Learning 4D Action Feature Models for Arbitrary View Action Recognition

Pingkun Yan, Saad M. Khan, Mubarak Shah
Computer Vision Lab,
University of Central Florida, Orlando, FL
http://www.eecs.ucf.edu/~vision/

Abstract

In this paper we present a novel approach using a 4D (x,y,z,t) action feature model (4D-AFM) for recognizing actions from arbitrary views. The 4D-AFM elegantly encodes shape and motion of actors observed from multiple views. The modeling process starts with reconstructing 3D visual hulls of actors at each time instant. Spatiotemporal action features are then computed in each view by analyzing the differential geometric properties of spatio-temporal volumes (3D STVs) generated by concatenating the actor's silhouette over the course of the action (x,y,t). These features are mapped to the sequence of 3D visual hulls over time (4D) to build the initial 4D-AFM. Actions are recognized based on the scores of matching action features from the input videos to the model points of 4D-AFMs by exploiting pairwise interactions of features. Promising recognition results have been demonstrated on the multi-view IXMAS dataset using both single and multi-view input videos.

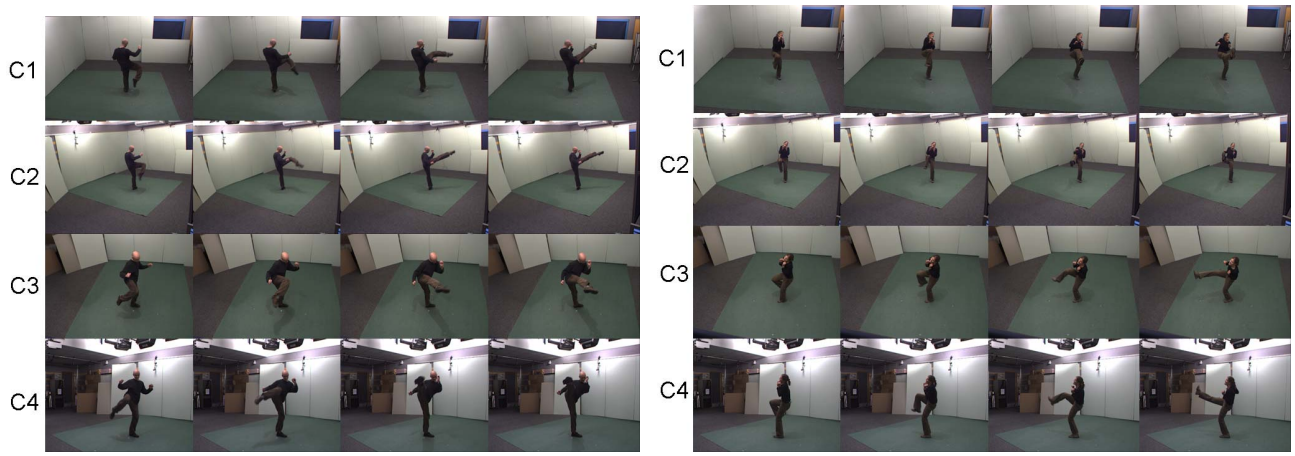
1. Introduction

Human action and event recognition from video has been actively investigated in the field of computer vision due to its fundamental importance to many video content analysis applications. In this paper, we address the problem of recognizing human actions from videos taken from arbitrary views. This is a very challenging problem in that the same action may look quite different when observed from different angles, in addition to the variation caused by the personal action difference of people as shown in Figure 1. Action recognition from arbitrary views entails a two pronged solution: first deciding what features are most suitable and second creating view invariance in the action description to allow actions to be learned and recognized.

Bounding boxes or articulated "cardboards" are suitable for representing rigid or semi-rigid objects [5, 7], but they are approximations and tend to discard much of the detailed shape information that can be highly discriminative for human activity recognition tasks. Image/volume patches based descriptors have also been used for action recognition [4, 8, 12] based on their success in object recognition. However, the overwhelming appearance suppresses the shape and motion in the video, which are essential for recognizing actions. On the other hand, studies in the field of object recognition in 2D images have demonstrated that silhouettes contain detailed shape information of objects [1]. When a silhouette is sufficiently detailed, people can readily identify the object, or judge its similarity to other shapes [18]. In the recent past, an interesting new approach has been to use a temporal concatenation of contours/silhouettes of person performing an action to create space-time action shape or object [2, 20]. Such a representation contains rich descriptive information about the action performed.

Although the spatiotemporal features can be quite powerful in recognizing actions observed from similar views, they tend to falter with changing viewpoint. Therefore, the second step for arbitrary view action recognition is to build the connections between the different views for modeling the actions in higher dimensional spaces. Several approaches have been proposed in literature [3, 17, 13, 14]. These approaches have focused on representations in which view dependent information is removed, often at the cost of an impoverished action model and without adding full flexibility in camera configurations. A very recent and interesting work is that of Lv *et al*. [11], where a graphical model called *Action Net* is built to connect 2D key poses of actors to represent 3D shapes for action recognition. However, the essential motions for recognizing actions are may not be well captured. On the other hand, another group of approaches tried to directly estimate 3D shapes and poses from multi-view inputs for action recognition [6, 9]. Most recently, Weinland *et al*. [19] proposed to identify actions by analyzing a sequence of 3D exemplars estimated from single 2D views using hidden Markov model. However, it is well known that reconstructing 3D pose from a single view is very difficult. More importantly, motion information could be missed out using only sampled time instances with the silhouette information.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE



(a) Kicking Action of Andreas (b) Kicking Action of Alba

Figure 1. Multi-view video frames of kicking action performed by two people. It can be seen that the same action may look quite different when being observed from different viewing angles. Variation also exists when the action is performed by different actors.

In our approach, we develop a 4D *action feature model* (4D-AFM) for representation and recognition of actions from arbitrary views. The AFM elegantly encodes shape and motion of actors observed from multiple views. The modeling process starts with reconstructing 3D shapes at each moment using *multi-view model videos*. In essence, this creates a *4D action shape* (spatio-temporal visual hull) of the action. Spatiotemporal action features are then computed *in each view* by analyzing the differential geometric properties of spatial temporal volumes (x, y, t) generated by concatenating the actor's silhouette over the course of the action. These features are then mapped to the 4D action shape to build the initial 4D-AFM. These training videos can be either multi-view videos or single view videos taken from arbitrary views with unknown camera parameters. Actions are recognized based on the scores of matching action features from the input videos to the model points of AFMs with pairwise interactions.

The rest of the paper is organized as follows. Section 2 presents the construction of 4D-AFM. The AFM based action recognition algorithm and the algorithm for learning the parameters of AFMs are provided in Sections 3 and 3.3, respectively. Section 4 presents the recognition results, and finally, Section 5 concludes the work.

2. Building 4D Action Feature Model

In this section, we present the approach for constructing the initial 4D-AFM. The AFM consists of the 4D action shape (sequence of 3D shapes) with spatiotemporal features attached. The process of building 4D-AFM is shown in Figure 3.

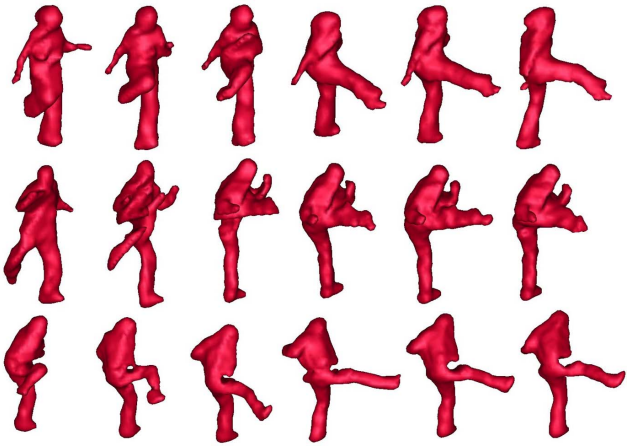


Figure 2. Kicking action of three different actors visualized in 4D. The similarity between the sequences becomes clear when the actions are reconstructed in 4D from multi-view videos.

2.1. Reconstructing Sequence of 3D Shapes

The modeling process of the 4D-AFM starts with the reconstruction of the sequence of 3D shapes (4D action shape) using multi-view video sequences. These videos are called *model videos* in our approach, which are captured by using multiple cameras around the actors. The INRIA Xmas Motion Acquisition Sequences (IXMAS) dataset is used in our work. The dataset was also used by [11] and [19]. In this dataset, multi-view videos were recorded by 5 calibrated cameras. The projection matrix of each camera is provided. At each moment, a 3D shape is reconstructed by computing the visual hulls via back projecting the multi-view 2D silhouettes into the 3D space like the method in [16]. In this work, we used the reconstructed 3D volumes provided together with the multi-view videos. The 4D action shape is

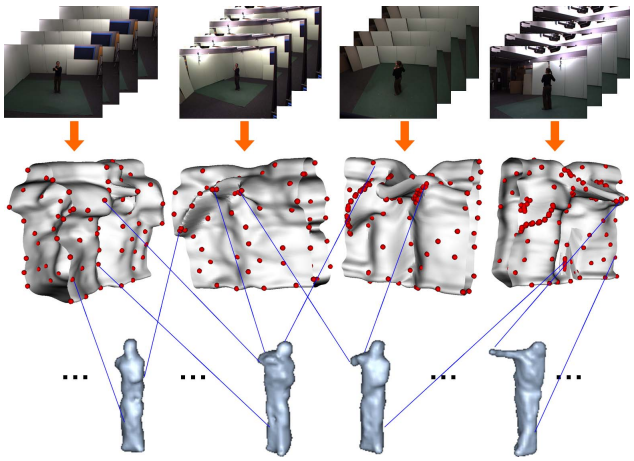


Figure 3. Illustration of constructing 4D-AFM. The first row shows videos from four different views. The second row shows the STVs extracted from the videos and the locations of the spatiotemporal action features on the surface of STVs. These action features are mapped to the 4D action shape as shown in the third row.

composed by concatenating in time the sequence of reconstructed 3D shapes as shown in Figure 2.

2.2. Extracting Action Features

The action features used in our work are computed from a STV of the actor's silhouette, by analyzing the differential geometric surface properties [20]. The first step in the approach is to automatically generate the STV from a sequence of contours. This task is completed by solving the correspondence problem between the contours in the consecutive frames using a graph theoretical approach as outlined in [20]. Once the STV is generated, a set of action features describing changes in direction, speed, and shape of parts of contour are computed, which is called the *action sketch*. There are eight fundamental surface types of STVs including peak, ridge, saddle ridge, flat, minimal, pit, valley, and saddle valley. These are defined by two metric quantities, the Gaussian curvature κ, and mean curvature h, computed from the first and second fundamental forms of the underlying differential geometry. The feature descriptors are not computed for every vertex of the surface, but only for some interest points. These interest points are selected from the surface vertexes, where the curvature maxima and minima occur as shown in Figure 3.

2.3. Mapping 3D Features to 4D-AFM

The reconstructed 4D action shape is not directly used in the proposed approach due to the aforementioned difficulties in Section 1. Instead, in order to efficiently relate the action features computed from videos (*i.e.* STVs) recorded in different views and performed by different people, a new representation for combining the action sketches is employed. In the proposed approach, all the features obtained from the multi-view model videos are attached to the surfaces of the previously reconstructed 4D action shape as shown in Figure 3. In this way, the spatiotemporal connections of action features are clearly modeled. Mapping the action features to 4D-AFM is performed by using the projection matrix $\mathbf{P}$ of each camera. Let $\mathbf{l}_i = (x_i, y_i, t_i)$ denote the location of the ith feature on the surface of a STV (at time t_i). The 4D location of the feature can be easily determined by back projecting the interest point on the 4D surface as $\mathbf{L}_i = \mathbf{P}^{+}\mathbf{l}_i$, where $\mathbf{L}_i = (X_i, Y_i, Z_i)$, with time index t_i and $\mathbf{P}^{+}$ is the pseudoinverse matrix of $\mathbf{P}$.

This novel representation has several advantages. Firstly, we can avoid storing all the multi-view 2D training frames and/or silhouettes. Thus, there is no need to build complicated connections between the views. The spatiotemporal relationship between the feature points from different views are readily available, which can be easily used when matching features for action recognition. Secondly, since the extracted spatiotemporal features instead of only sampled 3D shapes are used for describing actions, the recognition can be much simplified without the sophisticated inference of 3D shapes from single 2D view. Finally, it is worth noting that the model videos are only required for building the 4D-AFM at the initial step. The correspondences and the weights of the action features are automatically determined during the learning process. The details are provided in Section 3.3.

3. Arbitrary View Action Recognition

Recognition of the action contained in a given video V taken from an arbitrary view can be modeled as finding the action $\mathcal{A}^*$ such that

$$\mathcal{A}^* = \arg \max_{1 \le i \le n} P(\mathcal{A}_i|V), \tag{1}$$

where $\{\mathcal{A}_i | i = 1, 2, \ldots, n\}$ is a set of known actions. To compute the probability on the right side of Eqn. (1), we apply the Bayes' rule and have

$$P(\mathcal{A}_i|V) = \frac{P(V|\mathcal{A}_i)P(\mathcal{A}_i)}{P(V)}. \tag{2}$$

Since the denominator $P(V)$ on the right side of Eqn. (2) does not depend on the action $\mathcal{A}_i$, it can be ignored when estimating $\mathcal{A}^*$.

As described in Section 2, each known action A_i is represented by a 4D-AFM $\mathcal{M}_i$ in our work. Since the 4D-AFMs are built using multi-view video data, actions observed from arbitrary views can be recognized. For an input video V, a STV is first generated and then an action sketch F is computed. Since the input video can be represented using the abstract action features F, we are then able to model the

probabilities in Eqn. (2) as

$$P(\mathcal{A}_i|V) = P(\mathcal{M}_i|F) \sim P(F|\mathcal{M}_i)P(\mathcal{M}_i). \quad (3)$$

The probability of the AFM $\mathcal{M}_i$ can be computed as $P(\mathcal{M}_i) = n_i/N$, where n_i is the number of training samples in the ith action category and N is the number of total training videos. The key part for recognizing action is then to estimate the likelihood $P(F|\mathcal{M}_i)$, which can be solved by using feature matching, since 4D-AFM $\mathcal{M}_i$ is also composed by action features.

3.1. Matching Pairwise Action Features

Similar to the work by Leordeanu *et al*. [10] of using pairwise feature interactions in object recognition, we explicitly represent the pairwise relations between action features in the input action sketch and the learned 4D-AFMs for matching, in contrast to independent feature matching. The usefulness of pairwise geometric constraints between features is identified based on the observation that accidental alignments of features are rare and they can be effective in pruning the search for correspondences between sets of model and input features.

To find which feature in the test video best matches a model part, we formulate it as a quadratic assignment problem (QAP) [10], which incorporates the second order relationships. The matching score E is written as:

$$E_x = \sum_{ja;kb} x_{ja}x_{kb}G_{ja;kb}. \quad (4)$$

Here $\mathbf{x}$ is an indicator vector with an entry for each pair (j, a) such that $x_{ja} = 1$ if the model part i is matched to video action feature a and 0 otherwise. The mapping constraints can be enforced that one model part can match only one action feature and vice versa, *i.e*. $\sum_j x_{ja} = 1$ and $\sum_a x_{ja} = 1$.

Considering the action sketch and AFMs as connected graphs, the pairwise potential $G_{ja;kb}$ in Eqn. (4) reflects how well the model points j and k preserve their geometric relationship when being matched to features a and b in the video. As presented in Section 2, each action feature j can be seen as an abstract point with associated normals $\vec{n}_j$, Gaussian curvature κ_j, and mean curvature h_j. For a pair of model points (j, k), their relationship is captured in the vector $e_{jk} = \{\kappa_j, \kappa_k, h_j, h_k, \vec{n}_j, \vec{n}_k, \vec{d}_{jk}, \vec{d}_{kj}\}$, where $\vec{d}_{jk}$ is the directional distance from feature point j to feature point k and $\vec{d}_{kj}$ is the inverse. It is obvious that the pair vector e_{jk} is translation invariant. In the proposed approach, it is also scale invariant since the spatiotemporal space has been normalized. With the definition of the feature pairs, the pairwise potentials are modeled using logistic classifiers:

$$G_{ja;kb} = \frac{1}{1 + \exp(-\|\mathbf{w}^T(e_{jk} - e_{ab})\|^2)}, \quad (5)$$

where $\mathbf{w}$ is a vector containing the weight of each element. According to the dynamic ranges of the data, we empirically chose $\mathbf{w} = \{0.1, 0.1, 0.12, 0.12, 7.1, 7.1, 2.2, 2.2\}^T$.

The matched feature pairs are determined by computing the mapping assignment $\mathbf{x}^*$ that maximizes the matching score E in Eqn. (4) by picking the largest entries of $\mathbf{G}$,

$$\mathbf{x}^* = \arg\max(\mathbf{x}^T\mathbf{G}\mathbf{x}). \quad (6)$$

Motivated by the work of [15], which gives good approximate solutions, the mapping assignment $\mathbf{x}^*$ is efficiently computed by using a graph theoretical approach. It takes about 1.5 minutes to handle thousands of fully connected points on a 1.8GHz desktop computer.

3.2. Recognizing Actions

Once the matched feature pairs are found, we can compute the likelihood $P(F|\mathcal{M}_i)$ in Eqn. (3) as

$$P(F|\mathcal{M}_i) = \prod_{a,b} P(e_{jk}|x^*_{ja}, x^*_{kb}, m^i_{ab}), \quad (7)$$

by assuming that the matched pairs are independent from others, where e_{jk} denotes the pair of action features j and k and m^i_{ab} denotes the matched pair of model points a and b in the ith 4D-AFM indicated by the alignment $\mathbf{x}^*$. The probability $P(e_{jk}|x^*_{ja}, x^*_{kb}, m^i_{ab})$ is a function of several factors. Firstly, the probability depends on the quality of the match as given by the pairwise potentials. Better match results in higher probability and vice versa. Secondly, the probability should be closely related with the frequency of the matched model points appearing in the actions falling in the same category. The frequency of observing model points in actions indicates the relevance of the model points to these actions. The more relevant features contribute more to the recognition of actions than those less relevant. Therefore, in our work, a *relevance* parameter r_i is associated to each feature to measure how significant the feature can be for recognizing an action. A larger value of the relevance indicates that the feature is more related to an action and vice versa. The relevance vector $\mathbf{r}$ of the feature pairs are set during the learning process, which is described in detail in Section 3.3.

With both degree of match and relevance parameter considered, we formulate the likelihood probability as

$$P(e_{jk}|x^*_{ja}, x^*_{kb}, m^i_{ab}) \sim \exp\left(r_a r_b \sum_{j,k} x^*_{ja}x^*_{kb}G_{ja;kb}\right), \quad (8)$$

subject to normalization. Note that the likelihood probability does not depend on $\{j, k\}$. The reason is that only the matched feature pairs are counted for computing this probability. Given a pair of model points $\{a, b\}$, the feature pair

$\{j, k\}$ is fixed. For the efficiency of computation, we define a score S_i of recognizing the input action as the ith known action by taking log of Eqn. (7) and have:

$$S_i = \sum_{a,b} r_a r_b \sum_{j,k} x^*_{ja} x^*_{kb} G_{ja;kb}. \quad (9)$$

From the possible matches between the input video and the learned AFMs, we select the corresponding action $\mathcal{A}^*$ with the maximum matching score as the recognized action

$$\mathcal{A}^* = \arg \max_{1 \le i \le N} P(F|\mathcal{M}_i) = \arg \max_{1 \le i \le N} S_i, \quad (10)$$

where N is the total number of the known action categories.

3.3. Learning Action Feature Models

The most important parameter that needs to be learned in our approach is the relevance vector $\mathbf{r}$. We present the learning algorithm in this section. In our work, the relevance vector for each AFM $\mathcal{M}_i$ is learned separately. When learning the ith action, we binarize the ground truth into 1 for the current action category and 0 for all the others.

The relevance parameters are initialized with equal value $1/|\mathbf{r}|$, where $|\mathbf{r}|$ denotes the length of vector $\mathbf{r}$, which is also equal to the number of model points. The parameters are learned by minimizing the difference between the recognition results and the ground truth. The objective function to be minimized subject to relevance $\mathbf{r}$ is defined as

$$J = \sum_{m=1}^{M} \left(\phi(S_i^m) - t^m\right)^2, \quad (11)$$

where t^m is the ground truth for the mth action sketch and function $\phi(x)$ is defined as

$$\phi(x) = \frac{1 - \exp(-x)}{1 + \exp(-x)}. \quad (12)$$

The relevance parameter is then updated by using gradient descent method to minimize the function in Eqn. (11)

$$\Delta r_a = \eta(t^m - \phi(S_i^m))\psi(S_i^m)(1 - \psi(S_i^m))\frac{\partial S_i^m}{\partial r_a}, \quad (13)$$

where function $\psi(x)$ is a sigmoid function defined as

$$\psi(x) = \frac{1}{1 + \exp(-x)}, \quad (14)$$

and

$$\frac{\partial S_i^m}{\partial r_a} = \sum_b r_b \sum_{j,k} x^*_{ja} x^*_{kb} G_{ja;kb}. \quad (15)$$

The parameters are learned sequentially over all the training action in each loop. After each update, the vector $\mathbf{r}$ is normalized to make $\sum_i r_i = 1$.

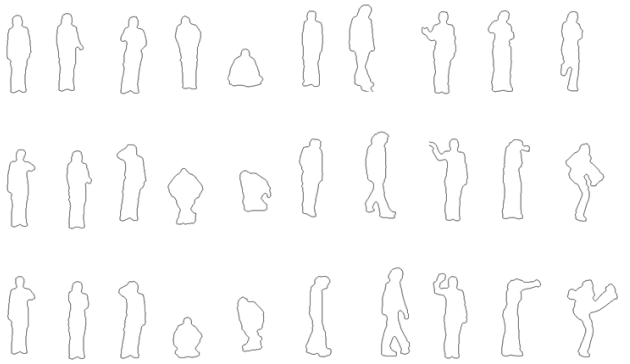


Figure 4. Examples of the contours of each action extracted from the videos for generating STVs. Each column contains one action. From left to right are the 11 actions for recognition in our work: checking watch, crossing arms, scratching head, sitting down, getting up, turning around, walking, waving hand, punching, kicking, picking up.

4. Experimental Results

The multi-view IXMAS dataset is used in our experiments. The dataset contains 13 daily-life motions and each was performed 3 times by 12 actors. The actors arbitrarily choose position and orientation. Each action is recorded by 5 cameras with frame rate of 23fps. These camera are calibrated and the parameters are provided. The silhouettes of actors in all the videos are also provided, which are directly used to compute STVs. Note that the frame rate has no influence on our algorithm, since STV is a continuous representation in the normalized time scale. This provides great flexibility for learning and recognizing actions with different lengths due to the difference between either the actors or the camera settings.

Since the trained 4D-AFM contains the 3D shapes and their motion, it can be used to recognize actions not only from single view videos but also by combining multi-view videos efficiently to improve the performance. In this section, we demonstrate the performance of 4D-AFM based approach on single and multi-view recognition separately. For the multi-view based action recognition, we used a simple weighted voting strategy for combining the single view recognition results $\hat{S}_i = \sum_v S_i^v$, where S_i^v is the score of the vth view recognized as the ith action. The action is recognized based on the multi-view score as

$$\hat{\mathcal{A}}^* = \arg \max_i \hat{S}_i. \quad (16)$$

In our experiments, the leave-one-out strategy is employed. In each run, we select one actor for testing. The multi-view videos of one of the other actors are taken as model videos for building the initial 4D-AFMs. When computing the action features, we only used the first 4 views, since the last view is from the top and not discriminative for

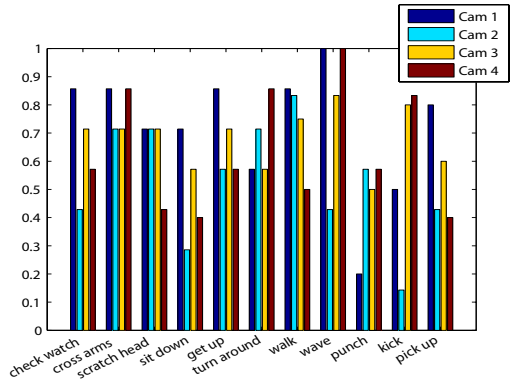


Figure 5. The recognition results using single views. The average recognition rates are 72%, 53%, 68% and 63% for cameras 1 to 4, respetively.

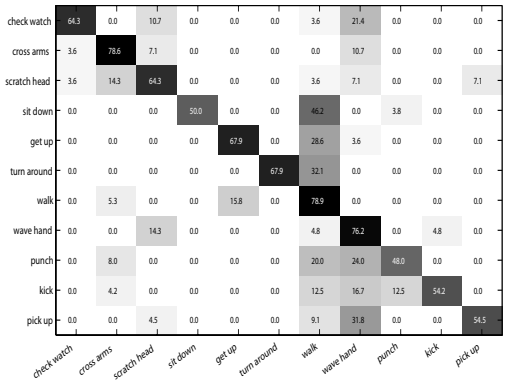


Figure 6. The confusion matrix of the recognition results using single views.

actions. Videos of the remaining actors are used for learning the 4D-AFM of each action. Some example contours of the actions used in our work are shown in Figure 4 including checking watch, crossing arms, scratching head, sitting down, getting up, turning around, walking, waving, punching, kicking, pointing, picking up, and throwing.

It is worth noting that although the cameras are fixed in the IXMAS dataset, the actors' orientations during performing the actions are not restricted. Therefore, the viewpoints of recorded actions are actually unknown and the dataset contains actions in arbitrary views. Thus, the dataset is suitable for testing the proposed 4D-AFM based action recognition method.

4.1. Single View Recognition

For a direct comparison to the results in [19], we used only 11 actions as shown in Figure 4, but unlike them we report results on four views instead of the three best. In this first experiment, after the initial 4D-AFM is built, learning is performed by using other views. In each round of testing, an action sketch extracted from a single view video is input into the system. An action label is then provided by computing the scores of matching the action sketch to the learned 4D-AFM. The action recognition results using single view videos for each action class from each camera are shown in figure 5. In our experiment, for actions like 'get up' and 'walk', the recognition results are consistently high. This we believe is due to their distinctness regardless of the view. On the average the recognition rates are 72%, 53%, 68%, 63% for camera 1, 2, 3 and 4, respectively.

As compared to the accuracy reported in [19] (55%, 64%, 60% for cameras 1, 2 and 4 respectively), our performance is better except for camera 2. There are two chief reasons for this. Firstly, the spatiotemporal information is fully exploited in our work thanks to the 4D action shape sequence and the attached spatiotemporal action features. While in the work of [19], only the temporal sequential information between the 3D exemplars is considered for recognizing actions. Secondly, action sketches from different views are integrated in the 4D-AFM in a unified manner, which makes the computation of matches between the model points and the action features in test video from an arbitrary view very efficient. In figure 6, we show the confusion matrix for the different actions averaged over all the single views.

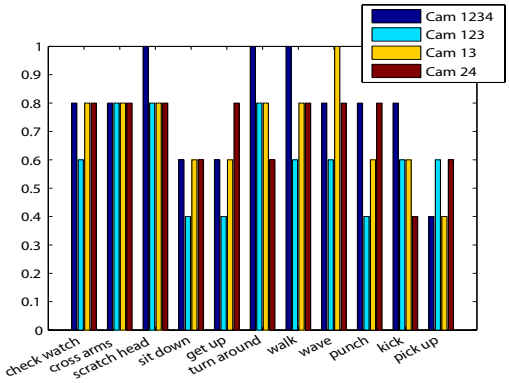


Figure 7. The recognition results using multi-view videos. The average recognition rates are 78%, 60%, 71% and 71% for camera combinations '1234', '123', '13', and '24', respetively.

4.2. Multiple View Recognition

We have also tested our approach by using multiple views simultaneously for recognition to simulate situations where more than a single view of the action may be available. We have tested using all four views as well as combinations of three and two views. In figure 7, we show recognition results of using the camera combinations '1234', '123', '13', and '24'. The average recognition results for these combinations are summarized in Table 1. It can be seen that the recognition performance can be improved when multi-view videos are available, which are comparable to the results present in [19].

Table 1. Average recognition rates by combining multi-view videos taken by multiple cameras. Results of different camera combinations are shown.

Cameras	1+2+3+4	1+2+3	1+3	2+4
Rate (%)	78	60	71	71

It is interesting to see that when all the 4 views are used, all the instances of the actions of 'scratching head', 'turning around', and 'walking' can be recognized in our experiment. The main reason is that the action features of these actions are quite discriminative. For example, the STV of 'scratching head' is significantly different from other actions, which results in the distinction of the action features extracted from the STV. Another possible cause of this could be that the 4D-AFMs of these actions contain dominating action features, which lead to high recognition accuracy of these actions with the price of increased false positives, as suggested by the 7th column of the confusion matrix in Figure 6.

5. Conclusions

In this paper, we have presented a new framework of learning feature models for recognizing actions in arbitrary views. Instead of directly using the 3D shapes and poses of actors, the proposed method builds a 4D-AFM for establishing spatiotemporal connections between videos recorded in different views. This is achieved by mapping action features from individual views to the surfaces of 4D action shapes obtained from time ordered multi-view 3D reconstructions of the actors. The proposed approach exploits the relationship between multi-view videos and their action sketches in a more unified manner. Experimental evaluation of the proposed method suggests collaborative information in the multi-view training videos can be represented efficiently and effectively through 4D-AFMs. We have also revealed that spatiotemporal action features containing both shape and motion can be salient properties to assist in action recognition. Performance of the proposed method has been evaluated using the IXMAS dataset and promising results have been demonstrated.

Acknowledgments

This research was funded in part by the U.S. Government VACE program.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002.

[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005.

[3] L. W. Campbell, D. A. Becker, A. Azarbayejani, A. F. Bobick, and A. Pentland. Invariant features for 3-D gesture recognition. In *Int. Conf. Automatic Face and Gesture Recognition (FG)*, pages 157–163, 1996.

[4] N. P. Cuntoor and R. Chellappa. Epitomic representation of human activities. In *CVPR*, 2007.

[5] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, pages 726–733, 2003.

[6] D. Gavrila and L. Davis. 3D model-based tracking of humans in action: a multi-view approach. In *CVPR*, 1996.

[7] S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated image motion. In *Int. Conf. Automatic Face and Gesture Recognition (FG)*, pages 38–44, 1996.

[8] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *ICCV*, pages 166–173, 2005.

[9] M. W. Lee and I. Cohen. Proposal maps driven mcmc for estimating human body pose in static images. In *CVPR*, pages 334–341, 2004.

[10] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. In *CVPR*, 2007.

[11] F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *CVPR*, 2007.

[12] J. C. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *CVPR*, 2007.

[13] V. Parameswaran and R. Chellappa. View invariance for human action recognition. *International Journal of Computer Vision*, 66(1):83–101, 2006.

[14] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, 50(2):203–226, 2002.

[15] L. Shapiro and R. Haralick. Structural descriptions and inexact matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 3(9):504–519, 1981.

[16] P. Y. S.M. Khan and M. Shah. A homographic framework for the fusion of multi-view silhouettes. In *ICCV*, 2007.

[17] T. Syeda-Mahmood, M. Vasilescu, and S. Sethi. Recognizing action events from multiple viewpoints. In *IEEE Workshop on Detection and Recognition of Events in Video*, pages 64–72, 2001.

[18] L. Wang and D. Suter. Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. In *CVPR*, 2007.

[19] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3D exemplars. In *ICCV*, 2007.

[20] A. Yilmaz and M. Shah. Action sketch: A novel action representation. In *CVPR*, 2005.