Expandable Data-Driven Graphical Modeling of Human Actions Based on Salient Postures

Wanqing Li, *Member, IEEE*, Zhengyou Zhang, *Fellow, IEEE*, and Zicheng Liu, *Senior Member, IEEE*

***Abstract*—This paper presents a graphical model for learning and recognizing human actions. Specifically, we propose to encode actions in a weighted directed graph, referred to as *action graph*, where nodes of the graph represent salient postures that are used to characterize the actions and are shared by all actions. The weight between two nodes measures the transitional probability between the two postures represented by the two nodes. An action is encoded as *one or multiple paths* in the action graph. The salient postures are modeled using Gaussian mixture models (GMMs). Both the salient postures and action graph are automatically learned from training samples through unsupervised clustering and expectation and maximization (EM) algorithm. The proposed action graph not only performs effective and robust recognition of actions, but it can also be expanded efficiently with new actions. An algorithm is also proposed for adding a new action to a trained action graph without compromising the existing action graph. Extensive experiments on widely used and challenging data sets have verified the performance of the proposed methods, its tolerance to noise and viewpoints, its robustness across different subjects and data sets, as well as the effectiveness of the algorithm for learning new actions.**

***Index Terms*—Action graph, Gaussian mixture model (GMM), human action, salient posture, silhouette, Viterbi path.**

I. Introduction

THE human body is often viewed as an articulated system of rigid links or segments connected by joints and human motion can therefore be considered as a continuous evolution of the spatial configuration of the segments or body posture [1]. Accordingly, effective characterization of the posture (shape) and its dynamics (kinematics) has been central to the research of recognition of human motion. Researchers have so far explored various types of visual information to describe human motion, including motion trajectories [2]–[4], sequences of silhouettes or contours of the human body [5], [6], spatio–temporal salient points [7], hierarchical configuration of body parts [8], [9], such as torso, arms, and legs, and shape volumes [10]–[12]. Among them, silhouettes have gained increasing attention in the recent years due to the advances in background modeling for the extraction of silhouettes, their ability to capture the spatio–temporal characteristics of human motion, and possibly lower complexity of computation. This paper is about the recognition of human motion based on sequences of silhouette images. In particular, we focus on the recognition of *human actions*, the smallest recognizable semantically meaningful motion units, such as *run*, *walk*, and *jump*.

An action recognition system is desired to be independent of the subjects who perform the actions, independent of the speed at which the actions are performed, robust against noisy extraction of silhouettes, scalable to large number of actions, and expandable with new actions. Despite the considerable research in the past few years, such a system is yet to be developed. In this paper, we propose an expandable graphical model of human actions that has the promise to realize such a system. Specifically, we characterize actions with sequences of finite salient postures and propose to model the dynamics or kinematics of the actions using a weighted directed graph, referred to as *action graph*, and to model the salient postures with Gaussian mixture models (GMM). In the action graph, nodes represent salient postures that are shared by the actions and the weight between two nodes measures the transitional probability between the two postures represented by the two nodes. This transitional probability is effectively governed by the kinematics of the human body. An action is encoded in *one or multiple paths* in the action graph. The GMM model of the salient postures provides a compact description of the spatial distribution of the contours belonging to the same salient posture and robust matching to imperfect or noisy silhouettes. Furthermore, the GMM together with the graphical model of actions create a mechanism for a trained system to learn a new action with small number of samples without compromising the existing system. In other words, our model is expandable to incorporate new actions into an existing system without the need for retraining the entire system.

The proposed modeling system is substantially differentiated from and possesses advantages over the previously proposed methods based on postures (or key frames) [13]–[16] and hidden Markov model (HMM) [17]–[19]. First, our model shares postures among the actions and, hence, enables efficient learning from a small number of samples rather than modeling each action with individual HMM, which often requires large number of samples to train. Second, we encode one action into multiple paths (or sequences of salient postures in the graph) to accommodate the variations of the action (e.g., performed by different persons or captured from different viewpoints) as opposed to one sequence of postures (or key frames) as featured in most methods proposed so far. Third, there are no specific beginning or ending postures for any action path. This allows continuous recognition of actions without segmentation. Moreover,

Manuscript received March 01, 2008; revised June 22, 2008. First published September 26, 2008; current version published October 29, 2008. This work was completed at Communication and Collaboration Systems, Microsoft Research, Redmond, WA. This work was supported in part by URC, University of Wollongong, Australia. This paper was recommended by Associate Editor D. Xu.

W. Li is with the Department of Computer Science and Software Engineering, University of Wollongong, Keiraville, NSW 2522, Australia (e-mail: wanqing@uow.edu.au).

Z. Zhang and Z. Liu are with Microsoft Research, Redmond, WA 98052 USA (e-mail: zhang@microsoft.com; zliu@microsoft.com).

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

Digital Object Identifier 10.1109/TCSVT.2008.2005597

cyclic and noncyclic actions can be dealt with in the same way. Fourth, the model facilitates different action decoding schemes (as described in Section III-B) that require different computing resources. From this perspective, our model can be considered as a generalization of the previous works, which usually employ only one of the decoding schemes. Last, our model can be easily scaled to incorporate a large number of actions without adversely impacting on the decoding speed or expanded to new actions without compromising the actions that have been previously learned in the model.

A. Contributions

The major contributions of the paper are as follows.

- We propose an *action graph* to effectively encode the dynamics of actions, in which each node represents a salient posture modeled by GMM. Five decoding schemes are derived. The proposed model offers sharing of knowledge (salient postures) among the actions and flexible decoding schemes. It can be trained with small number of samples and is tolerant to the variations of the actions. More importantly, the graphical model can be easily expanded with new actions.
- A two-stage method is developed to learn the salient postures and action graph from training samples: allocate the salient postures through unsupervised clustering based on joint shape and motion features and construct the action graph.
- A method is proposed for learning a new action with small samples and adding it into the system without the need for retraining the entire system. The algorithm adaptively utilizes the knowledge that has been already learned in the system and has little adverse impact on the system performance in recognizing the previously learned actions.
- Performance evaluation of the proposed graphical model and algorithms is carried out on a relatively large data set currently widely used in the research community not only through the leave-one-sample-out test, but also the leave-one-subject-out and cross-data-set test (i.e., training and test data are from different data sets). The results have verified that the proposed model is able to recognize actions effectively and accurately and it can be easily expanded to new actions. Quantitative and qualitative comparisons of the five decoding schemes are provided.

B. Organization

The rest of this paper is organized as follows. Section II gives a review of previous work related to silhouette-based action recognition. Section III details the proposed graphical model of actions and the five different decoding schemes derived from the model. In Section IV, system learning algorithms are described, which include finding salient postures through automatic clustering, modeling of the salient postures using GMM, and construction of the action graph. In Section V, an algorithm is proposed for adding a new action into an existing trained system by adaptively utilizing the previously learned postures. Kullback–Leibler (KL) divergence is adopted for deciding whether a posture should be shared by the new action. Experimental results on a widely used data set are presented in Section VI to demonstrate the effectiveness of the proposed graphical model for learning and recognition of actions. Comparison among the five different decoding schemes is made. Results on new action learning and the impact on the existing system are also presented and discussed in this section. Finally, this paper is concluded with remarks and future work in Section VII.

II. Related Work

A rich palette of diverse ideas has been proposed during the past few years on the problem of recognition of human actions by employing different types of visual information. A good review can be found in [6] and [20]–[22]. This section presents a review of the work related to silhouette-based action recognition.

Study of the kinematics of human motion suggests that a human action can be divided into a sequence of postures. The sequence is often repeated by the same subject at different times or different subjects with some variations. Methods proposed so far for silhouette-based action recognition differs in the way that the postures are described and the dynamics of the posture sequence is modeled. In general, they fall into two categories based on how they model the dynamics of the actions: *implicit* and *explicit* models. In an implicit model, *action descriptors* are extracted from the action sequences of silhouettes such that the action recognition is turned from a temporal classification problem to a static classification one. The action descriptors are supposed to capture both spatial and temporal characteristics of the actions. For instance, Bobick and Davis [18] proposed to stack the silhouettes into a motion-energy images (MEI) and motion-history images (MHI). Seven Hu moments [23] are extracted from both MEI and MHI to serve as action descriptors. Action recognition is based on the Mahalanobis distance between each moment descriptor of the known actions and the input one. Meng [24] extended the MEI and MHI into a hierarchical form and used a support vector machine (SVM) to recognize the actions. In the method proposed by Chen *et al.* [15], star figure models [25] are fitted to silhouettes to capture the five extremities of the shape that correspond to the arms, legs, and head. GMMs are used to capture the spatial distribution of the five extremities over the period of an action, ignoring the temporal order of the silhouettes in the action sequence. Davis and Yyagi [19] also used GMM to capture the distribution of the moments of the silhouettes of an action sequence.

Recently, Yilmaz and Shah [10] treated a sequence of silhouettes as a spatio–temporal volume and proposed to extract the differential geometric surface properties, i.e., Gaussian curvature and mean curvature, to form a descriptor for each action, known as an action sketch. Gorelick *et al.* [12], [26] extracted space-time features including space–time saliency, action dynamics, shape structure, and orientation by utilizing the properties of the solution to the Poisson equation and employed K-nearest neighborhood (KNN) to classify the actions.

The implicit modeling approach has the advantages that the recognition is relatively simple and is able to handle small number of training samples. However, it usually offers weak encoding of the action dynamics and requires good temporal segmentation before the actions can be recognized. In addition, periodic or cyclic actions have to be dealt with differently [27].

On the other hand, the explicit model follows the concept that an action is composed of a sequence of postures and usually consists of two components: description of the postures and modeling of the dynamics of the postures. Various features have been employed to describe the postures. They include binary masks [28], moments [23], Fourier shape descriptors [16], Kendall's shape description [29], and shape-context [30]. Strategies that have been proposed to model the dynamics include direct sequence matching (DSM) [13], [31], dynamic time warping (DTW) [27], spatio–temporal correlation [13], [32], HHM [17], [19], [33], [34], and their variants such as parameterized HMMs [35], entropic HMMs [36], variable-length HMMs [5], and layered HMMs [37]. Divis and Tyagi [19] used moments to describe shapes of a silhouette and continuous HMM to model the dynamics. In [16], Kellokumpu *et al.* chose Fourier shape descriptors and classified the postures into a finite number of clusters. Discrete HMM are then used to model the dynamics of the actions where the posture clusters are considered to be the discrete symbols emitted from the hidden states. Sminchisescu *et al.* [38] relaxed the HMM assumption of conditional independence of observations given the actions by adopting the conditional random field (CRF) model. Carlsson and Sullivan [39] took an extreme method to describe and match tennis strokes using single key frames. Veerarahavan *et al.* [32] proposed to use autoregressive (AR) model and autoregressive and moving average (ARMA) model to capture the kinematics of the actions. They adopted Kendall's representation of shape as shape features. Recently, Wang and Suter [27] employed locality preserving projection (LPP) to learn a subspace to describe the postures and DTW and temporal Huasdorff distance to classify the actions in the subspace. Colombo *et al.* [31] proposed to find the subspace for each type of actions through principal component analysis (PCA). Wei *et al.* [13] clustered the postures into a set of clusters, known as symbols, based on the shape context. DSM was applied to the symbolized sequences for recognition. Lv and Nevatia [14] took the approach a step further. They modeled the dynamics using an *unweighted directed graph*, referred to as action net, where nodes in the graph represented key postures learned from simulated actions based on the data captured from motion capture devices. The direct links indicate the allowed transition between postures. Each action is represented by *one path* in the action graph. Given an input sequence of silhouettes, the likelihood of each frame belonging to every posture is computed and the input is recognized as the action that gives the maximum accumulated likelihood along the path of the action. Similar to the implicit model, most proposed explicit modeling approaches mentioned above also require segmentation of the actions from the input sequence of silhouettes before an action can be recognized. In addition, the dynamics of the actions are modeled individually and separately (i.e., no connection among actions), such as the conventional HMM-based approach. As a result, they often require a large number of training samples, which can be costly and tedious to obtain.

It has to be pointed out that all methods reviewed above are view dependent. A few attempts have been made to address this issue by including silhouettes from multiple viewpoints or recovering 3-D postures from 2-D image/image sequences. Lv and Nevatia [14] included simulated multiple view silhouettes in each node of their action net. Ahmad and Lee [40] built multiple HMMs for each action, each HMM being for the action observed from a particular viewpoint. Pierobon *et al.* [41] used the 3-D postures recovered from multiple cameras. Green and Guan [34] recovered 3-D postures from monochrome image sequences.

III. Graphical Modeling and Decoding of Actions

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a sequence of n silhouettes and $\Omega = \{\omega_1, \omega_2, \ldots, \omega_M\}$ be the set of M salient postures that constitute actions. The corresponding posture sequence derived from X is denoted as $S = \{s_1, s_2, \ldots, s_n\}$, where $s_t \in \Omega$, $t = 1, 2, \ldots, n$. Assume that $\Psi = \{\psi_1, \psi_2, \ldots, \psi_L\}$ denotes a set of L actions and X is generated from one of the actions. The recognition of the most likely action that generates the observation of X can be formulated as

$$\begin{aligned}\psi^* &= \arg \max_{\psi \in \Psi, S \subset \Omega} p(X, S, \psi) \\ &\propto \arg \max_{\psi \in \Psi, S \subset \Omega} p(\psi) p(S|\psi) p(X|S, \psi) \\ &= \arg \max_{\psi \in \Psi, S \subset \Omega} p(\psi) p(s_1, \ldots, s_n|\psi) \\ &\quad \times p(x_1, \ldots, x_n|s_1, \ldots, s_n, \psi) \end{aligned} \tag{1}$$

where $p(\psi)$ is the prior probability of action ψ, $p(S|\psi)$ is the probability of S given action ψ, and $p(X|S, \psi)$ is the probability of X given S and ψ.

Assume that i) x_t is statistically independent of ψ given S, ii) x_t statistically depends only on s_t, and iii) s_t is independent of the future states and only depends on its previous state s_{t-1}. Then, (1) can be written as

$$\psi^* = \arg \max_{\psi \in \Psi, S \in \Omega} \underbrace{p(\psi) p(s_1, s_2, \ldots, s_n|\psi)}_{\text{specific knowledge}} \underbrace{\prod_{t=1}^{n} p(x_t|s_t)}_{\text{shared knowledge}} \tag{2}$$

where $p(x_t|s_t)$ is the probability for x_t to be generated from state or salient posture s_t. It is referred to as *posture or state model*. Contrary to conventional HMM, we assume the set of postures is known or can be computed from training data, and the first term of (2) is actually a Markov model with known states or visible Markov model (VMM) [42].

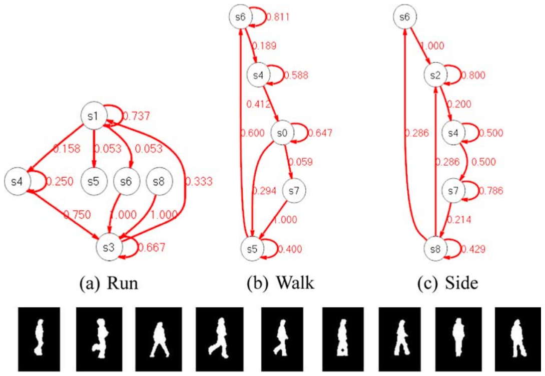
A. Action Graph

Equation (2) can be represented or interpreted as a set of weighted directed graphs G that are built upon the set of postures

$$G = \{\Omega, A, A_1, A_2, \ldots, A_L\} \tag{3}$$

where each posture serves as a node, $A_k = \{p(\omega_j|\omega_i, \psi_k)\}_{i,j=1:M}^{k=1:L}$ is the transitional probability matrix of the k'th action, and $A = \{p(\omega_j|\omega_i)\}_{i,j=1}^{M}$ is the global transitional probability matrix of all actions. We refer to G as an *action graph*.

In an action graph, each action is encoded in one or multiple paths. Fig. 1 shows an action graph for three actions: *run*, *walk*, and *side*. The three actions share nine states/postures whose rep-



(a) Run (b) Walk (c) Side

(d) The nine salient postures.

Fig. 1. Action graph for three actions with nine postures. In each graph, the number next to the links is the transitional probabilities: (a) action *run*; (b) action *walk*; (c) action *side*; and (d) the representative silhouettes of the nine salient postures (left to right), S0 to S8.

resentative silhouettes are shown in Fig. 1(d). Notice that a particular action may only undergo a subset of the postures. For instance, action *run* may go through postures S1, S4, and S3; action *walk* may go through postures S6, S4, S0, S7, and S5; and action *side* may undergo postures S6, S2, S4, S7, and S8. Clearly, the three actions share postures and each action has multiple paths in the action graph. In addition, action paths in the graph are usually cyclic and, therefore, there are no specific beginning and ending postures/states for the action from the recognition point of view.

With the graphical interpretation, a system that follows the model (2) can be described by a quadruplet

$$\Gamma = (\Omega, \Lambda, G, \Psi) \tag{4}$$

where

$$\begin{aligned}\Omega &= \{\omega_1, \omega_2, \dots, \omega_M\}\\ \Lambda &= \{p(x|\omega_1), p(x|\omega_2), \dots, p(x|\omega_M)\}\\ G &= (\Omega, A, A_1, A_2, \dots, A_L)\\ \Psi &= \{\psi_1, \psi_2, \dots, \psi_L\}.\end{aligned} \tag{5}$$

B. Action Decoding

Given a trained system $\Gamma = (\Omega, \Lambda, G, \Psi)$, the action of a sequence $X = \{x_1, x_2, \dots, x_n\}$ is generally decoded in three major steps: 1) find the most likely path in the action graph G that generates X; 2) compute the likelihood of each action $\psi_i \in \boldsymbol{\Psi}$; and 3) decode the action as the one having the maximum likelihood and its likelihood is greater than a threshold, otherwise, the action of X is unknown. Equation (2) offers a number of ways to find the most likely path and estimate the likelihood.

1) Action-Specific Viterbi Decoding: The most obvious one is to search for an action-specific Viterbi decoding (ASVD) in the action graph and calculate the likelihood as follows:

$$L(\psi_i) = \max_{S \in \Omega} p(\psi_i) \prod_{t=1}^{n} p(s_t|s_{t-1}, \psi_i) \prod_{t=1}^{n} p(x_t|s_t) \tag{6}$$

where $L(\psi_i),\ i = 1, 2, \dots, L$, is the likelihood of X belonging to action ψ_i and $p(s_1|s_0, \psi_i) = 1$. X is decoded as action ψ_k if the following condition is met:

$$k = \arg\max_i L(\psi_i), \qquad \text{if } \frac{L(\psi_k)}{\sum_{i=1}^{L} L(\psi_i)} \geq TH_1 \tag{7}$$

where TH_l is a threshold.

Besides the memory requirement for Viterbi search, ASVD decoding method can be computationally expensive when the number of recognizable actions L is large because it searches for the optimal path with respect to every action. A suboptimal, but computationally efficient, decoding scheme is to search for a Viterbi path with respect to the global transitional probability and decode the path with action-specific transitional probabilities. We refer to this method as global Viterbi decoding (GVD).

2) Global Viterbi Decoding: In GVD, the most likely path is the one, $s^* = \{s_1^*, s_2^*, \dots, s_n^*\}$, that satisfies

$$s^* = \arg\max_{s_t \in \Omega} \prod_{t=1}^{n} p(s_t|s_{t-1}) p(x_t|s_t). \tag{8}$$

The likelihood of an action that generates s^* can be computed either using uni-gram or bi-gram model as

$$L(\psi_i) = \arg\max_{\psi \in \Psi} p(\psi) \prod_{t=1}^{n} p(s_t^*|\psi) \quad \text{uni-gram} \tag{9}$$

$$L(\psi_i) = \arg\max_{\psi \in \Psi} p(\psi) \prod_{t=1}^{n} p(s_t^*|s_{t-1}^*, \psi) \quad \text{bi-gram.} \tag{10}$$

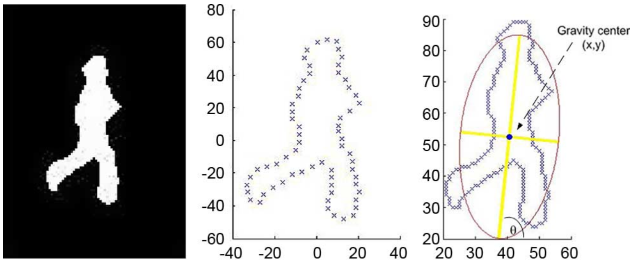
GVD decoding only requires about $1/L$ computational resources of what is required by ASVD.

3) Maximum-Likelihood Decoding (MLD): Both ASVD and GVD require memory to buffer previous frames for Viterbi search. A decoding method that does not require buffering can be devised by searching for the sequence of most likely states/postures rather than the most likely sequence of states (Viterbi path), i.e.,

$$s^* = \arg\max_{s_t \in \Omega} \prod_{t=1}^{n} p(x_t|s_t). \tag{11}$$

The likelihood of an action to generate the path s^* can be calculated using either (9) or (10).

In all, there are five different decoding schemes: 1) action-specific Viterbi decoding (ASVD), 2) uni-gram with global Viterbi decoding (UGVD), 3) bi-gram with global Viterbi decoding (BGVD), 4) uni-gram with maximum-likelihood decoding (UMLD), and 5) bi-gram with maximum-likelihood decoding (BMLD)



(a) (b) (c)

Fig. 2. Feature extraction. (a) A typical silhouette. (b) Normalized and resampled points of the contour; (c) The ellipse fitted to the contour and gravity center.

IV. System Learning

Learning a system Γ from training samples involves the estimation of the posture models Λ and construction of the action graph G. The set of postures Ω can be either derived from the kinematics and kinetics of human motion or automatically learned from the samples. In this paper, we adopted the latter. A simple approach is to cluster the sample silhouettes into M clusters.

A. Posture Models

A posture represents a set of similar poses. Considering the temporal nature of the human motion, we measure the similarity between two poses in terms of joint shape and motion, rather than shape or motion alone as used in most extant work [13], [14].

1) Shape Features and Dissimilarity: There are many shape descriptors available as mentioned in Section II. For the sake of scale invariance and noise tolerance, we choose a set of points on the silhouette contour after scale normalization as the shape descriptor. As shown in Fig. 2(b), the contour of a silhouette is first normalized and then resampled to a small number of points with two purposes: noise and computation reduction.

Let $f_{sp} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_b\}$ and $f'_{sp} = \{\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_b\}$ be the two shapes described by a set of b points on the contours, respectively, then their dissimilarity is defined as

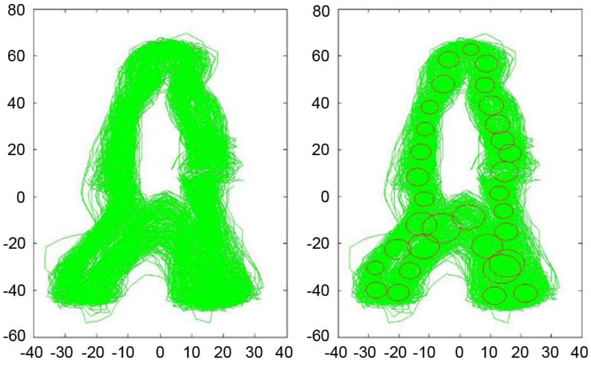
$$d_{sp} = \frac{1}{1+e^{-a(d_h(f_{sp},f'_{sp})-c)}} \tag{12}$$

where $d_h(X,Y)$ is the Hausdorff distance between X and Y; a and c are two constants.

2) Motion Features and Dissimilarity: Motion features include the change of the orientation of the entire body and the local motion of its gravity center. The orientation of the body is estimated by fitting an ellipse into the silhouette shape and measured as the angle (anticlockwise) between the horizontal axis and the major axis of the fitted ellipse as shown in Fig. 2(c).

Let $f_m = (\delta x, \delta y, \delta\theta)$ and $f'_m = (\delta x', \delta y', \delta\theta')$ be the motion feature vector of silhouette x and x', respectively, where $(\delta x, \delta y)$ is the locomotion of the gravity center and $\delta\theta$ is the change of the orientation. The dissimilarity of the x and x' in terms of motion is measured as follows:

$$d_{mt} = \frac{1}{1+e^{-a(\mathrm{corr}(f_m,f'_m)-c)}} \tag{13}$$



(a) (b)

Fig. 3. GMM representation of a salient posture. (a) The contours of a silhouette cluster. (b) The GMM fitted to the contours in (a) (each ellipse represents one Gaussian component).

where $\mathrm{corr}(\cdot,\cdot)$ represents correlation.

3) Unsupervised Clustering: We define the overall dissimilarity of two silhouettes as the product of their motion and shape dissimilarity, i.e.,

$$d = d_{sp} * d_{mt}. \tag{14}$$

Let $D = [d_{ij}]_{i,j=1}^{J}$ be the dissimilarity matrix of all pairs of the J training silhouettes, where D is a $J \times J$ symmetric matrix. The J silhouettes are then clustered into M clusters by employing a pairwise clustering algorithm, which takes the dissimilarity matrix of every pair of samples to be clustered. Choices of such a clustering algorithm include normalized cuts (NCuts) [43] and dominant sets (DSs) [44]. It is found, however, that the property of similarity propagation in both NCuts and DSs works unfavorably in the posture clustering. Therefore, we adopt the traditional non-Euclidean relational fuzzy (NERF) C-means [45]. The NERF C-means is derived from conventional fuzzy C-means specifically for the pairwise clustering where the dissimilarity measurement does not follow Euclidean properties.

4) Estimation of $p(x|s)$: After the clustering, a GMM is fitted using expectation and maximization (EM) algorithm to the shape component of a cluster to represent the spatial distribution of the contours of the silhouettes belonging to the same posture cluster, as shown in Fig. 3, and one Gaussian is fitted to its motion component to obtain a compact representation of the posture models.

Let

$$p_{sp}(y_{sp}|s) = \sum_{k=1}^{C} \pi_{k,s} N(y_{sp}; \mu_{k,s}; \Sigma_{k,s}) \tag{15}$$

$$p_{mt}(y_{mt}|s) = N(y_{mt}; \mu_{mt,s}; \Sigma_{mt,s}) \tag{16}$$

be, respectively, the GMM with C components for shape and Gaussian for motion, where s represents the s salient posture/state or cluster of the silhouettes and $N(\cdot)$ is a Gaussian function; y_{mt} represents the motion feature vector; $\mu_{mt,s}$ is the mean motion vector for salient posture s; $\Sigma_{mt,s}$ is a 3×3 matrix denoting the covariance of the motion features; y_{sp} represents the 2-D coordinates of a point on the contours of silhouettes; $\mu_{k,s}$ is the center of the kth Gaussian for state s; $\Sigma_{k,s}$ is a

2×2 covariance matrix; and π_k, s is the mixture proportion $\sum_{k=1}^{C} \pi_{k,s} = 1$.

The posture model can then be defined as

$$p(x|s) = p_{mt}(y_{mt}|s) \prod_{i=1}^{b} p_{sp}(y_{sp}^{i}|s) \tag{17}$$

where x is a silhouette and y_{mt} and y_{sp}^{i} represent, respectively, the motion feature and the i'th point on the resampled contour of x.

B. Action Graph

The action graph is built by linking the postures with their transitional probabilities. We estimate the action-specific and global transitional probability matrices $\{A_i\}_{i=1}^{L}$ and A from the training samples given the statistical independence assumptions introduced in Section III and the posture models

$$p(\omega_i|\omega_j) = \frac{\sum_{t=1}^{J} p(\omega_i|x_t) p(\omega_j|x_{t-1})}{\sum_{t=1}^{J} p(\omega_i|x_t)} \tag{18}$$

$$p(\omega_i|\omega_j, \psi_l) = \frac{\sum_{t=1}^{J_l} p(\omega_i|x_t, \psi_l) p(\omega_j|x_{t-1}, \psi_l)}{\sum_{t=1}^{J_l} p(\omega_i|x_t, \psi_l)} \tag{19}$$

where J is the total number of training silhouettes for all actions and J_l is the number of silhouettes contained in the training samples for action ψ_l. The marginalization of $p(\omega_i|\omega_j)$ and $p(\omega_i|\omega_j, \psi_l)$ gives the estimation of $p(\omega_i)$ and $p(\omega_i|\psi_l)$, respectively.

V. Learning a new Action

Obtaining training data of human actions can be costly and tedious [46], [47]. On the other hand, to retain all training data for retraining in the future would be impractical. It is desirable that a trained system, whenever needed, be expanded with new actions without a need for retraining the entire system. Our representation of the action graph and GMM postures enables this expansion. In this section, we present an algorithm to add a new action to an existing system without compromising the recognition of the previous learned actions.

Let $\Gamma = \{\Omega, G, \Lambda, \Psi\}$ be the system that has been trained for L actions. Assume that a new action ψ_{L+1} is required to be added to Γ. The new action ψ_{L+1} has K training sequences of silhouettes $\{y_t^k\}_{t=1:T_k}^{k=1:K}$, where T_k is the number of frames in the k'th training sequence. When the new action is included into the system, it is, in general, expected that both the action graph and postures need to be updated. To minimize the impact to the existing system and also considering that K is usually small in practice, it is reasonable and probably necessary to limit the update to the insertion of new postures required to describe ψ_{L+1}, modification of A, and insertion of A_{L+1}. Let us consider the following two cases.

- Ω has all the postures that are required to describe action ψ_{L+1}. In this case, postures should be shared and only new paths are required to be inserted into the action graph by updating A and A_{L+1}.
- Ω does not have all postures that are needed to describe action ψ_{L+1}. Therefore, new postures have to be created for ψ_{L+1} and the action graph needs to be expanded by updating A and A_{L+1}.

As seen, the key issue is how to judge whether new postures are required and how to create them if required. A simple approach is to find the salient postures for the new action first and, then, decide whether these postures have already been learned in the system by comparing the new postures to those residing in the existing system. Following this idea, we propose an algorithm for adding the new action ψ_{L+1} to Γ.

1) Clustering the samples of the new action into m postures, $\Omega' = \{\omega'_1, \omega'_2, \ldots, \omega'_m\}$, whose prototypes are $\Lambda' = \{p'(x|\omega'_1), p'(x|\omega'_2), \ldots, p'(x|\omega'_m)\}$ using the same method as the one used in the system learning.
2) For each new posture, $\omega'_i, i = 1, \ldots, m$, compare it with each posture in Ω. If ω'_i is *similar* to any one of the posture in Ω, then discard ω'_i. Otherwise, keep it in Ω'.
3) Set Ω^{new} as the union of Ω and Ω' and let Λ^{new} be the posture models of Ω^{new}.
4) Estimate the transitional probabilities A_{L+1} and A' from the K training samples for ψ_{L+1} based on Λ^{new}. Update A as follows:

$$A^{\text{new}} = A + \beta * A' \tag{20}$$

where $\beta \in (0, 1)$ is a weighting factor controlling the contribution of the new action samples to the global transition. Because the number of training samples K for the new action would be small compared to the number of samples used to train A, A' is often much less reliable than A, therefore, we limit the contribution of A' to the final global transitional probabilities by the factor of β, which should reflect the ratio of size of the new training samples to the size of the samples used to estimate A.

A. Similarity Between Postures

Because postures are modeled by a single Gaussian for motion and a GMM for shape, the similarity between two postures can be measured by KL divergence. We adopt the variational estimation of KL divergence recently proposed by Hershey and Olsen [48].

The KL divergences for motion and shape between posture s and s' are, respectively

$$\begin{aligned}
&\mathrm{KL}_{mt}(p_{mt}||p'_{mt}) \\
&\quad = D(N(y_{mt}; \mu_{mt}; \Sigma_{mt})||N(y_{mt}; \mu'_{mt}, \Sigma'_{mt})) \\
&\quad = \frac{1}{2}\Big[\log \frac{\det(\Sigma'_{mt})}{\det(\Sigma_{mt})} + \mathrm{tr}(\Sigma_{mt}^{-1}\Sigma_{mt}) - d \\
&\qquad\qquad + (\mu_{mt} - \mu'_{mt})^T \Sigma'^{-1}_{mt} (\mu_{mt} - \mu'_{mt})\Big]
\end{aligned}$$

where $\mathrm{KL}(p\|p')$ represents the KL divergence between the distribution p and p', and $D(N\|N')$ is the KL divergence between two Gaussians of dimension d, N, and N'

$$\mathrm{KL}_{sp}(p_{sp}\|p'_{sp}) = \sum_a \pi_a \log \frac{\sum_{a'} \pi_{a'} e^{-D(N_a\|N_{a'})}}{\sum_b \pi_b e^{-D(N_a\|N_b)}}. \tag{21}$$

s' will be discarded if the following condition is met:

$$(\mathrm{KL}_{mt} - \bar{\mathrm{KL}}_{mt}) < \alpha_{sp} * \sigma_{\mathrm{KL}_{mt}}$$

or

$$(\mathrm{KL}_{sp} - \bar{\mathrm{KL}}_{sp}) < \alpha_{mt} * \sigma_{\mathrm{KL}_{sp}} \tag{22}$$

where $\bar{\mathrm{KL}}_{mt}$, $\sigma_{\mathrm{KL}_{mt}}$, $\bar{\mathrm{KL}}_{sp}$, and $\sigma_{\mathrm{KL}_{sp}}$ are the means and standard deviation of the KL divergences of all pairs of postures in the system before updating, and $\alpha_{sp} \in (0, 1]$ and $\alpha_{mt} \in (0, 1]$ are constants.

B. Estimation of A_{L+1}

Estimation of A_{L+1} is critical to the recognition of the new action ψ_{L+1}. When the number of training samples is small, it is likely that the training samples only capture a small proportion of possible posture transition that are associated with the new action. This phenomenon is called "rare events" in learning grammars in speech recognition. Often, A_{L+1} will not be a reliable estimation of the true transition. Research in speech [49], [50] has suggested many strategies, known as smoothing, to compensate the small number of samples. Here, we adopt a simple and linear model to smooth A_{L+1}

$$p(s_i|s_j, \psi_{L+1}) = (1 - e^{-p(s_j, s_i, \psi_{L+1})}) \frac{p(s_j, s_i, \psi_{L+1})}{p(s_j, \psi_{L+1})} + e^{-p(s_i, s_j, \psi_{L+1})} p(s_i, \psi_{L+1}) \tag{23}$$

where $s_i,\ s_j \in \Omega^{\text{new}}$ and $p(s_j, s_i, \psi)$ is the joint probability of a frame being in posture s_j followed by another frame being in posture s_i. Equation (23) is actually an interpolation of bi-gram and uni-gram transitional probabilities. For unseen events, the transitional probability is set to be the uni-gram probability of the second posture of the bi-gram. Giving too much weight to uni-gram probability may result in faulty estimation if s_i is very frequent. Therefore, the value of the weight decreases exponentially with the number of bi-gram observations.

VI. Experimental Results

A. Data Sets

We evaluated our model on the most widely used data set created by Blank *et al.* [26]. The data set contains 93 low-resolution video (188 × 144, 25 fps) sequences for ten actions. These ten actions are *run, walk, wave with one hand, wave with two hands, galloping sideway, jumping-in-place, jumping, jumping jack, bend, and skip*. Nine subjects played each action once (with an exception that one subject played three actions twice). Silhouettes were obtained using simple background subtraction in color space. Global motion was removed by fitting quadratic function to the trajectory of the gravity centers. This data set is currently the most realistic and challenging one publicly available compared to those employed in other papers (e.g., [51]). Some silhouettes are noisy as shown in Fig. 4. Action *walk* and *jumping-in-place* appears very similar to action *galloping sideway* and *jumping*, respectively, when the global motion is removed from the silhouettes.

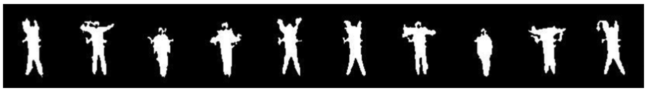


Fig. 4. Examples of noisy silhouettes.

B. Experimental Setup

As adopted in most previous works [3], [12], [26], [27] using the same data set, we conducted leave-one-sample-out test to verify the overall performance of the proposed model. To evaluate its robustness against various factors including the dependence on subjects, viewpoints, action speed and styles, and video capturing environment, we also conducted the following experiments:

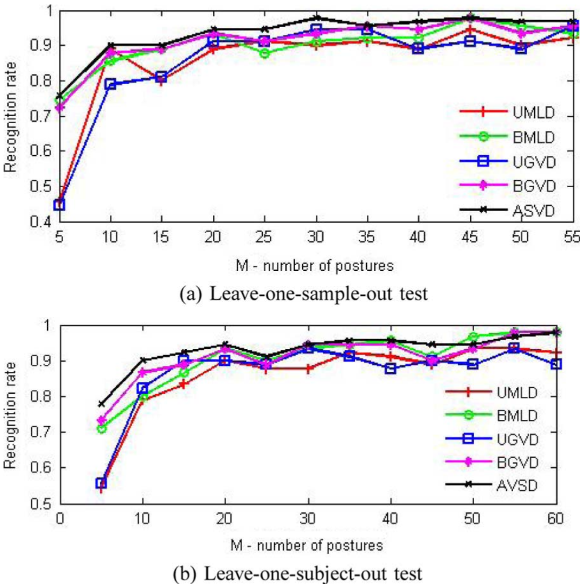
- leave-one-subject-out test;
- robust test against viewpoints and action styles for action *walk* using the sequences designed by Blank *et al.* [12], [26];
- cross-data-set test. In this test, we trained an action graph using Blank's data set and employed the action graph to recognize 68 sequences of actions *walk* and *run* extracted from the video sequences made available by Laptev *et al.* [52].

To test the algorithm for learning new actions, we intentionally left one action out when training the system and, then, added this action into the system using the proposed method. Recognition of the new actions and the impact on the performance of the system with respect to recognizing previously trained actions were evaluated.

In all experiments, silhouette contours were sampled to 64 points after normalization and GMMs with 32 spherical Gaussians were fitted to the shape of the contours. In the learning of new actions, both α_{sp} and α_{mt} were set to 0.3 and β was set to the ratio of the number of frames in the training samples for the new action to the number of frames in the sequences used to train the existing system. The following summarizes the experimental results.

C. Results

1) Leave-One-Sample-Out Test: In the leave-one-sample-out test, each sample was taken as the test sample and the residual samples were used as training samples to train the action graph. Recognition rate was calculated over all the actions in the data set. Fig. 5(a) shows the recognition rates of the five decoding schemes versus number of postures M. As expected, the two bi-gram decoding schemes (BMLD and BGVD) outperformed the two uni-gram schemes (UMLD and UGVD). The ASVD consistently outperformed both uni-gram and bi-gram decoding



(a) Leave-one-sample-out test

(b) Leave-one-subject-out test

Fig. 5. Recognition rates versus number of postures.

TABLE I
DECODING ERRORS FOR EACH TYPE OF ACTIONS IN LEAVE-ONE-SUBJECT-OUT TEST WHEN THE NUMBER OF POSTURES IS 60

	UMLD	BMLD	UGVD	BGVD	ASVD
Run	0	0	1	0	0
Walk	0	0	0	0	0
One-hand-wave	1	0	1	0	0
Two-hands-wave	0	0	0	0	0
Galloping-sideway	1	0	1	0	0
Jumping-in-place	1	1	3	1	1
Jumping	4	1	4	1	1
Jumping-jack	0	0	0	0	0
Bending	0	0	0	0	0
Skip	0	0	0	0	0
Overall Error (%)	7.78	2.22	9.00	2.22	2.22

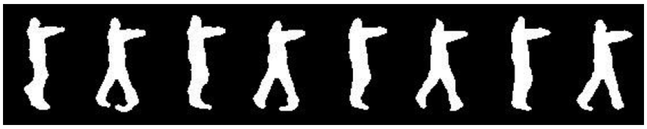


Fig. 6. Sample silhouettes of action *moonwalk*.

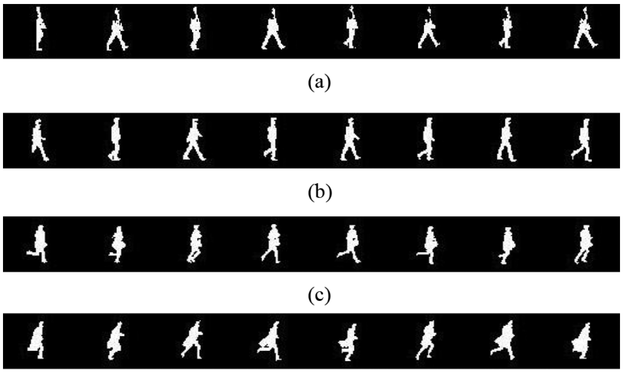
schemes for all M. Notice that the recognition rates of all decoding methods increase as the number of postures increases. When $M >= 20$, the recognition rates are all above 90%. When $M = 45$, the recognition rates of BMLD, BGVD, and ASVD have reached 97.8%, which are comparable to the best rates (96.5%–100%) obtained in [12], [26], and [27] and better than the rate (92.6%) achieved in [3]. It has to be pointed that in [12], [26], and [27], all training samples were kept and KNN was employed to classify the actions.

2) Leave-One-Subject-Out Test: In the leave-one-sample-out test, the training data set contained the samples of other actions performed by the same subject. This certainly helps the action graph to capture the styles of the postures performed by the subject and, therefore, benefits recognition. In the leave-one-subject-out test, we purposely took all samples performed by the same subject as the test samples and the samples performed by other subjects as the training samples. In other words, the trained action graph did not have any knowledge about the test subject. In addition, there was less number of training samples compared to the leave-one-sample-out test. Fig. 5(b) shows the recognition rates of the five decoding schemes versus number of postures M. The curves demonstrate similar patterns to those of the leave-one-sample-out test. BMLD, BGVD, and ASVD achieved recognition accuracies of 97.8% at $M = 60$. Table I shows the recognition errors for each action. As seen, *jumping* and *jumping-in-place* are the most challenging actions to recognize and both uni-gram decoding schemes had some difficulties to recognize them.

Because both leave-one-sample-out test and leave-one-subject-out test have shown that bi-gram and action-specific Viterbi decoding schemes are preferred to the uni-gram decoding schemes, we excluded the uni-gram decoding schemes from the following experiments.

3) Robustness Test: Together with the action data set, Blank *et al.* [26] also supplied additional 20 samples of the action *walk* captured from ten different viewpoints (0° to 81° relative to the image plan with steps of 9°) and ten different styles from zero degree viewpoint (normal, walking in a skirt, carrying briefcase, limping man, occluded legs, knees up, walking with a dog, sleepwalking, swinging a bag, and occluded by a "pole"). We trained an action graph with 30 postures using the 93 samples (from about zero degree viewpoint) for the ten actions (none of the 20 *walk* samples were included in the training data); BMLD, BGVD, and ASVD all recognized most samples and only failed to recognize the action in the cases of 72° and 81° viewpoints. For different walking styles, "occluded by a pole" was excluded in the test because the silhouettes in this case consist of disconnected regions and our method assumes the silhouette is a connected region. Among the rest nine different styles, BMLD, BGVD, and ASVD only failed to recognize the "moonwalk" (walking with arms being raised to the horizontal position). As shown in Fig. 6, it is probably not unreasonable to consider the "moonwalk" as another type of action.

4) Cross-Data-Set Test: We further evaluated the robustness of the proposed model by conducting a cross-data-set test. In this test, we trained an action graph using Blank's data set and employed it to recognize the action samples from a different data set. We chose the data set (video sequences) made available by Laptev [52]. The data set comes as uncompressed video sequences with spatial resolution of 160×120 pixels and comprises six actions (*walking*, *jogging*, *running*, *boxing*, *hand waving*, and *hand clapping*) performed by 25 subjects. Each subject performed each action in four different scenarios: 0° viewpoint, scale variations (from different viewpoints with the subject gradually approaching to or departing from the camera), different clothes (e.g., big pullovers or trench coats), and lighting variations. Two of the six actions, *walking* and *running*, overlap with the actions of Blank's data set. We implemented a simple median-filtering-based background modeling to extract the silhouettes. Because many sequences have severe jitter, the median filter failed to extract the silhouettes. Nevertheless, we managed to extract 36 samples of action *walk* and 32 samples of action *run*. These samples were performed by six



(a)

(b)

(c)

(d)

Fig. 7. Sample silhouette sequences from Laptev's data set: (a) and (b) *walk*; (c) and (d) *run*.

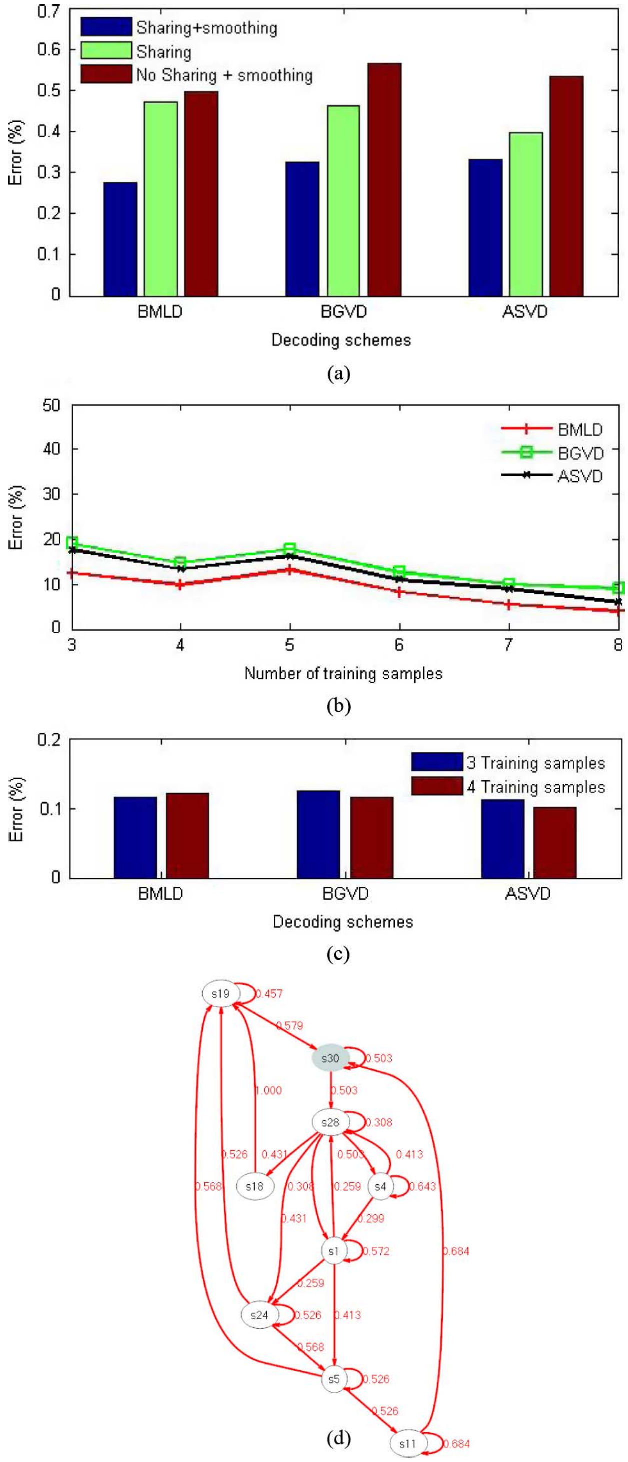
TABLE II
CROSS-DATA-SET TEST: RECOGNITION ERRORS (OUT OF 68) VERSUS NUMBER OF POSTURES FOR BMLD, BGVD, AND ASVD

# of Postures/Decoding Method	20	30	45	60
BMLD	11	5	1	0
BGVD	11	8	2	2
ASVD	14	7	5	3

different subjects. Fig. 7 shows a few examples of the extracted silhouettes. It can be seen that the silhouettes are noisy and, in Fig. 7(d), the subject wore a trench coat that distorted the silhouette shape. Table II is the number of recognition errors (out of 68) versus number of postures. As seen, the recognition rates are over 95% for BMLD, BGVD, and ASVD when the number of postures is 60. Notice that BMLD and BGVD performed better than ASVD. This is probably because ASVD is less generalized than BMLD and BGVD.

5) Learning New Actions: With respect to learning new actions, we first evaluated the significance of *smoothing*. Fig. 8(a) shows the recognition errors for the cases of sharing postures versus not sharing postures and smoothing versus not smoothing when the number of training samples for the new action is one. In sharing, we forced the algorithm not to create any new postures. In the case of not sharing, the algorithm was forced to create three new postures specifically for the new action. In each test, one sample of the action was used as training sample and the rest samples of the same action were used as test samples. The errors showed in the figure were averaged over all actions and all samples in each action. It is apparent that sharing and smoothing significantly reduced the recognition errors and are essential to learning a new action. Notice that, in the case of not sharing, the ASVD scheme is equivalent to the conventional methods where the model for each action is trained independently. It is obvious that our method outperforms the conventional ones.

Fig. 8(b) is the recognition errors of the added new action against the number of training samples. Surprisingly, the BMLD constantly outperformed BGVD and ASVD. On average, we achieved over 85% recognition rate for the new action even though there were only three to four training samples. When the number of training samples reached eight, the recognition rate was improved to over 95%.



(a)

(b)

(c)

(d)

Fig. 8. Learning new actions. (a) Study on the importance of sharing and smoothing. (b) Overall recognition rates of new actions versus number of postures. (c) Impact on the existing system when a new action is added. (d) A typical case that a new posture (S30) was created when*walk* was added to the system as a new action.

We also evaluated the impact on the recognition of previously learned actions when a new action was added. We trained a system by leaving one action out and tested the trained system against the training samples at $M = 30$. In all cases, the training samples were recognized without any error. We then added the left-out action to the system using the proposed method. The new system was evaluated against the samples used for training the previous system. Errors were recorded. Fig. 8(c) shows the averaged errors over all actions when the number of training samples for the new action was three and four. The error rates are around 0.1% for all of the bi-gram decoding schemes. In other words, the system was only degraded on average by 0.1% for the previously trained actions after it was updated with a new action.

Fig. 8(d) shows the action paths for *walk* and the new posture (S30) when *walk* was added as a new action to a system trained with 30 postures.

D. Discussion on Scalability

Our experiments have demonstrated that on average about three to five postures per action were required to model the actions in the data set. The average number of postures per action indicates the average length of the action paths in the graph. It is also noticed that an action graph of 30 postures that encodes the ten actions has sparse global and action-specific transitional probability matrices. In other words, many paths in the graph have not been utilized. This leaves much room for the action graph to be expanded with new actions. For an action graph with M postures that encodes L actions, there are on average $M^{M/L}$ paths with M/L postures. For instance, there are about 30^3 =27 000 paths with three postures in an action graph of $M = 30$ and $L = 10$, offering large capacity to encode a large number of actions and their variations.

VII. Conclusion and Future Work

Recognition of human actions is still in its infancy compared to other intensively studied topics like human detection and tracking. This paper has presented a graphical model of human actions and GMM modeling of postures. Experiments have verified that the proposed model is robust against the subjects who perform the actions, tolerant to noisy silhouettes and, to certain degree, viewpoints and action styles. Most importantly, it is scalable and expandable through adaptive sharing of postures. The scalability and expandability are desirable features for any action recognition systems, but these have rarely been studied before. In addition, the model is easy to train with small number of samples due to the sharing of the postures among the actions. It is found that there is no significant difference in performance between the decoding scheme BMLD and BGVD. ASVD can outperform BMLD and BGVD when there are sufficient training samples, but the gain in the performance is at the expense of more computational complexity with less flexibility for continuous decoding of actions.

The benefit of scalability and expandability becomes dramatically significant in a large scale action recognition system. Our intention is to further evaluate the proposed model on a larger data set. Meanwhile, the proposed model of actions opens a number of theoretical and practical questions to be researched. For instance, what is the optimal number of postures for a given set of actions and desired expandability, and how can the postures be learned from the samples such that the recognition errors can be minimized?

In the proposed algorithm for learning new actions, we only considered whether a new posture is similar to the postures in the trained system. It is also important to measure how an added path for the new action would compromise the existing action paths when no new postures are required. In addition, we assume that samples for a new action are collected first and then input to the system for learning. It is possible to relax this condition by letting the system to decide whether a sample is from a new action and, thus, to launch the new action learning process automatically. Solutions to these two problems could eventually lead to an online learning and recognition system for human actions.

Acknowledgment

The authors would like to thank M. Blank at the Weizmann Institute and I. Laptev at Computational Vision and Active Perception Laboratory (CVAP), NADA, KTH, Stockholm, Sweden, for sharing their data sets. They would also like to thank Dr C. Zhang for many inspiring discussions during the course of the project. W. Li would like to thank Communication and Collaboration Systems Group, Microsoft Research (MSR) for offering the support and opportunities to develop the work at MSR, Redmond, WA.

References

[1] V. M. Zatsiorsky, *Kinematics of Human Motion*. Champaign, IL: Human Kinetics Publishers, 1997.

[2] N. Nguyen, D. Phung, S. Venkatesh, and H. Bui, "Learning and detecting activities from movement trajectories using the hierarchical hidden Markov models," in *Proc. Comput. Vis. Pattern Recognit.*, 2005, vol. 2, pp. 955–960.

[3] S. Ali, A. Basharat, and M. Shah, "Chaotic invariants for human action recognition," in *Proc. Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[4] W. Lin, M.-T. Sun, P. Poovandran, and Z. Zhang, "Human activity recognition for video surveillance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 8, pp. 1128–1139, Aug. 2008.

[5] A. Galata, N. Johnson, and D. Hogg, "Learning variable-length Markov models of behavior," *Comput. Vis. Image Understand.*, vol. 81, pp. 398–413, 2001.

[6] T. B. Moeslund, A. Hilton, and V. Kruger, "A survey of advances in vision-based human motion capture and analysis," *Comput. Vis. Image Understand.*, vol. 104, pp. 90–126, 2006.

[7] A. Oikonomopoulos, I. Patras, and M. Pantic, "Spatiotemporal salient points for visual recognition of human actions," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 36, no. 3, pp. 710–719, Jun. 2006.

[8] J. Niebles and F.-F Li, "A hierarchical model of shape and appearance for human action classification," in *Proc. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[9] F. Lv and R. Nevatia, "Recognition and segmentation of 3-D human action using HMM and multi-class adaboost," in *Proc. Eur. Conf. Comput. Vis.*, 2006, vol. 4, pp. 359–372.

[10] A. Yilmaz and M. Shah, "Actions sketch: a novel action representation," in *Proc. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 984–989.

[11] Y. Ke, R. Sukthankar, and M. Hebert, "Event detection in crowded videos," in *Proc. Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[12] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.

[13] Q. Wei, M. Hu, X. Zhang, and G. Luo, "Dominant sets-based action recognition using image sequence matching," in *Proc. Int. Conf. Image Process.*, 2007, vol. VI, pp. 133–136.

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

[15] D.-Y. Chen, H.-Y. M. Liao, and S.-W. Shih, "Humnn action recognition using 2-D spatio-temporal templates," in *Proc. Int. Conf. Multimedia Expo*, 2007, pp. 667–670.

[16] V. Kellokumpu, M. Pietikainen, and J. Heikkila, "Human activity recognition using sequences of postures," in *Proc. IAPR Conf. Mach. Vis. Appl.*, 2005, pp. 570–573.

[17] J. Yamato, J. Ohya, and K. Ishii, “Recognizing human actions in time-sequential images using hidden Markov model,” in *Proc. Comput. Vis. Pattern Recognit.*, 1992, pp. 379–385.

[18] A. Bobick and J. Davis, “The recognition of human movement using temporal templates,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, Mar. 2001.

[19] J. W. Davis and A. Tyagi, “Minimal-latency human action recognition using reliable-inference,” *Image Vis. Comput.*, vol. 24, pp. 455–472, 2006.

[20] D. M. Gavrila, “The visual analysis of human movement: A survey,” *Comput. Vis. Image Understand.*, vol. 73, no. 1, pp. 82–98, 1999.

[21] W. Hu, T. Tan, L. Wang, and S. Maybank, “A survey on visual surveillance of object motion and behaviors,” *IEEE Trans. Syst. Man. Cybern. C, Appl. Rev.*, vol. 34, no. 3, pp. 334–351, Aug. 2004.

[22] T. B. Moeslund and E. Granum, “A survey of computer vision-based human motion capture,” *Comput. Vis. Image Understand.*, vol. 81, pp. 231–268, 2001.

[23] M. Hu, “Visual pattern recognition by moment invariants,” *IRE Trans. Inf. Theory*, vol. 8, no. 2, pp. 179–187, Feb. 1962.

[24] H. Meng, N. Pears, and C. Bailey, “A human action recognition system for embedded computer vision application,” in *Proc. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[25] H. Fujiyoshi and A. Lipton, “Real-time human motion analysis by image skeletonization,” in *Proc. 4th IEEE Workshop Appl. Comput. Vis.*, 1998, pp. 15–21.

[26] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” in *Proc. Int. Conf. Comput. Vis.*, 2005, pp. 1395–1402.

[27] L. Wang and D. Suter, “Learning and matching of dynamic shape manifolds for human action recognition,” *IEEE Trans. Image Process.*, vol. 16, no. 6, pp. 1646–1661, Jun. 2007.

[28] Y. Wang, H. Jiang, M. D. Z.-N. Lia, and G. Mori, “Unsupervised discovery of action classes,” in *Proc. Comput. Vis. Pattern Recognit.*, 2006, vol. 2, pp. 1654–1661.

[29] D. Kendall, D. Barden, T. Carne, and H. Le*, Shape and Shape Theory*. New York: Wiley, 1999.

[30] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[31] C. Colombo, D. Comanducci, and A. D. Bimbo, “Compact representation and probabilistic classification of human actions in videos,” in *Proc. IEEE Adv. Video Signal Based Surveillance*, 2007, pp. 342–346.

[32] A. Veeraraghavan, A. R. Chowdhury, and R. Chellappa, “Role of shape and kinematics in human movement analysis,” in *Proc. Comput. Vis. Pattern Recognit.*, 2004, vol. 1, pp. 730–737.

[33] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” in *Proc. IEEE*, Feb. 1989, vol. 77, no. 2, pp. 257–286.

[34] R. D. Green and L. Guan, “Quantifying and recognizing human movement patterns from monocular video images. Part i: A new framework for modelling human motion,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 2, pp. 179–190, Feb. 2004.

[35] A. Wilson and A. Bobick, “Recognition and interpretation of parametric gesture,” in *Proc. Int. Conf. Comput. Vis.*, 1998, pp. 329–336.

[36] M. Brand and V. Kettnaker, “Discovery and segmentation of activities in video,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 844–851, Aug. 2000.

[37] N. Oliver, A. Garg, and E. Horvits, “Layered representations for learning and inferring office activity from multiple sensory channels,” *Comput. Vis. Image Understand.*, vol. 96, pp. 163–180, 2004.

[38] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas, “Conditional models for contextual human motion recognition,” in *Proc. Int. Conf. Comput. Vis.*, 2005, vol. 2, pp. 1808–1815.

[39] S. Carlsson and J. Sullivan, “Action recognition by shape matching to key frames,” in *Proc. Workshop Models Versus Exemplars Comput. Vis.*, 2001, pp. 1–8.

[40] M. Ahmad and S.-W. Lee, “Human action recognition using multi-view image sequences features,” in *Proc. 7th Conf. Autom. Face Gesture Recognit.*, 2006, pp. 523–528.

[41] M. Pierobon, M. Marcon, A. Sarti, and S. Tubaro, “Clustering of human actions using invariant body shape descriptor and dynamic time warping,” in *Proc. IEEE Adv. Video Signal Based Surveillance*, 2005, pp. 22–27.

[42] C. D. Manning and H. Schütze*, Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 1999.

[43] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[44] M. Pavan and M. Pelillo, “Dominant sets and pairwise clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 167–172, Jan. 2007.

[45] R. Hathaway and J. Bezdek, “NERF C-means: non-Euclidean relational fuzzy clustering,” *Pattern Recognit.*, vol. 27, pp. 429–437, 1994.

[46] G. Zheng, W. Li, P. Ogunbona, L. Dong, and I. Kharitonenko, “Simulation of human motion for learning and recognition,” in *Lecture Notes in Artificial Intelligence 4304*, A. Sattar and B. Kang, Eds. Berlin, Germany: Springer-Verlag, 2006, pp. 1168–1172.

[47] G. Zheng, W. Li, and P. Ogunbona, “Human motion simulation and action corpus,” in *Digital Human Modeling, Lecture Notes in Computer Science 4561*, V. Duffy, Ed. Berlin, Germany: Springer-Verlag, 2007, pp. 314–322.

[48] J. R. Hershey and P. A. Olsen, “Approximating the Kullback Leibler divergence between Gaussian mixture models,” in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2007, vol. IV, pp. 317–320.

[49] C. Beccehetti and L. P. Ricotti*, Speech Recognition: Theory and C++ Implementation*. New York: Wiley, 2002.

[50] X. Huang, A. Acero, and A.-W Hon*, Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Englewood Cliffs, NJ: Prentice-Hall, 2001.

[51] A. Veeraraghavan, R. Chellappa, and Roy-Chowdhuryh, “The function space of an activity,” in *Proc. Comput. Vis. Pattern Recognit.*, 2006, pp. 959–968.

[52] I. Laptev and T. Lindeberg, “Space-time interest points,” in *Proc. Int. Conf. Comput. Vis.*, 2003, pp. 432–439.

Wanqing Li (M’02) received the B.Sc. degree in physics and electronics and the M.Sc. degree in computer science from Zhejiang University, Zhejiang, China, in 1983 and 1987, respectively, and the Ph.D. degree in electronic engineering from The University of Western Australia, Perth, W.A., Australia, in 1997.

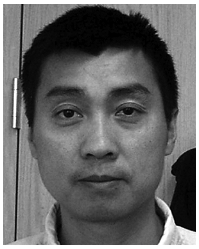
He was a Lecturer (1987–1990) and Associate Professor (1991–1992) at the Department of Computer Science, Zhejiang University. He joint Motorola Lab, Sydney, Australia (1998–2003) as a Senior Researcher and later a Principal Researcher. From December 2007 to February 2008, he was a Visiting Researcher at Microsoft Research, Redmond, WA. He is currently with SCSSE, University of Wollongong, Keiraville, N.S.W., Australia. His research interests include human motion analysis, audio and visual event detection, and object recognition.

Dr. Li has served as a publication chair of MMSP (2008), General Co-Chair of ASIACCS (2009) and DRMTICS (2005), and technical committee members of many international conferences including ICIP (2003–2007).

Zhengyou Zhang (SM’97–F’05) received the B.S. degree in electronic engineering from the University of Zhejiang, Zhejiang, China, in 1985, the M.S. degree in computer science from the University of Nancy, Nancy, France, in 1987, and the Ph.D. degree in computer science (specializing in computer vision) and the Dr. Sci. (Habil. diriger des recherches) diploma from the University of Paris XI, Paris, France, in 1990 and 1994, respectively.

He is a Principal Researcher with Microsoft Research, Redmond, WA. He has been with INRIA for 11 years and was a Senior Research Scientist from 1991 to 1998. During 1996–1997, he spent a one-year sabbatical as an Invited Researcher at the Advanced Telecommunications Research Institute International (ATR), Kyoto, Japan. He has published over 150 papers in refereed international journals and conferences, and has coauthored three books in computer vision.

Dr. Zhang is an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA, an Associate Editor of the *International Journal of Computer Vision* (IJCV), an Associate Editor of the *International Journal of Pattern Recognition and Artificial Intelligence* (IJPRAI), and an Associate Editor of *Machine Vision and Applications* (MVA). He served on the Editorial Board of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (PAMI) from 2000 to 2004, and of two other journals. He has been on the program committees for numerous international conferences, and was an Area Chair and a Demo Chair of ICCV (2003), a Program Co-Chair of ACCV (2004), a Demo Chair of ICCV 2005, a Program Co-Chair of MMSP (2006), and a Program Co-Chair of International Workshop on Motion and Video Computing (2006). He has given a number of keynotes in international conferences.



Zicheng Liu (SM'05) received the B.S. degree in mathematics from Huazhong Normal University, Wuhan, China, in 1984, the M.S. degree in operational research from the Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing, China, in 1989, and the Ph.D. degree in computer science from Princeton University, Princeton, NJ, in 1996.

He is a Researcher at Microsoft Research, Redmond, WA. Before joining Microsoft, he was a Member of Technical Staff at Silicon Graphics, focusing on trimmed nonuniform rational B-spline (NURBS) tessellation for computer-aided design (CAD) model visualization. His research interests include linked figure animation, face modeling and animation, face relighting, image segmentation, and multimedia signal processing.

Dr. Liu is an Associate Editor of *Machine Vision and Applications*. He was a Co-Chair of the 2003 IEEE International Workshop on Multimedia Technologies in E-Learning and Collaboration, a Program Co-Chair of the MMSP (2006), an Electronic Media Co-Chair of ICME (2007).