Signal Processing 110 (2015) 67–81

Contents lists available at ScienceDirect

Signal Processing

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

Combining unsupervised learning and discrimination for 3D action recognition

Guang Chen [a,b,*], Daniel Clarke [b], Manuel Giuliani [b], Andre Gaschler [b], Alois Knoll [a]

[a] Institut für Informatik VI, Technische Universität München, Boltzmannstr. 3, 85748 Garching, Germany
[b] fortiss GmbH, An-Institut Technische Universität München, Guerickestr. 25, 80805 Munich, Germany

ARTICLE INFO

Article history:
Received 1 March 2014
Received in revised form
1 August 2014
Accepted 16 August 2014
Available online 23 August 2014

Keywords:
Human action recognition
Depth camera
Unsupervised learning
Multi-kernel learning
Ensemble learning

ABSTRACT

Previous work on 3D action recognition has focused on using hand-designed features, either from depth videos or 2D videos. In this work, we present an effective way to combine unsupervised feature learning with discriminative feature mining. Unsupervised feature learning allows us to extract spatio-temporal features from unlabeled video data. With this, we can avoid the cumbersome process of designing feature extraction by hand. We propose an ensemble approach using a discriminative learning algorithm, where each base learner is a discriminative multi-kernel-learning classifier, trained to learn an optimal combination of joint-based features. Our evaluation includes a comparison to state-of-the-art methods on the MSRAction 3D dataset, where our method, abbreviated EnMkl, outperforms earlier methods. Furthermore, we analyze the efficiency of our approach in a 3D action recognition system.

1. Introduction

Human action recognition plays an important role in a number of real-world applications such as video surveillance, health care, and human-computer interactions. With recent developments in low-cost sensors such as Microsoft Kinect, depth cameras have received great attention among researchers and have led them to revisit problems such as object detection and action recognition in depth video data [1–3].

Compared to visible light cameras, depth sensors provide 3D structural information of a scene, which is invariant to lighting and color variation. Recently, with the emergence of 3D displays in consumer markets, there is a rise in commercially available 3D content. As an example, Hadfield and Bowden [4] extract 3D actions from movies and use commercial camera rigs to produce 3D consumer content. However, reconstructing depth from stereo cameras requires expensive computations and typically introduces substantial, undesirable artifacts. In contrast, depth sensors use structured light to generate real-time 3D depth maps rather reliably. These depth maps allow rather powerful human motion capturing techniques [5], which can recognize 3D joint positions of human skeletons in real-time.

In 3D action recognition, which is the topic of this paper, two significant questions arise when using depth video sequences. First, how can we represent depth video data efficiently? State-of-the-art techniques represent depth video data by extracting manually designed features, either directly from depth video data or extending hand-designed features from color-based video data [6–8]. Despite their good performance for 3D action recognition, these methods

* Corresponding author at: fortiss GmbH, An-Institut Technische Universität München, Guerickestr. 25, 80805 Munich, Germany.
E-mail addresses: guang@in.tum.de (G. Chen), clarke@fortiss.org (D. Clarke), giuliani@fortiss.org (M. Giuliani), gaschler@fortiss.org (A. Gaschler), knoll@in.tum.de (A. Knoll).

http://dx.doi.org/10.1016/j.sigpro.2014.08.024
0165-1684/

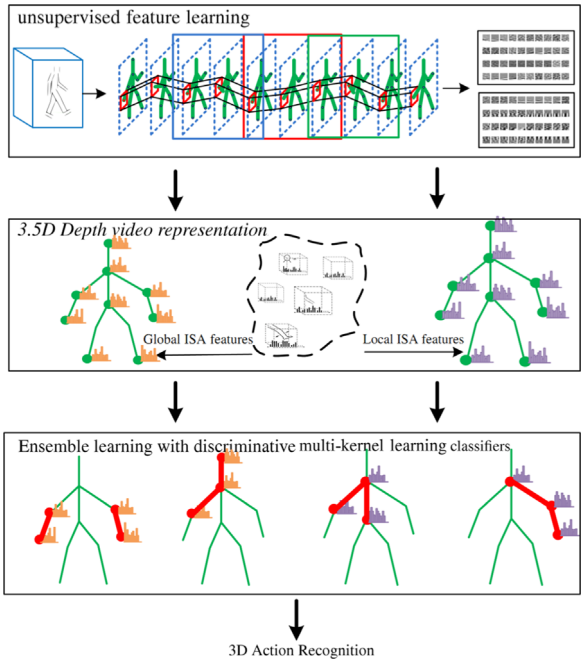

Fig. 1. The general framework of our proposed approach. Our framework consists of two main parts: unsupervised feature learning and discriminative feature mining. We develop two types of spatio-temporal features from depth video data using independent subspace analysis and apply an ensemble approach with discriminative multi-kernel-learning classifiers. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

suffer from one problem: designing features by hand requires a heavy, manual workload. In this work, we provide an unsupervised learning method to learn a 3.5D representation of depth video data inspired by [9,10] (see Fig. 1). At the heart of our method is the application of Independent Subspace Analysis (ISA). A main advantage of ISA is that it learns features that are robust to local translation, while being selective to rotation and velocity. A disadvantage of ISA is that it can be rather inefficient to train with highly dimensional data, such as video data. We therefore extend the original ISA algorithm for the use of depth video data and human skeleton data (see Figs. 2 and 3). Rather than training the model with the entire video in [11,10], we apply the ISA algorithm to local regions of joints and substantially improve the training efficiency. Based on depth video and estimated 3D joint positions, we develop two types of spatio-temporal features: Global ISA features (GISA) and Local ISA features (LISA). These spatio-temporal features can be treated as the resulting descriptors of spatio-temporal interest points. Interest points are dense sampled from the surrounding regions of the joints.

Second, how can we deal with noisy human skeleton data and improve the robustness of 3D action recognition systems? Skeleton data have natural correspondences over time, which model temporal dynamics and spatial structures explicitly. Together with other modalities (e.g. depth video data), skeleton data may improve the performance of 3D action recognition. However, when skeleton data contain irrelevant or redundant information, performance may be adversely affected. In order to tackle the problem of tracking errors in skeleton data and to handle intra-class variations more robustly, we propose *an ensemble learning approach with discriminative multi-kernel learning* (EnMkl) for 3D action recognition. In our implementation, we formulate the 3D action recognition task with depth video as a multiple-kernel learning problem. MKL is able to discover discriminative features for vision tasks automatically. The underlying idea for employing the MKL approach is that a certain action class is usually only associated with a subset of kinematic joints of the articulated human body. In our case, 3D actions are represented as a linear combination of joints, where each joint is associated with a weight. This weighted joint model is more robust to noisy features and it can better characterize intra-class variations. In addition, we integrate ensemble learning with discriminative MKL classifiers. Training and combining multiple classifiers, ensemble methods [12] are state-of-the-art techniques with strong generalization abilities.

Our contribution is therefore an original approach by combining unsupervised feature learning and discriminative feature mining to recognize 3D human actions. In summary, the novelty of our approach is four-fold. (1) Our algorithm is unsupervised and rather generic, and may therefore be applicable to a wider range of problems with unlabeled sensor data. To the best of our knowledge, this approach is the first attempt to learn spatio-temporal features from depth video data and skeleton data in an unsupervised way. (2) We propose an ensemble learning approach with discriminative multi-kernel learning classifiers, which allows for a better characterization of inter-class variations in the presence of noisy or erroneous skeleton data. (3) We improve our performance in terms of the recognition accuracy on MSRAction3D dataset [1] (see Table 3) and show an accuracy superior to the state-of-the-art. (4) We further investigate our model and analyze the efficiency of a 3D action recognition system. We find that a small subset of joints (1–6 joints) is sufficient to perform action recognition if action classes are targeted. This observation is important to allow online decisions and improvements to the efficiency of action recognition tasks. A preliminary version of this work appeared in [13].

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 describes our learning approach of the 3.5D representation of depth video data. Section 4 presents the ensemble learning approach with discriminative MKL classifiers. Section 5 discusses the results of our evaluation. Finally, Section 6 concludes the paper.

2. Related work

Methods for 3D human action recognition generally consist of two main stages: 3D video representation (extraction of suitable spatio-temporal features) and machine modeling of human actions (modeling and learning of dynamic patterns).

2.1. 3D video representation

A straightforward way to calculate spatio-temporal features from 3D video data is to extend methods based

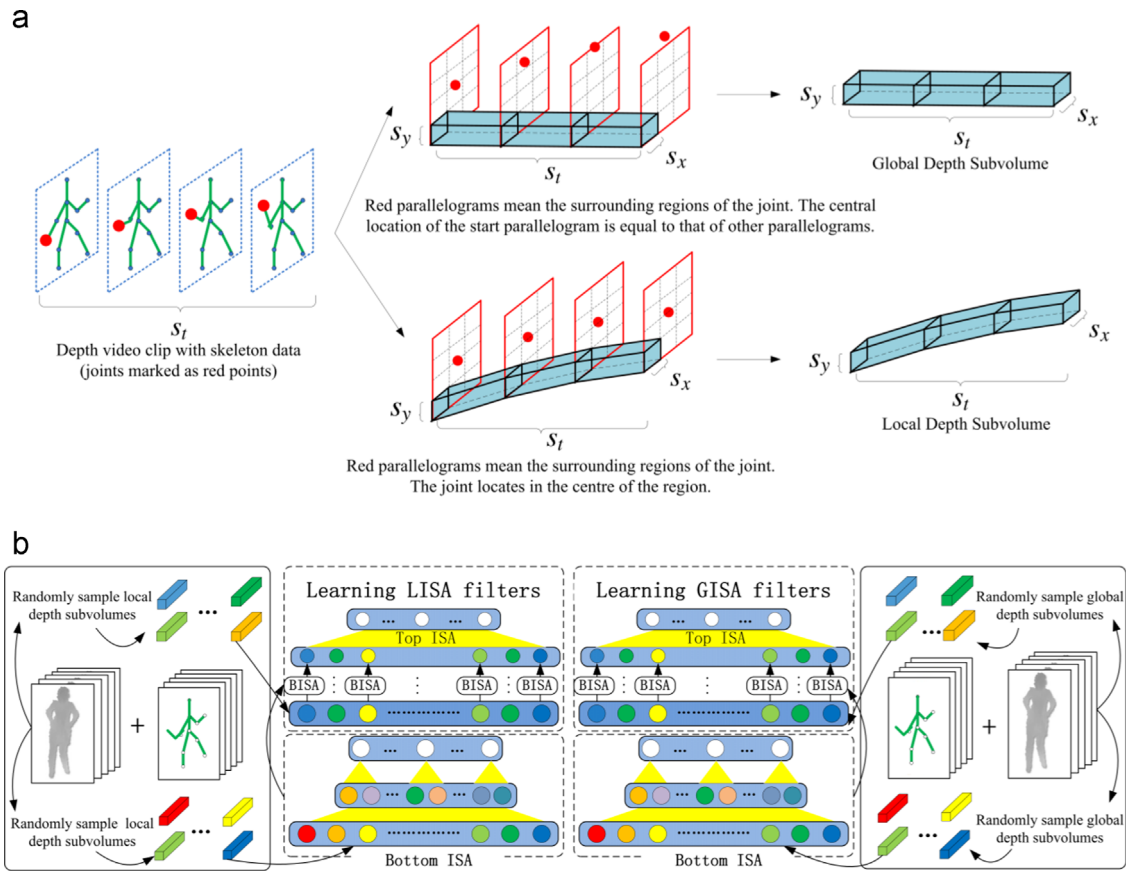

Fig. 2. Overview of our ISA model: (a) we randomly sample global depth subvolumes and local depth subvolumes from the surrounding regions of each joint. (b) These two types of subvolumes are given as inputs to the two-layer ISA network. Our ISA model learns two types of features: Global ISA feature (GISA) and Local ISA feature (LISA). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

design for 2D video data. Hernandez-Vela et al. [14] apply 3D Harris detectors [6] separately on RGB and depth video data. They treat the depth video as gray-scale video data, without using the spatial information along the depth direction. Recent work by Hadfield and Bowden [4] incorporate depth information while detecting spatio-temporal interest points. They extend 3D Harris detectors and 3D Hessian detectors [6] to 4D cases by exploiting the relationship between the spatio-temporal gradients of the depth stream and those of the appearance stream. Zhao et al. [15] and Ni et al. [16] use the HOF [7] and HOG [8] features to describe interest points in depth video data. Instead of appearance, motion and saliency be used in above descriptors and depth information can be utilized. Hadfield and Bowden [4] extend the HOG and HOF features to 4D descriptors (HODG), which encapsulate local structural information, in addition to local appearance and motion. Hadfield and Bowden [4] extend RMD (Relative Motion Descriptor) [17] to 4D (RMD-4D). RMD-4D makes use of saliency information within a 4D integral hyper-volume during interest point detection.

Recent research focuses on designing features to characterize unique properties of the depth video data more directly, rather than extending existing algorithms designed for 2D video data. Cheng et al. [18] design the comparative coding descriptor (CCD) to capture spatial geometric relations and related variations over time. The CCD feature essentially applies a comparative description idea used in Local Binary Patterns [19]. Inspired by the CCD and LBP features, Zhao et al. [15] develop the local depth pattern feature (LDP). In their work, local regions are partitioned into spatial cells, with the average depth value being computed for each cell. The differences of average depth values between every pair of cells form the LDP feature. In a different approach, Wang et al. [20] treat depth videos as a 4D volume. They employ four dimensional random occupancy patterns to construct their features (ROP). The ROP features therefore capture the occupancy pattern of a 4D subvolume.

The third type of 3D video representation is skeleton-based representations, which describe actions in depth video data by modeling the spatial-temporal structure of the human skeleton. Skeleton-based representations are widely used in computer vision tasks. Yu et al. [21,22] use skeletons for describing the gestures of characters. Yang and Tian [23] develop the EigenJoints features based on the differences of skeleton joints. EigenJoint is able to characterize action information, including static posture features, consecutive motion features, and offset features in each frame. Bloom et al. [24] use features based on

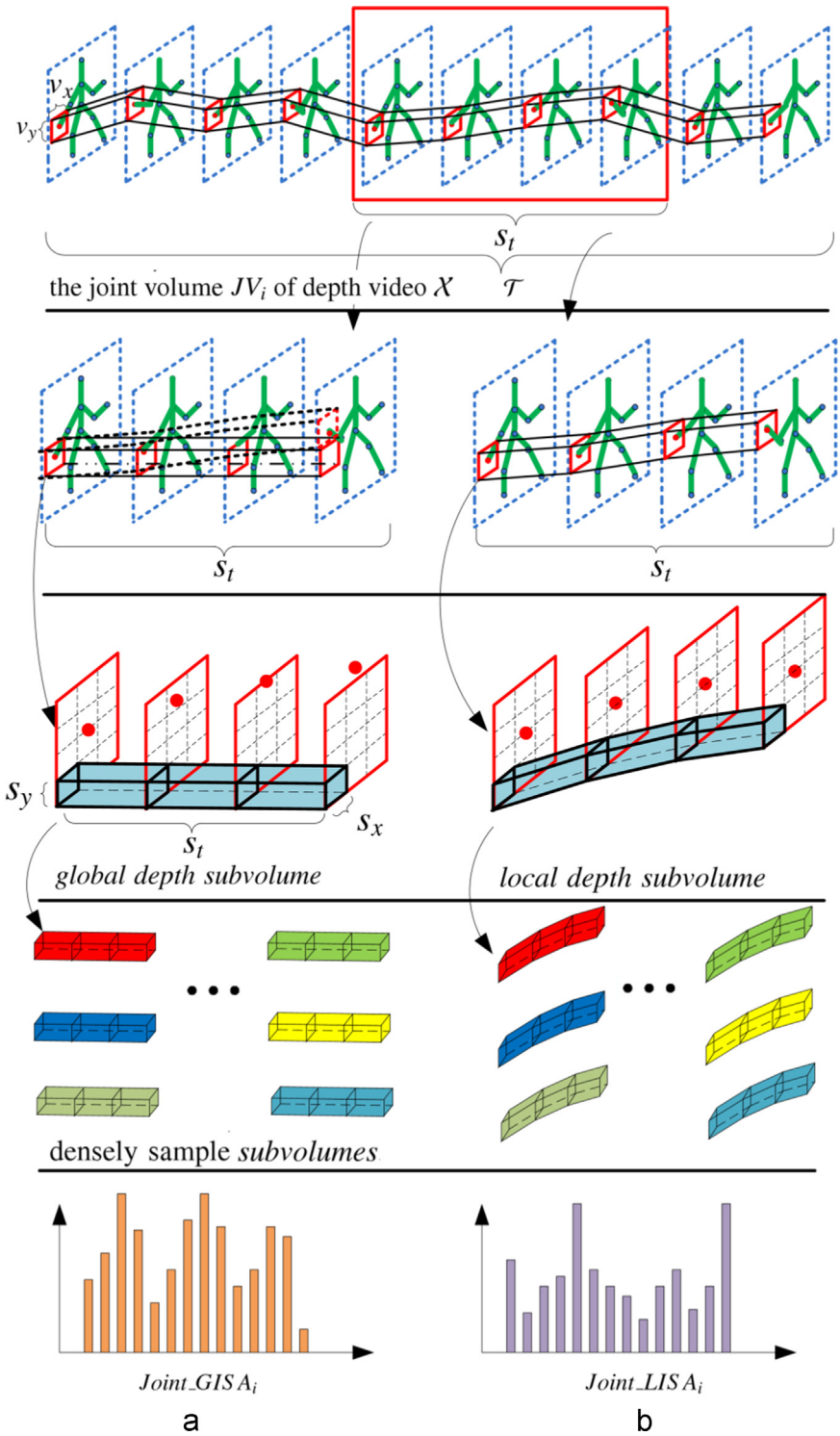

Fig. 3. The processing steps of learning (a) Joint_GISA features and (b) Joint_LISA features. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

human poses, such as position difference, position velocity, position velocity magnitude, angular velocity, and joint angles. Xia et al. [25] develop the histogram of 3D joints (HOJ3D), a viewpoint invariant representation of postures based on 3D skeleton joint locations. They compute HOJ3D from 12 of 20 joints in order to exclude possibly redundant information. Wang et al. [26] also apply pose-based features to action recognition. They use data mining techniques [27]

to obtain sets of distinctive co-occurring spatial and temporal configurations of body parts.

2.2. Machine modeling of actions

Machine learning is widely used in the computer vision tasks [28–31]. When a 3D video representation is provided for an observed sequence, human action recognition becomes a classification task. Data mining, used as a common step before actual classification, is becoming popular among recently developed approaches for 3D action recognition. It is used to cope with the new challenges—noisy data, and a rich collection of features—introduced by the new 3D modalities. Wang et al. [32] propose a data mining solution to discover discriminative conjunction rules. Their idea is inspired by successful applications of AND/OR graph learning in [33,34]. An *actionlet* is defined as an AND structure within the base representation. A mining algorithm is developed in [32] to mine a discriminative set of such actionlets. Chen et al. [35] use decision forests to discover the discriminative spatio-temporal cuboids in the dense sampling spatio-temporal space of the video data. Xia and Aggarwal [36] address this issue and mine discriminative feature sets from the feature pool based on F-scores. They rank feature prototypes by their F-scores and select features with high F-score. Oreifej and Liu [2] pay attention to quantization methods in order to build histogram-based descriptors. The bins of the histogram are voted by using only videos that correspond to the weighted set of support vectors from Support Vector Machine (SVM) classifiers.

Direct modeling approaches classify 3D video representations into action classes without modeling temporal variations explicitly. Wang et al. [32] employ MKL to learn a representation ensemble structure that combines discriminative representations in the data mining step, where each kernel corresponds to a discriminative representation. Ofli et al. [37] learn an MKL classifier by combining various modalities. Rehmani et al. [38] employ Random Forests (RF) for feature selection in conjunction with 3D action classification. RF is first trained using the set of all proposed features in [38], then discarding all features whose scores are below a specified threshold. The resulting compact feature vectors are then selected to train a new RF to be used for classification.

Recent works also utilize temporal state-based approaches in order to model the dynamics as a part of the classification process. Reyes et al. [39] present a 3D gesture recognition approach based on a dynamic time warping (DTW) framework. Depth features from human joints are compared through video sequences using DTW and weights are assigned to features based on inter–intra class gesture variability. Wang and Wu [40] develop a discriminative learning-based temporal alignment method, named maximum margin temporal warping (MMTW), to align two action videos and measure their matching scores. Gaschler et al. [41,42] train hidden Markov models (HMM) using human body posture and head pose estimation from depth cameras to recognize human social behaviors. Instead of performing off-line recognition, the action graph classifier which is proposed by Li et al. [43] has the advantage that it can perform classification without waiting until an action is finished. Vieira et al. [44] and Kurakin et al. [45] extend the action graph classifier from 2D video recognition to 3D. They employ action graphs to 3D action recognition and 3D dynamic hand gesture recognition, respectively. In addition, ensemble learning methods have also been widely applied in classification processes. Geng et al. [46] propose an efficient ensemble learning method and showed its effectiveness in real applications such as digit recognition and image classification. Yu et al. [47,48] apply ensemble learning to image classification and prediction of user behavior on websites. Xu et al. [49] develop an ensemble multi-instance multi-label learning approach for a video annotation problem.

The above approaches focus on hand-tuned features. In contrast, we generate spatio-temporal features in an unsupervised learning approach. In order to deal with tracking errors and redundancies in the skeleton data, we formulate the action recognition problem with multiple joints as an MKL approach. We further integrate an ensemble method into the MKL framework. In general, our unsupervised learning approach is not limited to 3.5D data, but can rather easily be adapted to other modalities.

3. Unsupervised learning of spatio-temporal features

This section describes the two types of spatio-temporal features that we use to represent the 3D actions: the local and global spatio-temporal features based on independent subspace analysis, abbreviated as LISA and GISA features, respectively. These features are learned directly from unlabeled depth video data using an extension of the independent subspace analysis algorithm (ISA). They are invariant to the translation of the human body and robust to noise. We first briefly describe the background of the ISA algorithm in Section 3.1. In Section 3.2, we elaborate our approach to generate LISA and GISA features from depth video data and skeleton data. Sections 3.3 and 3.4 then describe the implementation details of the 3.5D depth video representation.

3.1. Independent subspace analysis for depth video data

ISA is an unsupervised learning algorithm that learns features from unlabeled data and is widely used in the static image domain. Applying this model to the depth video domain is rather straightforward. First, random subvolumes are extracted from the depth video data. The set of subvolumes is then normalized and whitened. We treat a subvolume as a sequence of depth image patches and flatten them into a vector. This vector is then fed to ISA networks as an input unit. An ISA network [9] is described as a two-layer neural network (e.g. the bottom ISA in Fig. 2), with square and square-root nonlinearities in the first and second layers, respectively.

We start with any input unit $x^t \in \mathbb{R}^n$ for each randomly sampled subvolume. We split each subvolume into a sequence of image patches and flatten them into a vector x^t with dimension n. The activation of each second layer

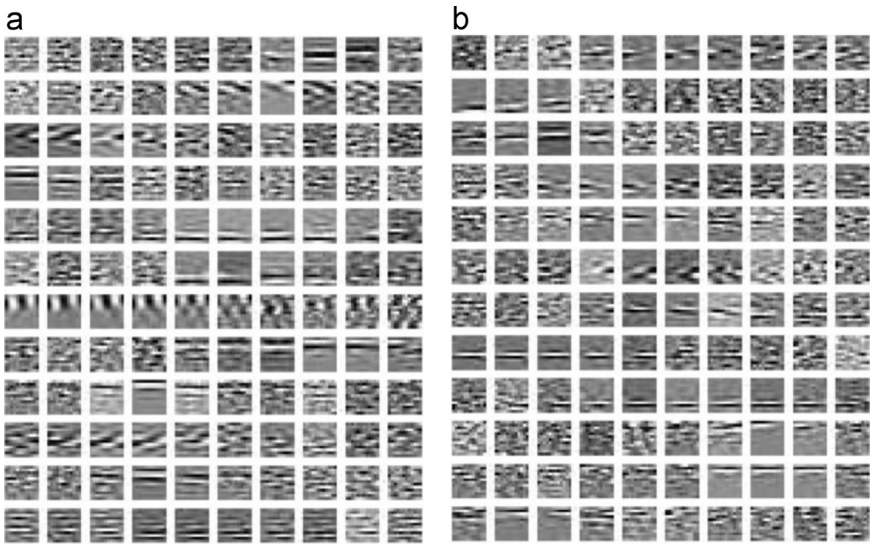

Fig. 4. Visualization of features learned by the model. (a) ISA features learning from Cross Subset3 of MSRAction3D dataset. The inputs of the ISA model are *global depth subvolumes*. (b) ISA features learning from Cross Subset3 of MSRAction3D dataset. The inputs of the ISA model are *local depth subvolumes*.

unit is

$$p_i(x^t; W, V) = \sqrt{\sum_{k=1}^{m} V_{ik} \left(\sum_{j=1}^{n} W_{kj} x_j^t \right)^2} \tag{1}$$

where i is the indicator of the activation of the second layer unit; $j = 1, \ldots, n$; $k = 1, \ldots, m$; n and m are the dimension of input unit x^t and the number of units in the second layer, respectively.

ISA learns the parameters W by finding sparse feature representations in the second layer, by solving

$$\begin{aligned} \min_{W} \quad & \sum_{t=1}^{T} \sum_{i=1}^{m} p_i(x^t; W, V) \\ s.t. \quad & WW^T = \mathbf{I} \end{aligned} \tag{2}$$

Here, $W \in \mathbb{R}^{u \times n}$ denotes the weights connecting the input units to the first layer units (u denotes the number of units in the first layer); $V \in \mathbb{R}^{m \times u}$ denotes the weights connecting the first layer units to the second layer units (V is typically fixed to represent the subspace structure of the neurons in the first layer); T is the number of the input units x^t. The orthonormal constraint is to ensure the features that are sufficiently diverse.

3.2. Learning LISA and GISA features

The standard ISA training algorithm degrades in efficiency when the input pattern x^t becomes large. The reason for this is the orthogonalization step that has to be called at each iteration of the projected gradient descent scheme, a problem that is addressed by [10]. In order to scale up the ISA algorithm to high-dimensional depth data, we use a stacked convolutional neural network architecture similar to [10]. The network progressively makes use of PCA and ISA as sub-units for unsupervised learning. The key ideas of this approach are as follows: we first train the bottom ISA network on small depth subvolumes. Then, we take the learned bottom ISA network and convolve with a larger region of the input depth subvolume. The combined responses of the convolution step of the bottom ISA are then given as an input to the top ISA network. The stacked model is trained greedily, and layer-by-layer, in the same manner as other algorithms described in [50,10]. Finally, we combine features from both the bottom and top ISA networks and use them as spatio-temporal features together with vector quantizations for classification.

In our implementation, we develop two types of spatio-temporal features (LISA and GISA features) according to the input patterns of the stacked ISA model (see Fig. 2). We define GISA features as the global spatio-temporal features learned by the stacked ISA model with the randomly sampled *global depth subvolumes* as input data. A *global depth subvolume* is a sequence of depth patches, where the depth patches of different timestamps have the same spatial size and image coordinates. Similarly, LISA features are learned by the stacked ISA model with randomly sampled *local depth subvolumes* as an input. A *local depth subvolume* is a sequence of depth patches, where the depth patches of different timestamps have the same spatial size but different image coordinates (see Fig. 2a). Fig. 4 shows the features learned by the bottom ISA layers. The bottom ISA model learns spatio-temporal features that detect moving edges in time. The learned feature (each row in Fig. 4(a) and (b)) is able to group similar features in a group, thereby achieving spatial invariance. These features have rather sharp edges, similar to Gabor filters [51]. This could be explained by the strong discontinuities that are prevalent at object boundaries in depth video data.

3.3. The 3.5D depth video representations

Based on the LISA and GISA features in the above section, we develop a new representation of human action, *3.5D Depth Video Representation*. It corresponds to the outcome of reconstructing 3.5D information from spatio-temporal

features (LISA and GISA features) and the skeleton data (3D joint positions).

Compared to interest point-based methods, which describe parts of interest in the scene, LISA and GISA features are used to describe general parts of the scene as they do not need apply interest point detection. This approach may save the time of interest point detection process, however it can be potentially time-consuming when the dimension of the input data is large. It is generally agreed that knowing the 3D joint position of human subject is helpful for action recognition. We therefore develop a 3.5D depth video representation to combine the 3D configuration of human skeletons and spatio-temporal features of each joint. In our implementation, we utilize LISA and GISA features as spatio-temporal features.

We borrow the term, *3.5D representation*, from stereoscopic vision [52], in which they use a 2.5 representation to describe actions in static imagery. A 3.5D representation $\mathcal{G}^{\mathcal{X}}$ describing a depth video $\mathcal{X}$ consists of V nodes connected by E edges. The nodes correspond to a set of key points (joints) of the human body. A node v is represented by the 3D position of this node p_v and the histogram feature $f_v^{\mathcal{X}}$ extracted in an image region surrounding this node in time. Adjacent nodes v and v' are connected by edge e. Finally, the 3.5D representation of a depth video is written as $\mathcal{G}^{\mathcal{X}}=\{f_{v_1}^{\mathcal{X}}, f_{v_2}^{\mathcal{X}} \ldots f_{v_k}^{\mathcal{X}}\}$, where k denotes the number of the joints.

3.4. Implementation

For a human subject in a depth video $\mathcal{X}$, the skeleton tracker tracks 20 joint positions [5] (see Fig. 5), which correspond to 20 nodes of a 3.5D representation $\mathcal{G}^{\mathcal{X}}$. For each joint i at frame t, its surrounding region S_t^i is of size (v_x, v_y) pixels. Let $\mathcal{T}$ denote the temporal dimension of the depth video $\mathcal{X}$. The depth video $\mathcal{X}$ is represented as the set of joint volumes $\{JV_1, JV_2 \ldots JV_{20}\}$. Each joint volume can be considered as a sequence of depth image patches $JV_i=\{S_1^i, S_2^i \ldots S_t^i\}$. The size of JV_i is $v_x \times v_y \times \mathcal{T}$ (see Fig. 3). Finally, $\mathcal{G}^{\mathcal{X}}$ is rewritten as $\mathcal{G}^{\mathcal{X}}=\{\mathcal{F}(JV_1), \mathcal{F}(JV_2) \ldots \mathcal{F}(JV_k)\}$, where $k=1, \ldots, 20$; $f_{v_k}^{\mathcal{X}}=\mathcal{F}(JV_k)$; $F(\cdot)$ is a function where the input joint volume JV_k is converted into a histogram feature $f_{v_k}^{\mathcal{X}}$.

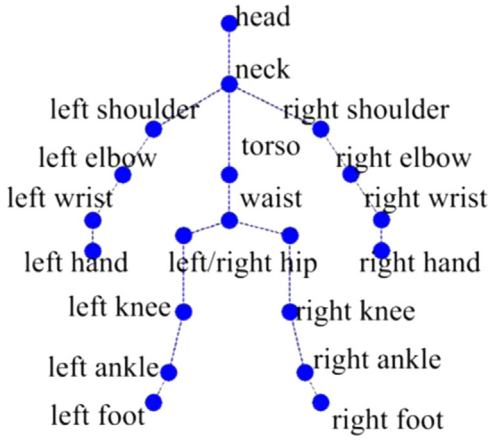

Fig. 5. Naming of the human joints tracked by the skeleton tracker [5].

As the surrounding region of each joint is small compared to the whole image, we reduce the dimensionality and greatly improve efficiency. Additionally, it is possible to dense sample the local region of a joint to capture more discriminative information. Moreover, the features are discriminative enough to characterize variations in different joints. Based on the above stacked ISA model, we compute the spatio-temporal features directly from JV_i for each joint. We treat the spatio-temporal features as the resulting descriptors of the spatio-temporal interest points. Each interest point is represented by a subvolume, which consists of s_t depth image patches of size $s_x \times s_y$ (see Fig. 2a and Fig. 3). The spatio-temporal interest points of GISA features and LISA features are *global depth subvolumes* and *local depth subvolumes*, respectively. We dense sample the interest points from JV_i. Then, we perform a vector quantization by clustering the spatio-temporal feature from the 20 joints, which result in a bag-of-word histogram feature of each joint. With two types of spatio-temporal features (LISA and GISA features), we obtain two histogram features at each joint, named $Joint\_GISA_i$, $Joint\_LISA_i$ (see Fig. 3 and Table 1). For each joint, we apply histogram operations (e.g. concatenation) to the histograms $Joint\_GISA_i$, $Joint\_LISA_i$, which results in a new histogram feature $Joint\_GL\_ISA_i=[Joint\_GISA_i, Joint\_LISA_i]$. The concatenation operation fuses two types of histogram features to provide robustness to classification problems, a technique which has proven useful in the image classification domain [53]. As different features present human actions from different perspectives, concatenation can further be enhanced by introducing broader characteristics. For this, each 3D joint is associated with two histogram features $Joint\_GISA_i$, $Joint\_LISA_i$ and their concatenation $Joint\_GL\_ISA_i$. Each of these corresponds to the feature $f_v^{\mathcal{X}}$ of a node v in $\mathcal{G}^{\mathcal{X}}$. In the following, we will refer to these three as *joint-based* features.

Inspired by the Spatial Pyramid approach [53], we group adjacent joints together as a *joint pair*, seeking to capture the hierarchical structure of the skeleton. In our

Table 1
Implementation details of six types of histogram features used in 3.5D depth video representations.

Video data	Spatio-temporal feature	Joint/joint pair	Histogram operation	Feature
Depth + skeleton	GISA feature	Joint		Joint_GISA
		Joint pair		Joint_GISAp
	LISA feature	Joint	N/A	Joint_LISA
		Joint pair		Joint_LISAp
	GISA+LISA feature	Joint		Joint_GL_ISA
		Joint pair	Concatenation	Joint_GL_ISAp

human skeleton model, there are 19 *joint pairs*. Each *joint pair* is represented by three histogram features $Joint\_GISAp_{ij}=[Joint\_GISA_i,\ Joint\_GISA_j]$, $Joint\_LISAp_{ij}=[Joint\_LISA_i, Joint\_LISA_j]$, and their concatenation $Joint\_GL\_ISAp_{ij}=[Joint\_GL\_ISAp_i,\ Joint\_GL\_ISAp_j]$. We call them *joint-pair-based features*. In total, we have defined six types of 3.5D representations $\mathcal{G}^{\mathcal{X}}$ for depth video $\mathcal{X}$ (see Table 1). For each type of representation, the features $f_v^{\mathcal{X}}$ are given by $Joint\_GISA_i$, $Joint\_LISA_i$, $Joint\_GL\_ISA_i$, $Joint\_GISAp_{ij}$, $Joint\_LISAp_{ij}$, and $Joint\_GL\_ISAp_{ij}$, respectively. To clarify further explanations, we will omit the subscript from the above six features in the rest of the paper.

4. Ensemble learning with discriminative MKL classifiers

In order to model intra-class variations better and provide more robustness against errors of the skeleton tracker [5], we propose an ensemble learning approach for action recognition using depth videos. The aim of our method is to learn a discriminative subset of joints for each action class. To achieve this, we combine two concepts: (1) *Discriminative training* to explore the 3.5D representation effectively; (2) *Ensemble learning* to learn a stronger classifier at a high efficiency. Specifically, we develop an ensemble multi-kernel learning framework (EnMkl) where each component classifier is a discriminative MKL classifier that is trained on a subset of training samples. In our setting, the discriminative training and the ensemble learning can benefit from each other. The MKL framework allows us to consider a subset of joints at a time, which allows us to explore the 3.5 representation efficiently, and in a systematic way. Ensemble learning selects a subset of training samples to explore the diversity of the sample data and therefore it can balance the distribution of the dataset (especially for a small size dataset) and reduce redundancy in the feature set.

An overview of the EnMkl approach we use is shown in Algorithm 1. We first describe the framework of our algorithm. Next, we give more details of the kernel design of the component classifiers.

4.1. Multi-kernel learning

Joint-based features provide useful, characteristic data to allow action recognition. However, redundant or irrelevant information may complicate classification; typically, joint data may be very noisy when occlusions occur, hindering the classifier from isolating relevant information.

When dealing specifically with skeletal data obtained by a skeleton tracker [5] from an RGBD camera, it can be seen that some joints are more important than others with respect to action recognition. Therefore, taking this observation into account, we investigate discriminative joint subsets for human actions by the MKL algorithm. MKL is used to learn an optimal combination of *joint-based* (or *joint pair-based*) features $\{f_{v_1}^{\mathcal{X}}, f_{v_2}^{\mathcal{X}} \ldots f_{v_k}^{\mathcal{X}}\}$. With each kernel corresponding to each feature, different weights are learned for each joint. Weights can therefore highlight more discriminative joints for an action and ignore irrelevant or unnecessary joints by setting their weight to zero.

4.2. Ensemble learning with MKL classifiers

The properties of training datasets such as size, distribution and number of attributes significantly contribute to the generalization error of a learning machine. In action recognition tasks, class imbalances or unevenly distributed sample data is rather common. Because of the large effort of acquiring video data and manually annotating these data, the size of the training data for action recognition is typically smaller than in other computer vision tasks. In addition, different subjects perform actions with considerable variation. These complications may—without precautions being taken—lead to models that suffer from overfitting.

To deal with these problems, randomization with under-sampling is an effective method. This technique uses a subset of majority class samples to train a classifier. Although the training set becomes balanced and the training process becomes faster, standard under-sampling often suffers from the loss of helpful information concealed in the ignored majority class samples. Inspired by [54], our method considers the distributions of different samples in the training dataset. Rather than randomly sampling subsets of the majority class, we try to balance randomization and discrimination during the training phase of the stronger classifier. For this, we define a threshold θ to evaluate component classifiers in the ensemble learning framework. This way, our algorithm can use random or discriminative sampling subsets of training samples to train a component classifier according to the performance of the component classifier in the previous iteration. (In a control experiment, we limit this ability by using only randomly sampling subsets, observing the recognition rate to drop by 0.7%.) Similar to other ensemble learning approaches, the AdaBoost algorithm [55] is used in our method to train a number of weighted component classifiers. An ensemble of all component classifiers together creates the final classifier. Here, for each class, a multiple kernel learning (MKL) classifier is used as the base learner of an ensemble. MKL is able to mine the dominating sets of joints and learn a linear combination of these discriminative joint-based features, details of which we present in the following section.

4.3. Kernel design of component classifiers

Our objective is to learn a component classifier that, rather than using pre-specified kernels, use kernels that are linear combinations of given base kernels. Suppose that the bags of the depth video $\mathcal{X}$ are represented as

$$f_{\mathcal{X}}=\{f_1, f_2, \ldots, f_{t-1}, f_t\} \tag{3}$$

where t is the number of the features for each depth video. The classifier defines a function $\mathcal{F}(f^{\mathcal{X}})$ that is used to rank the depth video $\mathcal{X}$ by the likelihood of containing an action of interest.

The function $\mathcal{F}$ is learned, along with the optimal combination of histogram features $f^{\mathcal{X}}$, by using the Multiple Kernel Learning techniques proposed in [56]. The function $\mathcal{F}(f^{\mathcal{X}})$ is the discriminant function of a Support

Vector Machine and is expressed as

$$\mathcal{F}(f^{\mathcal{X}}) = \sum_{i=1}^{M} y_i \alpha_i K(f^x, f^i) + b \tag{4}$$

Here, f^i, $i = 1, \ldots, M$ denote the feature histograms of M training depth video datasets, which are selected as a representative by the SVM. $y^i \in \{+1, -1\}$ are their class labels, and K is a positive definite kernel, obtained as a linear combination of base kernels

$$K(f^{\mathcal{X}}, f^i) = \sum_j w_j K(f_j^{\mathcal{X}}, f_j^i) \tag{5}$$

MKL learns both the coefficient α_i and the kernel combination weights w_j. For a multi-class problem, a different set of weights $\{w_j\}$ is learned for each class. We choose a one-vs.-rest strategy to decompose the multi-class problems.

Because of linearity, Eq. (4) can be rewritten as

$$\mathcal{F}(f^{\mathcal{X}}) = \sum_j w_j \mathcal{F}(f_j^{\mathcal{X}}) \tag{6}$$

where

$$\mathcal{F}(f_j^{\mathcal{X}}) = \sum_{i=1}^{M} y_i \alpha_i K(f_j^x, f_j^i) + b \tag{7}$$

With each kernel corresponding to each feature, there are 20 weights w_j to be learned for the linear combination of the *joint-based* features, and 19 weights w_j to be learned for the *joint pair-based* features. These weights represent how discriminative a joint is for an action; we can even ignore less discriminative joints by setting w_j to zero.

As MKL cannot give a posterior class probability $P(y = 1|X)$, we propose an approximation of the posteriors by a sigmoid function

$$P_m(y = 1|\mathcal{X}) \approx pro\left(\mathcal{F}_\tau^{\mathcal{X}}\right) \equiv \frac{1}{1 + \exp(A_m \mathcal{F}_\tau^{\mathcal{X}} + B_m)} \tag{8}$$

We follow Platt's method to learn A_m and B_m [57]. For each MKL model m, we then learn a sigmoid function $pro(\mathcal{F}_\tau)$.

Algorithm 1. EnMkl.

Input: For the training set of each action class, select all positive samples $\mathcal{P}$, and all negative samples $\mathcal{N}$, $|\mathcal{P}| < |\mathcal{N}|$, $y^i \in \{+1, -1\}$ are their class labels. Define T as the number of iterations to train an AdaBoost ensemble $\mathcal{C}$.

Weights initialization for each sample: $r_\tau^i = 1/(|\mathcal{P}| + |\mathcal{N}|)$, $i = 1, \ldots, |\mathcal{P}| + |\mathcal{N}|, \tau = 1$,

mode = *top*

while $\tau \le T$ **do**

Weights normalization: $\overline{r}_\tau^i = \frac{r_\tau^i}{\sum_i r_\tau^i}, \quad \forall i$ (9)

if *mode* = = *top* **then**

Select top weighted samples: a subset $\mathcal{N}_\tau$ from $\mathcal{N}$

end if

Train an MKLSVM component classifier, $\mathcal{F}_\tau$ on $\mathcal{P}$ and $\mathcal{N}_\tau$

Compute the performance of $\mathcal{F}_\tau$ over $\mathcal{P}$ and $\mathcal{N}$: $p_\tau = \sum r_\tau^i g_\tau^i (1 - abs(sgn(\mathcal{F}_\tau^i) - y^i))$

where $g_\tau^{i1} = ((1 - sgn(\mathcal{F}_\tau^i))/2 + pro(\mathcal{F}_\tau^i) sgn(\mathcal{F}_\tau^i))$ (10)

pro() denotes the probability output of $\mathcal{F}_\tau^i$

Choose $\alpha_\tau = -\frac{1}{2}\log\left(\frac{1-p_\tau}{p_\tau}\right) \quad \alpha_\tau > \theta$ **then**

mode = *top*; $\tau = \tau + 1$

Update the weights: $r_\tau^{i+1} = \overline{r}_\tau^i e^{(-2|g_\tau^i| + \alpha_\tau)(1 - abs(sgn(\mathcal{F}_\tau^i) - y^i))}, \quad \forall i$ (11)

else

mode = *random*; Select a random subset $\mathcal{N}_\tau$ from $\mathcal{N}$

continue

end if

end while

Output: $\mathcal{C} = \frac{\sum_{\tau=1}^{T} \alpha_\tau pro(\mathcal{F}_\tau)}{\sum_{\tau=1}^{T} \alpha_\tau}$ (12)

5. Evaluation

In this section, we first compare our algorithm quantitatively against current state-of-the-art 3D action recognition algorithms, measuring recognition accuracies on the MSRAction3D dataset. After that, we further analyze the efficiency of our approach in a 3D action recognition system. In addition, we study the general advantages of discriminative MKL classifiers in the field of action recognition. In a more specific evaluation, we discuss the discriminative joint subset for each action class, and we study how many joints in a depth video are sufficient to perform certain action detection and recognition tasks in our framework.

5.1. Experimental setup

The MSRAction3D dataset [1] is a public dataset that provides sequences of depth maps and skeletons captured by a Kinect RGBD camera. It includes 20 actions performed by 10 subjects facing the camera during performance. Each subject performs each action two or three times. As shown in Fig. 6, actions in this dataset reasonably capture a variety of motions related to arms, legs, torso, and their combinations. In order to facilitate a fair comparison, we follow the same experimental settings as [1,2,36] to split 20 actions into three subsets as listed in Table 2, each having 8 action classes. In each subset, half of the subjects are selected as training data and the other half for testing; we perform a two-fold cross validation.

5.2. Sensitively analysis

We analysis the effect of several parameters of our model: the size of the input unit of ISA model, dense sampling stride, codebook size, kernel type. We show the results across different parameter setting for LISA and GISA features by using a three-fold cross-validation on training data: Cross Subset 1 (see Table 4).

We first evaluate the effect of the size of the input units. The input units to the bottom layer of ISA model are of size $s_x \times s_y \times s_t$. We report results of our model with a different spatial size s_x ($s_x = s_y$) and a different temporal size s_t of the input units. Fig. 7 shows the average classification accuracies using cross-validation. Increasing the spatial and temporal size of the input units improves the performance up to $s_x = 12$. This is probably due to the fact that input units need to have a minimum size to dense sample enough interest points. We observe the best result with the size of the input unit of 12 pixels $\times$ 12 pixels $\times$ 10 frames.

With respect to the dense sampling stride, Fig. 8 presents the results for 1–4 pixels. The performance increases

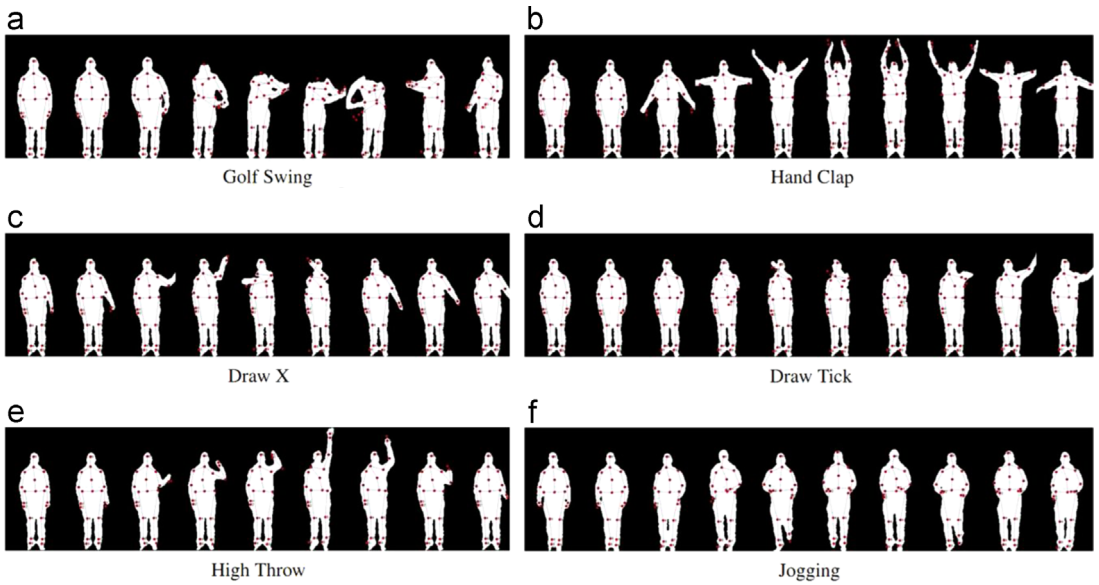

Fig. 6. The sequences of depth maps and skeleton for different action classes. Each depth image includes 20 joints (marked as red points). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Table 2
Partitioning of the MSRAction3D dataset into three subsets as used in our evaluation.

Subset Classes	Cross subset 1 (CS1)	Cross subset 2 (CS2)	Cross subset 3 (CS3)
Action classes	Tennis Serve (TSr)	High Wave (HiW)	High Throw (HT)
	Horizontal Wave (HoW)	Hand Catch (HC)	Forward Kick (FK)
	Forward Punch (FP)	Draw X (DX)	Side Kick (SK)
	High Throw (HT)	Draw Tick (DT)	Jogging (JG)
	Hand Cap (HCp)	Draw Circle (DC)	Tennis Swing (TSw)
	Bend (BD)	Hands Wave (HW)	Tennis Serve (TSr)
	Hammer (HM)	Forward Kick (FK)	Golf Swing (GS)
	Pickup Throw (PT)	Side Boxing (SB)	Pickup Throw (PT)

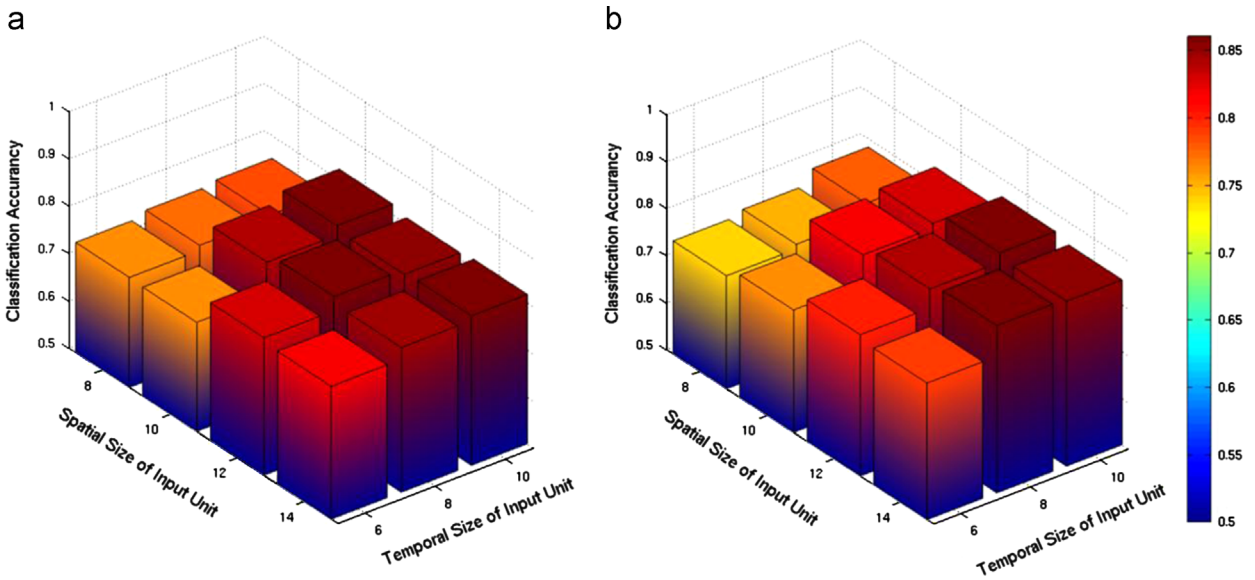

Fig. 7. Effect of spatial and temporal size of the input units with GISA feature (a) and LISA feature (b) on classification accuracy using cross-validation.

with a higher sampling density. This is also consistent with dense sampling at regular position where more features in general improve the results [58]. A sampling stride of 1 pixel samples every pixel which increase the computational complexity. We set the dense sampling stride as 2 pixels, which offers a good trade-off between speed and accuracy.

Fig. 9 shows the classification performance for different combinations of kernels and codebook sizes. The χ^2 kernel outperforms the intersection kernel. Larger codebook sizes have been reported to improve the classification performance. For both kernels, the performance saturates at codebook size = 700 or codebook size = 900.

5.3. Model details

We use the found optimal parameters for sensitive analysis to train our models and test our method. We train the ISA model on the MSRAction3D training sets. The input units to the bottom layer of ISA model are of size $12 \times 12 \times 10$, which are the dimensions of the spatial and temporal sizes of the subvolumes. The size is the same for global depth subvolumes (for learning GISA features) and local depth subvolumes (for learning LISA features). The subvolumes of the top layer of the ISA model are of the same size as those of the bottom layer.

We perform vector quantization by k-means on the learned spatio-temporal features for each joint. For the distance parameter in the dense sampling step for the local regions of each joint, we choose a region of 2 pixels. The codebook size k is 700 for both *Joint_GISA* feature and *Joint_LISA* feature. Therefore, each depth video is represented by 20 histogram features for *Joint_GISA*, *Joint_LISA*, *Joint_GL_ISA* or 19 histogram features for *Joint_GISAp*, *Joint_LISAp*, and *Joint_GL_ISAp*. We choose χ^2 as the histogram kernel for the multi-class SVM classifier. For EnMkl, we set the number of subsets $|\mathcal{N}_\tau| = 3|\mathcal{P}|$ and the rounds of the AdaBoost $T = 20$. The threshold for a good component classifier is set to 1.45. Across the three subsets, all parameters are set to the same values. Note that when we set the number of the samples in subsets $|\mathcal{N}_\tau| = |\mathcal{N}|$, and the rounds of the AdaBoost $T = 1$, EnMkl becomes equivalent to a multi-kernel learning problem; we call this special case EnMkl-s.

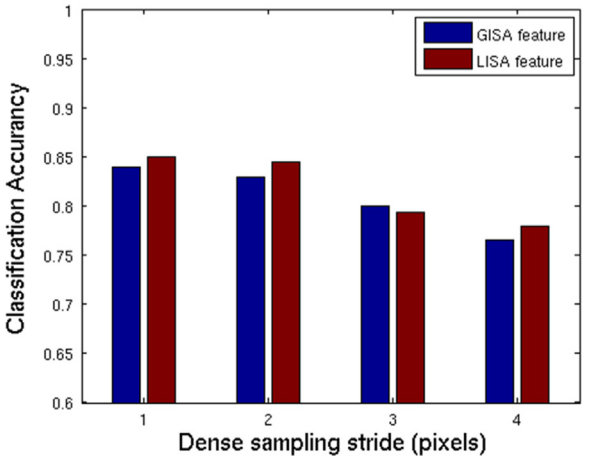

Fig. 8. Effect of the dense sampling stride with GISA feature and LISA feature on classification accuracy using cross-validation.

5.4. Experimental results

We compare our algorithm with several recent methods including: (1) Li et al. [1], where bags of 3D points are sampled from depth maps and an action graph is employed to model the dynamics of the actions; (2) Yang and Tian [23], who design a new type of feature set based on position differences of joints; (3) Wang et al. [20], where the depth sequence is randomly sampled and the most discriminative samples are selected and described using LOP descriptors; (4) Wang et al. [32], where local occupancy pattern features are used over the skeleton

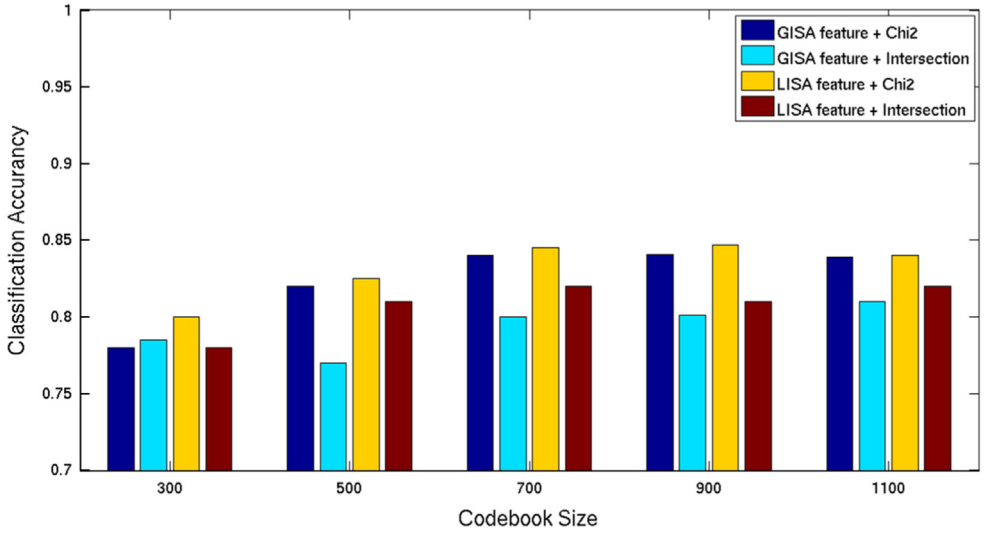

Fig. 9. Effect of the codebook size and kernel type with GISA feature and LISA feature on classification accuracy using cross-validation.

joints; (5) Oreifej and Liu [2], who describe the depth sequence using a histogram that captures the distribution of surface normal orientations in 4D space; (6) Xia and Aggarwal [36], who employ a filtering method to extract STIPs from depth videos; (7) Wang et al. [26], where observations are represented by histograms of activating spatial and temporal part-sets; (8) Gowayyed et al. [59], who design a 2D trajectory descriptor, the histogram of oriented displacements (HOD).

A comparison of our method against the best published results for the MSRAction3D dataset is reported in Table 3. Because we test six types of 3.5D representations $\mathcal{G}^{\mathcal{X}}$ for two models EnMkl and EnMkl-s, Table 3 shows 12 results in total. As can be seen from the table, our approach outperforms a wide range of methods. There is a clear increase in performance of our method EnMkl (with *Joint_GL_ISAp* feature) (92.3%) compared to the closest competitive method (91.3%). Note that the absolute performance is very good, considering that failures in the skeleton tracker are quite frequent and tracked joint positions are rather noisy. The obtained accuracy of EnMkl-s (with *Joint_GL_ISAp* features), a special case of EnMkl without using ensembles, is 91.2%. These encouraging results illustrate the effectiveness of our unsupervised learning features.

Table 3
Comparison of recognition accuracy between previous methods and our proposed approach on the MSRAction3D dataset.

Method	Accuracy
Action graph on bag of 3D points [1]	0.747
EigenJoints [23]	0.823
Random occupancy pattern [20]	0.865
Mining actionlet ensemble [32]	0.882
Histogram of oriented 4D normals [2]	0.889
ST depth cuboid similarity feature [36]	0.893
Pose-based action recognition [26]	0.902
Histogram of oriented displacements [59]	**0.913**
EnMkl-s + Joint_GISA	0.879
EnMkl-s + Joint_GISAp	0.896
EnMkl-s + Joint_LISA	0.895
EnMkl-s + Joint_LISAp	0.912
EnMkl-s + Joint_GL_ISA	0.894
EnMkl-s + Joint_GL_ISAp	**0.914**
EnMkl + Joint_GISA	0.887
EnMkl + Joint_GISAp	0.901
EnMkl + Joint_LISA	0.903
EnMkl + Joint_LISAp	0.920
EnMkl + Joint_GL_ISA	0.903
EnMkl + Joint_GL_ISAp	0.923

Compared to EnMkl-s, the improvement of EnMkl is about 1%. This indicates that the ensemble learning approach can better capture intra-class variations and is more robust against noise and errors in depth maps and joint positions. This observation is consistent with [32], who report that accuracy decreases when the ensemble approach is disabled in their experiments. It is also important to note that in our methods, accuracies obtained using LISA features (91.2% for EnMkl-s with *Joint_LISAp*) are better than using GISA features (89.6% for EnMkl-s with *Joint_GISAp*). This is probably because the skeletons have a natural correspondence over time and LISA features can model spatial structures more explicitly than GISA features. To further investigate the relationship between LISA features and GISA features, we study the most important joints discovered by *Joint_LISA* and *Joint_GISA* features with the EnMkl-s method. For each action class, the *top-weighted* joint is selected as the most important joint. Here, we define the *top-weighted* joint as the joint with the highest maximum. With *Joint_LISA* features, *right hand*, *right wrist*, and *left wrist* joints (the top three) receive the most votes in 20 action classes in the MSRAction3D dataset. With *Joint_GISA* features, *right hand*, *right shoulder*, and *left elbow* joints receive the most votes (the top three). The results indicate that LISA and GISA features have some qualities in common, as both of them select *right hand* as the highest weighted joint. Our results are consistent with [59], who conducted an experiment using features from only one joint to perform action recognition. Their results

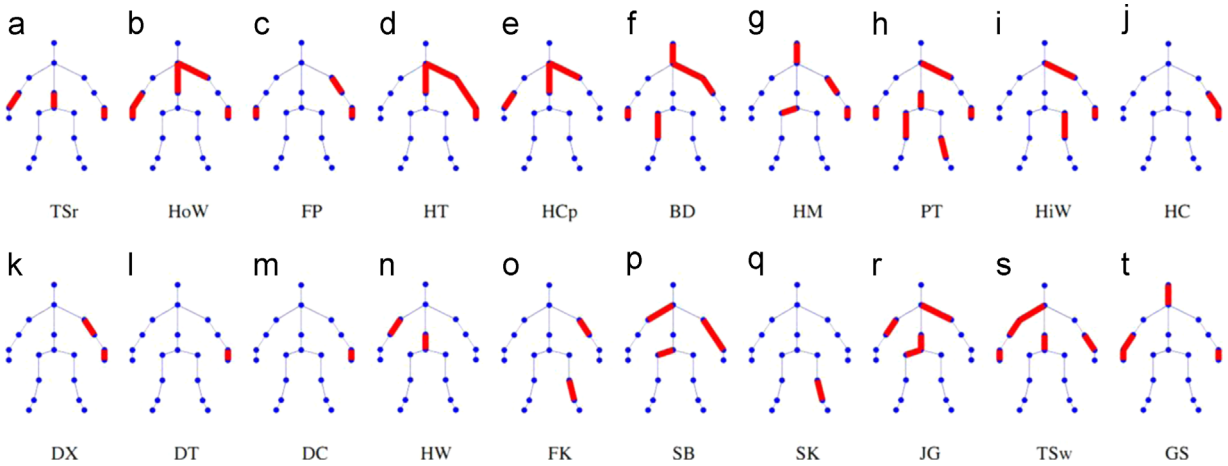

Fig. 10. The joint subsets used to recognize the 20 action classes in the MSRAction3D dataset. Our method can learn discriminative joint subsets for each action class. The weight associated with each joint describes how discriminative a joint is for that action. Joints with weights >0 are highlighted as thick, red lines. All abbreviations of action classes are defined in Table 2. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

show that using the right hand joint outperforms all other joints on the MSRAction3D dataset. Additionally, it is interesting to note that in our methods, accuracy obtained using *Joint_GISAp*|*Joint_LISAp* features is 90.1%/92% (EnMkl), which is better than using *Joint_GISA*|*Joint_LISA* features 88.7%/90.3% (EnMkl). These results show a clear advantage of the spatial pyramid approach, even though we simply group adjacent joints together as *joint pairs* and capture the hierarchical structure of human skeleton.

In Table 4, we report average accuracies of all three test sets (*Cross Subset* 1 (CS1), *Cross Subset* 2 (CS2), *Cross Subset* 3 (CS3)) and show the best performance results of the two methods, EnMkl-s and EnMkl. In Fig. 11, we illustrate the average accuracies of all action class. While the performance in CS2 and CS3 is promising, the accuracy in CS1 is relatively low. This is probably because actions in CS1 are performed with rather similar movements. For example, in CS1 *Hammer* tends to be confused with *Forward Punch* and *Horizontal Wave*, and *Pickup Throw* consists of *Bend* and *High Throw*. Although our method reaches an accuracy of 100% in 12 out of 20 actions, the accuracy of the *Hammer* in CS1 is only 37.7%. This is probably due to the significant variations of the action *Hammer* performed by different subjects; recognition performance could be improved by adding more subjects to the training set.

5.5. Advantages of multi-kernel learning

It is generally agreed that, although the human body has a large number of kinematic joints, a certain action is usually only associated with a subset of them. Additionally, feature extraction in action recognition is usually computationally expensive. A reduced feature subset leads to lower computational costs. This encourages us to investigate the following two questions: do more joints allow for better for action recognition? Do joints contribute equally to recognizing an action?

We address the first question by setting the following control experiment: we conduct two tests, where the first test uses 20 joint features with equal weights for action recognition and the second test uses a subset (the subset is obtained by the EnMkl method) of joint features with equal weights (manually setting w_j to 1). We perform both tests on the MSRAction 3D dataset. It is not surprising that the first test performs worse than the second, with a decline of 4.5% in accuracy on the MSRAction 3D dataset. This indicates that a subset of characteristic data may lead to a more successful recognition and a full set of data with irrelevant information may complicate the classification.

To answer the second question whether joints contribute equally to an action, we re-run the experiment with the same settings as in Section 5.3 and manually set w_j to 1. The results of this test show substantially worse accuracies than those of the previous experiments. More precisely, setting the weight to 1, accuracy drops by a significant amount of 5% for the MSRAction3D dataset. This confirms that the weights learned from MKL are indeed very relevant for successful action recognition.

Table 4
Recognition accuracy of our method on each of the three subsets. CS1, CS2 CS3 are the abbreviations of Cross subset 1, Cross subset 2, Cross subset3 (see Table 2).

Method	CS1	CS2	CS3
EnMkl-s +Joint_GL_ISAp	**0.882**	0.898	0.951
EnMkl + Joint_GL_ISAp	0.881	**0.927**	**0.959**

5.6. Mining discriminative joint subsets

In our EnMkl-s method, each action is represented as a linear combination of joint-based features. We learn their weights in a multiple kernel learning method to obtain discriminative joint subsets.

Fig. 10 illustrates the skeleton with joints weights obtained by our EnMkl-s method. Here, we only show the results of the *Joint_LISA* features as an example; the other five feature sets would show to very similar results. The *Joint_LISA* features with *weights* >0 are marked as thick, red lines. The average number of *Joint_LISA* features for 20 actions in the MSRAction3D dataset is four. Three of 20 action classes have only one discriminative *Joint_LISA* feature. This result is rather interesting: imagining we want to recognize or detect a specific action class; we only need to extract features from one joint rather than the entire video data. This can be implemented and executed at a high efficiency.

EnwMi-s is also able to deal with tracking errors in the skeleton data and can better capture intra-class variation.

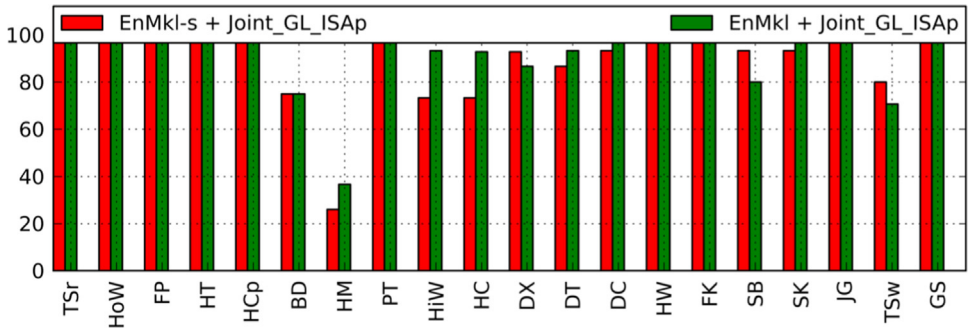

Fig. 11. Recognition accuracies for the 20 action classes of the MSRAction3D dataset. We compare EnMkl to EnMkl-s using Joint_GL_ISAp features. All abbreviations of action classes are defined in Table 2. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Fig. 10(t) shows that *Golf Swing* is represented by the combination of the joints *head, neck, right hand, right wrist, left hand, left wrist and left elbow* (see Fig. 5 for the definition of the joint labels). Fig. 10(a) shows that *Tennis Serve* is represented by the combination of the joints *left elbow, left wrist, right hand, right wrist, torso, and waist*. It is obvious that different action classes have different discriminative joint subsets. Fig. 10(r) shows that *Jogging* is represented by the combination of the joints *left elbow, left shoulder, left hip, waist, torso, neck and right shoulder*. Normally, one would expect *Jogging* to be related to the left and right feet or ankles. However, in the MSRAction3D dataset, the tracked positions of the joints *right/left foot, and right/left ankle* are very noisy (see Fig. 6). Therefore, these joints are not discriminative for the action class *Jogging*, which is consistent with Fig. 6(f). This shows that our method is rather robust against tracking errors in the skeleton data.

5.7. Computational complexity

The training phase of neural networks (e.g. unsupervised feature learning) is usually computationally expensive and requires much tuning. The ISA algorithm, however, does not need the tweaking with the learning rate or the convergence criterion. For the training stage, the ISA algorithm takes 3–4 h to learn the stacked ISA model using the setting in Section 5.3.

To analyze the computational complexity of the feature extraction, we extract features with dense sampling on 20 video clips from the MSRAction3D dataset. The run-time is obtained on a notebook with a 2.3 GHz double-core CPU and 8 GB RAM. The implementation is in unoptimized and un-parallelized MATLAB. Feature extraction using our method runs 1 frame per second with the entire video and 6 frames per second with specific portions of the video (the surrounding regions of the joints of the subject which performs the action in the video). As the extraction time relies heavily on matrix vector products, it can be implemented and executed much more efficiently on a GPU.

6. Conclusion

We present a novel ensemble learning approach, named EnMkl, which combines unsupervised feature learning and discriminative feature mining. For this, we develop two types of spatio-temporal features, applying independent subspace analysis to depth video data. Our approach is rather generic and unsupervised, and may therefore be applied to a wider range of problems with unlabeled sensor data. To the best of our knowledge, EnMkl is the first attempt to learn the spatio-temporal features from depth video data in an unsupervised way. Furthermore, we propose an ensemble learning approach with discriminative multi-kernel-learning classifiers, which allows for a better characterization of inter-class variations in the presence of noisy or erroneous skeleton data. In our evaluation, we analyze the efficiency of our 3D action recognition approach. In more detailed discussions, we investigate which joint subsets are discriminative for different types of actions, and we study which of these joints is sufficient to recognize these actions. Our experimental results of the EnMkl approach show a performance superior to existing techniques. Results also suggest that learning spatio-temporal features directly from depth video data may be a promising direction for future research, as combining these features with ensemble learning may further increase performance.

References

[1] W. Li, Z. Zhang, Z. Liu, Action recognition based on a bag of 3d points, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2010, pp. 9–14.
[2] O. Oreifej, Z. Liu, Hon4d: histogram of oriented 4d normals for activity recognition from depth sequences, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013.
[3] G. Chen, D. Clarke, A. Knoll, Learning weighted joint-based features for action recognition using depth camera, in: International Conference on Computer Vision Theory and Applications, 2014.
[4] S. Hadfield, R. Bowden, Hollywood 3d: recognizing actions in 3d natural scenes, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3398–3405.
[5] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from single depth images, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011,pp. 1297–1304.
[6] I. Laptev, On space-time interest points, Int. J. Comput. Vis. 64 (2005) 107–123.
[7] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[8] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 886–893.
[9] A. Hyvarinen, J. Hurri, P. Hoyer, Natural Image Statistics: A probabilistic approach to early computational verion, Spring-Verlag, 2009.
[10] Q. Le, W. Zou, S. Yeung, A. Ng, Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 3361–3368.
[11] G. Chen, F. Zhang, M. Giuliani, C. Buckl, A. Knoll, Unsupervised learning spatio-temporal features for human activity recognition from rgb-d video data, in: International Conference on Social Robotics, 2013.
[12] Z.-H. Zhou, Ensemble learning, in: Encyclopedia of Biometrics, 2009, pp. 270–273.
[13] G. Chen, M. Giuliani, D. Clarke, A. Gaschler, A. Knoll, Action recognition using ensemble weighted multi-instance learning, in: IEEE International Conference on Robotics and Automation, 2014.
[14] A. Hernandez-Vela, M. Bautista, X. Perez-Sala, V. Ponce, X. Baro, O. Pujol, C. Angulo, S. Escalera, Bovdw: bag-of-visual-and-depth-words for gesture recognition, in: International Conference on Pattern Recognition, 2012, pp. 449–452.
[15] Y. Zhao, Z. Liu, L. Yang, H. Cheng, Combing rgb and depth map features for human activity recognition, in: Signal Information Processing Association Annual Summit and Conference, 2012, pp. 1–4.
[16] B. Ni, G. Wang, P. Moulin, Rgbd-hudaact: a color-depth video database for human daily activity recognition, in: IEEE International Conference on Computer Vision Workshops, 2011, pp. 1147–1153.
[17] O. Oshin, A. Gilbert, R. Bowden, Capturing the relative distribution of features for action recognition, in: IEEE International Conference on Automatic Face and Gesture Recognition and Workshops, 2011, pp. 111–116.
[18] Z. Cheng, L. Qin, Y. Ye, Q. Huang, Q. Tian, Human daily action analysis with multi-view and color-depth data, in: ECCV Workshops and Demonstrations, Lecture Notes in Computer Science, vol. 7584, pp. 52–61.
[19] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: application to face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 2037–2041.
[20] J. Wang, Z. Liu, J. Chorowski, Z. Chen, Y. Wu, Robust 3d action recognition with random occupancy patterns, in: European Conference on Computer Vision, 2012, pp. 872–885.
[21] J. Yu, M. Wang, D. Tao, Semisupervised multiview distance metric learning for cartoon synthesis, IEEE Trans. Image Process. 21 (2012) 4636–4648.

[22] J. Yu, D. Liu, D. Tao, H.S. Seah, On combining multiple features for cartoon character retrieval and clip synthesis, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 42 (2012) 1413–1427.
[23] X. Yang, Y. Tian, Eigenjoints-based action recognition using Naïve–Bayes-nearest-neighbor, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 14–19.
[24] V. Bloom, D. Makris, V. Argyriou, G3d: a gaming action dataset and real time action recognition evaluation framework, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 7–12.
[25] L. Xia, C.-C. Chen, J. Aggarwal, View invariant human action recognition using histograms of 3d joints, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 20–27.
[26] C. Wang, Y. Wang, A. Yuille, An approach to pose-based action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 915–922.
[27] G. Dong, J. Li, Efficient mining of emerging patterns: discovering trends and differences, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999, pp. 43–52.
[28] D. Tao, X. Tang, X. Li, X. Wu, Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 1088–1099.
[29] D. Tao, X. Li, X. Wu, S. Maybank, General tensor discriminant analysis and Gabor features for gait recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 1700–1715.
[30] D. Tao, X. Li, X. Wu, S. Maybank, Geometric mean for subspace selection, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 260–274.
[31] G. Chen, F. Zhang, D. Clarke, A. Knoll, Learning to track multi-target online by boosting and scene layout, in: 12th International Conference on Machine Learning and Applications (ICMLA), vol. 1, 2013, pp. 197–202.
[32] J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining actionlet ensemble for action recognition with depth cameras, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1290–1297.
[33] L. Zhu, Y. Chen, Y. Lu, C. Lin, A. Yuille, Max margin and/or graph learning for parsing the human body, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[34] B. Yao, L. Fei-Fei, Grouplet: a structured image representation for recognizing human and object interactions, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 9–16.
[35] G. Chen, D. Clarke, D. Weikersdorfer, M. Giuliani, A. Gaschler, D. Wu, A. Knoll, Multi-modality gesture detection and recognition with un-supervision, randomization and discrimination, in: ChaLearn Looking at People Workshop, European Conference on Computer Vision (ECCV2014), 2014.
[36] L. Xia, J. Aggarwal, Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013.
[37] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, R. Bajcsy, Berkeley mhad: a comprehensive multimodal human action database, in: IEEE Workshop on Applications of Computer Vision, 2013, pp. 53–60.
[38] H. Rehmani, A. Mahmood, A. Mian, D. Huynh, Real time action recognition using histograms of depth gradients and random decision forests, in: IEEE Winter Applications of Computer Vision Conference, 2014.
[39] M. Reyes, G. Dominguez, S. Escalera, Feature weighting in dynamic time warping for gesture recognition in depth data, in: IEEE International Conference on Computer Vision Workshops, 2011, pp. 1182–1188.
[40] J. Wang, Y. Wu, Learning maximum margin temporal warping for action recognition, in: The IEEE International Conference on Computer Vision, 2013.
[41] A. Gaschler, S. Jentzsch, M. Giuliani, K. Huth, J. de Ruiter, A. Knoll, Social behavior recognition using body posture and head pose for human-robot interaction, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012.
[42] A. Gaschler, K. Huth, M. Giuliani, I. Kessler, J. de Ruiter, A. Knoll, Modelling state of interaction from head poses for social Human-Robot Interaction, in: ACM/IEEE HCI Conference on Gaze in Human-Robot Interaction Workshop, 2012.
[43] W. Li, Z. Zhang, Z. Liu, Expandable data-driven graphical modeling of human actions based on salient postures, IEEE Trans. Circuits Syst. Video Technol. 18 (2008) 1499–1510.
[44] A. Vieira, E. Nascimento, G. Oliveira, Z. Liu, M. Campos, Stop: space-time occupancy patterns for 3d action recognition from depth map sequences, in: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Lecture Notes in Computer Science, vol. 7441, 2012, pp. 252–259.
[45] A. Kurakin, Z. Zhang, Z. Liu, A real time system for dynamic hand gesture recognition with a depth sensor, in: European Signal Processing Conference, 2012, pp. 1975–1979.
[46] B. Geng, D. Tao, C. Xu, Y. Yang, X.-S. Hua, Ensemble manifold regularization, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2012) 1227–1233.
[47] J. Yu, D. Tao, M. Wang, Adaptive hypergraph learning and its application in image classification, IEEE Trans. Image Process. 21 (2012) 3262–3272.
[48] J. Yu, Y. Rui, D. Tao, Click prediction for web image reranking using multimodal sparse coding, IEEE Trans. Image Process. 23 (2014) 2019–2032.
[49] X.-S. Xu, Y. Jiang, X. Xue, Z.-H. Zhou, Semi-supervised multi-instance multi-label learning for video annotation task, in: Proceedings of the 20th ACM International Conference on Multimedia, 2012, pp. 737–740.
[50] G.E. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (2006) 1527–1554.
[51] J.-K. Kamarainen, V. Kyrki, H. Kälviäinen, Invariance properties of Gabor filter based features—overview and applications, IEEE Trans. Image Process. 15 (2006) 1088–1099.
[52] J. Read, G. Phillipson, I. Serrano-Pedraza, A. Milner, A. Parker, Stereoscopic vision in the absence of the lateral occipital cortex, PLoS One 5 (2010).
[53] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 2169–2178.
[54] X.-Y. Liu, J. Wu, Z.-H. Zhou, Exploratory undersampling for class-imbalance learning, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 39 (2009) 539–550.
[55] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1997) 119–139.
[56] S.V.N. Vishwanathan, Z. Sun, N. Theera-Ampornpunt, M. Varma, Multiple kernel learning and the SMO algorithm, in: Advances in Neural Information Processing Systems, vol. 23, 2010.
[57] J.C. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: Advances in Large Margin Classifiers, 1999, pp. 61–74.
[58] H. Wang, M.M. Ullah, A. Kläser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, in: British Machine Vision Conference, 2009.
[59] M.A. Gowayyed, M. Torki, M.E. Hussein, M. El-Saban, Histogram of oriented displacements (HOD): describing trajectories of human joints for action recognition, in: International Joint Conference on Artificial Intelligence, 2013, pp. 1351–1357.