BoMW: Bag of Manifold Words for One-shot Learning Gesture Recognition from Kinect

Lei Zhang, Shengping Zhang, Feng Jiang, Yuankai Qi, Jun Zhang, Yuliang Guo, Huiyu Zhou

***Abstract*—In this paper, we study one-shot learning gesture recognition on RGB-D data recorded from Microsoft's Kinect. To this end, we propose a novel bag of manifold words (BoMW) based feature representation on sysmetric positive definite (SPD) manifolds. In particular, we use covariance matrices to extract local features from RGB-D data due to its compact representation ability as well as the convenience of fusing both RGB and depth information. Since covariance matrices are SPD matrices and the space spanned by them is the SPD manifold, traditional learning methods in the Euclidean space such as sparse coding can not be directly applied to them. To overcome this problem, we propose a unified framework to transfer the sparse coding on SPD manifolds to the one on the Euclidean space, which enables any existing learning method can be used. After building BoMW representation on a video from each gesture class, a nearest neighbour classifier is adopted to perform the one-shot learning gesture recognition. Experimental results on the ChaLearn gesture dataset demonstrate the outstanding performance of the proposed one-shot learning gesture recognition method compared against state-of-the-art methods. The effectiveness of the proposed feature extraction method is also validated on a new RGB-D action recognition dataset.**

***Index Terms*—Gesture recognition, Covariance descriptor, Riemannian manifold, reproducing kernel Hilbert space, Kernel sparse coding.**

I. Introduction

Human gestures provide a very useful way for our daily communication. For examples, when two normal persons are talking, gestures can be used to help one understand the other one better. For deaf people, gestures are their only way to communicate with other people. On the other hand, gestures can also be used for the interaction between human and computers. For example, people can use their hand gestures to control electrical devices at home. All these applications raise an interesting question that training a computer to automatically recognize human gestures will be very useful in different aspects including human computer interaction [1], [2], [3], [4], robot control [5], [6], [7], sign language recognition [8], [9], augmented reality [10] and so on. Due to its potential applications, gesture recognition has been attracting increasing attention in the computer vision community.

In the past few decades, a huge number of gesture recognition approaches have been proposed in the literature [11], [12], [13], [14], [15], [16], [17], [18], [19]. Although much effort has been devoted, the performance of gesture recognition is still not good enough for practical use because of two main challenges: **1)** there are large variations in gesture movements. For example, when performing the same gesture, different people have different speeds, trajectories and spatial positions of the hands' movement. Even when the same person performs a specific gesture at different times, the movements of the hands are also not identical; **2)** it is very difficult to accurately track the hands. For example, when the hand and face are overlapped or the background is similar to skin color, tracking the hands may fail. These difficulties become more challenging for one-shot learning gesture recognition when only one sample per each class is given for training.

Recently, Microsoft's Kinect has attracted increasing interest in both industry and research communities [20] since it can capture both RGB and depth information of a scene. When applied to gesture recognition, the depth information recorded by Kinect can be used to accurately track the human. Due to this appealing feature, it has been widely used in human action recognition [21] and also gesture recognition [22], [17]. For example, [22] first detects the hands using scene depth information and then employs Dynamic Time Warping for recognizing gestures. [17] extracts the static finger shape features from depth images and measures the dissimilarity between shape features for classification. In [23], a Bag of Features (BoF) model is first built upon the descriptors extracted at the detected points of interest from both RGB and depth images and then a nearest neighbour classifier is adopted for the classification. In [24], a multi-layered framework is proposed for gesture recognition, which first segment the moving hands and then extracts features from both the segmented semantic units and the whole gesture sequence. Although, the depth

S. Zhang is supported by the National Natural Science Foundation of China under Grant 61672188, the Key Research and Development Program of Shandong Province under Grant 2016GGX101021 and HIT Outstanding Young Talents Program. F. Jiang is supported by the Major State Basic Research Development Program of China (973 Program 2015CB351804) and the National Natural Science Foundation of China under Grant No. 61572155. J. Zhang is supported by the Natural Science Foundation of China (61403116) and the China Postdoctoral Science Foundation (2014M560507). H. Zhou is supported by UK EPSRC under Grants EP/N508664/1, EP/R007187/1 and EP/N011074/1, and Royal Society-Newton Advanced Fellowship under Grant NA160342. (Corresponding author: Shengping Zhang and Feng Jiang.)

L. Zhang is with the School of Art and Design, Harbin University, Harbin, 150086, PR China. E-mail: cszhanglei@gmail.com

S. Zhang is with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai, 264209, PR China. E-mail: s.zhang@hit.edu.cn.

F. Jiang is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, PR China. E-mail: fjiang@hit.edu.cn.

Y. Qi is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, PR China. E-mail: yk.qi@hit.edu.cn.

J. Zhang is with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, P. R. China. E-mail: zhangjun@hfut.edu.cn.

Y. Guo is with the School of Engineering, Brown University, Providence, United States. E-mail:yuliang_guo@brown.edu.

H. Zhou is with the School of Electronics, Electrical Engineering and Computer Science, Queen's University of Belfast, Belfast, BT3 9DT, United Kingdom. E-mail: h.zhou@qub.ac.uk.

information recorded by Kinect can help us to detect and track the moving hands, accurately segmenting the fingers is still very challenging since fingers have many complicated articulations and they usually occlude each other. Therefore, the performance of the existing methods that extracts features from the segmented hands can be degraded especially when the gestures are performed in a complicated background.

To overcome the drawbacks of the existing gesture recognition methods. In this paper, we propose a novel bag of manifold words (BoMW) based feature representation for gesture recognition. In contrast to most existing feature extraction methods for gesture recognition, the proposed BoMW does not depend on the accurate segmentation of the body or fingers and therefore is suitable for more practical scenes. In particular, we densely sample spatio-temporal cubics from both the RGB and depth videos. Then we extract motion features from RGB cubics and texture features from depth cubics. The motion and texture features from each cubic are then represented by a covariance descriptor. Since covariance matrices are sysmetric positive definite (SPD) matrices and the space spanned by them is a SPD manifold, traditional learning methods in the Euclidean space such as sparse coding can not be directly applied to them. To overcome this problem, we propose to transfer the sparse coding on SPD manifolds to the one on the Euclidean space with the help of the Stein kernel, which maps SPD matrices into a vector space while preserving the geometry structures of the manifold. Subsequently, a bag of manifold words based feature representation is obtained for each video from each gesture class. Finally, a nearest neighbour classifier is adopted to perform the one-shot learning gesture recognition.

In summary, the contributions of the proposed method are three-folds:

- We propose to use covariance descriptors to extract motion and texture features from both RGB and depth videos. To our best knowledge, our work is the first to use covariance descriptors for RGB-D gesture recognition.
- We propose to transfer the sparse coding problem on SPD manifold to sparse coding on the Euclidean space via Stein kernel and then give the optimization solution.
- We carry out extensive experiments on the Chalearn gesture challenge dataset [25] and NTU RGB+D dataset [26] to demonstrate the effectiveness of the proposed algorithm with comparisions to the state-of-the-art methods. We also evaluate the performance of the proposed feature extraction method on a new RGB-D action recognition dataset.

The rest of the paper is organized as follows. Some related works are first reviewed in Section II. In Section III, we briefly review some basics of Hilbert spaces, CovDs and their geometry. The proposed BoMW method is introduced in Section IV. Experimental results are given in Section V. Secton VI concludes the paper.

II. Related Work

The widely used features for gesture recognition are color [27], [28], shapes [29], [30] and motion [31], [32]. Compared to color and shape features, motion features extracted from two consecutive frames are more discriminative for gesture recognition because most of gestures can be distinguished by different motion patterns. To exploit motion information for gesture recognition, Agrawal and Chaudhuri [33] use the correspondences between patches in adjacent frames and then compute 2D motion histogram to represent the motion information. Shao and Ji [34] compute optical flow field from each frame and then use different combinations of the magnitudes and directions of optical flow field to compute a motion histogram. Zahedi et al. [35] combine skin color features and different first- and second-order derivative features to recognize sign language. Wong et al. [36] use PCA on motion gradient images of a sequence to obtain features for a Bayesian classifier. To extract motion features, Cooper et al. [37] extend haar-like features from spatial domain to spatio-temporal domain and proposes volumetric Haar-like features for gesture recognition.

Recently, depth information recorded from Kinect are used together with RGB data for gesture recognition. To extract more robust features from depth images for gesture recognition, Ren et al. [17] propose part based finger shape features, which do not depend on the accurate segmentation of the hands. Ming et al. [38] propose a new feature called 3D MoSIFT that is derived from MoSIFT [39]. Wan et al. [23] extend SIFT to spatio-temporal domain and propose 3D EMoSIFT and 3D SMoSIFT to extract features from RGB and depth images, which are invariant to scale and rotation, and have more compact and richer visual representations. Wan et al. [40] propose a discriminative dictionary learning method on 3D EMoSIFT features based on mutual information and then use sparse reconstruction for classification. Based on 3D Histogram of Flow (3DHOF) and Global Histogram of Oriented Gradient (GHOG), Fanello et al. [41] apply adaptive sparse coding to capture high-level feature patterns. Wu et al. [42] utilize both RGB and depth information and an extended-MHI representation is adopted as the motion descriptors. In [24], a multi-layered feature extraction method is proposed, which extracts features from both the segmented semantic units and the whole gesture sequence and then sequentially classifies the motion, location and shape components.

In the literature, many classifiers are used for gesture recognition, e.g., Dynamic Time Warping (DTW) [43], [44], [45], [24], linear SVMs [41], neuro-fuzzy inference system networks [46], hyper rectangular composite NNs [47], 3D Hopfield NN [48], sparse coding [49], [50], [51], [52], [53], [54]. Due to the ability of modeling temporal signals, Hidden Markov Model (HMM) is possibly the most well known classifier for gesture recognition. Bauer and Kraiss [55] propose to use HMM as classifier with 2D motion features for gesture recognition. Vogler [12] presents to use a parallel HMM for continuous gesture recognition. Fang et al. [56] propose a self-organizing feature maps/hidden Markov model (SOFM/HMM) for gesture recognition in which SOFM is used as an implicit feature extractor for continuous HMM. Recently, Wan et al. [57] propose ScHMM to deal with the gesture recognition where sparse coding is adopted to find succinct representations and Lagrange dual is applied to learn the codebook.

One-shot learning gesture recognition is more challenging than traditional gesture recognition because only one training sample is available for each class. In the literature, several previous works have been focused on one-shot learning gesture recognition. For example, in [16], gesture sequences are viewed as third-order tensors and decomposed to three Stiefel sanifolds and a natural metric is inherited from the factor manifolds. A geometric framework for least square regression is further presented and applied to gesture recognition. Mahbub et al. [32] propose a space-time descriptor and apply Motion History Imaging (MHI) techniques to track the motion flow in consecutive frames. Seo and Milanfar [58] present a novel action recognition method based on space-time locally adaptive regression kernels. Escalante et al. [59] introduce principal motion components for one-shot learning gesture recognition. 2D maps of motion energy are obtained for each pair of consecutive frames in a video. Motion maps associated to a video are further processed to obtain a PCA model, which is used for gesture recognition with a reconstruction-error approach. More one-shot learning gesture recognition methods are summarized by [60]. Very recently, zero-shot learning [61] has also attracted increasing interest, which is more challenging to gesture recognition.

III. Background

To facilitate the presentation of our proposed method, we briefly review some basics of Hilbert spaces, CovDs and their geometry in this section.

Notations. We use $[n]$ to denote the set $\{1,\dots,n\}$. Vectors are always column vectors and are denoted by bold lower letters (e.g., $\boldsymbol{a}$). Notation a_i is used to indicate element at position the i of vector $\boldsymbol{a}$. Matrices are denoted by bold upper case letters (e.g., $\boldsymbol{A}$). Notation A_{ij} is used to indicate the (i,j)-th element of of matrix $\boldsymbol{A}$ and A_i the i-th column vector of matrix $\boldsymbol{A}$. The norm of matrix is always the entrywise norm. By default, $\|\cdot\|$ refers to ℓ_2-norm. For a symmetric invertible matrix $\boldsymbol{A}$, let $\boldsymbol{A}^{-1}$ denotes its inverse. We will use the following standard result in optimization.

Lemma 1. *Consider the following optimization problem for* $\boldsymbol{y}\in\mathbb{R}^n$, $\lambda>0$

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{x}-\boldsymbol{y}\|^2+\lambda\|\boldsymbol{x}\|_1.$$

This problem attains its minimum value at $\boldsymbol{x}=soft(\boldsymbol{y},\lambda/2)$.

Proof: Let $J(\boldsymbol{x}) = \|\boldsymbol{x}-\boldsymbol{y}\|^2+\lambda\|\boldsymbol{x}\|_1$. Let $\boldsymbol{x}=(x_1,\dots,x_n)$ and $\boldsymbol{y}=(y_1,\dots,y_n)$. We can rewrite $J(\boldsymbol{x})$ as

$$J(\boldsymbol{x})=\sum_{i=1}^{n}(x_i-y_i)^2+\lambda|x_i|,$$

and minimize $J(\boldsymbol{x})$ by minimizing each term in the summation. So we consider the scalar function $j(x_i)=(x_i-y_i)^2+\lambda|x_i|$. Taking the derivative with respect to x_i (assuming $x_i\neq 0$) we get, $j'(x_i)=2(x_i-y_i)+\lambda\cdot\text{sign}(x_i)$. Setting $j'(x_i)=0$ gives $x_i=y_i-(\lambda/2)\cdot\text{sign}(x_i)$. The minimizer for $j(x_i)$ is obtained by applying *soft-threshold* rule to y_i with threshold $\lambda/2$. The soft-threshold rule is the following non-linear function for $a,T\in\mathbb{R}$

$$\text{soft}(a,T)=\begin{cases}a+T \text{ if } a\leq -T,\\ 0 \text{ if } |a|\leq T,\\ a-T \text{ if } a\geq T.\end{cases}$$

The minimization of $j(x_i)$ is obtained by setting x_i to $\text{soft}(y_i,\lambda/2)$. Because the variables in the function $J(\boldsymbol{x})$ are uncoupled and the solution is obtained by minimizing with respect to each x_i individually, the minimizer of $J(\boldsymbol{x})$ is obtained by applying soft-thresholding rule to each element, that is $\boldsymbol{x}=\text{soft}(\boldsymbol{y},\lambda/2)$. ■

Definition 1. *Let* $\boldsymbol{F}=\big[\boldsymbol{f}_1|\boldsymbol{f}_2|\cdots|\boldsymbol{f}_m\big]$ *be a* $d\times m$ *matrix, obtained by stacking* m *independent observations* $\boldsymbol{f}_i\in\mathbb{R}^d$ *from an image (for example each observation may correspond to one pixel in an image). The covariance descriptor* $\boldsymbol{C}$*, as the name implies, is defined as*

$$\boldsymbol{C}=\frac{1}{m-1}\sum_{i=1}^{m}\big(\boldsymbol{f}_i-\mu\big)\big(\boldsymbol{f}_i-\mu\big)^T, \tag{1}$$

where $\mu=\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{f}_i$ *is the mean of the observations.*

In this paper, we will use the following feature vector $\boldsymbol{f}_i$ as:

$$\boldsymbol{f}_j=\left(I_j,x,y,\left|\frac{\partial I_j}{\partial x}\right|,\left|\frac{\partial I_j}{\partial y}\right|,\left|\frac{\partial^2 I_j}{\partial x^2}\right|,\left|\frac{\partial^2 I_j}{\partial y^2}\right|,f_j^1,\dots,f_j^8\right)^T, \tag{2}$$

where I_j is the gray (color) value(s) at location (x,y), and $\partial/\partial x$, $\partial/\partial y$, $\partial^2/\partial x^2$ and $\partial^2/\partial y^2$ are gradients and Laplacians along x and y dimensions, respectively. Apart from the widely used color, gradient and Laplacian features, we propose to use 8 extra textural features [62] for maintaining the tracking performance. In particular, f_j^1,f_j^2,f_j^3 and f_j^4,f_j^5,f_j^6 are the maximum responses across 6 orientations over 3 scales for two anisotropic filters, respectively. The remaining two features f_i^7 and f_i^8 are the responses of a Gaussian and a Laplacian of Gaussian filters both with $\sigma=10$, respectively. Note that these 8-dimensional features can be extracted efficiently using a fast anisotropic Gaussian filter[1] as shown in [63]. Fig. 1 shows the 8 textural responses of an example frame.

A CovD is a Symmetric Positive Definite (SPD) matrix with a well-known non-Euclidean structure. A $d\times d$, real SPD matrix $\boldsymbol{C}$ has the property that $\boldsymbol{v}^T\boldsymbol{C}\boldsymbol{v}>0$ for all non-zero $\boldsymbol{v}\in\mathbb{R}^d$. The space of $d\times d$ SPD matrices, denoted by $\mathcal{S}_{++}^d$, is clearly not a vector space since multiplying an SPD matrix by a negative scalar results in a matrix which does not belong to $\mathcal{S}_{++}^d$. Instead, $\mathcal{S}_{++}^d$ forms the interior of a convex cone in the $d(d+1)/2$-dimensional Euclidean space.

The $\mathcal{S}_{++}^d$ space is most studied when endowed with a Riemannian metric and thus forms a Riemannian manifold [64]. The geodesic distance between two CovDs $\boldsymbol{C}_1,\boldsymbol{C}_2\in\mathcal{S}_{++}^d$ induced by the Affine Riemannian Metric (AIRM) [64] is

[1] We use the publicly available code downloaded at http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html

Fig. 1. A illustration of the 8 textural responses.

defined as:

$$d_{GD}(\boldsymbol{C}_1, \boldsymbol{C}_2) = \sqrt{\sum_{i=1}^{d} \log^2 \left(\lambda_i(\boldsymbol{C}_1, \boldsymbol{C}_2)\right)}, \tag{3}$$

where $\lambda_i(\boldsymbol{C}_1, \boldsymbol{C}_2)$ corresponds to the generalized eigenvalues of $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$. This metric is affine invariant and has been the most widely used Reimannian metric over CovDs. unfortunately, it is computationally expensive to use this metric and thus many of the recent studies employed the log-Euclidean Riemannian metric which has the following form:

Definition 2. *The Stein or S metric is a symmetric member of Bregman matrix divergences and is defined as:*

$$S(\boldsymbol{C}_1, \boldsymbol{C}_2) \triangleq \operatorname{logdet}\left(\frac{\boldsymbol{C}_1 + \boldsymbol{C}_2}{2}\right) - \frac{1}{2}\operatorname{logdet}(\boldsymbol{C}_1\boldsymbol{C}_2). \tag{4}$$

From a geometric point of view, one of the suitable ways of handling SPD matrices is considering their Riemannian structure with the geometry induced by Affine Invariant Riemannian Metric (AIRM) [64]. The Stein metric shares several properties that are akin to the ones by AIRM. Moreover, computing the Stein metric is less demanding [65]. A property of the Stein metric, which is immensely useful in our application, is its embedding property. More specifically, the kernel

$$k_S(\boldsymbol{C}_1, \boldsymbol{C}_2) = \exp\{-\beta S(\boldsymbol{C}_1, \boldsymbol{C}_2)\}, \tag{5}$$

is positive definite for certain choices of $\beta > 0$ [65].

A positive definite kernel enables us to transfer the problems defined over $\mathcal{S}_{++}^d$ to familiar problems in Reproducing Kernel Hilbert Spaces (RKHS). This has two major advantages: Firstly, the embedding transforms the nonlinear manifold into a (linear) Hilbert space, thus makes it possible to utilize the algorithms designed for linear spaces with manifold-valued data. Secondly, as evidenced by the theory of kernel methods in Euclidean spaces, it yields a much richer high-dimensional representation of the original data, making the tasks such as classification much easier. In the later sections, we will find out how this property helps us to perform sparse coding on CovDs efficiently.

IV. Proposed Method

The proposed gesture recognition method is composed of several key parts including dense covariance descriptor extracting, cookbook learning on SPD manifold, BoW histogram representation and nearest neighbour classification. In this section, we present the details of each part.

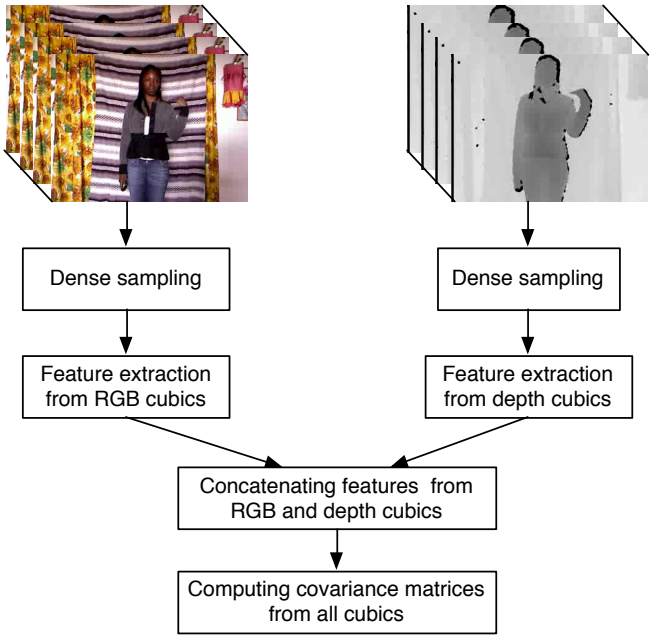


Fig. 2. The flow chart of the dense covariance descriptor extraction.

A. Dense Covariance Descriptor Extraction

Given the RGB video $\mathcal{C}$ and depth video $\mathcal{D}$ simultaneously recording a gesture, a set of dense 3D cubics can be sampled from them to extract local features. Let $w \times h \times t$ be the size of the sampled cubic and l_w, l_h and l_t be the sliding steps along three directions. For each pixel in the RGB cubic, we compute the following features.

1) Extracting motion features from RGB data: Let $I(x, y, t)$ be the intensity of the RGB video at pixel position (x, y, t) and $\mathbf{u}(x, y, t) = [u, v]^\top$ the corresponding optical flow vector. From the intensity and optical flow vector, we compute the following feature vector for the pixel

$$\mathbf{f}^m(x, y, t) = [u, v, u_t, v_t, Div, Vor, Gten, Sten] \tag{6}$$

where u_t and v_t are the 1-sf order partial derivatives of u and v with respect to t. Div, Vor, $Gten$ and $Sten$ are the divergence, vorticity and two tensor invariants of the optical flow, respectively. In particular, Div is the spatial divergence of the optical flow and can be computed as

$$Div(x, y, t) = \frac{\partial u(x, y, t)}{\partial x} + \frac{\partial v(x, y, t)}{\partial y} \tag{7}$$

Divergence captures the amount of local expansion in the fluid which can indicate gesture differences. Vor is the vorticity of the flow field and can be computed as

$$Vor(x, y, t) = \frac{\partial v(x, y, t)}{\partial x} - \frac{\partial u(x, y, t)}{\partial y} \tag{8}$$

Vorticity is used to measure local spin around the axis perpendicular to the plane of the flow field, which potentially captures locally circular motions of a moving pixel. To compute $Gten$ and $Sten$, we need to introduce two matrices, namely the gradient tensor $\nabla\mathbf{u}(x,y,t)$ and the rate of strain tensor $S(x,y,t)$

$$\nabla\mathbf{u}(x,y,t) = \begin{pmatrix} \frac{\partial u(x,y,t)}{\partial x} & \frac{\partial u(x,y,t)}{\partial y} \\ \frac{\partial v(x,y,t)}{\partial x} & \frac{\partial v(x,y,t)}{\partial y} \end{pmatrix} \quad (9)$$

$$S(x,y,t) = \frac{1}{2}\left(\nabla\mathbf{u}(x,y,t) + \nabla^{\top}\mathbf{u}(x,y,t)\right) \quad (10)$$

$Gten$ and $Sten$ are tensor invariants that remain constant no matter what co-ordinate system they are referenced in. They are defined in terms of $\nabla\mathbf{u}(x,y,t)$ and $S(x,y,t)$ as follows:

$$Gten(x,y,t) = \frac{1}{2}\left(tr^2(\nabla\mathbf{u}(x,y,t)) - tr(\nabla^2\mathbf{u}(x,y,t))\right)$$

$$Sten(x,y,t) = \frac{1}{2}\left(tr^2(S(x,y,t)) - tr(S^2(x,y,t))\right)$$

where $tr(\cdot)$ denotes the trace operation. $Gten$ and $Sten$ are scalar properties that combine gradient tensor components thus accounting for local fluid structures.

2) Extracting textural feature from depth data: Let $D(x,y,t)$ be the intensity value of the depth video at pixel position (x,y,t). To exploit the depth information, we propose to extract 8 textural features [62] for each pixel. The textural feature vector for pixel at (x,y,t) is computed as follows:

$$\mathbf{f}^t(x,y,t) = [f_1(x,y,t), f_2(x,y,t), \ldots, f_8(x,y,t)] \quad (11)$$

where $f_1(x,y,t)$, $f_2(x,y,t)$, $f_3(x,y,t)$ and $f_4(x,y,t)$, $f_5(x,y,t)$, $f_6(x,y,t)$ are the maximum responses across 6 orientations over 3 scales for two anisotropic filters, respectively. The remaining two features $f_7(x,y,t)$ and $f_8(x,y,t)$ are the responses of a Gaussian and a Laplacian of Gaussian filters both with $\sigma = 10$, respectively. Note that these 8-dimensional features can be extracted efficiently using a fast anisotropic Gaussian filter[2] as shown in [63].

3) Local Covariance descriptor computation: The final feature vector of the pixel at position (x,y,t) can be obtained by concatenating the motion features extracted from RGB video and the textural features extracted from the depth video as well as the position coordinates

$$\mathbf{f}(x,y,t) = [x,y,t,\mathbf{f}^m(x,y,t),\mathbf{f}^t(x,y,t)]^{\top} \in \mathbb{R}^d \quad (12)$$

where $d = 19$.

For any spatio-temporal cubic, Let $\boldsymbol{F} = \left[\boldsymbol{f}_1|\boldsymbol{f}_2|\cdots|\boldsymbol{f}_m\right]$ be a $d \times m$ matrix, obtained by stacking m feature vectors $\boldsymbol{f}_i \in \mathbb{R}^d$ extracted from all pixels inside the cubic. The covariance descriptor $\boldsymbol{C}$, as the name implies, is defined as

$$\boldsymbol{C} = \frac{1}{m-1}\sum_{i=1}^{m}\left(\boldsymbol{f}_i - \mu\right)\left(\boldsymbol{f}_i - \mu\right)^T, \quad (13)$$

where $\mu = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{f}_i$ is the mean of the feature vectors.

[2]We use the publicly available code downloaded at http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html

B. Coodbook learning on SPD Manifold

To learn a dictionary for sparse coding on SPD manifold, we operate in the kernel space via the Stein kernel with the associated mapping function ϕ. Let $\Phi(\mathbb{X}) = [\phi(\boldsymbol{X}_1), \ldots, \phi(\boldsymbol{X}_N)]$. The dictionary learning problem can be formulated as the following minimisation problem:

$$\arg\min_{\mathbb{D},\boldsymbol{H}} \|[\Phi(\mathbb{X}) - \Phi(\mathbb{D})\boldsymbol{H}\|_F^2 + \lambda\|\boldsymbol{H}\|_1. \quad (14)$$

where the i-th column of matrix $\boldsymbol{H} \in \mathbb{R}^{K\times N}$ is the coefficients associated with i-th training sample $\boldsymbol{X}_i$, λ is a regularization parameter. The ℓ_2-reconstruction error measures the quality of the approximation while the complexity is measured by the ℓ_1-norm of the optimal $\boldsymbol{H}$.

To solve Eq. 14, here we assume that the dictionary atoms lie within the subspace spanned by the input data, then we can write $\Phi(\mathbb{D})$ as a linear combination of $\Phi(\mathbb{X}) = [\phi(\boldsymbol{X}_1), \ldots, \phi(\boldsymbol{X}_N)]$. Formally, this means $\Phi(\mathbb{D}) = \Phi(\mathbb{X})\boldsymbol{A}$, where $\boldsymbol{A} \in \mathbb{R}^{N\times K}$ is the referred to as the atom representation dictionary.

1) Sparse Coding Phase: Given training samples atom representation dictionary $\boldsymbol{A}$, solving sparse coding coefficients for sample $\boldsymbol{X}$ can be formulated as:

$$\arg\min_{\boldsymbol{h}} \|\phi(\boldsymbol{X}) - \Phi(\mathbb{X})\boldsymbol{A}\boldsymbol{h}\|^2 + \lambda\|\boldsymbol{h}\|_1. \quad (15)$$

This above problem can be solved in many ways. Here, we describe an algorithm based on coordinate descent. We cyclically update over individual coordinates in $\boldsymbol{h} = (h_1, \ldots, h_K)$. Define, $\boldsymbol{r} = \phi(\boldsymbol{X}_i) - \Phi(\mathbb{X})\sum_{j=1,j\neq i}^{K} A_j h_j$. The update for h_i keeping all other variables in $\boldsymbol{h}$ fixed becomes:

$$\arg\min_{h_i} \|\boldsymbol{r} - \Phi(\mathbb{X})\boldsymbol{A}_i h_i\|^2 + \lambda|h_i|$$

The above optimization problem has a closed form minimum give by

$$h_i = \text{soft}(A_i^{\top}\Phi(\mathbb{X})^{\top}\boldsymbol{r}, \lambda/2)$$

where $A_i^{\top}\Phi(\mathbb{X})^{\top}\boldsymbol{r}$ can be computed as $A_i^{\top}\phi(\mathbb{X})^{\top}\phi(\boldsymbol{X}_i) - A_i^{\top}\phi(\mathbb{X})^{\top}\phi(\mathbb{X})\sum_{j=1,j\neq i}^{K} A_j h_j$.

2) Dictionary Learning Phase: Using the above representation of $\Phi(\mathbb{D}) = \Phi(\mathbb{X})\boldsymbol{A}$, Eq. (14) can be rewritten as

$$\arg\min_{\boldsymbol{A},\boldsymbol{H}} \|\Phi(\mathbb{X}) - \Phi(\mathbb{X})\boldsymbol{A}\boldsymbol{H}\|_F^2 + \lambda\|\boldsymbol{H}\|_1. \quad (16)$$

Let A_i denote the ith column in $\boldsymbol{A}$. We again optimize cyclically over individual A_i and $\boldsymbol{h}_i$ variables while keeping all other variables fixed.

Updating $\boldsymbol{h}_i$: Holding all variables except $\boldsymbol{h}_i$ fixed, the dictionary learning problem (16) can be reduced to:

$$\arg\min_{\boldsymbol{h}_i\in\mathbb{R}^n} \|\boldsymbol{P} - \Phi(\mathbb{X})A_i\boldsymbol{h}_i^{\top}\|_F^2 + \lambda\|\boldsymbol{h}_i\|_1 \quad (17)$$

where

$$\boldsymbol{P} = \Phi(\mathbb{X}) - \Phi(\mathbb{X})(\sum_{j\neq i} A_j\boldsymbol{h}_j^{\top})$$

is the residual matrix independent of A_i and $\boldsymbol{h}_i$. Note that $\boldsymbol{P}$ is the difference of a sparse matrix and rank one matrices. While $\boldsymbol{P}$ can possibly be a dense matrix, we never need to

evaluate it explicitly. In particular, our algorithm only needs to compute matrix vector products against $\boldsymbol{P}$, namely

$$\boldsymbol{P}^\top \Phi(\mathbb{X})A_i = \Phi(\mathbb{X})^\top \Phi(\mathbb{X})A_i - \sum_{j\neq i} \boldsymbol{h}_j A_j^\top \Phi(\mathbb{X})^\top \Phi(\mathbb{X})A_i.$$

We can use matrix-vector computations to evaluate $\boldsymbol{P}^\top A_i$ without evaluating $\boldsymbol{P}$ explicitly.

To finish the update equations for $\boldsymbol{h}_i$ note that

$$\|\boldsymbol{P}-\Phi(\mathbb{X})A_i\boldsymbol{h}_i^\top\|_F^2 = \mathrm{Tr}((\boldsymbol{P}-\Phi(\mathbb{X})A_i\boldsymbol{h}_i^\top)(\boldsymbol{P}-\Phi(\mathbb{X})A_i\boldsymbol{h}_i^\top)^\top).$$

Therefore, Eq. 17 can be rewritten as

$$\begin{aligned}
&\arg\min_{\boldsymbol{h}_i\in\mathbb{R}^n} \|\boldsymbol{P}-\Phi(\mathbb{X})A_i\boldsymbol{h}_i^\top\|_F^2 + \lambda\|\boldsymbol{h}_i\|_1 \\
&\equiv \arg\min_{\boldsymbol{h}_i\in\mathbb{R}^n} \frac{\|\boldsymbol{P}-\Phi(\mathbb{X})A_i\boldsymbol{h}_i^\top\|_F^2 + \lambda\|\boldsymbol{h}_i\|_1}{\|\Phi(\mathbb{X})A_i\|^2} \\
&\equiv \arg\min_{\boldsymbol{h}_i\in\mathbb{R}^n} \frac{\mathrm{Tr}((\boldsymbol{P}-\Phi(\mathbb{X})A_i\boldsymbol{h}_i^\top)(\boldsymbol{P}-\Phi(\mathbb{X})A_i\boldsymbol{h}_i^\top)^\top) + \lambda\|\boldsymbol{h}_i\|_1}{\|\Phi(\mathbb{X})A_i\|^2} \\
&\equiv \arg\min_{\boldsymbol{h}_i\in\mathbb{R}^n} \mathrm{Tr}\left(\left(\boldsymbol{h}_i - \frac{\boldsymbol{P}^\top\Phi(\mathbb{X})A_i}{\|\Phi(\mathbb{X})A_i\|^2}\right)\left(\boldsymbol{h}_i - \frac{\boldsymbol{P}^\top\Phi(\mathbb{X})A_i}{\|\Phi(\mathbb{X})A_i\|^2}\right)^\top\right) \\
&\qquad + \frac{\lambda\|\boldsymbol{h}_i\|_1}{\|\Phi(\mathbb{X})A_i\|^2} \\
&\equiv \arg\min_{\boldsymbol{h}_i\in\mathbb{R}^n} \left\|\boldsymbol{h}_i - \frac{\boldsymbol{P}^\top\Phi(\mathbb{X})A_i}{\|\Phi(\mathbb{X})A_i\|^2}\right\|^2 + \frac{\lambda\|\boldsymbol{h}_i\|_1}{\|\Phi(\mathbb{X})A_i\|^2}
\end{aligned}$$

Using Lemma 1, we get that the minimizer of Eq. 17 is

$$\boldsymbol{h}_i = \mathrm{soft}\left(\frac{\boldsymbol{P}^\top\Phi(\mathbb{X})A_i}{\|\Phi(\mathbb{X})A_i\|^2}, \frac{\lambda}{2\|\Phi(\mathbb{X})A_i\|^2}\right). \tag{18}$$

Updating A_i: Firstly note that for updating the dictionary, Eq. (16) can be reduced to (as there is no regularization term):

$$\arg\min_{\boldsymbol{A}\in\mathbb{R}^{N\times K}} \|\Phi(\mathbb{X}) - \Phi(\mathbb{X})\boldsymbol{A}\boldsymbol{H}\|_F^2. \tag{19}$$

Holding all variables except A_i fixed, the above dictionary learning problem can be reduced to:

$$\arg\min_{A_i\in\mathbb{R}^N} \|\boldsymbol{P}-\Phi(\mathbb{X})A_i\boldsymbol{h}_i^\top\|_F^2. \tag{20}$$

Like in the update of $\boldsymbol{h}_i$, after a simple numerical manipulation, Eq. 20 can be reduced as

$$\arg\min_{A_i\in\mathbb{R}^N} \left\|\Phi(\mathbb{X})A_i - \frac{P\boldsymbol{h}_i}{\|\boldsymbol{h}_i\|^2}\right\|^2.$$

The gradient of $\|\Phi(\mathbb{X})A_i - \boldsymbol{P}\boldsymbol{h}_i/\|\boldsymbol{h}_i\|^2$ with respect to A_i is $2\Phi(\mathbb{X})^\top(\Phi(\mathbb{X})A_i - \boldsymbol{P}\boldsymbol{h}_i/\|\boldsymbol{h}_i\|^2)$. Given the gradient, we can use gradient descent technique to solve the above minimization problem. The idea is to iteratively set

$$A_i^{(l)} = A_i^{(l-1)} - \alpha\left(2\Phi(\mathbb{X})^\top\Phi(\mathbb{X})A_i^{(l-1)} - \frac{2\Phi(\mathbb{X})^\top\boldsymbol{P}\boldsymbol{h}_i}{\|\boldsymbol{h}_i\|^2}\right), \tag{21}$$

where $A_i^{(l)}$ is the value of the variable A_i in the lth iteration and α is the step size. Note that using the definition of $\boldsymbol{P}$

$$\Phi(\mathbb{X})^\top\boldsymbol{P}\boldsymbol{h}_i = \Phi(\mathbb{X})^\top\Phi(\mathbb{X})\boldsymbol{h}_i - \Phi(\mathbb{X})^\top\Phi(\mathbb{X})\sum_{j\neq i} A_j(\boldsymbol{h}_j^\top\boldsymbol{h}_i). \tag{22}$$

Algorithm 1: Dictionary learning on SPD manifold

Input: SPD matrices $\Phi(\mathbb{X}) = [\phi(\boldsymbol{X}_1), \ldots, \phi(\boldsymbol{X}_N)]$
Output: $\boldsymbol{A}$
Initialize $A_1^{(0)}, A_2^{(0)}, \ldots, A_K^{(0)}$;
while *not(converge)* **do**
  for $i = 1$ **to** K **do**
    Updating $\boldsymbol{h}_i^\top$ (ith row of $\boldsymbol{H}$) using Eq. 18;
    Updating A_i (ith column of $\boldsymbol{A}$) by first solving Eq. 22 and then iterately performing
    for $l = 1$ **to** L **do**
      Solving Eq. 21.
    end
    $A_i = A_i^L$
  end
end

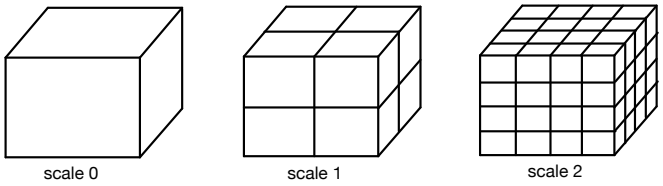


Fig. 3. An illustration of the spatial pyramid division of the video grid.

The detailed algorithm is summarized in Algorithm 1.

Convergence of Algorithm 1: Using a general result on convergence of Block Coordinate Descent, from [66], we can show that the limit point returned by Algorithm 1 is a stationary point of the objective function (16). In our experiments, we found that it averagely takes 10 iterations to converge.

C. Spatial Pyramid BoW Representation and Classification

Given a pair of RGB and depth videos from the k-th class, we adopt the spatial pyramid BoW representation to describe the global appearance of the video pair. As shown in Fig. 3, the video grid is divided into a set of cells at three scales. For example, $2^{3\times s}$ cells can be obtained at scale $s \in \{0, 1, 2\}$. Therefore, a total of $\sum_{s=0}^{2} 2^{3\times s}$ can be obtained. Let $\boldsymbol{C}_{c,s}^{(k)} = \boldsymbol{C}_{c,s,1}^{(k)}, \boldsymbol{C}_{c,s,2}^{(k)}, \ldots, \boldsymbol{C}_{c,s,N_{c,s}}^{(k)}$ denote the covariance descriptors computed from $N_{c,s}$ cubics inside the c-th cell at scale s from video pair of the k-th class. Using the sparse coding method introduced in subsection IV-B, we can compute the coefficient vectors for these covariance descriptors. Similarly, let $\boldsymbol{a}_{c,s,1}^{(k)}, \boldsymbol{a}_{c,s,2}^{(k)}, \ldots, \boldsymbol{a}_{c,s,N_{c,s}}^{(k)}$ denote the corresponding coefficient vectors. We can compute a histogram like representation for this cell as

$$\boldsymbol{h}_{c,s}^{(k)} = \sum_{i=1}^{N_{c,s}} \boldsymbol{a}_{c,s,i}^{(k)} \tag{23}$$

Therefore, the training video pair from the k-th class can be finally represented by a set of histograms $\{\boldsymbol{h}_{c,s} | s = 0, 1, 2, c = 1, 2, \ldots, 2^{3\times s}\}$.

At the recognition stage, given a test video pair, we can compute a set of histograms from this video pair using the

method above as $\{\boldsymbol{h}'_{c,s}|s = 0, 1, 2, c = 1, 2, \ldots, 2^{3\times s}\}$. We adopt the nearest neighbour classifier for recognition. The distance between the test sample and the training sample of the k-th class is computed as the $\mathcal{X}^2$ distance [67] between their pairwise histograms

$$\rho(k) = \sum_{s=0}^{2}\sum_{c=0}^{2^{3\times s}}\sum_{b=1}^{B}\frac{(\boldsymbol{h}'_{c,s}(b) - \boldsymbol{h}^{(k)}_{c,s}(b))^2}{\boldsymbol{h}'_{c,s}(b) + \boldsymbol{h}^{(k)}_{c,s}(b)} \tag{24}$$

Then the test sample is classified as the class which has the minimal distance.

V. Experiments

In this section, we evaluate the proposed one-shot learning gesture recognition method on the Chalearn gesture challenge dataset. We also validate the effectiveness of the proposed feature extraction methods based on bag of manifold words on a new RGB-D action recognition dataset. In the following, we present the experiment setup and results in detail.

A. Database

To validate the performance of the proposed method for one-shot learning gesture recognition, we compared our method with other state-of-the-art methods on the Chalearn gesture challenge dataset. Similar to [23], we also use 20 development batches (devel01 $\sim$ devel20), 20 validation batches (valid01 $\sim$ valid20) and 20 final batches (final21 $\sim$ final40) for testing. Each batch has a total of 47 gesture videos, which are split into a training set and test set. The training set includes a small set of vocabulary spanning from 8 to 15 gestures. Each test video contains 1 to 5 gestures. Detailed descriptions of the dataset can be found in [25]. All the samples are recorded with a Microsoft Kinect camera which provides both RGB and depth video clips.

B. Metric of Evaluation

We adopt the same metric of evaluation used in [25] which uses the Levenshtein distance to calculate the score between the classified labels and the ground truth labels. The Levenshtein distance between two strings is defined as the minimum number of operations (insertions, substitutions or deletions) needed to transform one string to the other. In our evaluation, one string contains the classified labels in all samples and the other string contains their ground truth labels. For all comparisons, we compute the mean Levenshtein distance (MLD) over all video clips and batches. Note that the smaller the MLD score is, the better the performance of an algorithm is.

C. Testing with different parameters

To evaluate the performance of the proposed method using different parameters, we first keep $\lambda = 0.02$ and change the values of B from 500 to 3500. Fig. 4 shows the performance when different B are used, from which we can see that the value of the parameter B significantly affects the performance when B increases from a smaller value 500 to a relatively larger value 3000 and then tends to be stable after being 3000. As well known the parameter λ in the sparse coding step controls how important the sparsity constraint is relative to the reconstruction error. As shown in Fig. 5, the MLD score has a significant drop when λ changes from 0.0001 to 0.005 and then achieves stable with little disturbance when λ changing from 0.005 to 0.03.

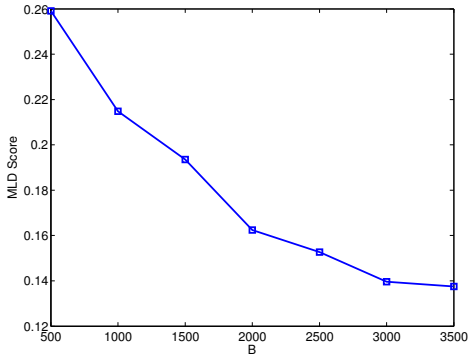


Fig. 4. The performance of the proposed method with $\lambda = 0.02$ and B changing.

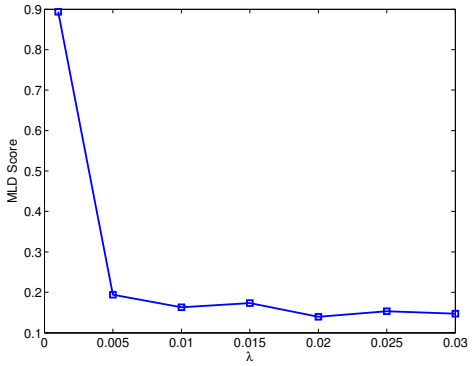


Fig. 5. The performance of the proposed method with $B = 3000$ and λ changing.

The other factor that may affect the performance is the pyramid structure. To test the performance of the proposed method when different pyramid divisions are used, we compute the MLD scores with parameters $B = 3000$ and $\lambda = 0.02$ but when different combinations of pyramid scales, e.g., $\{0\}, \{1\}, \{2\}, \{0, 1\}, \{0, 2\}, \{1, 2\}, \{0, 1, 2\}$ are used. The obtained MLD scores are shown in Table I. From this table, we can see that when the BoW representation is computed from the whole video grid, the performance is very poor. If dividing the video grid into a set of cells and then use the BoW histograms computed from all cells for the classification, the performance is significantly increased. We can also see that all three scales play important roles in improving the final performance.

D. Testing the effectiveness of key components

The proposed method (BoMW) has two key components including 1) extracting covariance descriptor to encode both the RGB and depth information and 2) learning codebook on SPD manifold. To validate the effectiveness of each key

TABLE I
THE MLD SCORES WHEN DIFFERENT COMBINATIONS OF PYRAMID SCALES ARE USED.

$s=\{0\}$	$s=\{1\}$	$s=\{2\}$	$s=\{0,1\}$	$s=\{0,2\}$	$s=\{1,2\}$	$s=\{0,1,2\}$
0.4382	0.3564	0.2983	0.3287	0.3054	0.1758	0.1396

component, we conducted several experiments to test the performance when a key component is replaced by other method. For example, to validate the effectiveness of the covariance descriptor extracted on both the RGB and depth information, we compare the performance of the proposed method when the covariance descriptor is extracted only from the RGB information or depth information. To validate the effectiveness of the proposed codebook learning method, we compare the proposed method with a SPD codebook learning method which uses the Log-Euclidean operator [68] to project covariance matrices to Euclidean vector space and then uses standard vector learning methods to learn the codebook. We use the abbreviation BoMW-RGB and BoMW-D to denote the proposed method using RGB and depth information, respectively. The abbreviation BoMW-LogE denotes the proposed method using Log-Euclidean operator to project covariance matrices to Euclidean vector space. Table II presents the experiment results of testing the effectiveness of key components. As we can see from Table II, when RGB or depth information is used alone, the resulting performance is worse than the proposed method when both RGB and depth information are used. By comparing RGB and depth information, we find that RGB information is important than depth information especially when there is significant contrast between the foreground and background. An example gesture is shown in Fig. 6(a). However, we also find that in some cases RGB information is not capable of discriminating different gestures. For example, in Fig. 6(b), the fingers of the subject have very similar colour information with the background and therefore when only RGB information is used, the proposed method fails to recognise the gesture. When the Log-Euclidean operator is used to project covariance matrices to Euclidean vector space, the resulting performance is worse than the proposed method using the Stein kernel to project covariance matrices to the RKHS.

TABLE II
EXPERIMENT RESULTS OF TESTING THE EFFECTIVENESS OF KEY COMPONENTS.

Method	MLD
BoMW-RGB	0.1725
BoMW-D	0.1958
BoMW-LogE	0.1538
BoMW	0.1396

E. Comparison with recent representative methods

We compare the proposed BoMW with several state-of-the-art feature extraction methods for gesture extraction including MoSIFT [39], 3D MoSIFT [38], 3D EMoSIFT [23] and Multi-Layered Features (MLF) [24]. All the results of the compared methods were obtained by running their methods on the same batches with the same training/testing splits. The compared

(a)

(b)

Fig. 6. Examples of challenging gestures. (a) An example of a gesture where RGB information is capable of recognising the gesture while depth information fails. (b) An example of a gesture where depth information is capable of recognising the gesture while RGB information fails.

results are shown in Table III, from which we can see that the proposed method is significantly superior to the MoSIFT method. It should be noted that MoSIFT was not originally proposed for gesture recognition from RGB-D data. In our experiments, we just use MoSIFT to extract features from both RGB and depth videos, respectively and then concatenate them together. Compared to three methods specifically designed for RGB-D data, the proposed method only has slight superiority in term of MLD. But we have to emphasize that the proposed method does not depend on the accurate segmentation of user's body and fingers, which makes our methods suitable for more complicated scenarios. In Tables IV and V, we also show the MLD scores of the compared methods on each development batch as well as each final batch. As shown in Table IV, in batches 2, 18 and 20, the proposed method does not achieve the best performance. However, the proposed method achieves the best performance in the remaining 17 batches. As shown in Table IV, in batches 24, 27, 30 and 34, the proposed method does not achieve the best performance. However, the proposed method achieves the best performance in the remaining 16 batches.

F. Evaluating on action recognition

In this section, we evaluate the effectiveness of the proposed feature extraction method for RGB-D action recognition. There are several large RGB-D action recognition datasets, such as MSR gesture 3D [69], NTU RGB+D [26], MSRC-12 [70], which can also be used to evaluated whether the

TABLE III
EXPERIMENT RESULTS OF THE COMPARED METHODS.

Method	MLD
MoSIFT [39]	0.4572
3D MoSIFT [38]	0.1837
3D EMoSIFT [23]	0.1629
MLF [24]	0.1483
BoMW	0.1396

TABLE IV
MLD SCORES OF THE COMPARED METHODS ON EACH BATCH (VALID01-VALID20).

Batch	MoSIFT	3D MoSIFT	3D EMoSIFT	MLF	BoMW
1	0.4012	0.1723	0.1421	0.1297	0.1201
2	0.3825	0.1539	0.1287	0.1102	0.1192
3	0.4831	0.2013	0.1823	0.1592	0.1472
4	0.4012	0.1796	0.1538	0.1321	0.1290
5	0.3927	0.1539	0.1291	0.1023	0.1021
6	0.4896	0.1876	0.1673	0.1490	0.1401
7	0.3536	0.1384	0.1462	0.1261	0.1224
8	0.4972	0.2017	0.2139	0.1823	0.1629
9	0.4793	0.1945	0.1733	0.1521	0.1491
10	0.4586	0.1838	0.1654	0.1467	0.1402
11	0.3965	0.1487	0.1374	0.1231	0.1031
12	0.3912	0.1526	0.1412	0.1245	0.1125
13	0.3627	0.1487	0.1401	0.1294	0.1099
14	0.4625	0.1987	0.1721	0.1581	0.1461
15	0.4135	0.1625	0.1524	0.1321	0.1293
16	0.4427	0.1793	0.1613	0.1481	0.1352
17	0.4824	0.2053	0.1856	0.1780	0.1762
18	0.5026	0.2169	0.1974	0.1824	0.1831
19	0.4987	0.2178	0.2012	0.2013	0.1925
20	0.4724	0.1926	0.1798	0.1723	0.1837

TABLE V
MLD SCORES OF THE COMPARED METHODS ON EACH BATCH (FINAL21-FINAL40).

Batch	MoSIFT	3D MoSIFT	3D EMoSIFT	MLF	BoMW
21	0.3942	0.1699	0.1499	0.1333	0.1298
22	0.3714	0.1603	0.1187	0.1203	0.1132
23	0.4626	0.1983	0.1749	0.1631	0.1391
24	0.4132	0.1732	0.1584	0.1421	0.1450
25	0.3893	0.1603	0.1301	0.1211	0.1121
26	0.4625	0.1814	0.1598	0.1356	0.1309
27	0.3618	0.1226	0.1502	0.1198	0.1213
28	0.4825	0.1941	0.2088	0.1725	0.1569
29	0.4613	0.1899	0.1793	0.1613	0.1511
30	0.4604	0.1802	0.1705	0.1527	0.1532
31	0.4012	0.1511	0.1401	0.1331	0.1124
32	0.3817	0.1490	0.1523	0.1300	0.1209
33	0.3725	0.1414	0.1519	0.1312	0.1115
34	0.4705	0.1904	0.1844	0.1609	0.1629
35	0.4213	0.1705	0.1624	0.1415	0.1390
36	0.4329	0.1811	0.1636	0.1516	0.1425
37	0.4793	0.1997	0.1725	0.1890	0.1839
38	0.4914	0.2104	0.1800	0.1876	0.1725
39	0.4895	0.2113	0.1923	0.2123	0.1834
40	0.4810	0.1874	0.1815	0.1800	0.1706

proposed method is also effective for action recognition or not. To this aim, we test the the proposed method on the NTU RGB+D dataset, which consists of 56,880 RGB+D video samples, captured from 40 different human subjects and in 80 distinct camera viewpoints, using Microsoft Kinect v2. This dataset has two evaluation settings: the cross-subject evaluation and cross-view evaluation. In the cross-subject evaluation, the 40 subjects are split into training and testing groups. Each group consists of 20 subjects. The IDs of training subjects in this evaluation are: 1, 2, 4, 5, 8, 9, 13, 14, 15, 16, 17, 18, 19, 25, 27, 28, 31, 34, 35, 38; remaining subjects are reserved for testing. For this evaluation, the training and testing sets have 40, 320 and 16, 560 samples, respectively. For the cross-view evaluation, the samples of camera 1 are picked as testing and the samples of cameras 2 and 3 are picked for training. In other words, the training set consists of front and two side views of the actions, while testing set includes left and right 45 degree views of the action performances. For this evaluation, the training and testing sets have 37, 920 and 18, 960 samples, respectively. After extracting features from each sample, LIBLINEAR SVM [71] is selected as the classifier. Please note that this dataset has four modals including RGB videos, depth sequences, skeleton data (3D locations of 25 major body joints), and infrared frames. However, we only use the RGB and depth modals for our experiments. A total of four methods in the literature are chosen for comparison including three map based methods (HOG^2 [72], Super Normal Vector [73] and HON4D [74]) and part-aware Long Short-Term Memory Networks (P-LSTM). The results of the proposed method and other compared methods are shown in Table VI. As we can see the proposed method is worse than P-LSTM, which is a new recurrent neural networks based learning framework. The proposed method is superior to other three depth based methods, which also indicates the proposed method is effective in combining both the RGB and depth information for action recognition.

TABLE VI
EXPERIMENT RESULTS OF THE TWO EVALUATION SETTINGS ON THE NTU RGB+D DATASET.

Method	Cross Subject Accuracy	Cross View Accuracy
HOG^2	32.24%	22.27%
Super Normal Vector	31.82%	13.61%
HON4D	30.56%	7.26%
P-LSTM	62.93%	70.27%
BoMW	48.25%	36.18%

VI. CONCLUSION

In this paper, we propose a novel feature extraction method for gesture recognition, namely, bag of manifold words (BoMW), which uses covariance matrices to combine both RGB and depth features from local spatio-templral blocks. Since covariance matrices are SPD matrices, which spans a SPD manifold. We further propose a novel sparse coding method on SPD manifolds and encode covariance matrices as the final feature representation in a bag of word fashion. The nearest neighbour classifier is finally adopted to perform the one-shot learning gesture recognition. Experimental results

on the ChaLearn gesture dataset demonstrate the outstanding performance compared to several state-of-the-art methods. The effectiveness of the proposed feature extraction method is also validated on a new RGB-D action recognition dataset.

REFERENCES

[1] V. Pavlovic, R. Sharma, and T. Huang, "Visual interpretation of hand gestures for human- computer interaction: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 677–695, 1997. 1

[2] Y. Zhu, G. Xu, and D. Kriegman, "A real-time approach to the spotting, representation, and recognition of hand gestures for human-computer interaction," *Computer Vision and Image Understanding*, vol. 85, no. 3, pp. 189–208, 2002. 1

[3] C. Li, L. Lin, W. Zuo, W. Wang, and J. Tang, "An approach to streaming video segmentation with sub-optimal low-rank decomposition," *IEEE Trans. Image Processing*, vol. 25, no. 5, pp. 1947–1960, 2016. 1

[4] W. Zuo, P. Wang, and D. Zhang, "Comparison of three different types of wrist pulse signals by their physical meanings and diagnosis performance," *IEEE J. Biomedical and Health Informatics*, vol. 20, no. 1, pp. 119–127, 2016. 1

[5] A. Malima, E. Ozgur, and M. Cetin, "A fast algorithm for vision-based hand gesture recognition for robot control," in *IEEE Signal Processing and Communications Applications*, 2006, pp. 1–4. 1

[6] C. Shan, T. Tan, and Y. Wei, "Real-time hand tracking using a mean shift embedded particle filter," *Pattern Recognition*, vol. 40, no. 7, pp. 1958–1970, 2007. 1

[7] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, "Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification," *IEEE Trans. Image Processing*, vol. 24, no. 12, pp. 4766–4779, 2015. 1

[8] T. Starner, J. Weaver, and A. Pentland, "Real-time American sign language recognition using desk and wearable computer based video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1371–1375, 1998. 1

[9] W. Gao, G. Fang, D. Zhao, and Y. Chen, "A chinese sign language recognition system based on sofm/srn/hmm," *Pattern Recognition*, vol. 37, no. 12, pp. 2389–2402, 2004. 1

[10] S. Reifinger, F. Wallhoff, M. Ablassmeier, T. Poitschke, and G. Rigoll, "Static and dynamic hand-gesture recognition for augmented reality applications," in *Proceedings of the 12th International Conference on Human-computer Interaction: Intelligent Multimodal Interaction Environments*, 2007, pp. 728–737. 1

[11] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden markov models for complex action recognition," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 994–999, 1997. 1

[12] C. P. Vogler, "American sign language recognition: reducing the complexity of the task with phoneme-based modeling and parallel hidden markov models," Ph.D. dissertation, University of Pennsylvania, 2003. 1, 2

[13] 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. 1

[14] H. Suk, B. Sin, and S. Lee, "Hand gesture recognition based on dynamic bayesian network framework," *Pattern Recognition*, vol. 43, no. 9, pp. 3059–3072, 2010. 1

[15] Z. Ren, J. Yuan, and Z. Zhang, "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera," in *Proceedings of the 19th International Conference on Multimedia*, 2011, pp. 1093–1096. 1

[16] Y. M. Lui, "Human gesture recognition on product manifolds," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 3297–3321, 2012. 1, 3

[17] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using Kinect sensor," *IEEE Transactions on Multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013. 1, 2

[18] K. Wang, X. Wang, L. Lin, M. Wang, and W. Zuo, "3d human activity recognition with reconfigurable convolutional neural networks," *ACM Multimedia*, pp. 97–106, 2014. 1

[19] Z. Chen, W. Zuo, Q. Hu, and L. Lin, "Kernel sparse representation for time series classification," *Inf. Sci.*, vol. 292, pp. 15–26, 2015. 1

[20] J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1297–1304. 1

[21] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1290–1297. 1

[22] P. Doliotis, A. Stefan, C. Mcmurrough, D. Eckhard, and V. Athitsos, "Comparing gesture recognition accuracy using color and depth information," in *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, 2011, p. 20. 1

[23] 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. 1, 2, 7, 8, 9

[24] F. Jiang, S. Zhang, S. Wu, Y. Gao, and D. Zhao, "Multi-layered gesture recognition with kinect," *Journal of Machine Learning Research*, vol. 16, pp. 227–254, 2015. 1, 2, 8, 9

[25] I. Guyon, V. Athitsos, P. Jangyodsuk, and H. J. Escalante, "The chalearn gesture dataset (CGD 2011)," *Machine Vision and Applications*, vol. 25, pp. 1929–1951, 2014. 2, 7

[26] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3d human activity analysis," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 8

[27] G. Awad, J. Han, and A. Sutherland, "A unified system for segmentation and tracking of face and hands in sign language recognition," in *Proceedings of the 18th International Conference on Pattern Recognition*, vol. 1, 2006, pp. 239–242. 2

[28] M. Maraqa and R. Abu-Zaiter, "Recognition of arabic sign language (ArSL) using recurrent neural networks," in *Proceedings of the First International Conference on the Applications of Digital Information and Web Technologies*, 2008, pp. 478–481. 2

[29] A. Ramamoorthy, N. Vaswani, S. Chaudhury, and S. Banerjee, "Recognition of dynamic hand gestures," *Pattern Recognition*, vol. 36, no. 9, pp. 2069–2081, 2003. 2

[30] E.-J. Ong and R. Bowden, "A boosted classifier tree for hand shape detection," in *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, 2004, pp. 889–894. 2

[31] R. Cutler and M. Turk, "View-based interpretation of real-time optical flow for gesture recognition," in *Proceedings of the 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, 1998, pp. 416–416. 2

[32] U. Mahbub, T. Roy, M. Shafiur Rahman, and H. Imtiaz, "One-shot-learning gesture recognition using motion history based gesture silhouettes," in *Proceedings of the International Conference on Industrial Application Engineering*, 2013, pp. 186–193. 2, 3

[33] T. Agrawal and S. Chaudhuri, "Gesture recognition using motion histogram," in *Proceedings of the Indian National Conference of Communications*, 2003, pp. 438–442. 2

[34] L. Shao and L. Ji, "Motion histogram analysis based key frame extraction for human action/activity representation," in *Proceedings of Canadian Conference on Computer and Robot Vision*, 2009, pp. 88–92. 2

[35] M. Zahedi, D. Keysers, and H. Ney, "Appearance-based recognition of words in american sign language," in *Proceedings of Second Iberian Conference on Pattern recognition and image analysis*, 2005, pp. 511–519. 2

[36] S.-F. Wong, T.-K. Kim, and R. Cipolla, "Learning motion categories using both semantic and structural information," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–6. 2

[37] H. Cooper, B. Holt, and R. Bowden, "Sign language recognition," in *Visual Analysis of Humans*, 2011, pp. 539–562. 2

[38] 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*, pp. 344–349, 2012. 2, 8, 9

[39] M. Chen and A. Hauptmann, "Mosift: Recognizing human actions in surveillance videos," *Technical Report*, 2009. 2, 8, 9

[40] J. Wan, V. Athitsos, P. Jangyodsuk, H. J. Escalante, Q. Ruan, and I. Guyon, "CSMMI: Class-specific maximization of mutual information for action and gesture recognition," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 3152–3165, 2014. 2

[41] S. R. Fanello, I. Gori, G. Metta, and F. Odone, "One-shot learning for real-time action recognition," in *Pattern Recognition and Image Analysis*, 2013, pp. 31–40. 2

[42] D. Wu, F. Zhu, and L. Shao, "One shot learning gesture recognition from RGBD images," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012, pp. 7–12. 2

[43] M. Reyes, G. Dominguez, and S. Escalera, "Feature weighting in dynamic time warping for gesture recognition in depth data," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2011, pp. 1182–1188. 2

[44] J. F. Lichtenauer, E. A. Hendriks, and M. J. Reinders, "Sign language recognition by combining statistical DTW and independent classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 2040–2046, 2008. 2

[45] Y. Sabinas, E. F. Morales, and H. J. Escalante, "A One-Shot DTW-based method for early gesture recognition," in *Proceedings of 18th Iberoamerican Congress on Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, 2013, pp. 439–446. 2

[46] O. Al-Jarrah and A. Halawani, "Recognition of gestures in Arabic sign language using neuro-fuzzy systems," *Artificial Intelligence*, vol. 133, no. 1, pp. 117–138, 2001. 2

[47] M.-C. Su, "A fuzzy rule-based approach to spatio-temporal hand gesture recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 30, no. 2, pp. 276–281, 2000. 2

[48] C.-L. Huang and W.-Y. Huang, "Sign language recognition using model-based tracking and a 3D Hopfield neural network," *Machine Vision and Applications*, vol. 10, no. 5-6, pp. 292–307, 1998. 2

[49] S. Zhang, H. Yao, X. Sun, K. Wang, J. Zhang, X. Lu, and Y. Zhang, "Action recognition based on overcomplete independent component analysis," *Information sciences*, vol. 281, pp. 635–647, 2014. 2

[50] Z. Chen, W. Zuo, Q. Hu, and L. Lin, "Kernel sparse representation for time series classification," *Information Sciences*, vol. 292, pp. 15–26, 2015. 2

[51] X. Cao, W. Ren, W. Zuo, X. Guo, and H. Foroosh, "Scene text deblurring using text-specific multiscale dictionaries," *IEEE Trans. Image Processing*, vol. 24, no. 4, pp. 1302–1314, 2015. 2

[52] L. Zhang, W. Zuo, and D. Zhang, "Lsdt: Latent sparse domain transfer learning for visual adaptation," *IEEE Trans. Image Processing*, vol. 25, no. 3, pp. 1177–1191, 2016. 2

[53] C. Li, L. Lin, W. Zuo, W. Wang, and J. Tang, "An approach to streaming video segmentation with sub-optimal low-rank decomposition," *IEEE Trans. Image Processing*, vol. 25, no. 5, pp. 1947–1960, 2016. 2

[54] P. Zhu, Q. Hu, C. Zhang, and W. Zuo, "Coupled dictionary learning for unsupervised feature selection," *AAAI Conference on Artificial Intelligence*, pp. 2422–2428, 2016. 2

[55] B. Bauer and K.-F. Kraiss, "Video-based sign recognition using self-organizing subunits," in *Proceedings of the 16th International Conference on Pattern Recognition*, vol. 2, 2002, pp. 434–437. 2

[56] G. Fang, W. Gao, and D. Zhao, "Large vocabulary sign language recognition based on fuzzy decision trees," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 34, no. 3, pp. 305–314, 2004. 2

[57] J. Wan, Q. Ruan, G. An, and W. Li, "Gesture recognition based on hidden markov model from sparse representative observations," in *Proceedings of the IEEE 11th International Conference on Signal Processing*, vol. 2, 2012, pp. 1180–1183. 2

[58] H. J. Seo and P. Milanfar, "Action recognition from one example," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 867–882, 2011. 3

[59] 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. 3

[60] I. Guyon, V. Athitsos, P. Jangyodsuk, H. J. Escalante, and B. Hamner, "Results and analysis of the Chalearn gesture challenge 2012," in *Proceedings of International Workshop on Advances in Depth Image Analysis and Applications*, 2013, pp. 186–204. 3

[61] H. Li, H. Wu, S. Lin, L. Lin, X. Luo, and E. Izquierdo, "Boosting zero-shot image classification via pairwise relationship learning," in *Proc. Asian Conference on Computer Vision*, 2016, pp. 85–99. 3

[62] M. Varma and A. Zisserman, "Classifying images of materials: Achieving viewpoint and illumination independence," *Proc. European Conference on Computer Vision*, pp. 255–271, 2002. 3, 5

[63] J. M. Geusebroek, A. W. M. Smeulders, and J. van de Weijer, "Fast anisotropic gauss filtering," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 938–943, 2003. 3, 5

[64] X. Pennec, P. Fillard, and N. Ayache, "A riemannian framework for tensor computing," *Int. Journal of Computer Vision*, vol. 66, no. 1, pp. 41–66, 2006. 3, 4

[65] S. Sra, "A new metric on the manifold of kernel matrices with application to matrix geometric means," *Proc. Advances in Neural Information Processing Systems*, pp. 144–152, 2012. 4

[66] D. P. Bertsekas, *Nonlinear programming*. Athena Scientific, 1999. 6

[67] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: a comprehensive study," *International Journal of Computer Vision*, vol. 73, no. 2, pp. 213–238, 2007. 7

[68] X. P. V. Arsigny, P. Fillard and N. Ayache, "Geometric means in a novel vector space structure on symmetric positive-definite matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 29, no. 1, pp. 328–347, 2007. 8

[69] A. Kurakin, Z. Zhang, and Z. Liu, "A real time system for dynamic hand gesture recognition with a depth sensor," in *Proceedings of the 20th European Signal Processing Conference*, 2012, pp. 1975–1979. 8

[70] S. Fothergill, H. Mentis, P. Kohli, and S. Nowozin, "Instructing people for training gestural interactive systems," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 1737–1746. 8

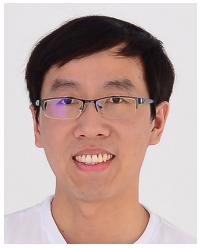
[71] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008. 9

[72] E. Ohn-Bar and M. Trivedi, "Joint angles similarities and hog^2 for action recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2013. 9

[73] X. Yang and Y. Tian, "Super normal vector for activity recognition using depth sequences," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014. 9

[74] O. Oreifej and Z. Liu, "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2013. 9

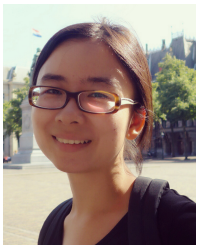
Lei Zhang received the Ph.D. degree in computer science from Harbin Institute of Technology, Harbin, China, in 2013. From September 2011 to August 2012, He was a research intern in Siemens Corporate Research, Inc., Princeton, NJ. From July 2015 to September 2016, He was a Post-Doctoral Research Fellow in College of Engineering, Temple University, PA. He is currently a Post-Doctoral Research Fellow in Department of Biomedical Informatics, Arizona State University, Scottsdale, AZ. He is also a lecturer in School of Art and Design, Harbin University, Harbin, China. His current research interests include machine learning, computer vision, visualization and medical image analysis.



Shengping Zhang received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2013. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology, China. He had been a post-doctoral research associate with Brown University, a post-doctoral research associate with Hong Kong Baptist University, and a visiting student researcher with University of California at Berkeley. He has authored or co-authored over 50 research publications in refereed journals and conferences. His research interests include deep learning and its applications in computer vision. Dr. Zhang is also an Associate Editor of Signal Image and Video Processing.

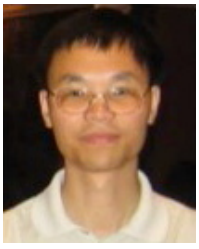
Feng Jiang received the B.S., M.S., and Ph.D. degrees in computer science from Harbin Institute of Technology (HIT), Harbin, China, in 2001, 2003, and 2008, respectively. He is now an Associated Professor in the Department of Computer Science, HIT. He had been a visiting scholar in the School of Electrical Engineering, Princeton University, United States. His research interests include computer vision, image and video processing and pattern recognition.

Yuankai Qi received the B.S. and M.S. degrees from Harbin Institute of Technology, China, in 2011 and 2013, respectively. He is currently working toward his Ph.D. degree in the School of computer science and technology, Harbin Institute of Technology, China. His research interests include object tracking, sparse coding, and machine learning.



Jun Zhang received her B.S., M.S, and PhD degrees in School of Computer Science and Information Engineering from Hefei University of Technology (HFUT), China in 2007, 2009 and 2013, respectively. From August 2010 to September 2012, and May 2015 to September 2015, she worked at the Department of Cognitive, Linguistic & Psychological Sciences (CLPS) at Brown University as a visiting research fellow. From July 2013 until now, she is a postdoctoral fellow in computer science, and also acts as a associate professor in School of Computer Science and Information Engineering of HFUT. Her current interests include computer vision, vision perception, and pattern recognition.

Yuliang Guo received his B.S.M.S degree in material science from Shanghai Jiaotong University, Shanghai, China, in 2009, M.S.C.E in computer engineering from Brown University, Rhode Island, USA in 2011, and is currently a Ph.D candidate in computer engineering at Brown University. His research focus on computer vision, specifically in exploiting graphical models integrated with geometric constraints, and applying to edge detection, contour extraction, animal pose detection and behavior analysis.



Huiyu Zhou obtained a Bachelor of Engineering degree in Radio Technology from the Huazhong University of Science and Technology of China, and a Master of Science degree in Biomedical Engineering from the University of Dundee of United Kingdom, respectively. He was then awarded a Doctor of Philosophy degree in Computer Vision from the Heriot-Watt University, Edinburgh, United Kingdom. Dr. Zhou is an assistant professor at School of Electronics, Electrical Engineering and Computer Science, Queens University of Belfast, United Kingdom. He has published over 130 peer-reviewed papers in the field. His research work has been or is being supported by UK EPSRC, EU, Royal Society, Leverhulme Trust, Puffin Trust, Invest NI and industry.