Recognizing Actions by Shape-Motion Prototype Trees

Zhe Lin, Zhuolin Jiang, Larry S. Davis
Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742
{zhelin,zhuolin,lsd}@umiacs.umd.edu

Abstract

A 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, first, an action prototype tree is learned in a joint shape and motion space via hierarchical k-means clustering; then a look-up table of prototype-to-prototype distances is generated. During testing, based on a joint likelihood model of the actor location and action prototype, the actor is tracked while a frame-to-prototype correspondence is established by maximizing the joint likelihood, which is efficiently performed by searching the learned prototype tree; then actions are recognized using dynamic prototype sequence matching. Distance matrices 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 very 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 91.07% *on a large gesture dataset (with dynamic backgrounds),* 100% *on the Weizmann action dataset and* 95.77% *on the KTH action dataset.*

1. Introduction

Action recognition is an important research topic in computer vision. Many studies have been performed on effective feature extraction and categorization methods for robust action recognition.

Feature extraction methods can be roughly classified into four categories: motion-based [6, 8, 29, 30], appearance-based [7, 25], space-time volume-based [2, 10, 13], and space-time interest points and local feature-based [5,12,19, 21,24]. Combining multiple features or visual cues [9,13, 17,18,23] has been shown to be an effective way to improve action recognition performance.

Action categorization methods are mostly based on machine learning or pattern classification techniques as in the object recognition literature. Classifiers commonly used for action recognition include NN/k-NN classifiers [2,6,19,25, 30, 31], Support Vector Machine (SVM) classifiers [5, 9, 12,16,23,24], boosting-based classifiers [8,13,21], Hidden Markov Model (HMM) [7].

Descriptor matching and classification-based schemes such as [2, 6] have been standard 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 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 experimented only on videos with simple backgrounds. However, how can we handle the influences of moving cameras or dynamic backgrounds which is common for human-robot interaction? The recognition problem becomes very difficult with dynamic backgrounds, because motion features can be greatly affected by background motion flows. Although some preliminary work has been done for recognizing actions in challenging movie scenarios [10,12,13,17] or group actions in sports scenarios [14], robustly recognizing actions viewed against a dynamic varying background is still an important challenge.

Motivated by these issues, we introduce a very efficient, prototype-based approach for action recognition which is robust in the presence of moving cameras and dynamic varying backgrounds. 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.

1.1. Related Work

Our approach is closely related to existing approaches representing a human action as a sequence of basic action units [7,25,30,31].

In [25], an action is represented as a set of pose primitives and n-Gram models are used for action matching. Ref. [31] models an action as a set of minimum distances from exemplars to action frames in an examplar-based embedding space. In [30], histograms of motion words are used as action representation and a latent topic model is learned for recognizing actions. These action representation methods are compact and efficient, but might be limited in capturing global temporal consistency between actions because they either use low-order statistics such as histograms and n-Grams, or use a minimum-distance-based representation which does not enforce temporal ordering.

Relaxing temporal constraints [25, 30] makes action representation more invariant to intra-class variation, and consequently might be effective in recognizing small numbers of actions, but when the number of action classes is large, global temporal consistency is very important for action recognition due to small inter-class variability (*i.e.* increased ambiguity between actions). In fact, there have been approaches modeling the global temporal consistency. For example, in [7], an action is modeled as a sequence of exemplars and temporal constraints are imposed by an HMM.

Compared to previous examplar-based approaches, our approach is more accurate due to incorporation of global temporal consistency and frame alignment-based computation of action-to-action distances, and more efficient due to fast prototype tree search and look-up table indexing in the testing phase.

1.2. Overview of Our Approach and Contributions

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 via k-means clustering and set as the resulting cluster centers, and each training sequence is mapped to a sequence of learned prototypes. Finally, a binary prototype tree is constructed via hierarchical k-means clustering [20] 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 likelihood of the actor location and action prototype. The optimal prototype is identified efficiently by repeated use of depth-first search (DFS) on the learned binary tree. Then, actions are recognized based on dynamic prototype sequence matching. Distance matrices used for the 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 four-fold:

- A prototype-based approach is introduced for robustly detecting and matching prototypes, and recognizing actions against dynamic backgrounds.

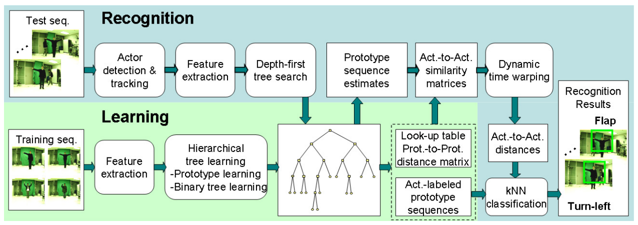


Figure 1. Overview of our approach.

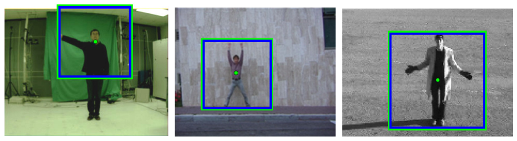


Figure 2. Examples of action interest regions illustrated for samples from three datasets: Gesture, Weizmann and KTH.

- 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.
- A new challenging dataset consisting of 14 gestures is introduced for public use.

2. Action Representation and Learning

For representing and describing actions, an action interest region is first determined around a person in each frame of an action sequence so that the representation is location and scale invariant. Given a human bounding box automatically obtained from background subtraction or human tracking, we define an action interest region as a square region[1] around the localized human bounding box. Examples of action interest regions are illustrated in Figure 2.

2.1. 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 subregions) $R_1 ... R_{n_s}$. For shape, we simply count the number of foreground pixels[2] in each region to form a raw shape feature vector. The feature vector is L_2 normalized to generate the shape descriptor D_s. L_2 normalization has been shown to be effective for concatenated, grid-based image descriptors [4]. In the training phase, shape observations are binary silhouettes obtained by background subtraction; and in the testing phase, the shape observations are either binary silhouettes from background subtraction (when cameras and backgrounds are static) or appearance-based likelihood (or probability) maps (with dynamic cameras and

[1]Its center is determined as a point on the vertical central axis of the human bounding box, and side-length is proportional to the height of the bounding box.

[2]The foreground used here can be computed using either binary silhouettes from background subtraction (under static backgrounds) or appearance-based likelihoods or probabilities (under dynamic backgrounds).

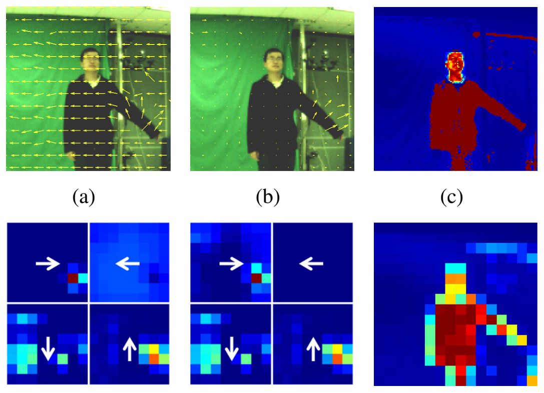


Figure 3. An example of computing the shape-motion descriptor of a gesture frame with a dynamic background. (a) Raw optical flow field, (b) Compensated optical flow field, (c) Combined, part-based appearance likelihood map, (d) Motion descriptor D_m computed from the raw optical flow field, (e) Motion 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.

backgrounds). An appearance-based likelihood map and the shape descriptor computed from it are shown in Figure 3(c) and 3(f), respectively. Our method of estimating appearance-based likelihoods is explained in Sec. 4.

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 [6] 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 [6]. For handling the influences of moving cameras and dynamic backgrounds, we use a median flow-based background motion compensation scheme. In contrast to [6] 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 $MF_x = F_x - median(F_x)$ and $MF_y = F_y - median(F_y)$. Intuitively, median flows estimate robust statistics of dominant background flows caused by camera movements and moving background objects. Figure 3(a) and 3(b) show an example of motion flow compensation for a gesture frame with 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

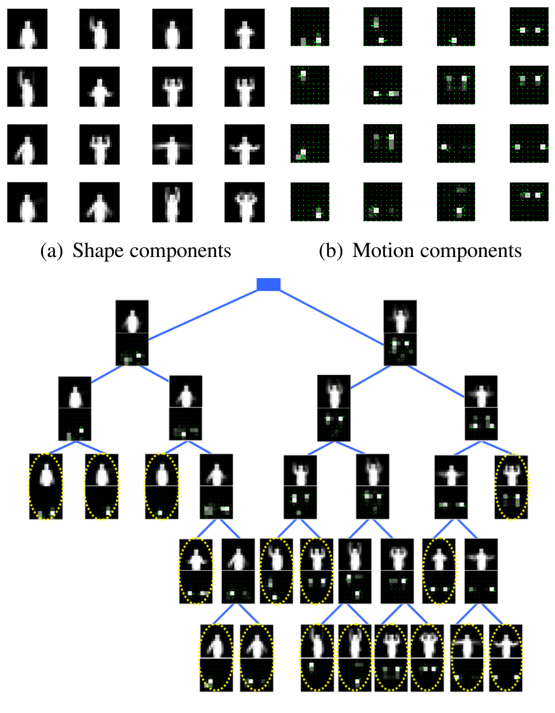


Figure 4. An example of learning. (a)(b) Visualization of shape and motion components of learned prototypes for $k = 16$. The shape component is represented by 16×16 grids and the motion component is represented by four (orientation channels) 8×8 grids. In the motion component, grid intensity indicates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

with a Gaussian kernel to form the low-level motion observations $(BMF_x^+, BMF_x^-, BMF_y^+, BMF_y^-)$ as in [6]. As in computing shape descriptors, we map each channel of the motion observations into low resolution by averaging them inside uniform grids overlaid on the interest region. The resulting four channel descriptors are L_2 normalized independently and L_2 normalized again after concatenation to form the motion descriptor D_m. Figure 3(d) and 3(e) visualize the motion descriptors for an example gesture frame with and without motion compensation, respectively.

We concatenate the shape and motion descriptors D_s and D_m to form a joint shape-motion descriptor[3]. The distance between two shape-motion descriptors is computed using the Euclidean distance metric.

2.2. Shape-Motion Prototype Tree

Motivated by [7, 25], we represent an action as a set of basic action units. We refer to these action units as action

[3]Based on the relative importance of shape and motion cues, we could learn a weighting scheme for the shape and motion components of $D_{sm} = (w_s D_s, w_m D_m)$ (where the weights are chosen such that $w_s^2 + w_m^2 = 1$), where the optimal weights w_s, w_m can be estimated using a validation set by maximizing the recognition rate.

prototypes $\Theta = (\theta_1, \theta_2...\theta_k)$. For learning a representative set of action prototypes Θ, we perform clustering on the 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 shape-motion space using the Euclidean distance for learning the action prototypes. Since both of our shape and motion descriptors are obtained by L_2 normalization, the Euclidean distance metric is reasonable for clustering the joint shape-motion descriptors. The cluster centers are then used as the action prototypes. In order to rapidly construct frame-to-prototype correspondence, we next build a binary prototype tree over the set of prototypes based on the hierarchical k-means clustering algorithm [20] and use DFS to traverse the tree and find the nearest neighbor prototypes for any given test frame (*i.e.* observation V) and hypothetical actor location, α, during testing.

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

3. Action Recognition

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

3.1. Frame-to-prototype matching

3.1.1. 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 location (x, y) and scale s). Then, the frame-to-prototype matching problem is equivalent to maximizing the joint likelihood $p(V, \theta, \alpha)$. Assuming the observation V is given, we decompose the joint likelihood $p(V, \theta, \alpha)$ into an actor localization term and a prototype matching term as follows:

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

For a test action sequence $\{G\}$ of length T with observation $\{V_t\}_{t=1...T}$, 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. 4). Based on the tracking information, the location prior $p(\alpha|V)$ is modeled as follows:

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

where α is defined over a 3D neighborhood around $\bar{\alpha}_t$, and L_{min}, L_{max} are the minimum and maximum limits of $L(\alpha|V)$ in that neighborhood, respectively. An example of the location likelihood map is shown in Figure 6. Details of computing $L(\alpha|V)$, actor localization and tracking are explained in Sec. 4.

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

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

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.

3.1.2. Joint Likelihood Maximization

Given the above model and the observation for frame t, V_t, we evaluate the joint likelihood over θ and α. In practice, we maximize the joint likelihood $p(V, \theta, \alpha)$ by uniformly sampling P points (instances) $\alpha_1, \alpha_2...\alpha_P$ around $\bar{\alpha}_t$ 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 can be rewritten as:

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

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

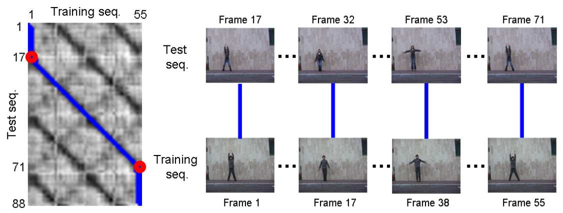
$$\alpha_p^* = \arg \max_{\{\alpha_p\}_{p=1,2...P}} J(\alpha_p). \quad (5)$$

A greedy search algorithm is an alternative method for maximizing J, but it can not guarantee a globally optimal solution. For efficiently finding the best prototype for any frame and sample actor location, we perform nearest neighbor classification by traversing the learned prototype tree using DFS. Different from traditional pose estimation problem, we only search using the set of learned action prototypes $\theta \in \Theta$ instead of the entire high-dimensional pose space, making the method computationally efficient. A further speedup is achieved in the nearest neighbor prototype classification (searching over the prototype space) by DFS on the learned binary tree. Example results of frame-to-prototype matching are shown in Figure 11.

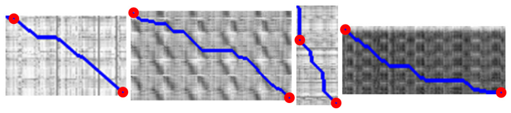
3.2. Prototype-based Sequence Matching

There have been approaches such as [27, 28] which used dynamic time warping (DTW) to align two action sequences and measure distances between them. Motivated by them, we use the FastDTW algorithm [22] to automatically identify optimal matching segments and compute alignment-based distances between two sequences.

Let $G_x = x_1, x_2, ..., x_{|X|}$ and $G_y = y_1, y_2, ..., y_{|Y|}$ be two actions of lengths $|X|$ and $|Y|$, and $W = \{(x_{l,i}, y_{l,j})\}_{l=1...L}$ be the minimum-cost path obtained by DTW. We estimate the optimal alignment path (a subsegment of the minimum-cost path) by removing redundant (non-matching) segments at the start and end of



(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.

Figure 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.

the path. Figure 5 shows examples of sequence matching. Based on the optimal alignment path $W^* = \{(x_{l,i}, y_{l,j})\}_{l=l_{start} \ldots l_{end}}$, the distance $Dist(G_x, G_y)$ (*i.e.* action-to-action distance) is given as the average of distances on the alignment-path:

$$Dist(G_x, G_y) = \frac{\sum_{l=l_{start}}^{l_{end}} dist(x_{l,i}, y_{l,j})}{l_{end} - l_{start} + 1}, \tag{6}$$

where $dist(x_{l,i}, y_{l,j})$ be the distance between two frames which can be computed directly via the Euclidean distance or using the look-up table of prototype-to-prototype distances.[4]

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.

4. Action Localization and Tracking

We use a generic human detector[5] such as [4] or simple foreground segmentation to localize the actor for initialization, and then perform fast local mode seeking such as [3] to track the actor in location and scale space. We compute the location likelihood (see below for details) used for tracking and joint likelihood computation based on foreground likelihood or segmentation maps which are obtained either by background subtraction or appearance-based likelihood computation.

Given image observation, V, such as foreground segmentation maps or foreground appearance-likelihood maps, the location likelihood $L(\alpha|V)$ is computed as the difference of average foreground segmentation maps or appearance-likelihood maps between the inside and the outside of a rectangle surrounding a hypothetical actor location. 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 shows an example of the location likelihood map.

We build a part-based appearance model of the actor in the first frame using kernel density estimation [3, 11] and use it to compute appearance-based likelihood maps in subsequent frames. This is done by dividing the human body into three parts: head, torso, and legs, and an appearance model is built for each part independently. The likelihood maps obtained by these part-based appearance models are linearly combined to generate the appearance-based likelihood map.

Figure 7 shows some examples of actor localization and tracking results on the Gesture and KTH Dataset.

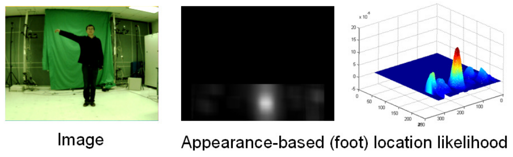


Figure 6. Location likelihood $L(\alpha|V)$ for a gesture frame.

(a) Examples from the Gesture dataset.

(b) Examples from the KTH dataset.

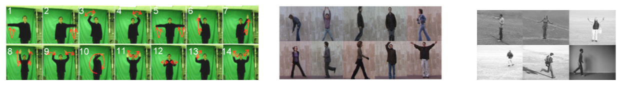
Figure 7. Examples of actor localization and tracking results. Note that our approach effectively handled interruption of a secondary person moving around in the scene on the Gesture data, and the influences of shadows, fast camera movements, low contrast, and poor foreground segmentation on the KTH data.

5. Experiments

We evaluated our approach on a locally collected gesture dataset and two public action datasets in terms of recognition rate and average computation time. The average time is computed as the average of computing an action-to-action similarity matrix. The shape-motion descriptor (vector) is 512-dimensional which consists of a 256-dimensional shape descriptor and a $64 \times 4 = 256$-dimensional motion

[4]Two versions of our approach are: (1) Descriptor distance-based approach directly computes frame-to-frame distances, (2) Prototype-based approach approximates frame-to-frame distances by indexing the look-up table (of prototype-to-prototype distances) precomputed during training.

[5]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 Gesture dataset); For data captured under static background(such as Weizmann and KTH dataset), we simply use background subtraction to localize actors.



(a) Gesture (b) Weizmann (c) KTH

Figure 8. Evaluation datasets.

descriptor. The value of k in k-means clustering was set by cross-validation on a validation set during training. For the Gesture 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.

5.1. Evaluation on the Gesture Dataset

We created a new dataset consisting of 14 different gesture classes[6], which are a subset of the military signals from [26], in our lab environment. Figure 8(a) shows sample training frames from the dataset. The 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.

5.1.1. Recognition against a Static Background

We evaluated our approach based on a leave-one-person-out experiment using the training data. Table 1 shows that the recognition rate of our approach using the joint shape-motion descriptor is 95.24%, which outperforms the 'shape only' descriptor or 'motion only' descriptor.

Table 2 shows that the results of our prototype-based approach for $k = 20 - 180$ in terms of recognition rate and average time. When $k = 180$, the prototype-based approach obtained 95.24% recognition rate, which is the same as the descriptor-based approach, but the computational cost is much lower.

5.1.2. Recognition against a Dynamic Background

This experiment was performed using a moving camera viewing the actor against a dynamic background, where one person (regarded as the actor) performed the specified fourteen gestures in a random order and the other person (regarded as 'noise') moved continuously behind the actor, making recognition more challenging. The results using different features are shown in Table 3. The joint shape-motion descriptor-based approach outperforms both 'shape only' and 'motion only' descriptor-based approaches.

As shown in Table 4, the prototype-based approach achieved an accuracy similar to the descriptor-based approach, but is an order of magnitude faster. Figures 9(a) and 9(b) show the confusion matrices for both the descriptor-based and the prototype-based approaches. Misclassifications are mainly from 'come near' and 'go back', which are visually similar.

[6]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'.

Table 1. Results using different features on the Gesture dataset (static background).

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

Table 2. Prototype-based recognition result using joint shape and motion features (static background).

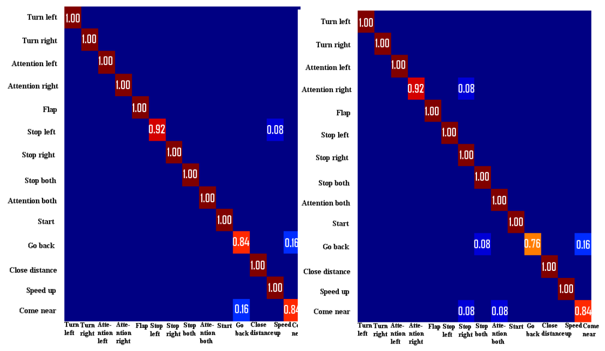
method	recog. rate (%)	avg. time (ms)
descriptor dist.	95.24	154.5
look-up(20 pr.)	90.48	21.8
look-up(60 pr.)	90.48	22.6
look-up(100 pr.)	92.86	25.6
look-up(140 pr.)	92.86	22.7
look-up(180 pr.)	95.24	25.6

Table 3. Results using different features on the Gesture dataset (moving camera, dynamic background).

method	motion only	shape only	joint shape & motion
recog. rate (%)	87.5	53.57	91.07

Table 4. Prototype-based recognition result using joint shape and motion features on the Gesture dataset (moving camera, dynamic background).

method	recog. rate (%)	avg. time (ms)
descriptor dist.	91.07	96.5
look-up(20 pr.)	55.36	7.2
look-up(60 pr.)	76.79	7.4
look-up(100 pr.)	80.36	7.2
look-up(140 pr.)	82.14	7.3
look-up(180 pr.)	89.29	7.8



(a) Descriptor-based (b) Prototype-based ($k = 180$)

Figure 9. Confusion matrices for gesture recognition using a moving camera viewing gestures against dynamic backgrounds.

5.2. Evaluation on the Weizmann Action Dataset

The Weizmann dataset [2] contains 90 videos of 10 actions performed by 9 different people. Example frames of this dataset are shown in Figure 8(b). We performed leave-one-person-out experiments to evaluate our approaches. Table 5 shows comparative results of our joint shape-motion descriptor-based approach with 'shape only' and 'motion

Table 5. Results using different features on the Weizmann dataset.

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

Table 6. Prototype-based recognition result using joint shape and motion features on the Weizmann dataset. The results of [2, 8, 9, 19, 23, 25] are copied from the original papers.

method	recog. rate (%)	avg. time (ms)
descriptor dist.	100	13.4
look-up(20 pr.)	82.22	0.5
look-up(60 pr.)	94.44	0.5
look-up(100 pr.)	97.78	0.5
look-up(140 pr.)	100	0.5
look-up(180 pr.)	100	0.5
Fathi [8]	100	N/A
Schindler [23]	100	N/A
Thurau [25]	94.40	N/A
Niebles [19]	90	N/A
Jhuang [9]	98.8	N/A
Blank [2]	99.61	N/A

only' descriptor-based approach in terms of recognition rate. The 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 6, the recognition rate reached 100% at $k = 140, 180$ which is the same as the descriptor-based approach. Comparing the average 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 [2, 8, 9, 19, 23, 25] in Table 6. Our approach achieved the same perfect recognition rate as [8, 23] and outperformed all the other approaches significantly.

5.3. Evaluation on the KTH Action Dataset

The KTH dataset [24] includes 2391 sequences of six action classes: 'boxing', 'hand clapping', 'hand 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 8(c). Previous work regarded the dataset either as a single large set (all scenarios in one) or as four different datasets (individual scenarios as one dataset trained and tested separately). We perform experiments using both of these settings.

In general, leave-one-out cross validation reflects the performance of an approach more reliably because it is more comprehensive than the splitting-based evaluation schemes [12, 21, 24]. So we evaluated our approaches using leave-one-person-out experiments. Table 7 shows the results of using different features under four different scenarios. As we see from the table, joint shape-motion descriptor achieved better recognition rates than 'shape only' and 'motion only' descriptors in all four scenarios.

In addition, we evaluated the performance of the prototype-based approach for individual scenarios using different numbers of prototypes, $k = 200, 240, 300$, and compared it to the descriptor-based approach. The experimental results in Table 8 show that the prototype-based approach achieves similar recognition rates as the descriptor-based approach, but is approximately 17 times faster. The comparison to state of art approaches [1, 9, 23] shows that our approaches achieved the highest recognition rates under the s1, s2 and s3 scenarios, and the results are comparable to [9, 23] under the s4 scenario.

Table 7. Results using different features on the KTH dataset.

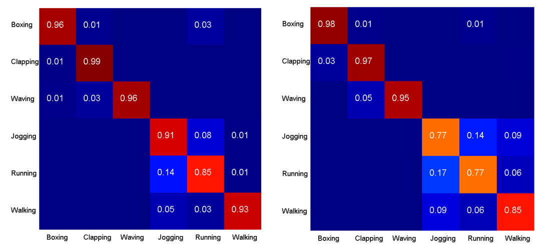
	recognition rate (%)			
method	s1	s2	s3	s4
motion only	92.82	78.33	89.39	83.61
shape only	71.95	61.33	53.03	57.36
joint shape and motion	98.83	94	94.78	95.48

Table 8. Prototype-based recognition result for individual scenarios on the KTH dataset. The results of [1, 9, 23] 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(240 pr.)	97.50 / 0.9	83.50 / 1.3	91.08 / 0.8	90.30 / 1.1
look-up(300 pr.)	96.66 / 0.9	86.17 / 1.2	90.07 / 0.8	89.97 / 1.1
Schindler [23]	93.0 / N/A	81.1 / N/A	92.1 / N/A	96.7 / N/A
Jhuang [9]	96.0 / N/A	86.1 / N/A	89.8 / N/A	94.8 / N/A
Ahmad [1]	90.17 / N/A	84.83 / N/A	89.83 / N/A	85.67 / N/A

Table 9. Average and all-in-one recognition results on the KTH dataset. The results of [1, 5, 8, 9, 16, 19, 21, 23, 24, 30] 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 [23]	split	90.73	92.7
Ahmad [1]	split	87.63	88.83
Jhuang [9]	split	91.68	N/A
Liu [16]	leave one out	94.15	N/A
Niebles [19]	leave one out	N/A	83.33
Dollar [5]	leave one out	N/A	81.17
Schuldt [24]	split	N/A	71.72
Fathi [8]	split	N/A	90.50
Nowozin [21]	split	N/A	87.04
Wang [30]	leave one out	N/A	92.43



(a) Descriptor-based (b) Prototype-based (k=220)

Figure 10. Confusion matrices for the 'all-in-one' experiments.

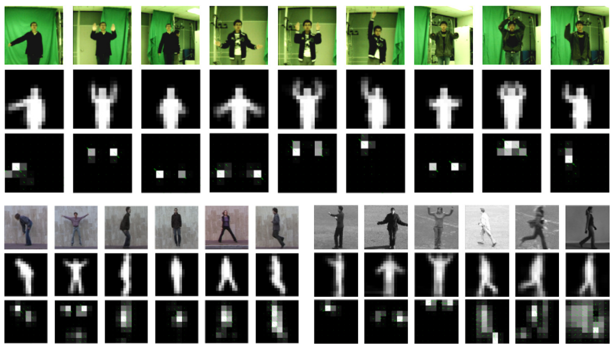


Figure 11. Examples of frame-to-prototype matching. Top: The Gesture dataset. Notice that the background against which the gesturer is viewed changes as we move through the figure, as does the location of the gesturer in the frame. Bottom-Left: The Weizmann dataset. Bottom-Right: The KTH dataset.

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 9, our 'average' recognition rate is 95.77% and 'all-in-one' recognition rate is 93.43%. To the best of our knowledge, both of them outperform published results in [1,5,8,9,16,18,21,23,24,30] on the KTH dataset. These results are also comparable in performance to recent results reported in [15,32]. Figure 10 shows confusion matrices of our approaches from the 'all-in-one' experiments. Misclassifications mainly occurred between 'Jogging', 'Running', and 'Walking', which is reasonable considering their substantial visual similarity.

Figure 11 shows some qualitative results of frame-to-prototype matching for the three datasets. An action recognition demo video is included in the supplemental material.

6. Conclusions

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. Although good overall recognition performance is achieved, our feature representation still has difficulties differentiating some ambiguous classes of actions and dealing with significantly changing backgrounds. A more sophisticated background motion compensation scheme than median compensation would be needed to overcome the effects of severe background motion. Also, discriminative feature analysis between different actions might mitigate the action 'ambiguity' issue to some degree. We are currently exploring these potential extensions for improving our recognition performance.

Acknowledgement

This work was funded, in part, by Army Research Laboratory Robotics Collaborative Technology Alliance program (contract number: DAAD 19-012-0012 ARL-CTA-DJH). We would like to thank Weihua Zhang for helpful discussions.

References

[1] M. Ahmad and S. Lee. Human action recognition using shape and clg-motion flow from multi-view image sequences. *Pattern Recognition*, 41(7):2237–2252, 2008.

[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *ICCV*, pp. 1395-1402, 2005.

[3] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. PAMI*, 25(5):564–577, 2003.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, pp. 886-893, 2005.

[5] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. *VS-PETS*, pp. 65-72, 2005.

[6] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. *ICCV*, pp. 726-733, 2003.

[7] A. Elgammal, V. Shet, Y. Yacoob, and L. S. Davis. Learning dynamics for exemplar-based gesture recognition. *CVPR*, pp. 571-578, 2003.

[8] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. *CVPR*, pp. 1-8, 2008.

[9] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. *ICCV*, pp. 1-8, 2007.

[10] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. *ICCV*, pp. 1-8, 2007.

[11] K. Kim and L. S. Davis. Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. *ECCV*, pp. 98-109, 2006.

[12] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. *CVPR*, pp. 1-8, 2008.

[13] I. Laptev and P. Perez. Retrieving actions in movies. *ICCV*, pp. 1-8, 2007.

[14] R. Li, R. Chellappa, and K. Zhou. Learning multi-modal densities on discriminative temporal interaction manifold for group activity recognition. *CVPR*, pp. 1-8, 2009.

[15] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". *CVPR*, pp. 1-8, 2009.

[16] J. Liu and M. Shah. Learning human actions via information maximization. *CVPR*, pp. 1-8, 2008.

[17] K. Mikolajczyk and H. Uemura. Action recognition with motion-appearance vocabulary forest. *CVPR*, pp. 1-8, 2008.

[18] J. C. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. *CVPR*, pp. 1-8, 2007.

[19] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *Int'l J. Computer Vision*, 79(3):299–318, 2008.

[20] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. *CVPR*, pp. 2161-2168, 2006.

[21] S. Nowozin, G. Bakir, and K. Tsuda. Discriminative subsequence mining for action classification. *ICCV*, pp. 1-8, 2007.

[22] S. Salvador and P. Chan. Fastdtw: Toward accurate dynamic time warping in linear time and space. *KDD Workshop on Mining Temporal and Sequential Data*, pp. 70-80, 2004.

[23] K. Schindler and L. V. Gool. Action snippets: How many frames does human action recognition require? *CVPR*, pp. 1-8, 2008.

[24] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. *ICPR*, pp. 32-36, 2004.

[25] C. Thurau and V. Hlavac. Pose primitive based human action recognition in videos or still images. *CVPR*, pp. 1-8, 2008.

[26] US-ARMY. Visual signals. Field Manual FM 21-60, 1987.

[27] A. Veeraraghavan, R. Chellappa, and A. K. Roy-Chowdhury. The function space of an activity. *CVPR*, pp. 959-968, 2006.

[28] S. N. Vitaladevuni, V. Kellokumpu, and L. S. Davis. Action recognition using ballistic dynamics. *CVPR*, pp. 1-8, 2008.

[29] Y. Wang and G. Mori. Learning a discriminative hidden part model for human action recognition. *NIPS*, pp. 1721-1728, 2008.

[30] Y. Wang, P. Sabzmeydani, and G. Mori. Semi-latent dirichlet allocation: A hierarchical model for human action recognition. *ICCV Workshop on Human Motion*, pp. 240-254, 2007.

[31] D. Weinland and E. Boyer. Action recognition using exemplar-based embedding. *CVPR*, pp. 1-7, 2008.

[32] J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. *CVPR*, pp. 1-8, 2009.