Recognizing Human Actions by Learning and Matching Shape-Motion Prototype Trees

Zhuolin Jiang, *Member, IEEE,* Zhe Lin, *Member, IEEE,* Larry S. Davis, *Fellow, IEEE*

***Abstract*— A shape-motion prototype-based approach is introduced for action recognition. The approach represents an action as a sequence of prototypes for efficient and flexible action matching in long video sequences. During training, an action prototype tree is learned in a joint shape and motion space via hierarchical K-means clustering and each training sequence is represented as a labeled prototype sequence; then a look-up table of prototype-to-prototype distances is generated. During testing, based on a joint probability model of the actor location and action prototype, the actor is tracked while a frame-to-prototype correspondence is established by maximizing the joint probability, which is efficiently performed by searching the learned prototype tree; then actions are recognized using dynamic prototype sequence matching. Distance measures used for sequence matching are rapidly obtained by look-up table indexing, which is an order of magnitude faster than brute-force computation of frame-to-frame distances. Our approach enables robust action matching in challenging situations (such as moving cameras, dynamic backgrounds) and allows automatic alignment of action sequences. Experimental results demonstrate that our approach achieves recognition rates of 92.86% on a large gesture dataset (with dynamic backgrounds), 100% on the Weizmann action dataset, 95.77% on the KTH action dataset, 88% on the UCF sports dataset and 87.27% on the CMU action dataset.**

***Index Terms*— Action recognition, shape-motion prototype tree, hierarchical K-means clustering, joint probability, dynamic time warping.**

I. Introduction

Action recognition is receiving more and more attention in computer vision due to its potential applications such as video surveillance, human-computer interaction, virtual reality and multimedia retrieval. Descriptor matching and classification-based schemes have been common for action recognition. However, for large-scale action retrieval and recognition, where the training database consists of thousands of action videos, such a matching scheme may require tremendous amounts of computation. Recognizing actions viewed against a dynamic varying background is another important challenge. Many studies have been performed on effective feature extraction and categorization methods for robust action recognition. Detailed surveys were reported in [1]–[3].

Feature extraction methods for activity recognition can be roughly classified into four categories: geometry-based [4]–[6], motion-based [7]–[10], appearance-based [4], [11], [12], and space-time feature-based [13]–[28]. The geometry-based approaches recover information about human body configuration, but they often heavily rely on object segmentation and tracking, which is typically difficult and time consuming. The motion-based approaches extract optical flow features for recognition, but they rely on segmentation of the foreground for reducing effects of background flows. The appearance-based approaches use shape and contour information to identify actions, but they are vulnerable to cluttered complex backgrounds. The space-time feature-based approaches either characterize actions using global space-time 3D volumes or more compactly using sparse space-time interest points.

Recently, methods have been introduced, *e.g.* [14], [29]–[35], that combine multiple features to detect and recognize actions. Laptev and Perez [14] used shape and motion cues to detect drinking and smoking actions. Jhuang *et al.* [29] introduced a biologically inspired action recognition system which used a hierarchy of spatial-temporal feature detectors. Liu *et al.* [30] combined quantized vocabularies of local spatial-temporal volumes and spin images. Shet *et al.* [31] combined shape and motion exemplars in a unified probabilistic framework to recognize gestures. Schindler and Gool [32] extracted both form and motion features from an action snippet to model and recognize actions. Niebles and Fei-Fei [33] introduced a hierarchical model and a hybrid use of static shape features and spatial-temporal features for action classification. Ahmad and Lee [34] combined shape and motion flows to classify actions from multi-view image sequences. Mikolajczyk and Uemura [35] extracted a large set of low dimensional local features to learn many vocabulary trees to allow efficient action recognition and perform simultaneous action localization and recognition. For recognizing human actions under view changes, there are some approaches proposed in recent years. Junejo *et al.* [36] proposed a self-similarity based descriptor for view-independent human action recognition. Parameswaran and Chellappa [37] modeled actions in terms of view-invariant canonical body poses and trajectories in 2D invariance space, to represent and recognize human actions from a general viewpoint. Souvenir and Babbs [38] learned the viewpoint manifolds to provide a compact representation of primitive actions for view-invariant action recognition and viewpoint estimation.

Categorization methods are mostly based on machine learning or pattern recognition techniques. Classifiers commonly used for action recognition include NN/K-NN classifiers [7], [10], [12], [17], [18], [39]–[42], Support Vector Machine (SVM) classifiers [13], [15], [20], [21], [29], [32], [43], boosting-based classifiers [9], [14], [23], [26], Hidden Markov Model (HMM) [11], [31], [44], dynamic time warping (DTW) [45], [46] and Hough-voting-based classifiers [47]. For example, the method in [12] used n-Gram models to represent local temporal context and recognized actions based on histogram comparisons. Marszalek *et al.* [43] exploited a context model between scenes and actions,

This work was funded by Army Research Laboratory Robotics Collaborative Technology Alliance program (contract number: DAAD 19-012-0012 ARL-CTA-DJH) and VIRAT program.

Zhuolin Jiang and Larry S. Davis are with the Institute for Advanced Computer Studies, University of Maryland, College Park, MD, 20742 USA E-Mail: {zhuolin—lsd}@umiacs.umd.edu.

Zhe Lin is with Advanced Technology Labs, Adobe Systems Incorporated, San Jose, CA, 95110 USA Email: zlin@adobe.com.

Digital Object Indentifier 10.1109/TPAMI.2011.147

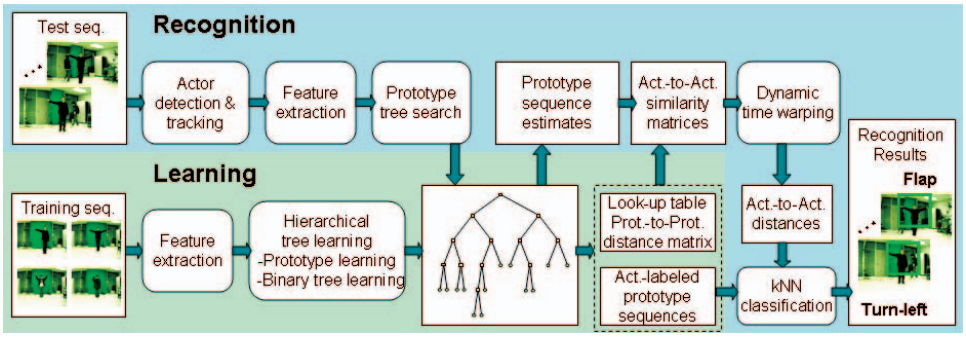


Fig. 1. Overview of our approach.

and integrated it with bag of features and SVM based classifiers to improve action recognition. Ref. [11], [31] incorporated temporal constraints between examplars using HMMs. Additionally, Li *et al.* [40] presented a weighted directed graph to learn and recognize human actions. Lv and Nevatia [41] modeled the dynamics of an action or between actions using an unweighted directed graph. Fanti *et al.* [42] presented a hybrid probabilistic model for human motion recognition and modeled human motion as a triangulated graph. Sminchisescu *et al.* [48] integrated discriminative Conditional Random Field (CRF) and Maximum Entropy Markov Models (MEMM) to recognize human actions.

Descriptor matching and classification-based schemes such as [7], [17] have been common for action recognition. However, for large-scale action recognition problems, where the training database consists of thousands of labeled action videos, such a matching scheme may require tremendous amounts of time for computing similarities or distances between actions. The complexity increases quadratically with respect to the dimension of action (frame) descriptors. Reducing the dimensionality of the descriptors can speedup the computation, but it tends to trade-off with recognition accuracy. In this regard, an efficient action recognition system capable of rapidly retrieving actions from a large database of action videos is highly desirable.

Many previous approaches relied on static cameras or considered only videos with simple backgrounds. The recognition problem becomes very difficult with dynamic backgrounds, because motion features can be greatly affected by background motion flows. Although some preliminary work [14], [15], [22], [35] has been done for recognizing actions in challenging movie scenarios, robustly recognizing actions viewed against a dynamic varying background is still an important challenge.

Motivated by these issues, we introduce an efficient, prototype-based approach for action recognition. Our approach extracts rich information from observations but performs recognition efficiently via tree-based prototype matching and look-up table indexing. It captures correlations between different visual cues (*i.e* shape and motion) by learning action prototypes in a joint feature space. It also ensures global temporal consistency by dynamic sequence alignment. In addition, it has the advantage of tolerating complex dynamic backgrounds due to median-based background motion compensation and probabilistic frame-to-prototype matching.

A. Overview

The block diagram of our approach is shown in Figure 1. During training, action interest regions are first localized and shape-motion descriptors are computed from them. Next, action prototypes are learned as the cluster centers of K-means clustering, and each training sequence is mapped to a sequence of learned prototypes. Finally, a binary prototype tree is constructed via hierarchical K-means clustering [49] using the set of learned action prototypes. In the binary tree, each leaf node corresponds to a prototype. During testing, humans are first detected and tracked using appearance information, and a frame-to-prototype correspondence is established by maximizing a joint probability of the actor location and action prototype. Given rough location of the actor by appearance-based tracking, joint optimization is performed to refine the location of the actor and identify the corresponding prototype. Then, actions are recognized based on dynamic prototype sequence matching. Distances needed for matching are rapidly obtained by look-up table indexing, which is an order of magnitude faster than the brute-force computation of frame-to-frame distances. Our main contributions are three-fold:

- A prototype-based approach is introduced for robustly detecting and matching prototypes, and recognizing actions against dynamic backgrounds.
- Actions are modeled by learning a prototype tree in a joint shape-motion space via hierarchical K-means clustering.
- Frame-to-frame distances are rapidly estimated via fast prototype tree search and look-up table indexing.

This paper is organized as follows. In Sec. II, we introduce our action representation and learning methods in detail. In Sec. III, we first describe a tree based approach for frame-to-prototype matching, and then introduce a prototype-based approach to measure distances between actions. In Sec. IV, we discuss our action localization and tracking methods. In Sec. V, we describe implementation details. In Sec. VI, we present experimental results and analysis. Finally, we conclude the paper and discuss possible future research directions in Sec. VII.

II. Learning Action Representation

For representing and describing actions, an action interest region (bounding box surrounding a person) is specified around a person in each frame of an action learning sequence. Examples of action interest regions are illustrated in Figure 2.

A. Shape-Motion Descriptor

A shape descriptor for an action interest region is represented as a feature vector $D_s = (s_1 ... s_{n_s}) \in \mathcal{R}^{n_s}$ by dividing the action interest region into n_s square grids (or sub-regions) $R_1 ... R_{n_s}$. Given the shape observations from background subtraction (when the camera and background are static) or from appearance likelihood

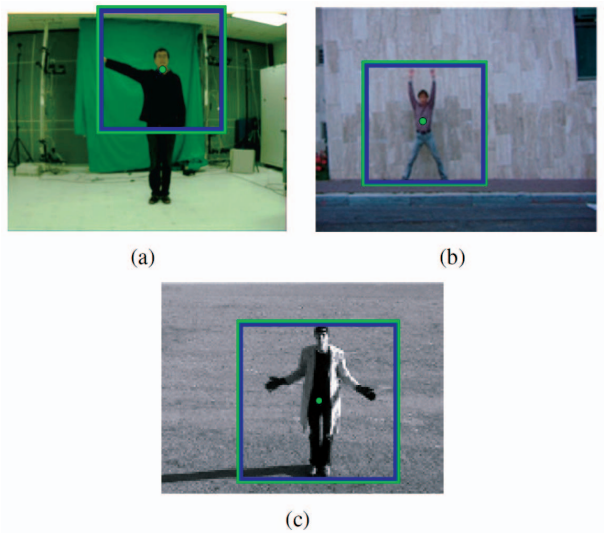


Fig. 2. Examples of action interest regions illustrated for samples from three datasets: (a) Keck gesture dataset, (b) Weizmann action dataset, (c) KTH action dataset.

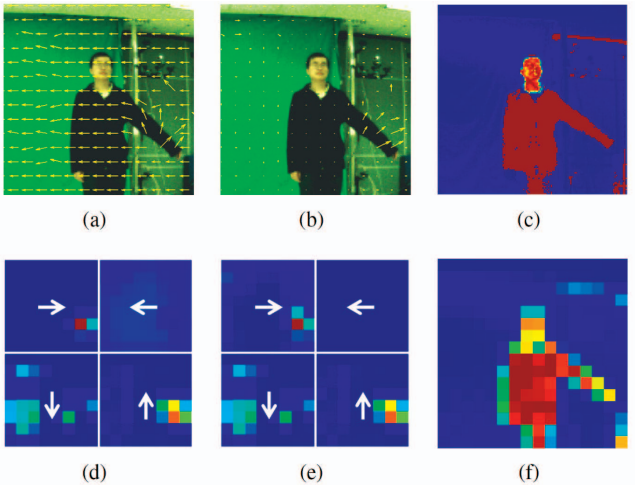


Fig. 3. An example of computing the shape-motion descriptor of a gesture frame with a dynamic background. (a) Raw optical flow field, (b) Compensated optical flow field, (c) Combined appearance likelihood map, (d) Motion descriptor D_m computed from the raw optical flow field, (e) Motion descriptor D_m computed from the compensated optical flow field, (f) Shape descriptor D_s. A motion descriptor is visualized by placing its four channels in a 2×2 grid.

maps (for dynamic cameras and background). In the latter case, a color appearance model is used to assign a probability of person occupancy for each pixel in the bounding box. Then, to form the un-normalized shape feature vector, we simply accumulate the probability of foreground pixel values ($0/1$ in the case of simple background subtraction). Alternatively, in situations in which it might be difficult to extract binary silhouettes or appearance likelihood maps, histogram of oriented gradient (HOG) can be used to encode the shape of each subregion, and then all the histograms can be concatenated to form a raw shape feature vector. The feature vector is L_2 normalized to generate the shape descriptor D_s. An appearance likelihood map and the shape descriptor computed from it are shown in Figure 3(c) and 3(f), respectively. Our method for estimating appearance likelihoods is explained in Sec. IV. The distance between two shape descriptors is computed using the Euclidean distance metric.

A motion descriptor for an action interest region is represented as a n_m-dimensional feature vector $D_m = (QBMF_x^+, QBMF_x^-, QBMF_y^+, QBMF_y^-) \in \mathcal{R}^{n_m}$, where '$QBMF$' refers to quantized, blurred, motion-compensated flow. We compute the motion descriptor D_m based on the robust motion flow feature introduced in [7] as follows. Given an action interest region, its optical flow field is first computed and divided into horizontal and vertical components, F_x and F_y as in [7]. In contrast to [7] which directly use F_x, F_y to compute the motion descriptors, we remove background motion components by subtracting from them the medians of flow fields to obtain median-compensated flow fields. Intuitively, median flows estimate robust statistics of dominant background flows caused by camera movement and moving background objects. Figure 3(a) and 3(b) show an example of motion flow compensation for a gesture frame with a dynamic background. We can see from the figure that this approach not only effectively removes background flows but also corrects foreground flows so that the extracted motion descriptors are more robust against dynamic, varying backgrounds.

The motion-compensated flow fields, MF_x and MF_y, are then half-wave rectified into four non-negative channels $MF_x^+, MF_x^-, MF_y^+, MF_y^-$, and each of them is blurred with a Gaussian kernel to form the low-level motion observations $(BMF_x^+, BMF_x^-, BMF_y^+, BMF_y^-)$ as in [7]. As in computing shape descriptors, we reduce the resolution of the motion observations by averaging them inside uniform grids overlaid on the interest region. The resulting four channel descriptors are L_2 normalized independently as in [7] and concatenated to generate a raw motion descriptor; the L_2 normalization makes the four channel descriptors have equal energy when generating the raw motion descriptor. Finally the raw motion descriptor is L_2 normalized to form the final motion descriptor D_m, and equalize the energy in the shape and motion components of the final joint descriptor. Figure 3(d) and 3(e) visualize the motion descriptors for an example gesture frame with and without motion compensation, respectively.

We concatenate shape and motion descriptors D_s and D_m to form joint shape-motion descriptors: $D_{sm} = (D_s, D_m) \in \mathcal{R}^{n_{sm}}$, where $n_{sm} = n_s + n_m$ is the dimension of the combined descriptor. The distance between two action frames, *i.e.* two shape-motion descriptors, D_{sm}^a and D_{sm}^b, is computed using the Euclidean distance metric. Based on the relative importance of shape and motion cues, we can learn a weighting scheme for the shape and motion components of $D_{sm} = (wD_s, (1-w)D_m)$, where the optimal weight w can be estimated using cross validation by maximizing the recognition rate in the training data [1].

B. Shape-Motion Prototype Tree

Motivated by [11], [12], we represent an action as a set of basic action units. We refer to these action units as action prototypes $\Theta = (\theta_1, \theta_2 ... \theta_K)$. For learning a representative set of action prototypes Θ, we perform clustering on a set of descriptors extracted from the training data.

Given the set of shape-motion descriptors for all frames of the training set, we perform K-means clustering in the joint

[1]The optimal w was estimated as 0.5 from a leave-one-person-out cross validation on the Keck gesture dataset, and we then simply set $w = 0.5$ in all our experiments.

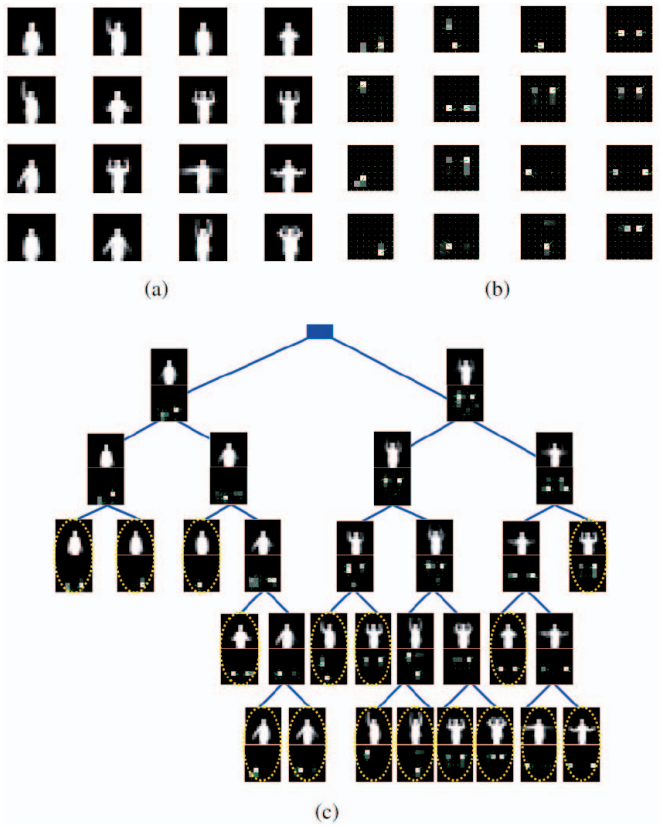


Fig. 4. An example of learning. (a)(b) Visualization of shape and motion components of learned prototypes generated by K-means ($K = 16$). The shape component in (a) is represented by 16×16 grids and the motion component in (b) is represented by four (orientation channels) 8×8 grids. In the motion component, grid intensity indicates motion strength and 'arrow' indicates the dominant motion orientation at that grid, (c) The learned binary prototype tree constructed by hierarchical K-means ($K = 2$). Leaf nodes, represented as yellow ellipses, are action prototypes.

shape-motion space using the Euclidean distance for learning the action prototypes. The cluster centers are then used as the action prototypes. In order to rapidly construct frame-to-prototype correspondence, similar to the online matching of shape exemplar by tree traversal in [50], we build a binary prototype tree over the set of prototypes based on hierarchical K-means clustering [49] ($K = 2$) and traverse the tree to find the nearest neighbor prototype for any given test frame (*i.e.* observation V) and hypothetical actor location, α, during testing.

During tree construction, an initial 2-means clustering process is run on the action prototypes to partition the entire set of prototypes into 2 groups. Then, the same procedure is applied recursively to each group. This process will generate a quantization tree for finding nearest prototypes (leaf nodes). During testing, each query descriptor is passed down the tree by comparing the 2 candidate cluster centers at each level and then choosing the closest one. This matching process continues until it arrives at a leaf node.

Examples of action prototypes and a binary prototype tree are shown in Figure 4. We construct a prototype-to-prototype (pairs of leaf nodes) distance matrix which is computed off-line in the training phase, and use it as a look-up table to speed up the action recognition process.

III. Action Recognition

The action recognition process is divided into two steps: frame-to-prototype matching and prototype-based sequence matching.

A. Tree-based Frame-to-prototype Matching

1) Problem Formulation: Let random variable V be an observation from an image frame, θ be a prototype random variable chosen from the set of K learned shape-motion prototypes $\Theta = (\theta_1, \theta_2 ... \theta_K)$, and $\alpha = (x, y, s)$ denote random variables representing actor location (image position (x, y) and scale s). Then, the frame-to-prototype matching problem is equivalent to maximizing the conditional probability $p(\theta, \alpha|V)$. The conditional probability $p(\theta, \alpha|V)$ is decomposed into a prototype matching term (prototype likelihood given the actor location) and an actor localization term:

$$p(\theta, \alpha|V) = p(\theta|V, \alpha)p(\alpha|V). \quad (1)$$

For a test action sequence of length T with observations $\{V_t\}_{t=1...T}$, a track of the actor's location ($\{\bar{\alpha}_t\}_{t=1...T}$) and location likelihood maps $L(\alpha|V_t), t = 1...T$ are provided by an actor tracker (see Sec. IV). Based on the tracking information, the actor localization term $p(\alpha|V)$ is modeled as follows:

$$p(\alpha|V) \propto \frac{L(\alpha|V) - L_{min}}{L_{max} - L_{min}}, \quad (2)$$

where α is defined over a 3D neighborhood (*i.e.* image position (x, y) and scale s) around $\bar{\alpha}_t$, and L_{min}, L_{max} are the global minimum and maximum limits of $L(\alpha|V)$ respectively. Details of modeling and computing $L(\alpha|V)$ are provided in Sec. IV. An example of a location likelihood map is shown in Figure 6(c).

We model the prototype matching term $p(\theta|V, \alpha)$ as:

$$p(\theta|V, \alpha) \propto e^{-d(D(V,\alpha), D(\theta))}, \quad (3)$$

where d represents the Euclidean distance between the descriptor $D(V, \alpha)$ determined by observation V at location α, and the descriptor $D(\theta)$ of prototype θ.

2) Conditional Probability Maximization: We maximize the conditional probability $p(\theta, \alpha|V)$ by uniformly sampling P points (instances) $\alpha_1, \alpha_2 ... \alpha_P$ around $\bar{\alpha}_t$, which is provided by tracking, and finding the nearest neighbor prototype θ^* for each of the instances α_p. Then, for each given instance α_p, the right-hand-side of Eq. 1 is proportional to the following score function:

$$J(\alpha_p) = e^{-d(D(V_t, \alpha_p), D(\theta^*(\alpha_p)))} \frac{L(\alpha_p|V_t) - L_{min}}{L_{max} - L_{min}}. \quad (4)$$

Finally, the maximum probability prototype is given as $\theta^*(\alpha_p^*)$ where the best location α_p^* is

$$\alpha_p^* = \arg \max_{\{\alpha_p\}_{p=1,2...P}} J(\alpha_p). \quad (5)$$

For efficiently finding the best prototype for any frame and sample actor location, we perform nearest neighbor classification by traversing the learned prototype tree. By only searching the set of learned action prototypes $\theta \in \Theta$ instead of the entire high-dimensional pose space, the method is computationally efficient. Example results of frame-to-prototype matching are shown in Figure 16.

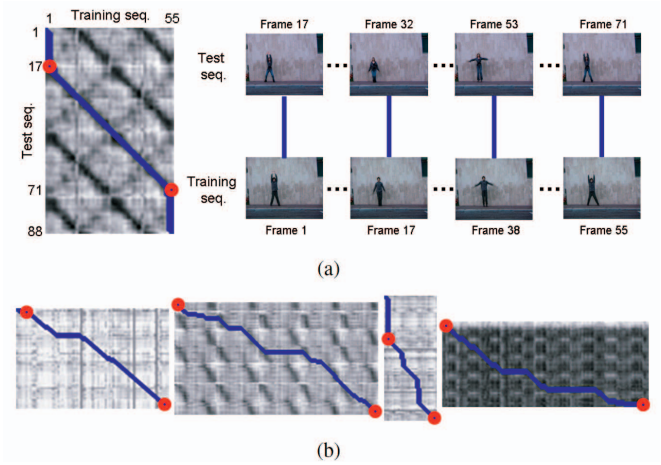


Fig. 5. Examples of sequence matching. Action distance matrices are visualized as gray-scale images and 'blue' alignment-paths obtained by dynamic sequence alignment are overlaid on them. 'red' circles mean start and end points of an optimal alignment path. (a) Same actions performed by different persons. The frame correspondence is shown based on the estimated alignment path. (b) Different actions performed by different persons.

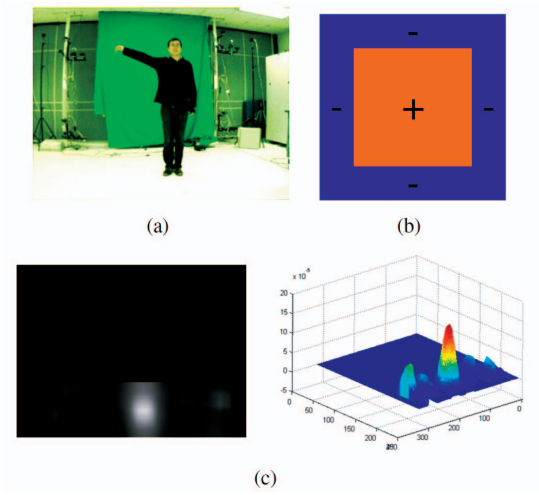


Fig. 6. Location likelihood $L(\alpha|V)$ for a gesture frame. (a) Image, (b) Filter, (c) Location likelihood.

Fig. 7. Examples of action localization result on the Keck Gesture dataset. Our localization method can avoid the influence of a secondary person moving around in the background and a moving camera.

B. Prototype-based Sequence Matching

1) Dynamic Sequence Alignment: There are previous approaches, such as [45], [46], which used dynamic time warping (DTW) to align two action sequences and measure distances between them. Motivated by these approaches, we use the Fast-DTW algorithm [51] to automatically identify optimal matching segments and compute alignment-based distances between two sequences. Let G_x and G_y be two actions of lengths $|X|$ and $|Y|$, with $G_x = x_1, x_2, ..., x_{|X|}$ and $G_y = y_1, y_2, ..., y_{|Y|}$.

Finding the best match between these sequences is equivalent to finding a minimum-cost path through a cost matrix to align the two sequences in time. The cost matrix is constructed by computing the distance $dist(x_i, y_j)$ between every pair of frames x_i and y_j from G_x and G_y. The warping path is defined as $W = w_1, w_2, ..., w_L$, where $w_l = (x_{l,i}, y_{l,j})$, L is the length of the path and satisfies $\max(|X|, |Y|) \leq L \leq |X| + |Y|$. Based on dynamic time warping, the minimum-cost warping path W^* is:

$$W^* = \arg\min_W \left(\sum_{l=1}^{L} dist(x_{l,i}, y_{l,j}) \right), \qquad (6)$$

where $dist(x_{l,i}, y_{l,j})$ is the distance between two frames $x_{l,i}$ and $y_{l,j}$ at the l-th element of the warping path. Distance $dist(x_{l,i}, y_{l,j})$ is computed using the look-up table of prototype distances (i.e. prototype-based approach). Alternatively, the raw feature descriptors can be used for distance computation (i.e. descriptor distance-based approach), but this is more expensive and does not lead to improved classification, as experiments will show.

2) Alignment-based Sequence Matching: After obtaining the minimum-cost path, we estimate the optimal alignment path (a sub-segment of the minimum-cost path) by removing redundant (non-matching) segments at the start and end of the path. Figure 5 shows examples of sequence matching. Based on the optimal alignment path $W^* = \{(x_{l,i}, y_{l,j})\}_{l=l_{start}...l_{end}}$, the distance $Dist(G_x, G_y)$ (*i.e.* action-to-action distance) is given as the average of distances on the alignment-path:

$$Dist(G_x, G_y) = \frac{\sum_{l=l_{start}}^{l_{end}} dist(x_{l,i}, y_{l,j})}{l_{end} - l_{start} + 1}, \qquad (7)$$

We use a k-NN classifier to recognize actions based on action-to-action distances computed using the optimal alignment. We reject non-modeled actions by thresholding action-to-action distances, where the threshold is estimated via cross-validation.

IV. Action Localization

We use a generic human detector[2] [52] or simple foreground segmentation to localize the actor for initialization, and then perform fast kernel-based tracking, such as [53], [54], to track the actor in location and scale space. The fast kernel-based tracker from [53] proved sufficient in our experiments, although the tracker from [54] enables online target appearance modeling. We compute the location likelihood (see below for details) used for tracking and joint probability computation based on background subtraction, appearance likelihood computation or color histogram matching.

Given image observation, V, such as foreground segmentation maps or foreground appearance-likelihood maps, for the Keck, Weizmann, KTH dataset, the location likelihood $L(\alpha|V)$ is computed as the difference of average foreground segmentation maps or appearance-likelihood maps between the inside

[2]The generic human detector is only used for complex cases where actors are viewed by a moving camera and against a dynamic background (such as Keck gesture dataset); For data captured with a static background(such as Weizmann and KTH dataset), we simply use background subtraction to localize actors.

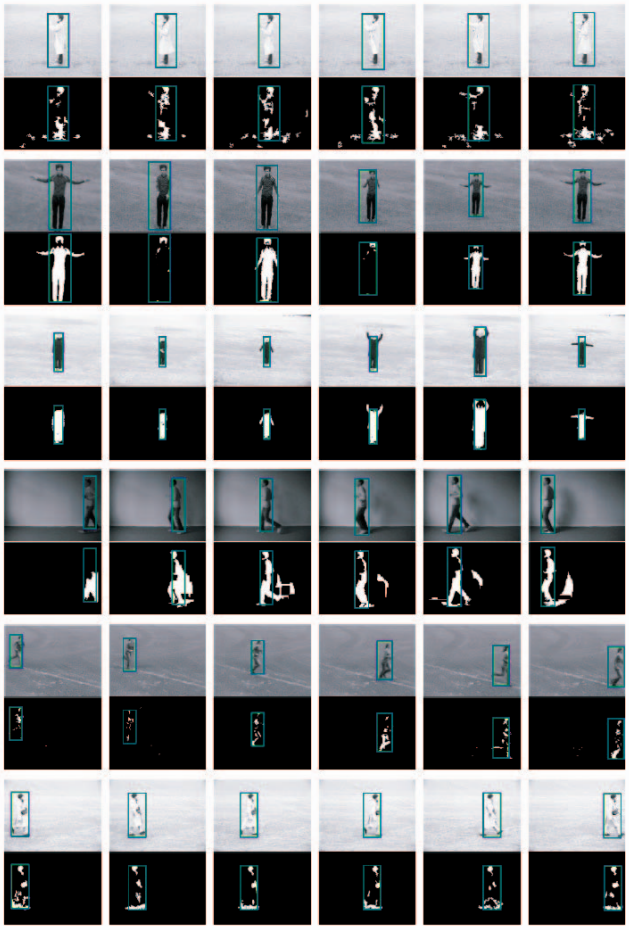


Fig. 8. Examples of action localization and tracking results on the KTH dataset. Our localization method effectively handled influences of shadows, fast camera movements, low contrast, poor background subtraction, and even missing human silhouettes for a short period of time.

and the outside of a rectangle surrounding a hypothetical actor location. It can be visualized as Figure 6(b). Intuitively, this is like a generalized Laplacian operator and favors situations in which the actor matches well inside a detection window, but not coincidentally because the image locally mimics the color distribution of the actor. Figure 6(c) shows an example of the location likelihood map, arbitrarily mapped to the foot location (bottom of the bounding box) after linearly combining all the part based likelihood maps. For the CMU and UCF dataset, $L(\alpha|V)$ is computed as $L(\alpha|V) \propto \exp(-\|h_{cur} - h_{pre}\|^2)$, where h_{cur} and h_{pre} are the color histograms of current frame and previous frame,respectively.

We build an appearance model for each body part in the first frame using kernel density estimation from [55] and use it to compute appearance likelihood maps in subsequent frames. We use a human body part geometric distribution model (learned from training data) to divide the body into three parts: head, torso and legs. Using the torso location as the reference, the geometric model represents the range of locations for the head and legs. The likelihood maps corresponding to body part appearance models are simply summed to generate the combined appearance likelihood map.

Figure 7 shows some examples of localization results on the Keck Gesture Dataset. There is a second person continuously moving in the background and the camera is often moving quickly. Our localization approach eliminates these influences. Figure 8 presents some examples of the localization results on the KTH dataset. From our experiments, we found that our localization method can detect and track people even with very poor silhouettes from background subtraction due to the robustness of the tracking algorithm.

V. Implementation Details

In this section we provide implementation details of our recognition approach. The standard deviation σ of the gaussian kernel for blurring the four non-negative channels $MF_x^+, MF_x^-, MF_y^+, MF_y^-$ is set to 5. Before concatenating a shape descriptor G_s and a motion descriptor D_m to form a joint shape-motion descriptor D_{sm}, the shape descriptor D_s and motion descriptor D_m are L_2 normalized independently. This independent channel normalization scheme is crucial for obtaining high recognition rates. In the training phase, we exclude frames (or descriptors) with no motion or shape information from the K-means clustering input data.

In the testing phase, computation of the shape descriptor D_s is different for different datasets used in our experiments. For the Keck gesture dataset, we use the combined appearance likelihood maps to compute the shape descriptors because background subtraction cannot be used to obtain binary silhouettes. For the Weizmann action dataset and the KTH action dataset, we perform background subtraction for each action sequence and simply compute the shape descriptors using the resulting binary silhouettes. For the UCF sports dataset and CMU action dataset, we use histograms of oriented gradients [52] to compute shape descriptors because the image resolution in CMU dataset is low and it is difficult to divide the human body in parts in UCF dataset.

The shape-motion descriptor (vector) in Keck, Weizmann and KTH dataset is 512-dimensions and consists of a $16 \times 16 = 256$-dimensional shape descriptor and a $8 \times 8 \times 4 = 256$-dimensional motion descriptor. For computing simplified HOG descriptors in the UCF and CMU dataset, we divide an action interest region into $n_h = 6 \times 6$ non-overlapping square cells and accumulate the votes over all pixels into $n_o = 9$ orientation bins for each cell. Hence the shape-motion descriptor is 648-dimensions and consists of a $6 \times 6 \times 9 = 324$-dimensional shape descriptor and $9 \times 9 \times 4 = 324$-dimensional motion descriptor.

The value of K in K-means clustering was set by leave-one-out cross validation during training. The optimal K is assigned to the value which achieves the best overall recognition rate in the training data. It is dependent on the characteristics of individual datasets. For the Keck gesture dataset and Weizmann dataset, varying K from 80 to 180 results in stable recognition rates, while for the KTH dataset, the optimal range of K is from 200 to 300. For the UCF and CMU dataset, the optimal range of K is from 40 to 80.

Our recognition approach classifies an action by matching it to all of the model actions in the training set and then performing k-NN classification. The best value of k is automatically chosen from the range 1 to 50 based on the recognition rates in the training data. For the three test datasets, the range of optimal k is 1 to 10.

In order to determine optimal parameters in our approaches, we perform leave-one-out cross validation approach using the training data. For the Keck, Weizmann and KTH dataset, we perform leave-one-person-out cross validation while we perform

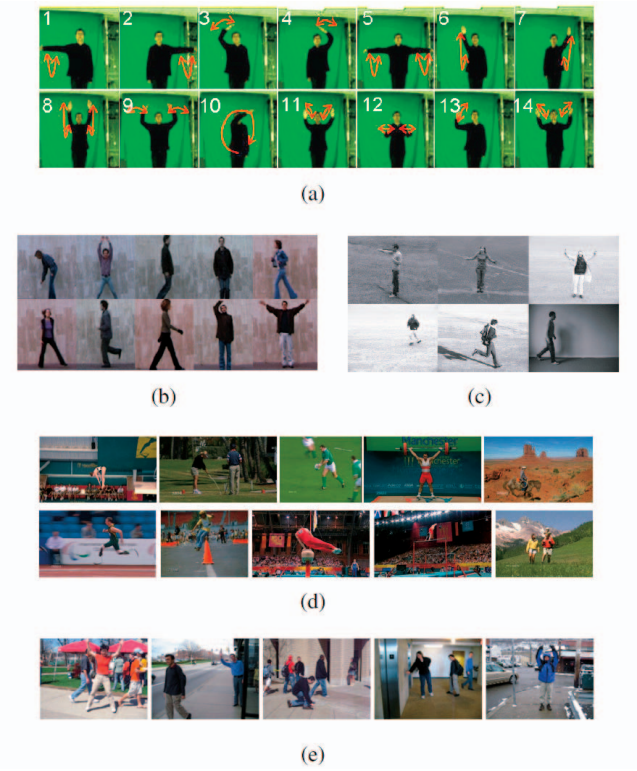


Fig. 9. Datasets. (a) The Keck gesture dataset consisting of 14 different gestures, (b) The Weizmann action dataset consisting of 10 different actions, (c) The KTH action dataset consisting of 6 different actions collected under 4 different scenarios, (d) The UCF sports action dataset consisting of 10 different actions, (e) The CMU dataset consisting of 5 different actions.

leave-one-sequence-out cross validation for the UCF and CMU dataset.

At the local search step for the prototype-based approach, the sampling points are generated around the locations of the actor obtained by tracking. $5 \times 5 = 25$ points are sampled uniformly around the tracked actor location where neighboring points are separated by 4 pixels [3].

For building an appearance model using kernel density estimation for each part, we use the normalized `rgs` color space ($r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, s = \frac{R+G+B}{3}$) and the Gaussian kernel in our experiment. The channel bandwidths are set to $\sigma_r = 0.02$, $\sigma_g = 0.02$ and $\sigma_s = 20$.

VI. Experiments

We evaluated our approach both on a locally collected gesture dataset (Keck Gesture Dataset[4]) and four public action datasets (Weizmann[5], KTH[6], UCF[7] and CMU[8]) in terms of recognition rate and computation time. The time is the average time needed to compute an action-to-action similarity matrix.

[3]We also tested sampling in joint spatial-scale space ($5 \times 5 \times 5$), but there were not much improvement over using the uniform scale ($5 \times 5 \times 1$). Hence we sampled points only in x, y directions.

[4]`http://www.umiacs.umd.edu/~zhuolin/Keckgesturedataset.html`

[5]`http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html`

[6]`http://www.nada.kth.se/cvap/actions/`

[7]`http://server.cs.ucf.edu/~vision/data.html`

[8]`http://www.yanke.org/research.htm`

TABLE I
Feature-based recognition result (leave-one-person-out procedure) on the Keck dataset.

method	motion only	shape only	joint shape and motion
recog. rate (%)	92.86	92.86	95.24

TABLE II
Prototype-based recognition result (leave-one-person-out procedure). Joint motion and shape descriptors are used for the evaluation.

method	recog. rate(%)	avg. time (ms)
descriptor dist.	95.24	154.5
look-up(20 prot.)	90.48	21.8
look-up(40 prot.)	90.48	22.2
look-up(60 prot.)	90.48	22.6
look-up(80 prot.)	90.48	23.2
look-up(100 prot.)	92.86	25.6
look-up(120 prot.)	90.48	22.3
look-up(140 prot.)	92.86	22.7
look-up(160 prot.)	95.24	23.2
look-up(180 prot.)	95.24	25.6
STIP+SVM(20 visual words)	64.29	N/A
STIP+SVM(40 visual words)	83.33	N/A
STIP+SVM(60 visual words)	88.10	N/A
STIP+SVM(80 visual words)	80.95	N/A
STIP+SVM(100 visual words)	80.95	N/A
STIP+SVM(120 visual words)	76.19	N/A
STIP+SVM(140 visual words)	71.43	N/A
STIP+SVM(160 visual words)	69.05	N/A
STIP+SVM(180 visual words)	69.05	N/A

A. Evaluation on the Keck Gesture Dataset

Similar to [31], we collected a dataset consisting of 14 different gesture classes [9]which are a subset of military signals [56]. Figure 9(a) shows sample training frames of the gesture data. This dataset is challenging due to camera motion, moving objects and dynamic backgrounds. These challenges are very common for human-robot interaction.

The gesture dataset is collected using a color camera with 640×480 resolution. Each of the 14 gestures is performed by three people. In each sequence, the same gesture is repeated three times by each person. Hence there are $3 \times 3 \times 14 = 126$ video sequences for training which are captured using a fixed camera with the person viewed against a simple, static background. There are 168 video sequences for testing which are captured from a moving camera and in the presence of background clutter and other moving objects.

1) Gesture Recognition against a Static Background: We evaluated our approach based on a leave-one-person-out experiment using the training data. The confusion matrix is shown in Figure 10. Table I shows that the recognition rate of our approach using the joint shape-motion descriptor outperforms using the shape only feature descriptor or motion only feature descriptor. The recognition rate for the joint shape-motion descriptor is 95.24%.

Table II shows that the prototype-based approach achieves the same recognition rate as using the more computationally demanding descriptor-based approach. When $K = 160$ or 180,

[9]The gesture classes include '1 turn left', '2 turn right', '3 attention left', '4 attention right',' 5 flap', '6 stop left', '7 stop right', '8 stop both', '9 attention both', '10 start', '11 go back', '12 close distance', '13 speed up' and '14 come near'.

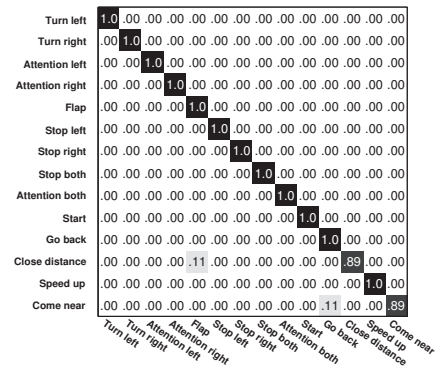


Fig. 10. Confusion matrix for gesture recognition against a static background.

TABLE III
FEATURE-BASED RECOGNITION RESULT USING A MOVING CAMERA VIEWING A DYNAMIC BACKGROUND.

method	motion only	shape only	joint shape and motion
recog. rate (%)	87.5	53.57	92.86

the prototype-based approach obtained a 95.24% recognition rate which is the same as the result of the descriptor-based approach, but the computational cost of the prototype-based approach is much lower than the descriptor-based approach. In addition, both our approaches (*i.e.* descriptor-based or prototype-based approach) outperform the spatio-temporal interest-point feature (STIP) [15] plus bag-of-words and RBF-SVM based approach.

2) Gesture Recognition against a Dynamic Background: This experiment was performed using a moving camera viewing the gesturer against a dynamic background, where one person (who is regarded as the gesturer) performed the specified fourteen gestures in a random order and the other person (who is regarded as 'noise') moved continuously around the gesturer making recognition more challenging. The experimental results using different features are shown in Table III. The joint shape-motion descriptor-based approach outperforms 'shape only' and 'motion only' descriptor-based approaches. The 'shape only' descriptor-based approach obtained poor performance. This is because appearance-based likelihood maps were too noisy, probably due to the strong color similarity between the gesturer and some parts of the background and the distracting person at times. The 'motion only' descriptor-based approach obtained quite good performance due to the robustness of the motion descriptors to small background flows in many gesture frames.

TABLE IV
PROTOTYPE-BASED RECOGNITION PERFORMANCE USING A MOVING CAMERA VIEWING A DYNAMIC BACKGROUND. THE OPTIMAL NUMBER OF PROTOTYPES IS ESTIMATED FROM A LEAVE-ONE-PERSON-OUT CROSS-VALIDATION PROCEDURE ON THE TRAINING SET.

method	recog. rate (%)	avg. time (ms)
descriptor dist.	92.86	96.5
look-up(20 prot.)	55.36	7.2
look-up(40 prot.)	75	7.3
look-up(60 prot.)	76.79	7.4
look-up(80 prot.)	85.71	7.2
look-up(100 prot.)	83.92	7.2
look-up(120 prot.)	89.29	7.2
look-up(140 prot.)	87.5	7.3
look-up(160 prot.)	87.5	7.7
look-up(180 prot.)	91.07	7.8
STIP+SVM(20 visual words)	25.00	N/A
STIP+SVM(40 visual words)	35.71	N/A
STIP+SVM(60 visual words)	41.07	N/A
STIP+SVM(80 visual words)	42.86	N/A
STIP+SVM(100 visual words)	42.86	N/A
STIP+SVM(120 visual words)	42.86	N/A
STIP+SVM(140 visual words)	50.00	N/A
STIP+SVM(160 visual words)	39.29	N/A
STIP+SVM(180 visual words)	48.21	N/A

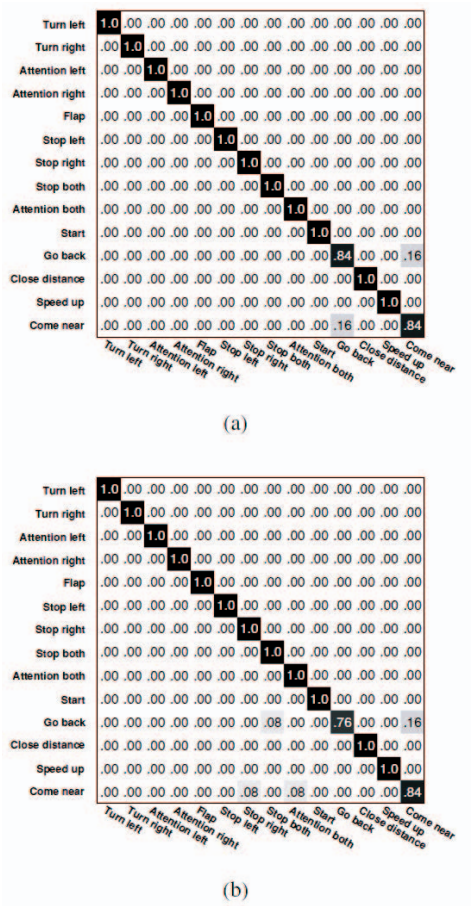


Fig. 11. Confusion matrices for gesture recognition using a moving camera viewing gestures against dynamic backgrounds. (a) Descriptor-based approach, (b) Prototype-based approach ($K = 180$).

As shown in Table IV, the prototype-based approach achieves an accuracy similar to the descriptor-based approach, but is an order of magnitude faster. The recognition results of our approaches are much better than the STIP plus bag-of-words and RBF-SVM based approach. The low recognition rates of the STIP-based approach are mainly caused by the noisy STIP features resulting from significant camera motion. Figures 11(a) and 11(b) show the confusion matrices for both the descriptor-based and the prototype-based approaches.

3) Evaluation with Rejection of Non-modeled Actions: We evaluated the performance of our approaches on rejecting non-modeled actions via thresholding action-to-action distances. Here, 10 action classes (*i.e.* gesture class 1 to 10) are used as the modeled action classes and the remaining 4 action classes (*i.e.* gesture class 11 to 14) are used as the nonmodeled action classes.

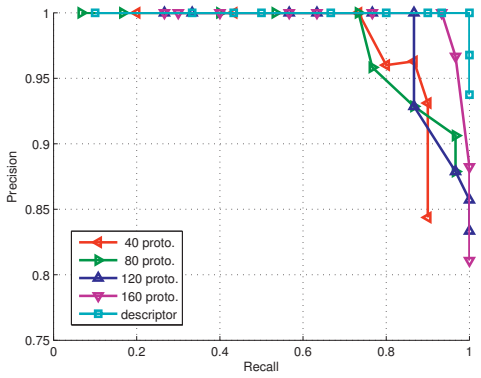


Fig. 12. Precision and recall curves of our approach on the Keck gesture dataset when four gestures are regarded as background gestures to be rejected.

Our recognition approaches are evaluated based on leave-one-person-out experiment using the training data. We generated the precision-recall curves for the descriptor-based and prototype-based approaches, which are shown in Figure 12. The descriptor-based approach can successfully reject the four nonmodeled action classes by thresholding action-to-action distances. The performance of the prototype-based approach is evaluated using different number of prototypes, $K = 40, 80, 120, 160$. As the number of prototypes increases, the prototype-based approach is comparable to the descriptor-based approach.

4) Effect of Local Search Strategy for the Prototype-based Approach: We evaluated the local search step (see sec. III-A) for the prototype-based approach using different number of prototypes, $K = 80, 100, 120, 140, 160, 180$. $5 \times 5 = 25$ points are sampled uniformly around the tracked actor location where the neighboring points are separated by 4 pixels. As shown in Table V, the local search step for joint maximization can improve the prototype-based approach over the tracking-only approach because jointly considering the prototype matching term and the actor localization term can help to better align actor location to the observation and consequently find correct prototypes for individual frames. It is possible to use the local search step for the descriptor-based approach, but the computation become very expensive for this case. This is because, at each sampling point, the descriptor-based approach has to compare each extracted descriptor with all the training descriptors. When the number of training descriptors is far larger than the number of prototypes, it is computationally prohibitive. Therefore we did not use the local search step for the descriptor-based approach in our experiments due to the large number of training descriptors. We did test the local search step for the descriptor-based approach on the set of test videos misclassified by the descriptor-based approach. For this 'hard' set of examples, the local search step did not improve the recognition rate.

B. Evaluation on the Weizmann Action Dataset

The Weizmann dataset [17] contains 90 videos of 10 actions performed by 9 different people. Example frames of this dataset are shown in Figure 9(b). We used background subtraction to obtain a single largest foreground blob and localize the person. We performed leave-one-person-out experiments using nearest neighbor classification to evaluate both the joint shape-motion descriptor and the prototype-based recognition method. Table VI shows comparative results of our joint shape-motion descriptor-based approach with 'shape only' and 'motion only' descriptor-based approaches in terms of recognition rate. The joint shape-motion descriptor-based approach obtained 100% recognition while 'shape only' and 'motion only' descriptor-based approaches obtained much lower recognition rates.

We also evaluated the performance of the prototype-based approach with respect to the number of prototypes K from 20 to 180, and compared these to the descriptor-based approach. As shown in Table VII, the prototype-based approach achieved an average recognition rate of 98.52%, and is robust to the selection of K. The recognition rate reached 100% at $K = 140, 180$ which is the same recognition rate as the descriptor-based approach. Comparing the computation times, the prototype-based approach is almost 26 times faster than the descriptor-based approach but with only a slight $1 - 2\%$ degradation of recognition rate. We have compared the experimental results with state of the art action recognition approaches [9], [12], [17], [18], [29], [30], [32], [33], [36], [57], [58] in Table VII. Our approach achieved the same

TABLE V
LOCAL SEARCH FOR THE PERFORMANCE OF PROTOTYPE-BASED APPROACH ON THE KECK GESTURE DATASET.

method	tracking only (%)	local search (%)
look-up(80 prot.)	83.92	85.71
look-up(100 prot.)	82.14	83.92
look-up(120 prot.)	87.5	89.29
look-up(140 prot.)	83.92	87.5
look-up(160 prot.)	82.14	87.5
look-up(180 prot.)	87.5	91.07

TABLE VI
FEATURE-BASED RECOGNITION RESULT ON THE WEIZMANN DATASET.

method	motion only	shape only	joint shape and motion
recog. rate (%)	88.89	81.11	100

TABLE VII
PROTOTYPE-BASED RECOGNITION PERFORMANCE ON THE WEIZMANN DATASET. THE RESULTS OF [9], [12], [17], [18], [29], [30], [32], [33], [36], [57], [58] ARE COPIED FROM THE ORIGINAL PAPERS.

method	recog. rate (%)	avg. time (ms)
descriptor dist.	100	13.4
look-up(20 prot.)	82.22	0.5
look-up(40 prot.)	91.11	0.6
look-up(60 prot.)	94.44	0.5
look-up(80 prot.)	96.67	0.5
look-up(100 prot.)	97.78	0.5
look-up(120 prot.)	97.78	0.6
look-up(140 prot.)	100	0.5
look-up(160 prot.)	98.89	0.6
look-up(180 prot.)	100	0.5
Natarajan [6]	99.5	N/A
Fathi [9]	100	N/A
Schindler [32]	100	N/A
Thurau [12]	94.40	N/A
Niebles [18]	90	N/A
Ali [57]	92.6	N/A
Jhuang [29]	98.8	N/A
Liu [30]	89.26	N/A
Niebles [33]	72.8	N/A
Wang [58]	97.78	N/A
Blank [17]	99.61	N/A
Junejo [36]	95.3	N/A

TABLE VIII
FEATURE-BASED RECOGNITION RESULT ON THE KTH DATASET. THE UNIT FOR RECOG. RATE IS PERCENTAGE.

	recognition rate (%)			
method	s1	s2	s3	s4
motion only	92.82	78.33	89.39	83.61
shape only	71.95	61.33	53.03	57.36
joint shape and motion	98.83	94	94.78	95.48

TABLE IX
PROTOTYPE-BASED RECOGNITION RESULT FOR INDIVIDUAL SCENARIOS ON THE KTH DATASET. THE RESULTS OF [29], [32], [34] ARE COPIED FROM THE ORIGINAL PAPERS.

	recognition rate (%) / time (ms)			
method	s1	s2	s3	s4
descriptor dist.	98.83 / 15.2	94 / 19.3	94.78 / 14.5	95.48 / 16.7
look-up(200 pr.)	96.83 / 0.9	85.17 / 1.2	92.26 / 0.8	85.79 / 1.1
look-up(220 pr.)	96.33 / 0.9	83.33 / 1.3	92.09 / 0.8	86.79 / 1.1
look-up(240 pr.)	97.50 / 0.9	83.50 / 1.3	91.08 / 0.8	90.30 / 1.1
look-up(260 pr.)	96.33 / 0.9	84.17 / 1.2	90.74 / 0.8	87.96 / 1.1
look-up(280 pr.)	96.83 / 0.9	85.67 / 1.2	90.40 / 0.8	86.79 / 1.1
look-up(300 pr.)	96.66 / 0.9	86.17 / 1.2	90.07 / 0.8	89.97 / 1.1
Schindler [32]	93.0 / N/A	81.1 / N/A	92.1 / N/A	96.7 / N/A
Jhuang [29]	96.0 / N/A	86.1 / N/A	89.8 / N/A	94.8 / N/A
Ahmad [34]	90.17 / N/A	84.83 / N/A	89.83 / N/A	85.67 / N/A

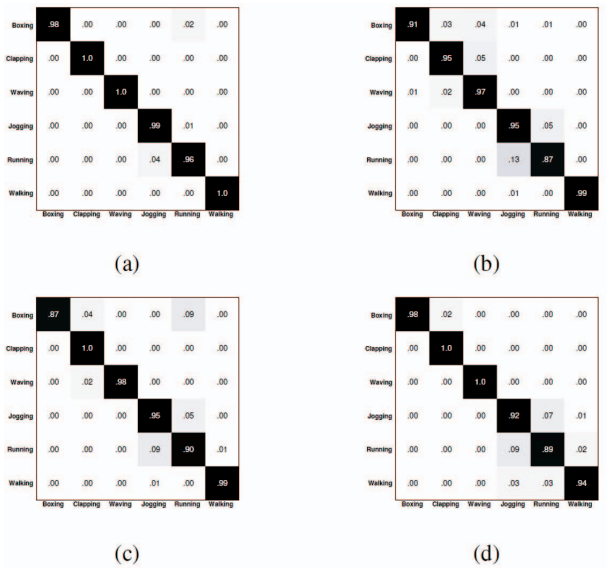


Fig. 13. Confusion matrices for individual scenarios using the descriptor-based approach. (a) s1 scenario, (b) s2 scenario, (c) s3 scenario, (d) s4 scenario.

perfect recognition rate as [9], [32] and outperformed all the other approaches.

C. Evaluation on the KTH Action Dataset

The KTH Action Dataset [13] includes 2391 sequences of six action classes: 'Boxing', 'Clapping', 'Waving', 'Jogging', 'Running' and 'Walking', performed by 25 actors in four scenarios: outdoors (s1), outdoors with scale variation (s2), outdoors with different clothes (s3) and indoors (s4). Example images from this dataset are shown in Figure 9(c). The KTH dataset is known to be more challenging than the Weizmann dataset due to low contrast, frequent shadows and scale variations. Previous work regarded the dataset either as a single large set (all scenarios in one) or as four different sub-datasets (individual scenarios as one sub-dataset trained and tested separately). We perform experiments using both of these settings to compare our results to others reported on this dataset.

In order to obtain reliable foreground segmentation, we combined the codebook-based background subtraction [59] and the local mode seeking tracker [53] to detect and track actors across action sequences. The codebook-based background subtraction algorithm [59] quantizes samples at each pixel into a set of codewords which represent a compressed background model. Foreground pixels are efficiently identified based on nearest neighbor distances to codewords. The algorithm handles moving backgrounds (vegetation under wind load) and multiple changing backgrounds due to factors such as illumination variation. Local mode tracking [53] is the classic mean shift-based tracking algorithm.

In general, leave-one-out cross validation reflects the performance of an approach more reliably because it is more comprehensive than splitting-based evaluation schemes. So, we evaluated our descriptor-based approach and prototype-based approach via the leave-one-person-out evaluation scheme. Table VIII presents recognition results for the four different scenarios using joint shape-motion descriptors, 'shape only', and 'motion only' descriptors. The joint shape-motion descriptors achieved better recognition rates than 'shape only' and 'motion only' descriptors in all four scenarios. Figure 13 shows the confusion matrices for action recognition using the descriptor-based approach. From this figure, we can observe that most recognition errors are produced by 'Jogging' and 'Running' actions. This is reasonable because even humans have difficulty in discriminating the two. Misclassifications between 'Boxing', 'Running', and 'Jogging' are possibly caused by inaccurate detection and localization of actors.

In addition, we evaluated the performance of our prototype-based approach using different number of prototypes, $K = 200, 220, 240, 260, 280, 300$, and compared it to the descriptor-based approach. The experimental results in Table IX show that our prototype-based approach obtains similar recognition rates to the descriptor-based approach in s1 and s3, while there is a little difference of recognition rates between two approaches in s2 and s4, but it is approximately 17 times faster in all

TABLE X
AVERAGE AND ALL-IN-ONE RECOGNITION RESULTS ON THE KTH DATASET. THE RESULTS OF [9], [10], [13], [15], [18]–[21], [23]–[25], [27], [29], [32], [34], [35] ARE COPIED FROM THEIR ORIGINAL PAPERS.

method	evaluation	recognition rate (%)	
		average of all scenarios	all scenarios in one
Our approach	leave one out	95.77	93.43
Schindler [32]	split	90.73	92.7
Ahmad [34]	split	87.63	88.83
Jhuang [29]	split	91.68	N/A
Liu [21]	leave one out	94.15	N/A
Rodriguez [24]	split	88.66	N/A
Mikolajczyk08 [35]	leave one out	93.17	N/A
Niebles [18]	leave one out	N/A	83.33
Dollar [20]	leave one out	N/A	81.17
Schuldt [13]	split	N/A	71.72
Fathi [9]	split	N/A	90.50
Nowozin [23]	split	N/A	87.04
Wang [10]	leave one out	N/A	92.43
Laptev [15]	split	N/A	91.8
Wong [25]	leave one out	N/A	86.7
Ke [19]	leave one out	N/A	80.9
Yuan [27]	split	N/A	93.3

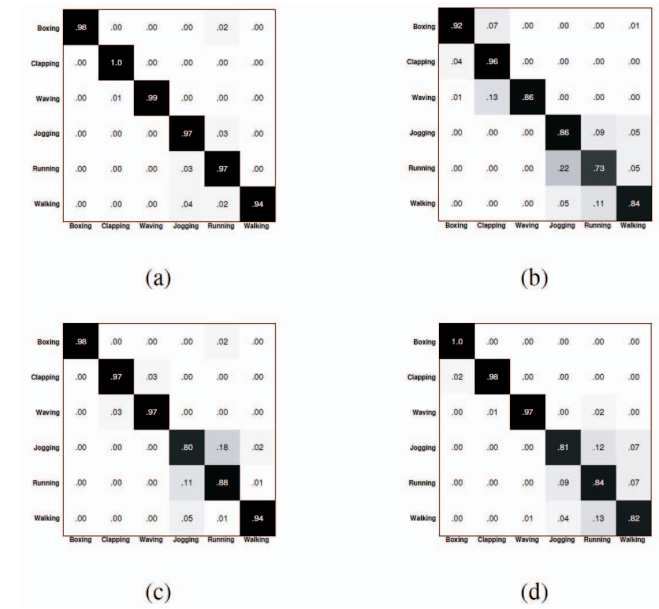


Fig. 14. Confusion matrices for individual scenarios using the prototype-based approach. (a) s1 scenario ($K = 240$), (b) s2 scenario ($K = 300$), (c) s3 scenario ($K = 200$), (d) s4 scenario ($K = 240$).

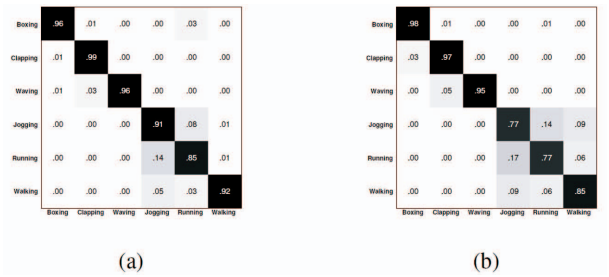


Fig. 15. Confusion matrices for the 'all-in-one' experiments. (a) Descriptor-based approach, (b) Prototype-based approach (K=220).

scenarios. We do not obtain comparable results for s2 and s4 unless we employ many prototypes; however, we then lose the computational advantage of the prototype-based approach. We also compared our results with state of art action recognition approaches [29], [32], [34]. Both versions of our method achieved the highest recognition rates in scenarios s1, s2 and s3, and the results are comparable to these approaches in scenario s4. Moreover, our average recognition rate for all four scenarios is 95.77%. Figure 14 displays confusion matrices for action recognition using the prototype-based approach. Misclassified cases are similar to those of the descriptor-based approach.

Finally, we evaluated our approach in terms of 'average' and 'all-in-one'(all scenarios in a single set) recognition rate. As shown in Table X, our 'average' recognition rate is 95.77% and 'all-in-one' recognition rate is 93.43%. Both of them outperform published results in [9], [10], [13], [15], [18]–[21], [23]–[25], [27], [29], [32], [34], [35] on the KTH dataset. Additionally, our results are comparable to the results reported in [26], [28]. Figure 15 shows confusion matrices of both methods for the 'all-in-one' experiments. Similar to the experiments on the individual scenarios, misclassifications here also mainly occurred amongst 'Jogging', 'Running', and 'Walking'.

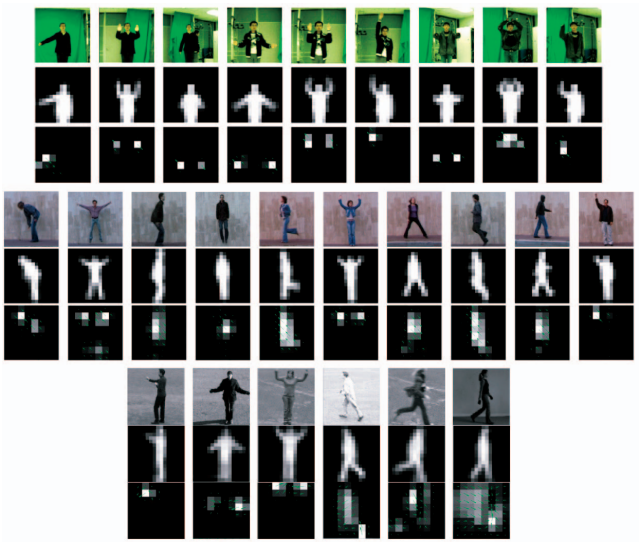


Fig. 16. Examples of frame-to-prototype matching. Top: The Keck gesture dataset. Notice that the background against which the gesturer is viewed changes as we move through the figure, as does the location of the gesturer in the frame. Middle: The Weizmann dataset. Bottom: The KTH dataset.

Figure 16 shows some qualitative results of frame-to-prototype matching for the Keck gesture dataset, Weizmann action datset, and the KTH action dataset. A demo video of gesture recognition is included in the Keck gesture dataset webpage[10].

D. Evaluation on the UCF sports action dataset

The UCF sports action dataset [24] contains 150 video sequences, which are collected from various broadcast sports channels such as BBC and ESPN. This dataset contains videos covering a wide range of scenarios and viewpoints. Example images from this dataset are shown in Figure 9(d). There are ten action classes included in this dataset: diving, swinging-1 (golf), kicking, lifting, horse riding, running, skateboarding, swinging-2 (pommel horse and floor), swinging-3 (high bar) and walking. Here we use the ground truth locations during training, while we use a state-of-art human detector [52] and the fast local mode tracker [53] to obtain the location of actors during testing. We evaluated the descriptor-based approach and our prototype-based approach via the leave-one-sequence-out evaluation scheme. Table XI presents recognition results using joint shape-motion descriptors, 'shape only', and 'motion only' descriptors in terms of recognition rate. The joint shape-motion descriptor-based approach significantly outperforms the shape only or motion only feature-based approach.

We also evaluated the prototype-based approach by varying the number of prototypes K from 10 to 80, and compared these results with the joint descriptor-based approach. The recognition rate reached 86% at $K = 50$ which approximates the results of the descriptor-based approach. We have compared these results with the state of art approaches [24], [28], [47], [60], [61] in table XII. The results of our approach are comparable to them. Figures 17(a) and 17(b) show the confusion matrices for both the descriptor-based and the prototype-based approaches.

[10] http://www.umiacs.umd.edu/~zhuolin/Keckgesturedataset.html

TABLE XI
FEATURE-BASED RECOGNITION RESULT ON THE UCF SPORTS ACTION DATASET.

method	motion only	shape only	joint shape and motion
recog. rate (%)	62	60.67	88

TABLE XII
PROTOTYPE-BASED RECOGNITION PERFORMANCE ON THE UCF SPORTS ACTION DATASET. THE RESULTS OF [24], [28], [47], [60], [61] ARE COPIED FROM THE ORIGINAL PAPERS.

method	recog. rate (%)	avg. time (ms)
descriptor dist.	88	16.6
look-up(10 prot.)	63	0.47
look-up(20 prot.)	69	0.47
look-up(30 prot.)	75	0.47
look-up(40 prot.)	81	0.47
look-up(50 prot.)	86	0.47
look-up(60 prot.)	83	0.47
look-up(70 prot.)	85	0.49
look-up(80 prot.)	83	0.49
Rodriguez [24]	69.2	N/A
Yao [47]	86.6	N/A
Yeffet [60]	79.20	N/A
Wang [61]	85.6	N/A
Kovashka [28]	87.27	N/A

E. Evaluation on the CMU Action Dataset.

This dataset consists of five action classes: 'jumping-jacks', 'one-handed-wave', 'pick-up', 'push-button' and 'two-handed-wave'. Totally there are 48 video sequences for training. Figure 9(e) shows some testing frames from the dataset. The testing data consists of 110 video clips (events) which are down-scaled to 160×120. The dataset is known to be very challenging, because it was captured by a hand-held camera in environments with moving people or vehicles in the background. The experimental results reported on this dataset are only for action detection [22], [62]. Here, we focus our evaluation on recognition performance.

We tested our approach by recognizing the actions in the 110 test videos. Table XIII shows comparative results of joint shape-motion descriptor-based approach with 'shape only' and 'motion only' descriptor-based approaches. The joint shape-motion descriptor achieved much better recognition rates than 'shape only' and 'motion only' descriptors. We also evaluated the recognition rate of our prototype-based approach varying the number of prototypes K from 10 to 80. As shown in Table XIV, the recognition rate reached 84.55% at $K = 60$, which is similar to the descriptor-based approach but is approximately 13 times faster. Figures 18(a) and 18(b) show the confusion matrices on the CMU dataset.

VII. CONCLUSIONS AND DISCUSSIONS

The experimental results demonstrate that our approach is both accurate and efficient for action recognition, even when the action is viewed by a moving camera and against a possibly dynamic background. This good performance is mostly due to the fact that our approach captures correlations between shape and motion by learning action prototypes in the joint feature space, and secondarily because it ensures global temporal consistency by dynamic sequence alignment. The simple median method for background motion compensation does allow us to tolerate complex dynamic backgrounds, although this approach should be replaced by a more principled method of camera motion compensation.

There are several limitations to our approach that should be addressed. First, the frame-to-prototype matching is performed by maximizing a conditional probability, which assumes that frame observations are independent and identically distributed. We did consider and evaluate HMM's for solving the prototype matching problem but did not obtain any improvements on any of the datasets, probably because there is insufficient training data to

TABLE XIII
FEATURE-BASED RECOGNITION RESULT ON THE CMU ACTION DATASET.

method	motion only	shape only	joint shape and motion
recog. rate (%)	49.09	27.07	87.27

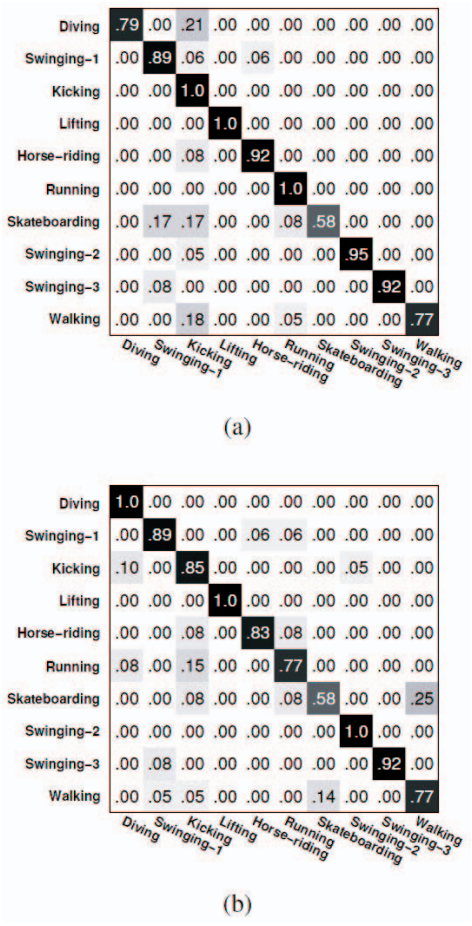


Fig. 17. Confusion matrices for the UCF sports action dataset. (a) Descriptor-based approach, (b) Prototype-based approach (K=50).

TABLE XIV
PROTOTYPE-BASED RECOGNITION PERFORMANCE ON THE CMU ACTION DATASET.

method	recog. rate (%)	avg. time (ms)
descriptor dist.	87.27	3.8
look-up(10 prot.)	57.27	0.3
look-up(20 prot.)	66.36	0.3
look-up(30 prot.)	69.09	0.3
look-up(40 prot.)	72.73	0.3
look-up(50 prot.)	78.18	0.3
look-up(60 prot.)	84.55	0.3
look-up(70 prot.)	76.36	0.3
look-up(80 prot.)	76.36	0.3

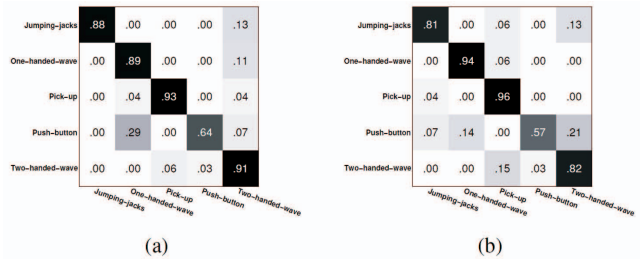


Fig. 18. Confusion matrices for the CMU action dataset. (a) Descriptor-based approach, (b) Prototype-based approach (K=60).

train the model. Second, we have only evaluated the prototype approach with respect to forced choice recognition protocols. More generally, one could use the approach for detection or 'spotting'. Integrating a temporal sliding window [26] or Hough voting scheme [47] into our approach should allow us to build a robust action detection system, which might handle more challenging situations such as the presence of multiple actions performed simultaneously by multiple actors.

REFERENCES

[1] T. B. Moeslund, A. Hilton, and V. Kruger, "A survey of adances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 90–126, 2006.

[2] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recogntion of human activities: A survey," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 8, pp. 1473–1488, 2008.

[3] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976–990, 2010.

[4] H. Li and M. Greenspan, "Multi-scale gesture recognition from time-varying contours," *Proc. IEEE Int'l Conf. Computer Vision*, vol. 1, pp. 236–243, 2005.

[5] Y. Shen and H. Foroosh, "View-invariant action recognition using fundamental ratios," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–6, 2008.

[6] P. Natarajan, V. Singh, and R. Nevatia, "Learning 3d action models from a few 2d videos for view invariant action recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2010.

[7] A. Efros, A. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," *Proc. IEEE Int'l Conf. Computer Vision*, vol. 2, pp. 726–733, 2003.

[8] G. R. Bradski and J. W. Davis, "Motion segmentation and pose recognition with motion history gradients," *Machine Vision and Applications*, vol. 13, pp. 174–184, 2002.

[9] A. Fathi and G. Mori, "Action recognition by learning mid-level motion features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[10] Y. Wang, P. Sabzmeydani, and G. Mori, "Semi-latent dirichlet allocation: A hierarchical model for human action recognition," *Proc. ICCV Workshop on Human Motion Understanding, Modeling, Capture and Animation*, pp. 240–254, 2007.

[11] A. Elgammal, V. Shet, Y. Yacoob, and L. S. Davis, "Learning dynamics for exemplar-based gesture recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 571–578, 2003.

[12] C. Thurau and V. Hlavac, "Pose primitive based human action recognition in videos or still images," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[13] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," *Proc. Int'l Conf. Pattern Recognition*, vol. 3, pp. 32–36, 2004.

[14] I. Laptev and P. Perez, "Retrieving actions in movies," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 1–8, 2007.

[15] I. Laptev, M. Marszalek, C.Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[16] E. Shechtman and M. Irani, "Space-time behavior-based correlation," *IEEE Trans. Pattern Analysis and Machine Intellegence*, vol. 29, no. 11, pp. 2045–2056, 2007.

[17] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *Proc. IEEE Int'l Conf. Computer Vision*, vol. 2, pp. 1395–1402, 2005.

[18] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *Int'l J. Computer Vision*, vol. 79, no. 3, pp. 299–318, 2007.

[19] Y. Ke, R. Sukthankar, and M. Hebert, "Spatio-temporal shape and flow correlation for action recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[20] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," *Proc. Int'l Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 65–72, 2005.

[21] J. Liu and M. Shah, "Learning human actions via information maximization," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[22] Y. Ke, R. Sukthankar, and M. Hebert, "Event detection in crowded videos," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 1–8, 2007.

[23] S. Nowozin, G. Bakir, and K. Tsuda, "Discriminative subsequence mining for action classification," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 1–8, 2007.

[24] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach: A spatio-temporal maximum average correlation height filter for action recogntion," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[25] S. Wong and R. Cipolla, "Extracting spatiotemporal interest points using global information," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 1–8, 2007.

[26] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2009.

[27] J. Yuan, Z. Liu, and Y. Wu, "Discriminative subvolume search for efficient action detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2009.

[28] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2010.

[29] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recognition," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 1–8, 2007.

[30] J. Liu, S. Ali, and M. Shah, "Recognizing human actions using multiple features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[31] V. D. Shet, V. S. N. Prasad, A. Elgammal, Y. Yacoob, and L. S. Davis, "Multi-cue exemplar-based nonparamtric medel for gesture recognition," *Proc. Indian Conf. on Computer Vision, Graphics and Image Processing*, pp. 656–662, 2004.

[32] K. Schindler and L. V. Gool, "Action snippets: How many frames does human action recognition require?" *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[33] J. C. Niebles and L. Fei-Fei, "A hierarchical model of shape and appearance for human action classification," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[34] M. Ahmad and S. Lee, "Human action recognition using shape and clg-motion flow from multi-view image sequences," *Pattern Recognition*, vol. 41, no. 7, pp. 2237–2252, 2008.

[35] K. Mikolajczyk and H. Uemura, "Action recognition with motion-appearance vocabulary forest," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[36] I. Junejo, E. Dexter, I. Laptev, and P. Perez, "View-independent action recognition from temporal self-similarities," *IEEE Trans. Pattern Analysis and Machine Intellegence*, vol. 99, pp. 1–13, 2010.

[37] V. Parameswaran and R. Chellappa, "View invariance for human action recognition," *Int'l J. Computer Vision*, vol. 66, no. 1, pp. 83–101, 2006.

[38] R. Souvenir and J. Babbs, "Learning the viewpoint manifold for action recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

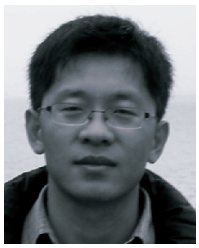
[39] D. Weinland and E. Boyer, "Action recognition using exemplar-based embedding," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–7, 2008.

[40] W. Li, Z. Zhang, and Z. Liu, "Expandable data-driven graphical modeling of human actions based on salient postures," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1499–1510, 2008.

[41] F. Lv and R. Nevatia, "Single view human action recognition using key pose matching and viterbi path searching," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 1–8, 2007.

[42] C. Fanti, L. Zelnik-Manor, and P. Perona, "Hybrid models for human motion recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 1166– 1173, 2005.
[43] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2009.
[44] Q. Shi, L. Wang, L. Cheng, and A. Smola, "Discriminative human action segmentation and recognition using semi-markov model," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2008.
[45] A. Veeraraghavan, R. Chellappa, and A. K. Roy-Chowdhury, "The function space of an activity," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 959–968, 2006.
[46] S. N. Vitaladevuni, V. Kellokumpu, and L. S. Davis, "Action recognition using ballistic dynamics," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2008.
[47] A. Yao, J. Gall, and L. V. Gool, "A hough transform-based voting framework for action recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2010.
[48] C. Sminachisescu, A. Kanaujia, Z. Li, and D. Metaxas, "Conditional models for contextual human motion recognition," *Proc. IEEE Int'l Conf. Computer Vision*, vol. 2, pp. 1808–1815, 2005.
[49] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 2161– 2168, 2006.
[50] D. M. Gavrila and V. Philomin, "Real-time object detection for 'smart' vehicles," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 87–93, 1999.
[51] S. Salvador and P. Chan, "Fastdtw: Toward accurate dynamic time warping in linear time and space," *Proc. KDD Workshop on Mining Temporal and Sequential Data*, pp. 70–80, 2004.
[52] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, 2005.
[53] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. PAMI*, vol. 25, no. 5, pp. 564–577, 2003.
[54] B. Han, D. Comaniciu, Y. Zhu, and L. D. Davis, "Sequential kernel density approximation and its application to real-time visual tracking," *IEEE Trans. PAMI*, vol. 30, no. 7, pp. 1186–1197, 2008.
[55] K. Kim and L. S. Davis, "Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering," *Proc. European Conf. Computer Vision*, pp. 98–109, 2006.
[56] US-ARMY, "Visual signals," *Field Manual*, pp. 21–60, 1987.
[57] S. Ali, A. Basharat, and M. Shah, "Chaotic invariants for human action recognition," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 1–8, 2007.
[58] L. Wang and D. Suter, "Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2007.
[59] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. S. Davis, "Real-time foreground-background segmentation using codebook model," *Real-time Imaging*, vol. 11, no. 3, pp. 167–256, 2005.
[60] L. Yeffet and L. Wolf, "Local trinary patterns for human action recognition," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 1–8, 2009.
[61] H. Wang, M. Ulah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," *Proc. British Machine Vision conference*, pp. 1–11, 2009.
[62] B. Yao and S. Zhu, "Learning deformable action templates from cluttered videos," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 1–8, 2009.

Zhuolin Jiang received the BEng degree in Computer Science from the China University of Petroleum in 2004 and the PhD degree in Computer Science from the South China University of Technology in 2010. He is currently working as a post-doctoral research associate in the Institute for Advanced Computer Studies at the University of Maryland, College Park. His research interests include action detection and recognition, object tracking, video content analysis and retrieval. He is a member of the IEEE.



Zhe Lin received the BEng degree in Automatic Control from the University of Science and Technology of China in 2002 and the MS degree in Electrical Engineering from the Korea Advanced Institute of Science and Technology in 2004, and the PhD degree in Electrical and Computer Engineering from the University of Maryland, College Park, in 2009. He has been a research intern at Microsoft Live Labs Research. He is currently working as a research scientist at the Advanced Technology Labs, Adobe Systems Incorporated, San Jose, California. His research interests include object detection and recognition, content-based image and video retrieval, human motion tracking and activity analysis. He is a member of the IEEE.

Larry S. Davis received the BA degree from Colgate University in 1970 and the MS and PhD degrees in computer science from the University of Maryland in 1974 and 1976, respectively. From 1977 to 1981, he was an assistant professor in the Department of Computer Science at the University of Texas, Austin. He returned to the University of Maryland as an associate professor in 1981. From 1985 to 1994, he was the director of the University of Maryland Institute for Advanced Computer Studies. He is currently a professor in the institute and in the Computer Science Department, as well as the chair of the Computer Science Department. He was named a fellow of the IEEE in 1997. He is known for his research in computer vision and high-performance computing. He has published more than 100 papers in journals and has supervised more than 20 PhD students. He is an associate editor of the International Journal of Computer Vision and an area editor for Computer Models for Image Processing: Image Understanding. He has served as the program or general chair for most of the fields major conferences and workshops, including the Fifth International Conference on Computer Vision, the 2004 Computer Vision and Pattern Recognition Conference, the 11th International Conference on Computer Vision. He is a fellow of the IEEE.