International Journal of Control, Automation and Systems 16(5) (2018) 2393-2404
http://dx.doi.org/10.1007/s12555-017-0532-1
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Progressive Filtering Approach for Early Human Action Recognition

Tehao Zhu, Yue Zhou, Zeyang Xia*, Jiaqi Dong, and Qunfei Zhao

Abstract: Human action recognition plays an important role in vision-based human-robot interaction (HRI). In many application scenarios of HRI, robot is required to recognize the human action expressions as early as possible in order to ensure a suitable response. In this paper, we proposed a novel progressive filtering approach to improve the robot's performance in identifying the ongoing human actions and thus to enhance the fluency and friendliness of HRI. Human movement data were captured by a Kinect device, and then the human actions were constituted by the refined movement data using robust regression-based refinement. Motion primitive, including both spatial and temporal information concerning the movement, was considered as an improved representation of action features. Then, the early human action recognition was accomplished based on an improved locality-sensitive hashing algorithm, by which the ongoing input action can be classified progressively. The proposed approach has been evaluated on four datasets of human actions in terms of accuracy and recall curves. The experiments showed that the proposed progressive filtering approach achieves high recognition rate, and in addition, can make the recognition decision at an earlier stage of the ongoing action.

Keywords: Early human action recognition, human-robot interaction, locality-sensitive hashing, motion primitive, progressive filter.

1. INTRODUCTION

Human action recognition (HAR) is one of the most important procedures in vision-based human-robot interaction (HRI). The traditional HAR algorithms focus on the recognition of completed human actions. Therefore, a recognition algorithm can begin to work only after the actions have been finished. This latency limits the fluency and friendliness of HRI, regardless of how fast the algorithm is. However, in many application scenarios, such as surveillance, nursing care, education, sports, and entertainment, an HRI system is required to recognize unfinished human actions. Here, we hope to achieve early HAR, i.e., the recognition of human actions at their early stages [1–4] to solve the latency problem in HRI.

The major challenge for implementing early HAR is to infer the action class depending on the observed part of the action sequence. Existing approaches of early HAR have introduced several ideas to tackle this issue, such as sequence segmentation, feature representation for stream data, and model learning for ongoing action. Mori *et al*. [5] improved the dynamic programming of early recognition and motion prediction, and proposed the concept of gesture networks to deal with the ambiguity in an action's beginning. Ryoo [6] proposed an extended bag-of-words paradigm to represent human action, combined with dynamic programming to predict ongoing activities. Tormene *et al*. [7] used open-end dynamic time warping (OE-DTW) to compute the similarity between an incomplete input stream of data and a reference time series, and also built nearest-neighbor classifiers to achieve early recognition and accurate class prediction of human action. Bloom *et al*. [8] utilized temporal Laplacian eigenmaps and k-means to reduce the data dimensionality and extract key poses. Then the newly captured pose fragment was matched with templates by dynamic time warping. The lowest distance over all the actions represented the matched action class. Weber *et al*. [9] applied long short-term memory (LSTM) networks for recognizing human action patterns. The parameters in the networks are optimized using an evolutionary algorithm. Li and Fritz [10] defined a hierarchical label space to realize accuracy-specificity trade-offs. Their model can predict ongoing complex activities from a coarse result to a fine one. Vats *et al*. [11] used Bandler and Kohout's fuzzy sub-triangle product to model partial actions. The fuzzy capabilities of

Manuscript received September 2, 2017; revised February 11, 2018; accepted March 25, 2018. Recommended by Associate Editor Kang-Hyun Jo under the direction of Editor Euntai Kim. This work was supported by the Major Research Plan of the National Natural Science Foundation of China (No. 91646205), the National Natural Science Foundation of China (No. 61773365), and the Major Project of Guangdong Province Science and Technology Department (No. 2014B090919002).

Tehao Zhu, Yue Zhou, Jiaqi Dong, and Qunfei Zhao are with the School of Electronic Information and Electric Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mails: {zthjoe, zhouyue, saberfate, zhaoqf}@sjtu.edu.cn). Zeyang Xia is with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: zy.xia@siat.ac.cn).
* Corresponding author.

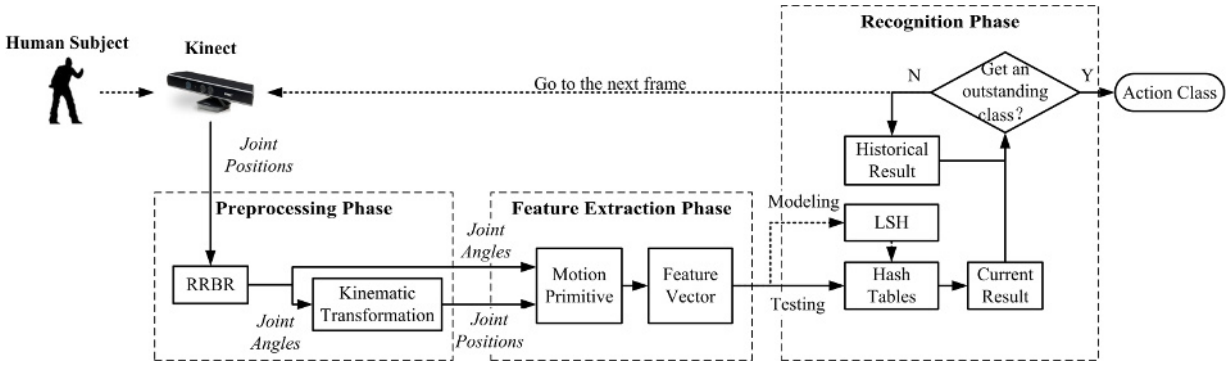

Fig. 1. Structure of the proposed approach.

this inference mechanism reduce uncertainties during the recognition of ongoing human actions. Ji *et al.* [12] adopted one-shot learning to automatically separate and define patterns in action sequences. The pattern transition maps were learnt using Q-learning. Finally, the early recognition was implemented based on soft-regression.

In several early HAR approaches, the key frames or poses need to be labeled in the preprocessing or training phase. Although this measure can minimize data redundancy, it is not convenient for practical application. Because if the HRI system is required to recognize a new action, the key frames or poses of the new templates must be also labeled. Moreover, to those approaches based on neural networks, changing the recognizable action classes may need to retrain the whole model.

The main contribution of this paper is proposing a novel approach called progressive filtering recognition (PFR) to improve the robot's performance in identifying the ongoing human actions and thus to enhance the fluency and friendliness of HRI. Our work includes several highlights: (1) the joint angles and distances are adopted to represent the human movements; (2) movement data in several frames are packaged as a motion primitive, including both spatial and temporal information; and (3) the early recognition is implemented based on an improved locality-sensitive hashing (LSH) algorithm.

The principal advantage of the proposed approach is that it requires neither the motion frames labeling, nor the complex action model training. The fast retrieval ability of LSH is utilized to establish an iterative recognition scheme, with the result that the ongoing input action can be classified progressively. Furthermore, when the recognizable actions are changed, only the matching relations associated with the changed actions in LSH need to update, while the rest part of LSH stays the same. So PFR will be very convenient, flexible and robust in practical applications.

The framework of the proposed approach is shown in Fig. 1. Chronological three-dimensional (3D) joint positions of the upper limbs are captured by a Kinect [13] device. In the preprocessing phase, a robust regression-based refinement algorithm developed in our previous work [14] is used to improve the accuracy and reliability of the captured joint position data. Then, the elbow and wrist positions are restored through a kinematic transformation, waiting for subsequent use. In the feature extraction phase, the refined joint angles and positions in several frames are packaged as a motion primitive, and its feature vector with powerful representative ability is then extracted. In the recognition phase, the hash tables are created using LSH. An iterative recognition scheme is established, and the ongoing input action is classified progressively.

The remainder of this paper is organized as follows. Section 2 introduces the preprocessing algorithm, concerned primarily with the principle of the regression-based refinement [14] algorithm. Section 3 proposes the feature extraction method. Section 4 provides the detailed scheme of the PFR. Section 5 shows the experimental results, and Section 6 presents a summary.

2. DATA PREPROCESSING

Kinect [13], a markerless human motion capture device developed by Microsoft, is the capture device in our system. It can capture 3D information of human joint positions in real time. In addition, Kinect is inexpensive and convenient to operate, and thus has vast potential for HRI applications. However, the 3D joint positions captured by Kinect are problematic in that the original data contain outliers and noises. The progressive filtering approach for early HAR focuses on daily interactive actions and commands, which are completed primarily by upper limbs [15, 16]. In our previous work [14], we proposed a robust regression-based refinement algorithm to preprocess the upper limb data captured by Kinect. We summarized the causes of the outliers in the Kinect data. Inverse kinematics was used to transform the captured 3D joint positions into joint angle values. Then, a stepwise robust regression strategy was designed to refine the joint angles from

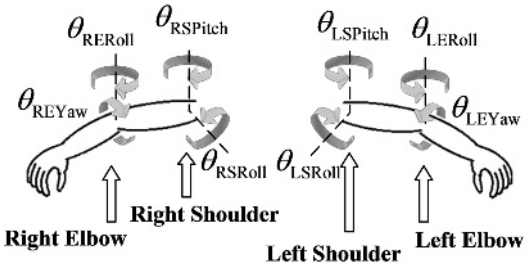

Fig. 2. Upper limb joints and angles.

the shoulder to the elbow. A Kalman filter was used to smooth the trajectories. As the elbow and wrist distances will be used in the subsequent feature extraction phase, elbow and wrist positions are restored through kinematic transformation based on refined joint angle data.

3. FEATURE EXTRACTION

3.1. Motion frame and action

We define the spatial information in a frame captured by Kinect as a *motion frame*. Several joint angles and distances are used to describe the spatial information. The advantage of using joint angle has been discussed in our previous work [14]. In order to facilitate perception and understanding the human action in HRI, the structure of humanoid robot can also be adopted to model the human's. In this work, the joint definitions of a humanoid robot NAO developed by Aldebaran Robotics [17] are used to summarize the human joint angles. As shown in Fig. 2, eight joint angles in the upper limbs are:

$$\boldsymbol{\theta}_{\text{LA}} = [\theta_{\text{LSPitch}} \quad \theta_{\text{LSRoll}} \quad \theta_{\text{LEYaw}} \quad \theta_{\text{LERoll}}], \tag{1}$$

$$\boldsymbol{\theta}_{\text{RA}} = [\theta_{\text{RSPitch}} \quad \theta_{\text{RSRoll}} \quad \theta_{\text{REYaw}} \quad \theta_{\text{RERoll}}]. \tag{2}$$

The joint distance reflects the position relationships between joints. In the following research, the joint distances between two elbows and wrists are used, representing by D_{E} and D_{W}, respectively.

Finally, the motion primitive includes above eight joint angles and two joint distances. We use $\boldsymbol{H}(t)$ to represent the content of the motion frame at the capture moment t:

$$\boldsymbol{H}(t)^{(10\times 1)} = [\theta_{\text{LA}}(t) \quad \theta_{\text{RA}}(t) \quad D_{\text{E}}(t) \quad D_{\text{W}}(t)]^{\text{T}}. \tag{3}$$

A series of motion frames constitute a complete and meaningful *action* $\boldsymbol{a}_p$:

$$\boldsymbol{a}_p = \left[\boldsymbol{H}(t_1)\, \boldsymbol{H}(t_2) \cdots \boldsymbol{H}(t_i) \cdots \boldsymbol{H}(t_{f_p})\right], \tag{4}$$

where f_p is the total frame number of $\boldsymbol{a}_p$, and t_i for $i = 1,2,\cdots,f_p$ are the moments when Kinect captures the spatial information.

3.2. Motion primitive

The motion frame contains only the spatial information, which cannot express the ongoing changes in action. In order to involve the temporal information, we package several motion frames with a d-width sliding window as a *motion primitive*. The motion primitive at t_i is formulated as follows:

$$\boldsymbol{S}(t_i) = [\boldsymbol{H}(t_{i-d+1}) \quad \cdots \quad \boldsymbol{H}(t_{i-1}) \quad \boldsymbol{H}(t_i)], \quad i \geq d, \tag{5}$$

where the value of d is related to the capture frequency. According to our experiment, $1/5$ of the capture frequency is a fine choice, i.e., packaging the data in 200 ms as a motion primitive. If d is too small, insufficient temporal information is included. On the other hand, if d is too large, the available iteration times are limited, as the durations of most actions are only a few seconds. Because the capture frequency of Kinect is 30 Hz, we set $d = 6$ in our experiment.

3.3. Feature vector of motion primitive

$\boldsymbol{S}(t_i)$ is a 10×6 matrix ($d = 6$). The vertical dimension of $\boldsymbol{S}(t_i)$ is the spatial dimension, whereas the horizontal dimension is the temporal dimension. To reduce the data size and preserve the typical information, the proper features must be extracted from the motion primitive.

The features must include the relationships among the data in both dimensions. For the spatial dimension, $D_{\text{E}}(t_i)$ and $D_{\text{W}}(t_i)$ contain the position relationships between joints. For the temporal dimension, the traditional relationship is the gradient among frames. However, the gradient of motion primitive might include style variance for different people. So we use the mean value of each spatial dimension in a motion primitive to construct the feature vector $\boldsymbol{F}(t_i)$:

$$\boldsymbol{F}(t_i) = \left[\bar{\boldsymbol{S}}_{1,:}(t_i) \quad \bar{\boldsymbol{S}}_{2,:}(t_i) \quad \cdots \quad \bar{\boldsymbol{S}}_{10,:}(t_i)\right]^{\text{T}}, \tag{6}$$

where $\boldsymbol{S}_{r,:}(t_i)$ for $r = 1,2,\cdots,10$ is the mean value of the r-th row in $\boldsymbol{S}(t_i)$. The mean value is more robust than the gradient, but it loses the change trend information. This issue is solved by using the temporal ordered $\boldsymbol{F}(t_i)$ in the subsequent progressive filtering recognition. In this case, both the temporal dynamics and spatial feature are considered. Accordingly, $\boldsymbol{F}(t_i)$ can improve the style invariant among similar motion primitives to some extent, although it has the same form as a single $\boldsymbol{H}(t_i)$.

In the subsequent modeling step of recognition phase, a certain amount of template actions are picked out from the database to model the feature mapping for early HAR, and the action classes are categorized by $\boldsymbol{C} = \{C_1, C_2, \cdots, C_g, \cdots, C_G\}$. Suppose that there are N motion primitives segmented from all of the template actions. Each motion primitive is denoted by $\boldsymbol{S}_n \in \boldsymbol{S}$, for $n = 1,2,\cdots,N$, and its feature vector is denoted by $\boldsymbol{F}_n \in \boldsymbol{F}$.

Table 1. Representative abilities of $\boldsymbol{S}$ and $\boldsymbol{F}$. "Avg." and "Std." are the average distance value and standard deviation, respectively, and "Avg. Ratio" is the ratio of the average inter-class distance to the average intra-class distance.

Form	Intra-class Dis.		Inter-class Dis.		Avg. Ratio
	Avg.	Std.	Avg.	Std.	
$\boldsymbol{S}$	2.8536	0.6129	7.1886	1.1141	2.5191
$\boldsymbol{F}$	1.0609	0.1921	2.8965	0.4939	2.7304

Additionally, we use $c_n \in \boldsymbol{C}$ to represent the action class that $\boldsymbol{S}_n$ belongs to, and use t_{f_n} to represent the appearing moment.

3.4. Representative ability

To verify the representative ability of $\boldsymbol{F}$, we use k-means method to compare the clustering effects of the motion primitives in the form of the original $\boldsymbol{S}$ and its feature vector $\boldsymbol{F}$. The larger the ratio between the inter-class distance and the intra-class distance is, the stronger the representative ability of the form is. The data used for this verification are from the MSRC-12 dataset developed by Microsoft Research and Cambridge University (introduced in Section 5). In the experiment, all of the motion primitives are gathered into ten clusters.

As k-means is sensitive to the initial cluster centroids, additional measures are required to ensure the reliability of the clustering results: The number of motion primitives in each cluster must be no less than 30% of the number by average division; the clustering was repeated ten times, and the average result was taken.

The clustering results are shown in Table 1. Although the intra-class distance and the inter-class distance when using $\boldsymbol{F}$ are both smaller than those when using $\boldsymbol{S}$, the ratio when using $\boldsymbol{F}$ is larger. Furthermore, the data size of $\boldsymbol{F}$ is only $1/6$ that of $\boldsymbol{S}$, with the result that the representative ability of $\boldsymbol{F}$ is more powerful.

4. PROGRESSIVE FILTERING RECOGNITION

Consider how we human beings recognize others' actions in our daily life. During the interaction, we observe others movement continuously to speculate about and predict their actions. When we are sure what others want to perform, we will make our reactions. Similarly, we hope the recognition approach can identify the ongoing action progressively based on the accumulated human movement data.

In the recognition phase, a progressive filter for recognizing the ongoing action is proposed based on LSH [18, 19], which is a retrieval method with the ability of finding similar motion primitives in the massive sample data rapidly and efficiently. It is unnecessary to label samples or select representative postures for the progressive filter, and unnecessary to retrain the filter model totally when the recognizable actions are changed.

4.1. Locality-sensitive hashing

To recognize human action, locality-sensitive hashing (LSH) [18,19] is introduced to search the best-matched action class. LSH overcomes the poor efficiency of searching massive and high-dimensional data that arises from using traditional retrieval methods. The basic principle of LSH is to map the original data to several "buckets" through a series of hash functions. Then, the retrieval task works in only one bucket [20]. Often, we use several groups of hash functions to enhance LSH. Each group is called a hash table. Integrating the results of multiple hash tables can reduce the matching error rate. In this work, the normal Gaussian distribution is chosen as the form of the hash function [21]:

$$h(\boldsymbol{F}_n) = \left\lfloor \frac{\boldsymbol{R}\boldsymbol{F}_n + b}{w} \right\rfloor, \tag{7}$$

where w is the width of the bucket, R is the random-number-based Gaussian distribution vector, and b is the uniform distribution offset in the range of $[0, w]$. As for one hash function, all the feature vectors of motion primitives are mapped to several hash values. Then we have:

$$\boldsymbol{b}^u = \{\boldsymbol{F}_n | n = 1, \cdots, N, h(\boldsymbol{F}_n) = v(u)\}, \tag{8}$$

where U is the total number of hash values that $\boldsymbol{F}_n$ are mapped to, $u = 1, \cdots, U$ is the hash value index, and $\boldsymbol{b}^u$ is the set consisted by $\boldsymbol{F}_n$ whose $h(\boldsymbol{F}_n)$ are equal to hash value $v(u)$.

Let K be the number of hash tables and L be the number of hash functions in each hash table. The hash function can be redefined based on (7):

$$h_{k,l}(\boldsymbol{F}_n) = \left\lfloor \frac{\boldsymbol{R}_{k,l}\boldsymbol{F}_n + b_{k,l}}{w} \right\rfloor, \tag{9}$$

where $k = 1, \cdots, K$ and $l = 1, \cdots, L$ are the indices of hash tables and hash functions. Then the set of $\boldsymbol{F}_n$ whose $h_{k,l}(\boldsymbol{F}_n)$ are the same is denoted by $\boldsymbol{b}^u_{k,l}$, and the corresponding hash value is denoted by $v_{k,l}(u)$.

4.2. Progressive filtering

Standard LSH cannot implement recognition directly unless the motion primitives have already been labeled. Here we devised an iterative scheme to avoid the labeling process, and the ongoing action can be classified progressively.

The hash value of $\boldsymbol{F}(t_i)$ is acquired by each hash function, denoted by $h_{k,l}(\boldsymbol{F}(t_i))$. The template motion primitives whose feature vectors have the same hash value are selected. In each $\boldsymbol{b}^u_{k,l}$, we limit the searching range to a 2d-frame-width before and after the current frame. This

measure directly excludes the motion primitives whose t_{f_n} values are apparently not in the same range as t_i. As a result, the incorrect action classes those $\boldsymbol{F}_n$ belong to can be eliminated effectively. The selected feature vectors of motion primitives $\tilde{\boldsymbol{b}}_{k,l}(t_i)$ can be given as follows:

$$\tilde{u} = \{u | v_{k,l}(u) = h_{k,l}(\boldsymbol{F}(t_i))\}, \tag{10}$$

$$\tilde{\boldsymbol{b}}_{k,l}(t_i) = \left\{\boldsymbol{F}_n | \boldsymbol{F}_n \in \boldsymbol{b}_{k,l}^{\tilde{u}}, |i - f_n| < 2d\right\}. \tag{11}$$

The matched feature vectors of motion primitives from all of the hash tables are integrated as:

$$\boldsymbol{B}_k(t_i) = \bigcap_{l=1}^{L} \tilde{\boldsymbol{b}}_{k,l}(t_i), \tag{12}$$

$$\boldsymbol{B}(t_i) = \bigcup_{k=1}^{K} \boldsymbol{B}_k(t_i). \tag{13}$$

We use $\boldsymbol{P}(t) = [p_1(t_i), p_2(t_i), \cdots, p_g(t_i), \cdots, p_G(t_i)]$ to represent the temporary recognition percentages of the action classes at t_i, where $p_g(t_i)$ is the proportion of the feature vectors of motion primitives in $\boldsymbol{B}(t_i)$ that belong to C_g:

$$p_g(t_i) = \frac{size\left(\{\boldsymbol{F}_n | \boldsymbol{F}_n \in \boldsymbol{B}(t_i), c_n = C_g\}\right)}{size\left(\boldsymbol{B}(t_i)\right)}, \tag{14}$$

where $size(\boldsymbol{X})$ represents the number of elements in $\boldsymbol{X}$, and $\sum_{g=1}^{G} p_g(t_i) = 1$.

The integrated recognition result is iteratively fused by the $\boldsymbol{P}(t_i)$ of multiple frames during the ongoing actions. Various classes of actions are often similar at the beginning, and their differences become salient as time passes. Accordingly, the new captured motion primitives are more helpful for recognition than the earlier ones. In other words, the new $\boldsymbol{P}(t_i)$ must be assigned a heavier fusion weight than the earlier ones. A temporal mask $\boldsymbol{M}$ including six elements is defined to assign the fusion weights, and then the integrated recognition percentages for multiple frames can be represented by $\boldsymbol{Q}(t_i)$:

$$\boldsymbol{M} = [m_h \quad m_1 \quad m_2 \quad \cdots \quad m_5], \tag{15}$$

$$\begin{aligned}\boldsymbol{Q}(t_i) &= [q_1(t_i)\ q_2(t_i)\ \cdots\ q_g(t_i)\ \cdots\ q_G(t_i)] \\ &= M \begin{bmatrix} \boldsymbol{Q}(t_{i-1}) \\ \boldsymbol{P}(t_{i-4}) \\ \vdots \\ \boldsymbol{P}(t_{i-1}) \\ \boldsymbol{P}(t_i) \end{bmatrix}, \quad i \geq d+5,\end{aligned} \tag{16}$$

where m_h is the weight for the previous integrated recognition percentages $\boldsymbol{Q}(t_{i-1})$, and m_1 m_5 are the weights for the latest five $\boldsymbol{P}(t_i)$. When $d \leq t < d+5$, i.e., at the beginning frames of an action, we only use a portion of M. We have:

$$\boldsymbol{Q}(t_i) = [m_{5+d-i} \ \cdots \ m_5] \begin{bmatrix} \boldsymbol{P}(t_d) \\ \vdots \\ \boldsymbol{P}(t_i) \end{bmatrix}, \quad d \leq i < d+5. \tag{17}$$

The value of m_h and m_1 m_5 must satisfy $m_h < m_5$, $m_1 < m_2 < \cdots < m_5$ and $m_h + m_1 + \cdots + m_5 = 1$. Based on repeated trials, the following setting of $\boldsymbol{M}$ is suitable for various situations:

$$\boldsymbol{M} = [0.2 \quad 0.06 \quad 0.11 \quad 0.16 \quad 0.21 \quad 0.26]. \tag{18}$$

If $\max_{g=1,\cdots,G} q_g(t_i)$ is greater than the given threshold q_{T}, we consider that the action class has been recognized and is $C_{\arg\max_{g=1,\cdots,G} q_g(t_i)}$. Otherwise, we store the result of the current frame, and continue on to the recognition procedure of the next frame.

The complete procedure of PFR is given in Algorithm "Progressive filtering recognition". There are no procedures of clustering or labeling massive motion primitives in this algorithm. This algorithm is very flexible when the recognizable actions are changed: The newly added feature vectors of motion primitives are assigned to the corresponding buckets based on the hash values, the feature vectors of motion primitives belonging to the deleted action classes are discarded directly, and the mappings for the feature vectors of motion primitives belonging to the unchanged action classes do not require any changes.

5. EXPERIMENTAL RESULTS

We evaluated different early HAR algorithms on four datasets: MSRC-12 [22], MAD [23], CR-UESTC [24], and a newly built daily interactive action (DIA-9) dataset. Three groups of experiments were designed: First, the performances of different features (joint positions, distances between joints, and the angles and distances proposed in Section 3) were evaluated to certify the effectiveness of the proposed one. Then, the proposed PFR was compared with two algorithms, OE-DTW [7] and LSTM [9]. In the above two experiments, the performances are compared in terms of accuracy and recall curves. Finally, the process time of PFR was analyzed, and more effective forms of PFR were discussed.

5.1. Dataset introduction

MSRC-12 dataset: The MSRC-12 dataset was developed by Microsoft Research and Cambridge University [22,25]. The actions were captured by Kinect. Thirty participants performed 12 classes of actions, of which five actions (lift arms, push right, goggles, wind it up, and change weapon, as shown in Fig. 3 [26]) are mainly upper limb movements. Every participant performed each class of action ten times.

Algorithm 1: Progressive filtering recognition

Determine the number of hash tables K and the number of hash functions L in each hash table.
In each hash table, use (9) to group all of the feature vectors of motion primitives into several buckets.
Determine the temporal mask $\boldsymbol{M}$ and the recognition threshold q_{T}.
if $i < d$ **then**
    Go to t_{i+1}.
end if
Generate the current motion primitive $\boldsymbol{S}(t_i)$ by (5), and extract $\boldsymbol{F}(t_i)$ from $\boldsymbol{S}(t_i)$ by (6).
Calculate the hash values $h_{k,1}(\boldsymbol{F}(t_i)), \cdots, h_{k,L}(\boldsymbol{F}(t_i))$ in hash table k by (9).
Select similar feature vectors from all of the hash tables to form $\boldsymbol{B}(t_i)$ by (10)-(13).
Calculate $\boldsymbol{P}(t_i)$ of $\boldsymbol{B}(t_i)$ following (14).
if $t \leq d+5$ **then**
    Use (16) to calculate $\boldsymbol{Q}(t_i)$.
else
    Use (17) to calculate $\boldsymbol{Q}(t_i)$.
end if
Check the maximum value $q_{\hat{g}}(t_i)$ in $\boldsymbol{Q}(t_i)$.
if $q_{\hat{g}}(t_i) > q_{\mathrm{T}}$ **then**
    The action is recognized. Output $C_{\hat{g}}$ and the algorithm ends.
else if the action is not finished **then**
    Preserve $\boldsymbol{P}(t_i)$ and $\boldsymbol{Q}(t_i)$, go to Step 4, and recognize the subsequent motion primitive at t_{i+1}.
else
    The recognition fails. No result is output. The algorithm ends.
end if

Daily interactive action (DIA-9) dataset: As the actions in MSRC-12 are not daily interactive actions, we specially record some interactive actions using Kinect and establish a dataset. We designed nine actions (shown in Fig. 4) and invited three participants to perform them. Each action was performed 20 times by each participant.

Multimodal action database (MAD): The MAD database was collected by Carnegie Mellon University [23]. This dataset contains the multimodal activities of 20 subjects recoded with Kinect. Each subject repeats the set of 35 actions twice. We choose 26 kinds of actions mainly performed by upper limbs in our experiment, which are the actions of No. 6-15, 18-26, and 30-35 according to the action list [27]. As there are so many action categories in MAD dataset, it will be a great challenge for recognition algorithms to distinguish similar actions and achieve high precision.

CR-UESTC interaction database: The CR-UESTC database was proposed by University of Electronic Science and Technology of China [24]. The movement data

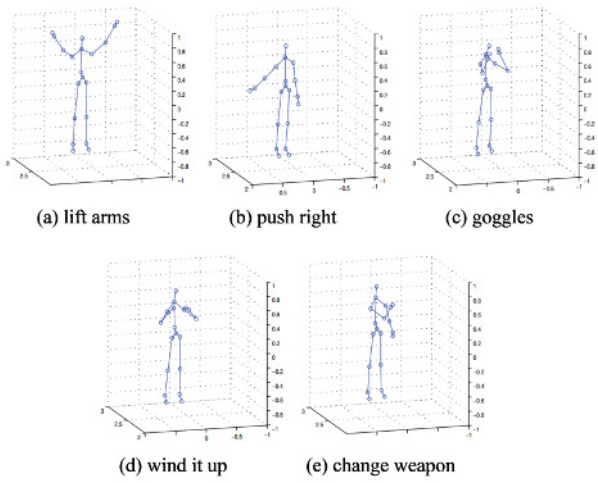

Fig. 3. Selected actions in MSRC-12 dataset.

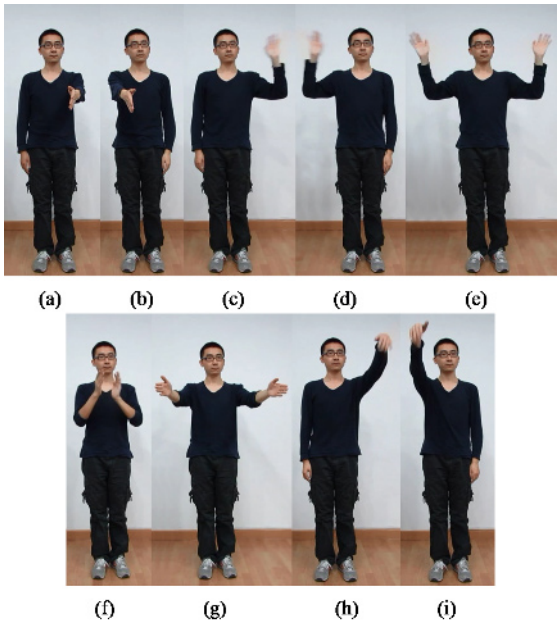

Fig. 4. Actions of DIA-9 dataset. (a) Shake left hand, (b) shake right hand, (c) wave left hand, (d) wave right hand, (e) wave both hands, (f) clap, (g) hug, (h) come here (left), and (i) come here (right).

were captured by the Kinect 2.0. This database contains 10 interaction categories acted by 25 pairs of persons. Eight action categories shown in Fig. 5 are mainly completed by upper limbs, so they are chosen in our experiments. As the subjects faced Kinect sideways, and two persons' skeletons sometimes overlapped each other, the data were noised greatly. The early HAR will be much more difficult on this database comparing with on the above ones. In addition, only the skeleton data of the person who dominates the interaction (i.e., the left person in the pictures of Fig. 5) are used in our experiments.

Table 2. Accuracies (Acc.) and timelinesses (Tim.) of different features on four datasets.

Feature	MSRC-12		DIA-9		MAD		CR-UESTC	
	Acc.	Tim.	Acc.	Tim.	Acc.	Tim.	Acc.	Tim.
Position	86.41%	25.48%	88.34%	19.82%	74.13%	29.47%	66.00%	37.80%
Distance	88.24%	28.13%	91.20%	28.53%	72.78%	28.83%	69.00%	39.13%
Angle+Distance	96.00%	30.12%	95.71%	29.87%	79.23%	25.46%	74.00%	39.74%

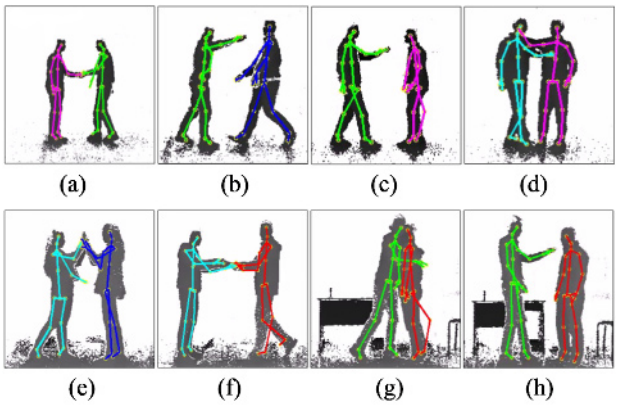

Fig. 5. Selected Actions in CR-UESTC dataset. The skeleton data of the left person in the pictures are used. (a) Handshake, (b) push, (c) punch, (d) arm round shoulder, (e) high five, (f) handover, (g) hug, and (h) pat shoulder.

5.2. Performance evaluation

We use two curves, the accuracy and recall curves, to evaluate the performance of the early HAR algorithms. The horizontal axes of the two curves are the normalized time to detection (NTtoD) [28], i.e., the completed percentage of the action when an algorithm outputs the recognition result. The vertical axes of the two curves are the accuracy and recall, respectively. The definitions of accuracy and recall are:

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{TN}+\text{FN}}, \tag{19}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}, \tag{20}$$

where TP, FP, TN, and FN are the number of true positive, false positive, true negative, and false negative results, respectively. The recognition algorithm can get a set of NTtoD, accuracy, and recall values under a given detection threshold. When the threshold takes different values, the combinations of (NTtoD, Accuracy) and (NTtoD, Recall) can form the accuracy and recall curves.

The accuracy curve evaluates the accuracy and timeliness of an algorithm. The final accuracy of an algorithm is represented by the peak value of the accuracy curve, and the corresponding NTtoD is considered as the timeliness. The recall curve evaluates the reliability of the recognized result, and the ideal shape of it is a monotonically increasing curve.

5.3. Experimental results

Exp. 1: PFR with different features. In Section 3, the motion primitive is composed of joint angles and distances. To certify this feature's performance, we compared it with other two sets of features. One includes the elbow and wrist positions. The other includes the joint distances in intra-inter-frames [24]. To the original latter feature, the joint distances refer to relative distances for interactive body pairs of two persons. In our experiment, we modified them to the distances for the joints of the left arm and those of the right arm.

The size of hash tables LSH are set to $K = 5$ and $L = 100$. According to our experiment, the best bucket width w is set to 3, 5, and 25 for position, distance, and the proposed "angle+distance" features, respectively. The recognitions are executed under different q_{T} from 0% to 100% to form the accuracy and recall curves. If $q_{\hat{g}}(t_i)$ is larger than a given q_{T}, then the corresponding action class is considered as the recognition result.

The experimental results are shown in Fig. 6. First, the recall curves using different features are monotonically increasing, meaning all the recognition results are reliable and trustworthy. Then, it is very clear that the peak values of four accuracy curves for "angle+distance" feature are the highest, thus the accuracy of the proposed "angle+distance" feature is the best.

The specific values of accuracy and timeliness values are listed in Table 2. The comprehensive performance of "angle+distance" feature is competitive. Although the timelinesses of position and distance features on MSRC-12, DIA-9, and CR-UESTC datasets are better than that of "angle+distance" feature, the advantage of the accuracy of "angle+distance" feature is remarkable. After all, a faster but wrong result is worthless.

Exp. 2: Early HAR by different algorithms. In this experiment, we compared the proposed PFR with two algorithms, OE-DTW [7] and LSTM [9], on multiple datasets. The motion primitive in terms of "angle+distance" is used as the feature. For our PFR, the parameters remained the same as those used in Exp. 1. For OE-DTW, we chose the Sakoe-CChiba band [29] as the global path restriction method, with the width set to 15. For LSTM, all the weights were initialized from a stan-

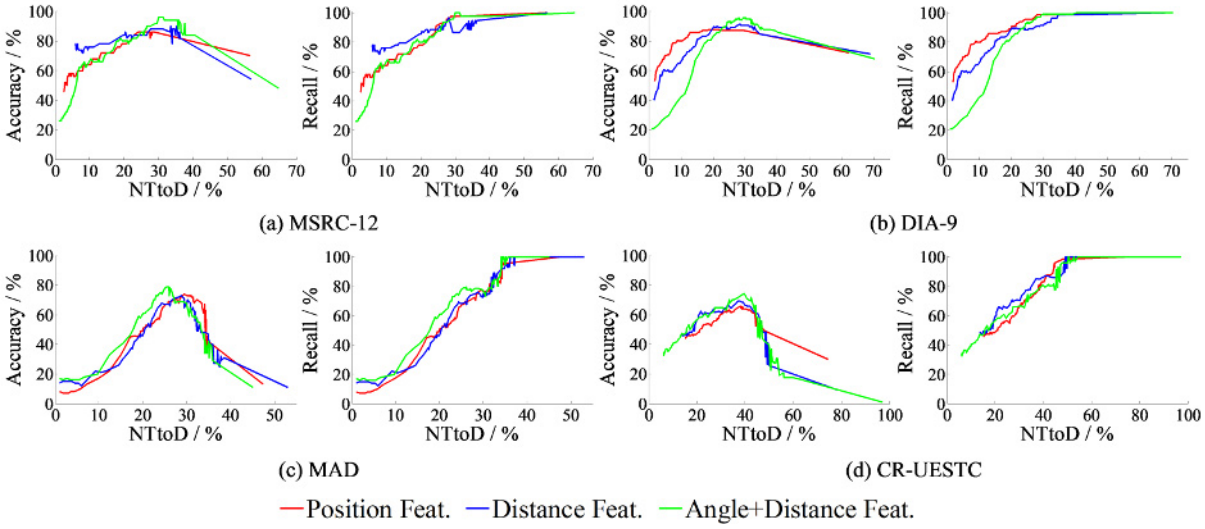

Fig. 6. Accuracy and recall curves of PFR with different features on four datasets.

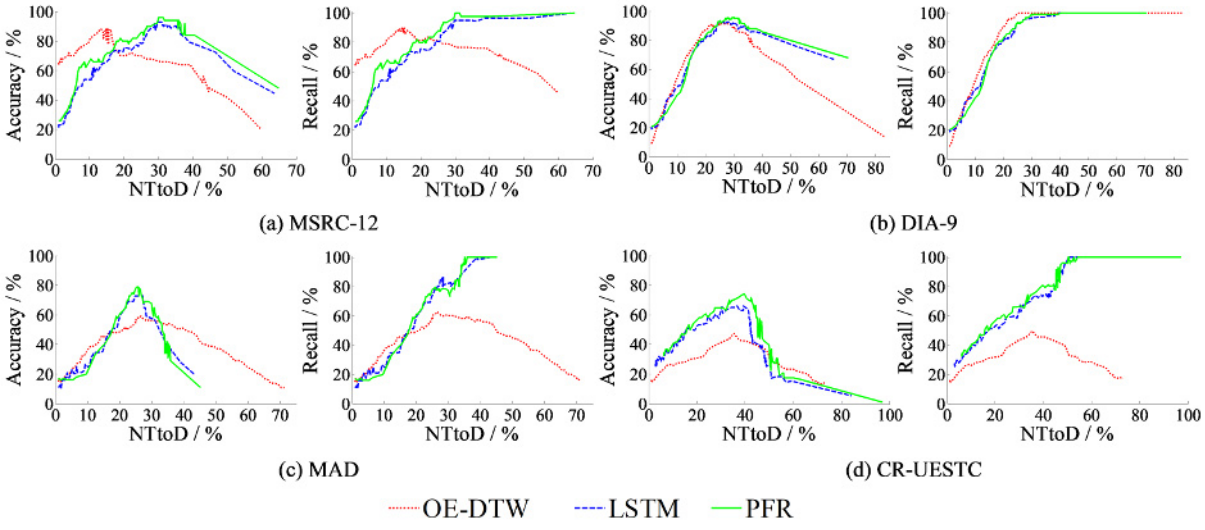

Fig. 7. Accuracy and recall curves of three algorithms on four datasets.

dard Gaussian distribution followed by a SVD orthogonalization [30], and the result was obtained using the softmax function.

The accuracy and recall curves for the compared two algorithms are generated when different ratios or thresholds are selected. For OE-DTW, if the ratio between the shortest distance and the second shortest is less than the given ratio, then the action class that the shortest distance corresponds to is the recognition result. For LSTM, if the softmax result is larger than the given threshold, then the corresponding action class is considered as the recognition result.

The experimental results are illustrated in Fig. 7. The performances of different algorithms on MSRC-12 dataset are shown in Fig. 7(a). Although OE-DTW is the fastest algorithm according to the accuracy curve (its timeliness is about 15%), the accuracy is only 88%. And according to the recall curve, OE-DTW is unreliable for the MSRC-12 dataset, as the recall of OE-DTW decreases when the ratio is strict (less than 0.4). LTSM's peak accuracy is 93%, whereas PFR's is 96%, the highest among the three algorithms. The timelinesses of LSTM and PFR are both about 30%. In contrast to OE-DTW, the recall curves of LSTM and PFR present a monotonically increasing trend, meaning both algorithms are reliable.

In Fig. 7(b), the accuracy and recall curves generated by different algorithms on DIA-9 dataset are presented. The results of the three algorithms are similar. For the accuracy curve, OE-DTW achieves its peak value of 91% at a NTtoD of approximately 27%, which is the fastest. LSTM achieves its peak accuracy of 92.7% at a NTtoD of approximately 28%. PFR achieves the best peak accuracy

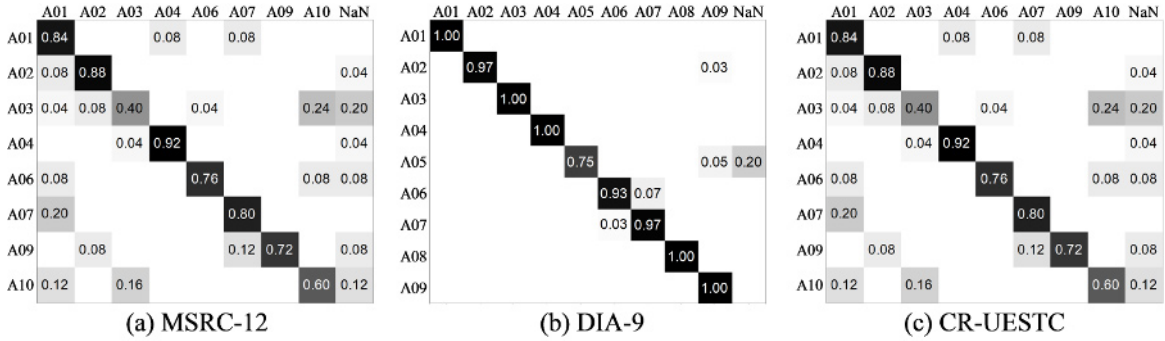

Fig. 8. Confusion matrices of PFR on MSRC-12, DIA-9, and CR-UESTC datasets.

of 95.7%, but its NTtoD is also the largest, approximately 30%. For the recall curves, all three algorithms present a monotonically increasing trend, in which the OE-DTW curve reaches 100% recall first.

The experimental results on MAD dataset are shown in Fig. 7(c). As there are up to 26 kinds of actions, the accuracy of each algorithm is lower than that on the above two datasets. The peak accuracy of PFR is approximately 80%, while those of OE-DTW and LSTM are approximately 60% and 73%, respectively. As for timeliness, the three algorithms are similar (approximately 25%). The recall curve of OE-DTW is just like its performance on MSRC-12 dataset, indicating that the recognition result is unreliable. On the contrary, the recall curves of LSTM and PFR take on similar monotonically increasing trend.

The two evaluation curves formed by the three algorithms on CR-UESTC dataset are illustrated in Fig. 7(d). The noises and outliers contained in the skeleton data caused huge trouble for each algorithm. We can see that OE-DTW can only achieves the accuracy less than 50%, and the recognition results are still unreliable according to its recall curve. The peak accuracies of LSTM and PFR are 66% and 74%, respectively. The timelinesses of OE-DTW and LSTM are about 36%, while that of PFR is a little slower (approximately 40%).

Furthermore, we show the confusion matrices of PFR on the four databases in Figs. 8 and 9, where each column represents the recognized class and each row represents the actual class. The column "NaN" represents the action cannot be recognized. The confusion matrices on MSRC-12 and DIA-9 indicate that almost all the actions are recognized correctly. However, on the challenging MAD and CR-UESTC datasets, several actions are quite similar, so the recognition results are confused among them, or even PFR cannot provide a decision. For example, the action "left/right arm wave (A08/A20)" is similar with "left/right arm swipe to the right/left (A07/A19)" and "left/right arm dribble (A10/A22)" in MAD to some extent, and the actions "punch (A03)" and "pat shoulder (A10)" in CR-UESTC look the same. As a result, the accuracies of the mentioned classes are lower than average.

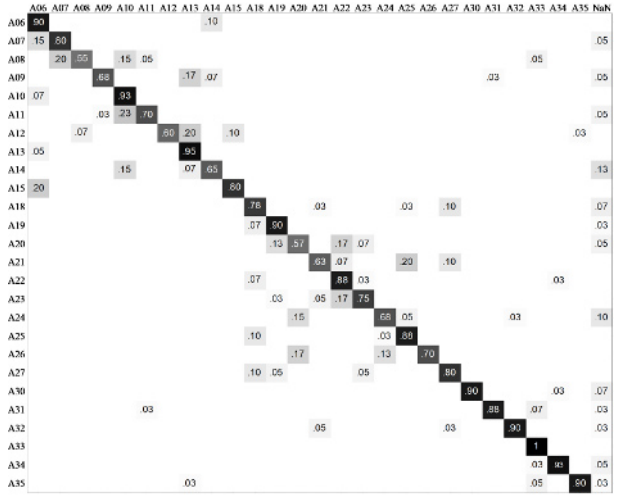

Fig. 9. Confusion matrix of PFR on MAD dataset.

Exp. 3: Processing time of PFR. The main processing time of PFR spends on hash table retrieval, and the retrieval time depends on the number of template motion primitives. We ran the PFR approach on a computer with 2.50 GHz Intel Core i5 CPU and 8 GB RAM. The OS is Windows 10 64-bit, and the working software is MATLAB.

Under this circumstance, the processing time per fame is 0.15s when the number of template motion primitives is about 3000. Let's consider an action containing $\mathcal{F}$ frames. In Fig. 8, the timeliness of PFR is about 30% in general, so the action can be recognized at the frame of about $0.3\mathcal{F}$. The processing time of PFR is $0.3\mathcal{F} \times 0.15 = 0.045\mathcal{F}(s)$. As the capture frequency of Kinect is 30 Hz, the total action lasts $\mathcal{F}/30 \approx 0.03\mathcal{F}(s)$.

The processing time can be reduced to satisfy the practical requirement by changing the recognition frequency. If PFR works per two frames, the processing time per action will reduce to $0.045\mathcal{F}/2 = 0.0225\mathcal{F}(s)$ theoretically. And if PFR works per three frames, the processing time per action will be $0.045\mathcal{F}/3 = 0.015\mathcal{F}(s)$, which is half time of the total action. We verified this measure by experiment. The results indicated that the actual processing

Table 3. Accuracies (Acc.) and timelinesses (Tim.) of different algorithms on four datasets.

Algorithm	MSRC-12		DIA-9		MAD		CR-UESTC	
	Acc.	Tim.	Acc.	Tim.	Acc.	Tim.	Acc.	Tim.
OE-DTW	88.00%	15.02%	91.11%	26.82%	58.56%	26.62%	47.50%	35.30%
SOM	90.80%	50.05%	92.40%	46.53%	64.42%	48.76%	57.50%	62.15%
LSTM	92.96%	30.74%	92.72%	28.14%	73.00%	24.90%	66.00%	36.74%
PFR	96.00%	30.12%	95.71%	29.87%	79.23%	25.46%	74.00%	39.74%

time coincides with the above derivations. Meanwhile, the accuracies and timelinesses when running PFR per two or three frames are almost the same as those in Table 3.

The number of template motion primitives can be controlled by taking one motion primitives every few frames. Alternatively, when the motion primitives of a template action are mapping to the hash tables in time order, the latest motion primitive and the previous one should have differences to some extent. Otherwise, the latest motion primitive can be discarded.

5.4. Discussion

Additionally, we compared the self-organizing map and LSH (SOM+LSH) based method proposed in [19]. The parameters of SOM+LSH method were set according to [19]. As there were no changeable thresholds for obtaining accuracy and recall curves in this method, we can only give the average accuracy and timeliness values. The accuracies and timelinesses of different algorithms are listed in Table 3.

OE-DTW [7] matches the action subsequences by temporal scaling, so its matching scope is wider than others', and similar templates are more. However, the incomplete action with tiny variation may be closer to some subsequence of an incorrect class in unexpected scales. As a result, OE-DTW is the fastest, but its precision is the lowest.

SOM+LSH [19] gets a fairly nice accuracy, indicating that the idea of matching the current gesture with representative ones is effective. However, the timeliness is much worse than other algorithms due to the fact that the representative gestures often appear in the middle or rear part of an action. In addition, the training of SOM networks is time-consuming.

The precision of LSTM [9] is further improved because of the recurrent network structure. Though the timeliness is not as good as OE-DTW, it is still satisfying. This result indicates that the temporal continuity is important to early HAR.

The iterative structure of the proposed PFR approach ensures the robust and reliability of the online recognition result. The outstanding action class is determined by continuous filtering, which also considers the temporal continuity. The result showed that the precision of PFR is the highest, and the timeliness is similar to that of LSTM. Moreover, if the recognizable actions are changed, PFR only needs to modify the parameters associated with the changed actions, but LSTM should retrain the networks. This is the advantage of PFR.

6. CONCLUSION

In many application scenarios of HRI, robot is required to recognize the human action expressions as early as possible in order to ensure a suitable response. For this purpose, we proposed a novel approach called progressive filtering recognition (PFR) to improve the robot's performance in identifying the ongoing human actions. There were three phases in PFR. In the preprocessing phase, human actions captured by Kinect were refined through robust regression-based refinement. Then, in the feature extraction phase, the refined data were packaged into motion primitives, and the feature vector with powerful representative ability was extracted. Finally, in the recognition phase, a novel progressive filtering approach based on LSH was designed to classify the ongoing input action progressively. The principal advantage of PFR is that it requires neither the motion frames labeling, nor the complex action model training, and is well-adapted when the recognizable actions are changed.

We evaluated PFR on four datasets. The experiments showed that PFR is more competitive than other algorithms. PFR achieves high recognition rate, and in addition, can make the recognition decision at an earlier stage of the ongoing action.

In future work, we aim to make PFR more robust to occlusion, abnormal movement, and missing data, as it can be seen there is still room for improvement on MAD and CR-UESTC datasets. Furthermore, we intend to enhance the PFR approach to recognize the whole body actions, and thus make an extension to cover more application scenarios, such as surveillance, nursing care, and medical rehabilitation.

REFERENCES

[1] M. Barnachon, S. Bouakaz, B. Boufama, and E. Guillou, "Ongoing human action recognition with motion capture," *Pattern Recognition*, vol. 47, no. 1, pp. 238-247, January 2014.

[2] Y. M. Chen, Z. Y. Ding, Y. L. Chen, and X. Y. Wu, "Rapid recognition of dynamic hand gestures using leap motion," *Proc. of IEEE International Conf. on Information and Automation*, pp. 1419-1424, August 2015.

[3] M. Kawashima, A. Shimada, H. Nagahara, and R.-I. Taniguchi, "Adaptive template method for early recognition of gestures," *Proc. of 17th Korea-Japan Joint Workshop on Frontiers of Computer Vision*, pp. 1-6, February 2011.

[4] R. Muscillo, M. Schmid, S. Conforto, and T. D'alessio, "Early recognition of upper limb motor tasks through accelerometers: real-time implementation of a DTW-based algorithm," *Computers in Biology and Medicine*, vol. 41, no. 3, pp. 164-172, March 2011.

[5] A. Mori, S. Uchida, R. Kurazume, R. Taniguchi, T. Hasegawa, and H. Sakoe, "Early recognition and prediction of gestures," *Proc. of 18th International Conf. on Pattern Recognition*, pp. 560-563, August 2006.

[6] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," *Proc. of IEEE International Conf. on Computer Vision*, pp. 1036-1043, November 2011.

[7] P. Tormene, T. Giorgino, S. Quaglini, and M. Stefanelli, "Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation," *Artificial Intelligence in Medicine*, vol. 45, no. 1, pp. 11-34, January 2009.

[8] V. Bloom, V. Argyriou, and D. Makris, "Linear latent low dimensional space for online early action recognition and prediction," *Pattern Recognition*, vol. 72, pp. 532-547, December 2017.

[9] M. Weber, M. Liwicki, D. Stricker, C. Scholzel, and S. Uchida, "LSTM-Based Early Recognition of Motion Patterns," *Proc. of 22nd International Conf. on Pattern Recognition*, pp. 3552-3557, August 2014.

[10] W. Li and M. Fritz, "Recognition of ongoing complex activities by sequence prediction over a hierarchical label space," *Proc. of IEEE Winter Conf. on Applications of Computer Vision*, pp. 1-9, March 2016.

[11] E. Vats, C. K. Lim, and C. S. Chan, "Early human actions detection using BK sub-triangle product," *Proc. of IEEE International Conf. on Fuzzy Systems*, pp. 1-8, August 2015.

[12] Y. L. Ji, Y. Yang, X. Xu, and H. T. Shen, "One-shot learning based pattern transition map for action early recognition," *Signal Processing*, vol. 143, pp. 364-370, February 2018.

[13] Microsoft, "Kinect - Windows app development," http://developer.microsoft.com/en-us/windows/kinect.

[14] T. H. Zhu, Q. F. Zhao, W. B. Wan, and Z. Y. Xia, "Robust regression-based motion perception for online imitation on humanoid robot," *International Journal of Social Robotics*, vol. 9, no. 5, pp. 705-725, November 2017.

[15] A. López-Méndez, M. Alcoverro, M. Pardàs, and J. R. Casas, "Real-time upper body tracking with online initialization using a range sensor," *Proc. of IEEE International Conf. on Computer Vision Workshops*, pp. 391-398, November 2011.

[16] Y. Xiao, Z. J. Zhang, A. Beck, J. S. Yuan, and D. Thalmann, "Human-robot interaction by understanding upper body gestures," *Presence: Teleoperators and Virtual Environments*, vol. 23, no. 2, pp. 133-154, August 2014.

[17] Aldebaran, "H25 - Joints," http://doc.aldebaran.com/2-1/family/nao_h25/joints_h25.html.

[18] P. Indyk, and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, pp. 604-613, May 1998.

[19] Y. Ko, A. Shimada, H. Nagahara, and R. I. Taniguchi, "Hash-based early recognition of gesture patterns," *Artificial Life and Robotics*, vol. 17, no. 3-4, pp. 476-482, February 2013.

[20] M. Slaney, and M. Casey, "Locality-sensitive hashing for finding nearest neighbors," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 128-131, March 2008.

[21] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, pp. 253-262, June 2004.

[22] S. Fothergill, H. Mentis, P. Kohli, and S. Nowozin, "Instructing people for training gestural interactive systems," *Proceedings of the SIGCHI Conf. on Human Factors in Computing Systems*, pp. 1737-1746, May 2012.

[23] D. Huang, S. Yao, Y. Wang, and F. De La Torre, "Sequential max-margin event detectors," *Proc. of European Conf. on Computer Vision*, pp. 410-424, September 2014.

[24] Y. L. Ji, H. Cheng, Y. L. Zheng, and H. X. Li, "Learning contrastive feature distribution model for interaction recognition," *Journal of Visual Communication and Image Representation*, vol. 33, pp. 340-349, November 2015.

[25] Microsoft Research Cambridge, "Kinect Gesture Data Set - Microsoft Research," http://research.microsoft.com/en-us/downloads/4e1c9174-9b94-4c4d-bc5e-0a9c929869a7/.

[26] X. B. Jiang, F. Zhong, Q. S. Peng, and X. Y. Qin, "Online robust action recognition based on a hierarchical model," *The Visual Computer*, vol. 30, no. 9, pp. 1021-1033, September 2014.

[27] D. Huang, S. T. Yao, Y. Wang, and F. De La Torre. "Action table of MAD database," http://humansensing.cs.cmu.edu/mad/data/action_table.txt.

[28] M. Hoai and F. De La Torre, "Max-margin early event detectors," *International Journal of Computer Vision*, vol. 107, no. 2, pp. 191-202, April 2014.

[29] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43-49, February 1978.

[30] N. Zhang, W. L. Zheng, W. Liu, and B. L. Lu, "Continuous vigilance estimation using LSTM neural networks," *Proc. of International Conf. on Neural Information Processing*, pp. 530-537, October 2016.

Tehao Zhu received the B.S. degree in automation from the Northwest Polytechnical University, Xi'an, China, in 2009, and the M.S. degree in pattern recognition and intelligent system from University of Science and Technology of China, Hefei, China, in 2012. He is currently a Ph.D. Candidate at Shanghai Jiao Tong University, Shanghai, China. His current research interests include human-robot interaction, machine learning, and image processing.

Yue Zhou received his B.S. degree in electronics engineering technology, an M.S. degree in mechatronic engineering, and a Ph.D. degree in signal and information processing from Northwest Polytechnical University, Xi'an, China, in 1991, 1997, and 2000, respectively. He was a postdoctoral research fellow in Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, from 2000 to 2002. He is currently an Associate Professor at School of Electronic Information and Electric Engineering, Shanghai Jiao Tong University, China. His research interests include visual-based object detection, object tracking, and scenario understanding.

Zeyang Xia received the B.S. degree in mechanical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2002, and the Ph.D. degree in mechanical engineering from Tsinghua University, Beijing, China, in 2008. He is currently a Professor at Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. His research interests include humanoid robotics, medical robotics and biomechanics.

Jiaqi Dong received the B.S. degree in automation from Shanghai Jiao Tong University, Shanghai, China, in 2014. She is currently a Ph.D. Candidate at Shanghai Jiao Tong University, Shanghai, China. Her current research interests include human-robot interaction and pattern recognition.

Qunfei Zhao received the B.S.E.E. degree from Xi'an Jiao Tong University, Xi'an, China, in 1982, and the Sc.D. degree in system science from Tokyo Institute of Technology, Tokyo, Japan, in 1988. He is currently a Professor at School of Electronic Information and Electric Engineering, Shanghai Jiao Tong University, China. His research interests include robotics, machine vision, and optimal control of complex mechatronic systems.