Journal of Electronic Imaging

SPIEDigitalLibrary.org/jei

3D SMoSIFT: three-dimensional sparse motion scale invariant feature transform for activity recognition from RGB-D videos

Jun Wan
Qiuqi Ruan
Wei Li
Gaoyun An
Ruizhen Zhao

3D SMoSIFT: three-dimensional sparse motion scale invariant feature transform for activity recognition from RGB-D videos

Jun Wan,[a,b,*] Qiuqi Ruan,[a,b] Wei Li,[a,b] Gaoyun An,[a,b] and Ruizhen Zhao[a,b]
[a]Beijing Jiaotong University, Institute of Information Science, Beijing 100044, China
[b]Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China

Abstract. Human activity recognition based on RGB-D data has received more attention in recent years. We propose a spatiotemporal feature named three-dimensional (3D) sparse motion scale-invariant feature transform (SIFT) from RGB-D data for activity recognition. First, we build pyramids as scale space for each RGB and depth frame, and then use Shi-Tomasi corner detector and sparse optical flow to quickly detect and track robust keypoints around the motion pattern in the scale space. Subsequently, local patches around keypoints, which are extracted from RGB-D data, are used to build 3D gradient and motion spaces. Then SIFT-like descriptors are calculated on both 3D spaces, respectively. The proposed feature is invariant to scale, transition, and partial occlusions. More importantly, the running time of the proposed feature is fast so that it is well-suited for real-time applications. We have evaluated the proposed feature under a bag of words model on three public RGB-D datasets: one-shot learning Chalearn Gesture Dataset, Cornell Activity Dataset-60, and MSR Daily Activity 3D dataset. Experimental results show that the proposed feature outperforms other spatiotemporal features and are comparative to other state-of-the-art approaches, even though there is only one training sample for each class. *© 2014 SPIE and IS&T* [DOI: 10.1117/1.JEI.23.2.023017]

Keywords: three-dimensional sparse motion scale-invariant feature transform; bag of words model; spatiotemporal feature; optical flow; RGB-D data.

Paper 13651 received Nov. 19, 2013; revised manuscript received Mar. 4, 2014; accepted for publication Mar. 13, 2014; published online Apr. 8, 2014.

1 Introduction

Vision-based human activity (e.g., gesture or action) recognition has been an active research topic in computer vision over the last decade.[1–5] However, the sensing devices used to record RGB videos can be capable of only color information and are still restricted in complex scenes, such as occlusions, clutter, and illumination changes. As human activities are, in essence, three-dimensional (3D), information loss in the depth channel could cause significant degradation of the representation and discrimination of capability for these feature representations. Fortunately, the Kinect™ camera launched by Microsoft has revolutionized the field of computer vision[6,7] by making available low-cost 3D cameras recording both RGB and depth data, using a structured light infrared sensor. Among computer vision areas, human activity recognition based on RGB-D data has gained a lot of attention.[8–14] Two significant aspects arise to explore more informative data from RGB-D videos. One way is to select 3D points of the joints from a skeleton detector.[15] For instance, an actionlet ensemble model[9] is proposed to estimate 3D joint positions and calculate the local occupancy pattern feature for action recognition. In order to figure up the human pose feature, the authors used the skeleton model, which can move with 15 joints, to calculate 3D coordinate and orientation of each joint.[10] However, the skeleton model is not stable enough to capture 3D positions of tracked joints if serious occlusions occur, which will lead to increased intraclass variations in actions.

Another approach is to adopt conventional color-based (or gray-scale-based) methods to extract spatiotemporal interest points (STIPs)[16,17] for RGB-D or only depth sequences. Ni et al.[8] proposed a depth-layered multichannel STIPs framework, which divides STIPs into several depth-layered channels and then STIPs within different channels are pooled independently. Hernández-Vela et al.[11] used Harris3D detector[16] to detect keypoints on RGB and depth sequences, respectively. Then histogram of oriented gradients (HOG) and histogram of optical flow (HOF) features around the keypoint volume are extracted on RGB data and 3D point histograms are extracted from depth data. Finally, the authors[11] fused the extracted features and used the bag of words (BoW) model[18] to achieve gesture recognition. Ming et al.[13] proposed a 3D motion scale-invariant feature transform (3D MoSIFT) feature, which fuses RGB data and depth information to compute SIFT-like descriptors. However, 3D MoSIFT is sensitive to detect the robust keypoint around the body motion regions if some slight movements occur in the background. Later, Wan et al.[14] extended the works of 3D MoSIFT and developed a new feature named 3D enhanced MoSIFT (3D EMoSIFT). 3D EMoSIFT is more robust to detect keypoints and achieves better performance. Nevertheless, owing to build Gaussian pyramid and dense optical flow pyramid, both 3D MoSIFT and 3D EMoSIFT are time-consuming (~900 ms/f for 320×240 images[14]) to detect keypoints.

*Address all correspondence to: Jun Wan, E-mail: 09112088@bjtu.edu.cn

Inspired by previous works,[13,14] we propose a new feature named 3D SMoSIFT in this paper. The new feature has some more impressive aspects that 3D MoSIFT and 3D EMoSIFT do not have. The main contributions of this work are fivefold.

- For 3D (E)MoSIFT feature, it first builds Gaussian pyramid and difference of Gaussian (DoG) pyramid to find local extrema points, which is the same as SIFT algorithm.[19] As each point in DoG pyramid should be compared to its 26 neighbors in 3×3 regions at the current and adjacent scales, it will cost much time to detect all keypoints. However, we propose a novel method to detect keypoints via simple corner detector[20] and the tracking technique.[21] Due to only some corner points being considered, our method for keypoint detection is sparse. That is why the new feature is called 3D sparse MoSIFT (or 3D SMoSIFT).
- We know that 3D (E)MoSIFT consists of gradient and motion features. But from experimental results of 3D EMoSIFT,[14] the authors proved that the performance of motion features is less than that of gradient features. That is probably because depth frames are often contaminated with undefined depth points, which appear in the sequence as spatially and temporally discontinuous black regions.[22] The missing depth points may affect the motion feature. So we extract the local patches around keypoint regions and smooth them using Gaussian filter. Besides, the 3D modified motion space is proposed to calculate motion features.
- The new feature is invariant to scale, transition, and partial occlusions.
- Compared with 3D (E)MoSIFT, which is time-consuming, the new feature can be applied in real-time applications.
- The proposed feature has obtained high performances on some human activity datasets, such as Chalearn Gesture Dataset, Cornell Activity Dataset-60, and MSR Daily Activity 3D Dataset.

The rest of this paper is organized as follows. Section 2 reviews the background of both local spatiotemporal features and BoW model. Then we describe 3D SMoSIFT in Sec. 3. Section 4 presents our experimental results. Finally, Sec. 5 concludes the paper and shows future works.

2 Related Work

2.1 *Local Spatiotemporal Features*

Modeling a human activity in a video sequence starts with a powerful video representation. A popular approach is to describe an action video with some kind of motion features. The motion features capture some key spatiotemporal patterns that characterize a particular class as well as discriminate it from other classes. Therefore, we describe some spatiotemporal features that represent state-of-the-art techniques on activity recognition tasks. Cuboid detector[16] depends on a set of linear filters for computing a response function of a video clip $V(x, y, t)$. The response function has the form $R = (V * g * h_{ev})^2 + (V * g * h_{od})^2$, where $g(x, y, \sigma)$ is a two-dimensional (2-D) Gaussian smoothing function applied in the spatial domain, and h_{ev} and h_{od} are a quadrature pair of one-dimensional Gabor filters applied in the temporal direction. These are defined as $h_{ev} = -\cos(2\pi tw)e^{-t^2/\tau^2}$ and $h_{od} = -\sin(2\pi tw)e^{-t^2/\tau^2}$, where $w = 4/\tau$. The parameters σ and τ roughly correspond to the spatial and temporal scales. And keypoints are detected at the local maxima of the response function. Then the video patches extracted around every keypoint are converted to a descriptor. Last, principal component analysis is used to project the feature vector to a lower-dimensional space.

Harris3D detector[17] is an extension of Harris detector.[23] The author computes a spatiotemporal second-moment matrix at each video point $\mu(\cdot; \sigma, \tau) = g(\cdot; s\sigma, s\tau) * \{\nabla L(\cdot; \sigma, \tau)[\nabla L(\cdot; \sigma, \tau)]^T\}$ using independent spatial and temporal scale values σ and τ, a separable Gaussian smoothing function g, and space-time gradients ∇L. The final locations of space-time interest points are given by the local positive spatiotemporal maxima of $H = \det(\mu) - k\,\mathrm{trace}^3(\mu)$, $H > 0$. Then two types of descriptors are calculated at each keypoint, which are HOG and HOF.

MoSIFT[24] is derived from SIFT algorithm[19] and optical flow.[21] First, a pair of Gaussian pyramids, L^t, L^{t+1}, is built for two successive frames at time t and $t+1$. The form of Gaussian pyramid is $L_{i,j} = G(\cdot, k^j\sigma) * L^I_{i,0}$, $0 \le i < n$, $0 \le j < s+3$, where n is the number of octaves and s is the number of intervals, $*$ is the convolution operation, $G(\cdot, k^j\sigma)$ is a Gaussian function, σ is the initial smoothing parameter, and $k = 2^{1/s}$.[19] So an optical flow pyramid can be built via L^t and L^{t+1}. Then DoG pyramid Df is calculated at time t, $Df_{i,j} = L^t_{i,j+1} - L^t_{i,j}, 0 \le i < n, 0 \le j < s+2$. And the local extreme points can be found in DoG pyramid (the same as SIFT algorithm). Next, the extreme points can only become keypoints if they have sufficient motion in optical flow pyramid. Finally, as the process of SIFT descriptor calculation, MoSIFT descriptors are computed from Gaussian pyramid and optical flow pyramid around keypoints regions, respectively.

3D MoSIFT feature[13] fuses RGB data and depth information to calculate feature descriptors. First, 3D MoSIFT adopts the same strategy in MoSIFT to detect keypoints. Then, 3D gradient and motion spaces are constructed from the regions around keypoints by fusing RGB-D data. In each 3D space, they map 3D space into three 2-D planes: xy plane, yz plane, and xz plane. Next, SIFT descriptors are calculated on each plane. Hence, 3D MoSIFT consists of six SIFT descriptors.

Later, Wan et al.[14] found that some futile points from the background or torso regions are detected as keypoints in both MoSIFT and 3D MoSIFT. Therefore, 3D EMoSIFT feature is proposed to reduce these futile points. First, interest points are detected by the same strategy of MoSIFT. Then, interest points will become keypoints when the depth value of these points has changed enough in depth changing pyramid. So some redundant points with slight motion will be filtered out. Compared with 3D MoSIFT, 3D EMoSIFT can detect less keypoints but capture more compact visual representations. For more details about the differences among three MoSIFT-based features, please refer to the works.[14]

2.2 *BoW Model*

After spatiotemporal feature extraction, a certain human activity is usually represented as a collection of codewords in a pretrained codebook. This is the well-known BoW

model, which has been adopted by many computer vision researchers.[11,16]

In the BoW model, a codebook is commonly learned by clustering (e.g., k-means). That means the codebook is denoted by clustering centers and each clustering center is treated as one codeword. Then each sample vector is allowed to be approximated by one codeword. For example, when vector quantization[25] is used, each vector is assigned to one codeword that is closest to it in terms of Euclidean distance. So each video can be represented by a histogram of the codebook. In the training stage, the histograms of training videos are used to train a support vector machine[26] or k-nearest neighbor classifier. The BoW model is illustrated in Fig. 1.

3 3D SMoSIFT

The proposed feature broadly consists of three stages. First, the scale place of every frame (including RGB and depth images) is constructed by pyramid representations. Second, keypoint detection is applied in different levels of scale space. Third, 3D gradient and modified motion spaces are constructed in local patches around keypoints, and then SIFT-like descriptors are calculated on both 3D spaces. Last, a short summary of the proposed feature is given. To more intuitively understand the proposed feature, we will use two pairs of consecutive frames (one pair of RGB frames, one pair of depth frames) as an example to illustrate every processing procedure.

3.1 *Pyramid Representation for RGB-D data*

For a given sample including two videos (an RGB video and a depth video), we can obtain a gray-scale image G_t (converted from RGB frame) and a depth image D_t at time t. (The depth values are normalized to [0 255] in depth videos.) Then one pyramid can be built from G_t or D_t via down-sampled way. 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 and (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, $P_G^t = \{G_t^1, G_t^2, \ldots, G_t^L\}$ and $P_D^t = \{D_t^1, D_t^2, \ldots, D_t^L\}$. For typical image sizes, $L = 2, 3, 4$. For instance, for an image I with 640×480, the pyramid with four levels consists of I^1, I^2, I^3, I^4, which are of sizes 640×480, 320×240, 160×120, and 80×60, respectively. If $L = 5$, I^5 is of size 40×30, which is small and makes no sense in most of the cases.

Figure 2 shows two pyramids P_G^t and P_G^{t+1} (or P_D^t and P_D^{t+1}) built from two consecutive gray-scale (or depth) frames at times t and $t+1$. The frames are of size 320×240. As shown in Fig. 2, 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.

3.2 *Keypoint Detection*

3.2.1 *Detection of initial interest points*

In Shi-Tomasi corner detector[20] algorithm, a 2×2 Hessian matrix H_p can be computed for every point p in an image I. H_p is a square matrix of second-order partial derivatives of I.

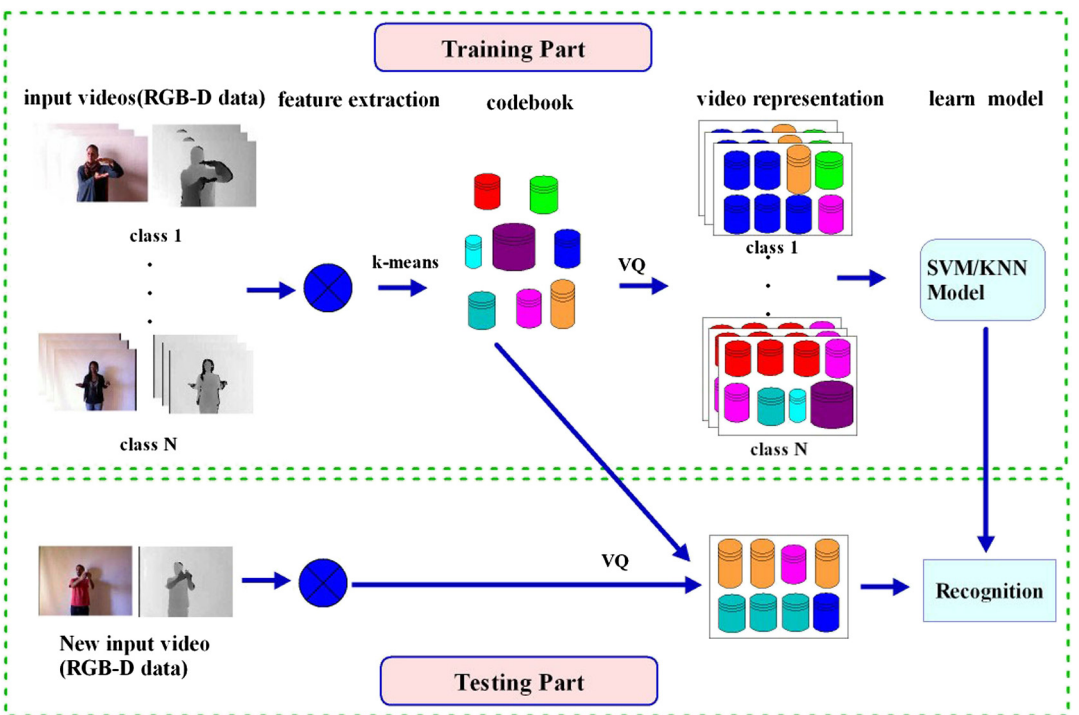

Fig. 1 An overview of the traditional bag of words model using RGB-D data.

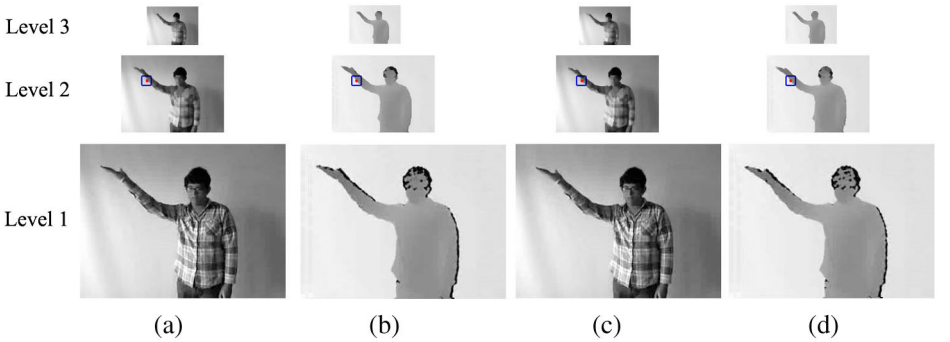

Fig. 2 Building four pyramids from two pairs of consecutive frames. (a) P_G^t at time t. (b) P_D^t at time t. (c) P_G^{t+1} at time $t+1$. (d) P_D^{t+1} at time $t+1$.

$$H_p = \begin{bmatrix} \frac{\partial^2 I}{\partial^2 x} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial^2 y} \end{bmatrix}_p \doteq \begin{bmatrix} a & b \\ b & c \end{bmatrix}_p, \tag{2}$$

where $a = (\partial^2 I)/(\partial^2 x)$, $b = (\partial^2 I)/(\partial x \partial y)$, and $c = (\partial^2 I)/(\partial^2 y)$; the partial derivatives a, b, and c are estimated by Sobel operator.

A point p is defined as a corner if its eigenvalues $\lambda_{p1}, \lambda_{p2}$ of H_p are larger than a threshold λ.

$$\min(\lambda_{p1}, \lambda_{p2}) > \lambda. \tag{3}$$

For the Hessian matrix H_p, its eigenvalues are $\lambda_{p1} = [(a+c) + \sqrt{(a-c)^2 + b^2}]/2$ and $\lambda_{p2} = [(a+c) - \sqrt{(a-c)^2 + b^2}]/2$ ($\lambda_{p1} \geq \lambda_{p2}$). Therefore, Eq. (3) can be written as

$$\lambda_{p2} = \frac{(a+c) - \sqrt{(a-c)^2 + b^2}}{2} > \lambda. \tag{4}$$

Lower values of λ allow us to detect more interest points. A commonly used value of λ is $\lambda = \alpha \times \max\{\lambda_{p_i 2}\}$, $p_i \in I$ in the literature[27] and α is equal to 0.001.

Because there are two frames (RGB and depth data) at time t, we can detect interest points in RGB or depth frame. That is, say interest points are detected either in the pyramid P_D^t [see Figs. 3(a) and 3(b)] or P_G^t [see Figs. 3(c) and 3(d)]. Here, we use Fig. 3(a) as an example to illustrate the interest point detection. We first detect interest points at different levels in P_D^t via Shi-Tomasi algorithm, and the detected points are labeled with green dots. We can see that green dots are detected around the body's mask, especially around motion regions (e.g., the right moving hand in Fig. 2). Besides, when an image is in a higher level of a pyramid, the detected points are fewer. That is because the image at a higher level is more smaller and some detailed information are missing. Intuitively, we want to detect interest points around motion regions and some futile points would be filtered out. So we will select a part of interest points as keypoints via tracking and filtering techniques in the next section.

3.2.2 *Keypoint detection via tracking and filtering*

At time t, we suppose an interest point $p = [\, p_x \quad p_y \,]^T$ on the image I^t is given. The goal is to find the location $q = p + v = [\, p_x + v_x \quad p_y + v_y \,]^T$ on the image I^{t+1} at time $t+1$. And the vector $v = [\, v_x \quad v_y \,]^T$ is the velocity of p, which is known as optical flow of p. We define the absolute velocity $|v| = \sqrt{v_x^2 + v_y^2}$. The velocity v is to minimize the residual function ϵ, and ϵ is defined as

$$\begin{aligned} \epsilon(v) &= \epsilon(v_x, v_y) \\ &= \sum_{x=p_x-N}^{x=p_x+N} \sum_{y=p_y-N}^{x=p_y+N} [I^t(x,y) - I^{t+1}(x+v_x, y+v_y)]^2, \end{aligned} \tag{5}$$

where ϵ is measured on a small widow of size $(2N+1) \times (2N+1)$. To optimize Eq. (5), the first derivative of ϵ is set to 0.

$$\frac{\partial \epsilon(v)}{\partial v} = [\, 0 \quad 0 \,]^T. \tag{6}$$

Then, the optimized optical flow vector is followed.

$$\begin{aligned} v = &\left(\sum_{x=p_x-N}^{p_x+N} \sum_{y=p_y-N}^{p_y+N} \begin{bmatrix} \frac{\partial^2 I^t}{\partial^2 x} & \frac{\partial^2 I^t}{\partial x \partial y} \\ \frac{\partial^2 I^t}{\partial x \partial y} & \frac{\partial^2 I^t}{\partial^2 y} \end{bmatrix} \right)^{-1} \\ &\times \left(\sum_{x=p_x-N}^{p_x+N} \sum_{y=p_y-N}^{p_y+N} \begin{bmatrix} (I^t(x,y) - I^{t+1}(x,y)) \frac{\partial I^t}{\partial x} \\ (I^t(x,y) - I^{t+1}(x,y)) \frac{\partial I^t}{\partial y} \end{bmatrix} \right), \end{aligned} \tag{7}$$

where the gradient information $(\partial I^t)/\partial x$ and $(\partial I^t)/(\partial y)$ can be calculated in both x and y directions by Sobel or Scharr operator.

This is sparse optical flow, which tracks only a few points in two consecutive frames.[28] For each interest point at time t, we can calculate its velocity at time $t+1$ via Eq. (7). Because both gray-scale and depth frames can be used, we have two ways to track initial interest points. The first way is shown in Figs. 3(a) and 3(d), where interest points located in P_G^t at time t are used to predict the next position in P_G^{t+1} at time $t+1$. Another way shown in Figs. 3(b) and 3(c) is to use P_D^t and P_D^{t+1} to track locations of interest points.

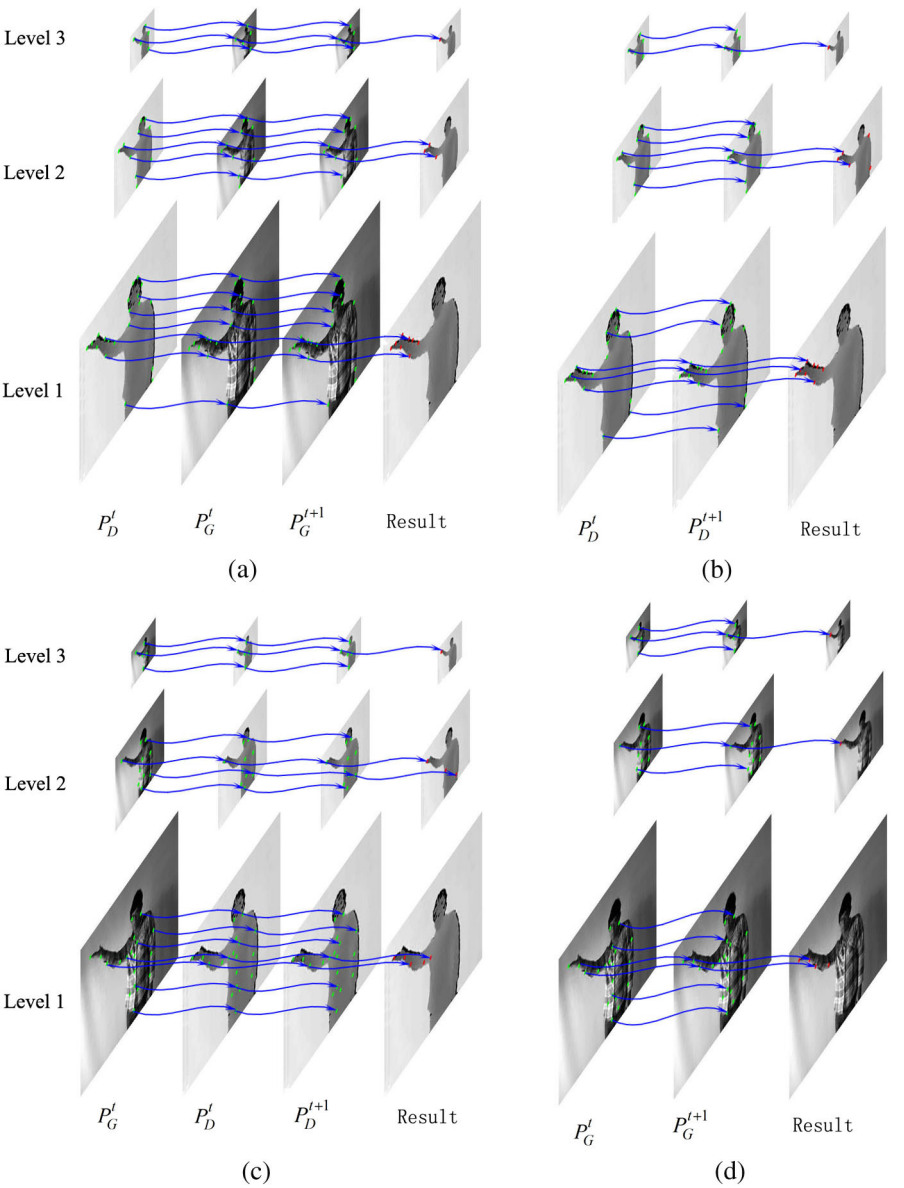

Fig. 3 The flow graph of keypoint detection from RGB-D data in the scale space via different types. The four types consist of two steps: initial interest point detection and interest point tracking. (a) SMoSIFT1: $P_D^t + \{P_G^t, P_G^{t+1}\}$. (b) SMoSIFT2: $P_D^t + \{P_D^t, P_D^{t+1}\}$. (c) SMoSIFT3: $P_G^t + \{P_D^t, P_D^{t+1}\}$. (d) SMoSIFT4: $P_G^t + \{P_G^t, P_G^{t+1}\}$, where the pyramid before "+" is used to detect initial interest points, and the pair of pyramids after "+" are used to track interest points. Initial interest points are labeled with green dots, while the keypoints are labeled with red dots.

Then we can calculate the absolute velocity of each interest point at different levels in the pyramid. Finally, we select some interest points around motion regions when these points satisfy the motion constraint. That is, when the absolute velocity $|v|$ of an interest point at the l'th level is larger than a given threshold τ^l, this point will become a keypoint. τ^l is defined as

$$\tau^l = \max\left[\underbrace{\max(\alpha|v_{\max}^l|, 0.5^{l-1}\beta)}_{\text{local constraint}}, \underbrace{\delta}_{\text{global constraint}}\right]$$
$$1 \le l \le L, 0 < \alpha < 1, \quad (8)$$

where $|v_{\max}^l|$ is the maximum value of absolute velocities of interest points at the l'th level in the pyramid. The parameter $\max(\alpha|v_{\max}^l|, 0.5^{l-1}\beta)$ determines the motion constraint at the l'th level. δ is the global parameters, which means the absolute velocity of each keypoint is not $<\delta$. From our experiments, 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.

As shown in Fig. 3, all the keypoints are labeled with red dots and four types of 3D SMoSIFT (SMoSIFT1, SMoSIFT2, SMoSIFT3, and SMoSIFT4) are generated.

3.3 *Feature Descriptor Calculation*

After keypoint detection, a feature descriptor can be computed from the local patch around a keypoint, such as Cuboid descriptor,[16] HOG and HOF descriptors,[17] or SIFT-like descriptors.[13,14,24] Here, we calculate SIFT-like

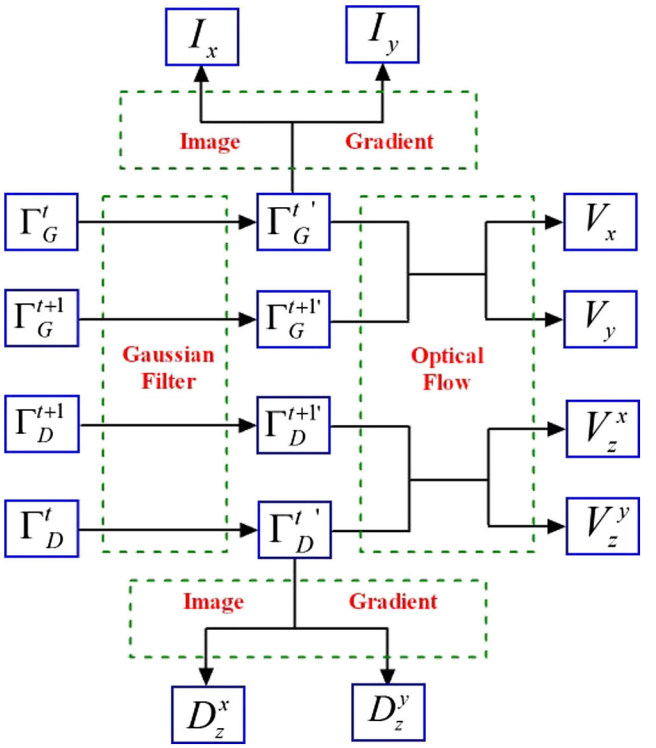

Fig. 4 The preprocessing portion of the descriptor calculation from the extracted local patches. Input images: Γ_G^t, Γ_D^t, Γ_G^{t+1}, and Γ_D^{t+1}. Output images: gradient images (I_x, I_y, D_z^x, D_z^y) and motion fields (V_x, V_y, V_z^x, V_z^y).

descriptor introduced by the work in Ref. 14. SIFT-like descriptor fuses RGB-D data and it is invariant to scale, translation, and partial occlusions. Moreover, compared with 3D EMoSIFT,[14] we propose a 3D modified motion space. And we will demonstrate that 3D SMoSIFT outperforms the previous works[13,14] in Sec. 4.

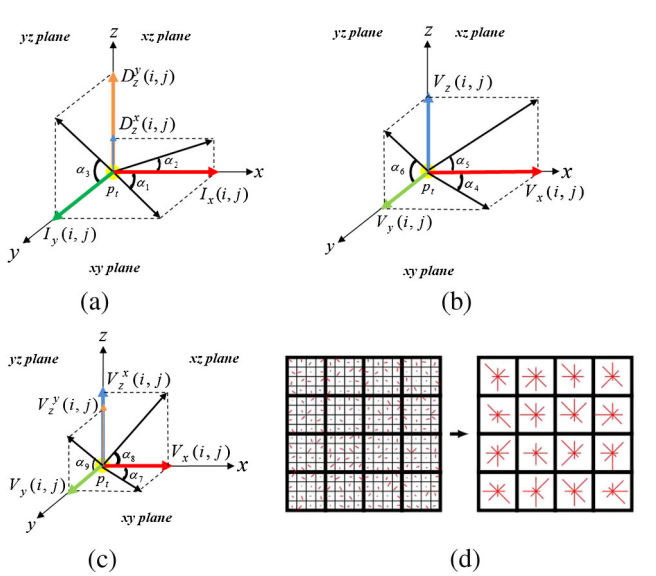

Fig. 5 Computing the descriptor on both three-dimensional (3D) gradient and motion spaces. (a) 3D gradient space. (b) 3D motion space. (c) 3D modified motion space. (d) SIFT-like descriptor calculation.

Suppose that a keypoint is detected at the l'th level in the pyramid at time t. Then we can extract local patches around the keypoint from four pyramids: P_G^t, P_G^{t+1}, P_D^t, and P_D^{t+1}. For instance, a keypoint is detected at the second level in the pyramid at time t. Then we can know the corresponding local patches around the keypoint as shown in Fig. 2, where green dots denote keypoints, and four local patches are extracted from four blue rectangles. Γ_G^t denotes the local patch from G_t^2, Γ_D^t from D_t^2, Γ_G^{t+1} from G_{t+1}^2, and Γ_D^{t+1} from D_{t+1}^2.

The four local patches are used to calculate gradient images and motion fields. First, to reduce the effects of noise, we use the Gaussian filter to smooth the extracted patches $(\Gamma_G^t, \Gamma_D^t, \Gamma_G^{t+1}, \Gamma_D^{t+1})$, and the corresponding results are $\Gamma_G^{t\prime}, \Gamma_D^{t\prime}, \Gamma_G^{t+1\prime}, \Gamma_D^{t+1\prime}$, as shown in Fig. 4. Second, we can calculate the dense optical flow[21] using two pairs of images $[(\Gamma_G^{t\prime}, \Gamma_G^{t+1\prime})$ and $(\Gamma_D^{t\prime}, \Gamma_D^{t+1\prime})]$. (We use the function cvCalcOpticalFlowLK in opencv library[27] to calculate dense optical flow.) Therefore, we can get the vertical and horizontal velocities (V_x, V_y, V_z^x, V_z^y) as shown in Fig. 4, where V_x and V_y are calculated from $\Gamma_G^{t\prime}$ and

Algorithm 1 3D SMoSIFT Feature Extraction from an RGB-D Video.

Input: (1) A sample with two videos: $V_G = [G_1, G_2, \ldots, G_Q]$ (gray-scale image), $V_D = [D_1, D_2, \ldots, D_Q]$ (depth image); (2) Number of frames: Q; (3) The number of pyramid levels: L.

Output: (1) The set of feature descriptors: X.

Initialization: $X = []$;

for $t = 1$ to $Q - 1$

Obtain the frames: G_t and G_{t+1} from V_G; D_t and D_{t+1} from V_D;

Build the pyramids with L levels: $P_G^t = \{G_t^1, \ldots, G_t^L\}$, $P_D^t = \{D_t^1, \ldots, D_t^L\}$,

$P_G^{t+1} = \{G_{t+1}^1, \ldots, G_{t+1}^L\}$ and $P_D^{t+1} = \{D_{t+1}^1, \ldots, D_{t+1}^L\}$ via Eq. 1

for $l = 1$ to L **do**

Obtain the images at the l'th level in the pyramids: G_t^l, D_t^l, G_{t+1}^l, and D_{t+1}^l;

Find the set of keypoints: $P = [p_1, \ldots, p_m]$ via Eqs. (4), (7), and 8 (see Fig. 3);

for $i = 1$ to m **do**

Get the local patches around a keypoint $p_i \subset P$ via Fig. 4;

Compute a descriptor in 3D gradient and motion spaces: $x \in \Re^{768}$ via Fig. 5;

$X = [X \quad x]$;

end

end

end

Table 1 Quantitative comparison among three-dimensional motion scale-invariant feature transform (3D MoSIFT)-based features. Experiments are performed in a workstation with Intel® Core™ *i*3-2120 CPU at 3.3 GHz and 8 GB RAM. All the features are written with c++ programs. The running time of all the features are tested, including both keypoint detection and descriptor calculation.

Feature name	Video size $w \times h \times N_{\text{frames}}$	s	n	L	Number of extracted features	Running time (s)	Average time (ms/f)
3D MoSIFT	$320 \times 240 \times 353$	5	6	—	5486	227.84	645.4
3D EMoSIFT	$320 \times 240 \times 353$	5	6	—	1897	218.53	619.1
3D SMoSIFT1	$320 \times 240 \times 353$	—	—	3	1837	7.2569	20.6
3D SMoSIFT2	$320 \times 240 \times 353$	—	—	3	4345	11.637	33.0
3D SMoSIFT3	$320 \times 240 \times 353$	—	—	3	4247	11.622	32.9
3D SMoSIFT4	$320 \times 240 \times 353$	—	—	3	2223	7.8141	22.1

$\Gamma_G^{t+1\prime}$, and V_z^x and V_z^y are calculated from $\Gamma_D^{t\prime}$ and $\Gamma_D^{t+1\prime}$. Third, at time t, the gradients can simply be calculated using $\Gamma_G^{t\prime}$ and $\Gamma_D^{t\prime}$ via Eq. (9).

$$\begin{aligned} I_x(i,j) &= \Gamma_G^{t\prime}(i,j+1) - \Gamma_G^{t\prime}(i,j) \\ I_y(i,j) &= \Gamma_G^{t\prime}(i+1,j) - \Gamma_G^{t\prime}(i,j) \\ D_z^x(i,j) &= \Gamma_D^{t\prime}(i,j+1) - \Gamma_D^{t\prime}(i,j) \\ D_z^y(i,j) &= \Gamma_D^{t\prime}(i+1,j) - \Gamma_D^{t\prime}(i,j), \end{aligned} \tag{9}$$

where (i,j) is the coordinate; I_x and I_y are the horizontal and vertical gradients from $\Gamma_G^{t\prime}$; and D_z^x and D_z^y are the horizontal and vertical gradients from $\Gamma_D^{t\prime}$.

3.3.1 *Feature descriptor in 3D gradient space*

As shown in Fig. 5(a), for a point p_t with the coordinate (i,j), the 3D gradient space can be constructed by $I_x(i,j), I_y(i,j), D_z^x(i,j)$, and $D_z^y(i,j)$. Now we use the xy plane to illustrate how to calculate the feature descriptor in the 3D gradient space. For each point p_t with its coordinate (i,j), we compute the gradient magnitude, $\text{mag}(i,j) = \sqrt{I_x(i,j)^2 + I_y(i,j)^2}$, and orientation, $\text{ori}(i,j) = \tan^{-1}[I_y(i,j)/I_x(i,j)]$ in the xy plane. Then, in xy plane, we divide the local patch of size 16×16 around the keypoints as shown in Fig. 5(d) and we can calculate SIFT descriptor with 128 dimensions. Similarly, we calculate SIFT descriptors in xz and yz planes. Therefore, the descriptor vector has 384 (128×3) dimensions in 3D gradient space.

3.3.2 *Feature descriptor in 3D motion space*

Figure 5(b) shows 3D motion space construction by Wan et al.[14] In 3D motion space, the authors[14] calculate the depth velocity by $V_z(i,j) = \Gamma_D^{t+1}[i+V_y(i,j), j+V_x(i,j)] - \Gamma_D^t(i,j+V_x)$. Although it is simple to calculate the depth velocity, the poor performance in 3D motion space is proved by Wan et al.[14] Therefore, we modified 3D motion space as shown in Fig. 5(c). In 3D modified motion space, we calculate the depth horizontal and vertical velocities (V_z^x, V_z^y) using optical flow (see Fig. 4). Here, we use the yz plane to illustrate how to calculate the feature descriptor in 3D modified motion space. For each point p_t with its coordinate (i,j), we compute the gradient magnitude, $\text{mag}(i,j) = \sqrt{V_y(i,j)^2 + V_z^y(i,j)^2}$, and orientation, $\text{ori}(i,j) = \tan^{-1}[V_z^y(i,j)/V_y(i,j)]$ in the yz plane. Then, in yz plane, we divide the local patch of size 16×16 around the keypoints as shown in Fig. 5(d) to calculate SIFT descriptor. Similarly, we can compute the magnitude and orientation for the local patch around the detected points in other two planes. Therefore, we obtain the descriptors with 384 dimensions in the 3D motion space. Finally, we integrate these two descriptor vectors into a long descriptor vector with 768 dimensions.

Fig. 6 Some samples from one-shot learning ChaLearn gesture database.

Table 2 Performances with different types and levels on validation batches (valid01 to valid20).

	Feature type			
Level L	SMoSIFT1	SMoSIFT2	SMoSIFT3	SMoSIFT4
1	0.2495	0.2745	0.2765	0.2335
2	0.1845	0.2185	0.222	0.182
3	0.1785	0.202	0.1925	0.174

We know that SIFT-like descriptor is calculated from a grid of size 16×16 [see Fig. 5(d)]. So the size of the extracted four patches (Γ_G^t, Γ_D^t, Γ_G^{t+1}, and Γ_D^{t+1}) should be larger than 16×16. In order to more accurately compute the gradients in the edge of the local patches, we define local patch size with 25×25.

3.4 *Overview of the 3D SMoSIFT Feature*

In this section, we propose a new spatiotemporal feature called 3D SMoSIFT. The new feature is invariant to scale, transition, and partial occlusions. For a given sample including an RGB video and a depth video, we can calculate feature descriptors between two consecutive frames. Then the sample can be represented by a set of feature descriptors extracted from video clips. Algorithm 1 illustrates how to calculate the proposed feature.

4 Performance Evaluation

A critical experimental evaluation of the proposed feature is presented in this section. Our main objective is to evaluate the strength of the proposed feature in different conditions (e.g., scale, transition, partial occlusions). The secondary goal is to compare with other state-of-the-art spatiotemporal features. Besides, we analyze the computational complexity of 3D SMoSIFT. Three public datasets that exhibit various motions in various scenes are used in our experiments. We have released the code (https://mloss.org/software/view/499/),

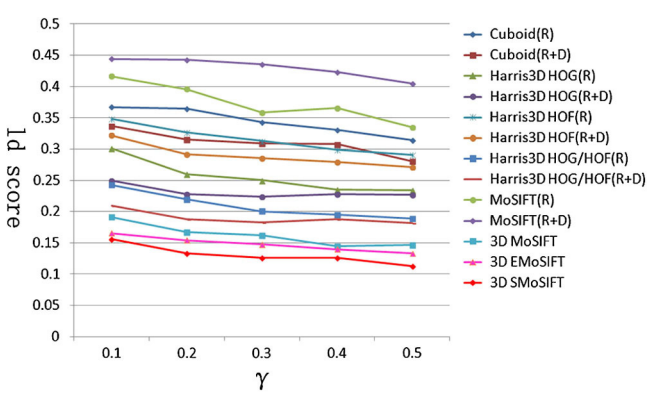

Fig. 7 Performance evaluation with different γ on final batches (final21 to final40). Our method consistently outperforms others. (R) represents that the features are extracted from RGB video. (R+D) denotes the features are extracted from both RGB and depth videos.

which includes 3D MoSIFT,[13] 3D EMoSIFT,[14] and 3D SMoSIFT features.

In our experiments, we use BoW model mentioned in Sec. 2.2 to evaluate the proposed feature, where nearest-neighbor classification is used for all datasets. For one-shot learning gesture recognition, levenshtein distance ld (normalized by the length of the truth labeling)[29] is used to evaluate the performance, which accounts for the number of edits that must be performed for taking a sequence of predictions into the ground truth labels. We use a parameter γ instead of the codebook size M (which is used in Ref. 14) in BoW 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 overclustering on some batches. The overclustering will affect the final performances.[14] Therefore, we can set different codebook sizes to different batches when we use a given value for γ. The corresponding codebook size can be calculated as $M = \gamma \times L_{tr}$, where L_{tr} is the number of features extracted from training samples on a certain batch. Unless mentioned otherwise, we set $\gamma = 0.5$, which can obtain a high performance by the work in Ref. 14. For Cornell Activity Dataset-60 (CAD-60), the precision and recall are used to evaluate the proposed feature. When the precision and recall are larger, the performance is better.

4.1 *Parameter Settings*

In the proposed feature, two main parameters have to be set. First, the number of levels L is used to build the pyramid. In order to keep the new feature invariant to scale space, the more larger L is, the better the performance will be (see Sec. 4.3). As in the previous discussion in Sec. 3.1, we usually set $L = 4$ for one image with 640×480 and $L = 3$ for one image with 320×240. The second parameter is to choose the best feature type from 3D SMoSIFT1 to 3D SMoSIFT4. In Fig. 3, we can see that when two consecutive gray-scale frames are used to track interest points, keypoints are detected in the motion regions [see Figs. 3(a) and 3(d)]. However, in Figs. 3(b) and 3(c), some futile points are detected as keypoints at the second level in the pyramid. The futile points will lead to poor performances, which will be proofed in Sec. 4.3.

4.2 *Complexity Analysis*

First, we analyze the computational complexity of 3D (E) MoSIFT. The most demanding part is to build Gaussian pyramids. For instance, if we build a Gaussian pyramid with s octaves and each octave has n intervals, the complexity is $s\Sigma_{i=1}^{n} O[(1/4)^{i-1}N] \sim sO\{[1-(1/4)^n]N\}$, where N is the number of pixels in the original image from a video. However, for 3D SMoSIFT, we first extract the local patches from the original frame, and then the complexity is $p\Sigma_{i=1}^{L} O[(1/4)^{i-1}N_1] \sim pO[1-(1/4)^L N_1]$, where L is the number of levels in the pyramid, $N_1 (N_1 \ll N)$ is the local patch size, and p is the number of keypoints. Because p is a small integral value and $N_1 \ll N$, the complexity of the proposed feature is much lower than 3D (E)MoSIFT. As shown in Table 1, whenever we use any type of the proposed feature, it is at least 18 times faster than 3D (E) MoSIFT, which indicates the proposed feature is very suitable for real-time applications (25 ms/f on average).

Fig. 8 Some samples from different datasets. (a) Untranslated. (b) Translated. (c) Scaled. (d) Occluded.

4.3 *CGD*

A comprehensive dataset of human actors performing a variety of gestures has been made available to researchers under Microsoft ChaLearn Gesture Challenge.[29] The goal of the challenge is to employ systems to perform gesture recognition from videos containing diverse backgrounds, using a single example per gesture, 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™ camera. The dataset 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 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 one to five gestures. Some samples are shown in Fig. 6.

Because one video probably includes multiple gestures, we should first split the video into isolate gestures before extracting spatiotemporal features. To achieve temporal segmentation, we used dynamic time warping algorithm, which is introduced in Ref. 14.

First, four types of the proposed feature are evaluated with different levels on validation batches (20 batches, ~2000 gestures), and the results (*ld* score) are shown in Table 2. We can see that when L is more larger, the performance is better. Besides, compared with 3D SMoSIFT1 and SMoSIFT4, 3D SMoSIFT2 and SMoSIFT3 have obtained relatively poor performances. That is because depth data used to track interest points have no texture information, which causes some futile keypoints to be detected, as shown in Figs. 3(b) and 3(c). In short, for one-shot learning, we set $L = 3$ and use 3D SMoSIFT4 as the proposed feature. That is to say we will use 3D SMoSIFT instead of 3D SMoSIFT4 in the next content, unless mentioned otherwise.

Second, we set different values for γ in BoW model and calculate average *ld* score via varied spatiotemporal features. Figure 7 shows the results on final batches (final21 to final40). We can see that the results of 3D SMoSIFT consistently exceed traditional features (e.g., Cuboid,[16] Harris3D,[17] MoSIFT,[24] 3D MoSIFT,[13] and 3D EMoSIFT[14]). More specifically, the least *ld* score (corresponding to the best recognition rate) for 3D SMoSIFT is 0.113, compared to 0.13311 for 3D EMoSIFT, 0.14476 for 3D MoSIFT, 0.28064 for Cuboid, 0.18192 for Harris3D, and 0.335 for MoSIFT. Also, we can give some conclusions from Fig. 7.

- From the previous works, traditional features have achieved promising results[16,17,24] in human activity recognition. However, those features may not be sufficient to capture the distinctive motion pattern only from RGB data when there is only one training sample per class. It indicates that those features based on RGB data are not suitable for one-shot learning.

Table 3 Results of different methods on translated and scaled dataset.

Method	Untranslated	Translated	Scaled	Team name
Motion signature analysis	0.2316	0.2255	0.2571	Alfine
Hidden Markov model+HOG/HOF	0.2896	0.5993	0.5296	Turtle Tamers
BoW+3D MoSIFT	0.2623	0.2612	0.2913	JoeWan
NA	0.3387	0.6278	0.5843	WayneZhang
BoW	0.3644	0.4252	0.4358	Manavender
Dynamic time warping+principle motion	0.4743	0.664	0.6066	HITCS
BoW+3D EMoSIFT	0.2635	0.253	0.2540	Ref. 14
BoW+3D SMoSIFT	0.257	0.2475	0.263	Our method

Table 4 Results of different methods on the unoccluded and occluded dataset.

Method	Unoccluded	Occluded
BoW+3D MoSIFT	0.1525	0.1823
BoW+3D EMoSIFT	0.1185	0.1375
BoW+3D SMoSIFT	0.114	0.1335

Table 5 Results on Cornell Activity Dataset-60 (CAD-60) using different methods.

		Maximum entropy Markov model		3D MoSIFT		3D EMoSIFT		3D SMoSIFT	
Location	Activity	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec
Bathroom	Brushing teeth	88.5	55.3	100.0	75.0	100.0	50.0	100.0	50.0
	Rinsing mouth with water	51.4	51.4	100.0	75.0	100.0	50.0	100.0	75.0
	Wearing contact lenses	78.6	88.3	80.0	100.0	63.6	87.5	72.7	100.0
	Average	72.7	65.0	93.3	83.3	87.9	62.5	90.9	75.0
Bedroom	Drinking water	70.7	71.7	50.0	25.0	25.0	25.0	100.0	25.0
	Opening pill container	95.0	57.4	80.0	100.0	75.0	50.0	85.7	100.0
	Talking on the phone	63.2	48.3	0	0	42.9	75.0	80.0	100.0
	Average	76.1	59.2	43.3	41.7	47.6	50.0	88.6	75.0
Kitchen	Cooking (chopping)	45.6	43.3	60.0	75.0	50.0	75.0	50.0	75.0
	Cooking (stirring)	24.8	17.7	100.0	50.0	0	0	66.7	50.0
	Drinking water	95.4	75.3	100.0	25.0	0	0	50.0	25.0
	Opening pill container	91.9	55.2	91.7	91.7	46.2	50.0	84.6	91.7
	Average	64.4	47.9	87.9	60.4	24.0	31.3	62.8	60.4
Living room	Drinking water	54.3	69.3	50.0	25.0	40.0	50.0	66.7	50.0
	Relaxing on couch	31.3	21.1	50.0	25.0	66.7	50.0	42.9	75.0
	Talking on couch	73.2	43.7	0	0	0	0	100.0	25.0
	Talking on the phone	51.5	48.5	100.0	50.0	50.0	50.0	40.0	50.0
	Average	52.6	45.7	50.0	25.0	39.2	37.5	62.4	50.0
Office	Drinking water	67.1	68.8	50.0	50.0	50.0	25.0	66.7	50.0
	Talking on the phone	69.4	48.2	100.0	25.0	40.0	50.0	50.0	50.0
	Working on computer	83.4	40.7	100.0	75.0	100.0	75.0	60.0	75.0
	Writing on whiteboard	75.5	81.3	100.0	75.0	66.7	50.0	100.0	100.0
	Average	73.8	59.8	87.5	56.3	64.2	50.0	69.2	68.8
Overall average		**67.9**	**55.5**	**72.4**	**53.3**	**52.6**	**46.3**	**74.8**	**65.8**

- Although 3D MoSIFT-based features are derived from MoSIFT, MoSIFT still cannot achieve satisfactory outcomes. That is because the descriptors captured by MoSIFT are simply calculated from RGB data, while 3D MoSIFT-based features can construct 3D gradient and motion space from the local patch around each keypoint by fusing RGB-D data.
- Among 3D MoSIFT-based features, the proposed feature achieves the best performance. In addition, the proposed feature is much faster (see Table 1) than both 3D MoSIFT and 3D EMoSIFT features, which indicates it can provide a new chance to apply in real-time applications for gesture recogntion.

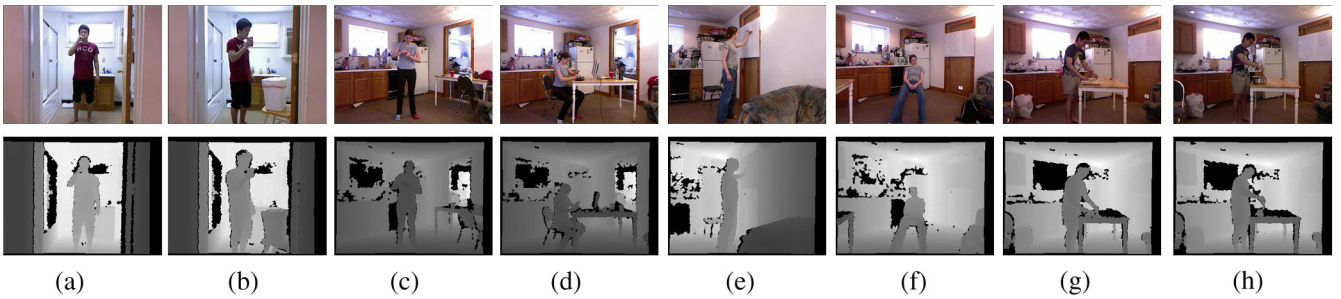

Fig. 9 Example frames from different actions obtained from Cornell Activity Dataset-60. (a) Brushing teeth. (b) Rinsing mouth with water. (c) Opening pill container. (d) Working on computer. (e) Writing on whiteboard. (f) Relaxing on couch. (g) Cooking (chopping). (h) Cooking (stirring).

Table 6 A comparison of different methods on CAD-60.

Reference	Published year	Precision (%)	Recall (%)
31	2012	65.32	—
13	2012	72.4	53.3
32	2013	71.9	66.6
33	2013	74.70	—
14	2013	52.6	46.3
3D SMoSIFT (the proposed feature)		74.8	65.8

4.3.1 *Translated and scaled dataset*

There are some additional batches used to test the robustness of recognition to translated or scaled data. The untranslated batches are the same as the previous Chalearn gesture data, where the user sits in a fixed position relative to a camera. However, it can happen that the user shifts his position on the translated batches. Besides, the scaled batches mean the gestures are performed in different scale space. Some samples are shown in the first sixth columns of Fig. 8. We compared the proposed feature with the top ranking methods on one-shot learning gesture recognition competition (round 2). The top ranking methods were introduced by Guyon et al.[30] Besides, we also compared the proposed feature with 3D (E)MoSIFT. The results are reported in Table 3. We can see that three teams' methods (Alfine, Joewan, and Manavender) are robust against translation and scale. However, other three teams exhibit important performance degradation between untranslated and translated (or scaled), where the three teams use features rigidly positioned on image feature maps. Compared with 3D (E)MoSIFT, the proposed feature can get better results in most cases. Moreover, our method is comparative to the method of Alfine, who got the best performance in the competition, especially for the scaled and translated datasets (http://gesture.chalearn.org/data/translated-data).

4.3.2 *Synthetic occlusion dataset*

We have also tested the robustness of our approach against partial occlusion. For the occluded data, we first selected 20 batches from CGD as unoccluded dataset (unocc01 to

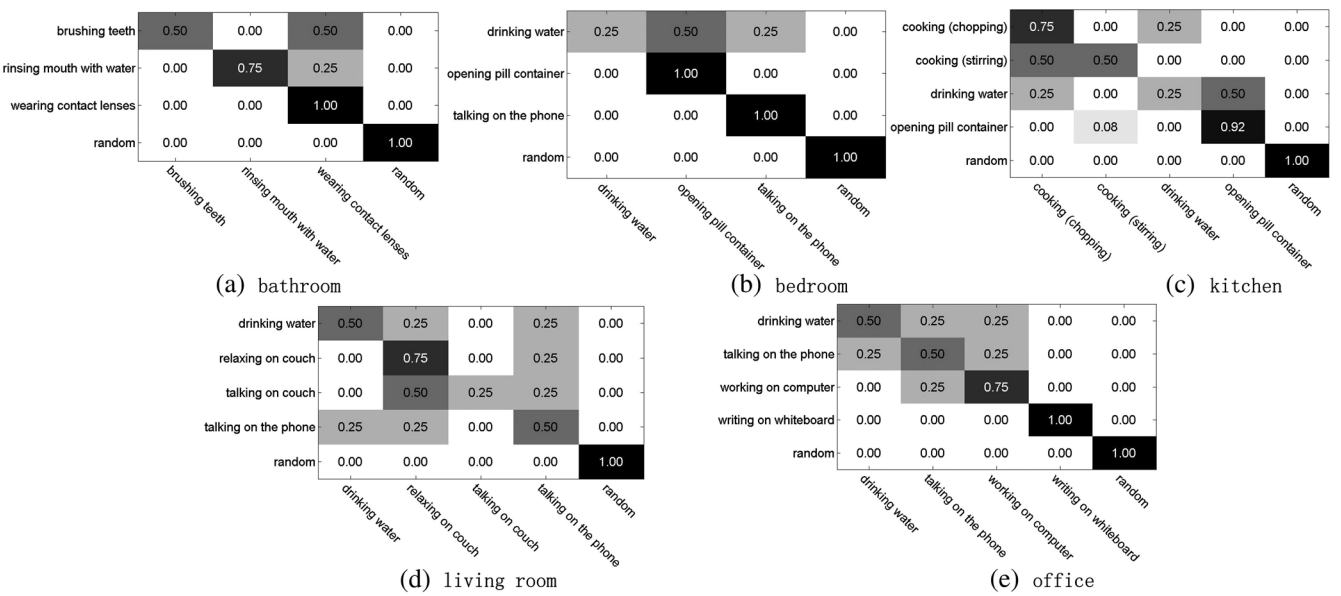

Fig. 10 Leave-one-out cross-validation confusion matrix for each location using 3D SMoSIFT feature.

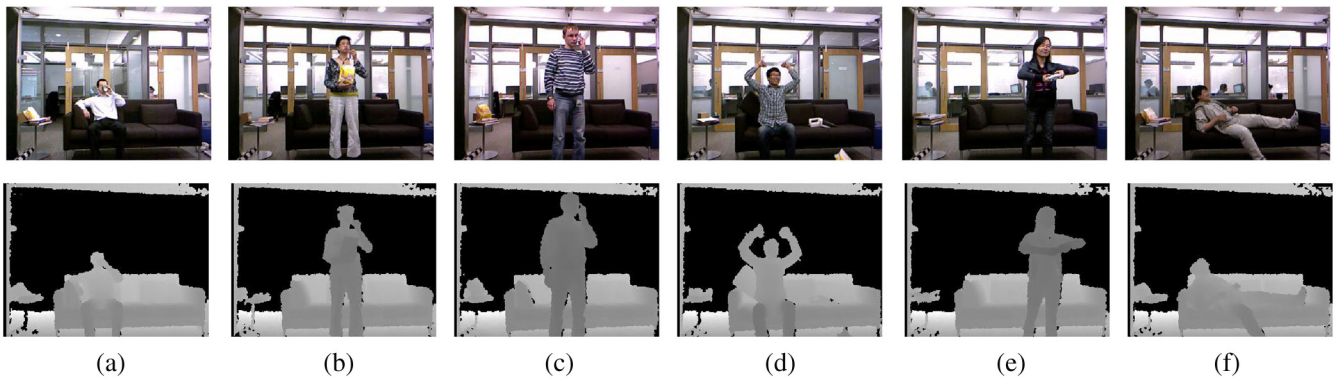

Fig. 11 Sample samples of MSR Daily Activity 3D Dataset. (a) Drink. (b) Eat. (c) Call cellphone. (d) Cheer up. (e) Play game. (f) Lay down.

unocc20). Then, we add a red rectangle with 10*240 pixels in the center of every frame of both RGB and depth videos. This red rectangle is treated as occlusions. And the new data is the occluded dataset (occlu01 to occlu 20) (http://gesture.chalearn.org/data/translated-data). Some examples are shown in Fig. 8(d). The results are given in Table 4. We can see that 3D MoSIFT-based features are robust to occluded data. Especially, 3D EMoSIFT and 3D SMoSIFT are better than 3D MoSIFT, and 3D SMoSIFT is slightly better than 3D EMoSIFT.

4.4 *CAD-60*

This dataset[10] has five different environments: office, kitchen, bedroom, bathroom, and living room. Three to four common activities were identified for each location (see Table 5), giving a total of 13 unique activities: brushing teeth, rinsing mouth with water, wearing contact lenses, drinking water, opening pill container, talking on the phone, cooking (chopping), cooking (stirring), relaxing on couch, talking on couch, working on computer, writing on whiteboard, random. The random activity contains sequence of random movements ranging from a person standing still to a person walking around and stretching his or her body, and the random activity is used in all locations. Compared with other activities, the random activity is meaningless and it is used to increase the difficulty of recognition. Data were collected from four different people: two males and two females. Some samples are shown in Fig. 9.

Following the experiments in Ref. 10, we considered the same five groups of activities based on their locations and used leave-one-out cross-validation to test each sample for each location. To extract skeletal HOG features,[10] we found the bounding box of the person in RGB-D videos and computed the local spatiotemporal features for that bounding box. Table 5 shows results of maximum entropy Markov model (MEMM),[10] 3D MoSIFT, 3D EMoSIFT, and our proposed feature. The proposed feature is able to classify with a precision/recall measure of 74.8%/65.8%, compared to 67.9%/55.5% for MEMM, 72.4%/53.3% for 3D MoSIFT, and 52.6%/46.3% for 3D EMoSIFT, which shows 3D SMoSIFT outperforms other methods for both precision and recall on average. That is because the proposed feature captures important local patch properties of motion. We can find that both MEMM and 3D MoSIFT-based features are able to classify well the activity containing some distinct characteristics, but perform poorly when the characteristics were subtler among different activities. Besides, we compared 3D MoSIFT-based features with other recent published papers, and the performance of 3D SMoSIFT is comparative to state-of-the-art methods as shown in Table 6.

Figure 10 shows confusion matrices between the activities in each location when we use 3D SMoSIFT. We can see that a lot of mistakes occur among similar activities. For example, talking on couch and talking on the phone are often confused. More interestingly, the proposed feature correctly classifies the random data in every location as shown in the bottom row of the confusion matrices.

4.5 *MSR Daily Activity 3D Dataset*

This dataset[9] 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, sit down. There are 10 subjects. Each subject performs each activity twice (standing position or sitting position). The total number of the activity samples is 320. Some examples from this dataset are shown in Fig. 11.

Table 7 The performance of our method on MSR daily activity 3D dataset, compared to previous approaches.

Reference	Method	Average accuracy (%)	Codebook size
36	Dynamic temporal warping	54	—
9	Local occupancy pattern features	42.5	—
9	Actionlet ensemble	85.75	—
34	Histogram of oriented 4D normals	80.0	—
13	3D MoSIFT+BoW	89.4	2000
35	Restricted graph-based genetic programming	90.4	—
14	3D EMoSIFT+BoW	90.9	2000
Our method	3D SMoSIFT+BoW	92.5	2000

For 3D MoSIFT-based features extraction, we found the bounding box of the person in RGB-D videos and computed the local spatiotemporal features for that bounding box. We used leave-one-out cross-validation to evaluate 3D MoSIFT-based features and the experimental results are given in Table 7, where we also gave some other state-of-the-art methods on this dataset. In Table 7, we compared 3D MoSIFT-based features with actionlet ensemble,[9] histogram of oriented 4D normals[34] features, and restricted graph-based genetic programming method.[35] Interestingly, 3D SMoSIFT outperforms other methods as shown in Table 7.

5 Conclusions and Future Works

In this paper, we propose a new method to extract the spatiotemporal feature from RGB-D videos. The proposed feature fuses RGB-D data to quickly detect keypoints and constructs 3D gradient and motion space to calculate SIFT-like descriptors. Compared with other 3D MoSIFT-based features,[13,14] the proposed feature gets a fast way to detect keypoints via tracking and filtering and it modifies the 3D motion space, which leads to the useful properties, including both speed and recognition accuracy. Compared with other existing features, such as Cuboid,[16] Harris3D,[17] and MoSIFT,[24] the proposed feature can achieve the best performance. Additionally, 3D SMoSIFT is invariant to scale, transition, and partial occlusions, and can capture more compact and richer video representations even though there is only one training sample for each class. Although the proposed method has achieved promising results, there are several avenues that can be explored. At first, most of the existing local spatiotemporal features are extracted from a static background or a simple dynamic background. In our feature research, we will focus on how to extract more compact features from cluttered backgrounds. Second, we will use the proposed feature to design a real-time system for gesture recognition in our future works.

Acknowledgments

This work was supported partly by the National Natural Science Foundation of China (61172128), National Key Basic Research Program of China (2012CB316304), New Century Excellent Talents in University (NCET-12-0768), the fundamental research funds for the central universities (2013JBZ003), Program for Innovative Research Team in University of Ministry of Education of China (IRT201206), Beijing Higher Education Young Elite Teacher Project (YETP0544), and Research Fund for the Doctoral Program of Higher Education of China (20120009110008). Besides, we would like to thank Isabelle Guyon, ChaLearn, Berkeley, California, who gave us insightful comments and suggestions to improve the works of three-dimensional MoSIFT-based features.

References

1. J. Aggarwal and M. Ryoo, "Human activity analysis: a review," *ACM Comput. Surv.* **43**(3), 16:1–16:43 (2011).
2. N. Zhang et al., "A generic approach for systematic analysis of sports videos," *ACM Trans. Intell. Syst. Technol.* **3**(3), 46:1–46:19 (2012).
3. L. Shao et al., "Spatio-temporal Laplacian pyramid coding for action recognition," *IEEE Trans. Cybern.* (2013).
4. L. Shao, S. Jones, and X. Li, "Efficient search and localisation of human actions in video databases," *IEEE Trans. Circuits Syst. Video Technol.* **24**(3), 504–512 (2014).
5. J. Wan et al., "Gesture recognition based on hidden Markov model from sparse representative observations," in *IEEE Int. Conf. on Signal Processing*, Vol. 2, pp. 1180–1183, IEEE (2012).
6. J. Han et al., "Enhanced computer vision with Microsoft Kinect sensor: a review," *IEEE Trans. Cybern.* **43**(5), 1318–1334 (2013).
7. H. J. Escalante et al., "Principal motion components for gesture recognition using a single-example," arXiv:1310.4822 (2013).
8. B. Ni, G. Wang, and P. Moulin, "RGBD-HuDaAct: a color-depth video database for human daily activity recognition," in *IEEE Conf. on Computer Vision Workshops*, pp. 1147–1153, IEEE (2011).
9. J. Wang et al., "Mining actionlet ensemble for action recognition with depth cameras," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1290–1297, IEEE (2012).
10. J. Sung et al., "Unstructured human activity detection from RGBD images," in *IEEE Conf. on Robotics and Automation*, pp. 842–849, IEEE (2012).
11. A. Hernández-Vela et al., "BoVDW: bag-of-visual-and-depth-words for gesture recognition," in *21st Int. Conf. on Pattern Recognition*, pp. 449–452, IEEE (2012).
12. M. R. Malgireddy, I. Inwogu, and V. Govindaraju, "A temporal Bayesian model for classifying, detecting and localizing activities in video sequences," in *IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 43–48, IEEE (2012).
13. Y. Ming, Q. Ruan, and A. G. Hauptmann, "Activity recognition from RGB-D camera with 3D local spatio-temporal features," in *IEEE Conf. on Multimedia and Expo*, pp. 344–349, IEEE (2012).
14. J. Wan et al., "One-shot learning gesture recognition from RGB-D data using bag of features," *J. Mach. Learn. Res.* **14**(9), 2549–2582 (2013).
15. J. Shotton et al., "Real-time human pose recognition in parts from single depth images," *Commun. ACM* **56**(1), 116–124 (2013).
16. P. Dollár et al., "Behavior recognition via sparse spatio-temporal features," in *2nd Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 65–72, IEEE (2005).
17. I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.* **64**(2–3), 107–123 (2005).
18. L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Vol. 2, pp. 524–531, IEEE (2005).
19. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.* **60**(2), 91–110 (2004).
20. J. Shi and C. Tomasi, "Good features to track," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 593–600, IEEE (1994).
21. B. D. Lucas et al., "An iterative image registration technique with an application to stereo vision," in *Int. Joint Conf. on Artificial Intelligence*, Vol. 81, pp. 674–679, AAAI Press (1981).
22. M. Camplani and L. Salgado, "Efficient spatio-temporal hole filling strategy for kinect depth maps," *Proc. SPIE* **8290**, 82900E (2012).
23. C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision Conf.*, pp. 147–151 (1988).
24. M.-y. Chen and A. Hauptmann, "MoSIFT: recognizing human actions in surveillance videos," (2009).
25. J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *Int. J. Comput. Vis.* **79**(3), 299–318 (2008).
26. C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Trans. Intell. Syst. Technol.* **2**(3), 27:1–27:27 (2011).
27. G. Bradski, "The OpenCV library," *Dr. Dobb's J. Softw. Tools* (2000).
28. J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," 2001, http://robots.stanford.edu/cs223b04/algo_tracking.pdf (20 March 2014).
29. I. Guyon et al., "ChaLearn gesture challenge: design and first results," in *IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 1–6, IEEE (2012).
30. I. Guyon et al., "Results and analysis of the ChaLearn gesture challenge 2012," in *Advances in Depth Image Analysis and Applications*, pp. 186–204, Springer, Berlin, Heidelberg (2013).
31. B. Ni, P. Moulin, and S. Yan, "Order-preserving sparse coding for sequence classification," *Lec. Notes Comput. Sci.* **7573**, 173–187 (2012).
32. X. Yang and Y. Tian, "Effective 3D action recognition using eigenjoints," *J. Vis. Commun. Image Represent.* **25**(1), 2–11 (2014).
33. J. Wang, Z. Liu, and Y. Wu, "Learning actionlet ensemble for 3D human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* (2013).
34. O. Oreifej and Z. Liu, "HON4D: histogram of oriented 4D normals for activity recognition from depth sequences," in *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 716–723, IEEE (2013).
35. L. Liu and L. Shao, "Learning discriminative representations from RGB-D video data," in *Proc. Int. Joint Conf. on Artificial Intelligence*, pp. 1493–1500, AAAI Press (2013).
36. M. Müller and T. Röder, "Motion templates for automatic classification and retrieval of motion capture data," in *ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, pp. 137–146, ACM (2006).

Jun Wan received his BS degree from China University of Geosciences, Beijing, China, in 2008. From September 2008 to August 2009, he was a master's student in the School of Electronic and Information Engineering Department, Beijing Jiaotong University. Since September 2009, he has been a master-doctoral program student in Institute of Information Science, Beijing Jiaotong University. His research interests include machine learning, computer vision, image processing, especially for gesture/action recognition, hand tracking, and segmentation.

Qiuqi Ruan is a professor and PhD supervisor at the Institute of Information Science, Beijing Jiaotong University, Beijing, China. He has published 380 papers and 4 books. His research interests include pattern recognition, computer vision, multimedia information processing, and image processing.

Wei Li received her BS degree in computer science from Beijing Jiaotong University, Beijing, China, in 2008. Since September 2008, she has been a master-doctoral program student at the Institute of Information Science, Beijing Jiaotong University. Her research interests include pattern recognition and image processing.

Gaoyun An is an associate professor at the Institute of Information Science, Beijing Jiaotong University, Beijing, China. His research interests include pattern recognition and image processing.

Ruizhen Zhao is a professor at the Institute of Information Science, Beijing Jiaotong University, Beijing, China. His research interests include pattern recognition and image processing.