Explore Efficient Local Features from RGB-D Data for One-shot Learning Gesture Recognition

Jun Wan, Guodong Guo, *Senior Member, IEEE,* and Stan Z. Li, *Fellow, IEEE*

Abstract— Availability of handy RGB-D sensors has brought about a surge of gesture recognition research and applications. Among various approaches, one shot learning approach is advantageous because it requires minimum amount of data. Here, we provide a thorough review about one-shot learning gesture recognition from RGB-D data and propose a novel spatiotemporal feature extracted from RGB-D data, namely mixed features around sparse keypoints (MFSK). In the review, we analyze the challenges that we are facing, and point out some future research directions which may enlighten researchers in this field. The proposed MFSK feature is robust and invariant to scale, rotation and partial occlusions. To alleviate the insufficiency of one shot training samples, we augment the training samples by artificially synthesizing versions of various temporal scales, which is beneficial for coping with gestures performed at varying speed. We evaluate the proposed method on the Chalearn gesture dataset (CGD). The results show that our approach outperforms all currently published approaches on the challenging data of CGD, such as translated, scaled and occluded subsets. When applied to the RGB-D datasets that are not one-shot (e.g., the Cornell Activity Dataset-60 and MSR Daily Activity 3D dataset), the proposed feature also produces very promising results under leave-one-out cross validation or one-shot learning.

Index Terms—one-shot learning, gesture recognition, RGB-D data, bag of visual words model.

1 Introduction

Vision-based gesture recognition [1] is a very important and active field of computer vision research due to a lot of varied applications, such as sign language recognition [2], human computer interaction (HCI) [3], robot control [4], video surveillance [5], augmented reality [6] and video annotation [7]. However, the traditional methods are based on strongly supervised learning [8] and require a large number of training samples [9], [10]. For example, the authors [10] demonstrated that the recognition rate can be achieved 96% when at least 50 training samples per class have used to train hidden markov models (HMM). Nevertheless, the recognition rate will be unstable when the number of training samples per class decreases dramatically [10]. In order to explore new algorithms or strategies that are robust to gesture recognition with limited training samples, some recent works [9], [11], [12], [13] have attempted to learn gestures at the other extreme: one-shot learning (that means only one training sample per class) gesture recognition from RGB-D data. In this paper, we also focus on this one-shot learning problem.

One-shot learning gesture recognition from RGB-D data has gained increasing attentions in recent years. Big challenge there is learning of discriminant features and a classifier from very limited training samples. Wu et al. [11] extracted extended motion history image (Extended-MHI) [14] from RGB and depth videos respectively, and then applied multiview spectral embedding (MSE) algorithm [15] to fuse these two Extended-MHI features, and finally used maximum correlation coefficient to achieve gesture recognition. Then a product manifold method [16] is presented, which characterize data tensors (gesture videos) as points on a Grassmann manifold and model it statistically using least squares regression. Later, Goussies et al. [17] proposed a transfer learning method based on decision forests to recognize gestures. Nevertheless, all of the mentioned methods got poor results ($\geq$24% in levenshtein distance (LD) scores[1]) on the Chalearn gesture dataset (CGD) [18]. From the experimental results [11], [16], [17], they demonstrate that the traditional features (e.g. Extended-MHI) and classification models (e.g. decision forest and product manifold) may be not very suitable for one-shot learning.

Fortunately, some other published papers have revealed promising results [9], [19], [20]. Some of them used bag of visual words (BoVW) model with traditional spatiotemporal features [21], [22] or specifically designed features [9]. For example, Wan et al. extended scale invariant feature transform (SIFT) [23] to spatiotemporal domain and proposed 3D enhanced motion SIFT (3D EMoSIFT) [9] and 3D Sparse Motion SIFT (3D SMoSIFT) [20] to extract features by fusing RGB-D data, which are invariant to scale and rotation, and have more compact and richer visual representations. The evaluations of both MoSIFT-based features under BoVW models were provided in [9], [20], which revealed that the best results were below 14% in LD scores on CGD. Furthermore, Konečnỳ and Hagara [19] extracted histogram of oriented gradients (HOG) [21] and histogram of optical flow (HOF) features and used dynamic time warping (DTW)

- *J. Wan and S.Z. Li are with Center for Biometrics and Security Research & National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Room 1411, Intelligent Building, 95 Zhongguancun Donglu, Haidian District, Beijing 100190, China. E-mail: jun.wan, szli@nlpr.ia.ac.cn.*
- *G. Guo is with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26506 USA (e-mail: guodong.guo@mail.wvu.edu).*

Manuscript received March 8, 2015; revised September 29, 2015.

1. LD distance between two strings is defined as the minimum number of operations (insertions, substitutions or deletions) needed to transform one string into the other. LD score is analogous to an error rate. Lower LD values indicate better performances.

to recognize gestures. The best results reported in [19] was about 11% in LD scores. From the top results, we can see that feature selections (whether the specifically designed features or traditional features) are very important for one-shot learning.

In this paper, we propose a novel spatiotemporal feature extracted from RGB-D data, namely mixed features around sparse keypoints (MFSK) for one shot learning of gesture recognition. First, spatial pyramids are built over scale spaces from every depth and grayscale (covered from RGB images) frames, and initial keypoints are detected using speeded-up robust features (SURF) [24] detector in scale spaces. Then, the velocity of each initial keypoint based on Lucas-Kanade optical flow algorithm [25] is calculated from two consecutive frames, with those keypoints whose velocities are below a predefined threshold discarded. After keypoint detection, various descriptors around keypoint volumes are calculated from RGB-D data: 3D SMoSIFT, HOG, HOF and motion boundary histograms (MBH). To cope with varying gesture velocity, the training samples are augmented by artificially building temporal scales. The main contributions of our work are summarized below:

- We provide a thorough review on published methods about one-shot learning gesture recognition from RGB-D data, in which the challenges are analyzed and some future research directions are suggested. As far as we known, it is the first such review which we believe it is a non-trivial work.
- A novel MFSK feature is proposed, inspired by our previous works [9], [20]. Compared with 3D EMoSIFT and SMoSIFT features, the proposed MFSK feature has two traits. The first is the robustness of keypoint detection based on the proposed feature. The other is the fusion of various descriptors, which leads to consistently higher performance than the 3D MoSIFT-based features. Furthermore, the proposed feature is invariant and robust to scale, rotation and partial occlusions.
- To cope with the problem of different velocity of gesture in the data, the training samples are augmented by artificially synthesizing versions of various temporal scales, which proves to be beneficial for the performance.

The rest of the paper is organized as follows. The review on one-shot learning gesture recognition from RGB-D data is presented in Section 2. The proposed MFSK feature and augment of training data is described in Section 3. Then, extensive experiments are provided in Section 4 to evaluate and compare our method with the state-of-the-art methods. Finally, a conclusion is given in Section 5.

2 Review of the State-of-the-Art Methods

Since the Chalearn gesture challenges took place in 2011 and 2012 based on CGD [18] dataset, one-shot learning gesture recognition from RGB-D data has gained increasing attentions and a lot of papers about this field have been published in recent four years.

In this section, we first introduce the differences and connections between RGB and RGB-D gesture recognition

Fig. 1. Some samples from CGD. The first row is RGB images and the corresponding depth images are shown in the second row. It is derived from [9].

as well as the related works on one-shot learning in image-based or video-based domain. Then, we thoroughly analyze these published papers, and introduce these methods into three parts: preprocessing, feature extraction, and temporal segmentation & recognition. In addition, we also introduce the CGD dataset about one-shot learning gesture recognition and discuss the challenges (that we are facing) and trends (which can improve the recognition performances).

2.1 Differences and Connections

Since the release of the Kinect^{TM} sensor (capturing RGB-D images) in late 2010 by Microsoft, RGB-D gesture recognition has gained a lot of attentions. Similar to the RGB gesture recognition, the traditional methods may be applied to gesture recognition based on RGB-D data. For example, HOG and HOF features [19], [26], [27] are also widely used to extract features, and HMM [27] and DTW [20], [28] are commonly applied in temporal segmentation or gesture recognition from RGB-D data.

However, compared with gesture recognition based on general RGB images, RGB-D sensors can capture RGB and depth images simultaneously. It provides an easy and inexpensive access to depth information, which is convenient to obtain the object mask using the simple ostu method [11], [29]. Besides, some elaborately designed features [9], [20], [30] are also proposed for fusing RGB-D data.

One-shot learning problem is primarily presented by L. Fei-Fei [31] in computer vision. It used a generative object category model and variational Bayesian framework for representation and learning of visual object categories. Comparing with one-shot learning in image-based domain [31], [32], we focus on one-shot learning in video-based domain, especially for RGB-D videos.

2.2 CGD Dataset

The goal of the CGD dataset is to employ systems to perform gesture recognition from videos containing diverse backgrounds, using a single example per class, i.e., one-shot learning. CGD comprises 54,000 different gestures divided into 540 batches. Gestures were recorded in RGB and depth video using Kinect^{TM} camera. The data set was divided into development (480 batches), validation (20 batches) and additional batches for evaluation (40 batches, referred to as final batches). Each batch is associated to a different gesture vocabulary, and it contains exactly one video from each gesture in the vocabulary for training and several videos containing sequences of gestures taken from the same vocabulary for testing. Each batch contains 100 gestures. The

Fig. 2. It shows some samples derived from [20] for the new subsets on CGD. (a) untranslated; (b) translated; (c) scaled; (d) occluded.

Fig. 3. It shows the results by the preprocessing step. (a) original frame, (b) image smoothing, (c) background removal. The image is derived from [33].

number of training videos/gestures ranges from 8 to 12 depending on the vocabulary. There are 47 videos in each batch (frame size 320×240, 10 frames/second, recorded by 20 different users) and each video contains 1 to 5 gestures. Some samples are shown in Fig. 1.

To test the robustness of recognition to body translation, images scaling and partial occlusions, CGD provided some more challenging subsets. Some representative images of these subsets are shown in Fig. 2 and their descriptions are illustrated below:

1) $utran$ data ($utran01 - utran20$): The untranslated data are selected batches from the original data including a large background area in which no gesture was taking place. It comprises 2000 gestures.

2) $trans$ data ($trans01 - trans20$): Using the $utran$ data, one made a different horizontal translation to generate the $trans$ data.

3) $scale$ data ($scale01 - scale20$): Similarly, one applied various scaling factors on the $utran$ data to generate the $scale$ data.

4) $uoccl$ data ($uoccl01 - uoccl20$): This subsets are selected from CGD as the unoccluded data. It comprises 2000 gestures.

5) $occlu$ data ($occlu01 - occlu20$): One added a red rectangle with 10*240 pixels in the center of every frame of both RGB and depth videos from the $uoccl$ data. This red rectangle is treated as occlusions.

In order to facilitate our discussions in the following parts, we list all the methods and recognition results of the published papers in Tables 1 and 2, and give some basic notions here.

Metric of Evaluation: LD score is used to evaluate the performances of different methods, which is explained in Section 1. The results shown in this paper are LD scores unless mentioned otherwise in our experiments.

Explanation of abbreviations: $devel$ means the development batches ($devel01 - devel20$); $valid$ means the validation batches ($valid01 - valid20$); $final1$ means the first 20 batches for evaluation ($final01 - final20$); $final2$ means the left 20 batches for evaluation ($final21 - final40$);

2.3 Preprocessing

Before feature extraction from RGB-D data, a few authors employed some image processing techniques to obtain the body mask [11], [29], [33]. Wu et al. [11] applied the otsu method of global image threshold [34] to segment human bodies from the background, then used a median filter and a morphological operator for noise reduction. As the authors' suggestion [11], the performances were improved about 9% via the preprocessing operators. Similarly, the otsu method and filter techniques are applied to achieve background or outlier removal in [29]. Besides, image smoothing via a median filter and in-painting [33] are used to restore the images in the sequence by filling all the holes. Some representative results are shown in Fig. 3.

In previous works, the aim of preprocessing is to obtain more pure human body mask which will be used to extract more efficient features [11], [29], [33].

2.4 Feature Extraction

As shown in Table 1, the motion and gradient features are widely used in the published papers. For instance, nearly 50% listed papers used HOG or HOF features. In [26], [27], Malgireddy et al. detected interest points by taking the difference between two consecutive depth images, and then extracted HOG and HOF features at each interest point. Owing to the calculation of descriptors around interest points, the extracted HOG and HOF features are local. However, in [19], [29], [33], [35], [36], the HOG or HOF features are extracted globally because they are directly computed from RGB or depth frames.

Another traditional features are motion energy information (MEI) [29], MHI [39] and its variants. These two features are used in [11], [17], [29]. Compared with the MEI or MHI, the performances of HOG and HOF features are better from the statistical results shown in Table 2. Moreover, a new descriptor VFHCRH [30] (Viewpoint Feature Histogram Camera Roll Histogram) was also introduced at the feature extraction stage.

Unlike the above mentioned features, some novel features are specifically designed for RGB-D videos. In [9], Wan et al. proposed the 3D EMoSIFT feature which fuses RGB and depth data to calculate SIFT-based descriptors. Later, the authors [9] extended their previous works and presented 3D SMoSIFT [20]. These two SIFT-based features first detect interest points from the scale spaces, and then build three dimensional gradient and motion spaces around each interest point, and finally calculate SIFT-like descriptors on both 3D spaces, respectively. The proposed 3D MoSIFT-based features are invariant to scale, transition, and partial occlusions, which can be found in Table 2. We also find that 3D MoSIFT-based features under the BoVW model achieve the best performances in most cases from Table 2.

As the above discussions, the traditional features (HOG and HOF) and the specifically designed features (3D E-MoSIFT and 3D SMoSIFT) have achieved promising performances. But HOG and HOF features are sensitive to scaled and translated data [13], that is because HOG and HOF features are rigidly positioned on image feature maps

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2513479, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 1
The survey of published methods about one-shot learning gesture recognition from RGB-D data.

Index	Papers	Preprocessing	Feature extraction	Temporal segmentation & Recognition methods	Simultaneously	Published year
1	[26] [27]	No	HOG, HOF	mcHMM + BoVW model + LDA	Yes	2012
2	[11]	Yes	Extended-MHI	Appearance-based & Maximum Correlation Coefficient	No	2012
3	[16]	No	Raw data (RGB-D)	Appearance-based & Product Manifolds	No	2012
4	[28]	No	Motion maps	DTW & PCA-based recognition error	No	2013
5	[35] [36]	Yes	3D Histogram of Scene Flow + Global HOG	Sliding window + Sparse coding + linear SVM	Yes	2013 2013
6	[9]	No	3D EMoSIFT	DTW & BoVW model + nearest neighbors (NN) classifier	No	2013
7	[29]	Yes	MEI + HOG	DTW & PCA-based recognition error	No	2013
8	[20]	No	3D SMoSIFT	DTW & BoVW model + nearest neighbors (NN) classifier	No	2014
9	[19]	No	HOG + HOF	DTW + Quadratic-Chi histogram distance	Yes	2014
10	[17]	No	MHI	DTW & Transfer learning decision trees + naive Bayes model	No	2014
11	[37]	No	3D EMoSIFT	DTW & CSMMI	No	2014
12	[30]	No	HOG, HOF, VFHCRH	Probability-based DTW & BoVW model	No	2014
13	[33]	Yes	HOG	Conditional level building (CLB)	Yes	2015
14	[38]	No	Improved Principle Motion	Appearance-based & Multi-layered Classifier	No	2015

TABLE 2
It shows the recognition performances of the published papers on different subsets of CGD. The bold value in each column is the best result of the corresponding subsets except the last two rows in which the results of our method are listed. Compared with the results of published papers, our methods achieve the best performances in all subsets of CGD.

Index	Papers	devel	valid	final1	final2	utran	trans	scaled	uoccl	occlu
1	[26], [27]	0.2409	0.2333	0.1847	0.1853	0.3594	0.3962	0.4152	-	-
2	[11]	0.26	0.2969	-	-	-	-	-	-	-
3	[16]	0.2873	-	-	-	-	-	-	-	-
4	[28]	0.3016	0.3178	0.2641	0.2124	-	-	-	-	-
5	[35], [36]	0.2511	-	-	-	-	-	-	-	-
6	[9]	0.1945	**0.1595**	**0.1382**	0.1259	0.2635	0.253	**0.254**	0.1185	0.1375
7	[29]	0.2241	-	-	-	-	-	-	-	-
8	[20]	0.1965	-	-	0.114	**0.257**	**0.2475**	0.263	**0.114**	**0.1335**
9	[19]	0.2199	0.2001	0.1702	**0.1098**	0.2896	0.5993	0.5296	-	-
10	[17]	0.3155	-	0.2834	0.2475	-	-	-	-	-
11	[37]	**0.1876**	-	-	-	-	-	-	-	-
12	[30]	0.2662	-	-	-	-	-	-	-	-
13	[33]	-	0.2105	0.1642	0.1687	-	-	-	-	-
14	[38]	0.1964	-	-	-	-	-	-	-	-
Ours	MFSK+BoVW	0.1645	0.1270	0.1395	0.0925	0.2390	0.2120	0.2375	0.1145	0.1150
	MFSK+BoVW+TS	0.1590	0.1242	0.1326	0.0900	0.2315	0.2102	0.2300	0.0970	0.1125

[19]. For example, in [19], the proposed method using HOG and HOF features can get 0.2896 on $utran$ data, while the performances drastically declined at least 24% on $trans$ and $scaled$ data. However, in [20], the results via the 3D SMoSIFT feature are very robust (about 1%) to $scale$ and $trans$ data. Therefore, 3D MoSIFT-based features can well handle more complex cases, such as scaled, translated or partially occluded data.

2.5 Temporal Segmentation & Recognition

In order to recognize gestures in continuous video streams, gesture recognition and temporal segmentation can be simultaneous or non-simultaneous depending on different algorithms.

1) ***Non-simultaneous segmentation and recognition***

As shown in Table 1 (see the fifth and sixth columns), temporal segmentation and gesture recognition are not done simultaneously in most of the published papers [9], [11], [16], [17], [20], [28], [29], [30], [37], [38]. Usually those approaches first perform temporal segmentation to localize isolate gestures, and then recognize each isolate gesture.

For automatic segmentation, the authors in [9], [17], [20], [28], [29], [37] used dynamic time warping (DTW) algorithm to split the continues gestures into some isolate gestures. More specifically, the difference image (see Fig. 4(b)) is first computed by subtracting consecutive frames in a video. Then a grid of equally spaced cells is defined over the difference image. The default size of the grid is 3×3 as shown in Fig. 4(c). For each cell, the average value is calculated in the difference image, so a 3×3 matrix is generated. Finally, this matrix can be flattened into a vector which is called motion feature [9]. After motion feature extraction, the DTW distance can be calculated between the testing sample and the traing samples, and then one

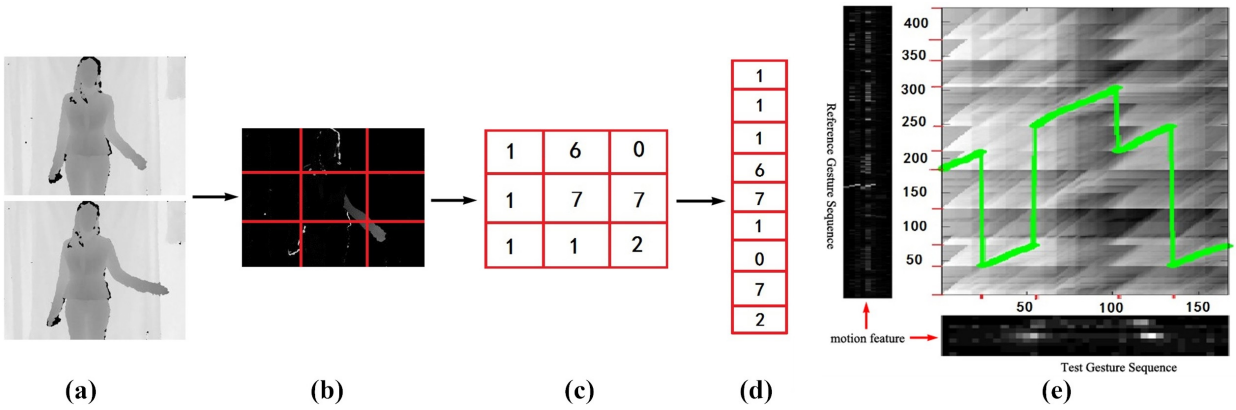


Fig. 4. Examples for the calculation of motion feature vector (a-d) and temporal segmentation by DTW algorithm (e). Images are derived from [9].

can apply the Viterbi algorithm [40] to find the temporal segmentation (see Fig. 4(e)). The code[2] of DTW was released by the organizers of the Chalearn gesture challenge.

Appearance-based method [11], [16], [38] is another way to achieve temporal segmentation, which find candidate cuts based on similarities with respect to the resting position or the amount of motion. Liu [16] thought that hands return to a resting position between each pair of neighboring gestures, and the correlation coefficient were calculated between the first frame (as a template) and subsequent frames. Then the gesture segments can be localized by identifying the peak locations from the correlations. Similarly, Wu et al. [11] found the frames that are similar to the beginning and ending frames in the unsegmented testing video sequence and defined them as the interval frames between two gestures in a video sequence. Jiang et al. [38] measured the quantity of movement of each frame and then got the candidate boundaries based on a predefined threshold, and finally refined the candidate boundaries using sliding windows.

After temporal segment, different features can be extracted from each isolate gesture. Then, in order to recognize gestures, a lot of methods can be selected, such as BoVW model with nearest neighbor (NN) classifier [9], [20], class-specific maximization of mutual information (CSMMI) [37], PCA-based recognition error [28], [29], product manifolds [16] , transfer learning decision trees (TLDT) with naive Bayes model [17], and maximum correlation coefficient [26], [27] , which are summarized in Table 1. In addition, we can see that the traditional generative models (i.e. DBN, CRF, ANN) are rarely used for one-shot learning. That is because it would be very difficult to train these models effectively due to the lack of training data, and very limited training samples can easily lead to the underfitting problem. On the contrary, nonparametric methods, such as the NN classifier [9], [20] can work surprisingly well for one-shot learning.

2) ***Simultaneous segmentation and recognition***

The simultaneous segmentation and recognition techniques are very popular to recognize gestures in traditional methods [41], [42]. For one-shot learning gesture recognition, some papers [19], [26], [27], [33], [35], [36] also execute gesture segmentation and recognition simultaneously.

2. http://gesture.chalearn.org/data/sample-code

Malgireddy et al. [26], [27] proposed a temporal Bayesian model for classifying, detecting and localizing gestures in video sequences. First, after HOG and HOF feature extraction, the feature descriptors over the entire space of gestures are converted to "visual words" via the BoVW model and latent dirichlet allocation (LDA) [43]. So each frame can be represented by a histogram over the visual words in that frame. Then, multiple channel HMM (mcHMM) is proposed for gesture spotting and recognition. Unlike classic HMM, this model has multiple channels, where each channel is represented as a distribution over the visual words corresponding to that channel. And mcHMM has multiple observations per state and channel.

Fanello et al. [35], [36] extracted motion and gradient features (3D Histograms of Scene Flow and Global HOGs), and then adopted sparse coding [44], [45] to capture high-level patterns from the extracted features, and finally used a sliding window with linear support vector machines (SVMs) [46], [47] to simultaneously segment and recognize gestures.

Besides, after feature extraction, Konečnỳ and Hagara [19] simply utilized DTW with Quadratic-Chi histogram distance to segment and recognize gestures. The proposed method achieves high performances on regular data, such as $valid$, $devel$ and $final$ data. But the proposed method is not good at scaled and translated data (see Table 2). That means the extracted HOG and HOF features are sensitive to scaling and translations.

Lastly, Krishnan and Sarkar [33] proposed the conditional level building (CLB) method to achieve this purpose. Unlike the original level building algorithm [46], [48], [49] using DTW as the distance measure, CLB is based on conditional distances. From the experimental results in Table 2, CLB can get a relatively high performance.

2.6 Challenges and Trends

Here, we first list some challenges that we are facing for one-shot learning gesture recognition, and discuss the trends which can improve recognition performances.

2.6.1 Challenges

We are facing the challenges on one-shot learning problem. First, there is one training sample per class. It is one of the biggest challenges to train a robust model using very limited

samples. Second, feature extraction is also one of the most difficult challenges. The effective features usually can get better performances. When some errors or omissions may be happened in performing gestures, it faces great challenges at the feature extraction stage. Third, large variances of intra-class and small variances of inter-class are also a challenging problem. For instance, some users are less skilled than others when performing gestures or some gestures from different classes are very similar. Fourth, partial occlusions may occur, and it may face more complex environments (i.e. variations in background, clothing, skin color, lighting, resolution). Lastly, only the RGB-D data is provided. The skeleton or audio data is not available. Moreover, outliers or black areas may occur in depth videos.

2.6.2 Trends

In order to improve the performance using limited training samples, we list some trends as below. We think that these trends may be useful for researchers in this field.

Designing new features from RGB-D data. Feature extraction is one of the biggest challenges. Different people have different speeds, trajectories and spatial positions to perform the same gesture. Even when a single person performs the gestures, the trajectories are not identical. In order to overcome these problems, the extracted features will be effective if they are robust to scale, translation, rotation and partial occlusions, i.e. 3D MoSIFT-based features [9], [20] shown in Table 2.

Fusing different features. The combination of different features can also boost the performance. In the following, we will prove this statement.

Selecting suitable models. Through analysis of the published papers in Tables 1 and 2, DTW [19] and BoVW models [9], [20] can achieve the top performances.

Augmenting training samples. In order to augment training samples, some strategies can be used to generate new training samples: 1) building temporal scales, which will be illustrated in Section 3; and 2) adding some noise in training samples to cover more variations in learning.

3 The Proposed Approach

We first introduce the MFSK feature and then give the detailed information about temporal scales to augment the training samples. Lastly, the inspirations of using MFSK features are given, and finally a short summary of the proposed approach is presented.

3.1 Mixed Features around Sparse Keypoints (MFSK)

As the previously proposed features [9], [20], [50], we first find keypoints and then calculate the descriptors around the regions of each keypoint. However, MFSK is different from these MoSIFT-based features [9], [20], [50]: 1) Having more robust keypoint detection strategies; 2) Having more descriptors around the detected keypoints.

Concretely, MFSK features broadly consist of three stages. First, the spatial pyramid as the scale space is built for every gray and depth frame, which is similar to the 3D SMoSIFT feature [20]. In order to make this paper more self-contained, we briefly introduce the processes of building spatial pyramids. Second, keypoint detection around the motion regions is applied in scale spaces via SURF detector [24] and tracking techniques. Third, different descriptors are calculated in local patches around keypoints.

3.1.1 Spatial Pyramid Building

For a given sample including two videos (an RGB video and a depth video[3]), we can obtain a grayscale image G_t (converted from RGB frame) and a depth image D_t at time t. Then one pyramid can be built from G_t or D_t via downsampling. Formally, at time t, two pyramids P_G^t and P_D^t can be constructed via Eq. (1).

$$\begin{aligned} G_t^l(x,y) &= G_t(2^{(l-1)}x, 2^{(l-1)}y) \qquad 1 \le l \le L \\ D_t^l(x,y) &= D_t(2^{(l-1)}x, 2^{(l-1)}y) \qquad 1 \le l \le L \end{aligned} \tag{1}$$

where G_t^l (or D_t^l) is the image at the l^{th} level in the pyramid, (x,y) is the coordinate of G_t^l (or D_t^l). Hence, at time t, the pyramids P_G^t and P_D^t can be built, that is $P_G^t = \{G_t^1, G_t^2, ..., G_t^L\}$, $P_D^t = \{D_t^1, D_t^2, ..., D_t^L\}$.

Fig. 5 shows two pyramids P_G^t, P_G^{t+1} (or P_D^t, P_D^{t+1}) built from two consecutive grayscale (or depth) frames at time t and $t+1$. The original frames are of size 320×240. As shown in Fig. 5, each pyramid has three levels and images in the first level are original frames from RGB-D videos. After building pyramids, we illustrate how to find robust keypoints around motion regions in both RGB and depth frames.

3.1.2 Sparse Keypoint Detection

After building spatial pyramids, keypoints around motion regions can be detected in each spatial scale by two steps.

Initial Keypoint Detection via SURF Detector. Bay et al. [24] proposed a Fast-Hessian detector. Concretely, for a point $p(x,y)$ in an image I, its Hessian matrix $H(p,\sigma)$ in p at scale σ is defined as follow:

$$H(p,\sigma) = \begin{bmatrix} L_{xx}(p,\sigma) & L_{xy}(p,\sigma) \\ L_{xy}(p,\sigma) & L_{yy}(p,\sigma) \end{bmatrix} \tag{2}$$

where $L_{xx}(p,\sigma)$ is the convolution of the Gaussian second order derivative $\frac{\partial^2 g(\sigma)}{\partial^2 x}$ with the image I in point p, and similarly $L_{xy}(p,\sigma) = \frac{\partial^2 g(\sigma)}{\partial xy}$, $L_{yy}(p,\sigma) = \frac{\partial^2 g(\sigma)}{\partial^2 y}$.

In order to achieve fast convolution calculation, the second order Gaussian derivatives in Eq. (2) can be computed at a very low computational cost using integral images [51], and the calculation time is independent of the gaussian filter size. This is important when big filter sizes are used.

Then, the $9\times$ 9 box filters in Fig. 6 are approximations of a Gaussian with $\sigma = 1.2$ for computing the blob response maps, which are denoted by D_{xx}, D_{yy} and D_{xy}. More specifically, the blob response maps are calculated via filtering one image with the box filters shown in Fig. 6. The weights applied to the rectangular regions are kept simple for computational efficiency. So the Hessian's determinant can be approximated as

$$det(H_{approx}) = D_{xx}Dyy - (wD_{xy}^2) \tag{3}$$

3. The depth values are normalized to [0 255] in depth videos.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2513479, IEEE Transactions on Pattern Analysis and Machine Intelligence

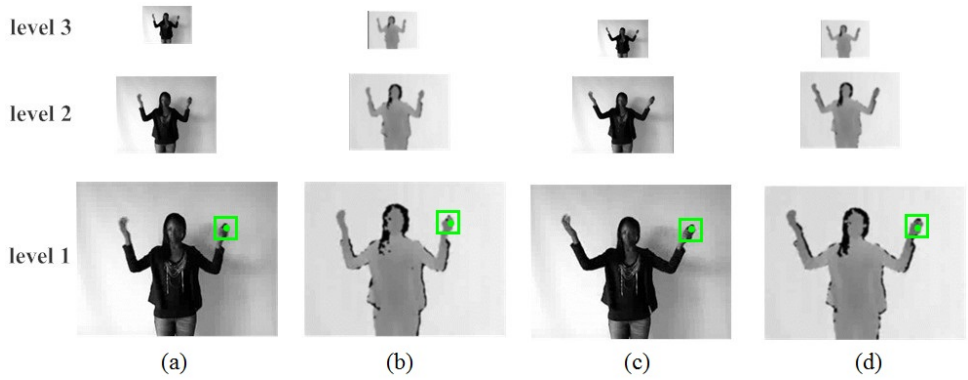


Fig. 5. Building four pyramids from two pair of consecutive frames. (a) P_G^t at time t; (b) P_D^t at time t; (c) P_G^{t+1} at time $t+1$; (D) P_D^{t+1} at time $t+1$. The detected keypoints are denoted by the green circle, and the extracted local patches are shown within the green rectangles. The four local patches are denoted by $g_t, d_t, g_{t+1}, d_{t+1}$ from left to right.

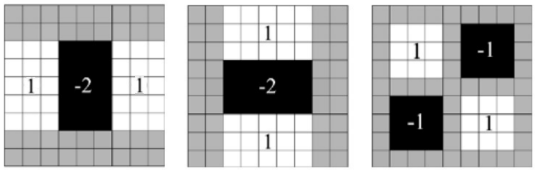


Fig. 6. The approximation for the second order Gaussian partial derivative in x-(D_{xx}), y-(D_{yy}) and $xy-$direction (D_{xy}). The grey regions are equal to zero.

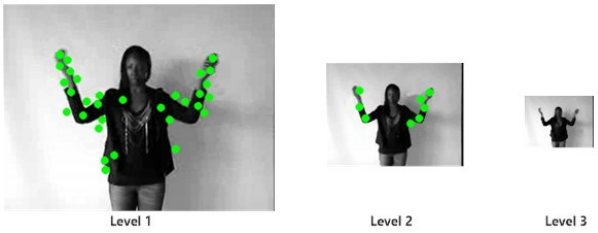


Fig. 7. The detected keypoints via SURF detector and motion filtering. The keypoints are detected in the first two levels from the pyramids of Fig. 6, and no keypoints are found in the third level.

where w is the relative weight of the filter responses that is used to balance the expression for the Hessian's determinant. And w is

$$w = \frac{|L_{xy}(1.2)|_F |D_{yy}(9)|_F}{|L_{yy}(1.2)|_F |D_{xy}(9)|_F} = 0.912... \simeq 0.9 \qquad (4)$$

where $|x|_F$ is the Frobenius norm.

Therefore, the approximated determinant of the Hessian for every point can be computed in the image. Then the maxima of the determinant of the Hessian matrix can be found in a $3\times3\times3$ neighborhood [24], and these points with maxima are our initial keypoints. Initial keypoints can be easily found via the function $SurfFeatureDetector$ from opencv library [52].

Keypoint Detection via Motion Filtering. We assume that the initial keypoints are found at time t, then we can use Lucas-Kanade optical flow method [25] to track these keypoints at time $t+1$ and calculate their velocities. If the velocity of one keypoint is very small, this means the region around this keypoint may have less motion. Therefore, these keypoints with small velocities will be discarded. That is to say, when the absolute velocity $|v|$ of a keypoint at the l^{th} level is larger than a given threshold τ^l, this point will become a keypoint. τ^l is defined as,

$$\tau^l = max(\overbrace{max(\alpha|v^l_{max}|, 0.5^{l-1}\beta)}^{local\ constraint}, \overbrace{\delta}^{global\ constraint}) \qquad 1 \le l \le L, 0 < \alpha < 1 \qquad (5)$$

where $|v^l_{max}|$ is the maximum value of absolute velocities of interest points at the l^{th} level in the pyramid. The parameter $max(\alpha|v^l_{max}|, 0.5^{l-1}\beta)$ determines the motion constraint at the l^{th} level. δ is the global parameters, which means the absolute velocity of each keypoint is not less than δ. According to [20], we fixed $\alpha = 0.15$, $\beta = 0.8$ and $\delta = 0.5$. If the values α, β and δ are larger, the selected keypoints will have large motions.

From the previous work [20], we know that when it uses the gray images to detect keypoints for both initial keypoint detection and motion filtering, it will get a relatively high performance. Therefore, we also use gray images for keypoint detection.

3.1.3 Feature Descriptor Calculation

After keypoint detection, the feature descriptors can be computed from the local patch around every keypoint. As shown in Fig. 5, we suppose that one keypoint denoted by the green circles has been detected. Then we can extract four local patches $(g_t, d_t, g_{t+1}, d_{t+1})$ around the keypoint, which are denoted by green rectangles. The size of the patch is $\Gamma\times\Gamma$ pixels. Finally, we compute four descriptors (3D SMoSIFT, HOG, HOF and MBH) from the extracted local patches as shown in Fig. 8, and the final descriptor is a concatenation of these descriptors. To embed structural information, the local patch is subdivided into $\gamma\times\gamma$ cells to calculate HOG, HOF and MBH descriptors.

3D SMoSIFT. As shown in [9], [20], 3D SMoSIFT yields excellent results for one-shot learning in comparison with other state-of-the-art descriptors. Especially, as shown in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2513479, IEEE Transactions on Pattern Analysis and Machine Intelligence

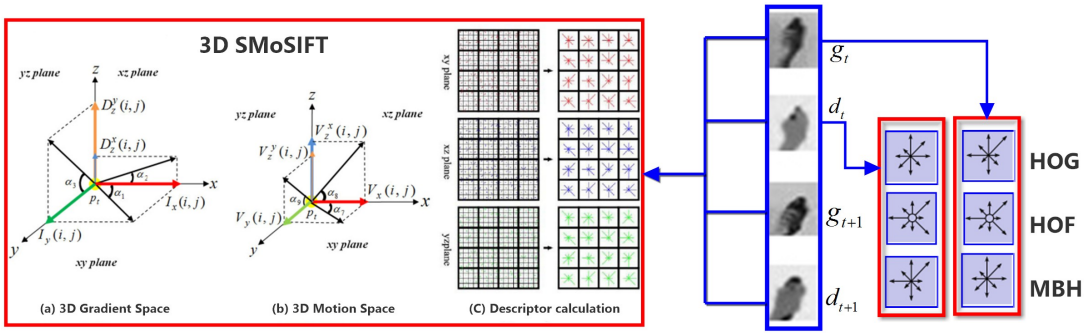


Fig. 8. Computing the descriptors (3D SMoSFIT, HOG, HOF, MBH) from the local patch around every keypoint.

Table 2, 3D SMoSIFT is robust to scale, translation and partial occlusions.

We compute the 3D SMoSIFT descriptors using the four patches $(g_t, d_t, g_{t+1}, d_{t+1})$. For 3D SMoSIFT, 3D gradient and 3D motion spaces are constructed by fusing RGB-D data shown in Fig. 8. Then in each 3D space, we map 3D space into three 2D planes: xy, yz and xz plane, and SIFT descriptors are calculated on each plane. Therefore, 3D SMoSIFT consists of six SIFT descriptors, and 3D SMoSIFT size is 768 (6×128). The detailed information about the calculation of 3D SMoSIFT can be found in [9], [20].

HOG and HOF. HOG and HOF descriptors [53] are widely used for action recognition, which have achieved very promising results [54]. Interestingly, as shown in Table 1, they are also widely used for one-shot learning gesture recognition. And some papers [19], [26], [27] also achieved high performances (see Table 2).

We compute the HOG and HOF descriptors in two patches g_t and d_t. For both HOG and HOF, orientations are quantized into η bins with full orientations, and the magnitudes are used for weighting [55]. The final descriptor size for both g_t and d_t is $2\times\gamma\times\gamma\times\eta$ for HOG and $2\times\gamma\times\gamma\times\eta$ for HOF.

Motion Boundary Histograms (MBH). To overcome the camera motion, Dalal et al. [56] proposed MBH descriptor by computing derivatives separately for the horizontal and vertical components of the optical flow. Since MBH represents the gradient of the optical flow, locally constant camera motion is removed and information about changes in the flow field (i.e., motion boundaries) is kept. MBH is more robust to camera motion than optical flow.

We compute MBH descriptors in two patches g_t and d_t. The MBH descriptor separates optical flow into its horizontal and vertical components. Spatial derivatives are computed for each of them and orientation information is quantized into histograms. The magnitude is used for weighting. The orientations are also quantized into η bins. Then we can obtain a η-bin histogram for each component (i.e. MBHx and MBHy). Then the final descriptor size for both g_t and d_t is $2 \times \gamma \times \gamma \times \eta$ for MBHx, and $2 \times \gamma \times \gamma \times \eta$ for MBHy, respectively.

3.2 Building Temporal Scales

To further boost the recognition performance, we artificially augment the training samples via building temporal scales. Here, we define the temporal scales as $\Omega = [\omega_1, ..., \omega_n]$, where n is the size of temporal scales, $\omega_i \in [0, 1]$. And we can generate n new videos from one training sample.

Specifically, we assume that a video is represented by $V = [f_1, f_2, ..., f_N]$ with N frames, where f_i is the i^{th} frame, V is the original video. For a temporal scale value $\omega_i \in [0, 1], 1 \leq i \leq n$, a new video V_i is generated via linear interpolation. The full description is given in Algorithm 1.

When there are n temporal scales $\Omega = [\omega_1, ..., \omega_n]$, we can generate n new videos $V_1, V_2, ..., V_n$ from the original video (one video per temporal scale).

Algorithm 1 Building one new video via a temporal scale value w_i

Input: A temporal scale value: w_i;
Input video: $V = [f_1, f_2, ..., f_N]$;
Output: A new video: V_i
1: Initialization: $V_i = [\,]$;
2: Calculate the frame number of V_i: $N' = floor(\omega_i \times N)$, where N' is an integer value;
3: Calculate the step Δ: $\Delta = N/(N' - 1)$;
4: **while** $idx \leq N$ **do**
5: Selecting one frame f: $f = V(idx)$
6: $V_i = [V_i\ f]$
7: $idx = floor(idx + \Delta)$, idx is a integer value
8: **end while**

3.3 Inspirations and Summary of the Proposed Approach

Inspirations. There are two reasons that we calculate these feature descriptors (3D SMoSIFT, HOG, HOF and MBH) for the MFSK feature.

1) For one-shot learning gesture recognition in RGB-D data, the 3D SMoSIFT, HOG and HOF can achieve high performances in previous works [9], [19], [20].
2) For video-based action recognition, the HOG, HOF and MBH are widely used [53], [54], [55]. For example, the dense trajectory method [55] is one of the state-of-the-art methods for action recognition, which also used HOG, HOF and MBH.

Owing to the above reasons, the MFSK feature is proposed. As far as we known, it is the first time the MBH feature is used for one-shot learning gesture recognition. From the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2513479, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 3
Parameters: cell number $\gamma = 2$, bin number $\eta = 8$. The LD scores are calculated with different block sizes ($devel01 - devel20$).

patch size $\Gamma \times \Gamma$	8×8	16×16	32×32	64×64
LD score	0.2550	0.1840	0.1645	0.1630

experiments in this paper, the MFSK feature (combined with 3D SMoSIFT, HOG, HOF and MBH) can get very promising results.

Summary. In this section, we propose the MFSK feature, which includes different descriptors (3D SMoSIFT, HOG, HOF, MBH). Therefore, the proposed features have similar properties of these four descriptors. According to our experimental results, the proposed feature outperforms currently published state-of-the-art methods on challenging data of CGD. Besides, in order to further boost the recognition performance, we propose a simple method to build several temporal scales for augmenting training samples.

4 Experimental Results

In our experiments, we use the BoVW model to evaluate the proposed feature, where NN classifier is used for all subsets of CGD. The LD score (normalized by the length of the truth labeling) [13] is used to evaluate the performance. We use a parameter ξ instead of the codebook size M (which is used in [9]) in BoVW model. That is because the number of extracted features from training samples is varied. If a given codebook size M is too large, it may cause over-clustering on some batches, which will affect the final performance [9]. Therefore, we set different codebook sizes to different batches when we use a given value for ξ. The corresponding codebook size can be calculated, $M = \xi \times L_{tr}$, where L_{tr} is the number of features extracted from training samples on a certain batch. Unless mentioned otherwise, we set $\xi = 0.5$ which can obtain a high performance as shown in [9].

First, we discuss the parameters of the proposed method which include parameters of MFSK features and temporal scales. Then, we compare our method with current state-of-the-art methods on CGD, Cornell Activity Dataset-60 (CAD-60) and MSR Daily Activity 3D datasets. And we note that we have released the code about the proposed MFSK feature (https://mloss.org/revision/view/1866/).

4.1 Parameter Settings

This part gives the discussion of parameters in the proposed method. First, we analyze the parameters of MFSK features: cell number γ, bin number η, the patch size $\Gamma \times \Gamma$. Then, we discuss the parameter of temporal scales: Ω. For determining the parameters of MFSK features, we test the proposed feature under the BoVW model without temporal scales. We should declare that according to the results of [20], we set the pyramid levels $L = 3$.

4.1.1 Parameters of MFSK Features

We use a simple strategy to decide these three parameters. At first, we keep $\gamma = 2, \eta = 8$ and set $\Gamma = [8, 16, 32, 64]$. The results are shown in Table 3, we can see that the performances will be better when the patch size increases, and the result of $\Gamma = 64$ is slightly better than $\Gamma = 32$. With the trade off between time complexity and recognition performance, we set $\Gamma = 32$.

TABLE 4
Parameters: patch size $\Gamma = 32 \times 32$. The LD scores are calculated with different cell number γ and bin nmuber η ($devel01 - devel20$). The values in the brackets are descriptor (HOG+HOF+MBH) sizes.

γ \ η	2	4	8	16
1	0.193 (16)	0.185 (32)	0.179 (64)	0.176 (128)
2	0.1795 (64)	0.1775 (128)	**0.1645** (256)	0.171 (512)
3	0.1715 (144)	0.176 (288)	0.171 (576)	0.1815 (1152)
4	0.1775 (256)	0.1765 (512)	0.1895 (1024)	0.2015 (2048)

TABLE 5
Parameters: $\gamma = 2$, $\eta = 8$, $\Gamma = 32$. The LD scores are calculated with different block sizes ($devel01 - devel20$).

temporal scale Ω	LD score
$[1]$	0.1645
$[0.4, 0.8, 1]$	0.1590
$[0.4, 0.6, 0.8, 1]$	0.1580

Then, we set different values for $\gamma \in [1, 2, 3, 4]$ and $\eta \in [2, 4, 8, 16]$. As shown in Table 4, the performances of the MFSK features are very stable, and the best performance is 0.1645 when $\gamma = 2, \eta = 8$. Therefore, we set $\gamma = 2, \eta = 8$.

4.1.2 Parameters of Temporal Scales

Here, we test three cases: $\Omega = [1]$, $\Omega = [0.4, 0.8, 1]$ and $\Omega = [0.4, 0.6, 0.8, 1]$. The results are shown in Table 5, where the results for $\Omega = [0.4, 0.8, 1]$ and $\Omega = [0.4, 0.6, 0.8, 1]$ are better than the original results by improving about 0.55%. When $\Omega = [1]$, it means temporal scales is not used. It also demonstrates that the performances are very stable when Ω is changed. And in our experiments, we set $\Omega = [0.4, 0.6, 0.8, 1]$, because it is more stable when gestures have different speeds.

From the above discussions, we set $\gamma = 2, \eta = 8, \Gamma = 32$ and $\Omega = [0.4, 0.6, 0.8, 1]$ unless mentioned otherwise in our experiments.

4.2 Comparisons

4.2.1 Comparison within MFSK Features

We know that the MFSK feature consists of four basic components, namely 3D SMoSIFT, HOG, HOF and MBH features. In order to find the effectiveness of each component, we evaluate these four features in our experiments separately or jointly, and the results are calculated under the BoVW model without temporal scales. The HOG and HOF features are used simultaneously, that is because HOGHOF usually outperforms the HOG and HOF features [9], [30], [55]. The results are shown in Fig. 9, where we can see that:

1. For individual features, the 3D SMoSIFT gets the best performance, and the order of performance decrease is 3D SMoSIFT $>$ HOGHOF $>$ MBH.

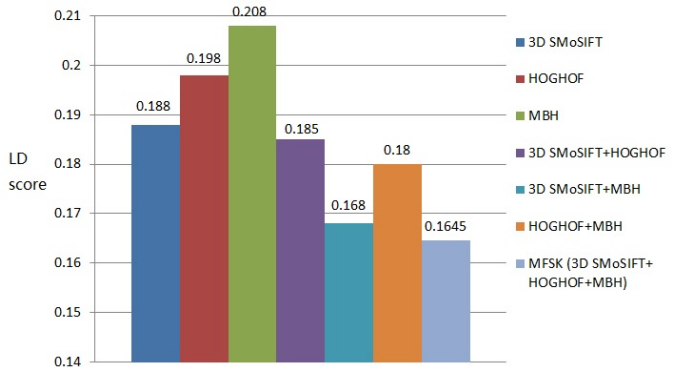


Fig. 9. The performances of different components in the MFSK feature ($devel01 - devel20$). It can be seen that the MFSK feature achieves the best performance.

2. For the composite features, MFSK achieves the best performance, which is slightly better than 3D SMoSIFT + MBH. And the performances of another two combined features (3D SMoSIFT + HOGHOF, HOGHOF + MBH) are poorer than the MFSK or 3D SMoSIFT + MBH.

3. Overall, the combination of features can deliver better performances than the individual features.

4.2.2 Comparison with Other Spatiotemporal Features

In our experiments, we use the BoVW model without temporal scales to evaluate traditional spatiotemporal features (i.e. Cuboid[4] [57], STIP[5] [58] (Harris3D+HOGHOF), dense trajectory[6]) [55], and the specially designed features for RGB-D data: 3D MoSIFT-based features[7] (3D MoSIFT [50], 3D EMoSIF [9], 3D SMoSIFT [20]). All features are used with their default parameters. For traditional features, we can extract feature descriptors from RGB or depth videos. Therefore, the results are calculated by these features under two type data (RGB or RGB-D data). As shown in Fig. 10, the proposed feature also achieves the best performance. Besides, the performance of the dense trajectory method outperforms other traditional spatiotemporal features, but is still 4.4% lower than the proposed MFSK feature.

4.2.3 Comparison with Other Methods

Here, the performances are executed by two settings: MFSK+BoVW and MFSK+BoVW+TS, where TS denotes temporal scales. First, we compare with the currently published papers. The results are shown in Table 2. One can see that our method consistently outperforms the state-of-the-art methods (with an improvement of about 2.31% on average), and our method is very robust to scale, translation, partial occlusions.

Then, we also compare with the results of all top 14 results on CGD data in Table 6. Those top 14 results are derived from [18]. This table shows that our method is comparable to the best performance of team "Alfnie". More importantly, our method is more robust than "Alfnie" on

4. http://vision.ucsd.edu/ pdollar/toolbox/doc/
5. http://www.di.ens.fr/ laptev/download.html
6. http://lear.inrialpes.fr/people/wang/dense_trajectories/
7. https://mloss.org/software/view/499/

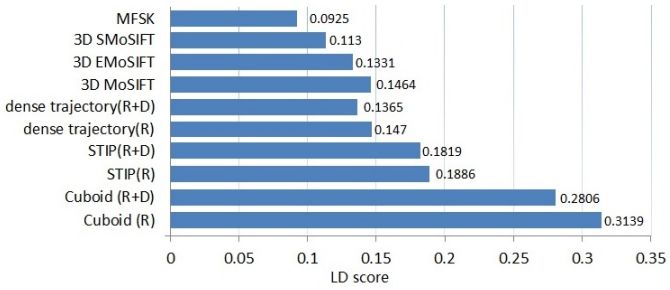


Fig. 10. The performances of different spatiotemporal features. It can be seen that MFSK achieves the best performance. More exactly, the order of performance decrease is MFSK > 3D SMoSIFT > 3D E-MoSIFT > dense trajectory(R+D) > 3D MoSIFT > dense trajectory(R) > STIP(R+D) > STIP(R) > Cuboid(R+D) > Cuboid(R).

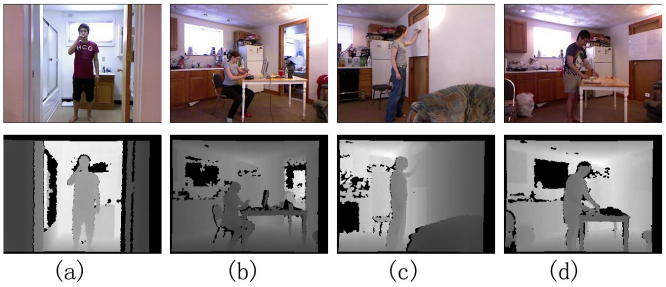


Fig. 11. Sample frames from different actions selected from CAD-60 dataset: (a) Brushing, (b) Opening pill container, (c) Writing on whiteboard, (d) Cooking (chopping).

challenging CGD subsets, such as $tran$ and $scaled$ data. For examples, the performances of "Alfine1" are decreased by 6.75% for $trans$ data and 9.31% for $scaled$ data, while our performances of "MFSK+BoVW+TS" are only decreased by 2.13% for $trans$ data, and only 0.15% for $scaled$ data, respectively.

4.3 Running Time Analysis

We randomly selected a sample from CGD and tested the average time with c++ programs and opencv library on a standard personal computer (CPU: Intel(R) Core(TM) i7-4790 @3.6 GHz, RAM: 24 GB). The average time of the MFSK feature is about 98 ms/f (including keypoints detection, feature description calculation and feature saving in the disk) without any optimization of the code. Moreover, we also test the calculation time among different descriptions: HOGHOF, MBH and 3D SMoSIFT. The running time of HOGHOF and MBH is very similar (about 18 ms/f) while 3D SMoSIFT costs more time (about 32 ms/f). However, as shown in Fig 9, 3D SMoSIFT can get better performances when these three parts are tested separately.

4.4 Experimental Results on Other RGB-D Datasets

Although the proposed MFSK feature has gained promising performances on one-shot learning gesture recognition from RGB-D data, it can also be evaluated on other RGB-D datasets. Here, we evaluate the proposed feature on Cornell Activity Dataset-60 (CAD-60) [59] and MSR Daily Activity 3D dataset [60]. Some samples from these two datasets

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2513479, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 6
Our method is compared with the results of all the top 14 results on the validation, final, untranslated, translated and scaled data of CGD. Our results are comparable to top 2 state-of-the-art methods on regular data ($valid$, $final1$, $final2$ and $utran$ data), and get the best performances on challenging data ($trans$ and $scaled$ data).

Name	$valid$	$final1$	$final2$	$utran$	$trans$	$scaled$
Alfnie1	0.1426	0.0996	0.0915	0.2316	0.2255	0.2573
Alfnie2	**0.0995**	**0.0734**	**0.0710**	**0.1635**	0.2310	0.2566
BalazsGodeny	0.2714	0.2314	0.2679	0.4347	0.5636	0.5526
HITCS	0.3245	0.2825	0.2008	0.4743	0.6640	0.6066
Immortals	0.2488	0.1847	0.1853	0.3594	0.3962	0.4152
Joewan	0.1824	0.1680	0.1448	0.2623	0.2612	0.2913
Manavender	0.2559	0.2164	0.1925	0.3644	0.4252	0.4358
OneMillionMonkeys	0.2875	0.1685	0.1819	0.3633	0.4961	0.5552
Pennect	0.1797	0.1652	0.1231	0.2589	0.4888	0.4068
SkyNet	0.2825	0.2330	0.1841	0.3901	0.4693	0.4771
TurtleTamers	0.2084	0.1702	0.1098	0.2896	0.5993	0.5296
Vigilant	0.3090	0.2809	0.2235	0.3817	0.5173	0.5067
WayneZhang	0.2819	0.2303	0.1608	0.3387	0.6278	0.5834
XiaoZhuWudi	0.2930	0.2564	0.2607	0.3962	0.6986	0.6897
Zonga	0.2714	0.2303	0.2191	0.4163	0.4905	0.5776
Ours(MFSK+BoVW)	0.1270	0.1395	0.0925	0.2390	**0.2120**	**0.2375**
Ours(MFSK+BoVW+TS)	0.1242	0.1326	0.0900	0.2315	**0.2102**	**0.2300**

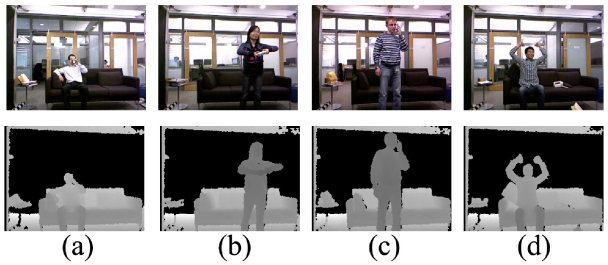


Fig. 12. Sample frames from different actions selected from MSR Daily Activity 3D Dataset: (a) Drink, (b) Play game, (c) Call cellphone, (d) Cheer up.

TABLE 7
Comparison with state-of-the-art on the CAD-60 dataset.

Approach	Precision(%)	Recall(%)	F_1 score(%)
Sung et al. 2012 [59]	67.9	55.5	61.08
Wan et al. 2014 [20]	74.8	65.8	70.01
Ni et al. 2013 [64]	75.9	69.5	72.56
Gupta et al. 2013 [65]	78.1	75.4	76.73
Zhang and Tian 2012 [66]	86.0	84.0	84.99
Ours (MFSK+BoVW)	87.1	83.8	85.42
Zhu et al. 2014 [61]	93.2	84.6	88.69
Parisi et al. 2015 [62]	91.9	90.2	91.47
Faria et al. 2014 [63]	91.1	91.9	91.50

are shown in Fig. 11 and Fig. 12, respectively. We can see that these two datasets are more challenging for the noise backgrounds (i.e. moving background, moving subjects).

In order to eliminate the noise effect and detect more accurate keypoints, we first found the local bounding box of the person based on skeletal information provided by these two datasets and then computed MFSK features within that bounding box.

4.4.1 Experiments on CAD-60

CAD-60 has five different environments: office, kitchen, bedroom, bathroom and living room. Three to four common activities were identified for each location, giving a total of 12 unique activities and a random action. This dataset is collected from 4 different people (2 males and 2 females).

Here, we test the proposed feature under two experimental settings: leave-one-out cross validation setting and one-shot-learning setting. Besides, the precision, recall and F_1 score are used as the evaluation criteria and the performances by different methods are ranked by the F_1 score. The F_1 score is the harmonic mean of precision and recall. It can be computed as:

$$F_1 = 2 \times precision \times recall/(precision + recall) \quad (6)$$

Leave-One-Out Cross Validation Setting Following the experiments in [20], [59] , we employed leave-one-out cross validation to test each person's data, which means the model was trained on three of the four people from whom data and tested on the fourth. In Table 7, we show a comparison of our results with the state-of-the-art methods. It shows that our model can obtain 87.1% precision, 83.8% recall and 85.42% F_1 score which indicates the proposed method exhibits a promising result. That is because: (1) we just used RGB-D data while other state-of-the-art methods [61], [62], [63] used skeleton information which is more easier to locate the human body. For examples, Faria et al. [63] used raw depth images to estimate their own skeleton model and then a dynamic Bayesian Mixture Model was used to classify motion relations between body poses. (2) We used the simple NN classifier. If nonlinear Support Vector Machine (SVM) is used, the performances may be improved further.

One-Shot Learning Setting Similar to the settings of CGD dataset, we select one sample per each class used for training and the left samples for testing. Owing to the varied movements of the random activity, it is almost impossible to recognize correctly the random activity if only one sample from the random activity is used in training stage. Therefore, we only evaluate the proposed method on the 12 unique activities. We randomly repeated the experiments five times

TABLE 8
Comparisons between the MFSK and individual feature descriptors on the CAD-60 dataset using one-shot learning settings.

Location	Activity	HOGHOF			MBH			3D SMoSIFT			MFSK		
		precision	recall	F_1 score	precision	recall	F_1 score	precision	recall	F_1 score	precision	recall	F_1 score
bathroom	brushing teeth	90	40	54	53.3	26.7	33.3	70	40	47.4	100	53.3	68
	rinsing mouth with water	90	100	94.3	77	93.3	81.5	90	100	94.3	95	85.6	97.1
	wearing contact lenses	83.1	97.1	89.5	80.9	91.4	85.2	83.2	94.3	87.9	85.6	100	92.2
	Average	87.7	79	79.3	70.4	70.5	66.7	81.1	78.1	76.5	93.5	84.4	85.8
bedroom	drinking water	36.9	73.3	47	27	40	32.1	15.7	33.3	21	43.3	80	53.7
	opening pill container	80.9	61.8	69.2	73.7	63.6	66.3	72	50.9	59.2	82.3	58.2	67.3
	talking on the phone	61.3	46.7	49.3	29	26.7	22.3	21.8	33.3	25	14.7	20	16.7
	Average	59.7	60.6	55.2	43.2	43.4	40.2	36.5	39.2	35.1	46.7	52.7	45.9
kitchen	cooking (chopping)	59.1	60	53.3	39	40	37.3	38	26.7	28	48.3	46.7	46.8
	cooking (stirring)	84.7	50.9	59.9	63.8	52.7	55.4	64.1	54.5	56.5	78	63.6	68.5
	drinking water	30.7	53.3	37.3	37.3	40	38.3	33.5	26.7	25.7	31	53.3	37.8
	opening pill container	31.7	33.3	30.7	31.9	46.7	36.8	27.6	46.7	33.3	26.9	46.7	33.6
	Average	51.5	49.4	45.3	43	44.8	42	40.8	38.6	35.9	46.1	52.6	46.6
living room	drinking water	47.6	80	56.1	33.7	66.7	43.8	26.7	33.3	26	34.5	60	43.1
	relaxing on couch	5	6.7	5.7	14.7	26.7	18.9	12.4	26.7	16.9	36.7	33.3	30.3
	talking on couch	23.3	26.7	24.8	20	6.7	10	33.3	26.7	29.3	53.3	40	45.3
	talking on the phone	25	26.7	25.1	10	6.7	8	11.4	26.7	16	19	20	18.7
	Average	25.2	35	27.9	19.6	26.7	20.2	21	28.3	22.1	35.9	38.3	34.4
office	drinking water	50.6	60	48.4	55.3	60	55.8	30.5	46.7	33.7	46.1	53.3	40.9
	talking on the phone	40	20	26	10	6.7	8	6.7	6.7	6.7	35	26.7	27.1
	working on computer	100	66.7	80	100	86.7	92	38	26.7	28	100	80	88
	writing on whiteboard	41.7	66.7	50.1	58.3	86.7	68.8	17.1	40	24	70.3	80	68.8
	Average	58.1	53.3	51.1	55.9	60	56.1	23.1	30	23.1	62.9	60	56.2
Overall Average		**56.4**	**55.5**	**51.8**	**46.4**	**49.1**	**45**	**40.5**	**42.8**	**38.5**	**57**	**57.6**	**53.8**

TABLE 9
Comparison on the MSR Daily Activity 3D dataset.

Approach	Average accuracy(%)
Wang et al. 2012 [60]	67.9
Oreifej et al. 2013 [67]	74.8
Ming et al. 2012 [50]	75.9
Li and Ling et al. 2013 [68]	78.1
Wan et al. 2013 [9]	86.0
Wan et al. 2014 [20]	93.2
Ours (MFSK+BoVW)	95.7

and calculated the average precision and recall. As shown in Table 8, the MFSK feature can obtain 57% precision, 57.6% recall and 53.8% F_1 score while the performances of other features is at least 2% lower than the proposed feature in F_1 score measure.

4.4.2 Experiments on MSR Daily Activity 3D Dataset

The dataset includes 16 activities: drink, eat, read book, call cellphone, write on a paper, use laptop, use vacuum cleaner, cheer up, sit still, toss paper, play game, lie down on sofa, walk, play guitar, stand up, and sit down. There are ten subjects. Each subject performs each action twice (standing or sitting). The total number of the activity samples is 320.

For the leave-one-out cross validation setting, we evaluate the proposed feature and the experimental results are shown in Table 9. The proposed feature outperforms other methods, demonstrating that the proposed feature is suitable for normal RGB-D video-based recognition.

In one-shot learning setting, it is similar to the experiments on CAD-60. The only difference is that we randomly selected two samples (standing and sitting) of each action per subject in the training stage. That is because each action is executed twice: standing and sitting positions. As shown in Table 10, the proposed feature outperforms other descriptors and achieves 41.2% accuracy, which is 4.5% higher than the 3D SMoSIFT.

TABLE 10
Descriptor comparisons in the MFSK feature on the MSR Daily Activity 3D dataset under one-shot learning seeting.

	3D SMoSIFT	HOGHOF	MBH	MFSK
accuracy (%)	36.7	32.1	27.3	41.2

5 Conclusion

We have thoroughly reviewed the research on one-shot learning gesture recognition from RGB-D data. We have analyzed the great challenges, and pointed out some future research directions. Then, we proposed the MFSK feature, which is robust and invariant to scale, rotation and partial occlusions. To further improve the recognition performance, we have presented a method to artificially augment the training samples, based on building temporal scales, which are beneficial for recognizing gestures with different speeds. We have also evaluated the proposed method on CGD and another two RGB-D datasets. Experimentally the proposed approach has achieved very promising results under either one-shot learning or leave-one-out cross validation.

Acknowledgment

This work was supported by the Chinese National Natural Science Foundation Projects ♯61203267, ♯61375037, ♯61473291, ♯61572501, ♯61502491, National Science and Technology Support Program Project ♯2013BAK02B01, Chinese Academy of Sciences Project No. KGZD-EW-102-2, and AuthenMetric R&D Funds.

References

[1] S. Mitra and T. Acharya, "Gesture recognition: A survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 3, pp. 311–324, 2007.

[2] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, "Sign language recognition using convolutional neural networks," in *European Conference on Computer Vision, Workshop*, 2014.

[3] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015.

[4] K. Qian, J. Niu, and H. Yang, "Developing a gesture based remote human-robot interaction system using kinect," *International Journal of Smart Home*, vol. 7, no. 4, 2013.

[5] F. Porikli, F. Brémond, S. L. Dockstader, J. Ferryman, A. Hoogs, B. C. Lovell, S. Pankanti, B. Rinner, P. Tu, and P. L. Venetianer, "Video surveillance: past, present, and now the future," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 190–198, 2013.

[6] S. Reifinger, F. Wallhoff, M. Ablassmeier, T. Poitschke, and G. Rigoll, "Static and dynamic hand-gesture recognition for augmented reality applications," in *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments*. Springer, 2007, pp. 728–737.

[7] M.-C. Roh, B. Christmas, J. Kittler, and S.-W. Lee, "Gesture spotting for low-resolution sports video annotation," *Pattern Recognition*, vol. 41, no. 3, pp. 1124–1137, 2008.

[8] T. Pfister, J. Charles, and A. Zisserman, "Domain-adaptive discriminative one-shot learning of gestures," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 814–829.

[9] J. Wan, Q. Ruan, W. Li, and S. Deng, "One-shot learning gesture recognition from rgb-d data using bag of features," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2549–2582, 2013.

[10] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden markov model," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1992, pp. 379–385.

[11] D. Wu, F. Zhu, and L. Shao, "One shot learning gesture recognition from rgbd images," in *Computer Vision and Pattern Recognition Workshops, 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 7–12.

[12] A. Farhadi, D. Forsyth, and R. White, "Transfer learning in sign language," in *Computer Vision and Pattern Recognition, IEEE Conference on*. IEEE, 2007, pp. 1–8.

[13] I. Guyon, V. Athitsos, P. Jangyodsuk, H. J. Escalante, and B. Hamner, "Results and analysis of the chalearn gesture challenge 2012," in *Advances in Depth Image Analysis and Applications*. Springer, 2013, pp. 186–204.

[14] D. Wu and L. Shao, "Silhouette analysis-based action recognition via exploiting human poses," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 23, no. 2, pp. 236–243, 2013.

[15] T. Xia, D. Tao, T. Mei, and Y. Zhang, "Multiview spectral embedding," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, no. 6, pp. 1438–1446, 2010.

[16] Y. M. Lui, "Human gesture recognition on product manifolds," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 3297–3321, 2012.

[17] N. A. Goussies, S. Ubalde, and M. Mejail, "Transfer learning decision forests for gesture recognition," *Journal of Machine Learning Research*, vol. 15, pp. 3667–3690, 2014.

[18] I. Guyon, V. Athitsos, P. Jangyodsuk, and H. J. Escalante, "The chalearn gesture dataset (cgd 2011)," *Machine Vision and Applications*, vol. 25, no. 8, pp. 1929–1951, 2014.

[19] J. Konečnỳ and M. Hagara, "One-shot-learning gesture recognition using hog-hof features," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2513–2532, 2014.

[20] J. Wan, Q. Ruan, W. Li, G. An, and R. Zhao, "3d smosift: three-dimensional sparse motion scale invariant feature transform for activity recognition from rgb-d videos," *Journal of Electronic Imaging*, vol. 23, no. 2, pp. 023 017–023 017, 2014.

[21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[22] A. Kläser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *British Machine Vision Conference*. British Machine Vision Association, 2008, pp. 995–1004.

[23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[24] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[25] B. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.

[26] M. R. Malgireddy, I. Inwogu, and V. Govindaraju, "A temporal bayesian model for classifying, detecting and localizing activities in video sequences," in *Computer Vision and Pattern Recognition Workshops, 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 43–48.

[27] M. R. Malgireddy, I. Nwogu, and V. Govindaraju, "Language-motivated approaches to action recognition," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2189–2212, 2013.

[28] H. J. Escalante, I. Guyon, V. Athitsos, P. Jangyodsuk, and J. Wan, "Principal motion components for gesture recognition using a single-example," *arXiv preprint arXiv:1310.4822*, 2013.

[29] B. Liang and L. Zheng, "Gesture recognition from one example using depth images," *Lecture Notes on Software Engineering*, vol. 1, no. 4, 2013.

[30] A. Hernández-Vela, M. Á. Bautista, X. Perez-Sala, V. Ponce-López, S. Escalera, X. Baró, O. Pujol, and C. Angulo, "Probability-based dynamic time warping and bag-of-visual-and-depth-words for human gesture recognition in rgb-d," *Pattern Recognition Letters*, vol. 50, pp. 112–121, 2014.

[31] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 4, pp. 594–611, 2006.

[32] E. G. Miller, N. E. Matsakis, P. Viola *et al.*, "Learning from one example through shared densities on transforms," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 1. IEEE, 2000, pp. 464–471.

[33] R. Krishnan and S. Sarkar, "Conditional distance based matching for one-shot gesture recognition," *Pattern Recognition*, vol. 48, no. 4, pp. 1298–1310, 2015.

[34] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[35] S. R. Fanello, I. Gori, G. Metta, and F. Odone, "One-shot learning for real-time action recognition," in *Pattern recognition and image analysis*. Springer, 2013, pp. 31–40.

[36] S. R. Fanello, I. Gori, G. Metta, and F. Odone, "Keep it simple and sparse: Real-time action recognition," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2617–2640, 2013.

[37] J. Wan, V. Athitsos, P. Jangyodsuk, H. Escalante, Q. Ruan, and I. Guyon, "Csmmi: Class-specific maximization of mutual information for action and gesture recognition," *Image Processing, IEEE Transactions on*, vol. 23, no. 7, pp. 3152–3165, 2014.

[38] F. Jiang, s. Zhang, S. Wu, S. Deng, G. Yang, and D. Zhao, "Multi-layered gesture recognition with kinect," *The Journal of Machine Learning Research*, vol. 16, pp. 227–254, 2015.

[39] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 3, pp. 257–267, 2001.

[40] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *Information Theory, IEEE Transactions on*, vol. 13, no. 2, pp. 260–269, 1967.

[41] D. Kim, J. Song, and D. Kim, "Simultaneous gesture segmentation and recognition based on forward spotting accumulative hmms," *Pattern Recognition*, vol. 40, no. 11, pp. 3012–3026, 2007.

[42] M. Elmezain, A. Al-Hamadi, and B. Michaelis, "A robust method for hand gesture segmentation and recognition using forward spotting scheme in conditional random fields," in *Pattern Recognition, 2010 20th International Conference on*. IEEE, 2010, pp. 3850–3853.

[43] J. D. Mcauliffe and D. M. Blei, "Supervised topic models," in *Advances in neural information processing systems*, 2008, pp. 121–128.

[44] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in neural information processing systems*, 2006, pp. 801–808.

[45] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent k-svd," in *Computer Vision and Pattern Recognition, 2011 IEEE Conference on*. IEEE, 2011, pp. 1697–1704.

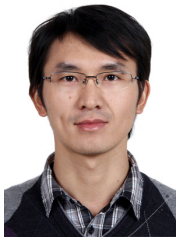
[46] V. N. Vapnik and V. Vapnik, *Statistical learning theory*. Wiley New York, 1998, vol. 1.

[47] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine learning*, vol. 46, no. 1-3, pp. 389–422, 2002.

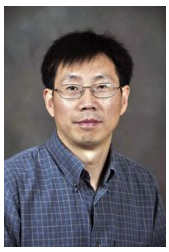
[48] H. Ney and S. Ortmanns, "Dynamic programming search for continuous speech recognition," *Signal Processing Magazine, IEEE*, vol. 16, no. 5, pp. 64–83, 1999.

[49] R. Yang, S. Sarkar, and B. Loeding, "Handling movement epenthesis and hand segmentation ambiguities in continuous sign lan-

guage recognition using nested dynamic programming," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 3, pp. 462–477, 2010.
[50] Y. Ming, Q. Ruan, and A. Hauptmann, "Activity recognition from rgb-d camera with 3d local spatio-temporal features," in *Proceedings of IEEE International Conference on Multimedia and Expo*, 2012, pp. 344–349.
[51] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–511.
[52] G. Bradski, *Dr. Dobb's Journal of Software Tools*, 2000.
[53] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Computer Vision and Pattern Recognition, IEEE Conference on*. IEEE, 2008, pp. 1–8.
[54] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *BMVC 2009-British Machine Vision Conference*. BMVA Press, 2009, pp. 124–1.
[55] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 60–79, 2013.
[56] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 428–441.
[57] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proceedings of IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005, pp. 65–72.
[58] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2, pp. 107–123, 2005.
[59] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from rgbd images," in *Robotics and Automation, IEEE Conference on*, 2012, pp. 842–849.
[60] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2012, pp. 1290–1297.
[61] Y. Zhu, W. Chen, and G. Guo, "Evaluating spatiotemporal interest point features for depth-based action recognition," *Image and Vision Computing*, vol. 32, no. 8, pp. 453–464, 2014.
[62] G. Parisi, C. Weber, and S. Wermter, "Self-organizing neural integration of pose-motion features for human action recognition," *Frontier in Neurobotics*, vol. 9, no. 3, 2015.
[63] D. R. Faria, C. Premebida, and U. Nunes, "A probabilistic approach for human everyday activities recognition using body motion from rgb-d images," in *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*. IEEE, 2014, pp. 732–737.
[64] B. Ni, Y. Pei, P. Moulin, and S. Yan, "Multilevel depth and image fusion for human activity detection," *Cybernetics, IEEE Transactions on*, vol. 43, no. 5, pp. 1383–1394, 2013.
[65] R. Gupta, A. Y.-S. Chia, and D. Rajan, "Human activities recognition using depth images," in *Proceedings of the 21st ACM international conference on Multimedia*. ACM, 2013, pp. 283–292.
[66] C. Zhang and Y. Tian, "Rgb-d camera-based daily living activity recognition," *Journal of Computer Vision and Image Processing*, vol. 2, no. 4, p. 12, 2012.
[67] O. Oreifej and Z. Liu, "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences," in *Computer Vision and Pattern Recognition, 2013 IEEE Conference on*, 2013, pp. 716–723.
[68] L. Liu and L. Shao, "Learning discriminative representations from rgb-d video data," in *Proc. Int. Joint Conf. on Artificial Intelligence*, 2013, pp. 1493–1500.



Jun Wan received the BS degree from China University of Geosciences, Beijing, China in 2008 and the Ph.D. degree in Institute of Information Science, Beijing Jiaotong University, Beijing, China, in 2015. Since January 2015, he has been an Assistant Professor in National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Science (CASIA). He was a recipient of the 2012 ChaLearn One-Shot-Learning Gesture Challenge Award, sponsored by Microsoft, at ICPR 2012. He was also a recipient of the 2013, 2014 Best Paper Award from the Institute of Information Science, Beijing Jiaotong University. His main research interests include computer vision, machine learning, especially for gesture and action recognition, face attribution recognition.



Guodong Guo (M'07-SM'07) received the B.E. degree in automation from Tsinghua University, Beijing, China, the Ph.D. degree in pattern recognition and intelligent control from Chinese Academy of Sciences, Beijing, China, and the Ph.D. degree in computer science from University of Wisconsin-Madison, Madison, WI, USA, in 2006. He is an Associate Professor with the Department of Computer Science and Electrical Engineering, West Virginia University (WVU), Morgantown, WV, USA. In the past, he visited and worked in several places, including INRIA, Sophia Antipolis, France; Ritsumeikan University, Kyoto, Japan; Microsoft Research, Beijing, China; and North Carolina Central University. He authored a book, *Face, Expression, and Iris Recognition Using Learning-based Approaches* (2008), co-edited a book, *Support Vector Machines Applications* (2014), and published over 60 technical papers. His research interests include computer vision, machine learning, and multimedia. He received the North Carolina State Award for Excellence in Innovation in 2008, Outstanding Researcher (2013-2014) at CEMR, WVU, and New Researcher of the Year (2010-2011) at CEMR, WVU. He was selected the "People's Hero of the Week" by BSJB under Minority Media and Telecommunications Council (MMTC) on July 29, 2013. Two of his papers were selected as "The Best of FG'13" and "The Best of FG'15", respectively.

Stan Z. Li received the BEng degree from Hunan University, China, the MEng degree from National University of Defense Technology, China, and the PhD degree from Surrey University, United Kingdom. He is currently a professor and the director of Center for Biometrics and Security Research (CBSR), Institute of Automation, Chinese Academy of Sciences (CASIA). He was at Microsoft Research Asia as a researcher from 2000 to 2004. Prior to that, he was an associate professor at Nanyang Technological University, Singapore. His research interests include pattern recognition and machine learning, image and vision processing, face recognition, biometrics, and intelligent video surveillance. He has published more than 200 papers in international journals and conferences, and authored and edited eight books. He was an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence and is acting as the editor-in-chief for the Encyclopedia of Biometrics. He served as a program cochair for the International Conference on Biometrics 2007 and 2009, and has been involved in organizing other international conferences and workshops in the fields of his research interest. He was elevated to IEEE fellow for his contributions to the fields of face recognition, pattern recognition and computer vision and he is a member of the IEEE Computer Society.