
Skeleton+BikNN	**63.57**

C3D related results. Other results are shown in Table VII. From the results, we can see that the proposed BikNN method performs better than SVM under the one-shot case.

V. CONCLUSION

We present a new problem, namely, visual EPE, which has been neglected in the research community. We show the significance of this problem and model it as a special SADI problem. Its specialty lies in the following aspects. It is a one-shot learning problem, while being an open-set problem that requires online output. Moreover, we propose a multi-modal feature fusion-based framework to solve the problem. The framework fuses the skeletal and silhouette data captured from two RGB-D sensors. By using this framework, the event progress can be estimated according to the action unit. In terms of methodology, a bidirectional kNN algorithm is proposed to extract representative and discriminative key poses from the training sample. Attention weighting scheme-based fusion lies between the feature and decision levels to handle the predicament of the high-dimensional features and one training sample. In addition, we construct our dataset to test our algorithm. Experimental results show that our scheme can be used to realize an EPE with acceptable performance.

REFERENCES

[1] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, no. 3, 2011, Art. no. 16.

[2] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1473–1488, Nov. 2008.

[3] A. Shahroudy, T.-T. Ng, Y. Gong, and G. Wang, "Deep multimodal feature analysis for action recognition in RGB+D videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1045–1058, May 2018.

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

[5] G. Chéron, I. Laptev, and C. Schmid, "P-CNN: Pose-based CNN features for action recognition," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3218–3226.

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

[7] P. Afsar, P. Cortez, and H. Santos, "Automatic visual detection of human behavior: A review from 2000 to 2014," *Expert Syst. Appl.*, vol. 42, no. 20, pp. 6935–6956, 2015.

[8] R. De Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, and T. Tuytelaars, "Online action detection," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 269–284.

[9] M. Takehara, S. Tamura, R. Tenmoku, T. Kurata, and S. Hayamizu, "The role of speech technology in service-operation estimation," in *Proc. Int. Conf. Speech Database Assessments (Oriental COCOSDA)*, Oct. 2011, pp. 116–119.

[10] R. Tenmoku *et al.*, *Service-Operation Estimation in a Japanese Restaurant Using Multi-Sensor and POS Data*. New York, NY, USA: Kennikat Press, 2011.

[11] X. Chang, Y.-L. Yu, Y. Yang, and E. P. Xing, "Semantic pooling for complex event analysis in untrimmed videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1617–1632, Aug. 2017.

[12] Y. Xiong *et al.*, "CUHK & ETHZ & SIAT submission to activityNet challenge 2016," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 1–4.

[13] T. Du, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2014, pp. 4489–4497.

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

[15] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1933–1941.

[16] L. Wang *et al.*, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 20–36.

[17] W. Lin, Y. Mi, J. Wu, K. Lu, and H. Xiong, "Action recognition with coarse-to-fine deep feature integration and asynchronous fusion," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 7130–7137.

[18] W. Lin *et al.*, "A tube-and-droplet-based approach for representing and analyzing motion trajectories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1489–1503, Aug. 2017.

[19] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4674–4683.

[20] Y. Li, W. Li, V. Mahadevan, and N. Vasconcelos, "VLAD3: Encoding dynamics of deep features for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1951–1960.

[21] X. Peng, L. Wang, X. Wang, and Y. Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," *Comput. Vis. Image Understand.*, vol. 150, pp. 109–125, Sep. 2016.

[22] B. Hartmann, C. Schauer, and N. Link, "Worker behavior interpretation for flexible production," *World Acad. Sci. Eng. Technol.*, vol. 3, no. 58, pp. 494–502, 2009.

[23] S. Tamura, T. Uno, M. Takehara, S. Hayamizu, and T. Kurata, "Multimodal service operation estimation using DNN-based acoustic bag-of-features," in *Proc. Eur. Signal Process. Conf.*, Aug./Sep. 2015, pp. 2291–2295.

[24] 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. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5344–5352.

[25] A. A. Chaaraoui, J. R. Padilla-López, and F. Flórez-Revuelta, "Fusion of skeletal and silhouette-based features for human action recognition with RGB-D devices," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 91–97.

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

[27] C. Liu, Y. Hu, Y. Li, S. Song, and J. Liu. (2017). "PKU-MMD: A large scale benchmark for continuous multi-modal human action understanding." [Online]. Available: https://arxiv.org/abs/1703.07475

[28] R. Poppe, "A survey on vision-based human action recognition," *Image Vis. Comput.*, vol. 28, no. 6, pp. 976–990, 2010.

[29] W. Liu, H. Liu, D. Tao, Y. Wang, and K. Lu, "Multiview Hessian regularized logistic regression for action recognition," *Signal Process.*, vol. 110, pp. 101–107, May 2015.

[30] W. Liu, Z.-J. Zha, Y. Wang, K. Lu, and D. Tao, "p-Laplacian regularized sparse coding for human activity recognition," *IEEE Trans. Ind. Electron.*, vol. 63, no. 8, pp. 5120–5129, Aug. 2016.

[31] T. Bagautdinov, A. Alahi, F. Fleuret, P. Fua, and S. Savarese, "Social scene understanding: End-to-end multi-person action localization and collective activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3425–3434.

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

[33] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3D points," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2010, pp. 9–14.
[34] A. W. Vieira, E. R. Nascimento, G. L. Oliveira, Z. Liu, and M. F. M. Campos, "On the improvement of human action recognition from depth map sequences using space–time occupancy patterns," *Pattern Recognit. Lett.*, vol. 36, no. 1, pp. 221–227, 2014.
[35] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1290–1297.
[36] X. Yang, C. Zhang, and Y. Tian, "Recognizing actions using depth motion maps-based histograms of oriented gradients," in *Proc. ACM Int. Conf. Multimedia*, 2012, pp. 1057–1060.
[37] Z. Cheng, L. Qin, Y. Ye, Q. Huang, and Q. Tian, "Human daily action analysis with multi-view and color-depth data," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 52–61.
[38] 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.
[39] C. Chen, M. Liu, B. Zhang, J. Jiang, J. Jiang, and H. Liu, "3D action recognition using multi-temporal depth motion maps and Fisher vector," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 3331–3337.
[40] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognit.*, vol. 68, pp. 346–362, Aug. 2017.
[41] Z. Ding, P. Wang, P. O. Ogunbona, and W. Li, "Investigation of different skeleton features for CNN-based 3D action recognition," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, Jul. 2017, pp. 617–622.
[42] J. Wan, G. Guo, and S. Z. Li, "Explore efficient local features from RGB-D data for one-shot learning gesture recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1626–1639, Aug. 2016.
[43] S. R. Fanello, I. Gori, G. Metta, and F. Odone, "Keep it simple and sparse: Real-time action recognition," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 2617–2640, 2013.
[44] M. R. Malgireddy, I. Nwogu, and V. Govindaraju, "Language-motivated approaches to action recognition," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 2189–2212, 2013.
[45] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception Psychophy*, vol. 14, no. 2, pp. 201–211, 1973.
[46] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," in *Proc. CVPR*, Jun. 2011, pp. 1297–1304.
[47] G. Tian, J. Yin, X. Han, and J. Yu, "A novel human activity recognition method using joint points information," *Robot*, vol. 36, no. 3, pp. 285–292, 2014.
[48] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition," *J. Vis. Commun. Image Represent.*, vol. 25, no. 1, pp. 24–38, 2014.
[49] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, and J. Liu, "Online human action detection using joint classification-regression recurrent neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 203–220.

Jianqin Yin received the Ph.D. degree from Shandong University, Jinan, China, in 2013.

She is currently a Professor with the Automation School, Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include service robot, pattern recognition, machine learning, and image processing.

Xiaoli Liu is currently pursuing the Ph.D. degree with the Automation School, Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include machine learning and image processing.

Fuchun Sun received the Ph.D. degree from Tsinghua University, Beijing, China, in 1997.

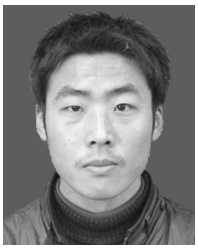
He is currently a Full Professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include intelligent control and robotics.

Dr. Sun was a recipient of the National Science Fund for Distinguished Young Scholars. He serves as an Associate Editor for a series of international journals including IEEE TRANSACTIONS ON FUZZY SYSTEMS, IEEE TRANSACTIONS ON MECHATRONICS, and IEEE TRANSACTIONS ON ROBOTICS AND AUTONOMOUS SYSTEMS.

Huaping Liu received the Ph.D. degree from Tsinghua University, Beijing, China, in 2004.

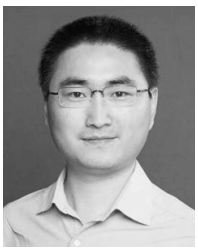
He is currently an Associate Professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include robot perception and learning.

Dr. Liu served as a Program Committee Member for RSS2016 and IJCAI2016. He serves as an Associate Editor for several journals including IEEE ROBOTICS AND AUTOMATION LETTERS, *Neurocomputing*, *Cognitive Computation*, and some conferences including the International Conference on Robotics and Automation and the International Conference on Intelligent Robots and Systems.



Zhiqiang Liu received the master's degree from University of Jinan, China. He is a Visitor Student with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include machine vision and image processing.

Bin Wang received the master's degree from Tsinghua University, Beijing, China. He is currently an Engineer with the Department of Computer Science and Technology, Tsinghua University. His research interests include soft engineering, machine vision, and image processing.



Jun Liu received the Ph.D. degree from University of Toronto, Toronto, Canada, in 2016.

He currently is a Post-Doctoral Fellow with the Dalio Institute of Cardiovascular Imaging, Cornell University, NY, USA. His research interests include micro-nano robotics and image analysis and interaction.

He have won multiple awards including the Best Student Paper Award and the Best Medical Robotics Paper Finalist Award from the IEEE International Conference on Robotics and Automation in 2014 and the IEEE Transactions on Automation Science and Engineering Best New Application Paper Award in 2018.

Yilong Yin received the Ph.D. degree from Jilin University, China, in 2000.

He is currently the Director of the MLA Group and a Professor with the School of Software Engineering, Shandong University, China. His research interests include machine learning and data mining.