J. Vis. Commun. Image R. 24 (2013) 1458–1468

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Depth sensor assisted real-time gesture recognition for interactive presentation

Hanjie Wang [a], Jingjing Fu [b,*], Yan Lu [b], Xilin Chen [a], Shipeng Li [b]

[a] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
[b] Microsoft Research Asia, Beijing 100080, China

ARTICLE INFO

Article history:
Received 11 April 2013
Accepted 21 October 2013
Available online 1 November 2013

Keywords:
Gesture recognition
Real-time interaction
Motion History Image
Depth sensor
Background subtraction
Hand tracking
Feature pooling
Discriminative model

ABSTRACT

In this paper, we present a gesture recognition approach to enable real-time manipulating projection content through detecting and recognizing speakers gestures from the depth maps captured by a depth sensor. To overcome the limited measurement accuracy of depth sensor, a robust background subtraction method is proposed for effective human body segmentation and a distance map is adopted to detect human hands. Potential Active Region (PAR) is utilized to ensure the generation of valid hand trajectory to avoid extra computational cost on the recognition of meaningless gestures and three different detection modes are designed for complexity reduction. The detected hand trajectory is temporally segmented into a series of movements, which are represented as Motion History Images. A set-based soft discriminative model is proposed to recognize gestures from these movements. The proposed approach is evaluated on our dataset and performs efficiently and robustly with 90% accuracy.

1. Introduction

Recently, gesture recognition has been attracting a great deal of attention as a natural human computer interface, since it allows users to control or manipulate devices in a more natural manner through intentional physical movements of figures, hands, arms, face, head, or body. So far, numerous studies [10,16] have been conducted on gesture recognition for human computer interaction, especially for hand and arm gesture recognition. Based on these technologies, a number of recognition applications are developed, including sign language recognition, game controlling, navigating in virtual environment, etc. In this paper, we present an efficient arm gesture recognition algorithm that enables natural interaction between speaker and presentation contents. During presentation, speakers often stand in front of the projection screen at a distance from the machine with projection contents. It is more natural for speakers to remotely control the page flipping, scrolling and clicking by arm gestures. Considering the potential light influences caused by projector on the color images, depth maps captured during presentation are employed for gesture recognition in our work. Fig. 1 shows the framework of proposed approach. In the framework, human body is segmented from noisy depth maps, and then Potential Active Regions (PARs) are derived from head position for meaningful gesture detection. Once the hand is observed in the PAR, its trajectory will be recorded and decomposed into a series of movements. (See green box in dash line in Fig. 1). These movements are represented as Motion History Images (MHIs) and compose a labeled gesture by utilizing proposed set-based soft discriminative model.

As depth data is noisy, a background subtraction technique is used to segment human body from background. The location and size of human body are determined by searching a proper bounding box in the generated human body's depth map. Given that human body may be incomplete in the bounding box because of occlusion or the limited view angle of depth sensor, head detection is applied to estimating the size of the complete human body. However, normal head detection method using face is impossible in presentation since speakers may turn their faces to the side (not facing the sensor) and skin color is also changed due to the strong light from the projector. As a result, we detect the head position by detecting the physical width variation in body's bounding box.

Referring the size and location of human body, PARs of human arms (see the boxes beside body in Fig. 1) are adaptively determined. Intuitively, PARs are the most discriminative regions. Therefore, the arm is only detected when reaching in PARs, which is shown in the framework of our system in Fig. 1. If no arm is detected in PARs in this frame, it is unnecessary to perform recognition step or even unnecessary to detect arm in the next few frames, which vastly reduces the computational complexity. On

* Corresponding author. Address: Tower 2, No. 5 Danling Street, Haidian District, Beijing, 100080, P.R. China.
E-mail address: jifu@microsoft.com (J. Fu).

1047-3203/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jvcir.2013.10.004

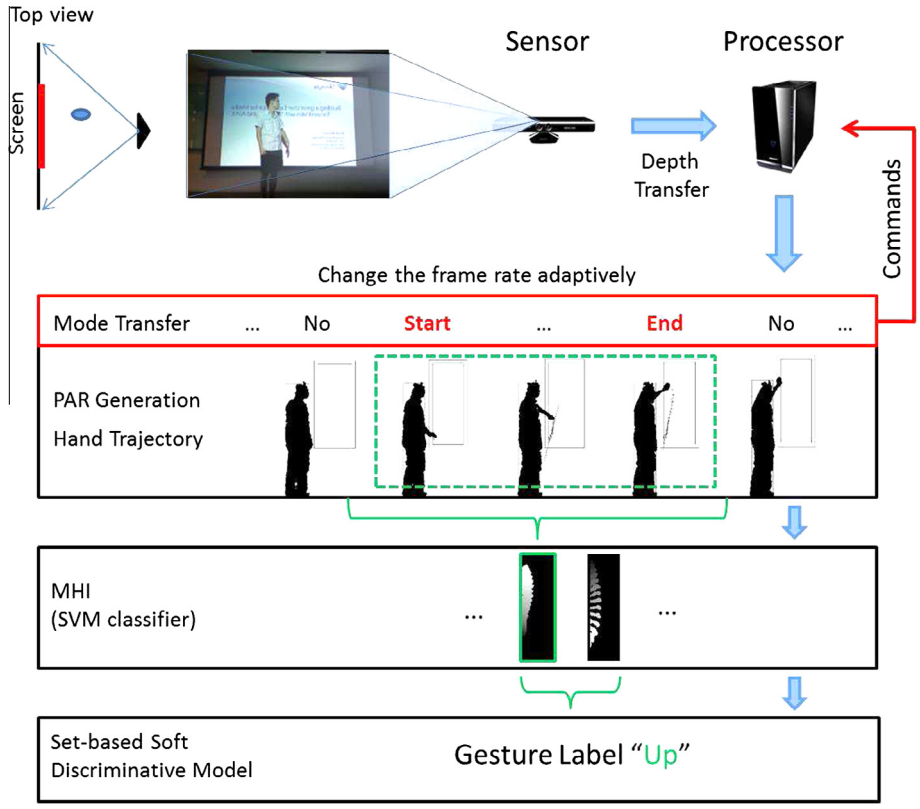

Fig. 1. Framework of proposed approach.

the contrary, once arm is observed in a PAR, the trajectory of hand will be recorded and decomposed into a series of movements. In our work, the hand trajectories are decomposed by interest points, which are searched online using a self-adaptive model. Therefore, the video containing continues gestures is automatically segmented by the proposed approach. It is one of the reasons that our approach can be used in real-time interaction system, not only be tested in some specific datasets.

These segmented movements are classified by Support Vector Machine (SVM) and compose an action, which consists of one or several movements and is then labeled and understood as a specific gesture. Considering that one misclassified movement in the action may influence the result of gesture recognition, a set-based soft discriminative model is originally proposed to correct a misclassified movement and designate a most likely gesture label to the set, making our approach robust. Experimental results show

Fig. 2. Five gestures ("click" is shown in depth since it is a gesture vertical to X–Y plane).

that this model performs much better than the traditional discriminative model.

Our contributions mainly lie in three folds. First, the noise and occlusion problems caused by depth sensors are solved by efficient preprocessing of background subtraction. Second, with PAR, the computational complexity is dramatically reduced by selecting active frames to implement further gesture recognition. Third, a set-based soft discriminative model is originally proposed to compose movement labels into a meaningful gesture. To test our method, we capture a dataset including five gestures: "up", "down", "go", "back" and "click" (see Fig. 2).

The rest of this paper is organized as follows. Related work is introduced in Section 2. Section 3 and 4 presents the preprocessing and the proposed gesture recognition approach. The experimental results in Section 5 demonstrate the efficiency of the proposed approach. Section 6 concludes the paper and gives the future directions.

2. Related work

In the last decades, most of recognition algorithms are designed for the color images captured by monocular camera sensors. One challenge of the color image based recognition is how to efficiently segment the object from the background. In order to obtain foreground silhouettes of objects, most of the recognition algorithms are restricted to pure and static backgrounds. For example, public dataset KTH human motion dataset [12] and Weizmann human action dataset [1] both record human actions under relatively static backgrounds. The rapid development of depth sensors open up the possibility of dealing with cluttered background by providing depth information. Even though, depth sensors like Kinect [14], Time of Flight camera [5] or stereo camera [15] still present two challenges: noise and occlusion. Due to the limited measurement accuracy of the depth sensor, noise is produced and decreases the quality of background subtraction. Occlusion occurs when there is an object (e.g., desk) in front of human body. It will ruin body detection because part of the human body is blocked. This is also a major problem while adopting monocular camera. What is worse, the Kinect sensor regards black objects such as black trousers or black hair as occlusions due to its generation principle. To address the issue of occlusion, some approaches employ the location of human face to indicate the location of human body. Face detection is usually implemented as the first step to detect human body. For example, Wang et al. [19] locate the face before obtaining skin model from face and use it to detect human hand. As stated before, they cannot work well in presentation due to the variant directions of face and abnormal skin color, which is influenced by projector light.

When equipped with depth sensor, many researchers make effort to compute 3D joint positions of human skeleton. Shotton et al. [14] provide a rather powerful human motion capturing technique. There are also many works in the field of action recognition directly using human skeleton. For example, Jiang et al. [18] track 20 joint positions by the skeleton tracker proposed by Shotton et al. [14] and use a local occupancy pattern to represent the interaction. Sadeghipour et al. [11] also use the 3D joint positions of human skeleton for gesture-based object recognition. Both of them adopt the Kinect as the depth sensor. When referred to the other depth sensors, robust and fast method to get human skeleton has not been presented yet to the best of our knowledge. That means, methods in [18,11] may not work so well without Kinect sensor. Besides, skeleton tracking by Kinect SDK performs best on the front view of human body with little occlusion. In the paper [17], Wang and his colleges has already found that the skeleton tracking method proposed by [14] may be inaccurate or even fail when serious occlusion occurs. They developed random occupancy pattern features to ensure the robustness to noise. By using the silhouettes, 2D features like MHI can also be generated for the same purpose. One step further, Depth Motion Maps (DMMS) is proposed by Yang et al. [22]. DMM stacks motion energy of all the depth maps projected onto three orthogonal Cartesian planes, make a fully use of the 3D information of depth sensor.

As PAR provides a spatial region that probably contains arm motion, the generation of MHI still asks for a temporal region to indicate the start and end of a gesture in a video sequence. In the most published 3D datasets, videos are manually segmented into sequences that contain an instance with a known set of action labels. For example, Sadeghipour et al. [11] capture the 3D Iconic Gesture dataset and segment the video by the moment when subject retracting their hands back to the rest position, so does NATOPS Aircraft Handling Signals Database captured by Song et al. [15].

As we know, representation of suitable feature and modeling of dynamic patterns are two important issues for recognition. A detailed taxonomy was summarized by Bobick [2] in an early survey. In other works, 3D low-level features are deeply studied in recent years. Most of them are extended from normal 2D features such as [9] (3D Harris corner detector), [21] (3D SURF descriptor), [8] (3D HOG descriptor) and [13] (3D SIFT descriptor). However, these local features are not discriminative ones in textureless depth maps. To compose low-level movements into a gesture in the proposed framework, generative model and discriminative model are generally used as two typical temporal state-space models. Generative model learns to model each class individually and always assume that the observations in time are independent, e.g., Feng and Perona [4] and Weinland et al. [20]. Discriminative model is trained to discriminate between action classes and model a conditional distribution over action labels given the observation. Jordan et al. [7] compare discriminative and generative learning as typified by logistic regression and naive Bayes. Our gesture recognition model is categorized to the discriminative model.

3. Preprocessing

This section gives a description of the two tasks in sequence of preprocessing. The first is the background subtraction from the noisy depth maps. The other is the PARs' derivation from head position for meaningful gesture detection. In addition, with hand detecting in the PARs, the reduction of the complexity of the system is also analyzed. "Detection mode transfer" is introduced to optimize our system.

3.1. Background subtraction

In presentation, touching the screen while performing gestures is a natural way to control the projection contents. However, measurement accuracy of the depth sensor is limited. As the result, even if the depth of background has been captured, segmenting arms from screen is impossible by background subtraction techniques. We define the scenes without speaker as backgrounds, and they are captured with a stationary Kinect sensor. Ideally, depth values of each point should be constants for all frames when the scene is fixed. Unfortunately, the environmental influence and systematic noise introduce random fluctuation on depth value. As a result, the depth value distribution of a pixel approaches to a Gaussian distribution, in which the mean is supposed to be a specific depth $d_{(x,y)}$ and its variance is supposed to be $\sigma^2_{(x,y)}$. Thus, we defined $D_{(x,y)} \sim N(d_{(x,y)}, \sigma^2_{(x,y)})$ as a distribution to fit the depths of pixel at the ordinate (x, y) in the background. Once a human body occurs in the frame, pixels' depth values change a lot where the

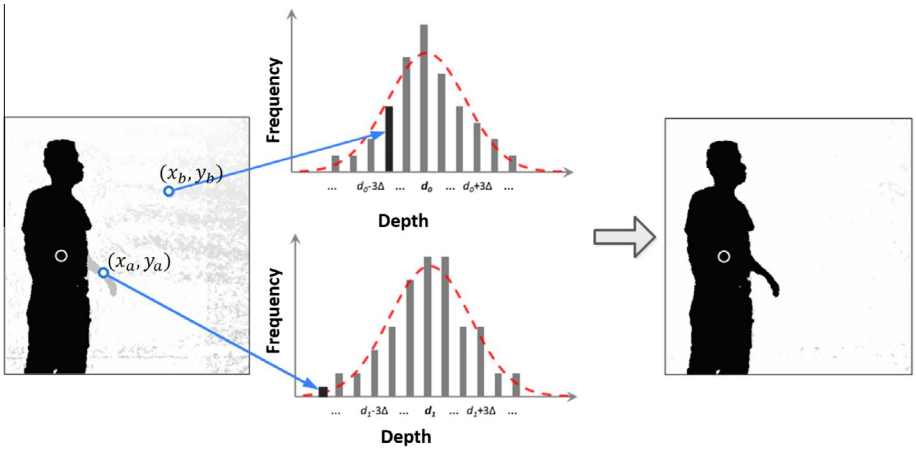

Fig. 3. Results of improved background subtraction method.

human body occurs. Because the new depth value of the pixel is far from the mean of Gaussian distribution, it has a small probability to belong to the background.

As illustrated in Fig. 3, the two histograms show the statistical models of two pixels' depths from frames without speaker. $D_{(x_a,y_a)}$ and $D_{(x_b,y_b)}$ fit the two statistical result and are shown as red dash curves in the histograms. The left image shows the segmenting result just using a threshold on depth. Noticing that human's left arm is missing since it is close to the background (the screen) and its distance to the Kinect is relatively far, making its pixels' depths below the threshold. The right image shows the result of our method. The depth of pixel $p(x_a,y_a)$ in the arm has small frequency on its corresponding Gaussian distribution $D_{(x_a,y_a)}$ and is labeled as human body rather than background while depth of pixel $p(x_b,y_b)$ has large frequency on $D_{(x_b,y_b)}$ and is labeled as background.

Though a threshold on frequency is still needed in our method, it is more robust than directly assigning a threshold on depth. That is because of the statistical model constructed in the former place. The improvement is obvious in Fig. 3.

3.2. PAR generation

The size and location of the whole human body is necessary for PAR generation. Therefore, a bounding box containing the whole body is constructed. The segmented depth map is scanned along vertical lines from left to right while recording the proportion of body pixels in the line. The location of the bounding box's left border is determined when the first time the proportion reaches a threshold. The locations of other three borders are determined in the similar way.

Notice that human body in such bounding box may be incomplete because of the occlusion or the limited view angle of sensor. Size of head is thus employed to estimate the size of the whole body according to the normal proportion of human figure. As we know, the physical width of a human body varies with height. By counting the pixel number of human body along horizontal direction in the body bounding box, a curve is drawn to show the proportion of human pixel (black) in each horizontal line, as illustrated in Fig. 4(a). We make the assumption that neck has the minimum width when compared with other parts of human body. Thus, human head can be segmented according to the curve in Fig. 4(a) by a self-adaptive threshold. The goal of head segmentation is to estimate the location and size of PAR. In a usual presentation, it is more natural for speakers to control the page flipping, scrolling and clicking by arm gestures. That makes region near the speakers' shoulders more important for detecting the meaningful gestures. Combined with human size, referring from size of human head, a Potential Active Region can be estimated. Two PARs are then constructed for left and right arms. Since PARs cover the most likely arm movement region, they can serve as the constraint for hand tracking and adaptive detection mode transfer. The detection mode transfer will be introduced in details in the Section 3.3.

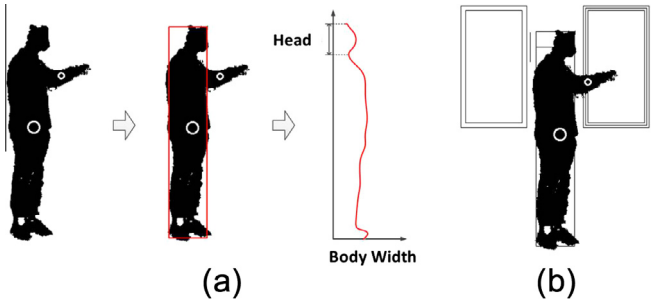

Fig. 4. (a) Method to obtain the size and location of head. (b) PARs for two arms.

3.3. Detection mode transfer

During presentation, most of speakers' arm movements are meaningless to presentation content control. Therefore, when hand trajectories are far away from screen or not in the PARs, it is unnecessary to record the hand trajectory or recognize the gesture. As an action is continuously in the adjacent depth frames, once the hand enters the PARs in the current depth frame, it is very likely that the hand moves within the PARs to perform meaningful actions in the sequential frames. The detection frequency can be reduced if the hand position is far from the PAR or its depth out of defined depth range. To adjust the detection frequency of arm movements, three hand detection modes are defined as "inactive", "semi-active" and "active", and each mode has different detection intervals k. In an inactive mode, hands will be re-detected after K frames, that is, $k = K - 1$. In the active mode, hands will be re-detected in the next frame, i.e., $k = 0$. If the detection mode keeps semi-active, the interval k is linearly increased from 1 to $(K - 1)$ over time.

When a hand is detected in PARs and is close enough to screen, the mode is immediately switched to "active" from "inactive" or "semi-active". Otherwise, the mode is switched to "semi-active" and gradually becomes "inactive" if hand is not detected in PARs for a long time. It should be noted that "active" mode is not directly

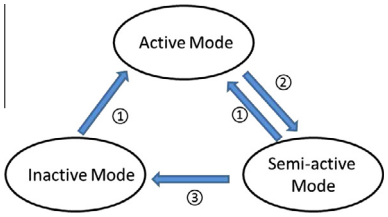

Fig. 5. (1) If a hand is detected in the extension of PARs, detection mode will be transferred to active mode but the hand's trajectory will not be recorded. If hand is detected in PARs, both mode transfer and trajectory recording will be executed. (2) Hand leaves the extension of PARs. (3) Hand leaves the extension of PARs for a long time.

changed to "inactive" mode. Because the detecting module may generate a false reject error, i.e., missing a hand in one frame. In "semi-active" mode, hand can be re-detected after fewer frames than in "inactive" mode. On the contrary, switching mode from "inactive" to "active" may require a relatively long time, during which hand may have already entered the PARs in those skipped frames. To avoid this phenomenon, an extension of PAR is generated for pre-detecting hand to ensure the completeness of trajectory. Besides, the hand trajectory excludes the hand detected in extension of PAR. In Fig. 4(b), around the PAR, there are two bounding boxes, which are defined as inside and outside. Region between the inside and the outside bounding box is the extension of PAR. The transfer between the three modes is illustrated in Fig. 5.

4. Gesture recognition

After depth video preprocessing, hand trajectory in PAR is recorded and decomposed into several movements, where MHI feature will be extracted. In terms of the trained multi-class SVM model, MHI features are classified and assigned to movement labels. Then, a soft discriminative model is originally proposed to compose movement labels into one meaningful gesture. Specifically, the 5 classes gestures (i.e., "up", "down", "go", "back" and "click") are composed of 12 classes elementary movements. Each gesture is a sequence including one or several elementary movements. These classes are learned from the training data.

4.1. Hand trajectory decomposition

The skeleton tracking in Kinect SDK provides a powerful tool for users to develop recognition related applications. Therefore, in many gesture recognition works [18,11], hand trajectory is derived directly from the skeleton extracted by Kinect SDK. The skeleton tracking results generated by Kinect SDK is robust for the depth sequences with little occlusion. However, in most cases of presentation scenario, speakers tend to touch the screen with their lateral views facing the Kinect and the skeleton tracking method may produce inaccurate results or even fail on the subjects' lateral views, where serious occlusion occurs (see Fig. 6(a)). In the left image of Fig. 6(a), the arm skeleton fails to match the silhouette, since part of the arm is occluded by the body. In the right image, the arm skeleton is out of the silhouette when the arm is completely occluded. In our work, the hand point (i.e., the endpoint of arm) is the most important skeleton point used for gesture recognition. Once it occurs in PARs, a gesture begins to be recorded for recognition. Otherwise, the hand trajectory will not be recorded. Inaccurate tracking of hand point will dramatically decrease the recognition rate. Therefore, we proposed an effective endpoint extraction method based on distance map to solve the problem.

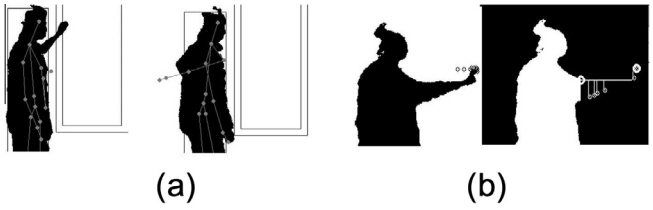

Fig. 6. (a) Failure cases of skeleton tracking. (b) Distance map using chamfer distance.

A chamfer distance map is derived when a hand enters the PAR. As the farthest endpoint from the edge of PAR, the hand is detected and tracked in PAR before leaving (see Fig. 6(b)). To recognize gestures from the complex trajectories, a self-adaptive approach is proposed to decompose the trajectories into movements, classify the movements and then compose a meaningful gesture. However, it is difficult to recognize gestures directly from the hand trajectories since our dataset has the following two characteristics. First, the dataset has small inter-class variances. The gesture "up" is very similar to "down" while the trajectories of "go" and "back" are similar if not considering the order of frames. Second, it also has large intra-class variances. There are variant ways to act the same gesture by different subjects. Even the same gesture performed by the same subject can be different each time. Therefore, we represent the segmented movements as MHIs. Since MHI only requires start and end of the trajectory, the interest points become more important than other points in the trajectories. See from Fig. 7, the subject successively performs four actions in one video. Among the complex trajectory, interest points are required to help segmenting it into movements.

As we know, interest points are the sudden changes of trajectories in a video sequence. Based on that common sense, Method of Least Squares (MLS) is used to detect the interest points on the trajectories. The MLS assumes that the best-fit curve of a given type is the curve that has the minimal sum of the deviations squared (least square error) from a given set of data. The type of straight line $y = ax + b$ is employed to approximate a given set of points. A new point is determined as an interest point when deviation $d_{(n+1)}$ of its coordinate $(x_{(n+1)}, y_{(n+1)})$ is larger than a threshold d_{th}, which is updated with the deviations by equation $d_{th} = \alpha(\sum_{i=1}^{n} d_i)/n$. So long as two interest points are found, the trajectory between the two points is segmented, indicating that a movement is detected (see Fig. 7).

4.2. Feature extraction and classification

We adopt coding-pooling-classification pipeline to extract global features from the silhouettes. A classifier is trained and tested by SVM using MHI features. Though the sizes of PARs are variant with time, the normalizations of PARs are potentially conducted in the coding and pooling steps. Hence, all the feature vectors have the same lengths for movement classification.

4.2.1. Coding

The silhouettes of human arm belonging to the same movement in a PAR are accumulated to generate the MHI, which is originally proposed by Bobick et al. [3]. In a MHI, pixel intensity records the temporal history of motion at each position. See Eq. (1) as follows:

$$MHI(x,y,t) = \begin{cases} 0 & t = 0 \\ MHI(x,y,t-1) + 1 & otherwise \end{cases}, \quad (1)$$

where (x, y) is the ordinate of a pixel and t is its frame index in its corresponding trajectory segmentation. It will be normalized to $[0, 255]$ before being used as a feature, and denoted as $\overline{MHI}(x, y)$.

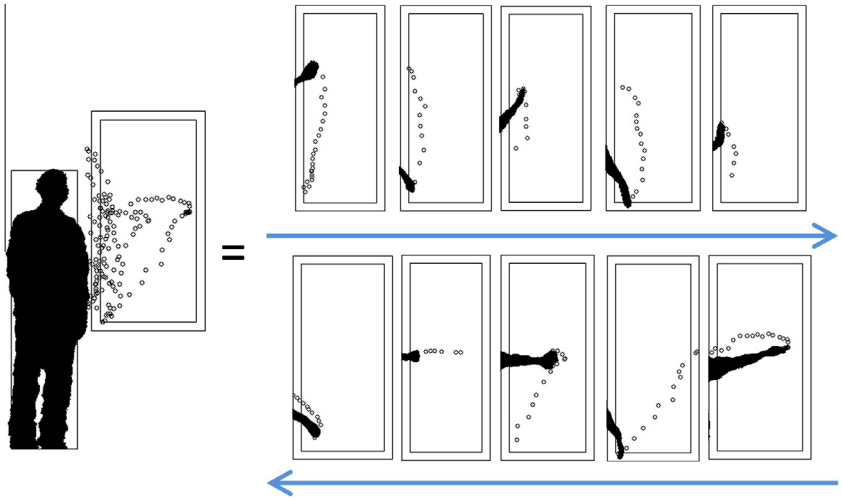

Fig. 7. Decomposing the movement trajectories. The start and end points in each segmentation are interest points. 10 Movements are detected in this example.

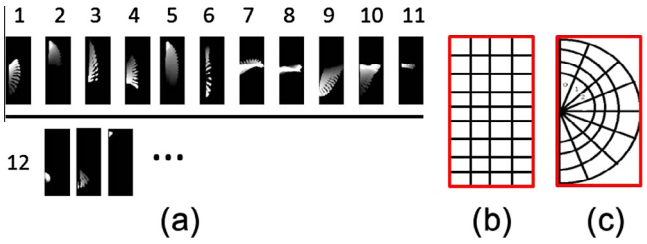

Fig. 8. (a) Samples of 12 classes of movements. Class 12 includes meaningless movements. Three of them are listed (b) Uniform rectangle spatial region. (c) Uniform semi-circle spatial region.

The MHIs of decomposed movements are shown in Fig. 8(a) and samples of gestures consist of a series of MHIs will be shown in section Experiment.

4.2.2. Pooling

Designing a proper spatial region for pooling has a significant impact on a feature's capacity to distinguish [6]. Since arms rotate along shoulder joints, the pixel intensity distributes in a semi-circle manner. Features pooled from the uniform spatial region like that in Fig. 8(b) are improper for further classification. For more effective representation, we choose the spatial regions shown in Fig. 8(c). The blocks in the semi-circle region are represented as $R_1, R_2, \ldots, R_N$. $\overline{MHI(x,y)}$ is the intensity of pixels at the position (x,y) in the MHI and the final global feature is denoted as $V = \{v_1, v_2, \ldots, v_N\}$. v_i is computed as follows:

$$v_i = \frac{1}{|R_i|} \sum_{(x,y)\in R_i} \overline{MHI(x,y)}, \quad i = 1,2,\ldots,N, \tag{2}$$

where $|R_i|$ represents the pixel number in each block.

4.2.3. Classification

SVM is applied for classifying movements. The standard SVM divides two classes with a clear gap that is as wide as possible. The classifier predicts new examples for a category based on which side of the gap they fall on. 12 classes of movements are designed and shown in Fig. 8(a). The formal 11 classes are meaningful movements while class 12 is meaningless movements. A multi-class SVM model is trained when given the manually labeled training data. In practice, a one against one strategy is applied to multi-class SVM and the output is a distribution on all the labels for each movement.

4.3. Set-based soft discriminative model

As has been pointed out, the intra-class variances of our dataset are large since one gesture may have multiple rules to be composed of. In the discriminative model, the rules and their corresponding probabilities can be learned from the training data. Movement sets, both meaningful and meaningless, are considered as rules in our work. Suppose x is the movements set (observation) and y is the gesture label (hidden state). Normal discriminative classifiers directly model the posterior $p(y|x)$. Suppose each movement in the set has a probability P_{w_mov} to be wrongly classified. Therefore, the gesture consisting of movements in such a set has a probability of $P_{w\_ges} = 1 - (1 - P_{w\_mov})^{n_s}$ to be wrongly recognized, where n_s is the number of movements in one set. Though P_{w_mov} may be small, P_{w_ges} can be quite large so that it can hardly be tolerated in real-time applications.

In our work, P_{w_mov} is relatively small by the proper feature and classifier. Instead of keep decreasing the P_{w_mov}, we propose a soft discriminative model to directly decrease the P_{w_ges}. Let M_i denote the label of movement and G_j denote the label of gesture. A set of M_i is represented as MS (see Eq. (3)), also known as assembly rule.

$$MS_i = \langle M_{i1}, M_{i2}, \ldots, M_{in_i} \rangle \quad i = 1,2,3\ldots L. \tag{3}$$

Each detected movement m has a distribution on all labels via multi-label SVM. In the distribution, $p_m^{M_i}$ is denoted as the probability of movement m classified to label M_i. The gesture probability $p(G_j)$ is computed as follows:

$$p(G_j) = \max_i \sqrt[n_i]{\left[p(G_j|MS_i) \times p(MS_i)\right]}, \tag{4}$$

where

$$p(MS_i) = \prod_{k=1}^{n_i} p_{m_k}^{M_{ik}}, \tag{5}$$

and n_i-th root is used for normalizing since the number of the movements in one set is variant. The detailed implementation is described by Algorithm 1.

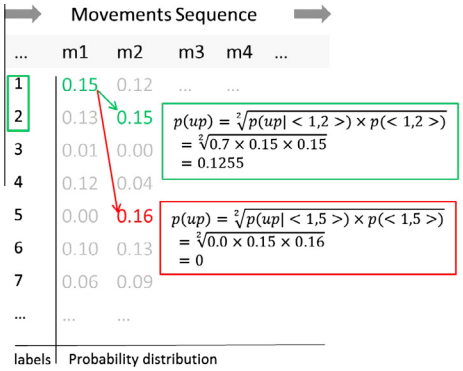

Fig. 9. Suppose the priori $p(up|\langle 1,2\rangle) = 0.7$ and $p(up|\langle 1,5\rangle) = 0$. See from the table, this movement set has the largest joint probability on $\langle 1,5\rangle$ after classification and has no chance to be labeled as "up". However, in soft discriminative model, this set still has the probability of 0.1255 to be labeled as "up" when $\langle 1,2\rangle$ is chosen.

Algorithm 1. Gesture recognition from movement sets.

1. Train L rules $MS_i = \{M_{i1}, M_{i2} \ldots M_{in_i}\}$ where $i = 1, 2, \ldots, L$;
2. **repeat**
3. Add a movement m_{n+1} to a queue of movements;
4. **for** *rule index* $i = 1$ *to* L **do**
5. Compute the probability $p(MS_i)$ for all the sets with equal length n_i in the queue of movements;
6. Multiply $p(MS_i)$ by posterior $p(G_j|MS_i)$;
7. Normalize $p(G_j)$;
8. **end**
9. Choose the max $p(G_j)$;
10. **if** $p(G_j) >$ *threshold* **then**
11. Detect an action and label it as G_j;
12. Delete the movements from queue;
13. **else**
14. No action is detected;
15. **end**
16. **until** *no movement is detected any more*;

Actually, Eq. (4) replaces traditional x with a set MS, fully utilizing the distribution of classification output p_m^M. Each rule MS_i is evaluated on the subsets of a given movement sequence over time. The MS can be regarded as a sparse joint distribution of the movements in the sequence. It provides a soft observation of the discriminative model. To explain it clearly, Fig. 9 gives a simple example of this model.

Two advantages of set-based discriminative model are listed as follows. First, it is able to correct wrongly classified movements and output the right recognition result by thoroughly take advantage of each movement's distribution on labels. Second, this model is completely fit the online processing since the movement set MS implicitly give the start and finish state, i.e., the first and last movement elements of the set. A movement stream will be automatically segmented by these sets.

5. Experiment

5.1. Dataset and correct rate

To the best of our knowledge, no suitable database is accessible with the scenario of presentation and we collect a new dataset of interactive presentation gestures that contain five typical presentation gesture classes: "up", "down", "go", "back" and "click" intuitively corresponding to the "up", "down", "left", "right" and "enter" in the keyboard. Part of the rules to compose gestures are shown in Fig. 10, in which movement sets are represented in the form of MHI. The number of movements in one action and the appearance of the same class of MHI are variant, since different subjects act in a quite different way. It makes the corresponding dataset a challenge one, with large intra-class variations and small inter-class variations.

In our dataset, each gesture was performed by three subjects for five times under two different lighting conditions: projection light and normal light. Unlike pre-segmented dataset, a long video record the five classes of gestures successively by one subject each time. Our method can locate the five gestures automatically in the video without pre-segmentation. Each subject performs the gestures three times at a normal speed (4 s/gesture) and the other two times at a fast speed (1 s/gesture) under each lighting condition. The distance between the Kinect and the subjects is 1.5–2.5 m. The depth maps were captured at about 30 frames per second with a resolution of 640×480. In addition, the size of PAR is normalized to 120×260. Altogether, the dataset has $3(subjets) \times 5(times) \times 5(gestures) \times 2(illuminations) = 150$ gestures, i.e., 30 samples for each class. To test the movement classification and the gesture recognition, 60 gestures $(3(subjets) \times 2(times) \times 5(gestures) \times 2(illuminations))$ serve as training data and the rest serve as test data. The dataset is labeled manually before training and testing. The confusion matrix of 12 classes of movements is shown in Table 1. The confusion matrix of 5 classes of gestures is shown in Table 2. The method achieves a correct classification rate of 95.23% for movements in 5-folds Cross Validation and a correct recognition rate of 90.00% for gestures in the test. Since each gesture consists of more than one frame and most of gestures are meaningless in the long video, we denote frames without meaningful gestures as "None" gesture frames. It is possible that meaningful gestures are recognized in "None" gesture frames. For example, in Table 2, "Go" and "Click" are recognized in some "None" gesture frames.

The cross-subject-test is more desirable in real world application. For each gesture class, 20 samples from two subjects are used for training, the other 10 samples from another subject are used as test data. The experiment is repeated 3 times to test each subject by using the model from the other two. So, each gesture class has to be tested 30 times. The result is recorded in Table 3. The average correct rate is 80.00%. The decreasing of correct rate is due to the limited number of subjects and training data. Only 20 samples from 2 subjects are used to generate the classifier model. With more training data from more subjects, the correct rate of cross-subject-test will be improved.

5.1.1. Complexity reduction

In our work, the gesture detection complexity adapts to the arm's movement, that is, the detection frequency is controlled by detection mode derived from the relationship between PARs and hand position in the previous depth map. In the dataset, 18 videos are captured for testing. When collecting the dataset, one subject successively performs five different gestures in one video, and each video contains 600 frames. Among these depth frames, only partial of them are selected as active frames for gesture detection. As a result, the computing complexity is vastly reduced, and its reduction ratio is proportional to the number of the inactive frame. Fig. 11 (left) shows the active frame ratio over all frames of 18 videos. See from Table 4, the active frame ratio reduces to 46.47% and the correct recognition rate does not decrease much. It means that half of depth frames will not be calculated for detection. When speaker spends more time on presentation instead of interaction, the active frame ratio will be further reduced. This method can be potentially used for wireless transfer of frames.

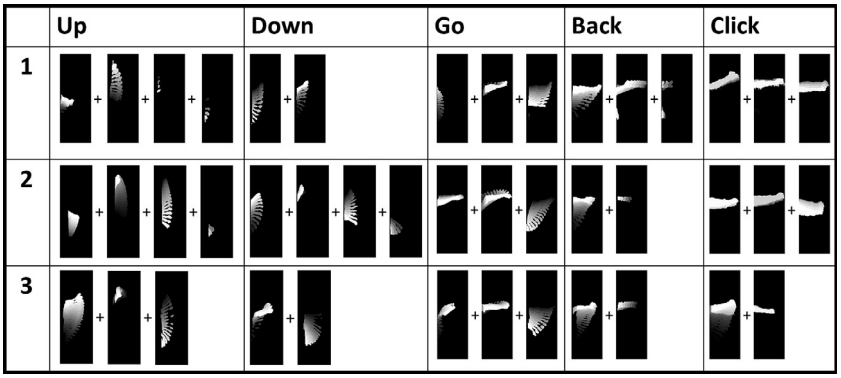

Fig. 10. Samples of five classes of gestures performed by 3 subjects.

Table 1
Confusion matrix of 12 classes of movements in the test. Our method achieves a correct recognition rate of 95.23% in 5-folds Cross Validation.

No.	01	02	03	04	05	06	07	08	09	10	11	12
01	.84		.02				.02			.02	.04	.06
02		.85			.07							.08
03			1									
04				.91		.04			.05			
05					1							
06						.93					.07	
07							.89	.05				.06
08								1				
09									1			
10										1		
11	.06		.06				.05				.83	
12				.01								.99

Table 2
Confusion matrix of 5 classes of gestures (18 samples for each class). "None" means no gesture in those frames. Achieve a correct rate of 90.00% excluding "None".

Gestures	Up	Down	Go	Back	Click	None
Up	16	2				
Down		17				1
Go		1	16			1
Back				14		4
Click					18	
None			2		1	

Table 3
Confusion matrix of cross-subject-test. None means no gesture in those frames. Achieve a correct rate of 80.00% excluding "None".

Gestures	Up	Down	Go	Back	Click	None
Up	19	1				10
Down	2	27		1		
Go		2	21			7
Back				23		7
Click					30	
None	1	2			3	

Table 5 gives the computational complexity in different detection modes. Fig. 11 (right) shows the complexity from one of the 18 videos, which containing four gestures. At the beginning, hand is detected every 10 frames in an "inactive" mode. Once hand is detected in the PARs or the extension of PARs and is close enough to the screen, the detecting mode is switched to "active". After the hand leaves the PARs, the frame grabbing interval for hand detecting increases gradually.

5.1.2. Comparison on features

In the field of gesture recognition, the trajectories of gestures are always represented as a set of points (e.g., sampled positions of the head, hand, and eyes) in a 2-D space before being decomposed. For example, HoGS is a descriptor proposed by Sadeghipour et al. [11]. They combine this feature with SVM to solve the challenging problem of gesture-based object recognition. Though trajectory is obtained by tracking hand in the first place, MHI is the feature we finally used. The reason lies in two folds. First, compared with sensitive point detection, the method to generate MHI is more robust since it simulates original silhouettes. Second, the MHI implicitly represents the history of movement while a trajectory of hand points provides less information on that. To compare the two features, an experiment using trajectory as feature is conducted on our dataset.

As the trajectories have been segmented by MLS, some attributes can be extracted from the curve of segmentations like the method in [11]. Five attributes are used: height, width, length, orientation and center of the curve (see Fig. 12(c)). Combined with SVM, this feature has a low movement correct classification rate and gesture correct recognition rate (see Table 7). As stated above, the main reason is the sensitiveness of points on trajectories. For example, picture (a) and (b) in Fig. 12 are the comparisons between trajectory feature and MHI feature. In (a), the two features have almost the same discriminating power and are both correctly classified in experiment. In (b), since the clothes of the subject used to enter the bounding box and produce some outliers at the left bottom region, trajectory fails to describe this movement because of a wrongly connected straight line while the MHI feature is still correctly classified. That mainly owns to the abundant original information the MHI feature contains.

An extra experiment on directly using skeleton from Kinect SDK is also conducted. Fig. 13 shows the comparison result of using skeleton from Kinect SDK with proposed approach. The average correct rate by using skeleton is 67.78%. Our method of detecting the hand point (i.e., the endpoint of arm) is superior as shown and the reason is explained in Section 4.1. "Click", which is always correctly recognized, is judged by special depth changes when the hand is relatively static in X-Y plane. Therefore, both the methods can locate the hand correctly and performs well on that gesture.

Our dataset is similar to the Microsoft Research (MSR) Action3D dataset. DMM-HOG descriptor is reported to be the state-of-the-art methods on MSR Action3D dataset on year 2012 [22]. They compute HOG features from Depth Motion Maps, which is generated by projecting depth maps on three orthogonal planes and accumulating global activities through entire video sequences. However, our method is designed for specific application and is an online recognition system, MSR Action3D dataset is not thoroughly suitable to test our method. We'd better to test DMM-HOG on our

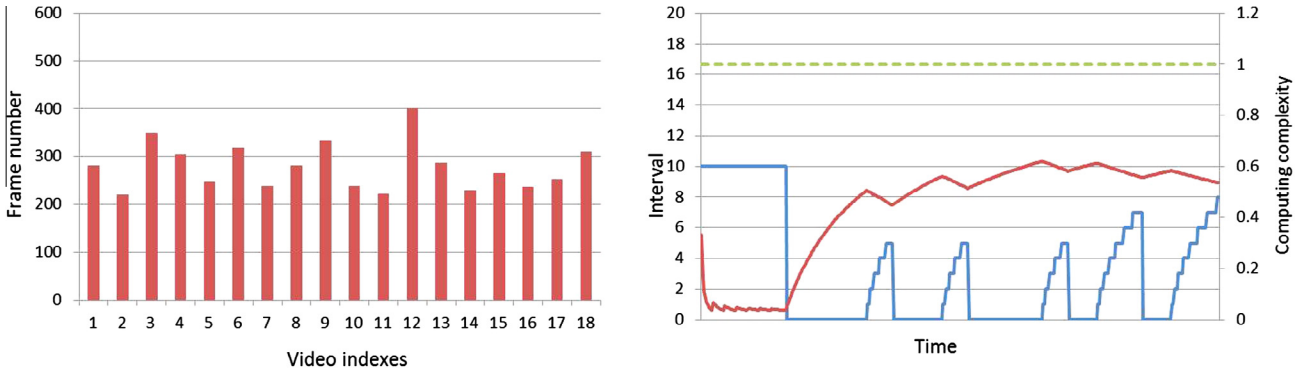

Fig. 11. Left: active frames numbers of all the 18 videos. Right: interval (blue line) and computational complexity curve of one of our samples. Red line represents our computational complexity while green dash line represents normal computational complexity (Best view in color). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
The active frame number and correct recognition rate. (AFP: Active Frame Percentage. CCR: Correct Recognition Rate.)

	AFP	CCR
All frames	100%	90.00%
Avtive frames	46.47%	85.56%

Table 5
Computational complexity in different detection mode. n is the number of pixels in PARs. Chamfer distance maps can be executed in linear (O (n)) time.

Detecting Mode	Processing	Frequency
Inactive mode	Detecting ($O(n)$)	Every 10 frames
Semi-active mode	Detecting ($O(n)$)	Every k frames ($1 < k < 10$)
Active mode	Detecting ($O(n)$) + Recording ($O(n)$) + Feature extraction ($O(n)$)	Every frame

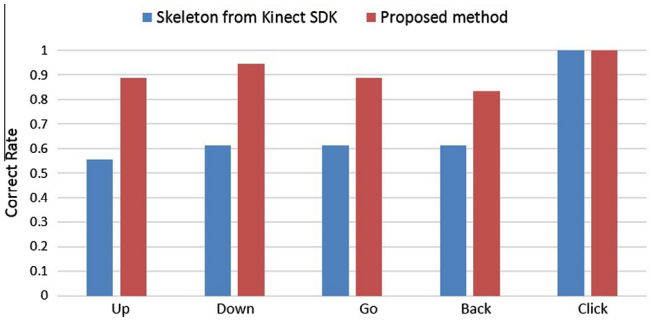

Fig. 13. Correct rate comparison between Kinect SDK and the proposed approach.

dataset. The source code is not accessible and we generate the three orthogonal planes and compute the HOG feature by ourselves, see an example in Fig. 14.

It should be mentioned that method in [22] cannot recognize continues actions while ours is an online system, which is able to recognize continues actions by our set-based soft discriminative model. In order to test DMM-HOG, we manually divide the 30 videos from our dataset into 150 videos, so that each video contains only one action. The dataset is also divided as training data and test data. 60 gestures ($3(subjets) \times 2(times) \times 5(gestures) \times 2(illuminations)$) serve as training data and the rest serve as test data. After extracting the DMM-HOG feature, SVM classifier is used to recognize the actions. Except for extracting HOG features on 3 views, we also carry an experiment on front view only. The result is shown in Table 6.

See from the table, our method is superior to DMM-HOG despite that our method is tested on videos without being segmented. Besides, DMM-HOG using front view is better than using all the 3 views. The reason is supposed to be the similarity of actions in our dataset. See from Fig. 14, the four actions are very similar after accumulating global activities through entire video sequences. While in our method, we solve this problem by segmenting the video sequence into elementary movements. The video is segmented atomically from the entire video by "hand trajectory decomposition". Afterwards, set-based soft discrimina-

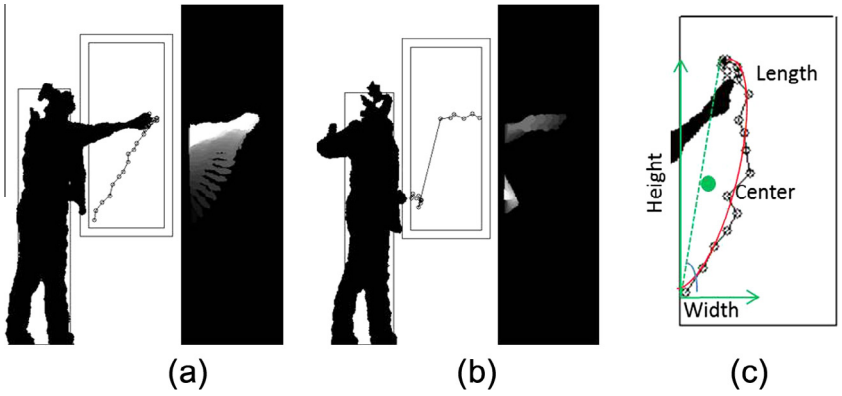

Fig. 12. (a) Both trajectory and MHI are correctly classified. (b) Trajectory has problem with outliers (clothes), the straight line is wrongly connected. (c) The five attributes of trajectory in our experiment.

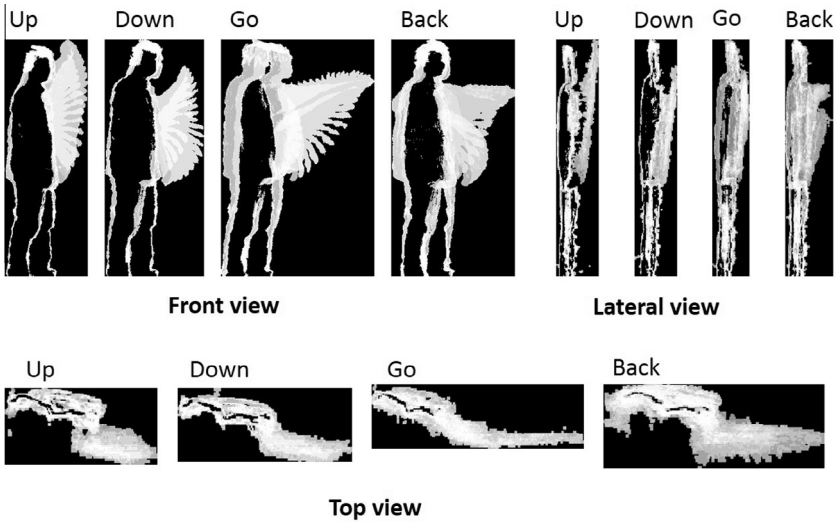

Fig. 14. An example of 3 views generated in experiment.

Table 6
The comparison result. Both methods are tested on our dataset.

Method	Correct Rate
Our method (without manual segmentation on videos)	90.00%
DMM-HOG (3 views: front, top, lateral)	71.11%
DMM-HOG (front view only)	76.67%

tive model composes them into meaningful and discriminative gestures. The appearance of the lateral and top views are more similar between different gestures than that of the front view. It is the major reason that DMM-HOG using 1 view is more discriminative than concatenating HOG features from 3 views.

5.1.3. Set-based soft discriminative model

On the training data set including 60 samples, a posterior $p(G|MS)$ is trained. In traditional discriminative model, MS includes the movement sets that occur at least once in the training data. We know that, though an element of MS_i may be wrongly classified, the MS_i could still be recognized as a meaningful gesture. For example, set $\langle 1,2,3,4\rangle$ means gesture "go". However, a set $\langle 4,2,3,4\rangle$ may also be the gesture "go", just because movement 1 is wrongly classified as movement 4. We define these sets as "new movement sets". These new movement sets have no chance to be changed to the nearest corrected one in the MS in the traditional discriminative model while the soft discriminative model does. That is because soft discriminative model chooses the best one according to the distribution of movements m_i (see Eq. (4)). After using the soft discriminative model, the correct recognition rate is 90.00% (see Table 7).

We also did the experiment comparing a series of movements m_i with trained templates MS in traditional discriminative model.

Table 7
Comparison result. Movement correct classification rate are computed by 5-folds CV. (DM: Discriminative Model; MCCR: Movement Correct Classification Rate; GCRR: Gesture Correct Recognition Rate). The last row in bold shows the best results of MCCR and GCRR when compared with other combinations of features and models.

Feature	Model	MCCR	GCRR
MHI + Semi-circle	Traditional DM	95.23%	76.67%
MHI + Rectangle	Soft DM	79.31%	76.67%
Trajectory	Soft DM	48.32%	62.23%
MHI + Semi-circle	**Soft DM**	**95.23%**	**90.00%**

m_i is represented as a single label rather than a distribution. For observation movements $\langle m_1, m_2, \ldots, m_l\rangle$ and a set with length k ($k \leqslant l$) as the template, after C_l^k times comparisons, we obtain the similarities. The label of template with the maximum similarity is chosen as the gesture label of the movements $\langle m_1, m_2, \ldots, m_l\rangle$. Such traditional model is also tested on the test set including the rest 90 samples and the correct recognition rate is 76.67% (see Table 7). This result shows that the correct recognition rate is limited by the scale of train set, since the traditional discriminative models requires pretty number of observations and our dataset is not large enough. In addition, our observation is set-based, some observation sets on the test set never occurs on the training set. In our approach, this observation set is mapped to the one existing on the soft discriminative model trained by small scale train set.

Let us define the recognition result by using the traditional discriminative method as a base. We can infer from Table 7, many gestures are corrected from the base, since the GCRR improves about 13.4%. However, some gestures on the contrary are wrongly corrected from the base. That limitation of our method lies in the "new movement sets". For example, MS_1 ($\langle 1,2,3,4\rangle$, up) and MS_2 ($\langle 4,2,3\rangle$, down) both exist in our rule. Now, an observation

Fig. 15. Perform the gesture using right arm, left arm and even face to the screen.

$\langle 4,2,3,4\rangle$ can be judged as MS_1 $\langle 1,2,3,4\rangle$ under the assumption that movement 1 is misclassified as movement 4. Or, it can be judge as MS_2 $\langle 4,2,3\rangle$ under the assumption that the forth movement 4 is redundant. That is to say, the edit distance between MS_1 and the observation is 1, the same with that between MS_2 and the observation. Our approach ignores the edit distance and just only compute $p(G_j|MS_i) \times p(MS_i)$ as shown in Eq. (4). That is the reason why some gestures on the contrary can be wrongly corrected from the base.

5.1.4. Robustness

The robustness of our system lies in three aspects. The set-based soft discriminative model is the first one, since it has the ability to correct wrongly classified movements and improve 13.4% of GCRR from the base on our dataset. In online testing, this model also ignores meaningless gestures performed in a presentation. The second one is the free style of presentation. One can perform the gesture using either right arm or left arm. The speaker can even face to the screen (see Fig. 15). The third one is the choice of stereo cameras. Though Kinect is what we used, other depth cameras can also be used such as binocular camera, whose depth map calculated in real-time is not as accurate as Kinect's.

6. Conclusions and future work

We propose an approach for real-time gesture recognition based on the Kinect depth data and test it in our dataset. In this paper, PARs, feature extraction method and set-based soft discriminative model are designed originally to fit the online processing. With the assistance of PARs, the detection mode is adjusted to exclude most of meaningless gestures before further gesture recognition. Experimental results show that our approach is efficient due to the detection mode transfer and robust due to the MHI feature and soft discriminative model.

Acknowledgement

Hanjie Wang and Xilin Chen was partially supported by the NSFC under contract Nos. 61025010 and 61001193, and the FiDiPro program of Tekes.

References

[1] M. Blank, L. Gorelick, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, in: ICCV, vol. 2, IEE, 2005, pp. 1395–1402.
[2] A. Bobick, Movement, activity and action: the role of knowledge in the perception of motion, Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences 352 (1358) (1997) 1257–1265.
[3] A. Bobick, J. Davis, The recognition of human movement using temporal templates, PAMI 23 (3) (2001) 257–267.
[4] X. Feng, P. Perona, Human action recognition by sequence of movelet codewords, in: 3D Data Processing Visualization and Transmission, IEEE, 2002, pp. 717–721.
[5] K. Fujimura, X. Liu, Sign recognition using depth image streams, in: FG, IEEE, 2006, pp. 381–386.
[6] Y. Jia, C. Huang, T. Darrell, Beyond spatial pyramids: receptive field learning for pooled image features, in: CVPR, IEEE, 2012, pp. 3370–3377.
[7] A. Jordan, On discriminative vs. generative classifiers: a comparison of logistic regression and naive bayes, Advances in neural information processing systems 14 (2002) 841.
[8] A. Klaser, M. Marszalek, C. Schmid, A Spatio-Temporal Descriptor Based on 3D-Gradients, in: BMVC, 2008, pp. 275:1–275:10.
[9] I. Laptev, On space-time interest points, IJCV 64 (2) (2005) 107–123.
[10] R. Poppe, A survey on vision-based human action recognition, IVC 28 (6) (2010) 976–990.
[11] A. Sadeghipour, L. Morency, S. Kopp, Gesture-based object recognition using histograms of guiding strokes, in: BMVC, 2012.
[12] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local svm approach, in: ICPR, vol. 3, IEE, 2004, pp. 32–36.
[13] P. Scovanner, S. Ali, M. Shah, A 3-dimensional sift descriptor and its application to action recognition, in: Proceedings of the 15th International Conference on Multimedia, ACM, 2007, pp. 357–360.
[14] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from single depth images, in: CVPR, IEEE, 2011, pp. 1297–1304.
[15] Y. Song, D. Demirdjian, R. Davis, Tracking body and hands for gesture recognition: Natops aircraft handling signals database, in: FG, IEEE, 2011, pp. 500–506.
[16] P. Turaga, R. Chellappa, V. Subrahmanian, O. Udrea, Machine recognition of human activities: a survey, IEEE Transactions on Circuits and Systems for Video Technology 18 (11) (2008) 1473–1488.
[17] J. Wang, Z. Liu, J. Chorowski, Z. Chen, Y. Wu, Robust 3d action recognition with random occupancy patterns, in: ECCV, Springer, 2012, pp. 872–885.
[18] J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining actionlet ensemble for action recognition with depth cameras, in: CVPR, IEEE, 2012, pp. 1290–1297.
[19] Q. Wang, X. Chen, W. Gao, Skin color weighted disparity competition for hand segmentation from stereo camera, in: BMVC, 2010, pp. 66.1–66.11.
[20] D. Weinland, E. Boyer, R. Ronfard, Action recognition from arbitrary views using 3d exemplars, in: ICCV, IEEE, 2007, pp. 1–7.
[21] G. Willems, T. Tuytelaars, L. Van Gool, An efficient dense and scale-invariant spatio-temporal interest point detector, in: ECCV, 2008, pp. 650–663.
[22] X. Yang, C. Zhang, Y. Tian, Recognizing actions using depth motion maps-based histograms of oriented gradients, in: Proceedings of the 20th ACM International Conference on Multimedia, ACM, 2012, pp. 1057–1060.