# Action Recognition From Arbitrary Views Using Transferable Dictionary Learning

Jingtian Zhang, *Student Member, IEEE*, Hubert P. H. Shum, *Member, IEEE*, Jungong Han, *Member, IEEE*, and Ling Shao, *Senior Member, IEEE*

***Abstract*— Human action recognition is crucial to many practical applications, ranging from human-computer interaction to video surveillance. Most approaches either recognize the human action from a fixed view or require the knowledge of view angle, which is usually not available in practical applications. In this paper, we propose a novel end-to-end framework to jointly learn a view-invariance transfer dictionary and a view-invariant classifier. The result of the process is a dictionary that can project real-world 2D video into a view-invariant sparse representation, and a classifier to recognize actions with an arbitrary view. The main feature of our algorithm is the use of synthetic data to extract view-invariance between 3D and 2D videos during the pre-training phase. This guarantees the availability of training data, and removes the hassle of obtaining real-world videos in specific viewing angles. Additionally, for better describing the actions in 3D videos, we introduce a new feature set called the *3D dense trajectories* to effectively encode extracted trajectory information on 3D videos. Experimental results on the IXMAS, N-UCLA, i3DPost and UWA3DII data sets show improvements over existing algorithms.**

***Index Terms*— Action recognition, 3D dense trajectories, view-invariance, transfer dictionary learning.**

## I. Introduction

2D VIDEO based human action recognition has attracted a lot of attention in security surveillance and human-computer interaction. Various spatio-temporal appearances generated from the movements can be considered as the feature descriptors for action recognition. These include spatio-temporal pattern template [1], spatio-temporal interest points [2]–[5], shape matching [6], [7] and motion trajectories based descriptors [8]–[11]. Among them, dense trajectories based methods have achieved state-of-the-art results by extracting densely sampled trajectories-aligned descriptors in the optical flow fields. Deep learning networks have also achieved significant success in the 2D action recognition area [12]–[15]. These methods can automatically learn spatial-temporal feature representations and identify different action categories. However, [12]–[15] are only effective for single view action recognition and the recognition performance degrades significantly when the viewpoint is changed. The reason behind is that the appearances of a particular action from different viewpoints vary dramatically, which results in dissimilar trajectories.

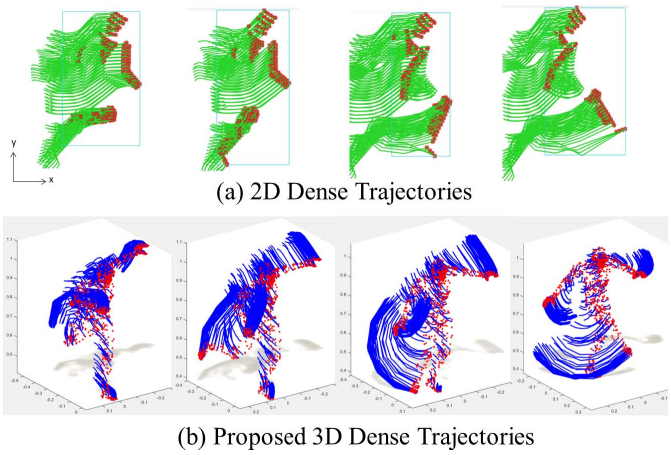



Fig. 1. Leveraging view-invariance from 3D model is a popular idea to tackle arbitrary-view and cross-view action recognition. (a) Existing works [16], [17] project a simplified 3D cylindrical model into as many viewpoints as possible to produce 2D training videos and extract *2D dense trajectories* from these projections. However, some human appearance information could be lost due to the unrealistic 3D reconstruction and the simplified cylindrical model. The discrete projection angles also inevitably result in the loss of 3D geometric information. (b) The proposed *3D dense trajectories* are extracted directly from high-quality 3D human surface model without any projection.

As a result, cross-view action recognition is proposed for bridging the appearance differences between different viewpoints. The main idea is to transfer the knowledge from the source view to the target view, allowing the system to recognize actions from a view that is not included in the training set. Blank *et al.* [1] presented a dynamics-based feature called hankelet that can capture the invariant property in viewpoint change using short tracklets for cross-view recognition. Kovashka and Grauman [18] used an AND-OR graph representation to compactly express the appearance and

Manuscript received June 27, 2017; revised October 30, 2017, March 10, 2018, and April 6, 2018; accepted April 25, 2018. Date of publication May 15, 2018; date of current version June 27, 2018. This project was supported in part by the Engineering and Physical Sciences Research Council under Grant EP/M002632/1 and in part by the Royal Society under Grant IE160609. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xudong Jiang. *(Corresponding author: Ling Shao.)*

J. Zhang and H. P. H. Shum are with the Department of Computer and Information Sciences, Northumbria University, Newcastle Upon Tyne NE1 8ST, U.K. (e-mail: jingtian.zhang@northumbria.ac.uk; hubert.shum@northumbria.ac.uk).

J. Han is with the School of Computing and Communications, Lancaster University, Lancaster LA1 4YW, U.K. (e-mail: jungonghan77@gmail.com).

L. Shao is with the Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates. (e-mail: ling.shao@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2018.2836323

motion variance during viewpoint changes. Lin *et al.* [6] and Lv and Nevatia [7] constructed a continuous path between the target view and the source view to facilitate cross-view action recognition. Farhadi and Tabrizi [19] generated the same split-based features for correspondence video frames from both training and testing views. Such systems are computationally expensive as they not only require feature-to-feature correspondence, but also require mapping between the split-based and the original feature. Liu *et al.* [20] used a bipartite graph to model the relationship between the two codebooks from the source view and target view. Wang *et al.* [21] proposed a Statistical Translation Framework (STF) to estimate the transfer probabilities of the visual words from the source to target views. Huang *et al.* [22] built a correlation subspace to produce joint representation from different views by using canonical correlation analysis. In spite of discovering the correspondence between codebooks from two or more different views, the above approaches cannot guarantee that videos captured from different views share similar features. Also, all these methods require viewpoint information for both source view and target view, which is usually not available in practical applications.

As a solution, arbitrary-view action recognition is proposed, in which viewpoint information is not required during testing and action from unseen views can be recognized. The main idea is to remove view-dependent information from the feature representation. Previous attempts to realize arbitrary-view action recognition have met with varying levels of success. Lv and Nevatia [7] use a graphical model to calibrate 2D key poses of actors to represent 3D surface models for arbitrary view action recognition. However, the motion information for recognizing actions may not be well captured. Weinland *et al.* [23] propose to recognize human actions by estimating 3D exemplars from a single 2D view angle using the hidden Markov model. However, reconstructing these 3D exemplars from a single view is unreliable. Also, detailed action information may be lost as only discrete samples of silhouette information are used. Yan *et al.* [24] present a 4D (i.e., 3D spatial and 1D temporal dimensions) action feature using the time-ordered 3D reconstruction of the actors from multi-view video data. The recognition accuracy depends heavily on the performance of the 3D reconstruction, and the framework requires training data to be captured from carefully designed viewpoints. Gupta *et al.* [16] propose to project the 3D motion capture sequence in the 2D space and explore the best match of each training video using non-linear circular temporary encoding. However, since discrete 2D projection, instead of full 3D information, is used for training, the accuracy depends on the number of projected views. Rahmani *et al.* [25] propose R-NKTM to transfer knowledge of human actions from any unknown view to a shared high-level virtual view by finding a non-linear virtual path that connects the views. They generate the training data by projecting the 3D exemplar to 108 virtual views. The use of so many projected views results in enhanced system performance, but result in a computationally expensive training process. Ideally, we would like to have a framework that relies on easy-to-obtain training data and performs robustly in runtime.

Most of the existing works leverage view-invariance provided by 3D models to realize cross-view or arbitrary-view action recognition. Traditionally, simplified cylindrical models are used [16], [17], which does not generate realistic movement appearance. High-quality reconstruction models are proposed by calculating them from multi-view 2D videos [24]. In order to increase the system robustness to viewpoint changes, training data is forced to cover as much 2D data projected along as many viewpoints as possible. All these approaches suffer from the following problems: (1) The recognition accuracy is highly related to the quality of 3D models. Some human appearance information could be lost due to the unrealistic 3D reconstruction and the simplified cylindrical model; (2) Despite the effort to project the 3D model into as many viewpoints as possible, these discrete projection angles will inevitably result in the loss of 3D geometric information. A large amount of 2D projections also requires larger system capacity and training cost.

To solve the problems, we proposed to synthesize training data using high-quality human models with captured 3D motion data. We employ primary deformation [26] to drive the movement of the models, and motion retargeting [27] to adjust the movement based on the body sizes of the models. We further propose a new 3D feature set called the *3D dense trajectories* including 3D trajectories, 3DHOF and 3DMBH. This allows us to extract the feature directly from 3D videos and avoid geometric information loss due to discrete projection. Finally, we propose a new view-invariant transfer dictionary learning framework, which extracts the view-invariance between 3D and 2D video, to perform arbitrary view action recognition. We pre-train the system with a large number of automatically synthesized 3D and 2D videos. This allows us to train a view-invariant action classifier using only a small number of real-world 2D videos, in which the view information is not annotated. Experimental results show that our system achieves better accuracy when compared with previous work in arbitrary-view and cross-view action recognition.

This paper has three main contributions:

- We propose a new transfer dictionary learning framework that utilizes synthetic 2D and 3D training videos generated from realistic human models to learn a dictionary that can project a real world 2D video into a view-invariant sparse representation, which allows us to train an action classifier that works in an arbitrary view.
- We release our synthetic 2D and 3D dataset for public usage. This is the first structured action dataset built with realistic human models for high-quality action classification.
- We propose a new 3D feature set called the 3D dense trajectories consisting of 3D trajectories, 3DHOF and 3DMBH for a better description of motion in 3D. This can be considered as a 3D counterpart of the popular 2D feature dense trajectories [21].

This paper is based on our previous work presented in [28], but it substantially extends the work in four aspects, which are: (1) We replace the cylinder-based 3D model with several

more realistic 3D human models. The motion is retargeted according to the bone dimensions [27] and skinned to the realistic models [26]. (2) We propose the 3D dense trajectories including 3D trajectories, 3DHOF and 3DMBH to better describe the motion in 3D videos. (3) By jointly training the transfer dictionary pair and the classifier, we build an end-to-end framework with an updated objective function to improve the efficiency and performance of the system. (4) We perform more detailed system evaluation with two more datasets: i3DPost and UWA3DII.

The rest of this paper is organized as follows. In Section II, we introduce some related applications and approaches for the view-invariant action recognition. In Section III, we give an overview of our view-invariant human action recognition frame. In Section IV, we present the synthesis and feature extraction process on our 2D and 3D video data. Section V provides the details of our view-invariant dictionary learning algorithm. Section VI presents the experimental results, and Section VII concludes the paper.

## II. Related Work

The general process for view-invariant action recognition can be divided into three major parts. (1) Synthesized 3D exemplars are used for producing the 2D videos covering as many viewpoints as possible. (2) Then, the feature extraction methods, especially some interest points and trajectory-based feature extraction methods, are developed for describing the action on the 2D videos. (3) At last, transfer learning algorithms are used to transfer the action information across different views in order to realize view-invariant action recognition. Therefore, in this part, some previous works related to these three major processes will be introduced respectively.

### *A. 3D Exemplar-Based Methods*

One popular idea is to utilize 3D exemplars for view-invariant feature extraction and description. Some researchers are only using the static 3D exemplars. For example, Ankerst *et al.* [29] propose the histogram of shape which is very similar to the 3D shape context proposed by Korgen *et al.* [30]. Subsequently, Huang and Hilton [31] combine the histogram of shape with color information. All these methods are mainly based on static descriptors such as poses and shape while the state-of-the-art descriptors integrate static descriptors with motion information.

Instead of relying on the static feature only, some researchers utilize the changing of static descriptors over time in order to capture the temporal information by simply accumulating static descriptors, applying sliding windows, or tracking human pose information [23], [32]–[34]. Cohen and Li [35] present a 3D human shape model for view-invariant human identification. Later, this 3D human shape model was developed by Pierobon *et al.* [34] for human action recognition. Weinland *et al.* [33] propose the Motion History Volume (MVH) as a 3D extension of Motion Histogram Images (MHIs). MHV is calculated by accumulating human postures over time in cylindrical coordinates. A different strategy is proposed by Yan *et al.* [24], where they develop a 4D action feature model (4D-AFM) for arbitrary view action recognition based on spatio-temporal volumes (STVs). However, the performance of the above 3D exemplars-based systems is strictly limited to the result of 3D reconstruction. Normally, the reconstructed 3D exemplars are not very realistic.

Some other researchers construct the 3D exemplar with the aid of depth sensors. Zhang *et al.* [36] present a low-cost descriptor called 3D histogram of textures (3DHoTs) to extract discriminative features from a sequence of depth maps. They combine depth maps and texture description by projecting depth frames onto three orthogonal Cartesian planes to describe the salient information of a specific action. Liu *et al.* [37] presents a multi-scale energy-based Global Ternary Image (GTI) representation, which efficiently encodes both the spatial and temporal information of 3D actions. Skeleton information can be easily collected from the depth map. Liu *et al.* [38] propose a sequence-based transform method, which maps skeleton joints into a view-invariant high dimensional space. Then, they use color images to visualize this space and adopt CNN to extract deep features from these enhanced color images. Wang *et al.* [39] realize non-rigid reconstruction and motion tracking without any template using a single RGB-D camera. Jia and Fu [40] present a tensor subspace, whose dimension is learned automatically by low-rank learning for RGB-D action recognition. Kong and Fu [41] propose a discriminative relational feature learning method for fusing heterogeneous RGB with depth modalities and classifying the actions in RGB-D sequences. Even though the depth information has a superior descriptive ability on 3D exemplars, most videos in the real world are captured without depth information. Therefore, we focus on techniques for extracting 3D information from RGB only videos, which have more prospective applied areas.

### *B. Interest Points and Trajectory-Based Methods*

To better describe the spatio-temporal interest points, Dollar *et al.* [2] build the descriptors upon brightness, optical flow and gradient information. The SIFT descriptor is extended to the spatio-temporal interest points by Scovanner *et al.* [42]. Willems *et al.* [43] extend the SURF descriptor to the video domain by computing weighted sums of response of Haar wavelets.

Due to the fact that spatio-temporal interest points are at fixed location in the video, only interest points based descriptor cannot capture motion information in the video. In contrast, trajectory tracks the given interest point over time so that it can capture the motion information. Messing *et al.* [44] extract trajectories by tracking Harris3D interest points with a KLT tracker. They use a sequence of log-polar quantized velocities to represent trajectories. Matikainen *et al.* [45] extract trajectories with a standard KLT tracker, then they cluster these trajectories for the action classfication. Sun *et al.* [46] match SIFT descriptor between two frames to compute trajectories. Later, they combine both SIFT matching and KLT tracker to extract long-duration trajectories [47]. Wang *et al.* [9] compute trajectories by tracking the interest points in the optical

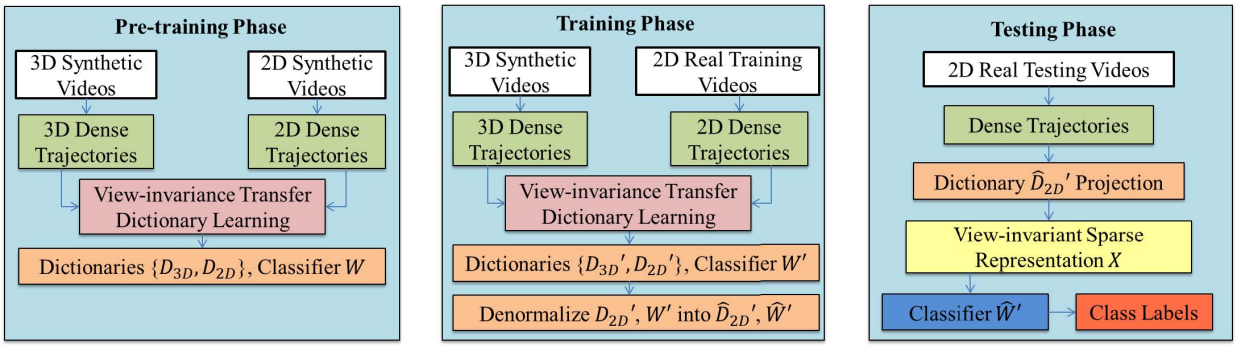



Fig. 2. The overview of our view-invariant transfer dictionary learning system. (Left) In the pre-training phase, we learn the dictionaries $D_{3D}$, $D_{2D}$ and a linear classifier $W$ simultaneously from the synthetic 3D videos and the synthetic 2D videos. (Middle) In the training phase, we replace the synthetic 2D videos with 2D real training videos for adapting the dictionaries $D_{3D}'$, $D_{2D}'$ and the classifier $W'$. The 2D dictionary and the classifier are denormalized into $\widehat{D}_{2D}'$ and $\widehat{W}'$ respectively. (Right) In the testing phase, given any real 2D video, we apply $\widehat{D}_{2D}'$ to encode the features into a view-invariant sparse representation $X$, and use $\widehat{W}'$ for classification.

flow field, then they compute Histogram of Gradient (HOG), Histogram of Optical Flow (HOF) and Motion Boundary Histogram (MBH) to model the action in the video. However, the optical flow field is just a 2D approximation of the 3D motion field and cannot accurately describe the 3D motion information.

### C. Transfer Learning and Dictionary Learning

Transfer learning has been widely used in cross-domain action recognition problems to store knowledge learnt from one dataset and apply it to a different but related one. Liu *et al.* [48] present a simple-to-complex action transfer learning model (SCA-TLM) for complex human action recognition. It improves the performance of complex action recognition by leveraging the abundant labeled simple actions. In particular, it optimizes the weight parameters, enabling the complex actions to be learned and to be reconstructed by simple actions. Xu *et al.* [49] propose a novel dual many-to-one encoder architecture to extract generalized features by mapping raw features from source and target datasets to the same feature space. Rahmani *et al.* [25] propose R-NKTM to transfer knowledge of human actions from any unknown view to a shared high-level virtual view by finding a non-linear virtual path that connects the views.

Recently, dictionary learning for sparse representation has been successfully applied in many computer vision applications, such as image de-noising [50] and face recognition [51]. With an over-complete dictionary, input signal can be approximately represented by a sparse linear combination of items in the dictionary. Previously, many methods [52] have been presented to learn such a dictionary based on different criteria. Among them, the K-Singular Value Decomposition (K-SVD) [53] is a typical dictionary learning method that uses the K-means clustering algorithm for optimizing dictionary items to learn an over-complete dictionary. Even though the K-SVD method has the re-constructive ability, due to the unsupervised learning process, the discriminative ability has not been considered. Later, [54] proposed a dictionary transformation method to transform the dictionary from one domain to another. It can handle the problem that the testing instances are different from the training instances. In addition, they use correspondences between the source view and the target view to construct pairwise dictionaries for the cross-view action recognition problem. Zheng *et al.* represents the videos in each view using a view-specific dictionary and the common dictionary. More importantly, it encourages the set of videos taken from different views of the same action to have the similar sparse representations [55].

Unlike the above approaches, our approach simultaneously learns pairwise dictionaries and a classifier while considering re-constructive ability, discriminative ability and domain adaptability during the dictionary learning process. The data in 3D source domain and 2D target domain are with completely different formats. View-invariance from 3D data can be smoothly transferred to 2D data with jointly optimizing the pairwise dictionaries.

## III. System Overview

As illustrated in Fig. 2 Left, in the pre-training phase, we synthesize 3D video sequences using motion capture data. We propose a new 3D dense trajectories feature extracted from a source 3D synthetic video, and $Y_{3D} = [\boldsymbol{y_{3D}}^1, \ldots, \boldsymbol{y_{3D}}^K] \in R^{S \times K}$ denotes the $K$ $S$-dimensional features. The synthetic 3D video is projected into different viewpoints to create multiple synthetic 2D videos. $Y_{2D} = [\boldsymbol{y_{2D}}^1, \ldots, \boldsymbol{y_{2D}}^K] \in R^{T \times K}$ denotes the $K$ $T$-dimensional features extracted from a target synthetic 2D video. We build 3D videos and 2D videos pairwisely in order to train the correspondence between them. We use $K$ to denote both the numbers of 2D videos and 3D videos used in the pre-training phase.

We then train the 3D and 2D dictionaries simultaneously from the synthetic 3D and 2D videos respectively, which projects the respective video data into a common view-invariant sparse feature space. They are represented as $D_{3D} = [\boldsymbol{d_{3D}}^1, \ldots, \boldsymbol{d_{3D}}^N] \in R^{S \times N}$ and $D_{2D} = [\boldsymbol{d_{2D}}^1, \ldots, \boldsymbol{d_{2D}}^N] \in R^{T \times N}$, where $N$ is the dimension of the sparse feature space. Records belonging to the same action class in both 3D and 2D data are constrained to share the same sparse representation. We construct the action classifier $W$ in an end-to-end manner for better accuracy, by jointly minimizing the classification

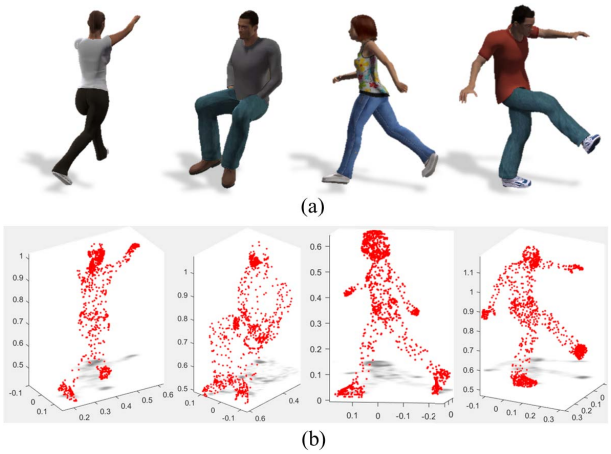



Fig. 3. (a) Some example frames from the synthetic 3D video. Using motion retargeting techniques, we can retarget the captured motion to 3D models of different body sizes to increase the database diversity. (b) The interest points obtained according to the vertices of the 3D models.

error rate and the dictionary quantization error. This improves training efficiency and system accuracy.

Then, as illustrated in Fig. 2 Middle, in the training phase, we replace the synthetic 2D videos with the 2D real training videos and perform system fine-tuning. This allows us to adapt the dictionaries ($D'_{3D}$, $D'_{2D}$) and the classifier ($W'$) originally trained from synthetic data into real-world data. Because of the pre-training phase, only a small amount of real training videos are needed. We finally denormalize the 2D dictionary and the classifier into $\widehat{D}'_{2D}$ and $\widehat{W}'$ respectively.

In the testing phase illustrated in Fig. 2 Right, given any real 2D video, we apply $\widehat{D}'_{2D}$ to encode the features into a view-invariant sparse representation $X = [\boldsymbol{x}^1, \ldots, \boldsymbol{x}^K] \in R^{N \times K}$. We then apply $\widehat{W}'$ to identify the class label of the video. Due to the use of the view transfer dictionary, our system can identify actions from an arbitrary 2D view.

## IV. Video Synthesis and Feature Extraction

In this section, we explain how we synthesize 3D videos and project them to generate synthetic 2D videos. We then explain how we extract a corresponding set of 3D and 2D features.

### A. Synthesizing 3D and 2D Videos

Here, we explain the process of synthesizing 3D and 2D video data.

To synthesize the 3D motion models, we utilize the motion capture data from the Carnegie-Mellon Graphics Lab [56] and the Truebones dataset [57]. The motions are represented with 3D joint angles in a skeletal body hierarchy at 25 frames per second (FPS). Instead of using simplified cylindrical model to represent surface information as in past research [16], [28], we use different high-resolution 3D human models instead. This requires a process known as *primary deformation* [26] to deform the human models based on the skeletal movement over time. The advantage of using 3D motion data is that we can apply *motion retargeting* techniques to synthesize the motion performed by human models of different body sizes, as shown in Fig. 3a. Such an automatic process enhances the diversity of the database by adjusting the movement according to the bone length.

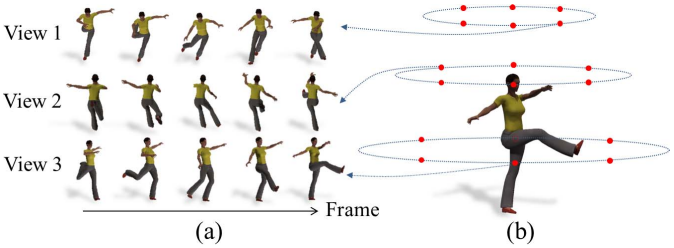



Fig. 4. (a) Example frames of synthetic 2D videos obtained by projecting a 3D video into different viewpoints. (b) Virtual cameras are placed on the hemisphere looking towards the center of the sphere to generate different viewpoints.

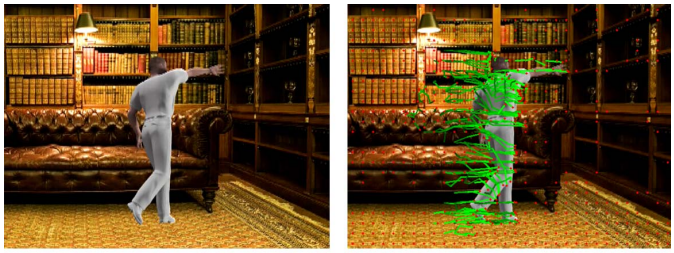



Fig. 5. (a) Synthesized 2D video (b) Extracted dense trajectories (red points are interest points, green curves are trajectories).

In order to produce synthetic 2D video, we project the synthesized 3D videos uniformly in a set of pre-defined viewpoints. Fig. 4 shows example frames of 2D videos projected from various viewpoints. Notice that in our system, we do not require any information about the viewpoints to perform classification.

### B. 2D Dense Trajectories

For both 2D synthetic videos and 2D real videos, we employ dense trajectories [10], a powerful action representation, for feature extraction. It considers both holistic and local information of 2D motion by combining dense sampling and trajectory tracking. Specifically, it consists of a set of low-level descriptors, including trajectory descriptor, Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF) and Motion Boundary Histogram (MBH). Among them, HOG can extract the static appearance of the videos while HOF and MBH can extract the motion information. Fig. 5 shows an example of dense trajectories extracted from a synthetic 2D video.

### C. Proposed 3D Dense Trajectories

Our transfer learning involves transferring 3D and 2D features into a common sparse feature space, and hence it is preferable that both of them have similar logical meanings. Therefore, we propose a 3D version of dense trajectories that corresponds to the 2D one. The proposed feature consists

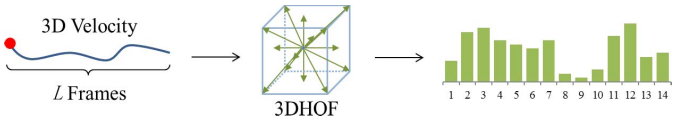



Fig. 6. The 14 3D velocity bins visualized with a 3D cube. 6 directions point towards the faces of the cube, and 8 directions point towards the corners of the cube.


Fig. 7. The 3DMBH components in X, Y and Z directions are quantized into 8 bins each. The 3DMBH is defined as the concatenation of 3DMBHx, 3DMBHy and 3DMBHz along each vertex trajectory.

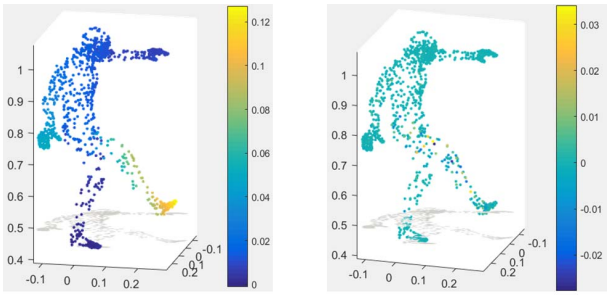



Fig. 8. (a) The Y component of 3D velocity field for the example frame. (b) 3DMBHy is obtained by computing the gradient of Y component of 3D velocity field.

of three components: 3D trajectories, 3DHOF and 3DMBH. Notice that HOG is not included here, as the surface texture of a 3D model remains unchanged over time.

An advantage of synthetic 3D videos is that both the vertices geometry on the human model surface and the vertices correspondence across frames are available. We first obtain a set of interest points over time according to the surface vertices of the 3D models, as shown in Fig. 3b. For each point, we extract the motion trajectory across frames $(P_t, P_{t+1}, P_{t+2}, \ldots)$, where $P_t$ is the 3D Cartesian coordinate of the vertex at frame $t$, as shown in Fig. 1b.

The 3D trajectory is defined as:

$$Tr' = \frac{(\Delta P_t, \ldots, \Delta P_{t+L-1})}{\sum_{j=t}^{t+L-1} \|\Delta P_j\|} \tag{1}$$

where $L$ is a user-defined value that represents the number of frames to be considered in a trajectory, and $\Delta P_t = (P_{t+1} - P_t)$ indicates the displacement across two frames. The denominator is the total length of the trajectory, which is used for normalization.

2D HOF is the pattern of apparent motion of objects and surface in a visual scene caused by the relative motion between an observer and a scene. A logically similar representation in 3D, which we named the 3D Histogram of Optical Flow (3DHOF), is the velocity field of the surface vertices. We first define the velocity of a vertex as:

$$V_t = \frac{\Delta P_t}{1/FPS} \tag{2}$$

where $FPS$ is the frame rate of the 3D video, and is set to 25 in our experiments. We then quantize the 3D velocity orientations into 14 bins $H(h_1, h_2, \ldots, h_{14})$ as shown in Fig. 6. 3DHOF is defined as the binned histogram along each vertex trajectory:

$$h_i = \frac{\sum_{t \in T_i} \|V_t\|}{\sum_{j=t}^{t+L-1} \|V_t\|} \tag{3}$$

where $T_i$ is a set that contains the frame's number in which the velocity direction of the interest point belongs to i on a $L$-frame trajectory. $\|V_t\|$ is the magnitude of the velocity, which is used for weighting.

The 2D MBH (motion boundary histogram) is the derivative of the optical flow field computed separately for the horizontal and vertical components to encode the relative motion between pixels. This is to compensate the HOF descriptor, which can only compute absolute motion information. Inspired by this, we proposed the 3DMBH that encodes the relative motion between neighbor interest points on our 3D model. Similar to the 2D MBH implementation, we compute the derivatives separately along the X, Y, Z axes in the 3D velocity field. We quantize each 3DMBH component into 8 bins and the 3DMBH is defined as the concatenation of 3DMBHx, 3DMBHy and 3DMBHz along each vertex trajectory. The process is visualized as in Fig. 7. For example, the Y component of 3D velocity field is shown in Fig. 8a, and we compute its gradient to describe the relative motion between neighboring interest points of that frame as shown in Fig. 8b.

## V. View-Invariant Action Classification

In this section, we explain how we train the view-invariant dictionaries and the classifier from synthetic 3D and 2D video data using dictionary learning. The processes are summarized as the algorithm shown in Fig. 9.

### A. The Pre-Training Phase

Here, we introduce the basic theory of dictionary learning [58], and explain how we learn the view-invariant transfer dictionary for the 3D and 2D synthetic videos.

Dictionary learning generates a sparse representation for a high dimensional signal using linear projection with a projection dictionary. Let $\boldsymbol{y} \in R^P$ denote a $P$-dimensional input signal that can be reconstructed by the $Q$-dimensional projection coefficient $\boldsymbol{x} \in R^Q$ via a linear projection dictionary $D = [\boldsymbol{d}^1, \ldots, \boldsymbol{d}^Q] \in R^{P \times Q}$. To obtain an over-completed dictionary, $P$ should be much larger than $Q$. Assuming the reconstruction error to be $E(\boldsymbol{x})$, the projection process is formulated as:

$$\boldsymbol{y} = D\boldsymbol{x} + E(\boldsymbol{x}) \tag{4}$$

The objective function is defined as:

$$argmin_{\boldsymbol{x},D} \|\boldsymbol{y} - D\boldsymbol{x}\|_2^2 \quad s.t. \ \|\boldsymbol{x}\|_0 \leq M \tag{5}$$

**Input:** 3D feature matrix $Y_{3D}$, 2D feature matrix $Y_{2D}$, target domain class label $H$, discriminative sparse code $Q$, sparsity constraint $M$, dictionary size $N$, trade-off parameter $\alpha, \beta, \gamma$, iteration steps $I$.

**Output:** Dictionary $\widehat{D}_{3D}{}'$, $\widehat{D}_{2D}{}'$, linear classifier parameter $\widehat{W}'$.

**Pre-training:** (Synthetic 3D Video, Synthetic 2D Video)

Initialize $D_{3D}$, $D_{2D}$, $A$ and $W$ using K-SVD and Eq. 11, 12;

Reformulate $Y_0 = \begin{pmatrix} \sqrt{\alpha}Y_{3D} \\ Y_{2D} \\ \sqrt{\beta}Q \\ \sqrt{\gamma}H \end{pmatrix}$, $D_0 = \begin{pmatrix} \sqrt{\alpha}D_{3D} \\ D_{2D} \\ \sqrt{\beta}A \\ \sqrt{\gamma}W \end{pmatrix}$;

**For** $i = 1$ **to** $I$

// Sparse coding using OMP

$argmin_{X,D_0} \|Y_0 - D_0 X\|_2^2 \; s.t. \forall i, [\|\boldsymbol{x}^i\|_0] \le M$;

// Dictionary updating using SVD

**For** $k = 1$ **to** $N$

$argmin_{\boldsymbol{d_k}, \tilde{\boldsymbol{x}}_k} \|\tilde{E}_k - \boldsymbol{d_k}\tilde{\boldsymbol{x}}_k\|_2^2$

$SVD(\tilde{E}_k) = U \sum V^T$

$\boldsymbol{d_k} = U(:,1)$

$\tilde{\boldsymbol{x}}_k = \sum(1,1)V(1,:)$;

**End**

Update $D_0$

**End**

**Training:** (Synthetic 3D Video, Real 2D Video)

Initialize $D_{3D}{}'$, $D_{2D}{}'$, $A'$ and $W'$ with pre-trained $D_{3D}$, $D_{2D}$, $A$ and $W$;

Apply the same optimization strategy as the pre-training phase;

Denormalize the trained $D_{2D}{}'$ and $W'$ to obtain $\widehat{D}_{2D}{}'$ and $\widehat{W}'$;

**Testing:** (Real 2D Video)

Use Eq. 14 for classification.

Fig. 9. The algorithm for transferring view-invariance from 3D video to 2D video by transfer dictionary learning.

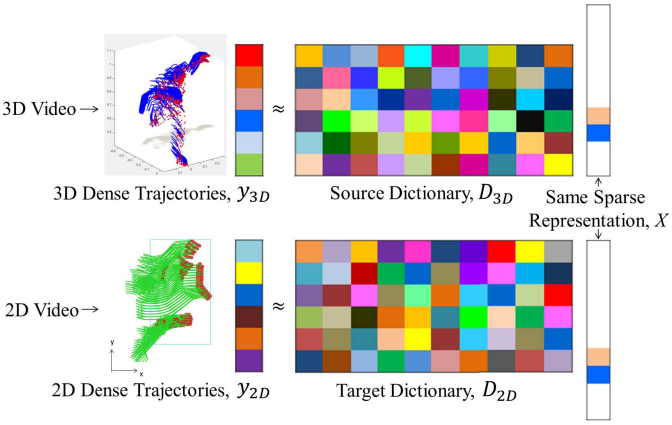



Fig. 10. Optimizing the 3D (source) and 2D (target) dictionaries to constraint that the same action in synthetic 3D and 2D videos has the same sparse representations.

where $\|\boldsymbol{y} - D\boldsymbol{x}\|_2^2$ denotes the reconstruction error. $s.t. \|\boldsymbol{x}\|_0 \le M$ denotes the sparsity constraint. $M$ is the $L_0$-norm sparsity constraint factor that limits the number of non-zero elements in the sparse codes.

Due to the different number of trajectories across action videos, we use a bag-of-words descriptor to ensure that the features extracted from the action videos share the same dimension, following [8]–[11]. Specifically, we use K-means to cluster the trajectory-based descriptors in each action video into a fixed number of visual words. This allows us to represent the action videos with histograms of the same dimension.

We design a transfer dictionary learning system to transfer the view-invariance of the synthetic 3D videos to the synthetic 2D videos. We train two dictionaries simultaneously, with one for 3D (i.e. source - $D_{3D}$) and one for 2D (i.e. target - $D_{2D}$). The main idea is to optimize the dictionaries such that the same action in both 3D and 2D videos has the same sparse representations, as visualized in Fig. 10. Upon successful training, $D_{2D}$ is able to project the feature vector of a 2D video into a sparse representation that is similar to that of a 3D video. In other words, such a sparse representation is view-invariant.

We divide the dictionary into a number of disjoint subsets, and each of these is used exclusively for one action category. 3D and 2D videos with the same action category are therefore represented by the same subset of the dictionary. Those with different action categories are represented with disjoint subsets of the dictionary. This design enables the 3D and 2D videos with the same action category to share the same sparse representation pattern. Conversely, those with different action categories tend to have different representations.

Specifically, the dictionary optimization function is designed as:

$$\begin{aligned} &argmin_{X,D_{3D},D_{2D},A} \; \alpha\|Y_{3D} - D_{3D}X\|_2^2 + \|Y_{2D} - D_{2D}X\|_2^2 \\ &\qquad + \beta\|Q - AX\|_2^2 \\ &s.t \; \forall i, \;\; \|\boldsymbol{x}^i\|_0 \le M \end{aligned} \tag{6}$$

where $\alpha$ and $\beta$ are trade-off parameters, $\|Y_{3D} - D_{3D}X\|_2^2$ and $\|Y_{2D} - D_{2D}X\|_2^2$ are two terms to minimize the error of the 3D and 2D dictionaries respectively, and $\|Q - AX\|_2^2$ is a label consistent regularization term to minimize the difference in sparse representation for the same class of action as introduced in [59] and [60]. $A$ is a linear transformation matrix that maps the original sparse codes $X$ to be consistent with the discriminative sparse codes $Q = [\boldsymbol{q}^1, \ldots, \boldsymbol{q}^K] \in R^{N \times K}$ of input signal $(\boldsymbol{y_{3D}}^j, \boldsymbol{y_{2D}}^j)$, in which the index $j$ indicates the index of 2D and 3D action video pairs. Specifically, each vector $\boldsymbol{q_j} = [\boldsymbol{q_j}^1, \ldots, \boldsymbol{q_j}^N] = [0 \ldots 1, 1 \ldots 0] \in R^N$, and the non-zero occurs at those indices where the input signal $(\boldsymbol{y_{3D}}^j, \boldsymbol{y_{2D}}^j)$ and the dictionary items $(\boldsymbol{d_{3D}}^n, \boldsymbol{d_{2D}}^n)$ share the same label. In our dictionary design, dictionary item $\boldsymbol{d_{3D}}^n$ and $\boldsymbol{d_{3D}}^n$ always have the same label. For example, assuming the $Y_{2D} = [\boldsymbol{y_{2D}}^1, \ldots, \boldsymbol{y_{2D}}^6]$ and $D_{2D} = [\boldsymbol{d_{2D}}^1, \ldots, \boldsymbol{d_{2D}}^6]$, where $\boldsymbol{y_{2D}}^1$, $\boldsymbol{y_{2D}}^2$ and $\boldsymbol{d_{2D}}^1$, $\boldsymbol{d_{2D}}^2$ are from class 1, $\boldsymbol{y_{2D}}^3$, $\boldsymbol{y_{2D}}^4$ and $\boldsymbol{d_{2D}}^3$, $\boldsymbol{d_{2D}}^4$ are from class 2, $\boldsymbol{y_{2D}}^5$, $\boldsymbol{y_{2D}}^6$ and $\boldsymbol{d_{2D}}^5$, $\boldsymbol{d_{2D}}^6$ are from class 3, then $Q$ can be defined as:

$$Q = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{7}$$

Inspired by [60], we propose to include the action classification error of a linear prediction classifier into the object function to build an end-to-end system. This enhances the system training efficiency and results in better classification

accuracy. The new objective function is therefore updated as

$$\begin{aligned} &argmin_{X,D_{3D},D_{2D},A,W}\ \alpha\|Y_{3D} - D_{3D}X\|_2^2 + \|Y_{2D} \\ &\qquad - D_{2D}X\|_2^2 + \beta\|Q - AX\|_2^2 + \gamma\|H - WX\|_2^2 \\ &s.t.\ \forall i,\ \|\boldsymbol{x}^i\|_0 \leq M \end{aligned} \tag{8}$$

where $\|H - WX\|_2^2$ is the proposed action classification error term, $W \in R^{C\times N}$ denotes the classifier parameters and $H = [\boldsymbol{h}^1, \ldots, \boldsymbol{h}^K] \in R^{C\times K}$ are the class label of input signals $Y_{2D}$. $\boldsymbol{h}^j = [0 \ldots 1 \ldots 0]^T \in R^C$ is a label vector corresponding to an input signal $\boldsymbol{y_{2D}}^j$, where the nonzero position indicates the class of $\boldsymbol{y_{2D}}^j$.

### B. Optimization

Here, we explain how we obtain the solution for Eq. 8. Since the three terms on the right hand side of Eq. 8 have the same format, we first rewrite Eq. 8 as follows:

$$argmin_{X,D_0}\|Y_0 - D_0X\|_2^2 \quad s.t.\ \forall i,\ \|\boldsymbol{x}^i\|_0 \leq M \tag{9}$$

where

$$Y_0 = \begin{pmatrix} \sqrt{\alpha}Y_{3D} \\ Y_{2D} \\ \sqrt{\beta}Q \\ \sqrt{\gamma}H \end{pmatrix}, D_0 = \begin{pmatrix} \sqrt{\alpha}D_{3D} \\ D_{2D} \\ \sqrt{\beta}A \\ \sqrt{\gamma}W \end{pmatrix}.$$

Such an objective function shares the same form as Eq. 5, which can be optimized using the K-SVD algorithm [53]. Specifically, Eq. 9 is solved through both dictionary atom updating and sparse representing.

For the dictionary atom updating stage, each dictionary atom is updated sequentially to better represent both 3D videos and 2D videos. When pursing the better dictionary $D_0$, the sparse representation $X$ is fixed, and each dictionary atom is updated by tracking down a rank-one approximation to the matrix of residuals.

Following K-SVD, the $k^{th}$ atom of dictionary $D_0$ and its corresponding coefficients are denoted as $\boldsymbol{d_k}$ and $\boldsymbol{x_k}$ respectively. Let $E_k = Y_0 - \sum_{j\neq k}\boldsymbol{d_j}\boldsymbol{x_j}$ and we further denote $\widetilde{\boldsymbol{x}}_{\boldsymbol{k}}$ and $\widetilde{E}$ as the result obtained when all zero entries in $\boldsymbol{x_k}$ and $E_k$ are discarded respectively. Each dictionary atom $\boldsymbol{d_k}$ and its corresponding non-zero coefficients $\widetilde{\boldsymbol{x}}_{\boldsymbol{k}}$ can be computed by:

$$argmin_{\boldsymbol{d_k},\widetilde{\boldsymbol{x}}_{\boldsymbol{k}}}\|\widetilde{E}_k - \boldsymbol{d_k}\widetilde{\boldsymbol{x}}_{\boldsymbol{k}}\|_2^2 \tag{10}$$

The approximation in Eq. 10 is achieved through Singular Value Decomposition (SVD) on $\widetilde{E}_k$:

$$\begin{aligned} SVD(\widetilde{E}_k) &= U\sum V^T \\ \boldsymbol{d_k} &= U(:,1) \\ \widetilde{\boldsymbol{x}}_{\boldsymbol{k}} &= \sum(1,1)V(1,:) \end{aligned} \tag{11}$$

where $U(:,1)$ indicates the first column of $U$ while $V(1,:)$ indicates the first row of $V$.

At the sparse representation stage, we compute the best matching projection $X$ of the multidimensional training data for the updated dictionary $D_0$ using Orthogonal Matching Pursuit (OMP) algorithm.

*1) Initialization:* $D_{3D}$, $D_{2D}$, $A$ and $W$ are required to be initialized before pre-training. In our system, for $D_{3D}$ and $D_{2D}$, we run a few iterations of K-SVD within each action class and initialize the label of the dictionary items based on the corresponding action labels. To initialize $A$ and $W$, we use the multivariate ridge regression model [61] with the $L_2$-norm:

$$\begin{aligned} A &= argmin_A\|Q - AX\|_2^2 + \varphi_1\|A\|_2^2 \\ W &= argmin_W\|H - WX\|_2^2 + \varphi_2\|W\|_2^2 \end{aligned} \tag{12}$$

where $\varphi_1$ and $\varphi_2$ are manually defined constants and are empirically set as 0.5 in our system. The equation yields the following solutions:

$$\begin{aligned} A &= (XX^t + \varphi_1 I)^{-1} \\ W &= (XX^t + \varphi_2 I)^{-1} \end{aligned} \tag{13}$$

where $X$ is calculated with the initialized $D_{3D}$ or $D_{2D}$.

*2) Convergence Analysis:* The convergence proof of the proposed method is similar with the K-SVD algorithm. In the dictionary updating stage, each atom $d_k$ and its corresponding coefficients $\tilde{x}_k$ minimize the objective function, while the rest of dictionary atoms are updated iteration by iteration. Therefore, the Mean Squared Error (MSE) of the reconstruction error should be monotonically decreasing. At the sparse representation stage, the MSE is also reduced due to the computation of the best matched coefficients under the $L_0$-norm constraint of the OMP algorithm. In addition, since MSE is non-negative, the optimization process should be monotonically reducing and bounded by zero. Therefore, the convergence of the proposed transfer dictionary learning method is guaranteed.

### C. The Training Phase

Here, we explain how to adapt the pre-trained dictionaries and classifier into real video.

We fine-tune the dictionaries and the classifier pre-trained by the synthetic data in order to adapt them into real-world data. Specifically, we use $D_{3D}$, $D_{2D}$, $A$, $W$ in the pre-training phase to initialize the training phase. We also replace the 2D synthetic videos with 2D real training videos. Then, we follow the same optimization strategy in Section V-B and apply the same number of iterations as the pre-training phase. After the optimization, we denote the trained dictionaries and classifiers as ($D_{3D}{}'$, $D_{2D}{}'$) and $W'$ respectively.

Since $D_{3D}{}'$, $D_{2D}{}'$, $W'$ are jointly $L2$-normalized during the optimization process, we need a step of de-normalization before they can be used for classification. Following [60], the denormalized 2D dictionary $\widehat{D}'_{2D}$ and the classification parameter $\widehat{W}'$ are calculated as:

$$\begin{aligned} \widehat{D}'_{2D} &= \left(\frac{\boldsymbol{d_{2D\_1}}'}{\|\boldsymbol{d_{2D\_1}}'\|_2}, \frac{\boldsymbol{d_{2D\_2}}'}{\|\boldsymbol{d_{2D\_2}}'\|_2}, \ldots, \frac{\boldsymbol{d_{2D\_N}}'}{\|\boldsymbol{d_{2D\_N}}'\|_2}\right) \\ \widehat{W}' &= \left(\frac{\boldsymbol{w_1}'}{\|\boldsymbol{w_1}'\|_2}, \frac{\boldsymbol{w_2}'}{\|\boldsymbol{w_2}'\|_2}, \ldots, \frac{\boldsymbol{w_N}'}{\|\boldsymbol{w_N}'\|_2}\right) \end{aligned} \tag{14}$$

where $\boldsymbol{d_{2D\_n}}'$ denotes the $n^{th}$ atom of the dictionary $D'_{2D}$, $\boldsymbol{w_N}'$ denotes the $n^{th}$ atom of $W'$. Notice that we do not denormalize $D_{3D}{}'$ as it is no longer needed in the next phase.

TABLE I

CROSS-VIEW RECOGNITION ACCURACY OF ALL POSSIBLE VIEWPOINT COMBINATIONS ON IXMAS DATABASE. THE HORIZONTAL AXIS LABELS ARE FORMATTED AS "SOURCE VIEW|TARGET VIEW"

| Methods | 0\|1 | 0\|2 | 0\|3 | 0\|4 | 1\|0 | 1\|2 | 1\|3 | 1\|4 | 2\|0 | 2\|1 | 2\|3 | 2\|4 | 3\|0 | 3\|1 | 3\|2 | 3\|4 | 4\|0 | 4\|1 | 4\|2 | 4\|3 | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DVV [62] | 72.4 | 13.3 | 53.0 | 28.8 | 64.9 | 27.9 | 53.6 | 21.8 | 36.4 | 40.6 | 41.8 | 37.3 | 58.2 | 58.5 | 24.2 | 22.4 | 30.6 | 24.9 | 27.9 | 24.6 | 56.4 |
| CVP [63] | 78.5 | 19.5 | 60.4 | 33.4 | 67.9 | 29.8 | 55.5 | 27.0 | 41.0 | 44.9 | 47.0 | 41.0 | 64.3 | 62.2 | 24.3 | 26.1 | 34.9 | 28.2 | 29.8 | 27.6 | 38.2 |
| nCTE [16] | 94.8 | 69.1 | 83.9 | 39.1 | 90.6 | 79.7 | 79.1 | 30.6 | 72.1 | 86.1 | 77.3 | 62.7 | 82.4 | 79.7 | 70.9 | 37.9 | **48.8** | 40.9 | 70.3 | 49.4 | 67.3 |
| Hankelets [64] | 83.7 | 59.2 | 57.4 | 33.6 | 84.3 | 61.6 | 62.8 | 26.9 | 62.5 | 65.2 | 72.0 | 60.1 | 57.1 | 61.5 | 71.0 | 31.2 | 39.6 | 32.8 | 68.1 | 37.4 | 56.4 |
| Zhang et al. [28] | 91.7 | 70.2 | 84.7 | **44.4** | **92.3** | **81.4** | 84.1 | 45.4 | 66.5 | **87.3** | 75.5 | 58.7 | 84.3 | 80.9 | 66.7 | 45.8 | 32.4 | 48.9 | **74.8** | **53.3** | 68.5 |
| Without pre-training | 86.1 | 68.8 | 74.7 | 34.6 | 81.4 | 74.6 | 78.4 | 37.9 | 68.4 | 78.6 | 73.5 | 58.3 | 76.5 | 72.3 | 64.3 | 48.7 | 31.7 | 37.1 | 67.8 | 41.1 | 62.7 |
| Ours | **96.2** | **71.3** | **85.2** | 41.5 | 90.6 | 80.7 | **89.7** | **47.5** | **74.2** | 85.3 | **82.1** | **60.5** | **85.1** | **84.9** | **73.5** | **57.6** | 41.6 | **52.8** | 71.6 | 50.8 | **71.1** |

### *D. The Testing Phase*

Here, we explain how we apply our trained dictionary to perform view-invariant action classification.

Given a real 2D video query sample $\boldsymbol{y_{2D}}'$, its sparse representation $\boldsymbol{x}'$ can be computed with $\widehat{D}_{2D}'$. With the linear classification parameter $\widehat{W}'$, the label $\boldsymbol{l}$ can be predicted as:

$$\boldsymbol{l} = \widehat{W}'\boldsymbol{x}' \tag{15}$$

The label of $\boldsymbol{y_{2D}}'$ is the index corresponding to the largest element of $\boldsymbol{l}$.

## VI. EXPERIMENTAL RESULTS

In this section, we first provide experiment setup details. We then evaluate the performance of our method with four public multi-view datasets including the IXMAS, N-UCLA, UWA3DII and i3DPost datasets.

The synthetic 3D and 2D datasets we used for transfer dictionary learning are open to the public. They can be found at our project website. All experiments were performed on a desktop computer with an Intel i7-4790k CPU, a NVIDIA Quadro K2200 graphics card and 16GB RAM.

### *A. Implementation Details*

We used the software package Poser 2014 to retarget 3D motion capture data files in BVH format, animate 3D human models, and project the 3D scenes into 2D videos. We employed 5 high-quality 3D characters to synthesize the 3D video. For each action class, we synthesized 18 3D videos per character with 18 randomly selected motion files within the class. For each action class, we synthesized the 2D videos per character by projecting a randomly selected 3D video into 18 uniformly sampled viewpoints. The azimuthal angle of the projection was uniformly sampled as {0°, 60°, 120°, 180°, 240°, 300°} and the polar angle of the projection was sampled as {0°, −30°, −60°}. This setup allowed us to generate the same number of 3D and 2D videos (number of characters × number of views × number of action classes)as required by K-SVD for transfer dictionary learning.

During pre-training, for the experiments on the IXMAS dataset (11 action classes), the N-UCLA dataset (10 action classes), the i3DPost dataset (10 action classes) and the UWA3DII dataset (30 action classes), we synthesized 990, 900, 900, 2700 pairwise 3D and 2D videos, respectively. From our experience, a larger synthetic dataset resulted in better accuracy. The size used was chosen considering the trade-off between system accuracy and training complexity.

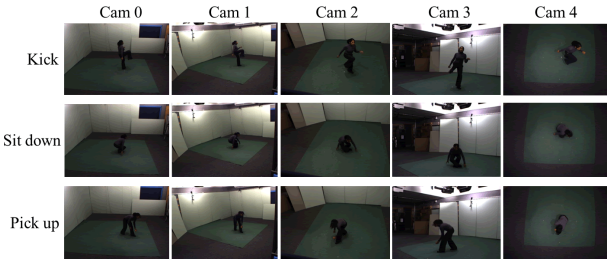



Fig. 11. Sampled frames from the IXMAS dataset.

We extracted dense trajectories from 2D synthetic videos, as well as 2D real videos from the IXMAS, N-UCLA, UWA3DII and i3DPost datasets. Afterwards, we constructed a codebook for each of the four descriptors in the dense trajectories separately. For each 2D descriptor, we applied k-means to cluster a subset of 100,000 dense trajectory features into 375 visual words. This resulted in a 2D feature $Y_{2D}$ of 1,500 dimensions. For 3D synthetic videos, similar to [23], we set the trajectory sample step to 5 frames, and the trajectory length to 15 frames. We constructed codebooks for 3D trajectories, 3DHOF and 3DMBH descriptors respectively. For each 3D descriptor, we applied k-means to cluster a subset of 100,000 3D dense trajectories into 500 visual words. This resulted in a 3D feature $Y_{3D}$ of 1,500 dimensions.

When training the transfer dictionaries, to initialize the dictionary pair $D_{3D}$ and $D_{2D}$, we employed k-means 5 times on the features $Y_{3D}$ and $Y_{2D}$ respectively. For IXMAS, N-UCLA, UWA3DII and i3DPost datasets, we set the dictionary sizes N to 1180, 1150, 2500 and 1150 respectively, for both $D_{3D}$ and $D_{2D}$. The 3D dictionary trade-off parameter $\alpha$ was set to 1.5. The label consistent trade-off parameter $\beta$ was set to be 2.0. The classification error trader-off parameter $\gamma$ was set to be 4.0. Finally, the numbers of iterations for the K-SVD algorithm in both pre-training and training phases were set to 60, 65, 100 and 65 for IXMAS, N-UCLA, UWA3DII and i3DPost datasets respectively.

### *B. Experiments on the IXMAS Dataset*

The IXMAS dataset [33] contains 11 daily-life actions including check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick, and pick up. Each action was performed three times by 10 subjects captured from 5 different viewpoints. Fig. 11 shows some examples.

In order to compare with existing works on cross-view action recognition that utilize view labels including DVV [62], CVP [63], nCTE [16], Hankelets [64], and our preliminary work [28], we conducted an experiment considering view

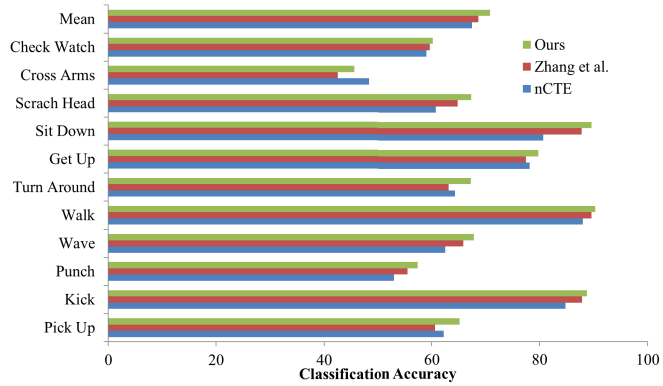



Fig. 12. Cross-view recognition accuracy per action class in IXMAS.

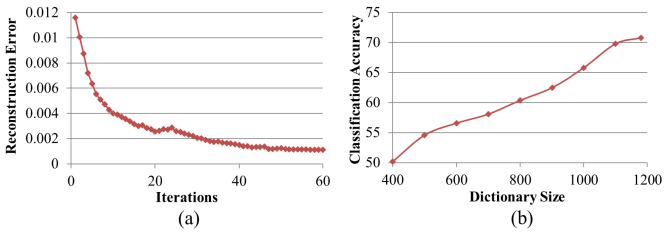



Fig. 13. Parameter analysis on the cross-view action recognition in IXMAS dataset. (a) The optimization process of the objective function with 50 iterations. (b) Performance with varying the dictionary size.

labels. Here, we grouped the videos in the IXMAS dataset into different views and evaluated the accuracy of transferring one view to another. We followed the leave-one-action-out cross-validation strategy from [16], [64]. Table I shows that our algorithm outperforms the state-of-the-art method nCTE in most cross-view pairs, as well as the average system accuracy. It also demonstrates that our proposed methodology enhancements over [28] have resulted in superior accuracy. We also compare with a baseline setup of our system that does not include the pre-training phase, which demonstrates the effectiveness of utilizing synthetic 2D and 3D videos for pre-training. Fig. 12 shows that our algorithm outperforms nCTE in most action classes, thereby indicating that our system can realize cross-view action recognition by transferring the view-invariance from 3D models. Notice that in our default setup, the system does not require any view information. This experiment was designed for the sake of comparison only.

In order to analyze the effect of the hyperparameters (i.e. $\alpha$, $\beta$ and $\gamma$), we experiment with 27 different settings within the searching range of $\alpha$ in [1, 2] on every 0.5 interval, $\beta$ in [1, 2] on every 0.5 interval and $\gamma$ in [2, 4] on every 1.0 interval. The result is visualized in Fig. 14.

Since the orientation of the actors is arbitrary in the IXMAS dataset, we compare with existing works on arbitrary view action recognition by calculating average accuracy for each camera. For example, C0 is the average accuracy when camera0 is used for training or testing. Table II shows that our algorithm outperforms most of the previous methods in some viewpoints. It is worth mentioning that NKTM [17] and R-NKTM [25] are deep learning based methods, at the core of which is the use of neural networks to transfer videos from different views to a canonical view. However, their method

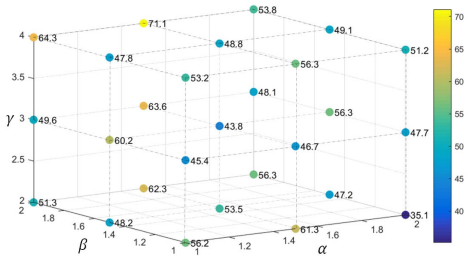



Fig. 14. Analysis on hyperparameters in Equation 7.

TABLE II

AVERAGE ACCURACY ON THE IXMAS DATASET FOR EACH CAMERA, E.G. C0 IS THE AVERAGE ACCURACY WHEN CAMERA 0 IS USED FOR TRAINING OR TESTING. EACH TIME, ONLY ONE CAMERA VIEW IS USED FOR TRAINING AND TESTING

| Methods | C0 | C1 | C2 | C3 | C4 |
|---|---|---|---|---|---|
| DVV [62] | 44.7 | 45.6 | 31.2 | 42.0 | 27.3 |
| CVP [63] | 50.0 | 49.3 | 34.7 | 45.9 | 31.0 |
| nCTE [16] | 72.6 | 72.7 | 73.5 | 70.1 | 47.5 |
| Hankelets [64] | 59.7 | 59.9 | 65.0 | 56.3 | 41.2 |
| NKTM [17] | 77.8 | 75.2 | 80.3 | 74.7 | 54.6 |
| R-NKTM [25] | **78.4** | 78.0 | **80.7** | 75.8 | **57.8** |
| Zhang et al. [28] | 70.8 | 76.5 | 72.6 | 71.9 | 50.5 |
| Ours | 73.2 | **78.5** | 74.9 | **76.1** | 52.9 |


Fig. 15. Sampled frames from the N-UCLA dataset.

requires the generation of 2D training video by projecting the 3D exemplar to 108 virtual views, while ours only needs 18 different views. Due to the lower amount of training data required, our method can save computation resources especially when constructing the system.

### C. Experiments on the N-UCLA Dataset

The N-UCLA dataset [21] contains 10 action classes captured from 3 different viewpoints with 10 different actors. The action categories include pick up with one hand, pick up with two hands, drop trash, walk around, sit down, stand up, donning, doffing, throw, and carry. Fig. 15 shows some sample frames from the N-UCLA dataset.

We evaluated our system accuracy in cross-view action recognition and in comparison with existing work including DVV [62], nCTE [16], CVP [63], and our preliminary work [28]. We followed the experimental setup in [16] and [63], which utilizes videos captured from two cameras

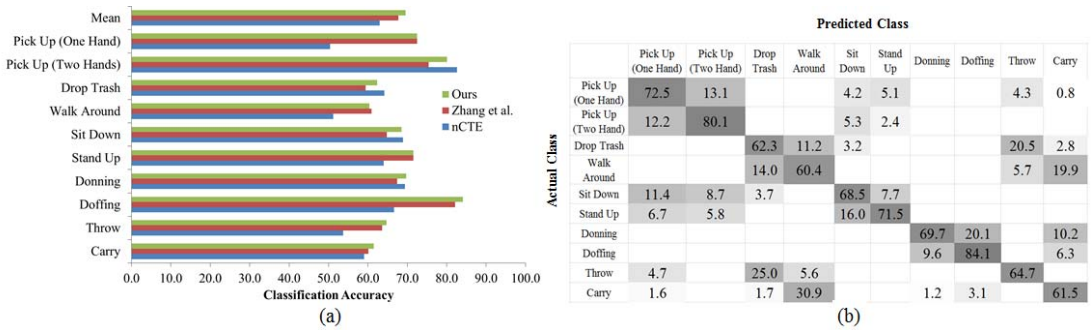



Fig. 16. (a) Cross-view recognition accuracy per action class in N-UCLA. (b) The confusion matrix of N-UCLA.

TABLE III
ACCURACY ON THE N-UCLA DATASET (TWO VIEWS FOR TRAINING AND ONE FOR TESTING)

| Methods | {1,2}\|3 | {1,3}\|2 | {2,3}\|1 | Mean |
|---|---|---|---|---|
| DVV [62] | 58.5 | 55.2 | 39.3 | 51.0 |
| CVP [63] | 60.6 | 55.8 | 39.5 | 52.0 |
| nCTE [16] | 68.6 | 68.3 | 52.1 | 63.0 |
| Zhang et al. [28] | 67.3 | 74.2 | 61.8 | 67.8 |
| Ours | **69.1** | **74.4** | **61.8** | **68.5** |

for training and the other one for testing. The accuracy was calculated using leave-one-action-out cross validation. As shown in Table III, our method outperforms existing algorithms in most of the cross-view setups and the overall result. Fig. 16 shows that our algorithm outperforms nCTE in most action classes. This demonstrates that our system can realize cross-view action recognition by transferring the view-invariance from 3D models. Notice that in our default setup, the system does not require view information. This experiment was designed for the sake of comparison only.

On the N-UCLA dataset, some actions are quite difficult to differentiate, such as "Drop Trash" vs. "Throw", "Carry" vs. "Walk around", as they both consist of similar body movement.

### *D. Experiments on the UWA3DII Dataset*

This dataset [65] consists of a variety of daily-life human actions performed by 10 subjects with different scales. It includes 30 action classes: one hand waving, one hand punching, two hand waving, two hand punching, sitting down, standing up, vibrating, falling down, holding chest, holding head, holding back, walking, irregular walking, lying down, turning around, drinking, phone answering, bending, jumping jack, running, picking up, putting down, kicking, jumping, dancing, moping floor, sneezing, sitting down (chair), squatting, and coughing. Each video is captured from one of four predefined viewpoints. This results in variations in actions across different viewpoints within the same action class. This dataset is challenging because of varying actor orientations, self-occlusion and high similarity among actions. Fig. 17 shows four sample actions from different viewpoints.

As shown in Table IV, our method outperforms existing algorithms in most of the cross-view setups and the overall

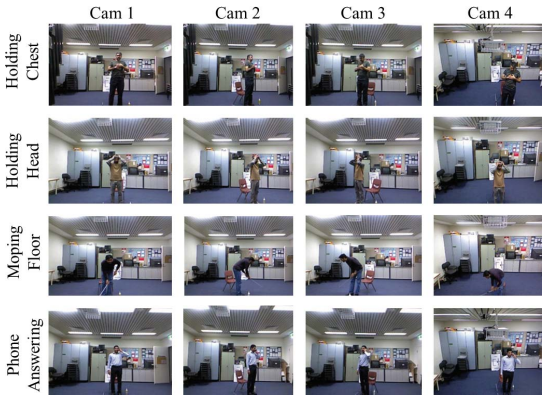



Fig. 17. Sampled frames from the UWA3DII dataset.

result. Fig. 18 shows that our algorithm outperforms our baseline in most action classes.

### *E. Experiments on the i3DPost Dataset*

The i3DPost dataset consists of 8 actors performing 10 different actions, where 6 are single actions: walk, run, jump, bend, hand-wave and jump-in-place, and 4 are combined actions: sit-stand-up, run-fall, walk-sit and run-jump-walk. The subjects have different body sizes, clothing and are of different sex and nationalities. The multi-view videos have been recorded by 8 calibrated and synchronized cameras in a high definition resolution (1920×1080), resulting in a total of 640 videos. For each video frames, an actor 3D mesh model of high detail level (20000-40000 vertices and 40000-80000 triangles) and the associated camera calibration parameters are available. The mesh models were reconstructed using a global optimization method proposed by Gkalelis *et al.* [65]. Fig. 19 shows multi-view actor/action examples from the i3DPost dataset.

We use leave-one-actor out strategy followed by [68]. This means that we use the 2D videos of one actor for testing, while using the rest of the dataset for training. Table V shows that our system achieves better result than previous methods.

### *F. Evaluation of Our 3D Dense Trajectories*

In this section, we evaluate our 3D dense trajectories by using 3D trajectories, 3DHOF and 3DMBH independently.

TABLE IV

ACCURACY ON THE UWA3DII DATASET (TWO VIEWS FOR TRAINING AND ONE FOR TESTING)

| Methods | {1,2}\|3 | {1,2}\|4 | {1,3}\|2 | {1,3}\|4 | {1,4}\|2 | {1,4}\|3 | {2,3}\|1 | {2,3}\|4 | {2,4}\|1 | {2,4}\|3 | {3,4}\|1 | {3,4}\|2 | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AOG [21] | 47.3 | 39.7 | 43.0 | 30.5 | 35.0 | 42.2 | 50.7 | 28.6 | 51.0 | 43.2 | 51.6 | 44.2 | 42.3 |
| Action Tube [66] | 49.1 | 18.2 | 39.6 | 17.8 | 35.1 | 39.0 | 52.0 | 15.2 | 47.2 | 44.6 | 49.1 | 36.9 | 37.0 |
| LRCN [67] | 53.9 | 20.6 | 43.6 | 18.6 | 37.2 | 43.6 | 56.0 | 20.0 | 50.5 | 44.8 | 53.3 | 41.6 | 40.3 |
| Zhang et al. [28] | 50.6 | 56.8 | 48.6 | 43.7 | 53.2 | 59.7 | **66.8** | 48.9 | 56.8 | 50.4 | 68.3 | 51.7 | 54.6 |
| Ours | **59.3** | **57.9** | **50.2** | **48.1** | **59.9** | **63.4** | 65.1 | **67.1** | **68.2** | **55.5** | **73.5** | **53.4** | **60.1** |


Fig. 18. Cross-view recognition accuracy per action class in the UWA3DII dataset.

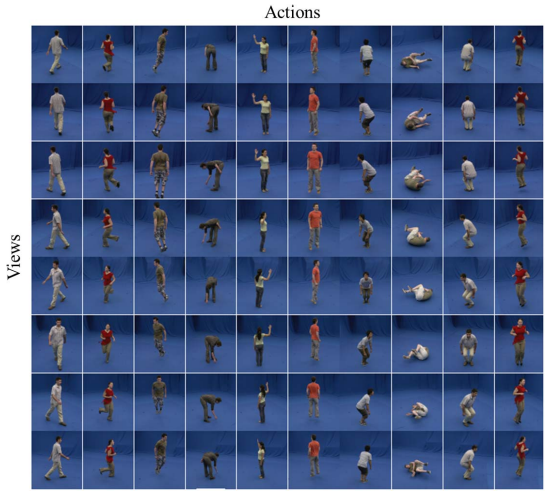



Fig. 19. Sampled frames from the i3DPost dataset.

TABLE V

AVERAGE ACCURACY FOR ARBITRARY VIEW RECOGNITION ON THE I3DPOST DATASET

| Methods | Mean |
|---|---|
| Holte et al. [68] | 92.2 |
| Iosifidis et al. [69] | 90.9 |
| Gkalelis et al. [65] | 90.0 |
| Zhang et al. [28] | 93.8 |
| Ours | **94.6** |

Table VI shows the comparison of cross-view action recognition results on the IXMAS, N-UCLA, UWA3DII and i3DPost dataset by using each descriptor independently and combining them together. Among the three descriptors, 3DHOF outperforms the other two in the most of dataset. However, it is clear that the combined feature produces far superior results that cannot be achieved by any single feature. This shows that our proposed features are complementary to each other.

TABLE VI

COMPARISON OF CROSS-VIEW ACTION RECOGNITION RESULTS ON THE IXMAS, N-UCLA, UWA3DII AND I3DPOST DATASET BY USING DIFFERENT FEATURES

| Features | IXMAS | N-UCLA | UWA3DII | i3DPost |
|---|---|---|---|---|
| 3D Trajectories | 58.1 | 57.1 | 48.6 | 90.1 |
| 3DHOF | 67.8 | 62.4 | 58.4 | 87.5 |
| 3DMBH | 66.3 | 56.3 | 53.2 | 81.6 |
| All Features Combined | **70.8** | **68.5** | **60.1** | **94.6** |

Fig. 20 and 21 show the performance of different descriptors according to different view transfer pairs on the IXMAS and UWA3DII datasets respectively. In all pairs, combining all the descriptors achieves better result than using them independently.

### *G. Evaluation of 2D Features Used in Our System*

While appearance information and movement information are both very important for describing the 2D action videos, such appearance information is quite different for 2D action videos captured from different points. We build a transfer learning framework to transfer 3D and 2D features into a common sparse feature space, and hence it is preferable that both of them have similar logical meanings. Therefore, any useful information on the 3D and 2D action videos such as appearance will assist our system. The reason we do not propose 3DHOG is that the surface texture of a 3D model remains unchanged over time. We conduct an experiment on the UWA3DII dataset to show the importance of appearance feature 2D HOG.

Fig. 22 shows the performance on only, without and with using 2D HOG respectively. Features combined with 2D Trajectories, 2D HOF and 2D MBH perform better than only using 2D HOG in all the view transfer pairs. Because the movement related descriptors contain more view-invariant information than appearance related descriptors on the 2D action videos. Combined features also perform better than

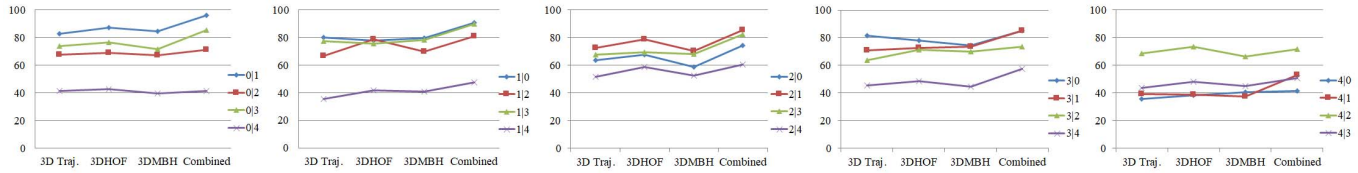



Fig. 20. Feature evaluation on IXMAS dataset according to different view transfer pairs.

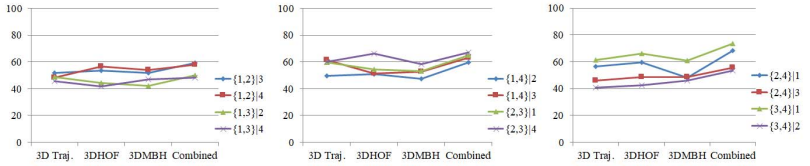



Fig. 21. Feature evaluation on UWA3DII dataset according to different view transfer pairs.

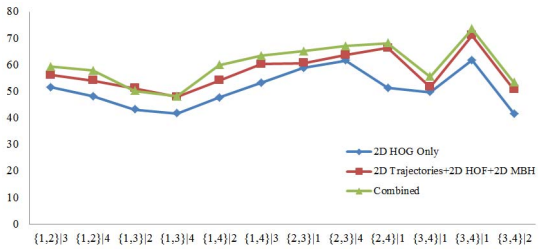



Fig. 22. 2D HOG evaluation of the UWA3DII dataset according to different view transfer pairs.

features without using 2D HOG, which shows the assistance of appearance information to our system.

## VII. Conclusion and Discussions

In this paper, we have proposed a view-invariant human action recognition framework. Unlike previous work, we construct a synthetic 3D and 2D video database using realistic human models, which is used to obtain the view-invariance through transfer dictionary learning. The trained dictionary is used to project real world 2D video into a view-invariant sparse representation, facilitating an arbitrary view action classifier. The use of synthetic data for initial training reduces the need for carefully captured video with view information. The synthetic dataset created in this project is open to the public, it is the first structured action dataset built with realistic human models for classification purposes. To enhance the quality of 3D motion description, we propose a new set of features known as the 3D dense trajectories, which consists of 3D trajectories, 3DHOF and 3DMBH. These features are complementary to each other and the combined feature set is highly effective for action classification. We demonstrate superior results in comparison to existing works in the IXMAS, NUCLA, UWA3DII and i3DPOST datasets.

In our system, we project the 3D and 2D videos into a common view-invariant sparse representation with the 3D and 2D dictionaries respectively. Theoretically speaking, it is possible to learn a dictionary that directly projects 2D video into 3D space, and consider the 3D space to be view-invariant. However, this is not practically possible. This is because 2D to 3D projection requires information that is not available in the 2D video. Even if a project matrix can be trained, the projected results will suffer from a large reconstruction error. In this research, we solve this problem by extracting the common view-invariant features in the 3D and 2D videos instead.

A main advantage of our framework is that the view-invariant transfer dictionary is pre-trained with a full synthetic dataset and fine-tuned with a small amount of real data. It is possible to include a large number of views in the synthetic dataset to learn a better view-invariant representation, even if the real data does not cover all of these views. Also, it is possible to introduce variations within each action class using computer graphics techniques such as motion style transfer to improve the richness of the dataset, which can enhance the classification accuracy. While existing work requires encoding and pooling parts to aggregate the local features, we use bag-of-words to effectively aggregate the local trajectories based features, motivated by the promising results from [8]–[11]. Specifically, we train a dictionary by using K-means to cluster the local features (e.g. HOG, HOF) into some visual words and then encode these local features by counting the occurrence of different visual words.

During the implementation, we found that the quality of the synthetic video could affect the classification accuracy of the system. This was the main motivation for us to utilize high-quality human models instead of simplified cylinder-based models as in previous works. In the future, we are interested to explore if more realistic rendering (such as photorealistic rendering with global illuminations) and more realistic character movement (such as introducing secondary deformation to simulate the involuntary movement of body fat and clothings) would further improve the system performance.

In many datasets, the facing angles of the actors are not aligned with that camera viewpoints. As a result, the same action may appear differently for the same viewpoints dependent on the faced direction. As a future direction, we are interested in introducing the facing angle into the classification framework, such that the system can understand how the action may appear dependent on the orientation of the actor. Furthermore, when creating synthetic 2D videos, our current

system samples projection viewpoints uniformly. With the facing angle, we may explore an optimal way of projection sampling that can optimize classification accuracy with a minimal number of synthetic 2D views.

Dictionary learning can be considered as a linear projection algorithm and can be limited in representing the view-invariance of 2D and 3D videos. In the future, we are interested in applying non-linear algorithms such as Neural Networks with synthetic training data to achieve better results. The potential challenges in using Neural Networks to learn the complex view-invariance is the need to tune a large number of hyper-parameters, as well as the need to design an optimal network architecture.

## REFERENCES


[1] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Oct. 2005, pp. 1395–1402.

[2] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc. 2nd Joint IEEE Int. Workshop Vis. Surveill. Perform. Eval. Tracking Surveill.*, Oct. 2005, pp. 65–72.

[3] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, nos. 2–3, pp. 107–123, 2005.

[4] J. Liu and M. Shah, "Learning human actions via information maximization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[5] J. Liu, Y. Yang, and M. Shah, "Learning semantic visual vocabularies using diffusion distance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 461–468.

[6] Z. Lin, Z. Jiang, and L. S. Davis, "Recognizing actions by shape-motion prototype trees," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sep. 2009, pp. 444–451.

[7] F. Lv and R. Nevatia, "Single view human action recognition using key pose matching and Viterbi path searching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–8.

[8] S. Wu, O. Oreifej, and M. Shah, "Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 1419–1426.

[9] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 60–79, 2013.

[10] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. IEEE Int. Conf. Comput. Vis. (CVPR)*, Jun. 2011, pp. 3169–3176.

[11] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 3551–3558.

[12] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (CVPR)*, Jun. 2014, pp. 1725–1732.

[13] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[14] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.

[15] L. Wang *et al.*, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 20–36.

[16] A. Gupta, J. Martinez, J. J. Little, and R. J. Woodham, "3D pose from motion for cross-view action recognition via non-linear circulant temporal encoding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 2601–2608.

[17] H. Rahmani and A. Mian, "Learning a non-linear knowledge transfer model for cross-view action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2458–2466.

[18] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2046–2053.

[19] A. Farhadi and M. K. Tabrizi, "Learning to recognize activities from the wrong view point," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 154–166.

[20] J. Liu, M. Shah, B. Kuipers, and S. Savarese, "Cross-view action recognition via view knowledge transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 3209–3216.

[21] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, "Cross-view action modeling, learning, and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 2649–2656.

[22] C.-H. Huang, Y.-R. Yeh, and Y.-C. F. Wang, "Recognizing actions across cameras by exploring the correlated subspace," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 342–351.

[23] D. Weinland, E. Boyer, and R. Ronfard, "Action recognition from arbitrary views using 3D exemplars," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2007, pp. 1–7.

[24] P. Yan, S. M. Khan, and M. Shah, "Learning 4D action feature models for arbitrary view action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–7.

[25] H. Rahmani, A. Mian, and M. Shah, "Learning a deep model for human action recognition from novel viewpoints," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 667–681, Mar. 2018.

[26] N. Iwamoto, H. P. H. Shum, L. Yang, and S. Morishima, "Multi-layer lattice model for real-time dynamic character deformation," *Comput. Graph. Forum*, vol. 34, no. 7, pp. 99–109, Oct. 2015.

[27] C. Hecker, B. Raabe, R. W. Enslow, J. DeWeese, J. Maynard, and K. van Prooijen, "Real-time motion retargeting to highly varied user-created morphologies," *ACM Trans. Graph.*, vol. 27, no. 3, 2008, Art. no. 27.

[28] J. Zhang, L. Zhang, H. P. H. Shum, and L. Shao, "Arbitrary view action recognition via transfer dictionary learning on synthetic training data," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 1678–1684.

[29] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl, "3D shape histograms for similarity search and classification in spatial databases," in *Proc. Int. Symp. Spatial Databases*, 1999, pp. 207–226.

[30] M. Körtgen, G.-J. Park, M. Novotni, and R. Klein, "3D shape matching with 3D shape contexts," in *Proc. 7th Central Eur. Seminar Comput. Graph.*, vol. 3, 2003, pp. 5–17.

[31] P. Huang and A. Hilton, "Shape-colour histograms for matching 3D video sequences," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Sep. 2009, pp. 1510–1517.

[32] S. Pehlivan and P. Duygulu, "A new pose-based representation for recognizing actions from multiple cameras," *Comput. Vis. Image Understand.*, vol. 115, no. 2, pp. 140–151, 2011.

[33] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volume," *Comput. Vis. Image Understand.*, vol. 104, nos. 2–3, pp. 249–257, 2006.

[34] M. Pierobon, M. Marcon, A. Sarti, and S. Tubaro, "3-D body posture tracking for human action template matching," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, vol. 2, 2006, pp. 501–504.

[35] I. Cohen and H. Li, "Inference of human postures by classification of 3D human body shape," in *Proc. IEEE Int. Workshop Anal. Modeling Faces Gestures (AMFG)*, Oct. 2003, pp. 74–81.

[36] B. Zhang, Y. Yang, C. Chen, L. Yang, J. Han, and L. Shao, "Action recognition using 3D histograms of texture and a multi-class boosting classifier," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4648–4660, Oct. 2017.

[37] M. Liu, H. Liu, C. Chen, and M. Najafian, "Energy-based global ternary image for action recognition using sole depth sequences," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, 2016, pp. 47–55.

[38] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognit.*, vol. 68, pp. 346–362, Aug. 2017.

[39] K. Wang, G. Zhang, and S. Xia, "Templateless non-rigid reconstruction and motion tracking with a single RGB-D camera," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5966–5979, Dec. 2017.

[40] C. Jia and Y. Fu, "Low-rank tensor subspace learning for RGB-D action recognition," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4641–4652, Oct. 2016.

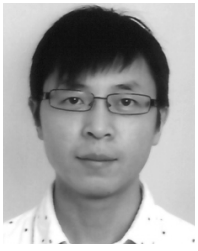
[41] Y. Kong and Y. Fu, "Discriminative relational representation learning for RGB-D action recognition," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2856–2865, Jun. 2016.

[42] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proc. 15th ACM Int. Conf. Multimedia*, 2007, pp. 357–360.
[43] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 650–663.
[44] R. Messing, C. Pal, and H. Kautz, "Activity recognition using the velocity histories of tracked keypoints," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sep. 2009, pp. 104–111.
[45] P. Matikainen, M. Hebert, and R. Sukthankar, "Trajectons: Action recognition through the motion analysis of tracked features," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Sep. 2009, pp. 514–521.
[46] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li, "Hierarchical spatio-temporal context modeling for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 2004–2011.
[47] J. Sun, Y. Mu, S. Yan, and L.-F. Cheong, "Activity recognition using dense long-duration trajectories," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2010, pp. 322–327.
[48] F. Liu, X. Xu, S. Qiu, C. Qing, and D. Tao, "Simple to complex transfer learning for action recognition," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 949–960, Feb. 2016.
[49] T. Xu, F. Zhu, E. K. Wong, and Y. Fang, "Dual many-to-one-encoder-based transfer learning for cross-dataset human action recognition," *Image Vis. Comput.*, vol. 55, pp. 127–137, Nov. 2016.
[50] M. Zhou, H. Chen, L. Ren, G. Sapiro, L. Carin, and J. W. Paisley, "Non-parametric Bayesian dictionary learning for sparse image representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 2295–2303.
[51] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.
[52] Y. Wang and G. Mori, "Hidden part models for human action recognition: Probabilistic versus max margin," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 7, pp. 1310–1323, Jul. 2010.
[53] M. Aharon, M. Elad, and A. Bruckstein, "$K$-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
[54] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa, "Domain adaptive dictionary learning," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 631–645.
[55] J. Zheng, Z. Jiang, and R. Chellappa, "Cross-view action recognition via transferable dictionary learning," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2542–2556, Jun. 2016.
[56] R. Gross and J. Shi, "The CMU motion of body (MoBo) database," Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-01-18, Jun. 2001.
[57] J. Tilmanne, R. Sebbe, and T. Dutoit, "A database for stylistic human gait modeling and synthesis," in *Proc. eNTERFACE Workshop Multimodal Interfaces*, Paris, France, 2008, pp. 91–94.
[58] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 689–696.
[59] F. Zhu and L. Shao, "Weakly-supervised cross-domain dictionary learning for visual recognition," *Int. J. Comput. Vis.*, vol. 109, nos. 1–2, pp. 42–59, Aug. 2014.
[60] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent K-SVD," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1697–1704.
[61] G. H. Golub, P. C. Hansen, and D. P. O'Leary, "Tikhonov regularization and total least squares," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 1, pp. 185–194, 1999.
[62] R. Li and T. Zickler, "Discriminative virtual views for cross-view action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 2855–2862.
[63] Z. Zhang, C. Wang, B. Xiao, W. Zhou, S. Liu, and C. Shi, "Cross-view action recognition via a continuous virtual path," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2690–2697.
[64] B. Li, O. I. Camps, and M. Sznaier, "Cross-view activity recognition using Hankelets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1362–1369.
[65] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, and I. Pitas, "The i3Dpost multi-view and 3D human action/interaction database," in *Proc. IEEE Conf. Vis. Media Prod. (CVMP)*, Nov. 2009, pp. 159–168.
[66] G. Gkioxari and J. Malik, "Finding action tubes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 759–768.
[67] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2625–2634.
[68] M. B. Holte, T. B. Moeslund, N. Nikolaidis, and I. Pitas, "3D human action recognition for multi-view camera systems," in *Proc. Int. Conf. 3D Imag., Modeling, Process., Vis. Transmiss. (3DIMPVT)*, 2011, pp. 342–349.
[69] A. Iosifidis, N. Nikolaidis, and I. Pitas, "Movement recognition exploiting multi-view information," in *Proc. IEEE Int. Workshop Multimedia Signal Process. (MMSP)*, Oct. 2010, pp. 427–431.

**Jingtian Zhang** received the B.Eng. degree in information engineering from the Nanjing University of Aeronautics and Astronautics, China, and the M.Sc. degree in electrical and electronic engineering from the University of Sheffield, U.K. He is currently pursuing the Ph.D. degree with the Department of Computer and Information Sciences, Northumbria University, U.K. His research interests include computer vision, computer graphics, motion analysis, and machine learning.

**Hubert P. H. Shum** received the B.Eng. and M.Sc. degrees from the City University of Hong Kong, and the Ph.D. degree from the School of Informatics, The University of Edinburgh, U.K. He was a Senior Lecturer with Northumbria University, U.K., a Lecturer with the University of Worcester, U.K., a Post-Doctoral Researcher with RIKEN, Japan, and a Research Assistant with the City University of Hong Kong. He is currently an Associate Professor (Reader) in computer science with Northumbria University and the Director of the Research and Innovation of the Computer and Information Sciences Department. His research interests include computer graphics, computer vision, motion analysis, and machine learning.

**Jungong Han** was a Faculty Member with the Department of Computer and Information Sciences, Northumbria University, U.K. He is currently a tenured Senior Lecturer (Associate Professor) with the Data Science Institute, Lancaster University. His research interests include computer vision, image processing, machine learning, and artificial intelligence.



**Ling Shao** is currently the CEO and Chief Scientist of the Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates. He is also a Professor with the School of Computing Sciences, University of East Anglia, Norwich, U.K. He is an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and several other journals.