Multimed Tools Appl (2018) 77:14115–14135
DOI 10.1007/s11042-017-5017-y

Supervised spatio-temporal kernel descriptor for human action recognition from RGB-depth videos

Maryam Asadi-Aghbolaghi[1] · Shohreh Kasaei[1]

Received: 9 January 2017 / Revised: 26 April 2017 / Accepted: 6 July 2017 /
Published online: 21 July 2017

Abstract One of the most challenging tasks in computer vision is human action recognition. The recent development of depth sensors has created new opportunities in this field of research. In this paper, a novel *supervised spatio-temporal kernel descriptor* (SSTKDes) is proposed from RGB-depth videos to establish a discriminative and compact feature representation of actions. To enhance the descriptive and discriminative ability of the descriptor, extracted primary kernel-based features are transformed into a new space by exploiting a supervised training strategy; i.e., *large margin nearest neighbor* (LMNN). The LMNN highly reduces the error of a nearest neighbor classifier by minimizing the intra-class variations and maximizing the inter-class distances. Subsequently, the *efficient match kernel* (EMK) is used to abstract the mid-level kernel features for a more efficient classification. The proposed approach is evaluated on five public benchmark datasets. The experimental evaluations demonstrate that the proposed method achieves superior performance to the state-of-the-art methods.

Keywords Action recognition · Supervised kernel descriptor · RGB-D video · LMNN · EMK

1 Introduction

From the early beginning of computer vision, human action analysis has been one of the significant research topic, due to the wide real-world applications in various fields like health

✉ Shohreh Kasaei
skasaei@sharif.edu

Maryam Asadi-Aghbolaghi
masadia@ce.sharif.edu

[1] Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

and medicine, sports and recreation, content-based video search, robotics and other systems which involve interactions between humans and electronic devices.

In past decades, research of human action recognition has mainly focused on recognizing actions from videos captured by traditional visible light cameras. The recent advent of low-cost and easy-operation depth sensors (like Kinect) have received a great deal of attention from researchers to reconsider problems such as activity recognition using depth images alongside color images. Compared with conventional RGB cameras, RGB-D cameras provide several advantages. First, depth cameras are insensitive to illumination changes (it can produce depth images even in a total darkness). Next, more discriminative information (like 3D geometric structural data of the scene) can be extracted from depth maps. Moreover, 3D positions of skeletal joints can be estimated from depth images quickly and accurately. Although the estimated skeleton brings benefits to activity recognition, some shortcomings limits its usage. For instance, the estimation is unreliable or even failed when the human is partly in view or touches background [50].

A substantial task in human action recognition is designing an efficient, compact, and robust video representation, despite the presence of challenging conditions. First, depth sensors (like Kinect) usually generate potentially noisy depth maps due to some special reflectance materials, fast movements, and porous surfaces. Second, there are significant intra-class variations in human action recognition on account of execution rates, personal styles, and different viewpoints. Next, overlaps among different action categories make characterizing the inter-class dissimilarities very difficult.

Most recent approaches recognize actions by constructing a histogram of descriptors of *spatio-temporal interest points* (STIP) in videos. The results of these approaches have been promising on RGB dataset; however the extension of these methods for depth images cannot be optimal, since depth images are much noisier than the RGB ones. For instance, undefined depth points appear as black regions on the surface of human body in depth images, therefore many interest point detectors falsely fire on these noisy regions [30]. In addition, almost all the hand-crafted feature descriptors are unsupervised. As such, they are barely able to handle inter- and intra-class variations for human action recognition.

To address the aforementioned challenges and design a more discriminative descriptor, in this paper, SSTKDes is proposed for human action recognition from the RGB-depth data. A general overview of the proposed method is depicted in Fig. 1. Note that to deal with noisy depth data, smoothing filters are first applied on depth videos.

It has been shown that descriptors based on low-level pixel attributes work fine on both RGB and depth images for object [5] and action recognition [22]. To properly characterize the spatio-temporal structure of actions and provide a discriminative descriptor, a primary rich pixel attribute is needed to be extracted from RGB-D videos. For action recognition, the descriptor should capture both shape and motion information. In this paper, the 3D (i.e., spatio-temporal) gradient is utilized as the primary attribute for both RGB and depth

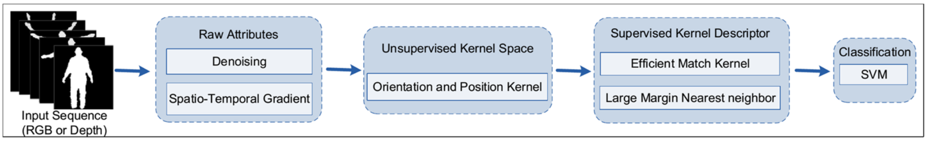

Fig. 1 General overview of the proposed method for action recognition

videos. Then, following the kernel descriptor proposed for visual recognition [5], the spatio-temporal attributes extracted from each pixel are transformed into a compact unsupervised kernel space.

Moreover, an efficient approach for coping with intra-class variations and inter-class similarities would be calculating the projections that can actively discriminate among classes; i.e., a supervised descriptor. Exploiting video labels for designing the feature descriptor can yield the method to achieve a more accurate, robust, efficient, and discriminative feature representation. Therefore, in the next step, a supervised strategy is utilized to transfer non-linear video features into a more discriminative feature space, motivated by [43]. For this transformation, a combination of LMNN [47] and EMK [6] is utilized.

The goal of LMNN is to find a supervised transformation of input space such that in the new space, the k-nearest neighbors have matching labels while samples from other classes are separated by large margins. Then, by using a convex optimization based on the hinge loss, this margin criterion is solved. From another sight, it can be beneficial to solve the problems of intra-class variations and inter-class similarities in human action recognition by transforming data into a new space in which samples with the same labels are closer to each other than those with different labels. The EMK is a kernel representation of well-known bag-of-words method. It has been proved to produce more accurate quantization and also learn nonlinear correlations among body parts in human action recognition [22].

2 Related work

Various RGB-based action recognition methods have been published in the literature [1, 31, 48]. Most of the methods extract STIP [24] and use the distribution of local features, like *histogram of optical flow* (HOF) [25] and *histogram of gradient* (HOG) [12], to represent the spatio-temporal patterns. During recent years, human action recognition based on 3D perception data has been wildly grown. Based on the representation of 3D data sequences, the methods can be divided into three main groups of skeleton data, point cloud, and depth map. Several surveys [2, 18, 56] have also been published in this regard.

In skeleton-based methods, the 3D position of body joints are utilized for action recognition. The 3D location of joints can be captured by multi-camera *motion capture* (MoCap) systems. Although this data is very accurate and almost free of noise, it is marker-based and therefore very expensive and difficult to be produced. On the other hand, some approaches exploit the position of human joints provided by Shotton et al. [34] which extracts human skeleton from depth map in real-time. The features used in this group are based on 3D joint locations [10, 33], relative position of joints from a reference joint or from each other [42, 54], angles between connected parts [16, 36], velocity of joints [46, 54], and joints trajectories along temporal dimension [17, 20, 37]. The estimated body joints from depth map are quiet accurate in experimental settings; however, their usage is limited to some especial cases. The situations in which the occlusion or self-occlusion occurs, a person touches the background, or a person is not in an upright position [50], makes the process of estimating the 3D position of joints very difficult or even impossible.

A point cloud is a collection of points in the 3D coordinate system. The point cloud can be acquired fully (or partially) by 3D reconstruction methods from multi-views or depth maps. The methods in the second group extract global or local features from the point cloud of human body [3, 4]. The point cloud of human body reveals important cues for recognizing actions. Although it can result in discriminative descriptors, representing actions based on the point cloud requires more computational cost.

By using the point cloud of human body from depth images, Jiang Wang et al. [41] proposed the *random occupancy pattern* (ROP). In this method, a sparse subset of the most discriminative sub-volumes, obtained from the whole 4D volume of human body during the action period, are selected by an Elastic-Net regularization method. The *depth motion map* (DMM), used in several methods [11, 44, 55], utilizes the point cloud computed from the depth map. To produce DMM, the point cloud of a human body is projected onto three orthogonal Cartesian views. Then, the global activity of the entire video sequences is accumulated on these planes. To classify actions, different features like HOG [55] and *local binary pattern* (LBP) [11] have been extracted from DMM. Vieira et al. [40] defined a 4D grid for a sequence of depth map by dividing the space and time axes into multiple cells. These cells typically consist of points on the silhouette or moving parts of the body. Then, they enhanced the roles of sparse cells by using a saturation scheme. Oreifej and Liu [30] proposed the *histogram of oriented 4D normals* (HON4D) descriptor based on the distribution of 4D normal vectors in some spatio-temporal cells of actions. Yang and Tian [53] proposed the *super normal vector* (SNV). It was a sparse dictionary-based method of low-level polynormals in which each poly-normal was calculated by clustering hypersurface normal vectors in each spatio-temporal neighborhood.

The depth map-based methods (the third group), usually use features, either local or global, which are extracted from a consecutive space-time volume. Xia and Aggarwal [50] extracted *depth-based spatio temporal interest points* (DSTIP) using a response function of spatio-temporal filtering. Then, *depth cuboid similarity feature* (DCSF) was proposed for describing the local 3D cuboid (x,y,t) with adaptable size centered at DSTIP. Lu et al. [29] proposed a descriptor for depth maps which was an extension of binary feature descriptor used in RGB video [9]. After partitioning the depth maps into three layers (named background, activity, and occlusion layers), features were extracted from some spatio-temporal local 3D cubes from the activity layer in depth sequences.

Due to its success in various classification tasks, many researches have utilized deep-learned features in action recognition from RGB [38, 45], depth [44], and skeletal data [14]. The *recurrent neural network* (RNN) [14, 27], *3D convolutional neural network* (3DCNN) [49], and 2D CNN with some primary motion features (e.g., DMM) [44] as their inputs are the most used networks. Among these methods, 3DCNN and RNN can deal with temporal information. The 3D convolution and 3D pooling layers of 3DCNN models allow capturing discriminative features along both spatial and temporal dimensions. The RNN takes into account the temporal data using recurrent connections in hidden layers. The original CNN network can only cope with images instead of videos. Simonyan and Zisserman [35] randomly sampled a fixed number of frames from a video, and then applied CNN on every individual frame. Finally, they used the average scoring of selected frames for classification. In another work, Yu et al. [57] extracted features from CNN for all frames of one video and then applied pooling on the frame-level features to get video-level features.

Action recognition has not gained a high performance from deep networks compared with other research areas (like image classification) [15]. It might be related to the fact that network performance is dependent on a large number of weights that have to be learnt from a large annotated data (like ImageNet), which is not currently available for action recognition purposes. In addition, such enormous data cannot be provided by many real-world problems [21]. Therefore, there is a need for methods that can achieve a high performance with small amount of data.

In some cases, for human action recognition, to cope with data from different sources, the *multiple kernel learning* (MKL) is used to combine the kernels which are established for each individual data source. For human action recognition, Xiao et al. [52] used MKL to select the most discriminative kernels in the function of composing of different kernels by setting weights for each kernel. In that work, the authors took advantage of the *Bacterial Chemotaxis* (BC) and the Powell optimization methods to find the weight of each kernel. The powerful local optimization ability of the Powell method is adopted to improve the local search precision of BC.

Bayesian logistic regression is also another framework for classification. To have more flexibility, variational transformations are used in order to approximate the likelihood function with a simpler and tractable exponential form by means of introducing extra variables known as variational parameters. To deal with the regression of several classes, variational Bayesian multinomial logistic regression has been proposed [19].

Among available algorithms for action recognition, only deep-models use the labels for producing features. To the best of our knowledge, all proposed hand-crafted descriptors for human action recognition are unsupervised. In object recognition, it has been demonstrated that supervised techniques (like *linear discriminate analysis* (LDA)), improve the result of *scale-invariant feature transform* (SIFT) [8].

The *kernel descriptor* (KDES) utilized the unsupervised *kernel principal component analysis* (KPCA) to learn a compact descriptor. A 3D extension of KDES was proposed for action recognition [22]. It achieved superior performance on RGB-D datasets. Wang et al. [43] proposed a supervised extension of kernel descriptor, called *supervised kernel descriptor* (SKDES), for objection recognition from RGB data. They took the advantage of KDES to design a low-level feature descriptor.

In this paper, SSTKDes is proposed for human action recognition using RGB-D data. The low-level attributes are spatio-temporal features extracted from RGB and depth videos. The attribute space is then transferred into a new space by a non-linear compact kernel-based transformation using a supervised process. Finally, the SVM classifier is applied on the final feature descriptor.

3 Proposed method

The aim of this paper is to design a global descriptor for human action recognition from RGB-D data. An overview of the proposed method is illustrated in Fig. 1. The input data is a sequence of RGB-D images in which just one person is in the scene and the person performs one action. The background in these datasets can be plain or textured (in some datasets the background has been subtracted).

First, hierarchy-levels are defined for each video. The hierarchy of three levels of video is shown in Fig. 2. The first level is the action *volume* which covers all spatio-temporal dimensions of the input sequence. The action volume is divided into some sub-volumes, called *part*. Then, each part is divided into smaller units, called *3D blob*. Each 3D blob contains a cubic spatio-temporal data (i.e., pixel level) in RGB or depth video.

The contribution of this work is first explained in Section 3.1. The denoising step is then presented in Section 3.2. Next, in Section 3.3 it is explained that how raw attributes for all pixels are transformed into an unsupervised kernel feature space. Finally, details on the supervised video descriptor are provided in Section 3.4.

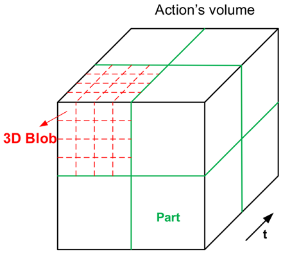

Fig. 2 Hierarchy levels of video

3.1 Contributions of this work

This work differs from the existing approaches as follow. The KDES [5] and CKSVD (EMK) [6] methods are unsupervised kernel descriptors proposed for object recognition. The HKDES [22] is a 3D hierarchical extension of KDES for human action recognition. Those methods are based on unsupervised kernel features while in this method a supervised kernel descriptor is proposed. Moreover, LMNN [47] is a supervised strategy that has been used for image, speech, and text classification. In the proposed method, LMNN is used for action recognition. The proposed method is particularly designed to extract rich information from both RGB and depth data for human action recognition while the supervised kernel method in [43] is designed for object recognition from only RGB data.

The key contributions of this work can be summarized as follows:

- A novel hierarchical framework for feature extraction of video is proposed. This structure is capable of discarding the irrelevant details while preserving the task-relating important feature.
- To reduce the effect of two kinds of noise (i.e., small variations of sensing device and undefined depth points) in (not highly noisy) depths map, a spatio-temporal smoothing function is used.
- First, a dictionary (generated through EMK) is exploited to encode the features. Then, the LMNN is utilized as a supervised learning plan to create a margin of safety around the kNN decision boundaries that separates videos with different labels.
- The proposed method is evaluated on five public action and gesture recognition datasets, and has achieved the state-of-the-art results on four datasets.

3.2 Denoising

Xia and Aggarwal [50] divided the noise in depth videos into three categories of: noise from variations in sensing devices, boundary of agents, and holes caused by: special reflectance materials, fast movements, porous surfaces, and other random effects. The first group can be reasonably removed by smoothing functions that consist of two separate spatial and temporal filters. Following [50], Gaussian filter is utilized as spatial smoothing function; which is a kind of low-pass filter that can remove high-frequency components in depth images. Mathematically, Gaussian smoothing is obtained by convolving the input signal (depth image) with a Gaussian function, as

$$\begin{aligned} d_s &= d(x, y, t) * g(x, y; \sigma) \\ g(x, y; \sigma) &= \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \end{aligned} \tag{1}$$

in which, $d(x, y, t)$ is the input depth map at time t, g is the 2D Gaussian function with standard deviation σ which controls the spatial scale, and d_s is the smoothed depth image. Also, the 1D complex Gabor filter is defined as

$$\begin{aligned} d_{st} &= (d_s(x, y, t) * h_{ev})^2 + (d_s(x, y, t) * h_{od})^2 \\ h_{ev}(t; \tau, \omega) &= \cos af;\ (2\pi \omega t)\ \exp af;\ (-t^2/\tau^2) \\ h_{od}(t; \tau, \omega) &= \sin af;\ (2\pi \omega t)\ \exp af;\ (-t^2/\tau^2) \end{aligned} \tag{2}$$

where τ controls the temporal scale and ω is usually used as a constant value related to τ (e.g., $\frac{0.6}{\tau}$). In the rest of this paper, d denotes the smoothed d_{st}.

In general, the result of smoothing filters is setting the value of each pixel to the weighted average of itself and its neighbors. Therefore, these filters set the value of each pixel into the closer harmony with the values of its neighbors. From another sight, noisy pixels with significantly higher or lower intensity than surrounding neighborhood, will be smoothed. Consequently, this method can approximately remove small variation of sensing device and holes in (not highly noisy) depths map. Xia and Aggarwal [50] defined a measure of temporal value variation (i.e., correction function) for each pixel to remove the second and third group of noise (namely, boundary of agents, and holes) in highly noisy depth map. But, in this paper, this function is not necessary. Their work is an STIP-based method, in which a response function is used to find interest points. These functions usually have large values in noisy pixels. As such, noisy points might be falsely selected as interest points. Fortunately, the proposed descriptor is not based on interest points. It uses a pooling function of attributes of all pixels in a video. Therefore, all points have the same weight to calculate the descriptor. Consequently, it reduces the effect of noise in the final descriptor.

3.3 Unsupervised kernel space

Some methods (SIFT and HOG), compute the histogram of orientation-based attributes of all pixels in small windows as the feature descriptor by quantizing the individual pixel attribute value into some bins. It is obvious that the quantization error decreases the accuracy of those methods. To overcome this problem, KDES [5] has been proposed to generate rich feature descriptors from pixel attributes for object recognition. It can capture more descriptive information lying in high dimensional space, compared to the SIFT and HOG.

By using Euclidean distance for measuring dissimilarity between two 3D blobs, as

$$\begin{aligned} Dist(B_1, B_2) &= (F(B_1) - F(B_2))^T (F(B_1) - F(B_2)) \\ &= 2 - 2\ F(B_1)^T F(B_2) \end{aligned} \tag{3}$$

it can be shown that the kernel view of two 3D blobs and the similarity of them are directly proportional. In [6], to compute the similarity between two blobs of two images, the match kernel is utilized which is a kernel function that averages over the continuous similarities among all pixel attributes in two blobs. In this paper, a 3D extension of the kernel representation of orientation histogram, in an unsupervised space, is used as the first step of the feature extraction process. The low-level pixel attribute exploited here is the 3D gradient for both RGB and depth videos which can capture the shape changes alongside both spatial and temporal dimensions. Following the formulation in [5], the match kernel between two 3D blobs of two action videos is calculated by

$$K_{3D}(B_1, B_2) = \sum_{z \in B_1} \sum_{z' \in B_2} \widetilde{m}(z)\ \widetilde{m}(z')\ k_o(\widetilde{\theta}_z, \widetilde{\theta}_{z'})\ k_p(z, z') \tag{4}$$

where B_1 and B_2 are two 3D blobs of two actions, $\widetilde{m}(z) = {}^{m(z)}/_{\sqrt{\sum_{z\in P} m(z)^2+\epsilon}}$ is the normalized 3D gradient magnitudes of z, $\widetilde{\theta}_z$ is the orientation of 3D gradient of z, $k_o(z, z') = \exp\left(-\gamma_o \,\|\widetilde{\theta}_z - \widetilde{\theta}_{z'}\|^2\right)$ is the Gaussian kernel of orientations of two pixels (which computes the similarity of these orientations), and $k_p(z, z') = \exp(-\gamma_p \,\|z - z'\|^2)$ is the Gaussian kernel of the 3D position of pixel in 3D blob (i.e., z, and measures the closeness of two pixels in spatio-temporal manner). By decomposing each Gaussian kernel into the inner product of two functions as $k_o(z, z') = \phi_o(\widetilde{\theta}_z)^T \ \phi_o(\widetilde{\theta}_{z'})$ and $k_p(z, z') = \phi_p(\widetilde{\theta}_z)^T \ \phi_p(\widetilde{\theta}_{z'})$, the feature representation for one 3D blob can be derived as

$$F_{3D}(B_1) = \sum_{z\in B_1} \widetilde{m}(z) \ \phi_o(\widetilde{\theta}_z) \otimes \phi_p(z) \tag{5}$$

in which $\otimes$ is the Kronecker product. Since the Gaussian kernel is used, the dimension of F_{3D} is infinite. For computational efficiency and for representational convenience, Bo et al. [5] provided a method to learn compact low dimensional features from match kernels. To do that, first a set of sufficient basis vectors, $\{x_i\}_{i=1}^{d_o}$ and $\{y_j\}_{j=1}^{d_p}$, are uniformly and densely sampled from the support regions of orientation and position, respectively. Then, the size of joint basis vectors is reduced using KPCA. As such, the final 3D kernel feature is obtained by

$$F^u_{3D}(B_1) = \sum_{i=1}^{d_o}\sum_{j=1}^{d_p} \alpha^u_{ij} \left\{ \sum_{z\in B_1} \widetilde{m}(z) \ k_o(\widetilde{\theta}_z, x_i) k_p(z, y_j) \right\} \tag{6}$$

where $\{x_i\}_{i=1}^{d_o}$ and $\{y_j\}_{j=1}^{d_p}$, are the basis vectors sampled from the support regions of orientation and position features, respectively, and d_o and d_p are the size of these basis vectors, α^u_{ij} is the $u^{th}(u = 1, ..., d_o \times d_p)$ projection coefficient computed by applying KPCA to the joint basis vectors $\left\{\phi_o(x_1) \otimes \phi_p(y_1), ..., \phi_o(x_{d_o}) \otimes \phi_p(y_{d_p})\right\}$.

3.4 Supervised kernel descriptor

By rewriting the obtained 3D blob features from the KDES into the vector form, we get $F(B) = A^T K$, where $A = \left[\boldsymbol{\alpha}^1, ..., \boldsymbol{\alpha}^{d_o \times d_p}\right]$ and each $\boldsymbol{\alpha}^u = [\alpha^u_1, ..., \alpha^u_{d_o\times d_p}]^T$ denotes the u_{th} principal components (kernel transform) in (6) and $K = \sum_{z\in B} \ \widetilde{m}(z) \left[k_o(\widetilde{\theta}_z, x_1) k_p(z, y_1), ..., k_o(\widetilde{\theta}_z, x_{d_o}) k_p(z, y_{d_p})\right]^T$. Note that the KPCA used in the previous session is an unsupervised transform while here, a supervised spatio-temporal kernel-based method is used for learning α^u_{ij}, to transfer non-linear video features into a more discriminative space. The proposed SSTKDes is depicted in Fig. 3. Each step is explained in the following sub-sections.

3.4.1 Hierarchy of feature levels

Features of action volume are obtained by concatenating the features of its parts, where the part feature is formed by average pooling of the encoded 3D blob features within it. The feature of each 3D blob is calculated by sum pooling of pixels included in it.

Pooling efficiently transfers from the feature representation to a new space, such that irrelevant details are discarded while task-relating important features are preserved [7]. Consequently, it significantly reduces the computational complexity and makes the representation compact. It is also used to achieve robustness to noise and invariance to the speed of the action. Sum pooling of features over a local neighborhood (i.e., 3D blob) reduces the

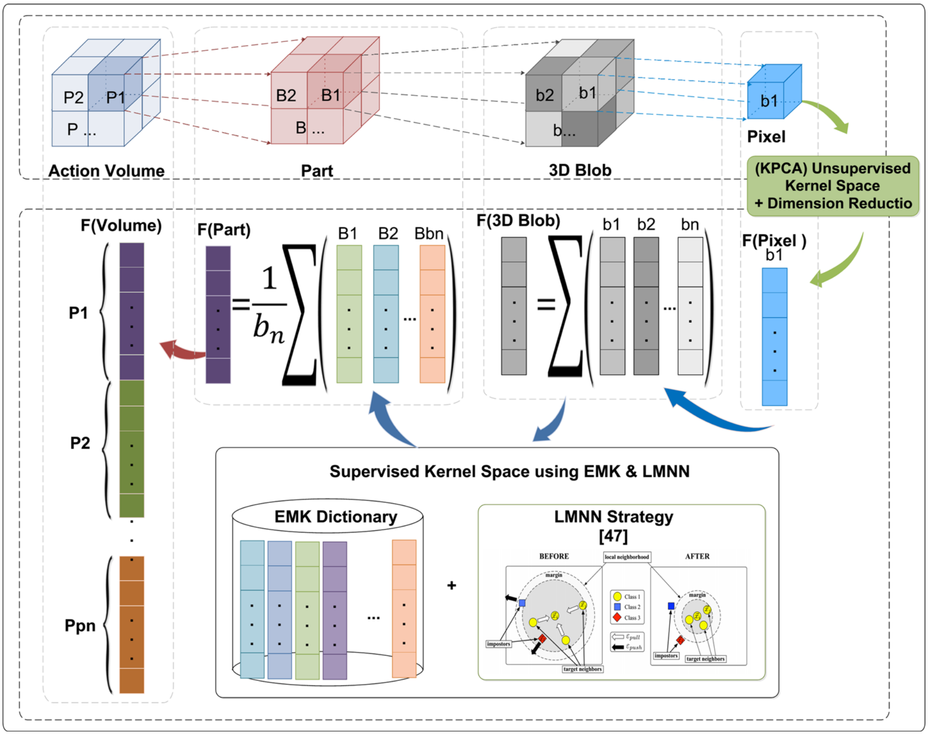

Fig. 3 Supervised spatio-temporal kernel descriptor (SSTKDes) algorithm

effect of noisy features. In addition, average pooling over 3D blobs contained in each part can yield the method to be invariant to the speed of the action, to certain extend. For the same actions with different speeds, the entire action volume is divided into the same number of parts. If there is no significant speed difference, corresponding parts contain almost the same information, but with different speeds. From another view point, different speeds can be expressed as different numbers of frames in each part. Thus, average pooling in each part results in almost the same information for the corresponding parts with different speeds.

To have a more accurate quantization process, the features of 3D blobs are encoded using a learnt dictionary generated through the *constrained singular value decomposition in kernel feature space* (CKSVD) [6] over 3D blob features. Thus, the part feature, which is a pooling of the encoded patch features, is

$$F_{P_s} = Pool_{b=1}^{|P_s|}\, g(C, A^T k_{bs}) \tag{7}$$

where P_s is the s^{th} part of the action volume, $|P_s|$ is the number of 3D blobs contained in the s^{th} part, F_{P_s} is the feature vector of the s^{th} part of the action volume, g is the encoding function, $Pool$ is a pooling operator, k_{bs} is the kernel feature of the b^{th} 3D blob contained in the s^{th} part, and C is the dictionary. In this paper, average pooling is used as the $Pool$ operator and ridge regression is used as the encoding function g. Dictionary C can be considered as $C = \left(A^T_{D\times D} F_{D\times\widetilde{b}}\right) Z_{n\times\widetilde{b}}^{\;T}$, where A is the matrix for transferring to the unsupervised space (defined in Section 3.4), F contains a set of $\widetilde{b}$ of 3D blob-level kernel features which are sampled over the whole 3D blob features, and Z is a matrix that transforms features to the dictionary space.

Therefore, if the encoded feature vector is $g(C, A^T k_{bs}) = c_{bs}$, then $F_{P_s} = \frac{1}{|P_s|}\sum_{b=1}^{|P_s|} c_{bs}$. Since the ridge regression is used to encode the 3D blob, the code c_{bs} has a closed-form solution of

$$\begin{aligned} c_{bs}^{*} &= \arg\min_{c_{bs}} \|A^T k_{bs} - C c_{bs}\|_2^2 + \mu \|c_{bs}\|_2^2 \\ &= (C^T C + \mu I)^{-1} C^T (A^T k_{bs}) \\ &= (Z F^T A A^T F Z^T + \mu I)^{-1} Z F^T A A^T k_{bs}. \end{aligned} \tag{8}$$

By setting $\mu > 0$, $C^T C + \mu I$ will be a positive definite matrix. As such, for each action volume, the feature vector F_V is obtined by

$$F_V = \cup_{s=1}^{PN}\left[F_{p_s}\right] = \cup_{s=1}^{PN}\left[\frac{1}{|P_s|}\sum_{b=1}^{|P_s|} c_{bs}\right] \tag{9}$$

where PN is the number of parts included in the action's volume and $\cup$ denotes a concatenation operator.

3.4.2 Supervised learning plan

The goal of this step is to find a compact kernel-based transformation in which a margin of safety is created around the kNN decision boundaries that separate samples with different labels; and consequently the k-nearest neighbors will always belong to the same class. In other words, the inter-class distance is maximized and the intra-class variation is minimized. To reach this goal, a supervised strategy, LMNN [47], is used.

For each sample of the action volume v_i, two kinds of neighbors are identified. Target neighbors, shown with $\mathcal{TN}(v_i)$, are k_p nearest neighbors of v_i with the same label. Target neighbors are desired to be the closest neighbors of v_i in the new space. In the current space, there might be some (k_n) differently labeled samples which are closer to the sample than the target, called impostors. These are shown by $\mathcal{IM}(v_i)$. The aim of the LMNN is to transform the feature space in such a way that the number of impostors is minimized. This is done by enlarging the safety margin around the kNN decision boundaries.

The loss function proposed for LMNN has two constraints. One constraint for penalizing the distance between each sample and its target neighbors to pull them closer to the sample (i.e., decreasing intra-class variations). The other constraint for penalizing the small distance between each sample and its impostors to push them further (i.e., increasing inter-class distances). In [43], to avoid overfitting and also to make the descriptor compact, a rank regularization is added to the loss function. The loss function used in this paper is

$$\begin{aligned} \min_A E(A) = &\sum_{v_i \in train}\left(\lambda \sum_{v_j \in \mathcal{TN}(v_j)} d_{ij} + \sum_{v_l \in \mathcal{IM}(v_j)} \sum_{v_j \in \mathcal{TN}(v_j)} \left[1 + d_{ij} - d_{il}\right]_+\right) \\ &+ \lambda_* \|A\|_* \end{aligned} \tag{10}$$

in which v_i is one sample of the training data, d_{ij} and d_{il} are distances between one sample with its target neighbors and its impostors, respectively, $[d]_+ = max\{d, 0\}$ is the hinge loss function, and $\|A\|_*$ is the nuclear norm of matrix A which is a convex surrogate of

$rank(A)$. By substituting the feature vector of action volume (9) in $d_{ij} = \|F_{V_i} - F_{V_j}\|_2^2 = (F_{V_i} - F_{V_j})^T(F_{V_i} - F_{V_j})$, we get

$$\begin{aligned} d_{ij} &= \sum_{s=1}^{PN} \left(k_{si} - k_{sj}\right)^T \mathbf{LL}^T \left(k_{si} - k_{sj}\right) \\ \mathbf{L} &= AA^T FZ^T (ZF^T AA^T FZ^T + \mu I)^{-1} \\ k_{si} &= \frac{1}{|P_s|} \sum_{b=1}^{|P_s|} k_{bs} \end{aligned} \tag{11}$$

in which k_{bs} are kernel features of the b^{th} 3D blob within the s^{th} part of the i^{th} video, and k_{si} are the kernel features of the s^{th} part of the i^{th} video. For $\mathbf{M} = \mathbf{LL}^T$, it can be shown [43] that $rank(\mathbf{M}) = rank(A)$. Thus, this minimization can be performed over the convex cone of positive semi-definite matrices $\mathbf{M}$. Therefore, the final convex version of the loss function, which is now a function of $\mathbf{M}$, is

$$\begin{aligned} \min_{\mathbf{M}} E(\mathbf{M}) = & \sum_{v_i \in train} \left(\lambda \sum_{v_j \in \mathcal{TN}(v_j)} d_{ij} + \sum_{v_l \in \mathcal{IM}(v_j)} \sum_{v_j \in \mathcal{TN}(v_j)} \left[1 + d_{ij} - d_{il}\right]_+ \right) \\ & + \lambda_* \mathrm{Tr}(\mathbf{M}) \\ & s.t.\ \mathbf{M} \succeq 0 \ and \ d_{ij} = \sum_{s=1}^{PN} (k_{si} - k_{sj})^T \mathbf{M} (k_{si} - k_{sj}) \end{aligned} \tag{12}$$

in which $\mathbf{M}$ is a semi-definite matrix. This optimization can be solved by using gradient-based algorithms. The complexity of each iteration is of order $O(ND_v k_p k_n)$, in which N is the size of the training data, D_v is the dimensionality of each video, and k_p and k_n are the average number of target neighbors and impostors, respectively. Since it is difficult to perform a general batch optimization with this complexity with the whole training data, a stochastic optimization is used to optimize this loss function. In fact, the *regularized dual averaging* (RDA) is used, which is generic and applicable to non-smooth losses (like hinge loss). (More details on the optimization process can be found in [43]).

3.4.3 Second kernel

If c_i and c_j are two encoded 3D blob features, then by applying the second kernel function (i.e., a *radial basis function* (RBF)) between a pair of encoded features, $k_M(c_i, c_j) = \exp(-\gamma_m (c_i - c_j)^T (c_i - c_j))$ is obtained. By setting the value of c_i and c_j, based on (8), in the second kernel function, then k_M is achieved as

$$k_M(c_i, c_j) = \exp(\gamma_m (k_i - k_j)^T \mathbf{M} (k_i - k_j)) \tag{13}$$

in which γ_m is the kernel parameter. Thus, the final features of each action volume can be calculated using $\mathbf{M}$ and therefore there will be no need to decompose $\mathbf{M}$ to obtain A. Using this kernel, following the formulation of EMK [6], the final feature vector for each action volume is $F_V = \cup_{s=1}^{PN} \left[\frac{1}{|P_s|} \mathbf{G} \sum_{m \in P_s} k_M(c_M, C)\right]$, where C is the dictionary constructed by EMK [6], and $\mathbf{G}$ is calculated using $\mathbf{G}^T\mathbf{G} = (k_M(C, C))^{-1}$. The process of calculating these features, based on learned dictionary is completely explained in [6]. Finally, the linear SVM is applied for the classification.

4 Experimental results

The proposed action recognition method is evaluated on five public RGB-D benchmark datasets of: MSR Action 3D dataset [26], MSR Gesture 3D dataset [41], MSR Action Pairs dataset [30], MSR Daily Activity 3D dataset [42], and UT Kinect [50]. The algorithm is also compared with the state-of-the-art methods of action recognition from RGB-D data. The empirical results show that the proposed method outperforms other methods.

4.1 Parameter setting

The parameters to be set for the denoising step are σ in 2D Gaussian smoothing and τ in 1D Gabor filtering, applied spatially and temporally, respectively. In this work, $\sigma = 1.5,\ 2.5,$ and 3.5 and $\tau = 1.5$ and 2.5 are tested. Figure 4 presents the final accuracy for different values of σ and τ. The experiment without smoothing is shown with value 0 as both σ and τ. It is noticed that smoothing functions increase the accuracy of SSTKDes, especially for MSR Daily Activity 3D dataset which is the noisiest one. It is also observed that the final accuracy is robust to different values used as smoothing parameters.

The kernel parameters γ_o and γ_p for orientation and position kernel have been set to 5 and 3, respectively; like the original kernel descriptor paper [5]. To handle the computational cost (like [22]), each video is resized in such a way that the size of its frames is no larger than 150 × 150. The effect of the 3D blob size along 3 dimensions on the final accuracy was tested by running the algorithm with different values. It is noteworthy that by changing the 3D blob size from 5 × 5 × 5 to 20 × 20 × 20 there was no significant changes in the final accuracy. It is worth mentioning that, having 3D blobs with large sizes (like the whole pixel size) reduces the accuracy. Since the features are pooled over all pixels in 3D blobs, the spatial and temporal order of motions in video will be lost. to have a more fair comparison, the 3D blob with size 16 × 16 × 16 and 50% overlap with neighbors is selected from the video (like [22]). In addition, as the number of 3D blobs is too large for constructing the dictionary, the dense sampling is used.

It might be worth mentioning that since the temporal resolution in the used benchmark datasets is smaller than the spatial resolution in a certain extent, it was thought that increasing the temporal resolution might affect the final accuracy. Therefore, in an experience, the temporal resolution was increased by a bilinear temporal interpolation. But, there was no improvement on the final accuracy. This might be related to the fact that simple bilinear

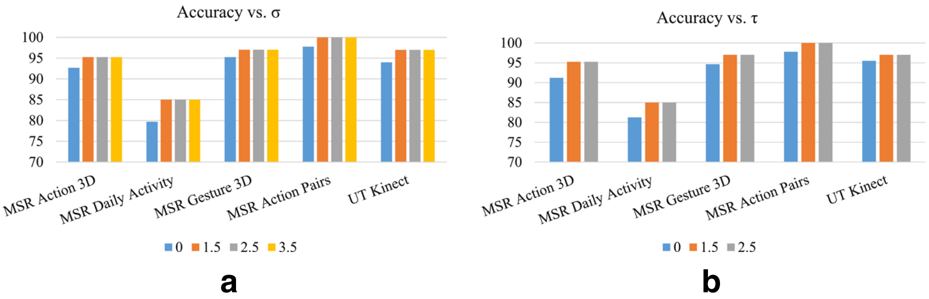

Fig. 4 **a** Accuracy vs. σ, **b** Accuracy vs. τ

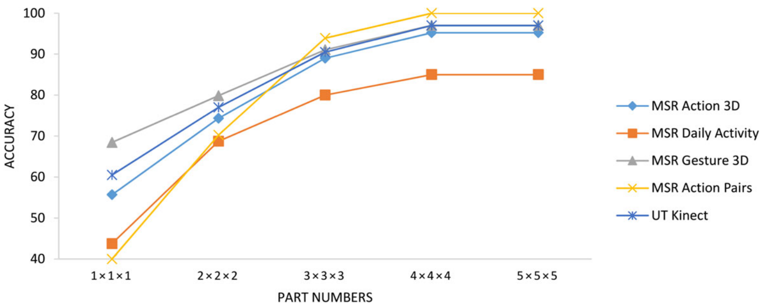

Fig. 5 Accuracy of actions vs. part numbers

interpolation does not have significant effects on 3D gradients used as the low-level features. In other words, it can only smooth the 3D gradients. For future work, the effect of more complicated interpolation methods, particularly the learning-based ones, can be tested.

In video hierarchy step, each video was divided to $1 \times 1 \times 1$, $2 \times 2 \times 2$, $3 \times 3 \times 3$, $4 \times 4 \times 4$, and $5 \times 5 \times 5$ parts. Figure 5 shows the overall accuracy of SSTKDes on all datasets with different part numbers. It is noticed that, there is an optimal number of words (i.e., $4 \times 4 \times 4$). By increasing the number of parts from $1 \times 1 \times 1$ to $4 \times 4 \times 4$ the accuracy of SSTKDes significantly increases and after that does not change. The concatenation of parts can yield the descriptor to preserve the general spatio-temporal order of actions while average pooling in each part removes redundant data. For instance, in Action Paris Dataset, actions *pick up* and *put down* have the same motion and shape but with inverse temporal order. In Fig. 5, it can be seen that by increasing the part numbers, the accuracy of this dataset is notably increasing.

The LMNN has also some parameters to be set. In (12), the number of target neighbors is set to $k_p = 4$. The parameters λ and λ_* are also set experimentally to 0.5 and 0.01, respectively.

One of the most effective parameters on the final accuracy is γ_m. It is used to calculate the final feature vector in the second kernel. To empirically study the performance of the proposed method, SSTKDes is trained with different values of γ_m. The accuracy for different values of this parameter over the range of 1 to 0.00001 is shown in Fig. 6a. It can be seen that value 0.0001 is optimal for all datasets. In (13), the exponential power consist of γ_m and another element. Hence, despite the fact that γ_m has a value close to zero; the exponential power is not a very small number.

The dictionary used in this method is trained with multiple codebook sizes. In Fig. 6b, the accuracies of all datasets with different codebook sizes are depicted. It can be seen that for every dataset, the accuracy is first seriously increasing with the number of words in dictionary and then it does not change anymore. It is also observed that 3000 and 4000 are the most optimal values for different datasets.LIBlinear [1] is also used for classification with linear kernel. Parameter c is empirically set to 10. For other SVM parameters, the default values are used.

[1] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

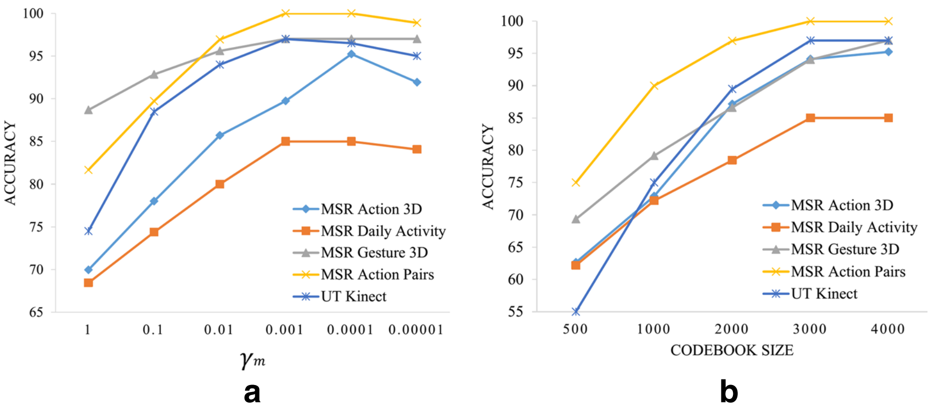

Fig. 6 **a** Accuracy vs. γ_m. **b** Accuracy vs. codebook size

4.2 MSR action 3D dataset

The MSR Action 3D Dataset contains gaming actions. It consists of depth sequences of 20 actions of: *high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, and pick up and throw*; each performed by 10 subjects for 2–3 times. The frame rate is 15 fps with resolution 320 × 240. The background of this dataset has been removed. The most important challenge of this dataset is the inter-action similarities. This dataset only contains depth videos, so our method is applied on this data.

In order to facilitate a fair comparison, the same experimental setting as other papers [42], is used; i.e., subjects 1,3,5,7,9 are used for training and the rest for testing. The confusion matrix on this dataset is presented in Fig. 7a. Table 1 lists the accuracy of the existing methods on this dataset. SSTKDes achieves the accuracy of 95.60% which outperforms the other methods. By comparing the confusion matrix of SSTKDes and the one of [22], it can be observed that confusions between similar actions in our method are fewer than [22]. In other words, SSTKDes effectively pushes the actions with different labels farther and pull

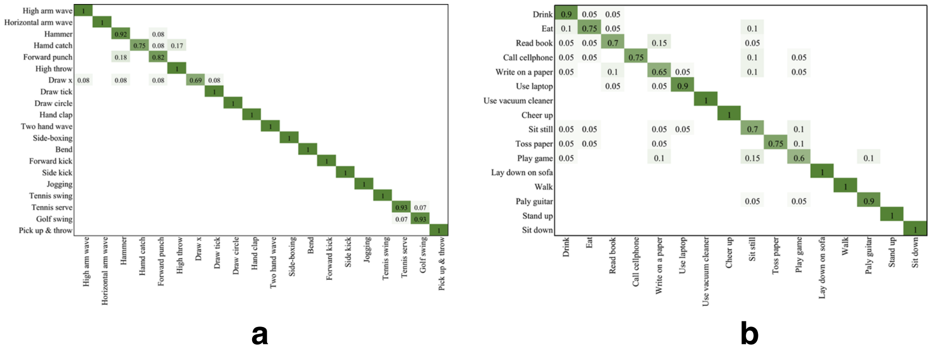

Fig. 7 Confusion matrix on: **a** MSR Action 3D dataset, **b** MSR Daily Activity 3D dataset

Table 1 Performance comparison on MSR Action 3D dataset

Method	Accuracy
Bag of 3D Points [26]	74.70
HOJ3D [51]	78.97
EigenJoints [54]	82.30
STOP [40]	84.80
ROP [41]	86.50
ActionLet [42]	88.20
HON4D [30]	88.36
DSTIP [50]	89.30
Vemulapalli et al. [39]	89.48
HOPC [32]	91.64
MovingPose [58]	91.70
HKDES [22]	92.73
Devanne et al. [13]	92.77
SNV [53]	93.09
ST-LSTM [27]	94.8
Proposed	95.60

the ones with the same label closer. Thus, it is more successful to handle the inter-class similarities and intra-class variabilities.

4.3 MSR daily activity 3D dataset

The MSR Daily Activity 3D dataset includes daily activities in a more realistic setting; i.e., two different poses with human object interaction in the living room. It consists of both RGB and depth sequences of 16 actions of: *drink, eat, read book, call cellphone, write on a paper, use laptop, use vacuum cleaner, cheer up, sit still, toss paper, play game, lay down on sofa, walk, play guitar, stand up, and sit down*; which are performed by 10 subjects twice in two different poses of "sitting on a sofa" and "standing". Subjects appear at different distances to the camera. Also, most of the actions involve human object interactions. These facts make this dataset very challenging. The proposed method is applied on both RGB and depth videos, and also a concatenation of both features from RGB and depth (i.e., RGB-D).

Figure 7b shows the confusion matrix of the best result (RGB-D) on this dataset. Table 2 compares the proposed method with the existing state-of-the-art methods. The result of our methods on depth data is better than the RGB one. The used raw attributes are the 3D gradients. Therefore, in a clutter background with more texture, many strong 3D gradients are extracted from background which are not related to the action. Since depth data has no texture, the strong 3D gradients are related to the performed action and probably noises. As it is discussed in Section 3.2, denoising step is able to suppressed a part of noise. Figure 4 also shows that smoothing function notably increase the performance of SSTKDes on this dataset. Hence, after denoising most part of strong 3D gradients will be related to the performed action. As a consequence, the depth data achieves better result than the RGB.

For this dataset, ActionLet [42] and SNV [53] get better accuracies than the proposed method. Those methods used the skeletal data for extracting the low-level features. In detail, the low-level features are extracted around a spatio-temporal neighborhood of the 3D location joints. In other words, they did not use all pixels in one frame for feature extraction,

Table 2 Performance comparison on MSR Daily Activity 3D dataset

Method	Accuracy
EigenJoints [55]	58.10
MovingPose [58]	73.8
Local HON4D [30]	80
HKDES [22]	83.13
DSTIP [50]	83.6
ActionLet [42]	85.75
SNV [53]	86.25
Proposed (RGB)	77.81
Proposed (Depth)	80.31
Proposed (RGB-D)	85

and thus, the noisy background has less effect on the video descriptor. However, based on the discussion about skeletal data in Section 2, this kind of data is not used and SSTKDes descriptor is computed by exploiting all pixels in frames. As a result, the noisy depth values influence the final accuracy. It is worth mentioning that although these methods get better result on this dataset, the SSTKDes likely results in better accuracy in real situations where the estimated human skeleton is not reliable. However, SSTKDes still achieves better accuracy than the HKDES [22] which is the effect of using the supervised strategy.

4.4 MSR gesture 3D dataset

The MSR Gesture 3D dataset contains depth sequences of: 12 dynamic *American Sign Language* (ASL) gestures, *bathroom, blue, finish, green, hungry, milk, past, pig, store, where, j, z*. Each gesture contains the segmented hand portion (above the wrist), and is performed by 10 subjects for 2–3 times. There is no available RGB data for this dataset. The confusion matrix in Fig. 8a is the best result of our method on depth videos of this dataset.

Table 3 addresses the performance of our method compared to the state-of-the-art methods. It can be indicated that SSTKDes outperforms all methods. The underlaying reason is that SSTKDes can efficiently take into account both spatial and temporal structure, formed by concatenating the features of parts. In fact, the motion (i.e., temporal information) of different hand parts (i.e., spatial information) are well organized by the hierarchy structure of actions. The used supervised strategy is also worthwhile to keep gestures with the same

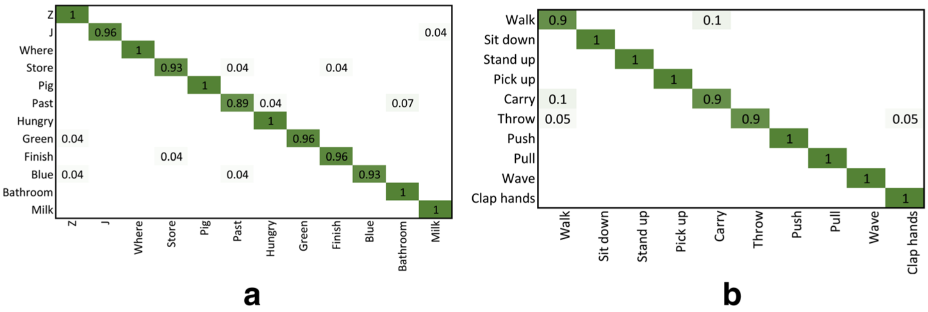

Fig. 8 Confusion matrix of: **a** Gesture3D dataset, **b** UT Kinect dataset

Table 3 Performance comparison on MSR Gesture 3D dataset

Method	Accuracy
Action Graph on Occupancy [23]	80.50
Action Graph on Silhouette [23]	87.70
ROP [41]	88.5
HON4D [30]	92.45
SNV [53]	94.74
HOPC [32]	95.29
HKDES [22]	96.09
Proposed	97.02

labels close to each other. As a result, it is capable of learning nonlinear (spatio-temporal) correlations of different parts of hand.

4.5 MSR action Paris dataset

The MSR Action Pairs dataset is a paired-activity dataset of both RGB and depth sequences of 6 pairs (12 activities) which are performed by 10 subjects for 3 times. It contains: *lift a box/place a box, pick up a box/put down a box, push a chair/pull a chair, put on a backpack/take off a backpack, stick a poster/remove a poster, and wear a hat/take off a hat*. The challenge of this dataset is the same shape of each action pair with reverse temporal order (like pick up and put down). In other words, considering the temporal order of frames is the most important factor for action recognition in this dataset.

Table 4 compares the performance of the proposed method with other state-of-the-art methods. It is indicate that SSTKDes achieves the best accuracy on this dataset; i.e., 100%. Therefore, no confusion matrix is depicted for this dataset. SSTKDes gains from preserving the temporal changes of the whole action volume by concatenation of parts. As a consequence, it can distinguish between actions with similar shape and different motion directions. The accuracies of RGB and depth data are very close to each other on this dataset.

4.6 UT kinect

The UT Kinect contains both RGB and depth sequences of 10 actions of: *hello, push, pull, boxing, step, forward-kick, side-kick, wave hands, bend, and clap hands*; performed twice by 10 subjects. The actions in this dataset cover the movements of hands, arms, legs, and

Table 4 Performance comparison on Action Pairs dataset

Method	Accuracy
ActionLet [42]	82.22
HON4D [30]	96.07
HOPC [32]	97.15
SNV [53]	98.33
HKDES [22]	100
Proposed (RGB)	100
Proposed (Depth)	99.44
Proposed (RGB-D)	100

Table 5 Performance comparison on UT Kinect dataset

Method	Accuracy
DSTIP [50]	85.7
HOJ3D [51]	90.92
Devanne et al. [13]	91.5
Liu et al. [28]	95.00
Vemulapalli et al. [39]	97.00
Proposed (RGB)	95
Proposed (Depth)	96
Proposed (RGB-D)	97

upper torso. The frame rate is 30 fps and its resolution is 320 × 240. Figure 8b shows the confusion matrix of the proposed method on this dataset which achieves 97%. Table 5 lists the performance of the proposed method and other stat-of-the-art methods. SSTKDes achieves the best result along with [39]. For this dataset, again, the depth data achieves better result than the RGB. The underlaying reason is that 3D gradients in depth data contains 3D geometrical information of the subject and the scene alongside the temporal information.

5 Conclusion

A novel supervised spatio-temporal kernel descriptor is proposed for human action recognition from RGB-D videos. 3D gradients are used as the low-level attributes for both RGB and depth videos, owing to the fact that it can capture both spatial and temporal information. In depth video, 3D gradient is also capable of taking 3D geometric information into account. The low-level 3D gradient attributes are then transfered into a kernel space. In the next level, by using a supervised strategy (LMNN) and a set of 3D blob kernel basis vectors (dictionary), generated through the EMK, features in kernel space are transformed into a more discriminate space. The success of this method was shown on object recognition [43].

In this paper, it has been shown that LMNN can efficiently minimize the intra-class variation and maximize inter-class dissimilarities for action recognition. Moreover, EMK combines the strengths of both bag of words and set kernels. It maps local features to a low dimensional feature space and then the set-level feature vector is formed by averaging the resulting feature vectors. It produces more accurate quantization and better performance. Finally, actions are classified by linear SVM using the feature vectors extracted from RGB, depth, or concatenation of them (RGB-D). The experimental results show the efficiency and superiority of SSTKDes.

References

1. Aggarwal JK, Ryoo MS (2011) Human activity analysis: a review. ACM Comput Surv (CSUR) 43(3):16
2. Aggarwal JK, Xia L (2014) Human activity recognition from 3d data: a review. Pattern Recogn Lett 48:70–80
3. Asadi-Aghbolaghi M, Kasaei S (2014) View invariant human action recognition using fourier-based and radon-based point cloud analysis. In: 2014 7th international symposium on telecommunications (IST). IEEE, pp 66–71
4. Asadi-Aghbolaghi M, Ramezanpour S, Kasaei S (2014) A new feature descriptor for 3d human action recognition. In: 2014 22nd Iranian conference on electrical engineering (ICEE). IEEE, pp 1157–1161

5. Bo L, Ren X, Fox D (2010) Kernel descriptors for visual recognition. In: Advances in neural information processing systems, pp 244–252
6. Bo L, Sminchisescu C (2009) Efficient match kernel between sets of features for visual recognition. In: Advances in neural information processing systems, pp 135–143
7. Boureau YL, Ponce J, LeCun Y (2010) A theoretical analysis of feature pooling in visual recognition. In: Proceedings of the 27th international conference on machine learning (ICML-10), pp 111–118
8. Brown M, Hua G, Winder S (2011) Discriminative learning of local image descriptors. IEEE Trans Pattern Anal Mach Intell 33(1):43–57
9. Calonder M, Lepetit V, Strecha C, Fua P (2010) Brief: binary robust independent elementary features. In: European conference on computer vision. Springer, pp 778–792
10. Chaaraoui AA, Padilla-López JR, Climent-Pérez P, Flórez-Revuelta F (2014) Evolutionary joint selection to improve human action recognition with rgb-d devices. Expert systems with applications 41(3):786–794
11. Chen C, Jafari R, Kehtarnavaz N (2015) Action recognition from depth sequences using depth motion maps-based local binary patterns. In: 2015 IEEE winter conference on applications of computer vision. IEEE, pp 1092–1099
12. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE Computer society conference on computer vision and pattern recognition (CVPR'05), vol 1. IEEE, pp 886–893
13. Devanne M, Wannous H, Berretti S, Pala P, Daoudi M, Del Bimbo A (2013) Space-time pose representation for 3d human action recognition. In: International conference on image analysis and processing. Springer, pp 456–464
14. Du Y, Wang W, Wang L (2015) Hierarchical recurrent neural network for skeleton based action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1110–1118
15. Feichtenhofer C, Pinz A, Zisserman A (2016) Convolutional two-stream network fusion for video action recognition. arXiv:160406573
16. Gu Y, Do H, Ou Y, Sheng W (2012) Human gesture recognition through a kinect sensor. In: 2012 IEEE international conference on robotics and biomimetics (ROBIO). IEEE, pp 1379–1384
17. Gupta A, Martinez J, Little JJ, Woodham RJ (2014) 3d pose from motion for cross-view action recognition via non-linear circulant temporal encoding. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2601–2608
18. Han F, Reily B, Hoff W, Zhang H (2016) Space-time representation of people based on 3d skeletal data: a review. arXiv:160101006
19. Jafari R, Ziou D (2012) Gaze estimation using kinect/ptz camera. In: 2012 IEEE international symposium on robotic and sensors environments (ROSE). IEEE, pp 13–18
20. Junejo IN, Dexter E, Laptev I, Perez P (2011) View-independent action recognition from temporal self-similarities. IEEE Trans Pattern Anal Mach Intell 33(1):172–185
21. Kang SM, Wildes RP, 2016 Review of action recognition and detection methods. arXiv:161006906
22. Kong Y, Satarboroujeni B, Fu Y (2016) Learning hierarchical 3d kernel descriptors for rgb-d action recognition. Comput Vis Image Underst 144:14–23
23. Kurakin A, Zhang Z, Liu Z (2012) A real time system for dynamic hand gesture recognition with a depth sensor. In: 2012 Proceedings of the 20th European signal processing conference (EUSIPCO). IEEE, pp 1975–1979
24. Laptev I (2005) On space-time interest points. Int J Comput Vis 64(2-3):107–123
25. Laptev I, Marszalek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies. In: IEEE conference on computer vision and pattern recognition, 2008. CVPR 2008. IEEE, pp 1–8
26. Li W, Zhang Z, Liu Z (2010) Action recognition based on a bag of 3d points. In: 2010 IEEE computer society conference on computer vision and pattern recognition-workshops. IEEE, pp 9–14
27. Liu J, Shahroudy A, Xu D, Wang G (2016) Spatio-temporal lstm with trust gates for 3d human action recognition. In: European conference on computer vision. Springer, pp 816–833
28. Liu Z, Feng X, Tian Y (2015) An effective view and time-invariant action recognition method based on depth videos. In: 2015 visual communications and image processing (VCIP). IEEE, pp 1–4
29. Lu C, Jia J, Tang CK (2014) Range-sample depth feature for action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 772–779
30. Oreifej O, Liu Z (2013) Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 716–723
31. Poppe R (2010) A survey on vision-based human action recognition. Image Vis Comput 28(6):976–990
32. Rahmani H, Mahmood A, Huynh DQ, Mian A (2014) Hopc: Histogram of oriented principal components of 3d pointclouds for action recognition. In: European conference on computer vision. Springer, pp 742–757

33. Reyes M, Domínguez G, Escalera S (2011) Featureweighting in dynamic timewarping for gesture recognition in depth data. In: 2011 IEEE international conference on computer vision workshops (ICCV workshops). IEEE, pp 1182–1188
34. Shotton J, Sharp T, Kipman A, Fitzgibbon A, Finocchio M, Blake A, Cook M, Moore R (2013) Real-time human pose recognition in parts from single depth images. Commun ACM 56(1):116–124
35. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems, pp 568–576
36. Sung J, Ponce C, Selman B, Saxena A (2012) Unstructured human activity detection from rgbd images. In: 2012 IEEE international conference on robotics and automation (ICRA). IEEE, pp 842–849
37. Tehrani AKN, Aghbolaghi MA, Kasaei S (2017) Skeleton-based human action recognition - a learning method based on active joints. In: Proceedings of the 12th international joint conference on computer vision, imaging and computer graphics theory and applications - vol 5: VISAPP (VISIGRAPP 2017), pp 303–310
38. Varol G, Laptev I, Schmid C (2016) Long-term temporal convolutions for action recognition. arXiv:160404494
39. Vemulapalli R, Arrate F, Chellappa R (2014) Human action recognition by representing 3d skeletons as points in a lie group. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 588–595
40. Vieira AW, Nascimento ER, Oliveira GL, Liu Z, Campos MF (2012) Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences. In: Iberoamerican congress on pattern recognition. Springer, pp 252–259
41. Wang J, Liu Z, Chorowski J, Chen Z, Wu Y (2012a) Robust 3d action recognition with random occupancy patterns. In: Computer vision–ECCV 2012. Springer, pp 872–885
42. Wang J, Liu Z, Wu Y, Yuan J (2012b) Mining actionlet ensemble for action recognition with depth cameras. In: 2012 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 1290–1297
43. Wang P, Wang J, Zeng G, Xu W, Zha H, Li S (2013) Supervised kernel descriptors for visual recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2858–2865
44. Wang P, Li W, Gao Z, Zhang J, Tang C, Ogunbona PO (2015a) Action recognition from depth maps using deep convolutional neural networks
45. Wang X, Farhadi A, Gupta A (2015b) Actions~ transformations. arXiv:151200795
46. Wei P, Zhao Y, Zheng N, Zhu SC (2013) Modeling 4d human-object interactions for event and object recognition. In: 2013 IEEE international conference on computer vision. IEEE, pp 3272–3279
47. Weinberger KQ, Saul LK (2009) Distance metric learning for large margin nearest neighbor classification. J Mach Learn Res 10(Feb):207–244
48. Weinland D, Ronfard R, Boyer E (2011) A survey of vision-based methods for action representation, segmentation and recognition. Comput Vis Image Underst 115(2):224–241
49. Wu D, Pigou L, Kindermans PJ, Nam L, Shao L, Dambre J, Odobez JM (2016) Deep dynamic neural networks for multimodal gesture segmentation and recognition
50. Xia L, Aggarwal J (2013) Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2834–2841
51. Xia L, Chen CC, Aggarwal J (2012) View invariant human action recognition using histograms of 3d joints. In: 2012 IEEE computer society conference on computer vision and pattern recognition workshops. IEEE, pp 20–27
52. Xiao Y, Xia L (2016) Human action recognition using modified slow feature analysis and multiple kernel learning. Multimedia Tools and Applications 75(21):13,041–13,056
53. Yang X, Tian Y (2014) Super normal vector for activity recognition using depth sequences. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 804–811
54. Yang X, Tian YL (2012) Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In: 2012 IEEE computer society conference on computer vision and pattern recognition workshops. IEEE, pp 14–19
55. Yang X, Zhang C, Tian Y (2012) Recognizing actions using depth motion maps-based histograms of oriented gradients. In: Proceedings of the 20th ACM international conference on multimedia. ACM, pp 1057–1060
56. Ye M, Zhang Q, Wang L, Zhu J, Yang R, Gall J (2013) A survey on human motion analysis from depth data. In: Time-of-flight and depth imaging. Sensors, algorithms, and applications. Springer, pp 149–187
57. Yu S, Cheng Y, Su S, Cai G, Li S (2016) Stratified pooling based deep convolutional neural networks for human action recognition. Multimedia Tools and Applications, pp 1–16

58. Zanfir M, Leordeanu M, Sminchisescu C (2013) The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In: Proceedings of the IEEE international conference on computer vision, pp 2752–2759

Maryam Asadi-Aghbolaghi Received both B.Sc. and M.Sc. degrees in Computer Engineering from Iran University of Science and Technology, Tehran, Iran in 2008 and 2011, respectively. Currently, she is a PhD student at Sharif University of Technology, Tehran, Iran. Her research interests include human action recognition, ambient Intelligence, 3D computer vision, machine learning, and data mining.

Shohreh Kasaei received the B.Sc. degree from the Department of Electronics, Faculty of Computer and Electrical Engineering, Isfahan University of Technology, Iran, in 1986. She then received the M.Sc. degree from the Graduate School of Engineering, Department of Electrical and Electronic Engineering, University of the Ryukyus, Japan, in 1994, and the Ph.D. degree from Signal Processing Research Centre, School of Electrical and Electronic Systems Engineering, Queensland University of Technology, Australia, in 1998. She was awarded as the best graduate student in engineering faculties of University of the Ryukyus, in 1994, the best Ph.D. student studied in overseas by the ministry of Science, Research, and Technology of Iran, in 1998, and as a distinguished researcher of Sharif University of Technology, in 2002 and 2010, where she is currently a full professor. She is the director of image processing lab (IPL). Her research interests are in image/video processing and 3D computer vision with primary emphasis on 4D pose estimation, 4D object tracking, 3D model building, multiresolution texture analysis, scalable video coding, image retrieval, video indexing, face recognition, hyperspectral change detection, and video restoration.