Activity Modeling Using Event Probability Sequences

Naresh P. Cuntoor, *Member, IEEE*, B. Yegnanarayana, *Senior Member, IEEE*, and Rama Chellappa, *Fellow, IEEE*

***Abstract*—Changes in motion properties of trajectories provide useful cues for modeling and recognizing human activities. We associate an event with significant changes that are localized in time and space, and represent activities as a sequence of such events. The localized nature of events allows for detection of subtle changes or anomalies in activities. In this paper, we present a probabilistic approach for representing events using the hidden Markov model (HMM) framework. Using trained HMMs for activities, an event probability sequence is computed for every motion trajectory in the training set. It reflects the probability of an event occurring at every time instant. Though the parameters of the trained HMMs depend on viewing direction, the event probability sequences are robust to changes in viewing direction. We describe sufficient conditions for the existence of view invariance. The usefulness of the proposed event representation is illustrated using activity recognition and anomaly detection. Experiments using the indoor University of Central Florida human action dataset, the Carnegie Mellon University Credo Intelligence, Inc., Motion Capture dataset, and the outdoor Transportation Security Administration airport tarmac surveillance dataset show encouraging results.**

***Index Terms*—Activity modeling, event detection, hidden Markov model (HMM).**

I. Introduction

HUMAN activities can be decomposed into a sequence of events that have a natural physical interpretation. This can be accomplished using semantic approaches [1]–[5] in which events are prespecified; or using statistical approaches [6]–[11], in which modeling is viewed as a problem of inferring (hidden) events from observed data. We present a statistical approach for modeling activities as a sequence of events.

At the outset, we briefly discuss the terms activities, actions, primitives and events. Existing approaches distinguish between actions and activities depending on the scale of representation [2], [12]; i.e., individual parts of the body are said to perform actions such as picking up and putting down objects, whereas human interaction with the environment constitutes activities. We do not make this distinction and use the term activities to denote both these cases. The distinction is not crucial since we are interested in applications such as activity recognition and anomaly detection. Primitives can be considered as building blocks of activities that are spread over an interval of time [5], [13]. We use the term events to denote instantaneous entities that can be thought of as boundaries between primitive segments.

Events can be defined based on dominant and persistent characteristics of the data. For example, events can be associated with key frames or exemplars [14], [15]. On the other hand, they can be defined using significant changes in velocity, curvature of motion trajectories and other motion properties [12], [16]–[20]. Change-based events are naturally suited to anomalous event detection. Also, as discussed in Section III, change-based events can characterize several commonly occurring activities.

We propose an event detection technique using the hidden Markov model (HMM) framework that focusses on stable state transitions under the hypothesis that certain state level transitions denote events. Transitions at the state level are robust to changes that are triggered by noisy measurements. Robustness to noise is enhanced by stable state changes. By stable change, we mean that the probability of event occurrence depends on the value of state variable for several frames before and after the event. Specifically, if the posterior probability of a state sequence $\{q_{t-p+1} = i, q_{t-p+2} = i, \ldots, q_{t-1} = i, q_t = j, q_{t+1} = j, \ldots, q_{t+p} = j\}$ for some distinct pair of states i, j, attains a persistent local maximum at time t over a range of values of p, the state change is said to be stable. During event detection, several state sequences of the HMM are explored as follows. Consider an observed data sequence (e.g., motion trajectory) of length T. In an ergodic HMM with N states, there are N^T possible state sequences, each of which can generate the observed data with some probability. The optimal state sequence which is one among these state sequences, maximizes this likelihood; but need not have a semantic interpretation. Given an observation sequence and a learned HMM, an event probability sequence $\{e_t^p, t = 1, \ldots, T, p = 2, \ldots, P\}$ is computed in which a local maximum denotes an event. Here p denotes the scale parameter. An efficient way of exploring state sequences to compute $\{e_t^p\}$ is presented.

It is desirable to find methods that are invariant to viewing conditions. The HMM, however, is view-dependent since 2-D motion trajectories are used in training. Multiple HMMs are required if the appearance of trajectories changes significantly. We describe the conditions on the HMMs that enable detection of similar events irrespective of viewing direction. Event probability sequences, unlike HMMs, are shown to be quasi-view invariant (Section VI).

Manuscript received August 7, 2006; revised November 25, 2007. This work was supported by the Advanced Research and Development Activity, a U.S. Government entity which sponsors and promotes research of import to the intelligence community. This work was done when N. P. Cuntoor was a graduate student in the University of Maryland. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Dan Schonfeld.

N. P. Cuntoor is with the Signal Innovations Group, Research Triangle Park, NC 27703 USA (e-mail: nareshpc@gmail.com).

B. Yegnanarayana is with the International Institute of Information Technology, Gachibowli, Hyderabad-500032 India.

R. Chellappa is with the Center for Automation Research and the Department of Electrical and Computer Engineering, University of Maryland, College Park MD 20742 USA (e-mail: rama@umiacs.umd.edu).

Digital Object Identifier 10.1109/TIP.2008.916991

The utility of event probability sequences for activity recognition and anomaly detection is demonstrated using both indoor and outdoor scenarios. It is common to find several samples of normal activities, but very few corresponding to anomalous events. It is not practical to model all possible anomalous activities, some of which can arise due to subtle, statistically insignificant deviations from normal cases. Events in the proposed method can be used to detect such anomalies since they are a result of local changes.

The rest of the paper is organized as follows. Section II discusses related work. Section III motivates the event model and Section IV describes the proposed event probability sequences. Computation of event probabilities is described in Section V. Section VI discusses view invariance of events. Section VII demonstrates the usefulness of the proposed approach for recognition and anomaly detection using the UCF human action dataset, the CMU motion capture dataset and the TSA airport surveillance dataset. Section VIII concludes the paper.

II. Prior Work

We briefly summarize related work in activity modeling. A detailed review may be found in [21].

A. Events and Actions

In the field of artificial intelligence (AI), activity modeling has been of interest for several decades [22]. In [1], a natural language representation with a hierarchy of verbs was presented using positions of the hand, head and body. In [5], actors, logical predicates, and temporal relations between subevents were used to describe activities. Similar methods have been presented in the computer vision area as well. Multiagent events were recognized using prespecified features such as distance, direction of heading and speed [3]. In [4], formal rule-based languages called video event representation language (VERL) and video event markup language (VEML) were presented. VERL allows the user to define events and choose the desired level of granularity in representation. Events were considered as long-term temporal objects at multiple temporal scales and the chi-squared distance between empirical distributions was used to compare event sequences [23].

Actions were segmented using changes in velocity curves in [20]. Sharp changes in curvature of trajectories were used in [12], instead of velocity curves. These changes are quasi view-invariant, but sensitive to noise, in part because of second-order derivatives used in their computation. Also, many activities do not contain trajectories with changes in curvature. View invariants for human actions in both 2-D and 3-D were developed [24]. In 3-D, actions were represented as curves in an invariance space and the cross ratio was used to find invariants.

The shape of activities formed by interacting objects was represented using Kendall's shape space theory in [17]. Dynamics was modeled on the shape's tangent space using continuous HMMs and its effectiveness was demonstrated using anomalous activity detection. In [25], the factorization approach was used to compute a set of shapes that represent normal activities.

B. State Space Models for Activity Modeling

HMMs have been applied in many vision and speech applications. We provide a representative review of HMM approaches in activity modeling. One of the earliest applications of HMMs in vision was for recognizing tennis strokes [6]. Since then, HMMs have been used for recognizing sign language [9], gait-based identification of humans [14], [26] and activities [27]–[29]. HMMs were augmented with stochastic context free grammar (SCFG) to overcome their limitation in expressing semantics [2]. Trajectories were modeled in two phases: primitives were represented by HMMs, and temporal sequencing between primitives was enforced using SCFG. Presegmentation for training and manual design of SCFG might limit scalability of the approach. In a related work, unsurpervised clustering based on factorization of an affinity matrix of the output of HMMs was used in place of SCFG [29].

Incorporating temporal information using HMMs was shown to outperform clustering trajectories using Gaussian mixture models (GMMs) [28]. Trajectories were divided into subtrajectories based on changes in curvature [30]. The subtrajectories were resampled to ensure that all segments have an equal number of elements. The top few principal components of the subtrajectories were modeled using GMMs and HMMs. Unlike [2], manual segmentation of trajectories into atomic units is not required. However, sensitivity of the second derivative in the curvature computation and resampling the subtrajectories may create artifacts.

Coupled HMMs were used to model actions involving body parts such as hands and head [7] in which states of HMMs representing the motion in different parts of the body are forced to make transitions simultaneously. Generally, deciding which HMM states have to be coupled may not be obvious.

Unlike the first-order HMM that models state transitions using one time-step, i.e., $a_{ij} = P(q_t = j|q_{t-1} = i)$, higher order HMMs model dependencies over several time steps in the past. The state transition matrix $A = [a_{ij}]$ is replaced by tensors of the form $A = [a_{ijk}]$, where $a_{ijk} = P(q_t = k|q_{t-1} = j, q_{t-2} = i)$ in second-order HMMs [31]. Variable length HMMs [32] allow for dependencies with varying number of time steps. Increasing the order is tantamount to augmenting the state space [31], which, in turn, increases computational complexity [33] and the amount of training data required for reliable parameter estimation. Tree pruning was suggested as a solution to this problem in [32]. Like [34], minimum entropy was used as a criterion during state estimation [32].

Our method uses the first-order HMM to compute the probability of an event occuring at every time instant by considering state sequences associated with several frames *before* and *after* each time instant. The state sequence remains first-order Markov, whereas the event probability sequence is not Markovian. An efficient way to compute event probabilities is developed using the forward and backward variables in the Baum–Welch algorithm. Tractability in parameter estimation and event detection is retained. Unlike higher order HMMs, we do not need quantities of the form a_{ijk}. However, if activities are modeled using higher order HMMs instead of the standard HMM, event probability sequences can still be computed.

In layered HMMs, activities are modeled at multiple levels of temporal granularity using several HMMs in parallel [35]. Generally, the states of an HMM need not represent physically meaningful entities when the model is constructed using the maximum likelihood criterion. Instead, entropy minimization was used to estimate the HMM parameters [34]. The resulting states were structured and interpretable.

Instead of using HMMs or its variants, domain knowledge can be explicitly modeled using dynamic Bayesian networks (DBN) [11], [36]. DBNs generalize HMMs; the graph structure reflects domain knowledge by design. If the structure changes, it has to be re-estimated. In [10], sampling was used to learn parameters of the DBN. Propagation-nets were proposed in [11] by incorporation duration modeling in HMMs. The states were manually chosen to reflect domain ontology.

C. Unsupervised Methods for Activity Modeling

Unsupervised classification of activities is well-suited for activity modeling where there are several repeated instances of normally occurring activities [16], [18], [19], [37]. Video clips were quantized into prototypes and unusual activities were detected using the graph cuts algorithm [16]. In [19], trajectories were smoothed and segmented based on changes in acceleration. A set of generic features such as velocity, acceleration and arclength were extracted from the segments, and matched for recognition. Motion trajectories collected over a long period of time were quantized into a set of prototypes representing the location, velocity and size of the objects in [38]. Assuming that the sequence of prototypes in each trajectory consists of a single activity, a co-occurrence matrix representing the probability that a pair of prototypes both occur in the same sequence was estimated and used for classification. Pixel change history was used to identify salient motion blobs modeled using Gaussian mixtures (GMM) [18], [37]. The learned GMM was used to initialize output distributions of a DBN. The DBN parameters were reestimated using an extended forward-backward algorithm [37]. The number of states of the DBN is assumed to be the number of events.

III. Motivation

We motivate an event based representation using commonly observed activities: opening and closing doors, picking up and putting down objects. Fig. 1 shows sample images depicting such activities, along with automatically extracted motion trajectories of the hand. Fig. 1(a) depicts trajectory of the hand when the door is opened; Fig. 1(b)–(d) show trajectories of the hand when an object is picked up from the shelf or the desk [the trajectories are isolated from the scene in Fig. 1(c)–(d)]. Though the three trajectories [Fig. 1(b)–(d)] represent the same activity, the variation in appearance is significant. The high degree of intraclass variability poses a challenge to existing approaches like the HMM. Moreover, for the activity *pick up an object* to occur, the time instant of picking up the object is more important than the rest of the trajectory. This suggests a representation that can highlight important time instants when events such as *start*, *pick up* and *stop* occur. At a finer resolution, we may say that the sequence of events *extend hand—make contact with object—pick up—withdraw hand* forms the activity. Each of these events represent a significant change in the motion trajectory that has a semantic interpretation. These changes are highlighted as peaks in the event probability sequence in the proposed model. Similarly, a typical activity in a parking lot may be represented as a sequence of events: *exit building—enter parking lot—enter car—exit parking lot*.

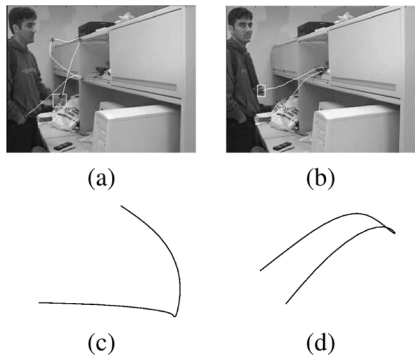

Fig. 1. Sample images and trajectories from UCF dataset. (a) Open cabinet door; (b) pickup object. (c) Smoothed trajectory of the hand for picking up an object from the desk and (d) picking up an object in the cabinet.

IV. Event Probability Sequence

We propose event probability sequences to quantify the notion of important characteristics of activities. Every motion trajectory is associated with an event probability sequence that is computed in two phases: an HMM is learned using the given motion trajectories, and event probability sequences are computed using the learned HMM and given motion trajectories. The HMM enables easy generalization, i.e., the structure of the model need not be manually specified for different activities. Using the HMM, we explore a subset of state sequences to detect events. The hypothesis is that significant changes in the video sequence are reflected as events, and a sequence of events forms an activity.

Let $O = \{o_1, o_2, \ldots, o_T\}$ represent the trajectory (observed sequence) of an object for T frames, where o_t is a 2-D vector of its location at frame t. The observed sequence O is assumed to be generated by an HMM whose hidden state sequence is denoted by $Q = \{q_1, q_2, \ldots, q_T\}$. Here, $q_t \in \{1, \ldots, N\}$, where N is the number of states. For every $t = 1, \ldots, T$, q_t can take any of N discrete values. So, there are N^T possible state sequences that can generate the observed motion trajectory O. Among these N^T state sequences, the optimal state sequence maximizes the probability that the given motion trajectory is observed. Instead of the optimal state sequence alone, we explore other state sequences to detect events.

The key idea is that stable transitions at the state level reflect significant changes in motion properties that are denoted as events. State-level transitions provide a robust representation of change compared to those defined at the data-level. Moreover, the number of distinct changes at the state level at any given time is finite (and equal to $N^2 - N$), and its probability of occurrence can be computed efficiently. The simplest change at the state level is passing from state i at time t to state j at time

$t+1$ (denoted by $i \rightarrow j$). In the example of picking up an object [Fig. 1(b)–(d)], there is a significant change associated with the *pick up* event when the hand, after having made contact with the object reverses direction and withdraws. The *pick up* event can be said to be a semantic change in state from state i before picking up the object to state j after picking up the object. The goal of activity modeling is to detect these changes in motion trajectories extracted from video sequences without an explicit semantic model.

The probability of state change $i \rightarrow j$, conditioned on the observation, can be expressed as

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j | O, \lambda) \tag{1}$$

for states $i, j \in \{1, 2, \ldots, N\}$. $\xi_t(i,j)$ is used in the Baum-Welch algorithm to efficiently compute the HMM parameters [39].

The change in (1) is based on one time-step before and after an event's occurrence. For stable transitions at the state level, instead of transitions of the form $i \rightarrow j$, (1) is modified so that the probability of change associated with the following state transitions $i \rightarrow i \rightarrow j \rightarrow j$ is used, where the inner $i \rightarrow j$ transition occurs at time t [40]. Given the observed trajectory, the probability of its generation by the following state sequence is computed: persistence in state i for two consecutive frames, then a transition to a state j and persistence in state j for two consecutive frames. This uses the following sequence of states to detect an event at time t

$$\underbrace{i \rightarrow i}_{2 \text{ frames}} \rightarrow \underbrace{j \rightarrow j}_{2 \text{ frames}} . \tag{2}$$

The event variable $\eta_t^{(2)}(i,j)$ is defined as follows:

$$\eta_t^{(2)}(i,j) = P(q_{t-1} = i, q_t = i, q_{t+1} = j, q_{t+2} = j | O, \lambda) \tag{3}$$

so that $\eta_t^{(1)}(i,j) \stackrel{def}{=} \xi_t(i,j)$. More generally, the region of support of an event can be extended to p frames by considering the following state sequence instead of (2):

$$\underbrace{i \rightarrow i \rightarrow \cdots \rightarrow i}_{p \text{ frames}} \rightarrow \underbrace{j \rightarrow j \rightarrow \cdots \rightarrow j}_{p \text{ frames}} . \tag{4}$$

Event variable $\eta_t^{(p)}(i,j)$ is defined as follows:

$$\begin{aligned} \eta_t^{(p)}(i,j) = P(&q_{t-p} = i, q_{t-p+1} = i, \ldots, q_t = i \\ &q_{t+1} = j, q_{t+2} = j, \ldots, q_{t+p+1} = j | O, \lambda) \end{aligned} \tag{5}$$

where $p = 2, 3, \ldots, P$. The maximum value P is fixed empirically, depending on the average length of trajectories. At every time instant, there are N^2 transitions between states (i,j). Of these, there are $N(N-1)$ transitions between distinct (i,j). A transition from state i to j, where $j \neq i$ represents change. The most likely change among the $N(N-1)$ transitions may be interpreted as an event. The event probability, parameterized by scale parameter p, is defined as follows:

$$e_t^{(p)}(k,l) = \max_{i \neq j} \eta_t^{(p)}(i,j) \tag{6}$$

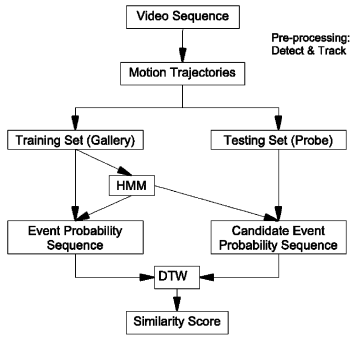

Fig. 2. Outline of the proposed algorithm for activity recognition.

where $(k,l) = \arg\max_{i \neq j} \eta_t^{(p)}(i,j)$. As p increases, the region of support for the event increases. The *before* and *after* states k and l are said to characterize the type of the event.

Thus, an event is specified by the following quantities:

- probability of the event $e_t^{(p)}(k,l)$;
- scale parameter p;
- event type (k,l) denoting the states *before* and *after* the event.

There are several reasons for choosing events based on transitions. It is a simple and robust way of representing change. The event probability sequence can be computed efficiently (Section V). Events are modeled using simple transitions between distinct states of the form given in (5). In practice, it is possible for events to involve more complex transitions. For instance, events may be caused by state transitions of the form $i \rightarrow \cdots j \cdots \rightarrow k$. Depending on the application and availability of data, such complex event probability sequences can be computed. The proposed event probability sequences consisting of simple, stable state transitions captures perceptually salient events, as outlined in Section III and demonstrated in Section VII.

Stable transitions at the state level yield events whose representation is tied to the underlying HMM. Since the HMMs are trained using 2-D motion trajectories, their parameters are view-dependent. This, however, is not a significant limitation provided viewing conditions do not change drastically. A drastic change, for example, is one in which straight-line trajectory in one view collapses to a point in a different view. Under certain assumptions, the event probability sequence is quasi invariant to changes in viewing conditions (Section VI).

Events (i.e., local maxima in event probability sequences) need not occur at the same time instant across multiple samples. So we use dynamic time warping (DTW), which is a dynamic programming technique, to handle changes in the location of events detected (Section V).

V. Approach

Fig. 2 shows an overview of the proposed method. During training, motion trajectories are extracted and used to train an HMM. Using the trained HMM, an event probability sequence $e_t^{(p)}(k,l)$ is computed for every trajectory of the activity. The computation of $e_t^{(p)}(k,l)$ is described next.

Let $\lambda = (A, B, \Pi)$ represent an HMM [39]. In our notation, we have tried to maintain consistency with the commonly used

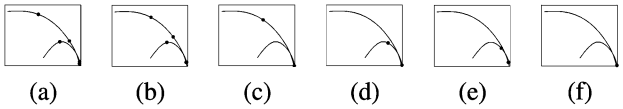

Fig. 3. Events detected at different scales for trajectory of picking up an object. The scale parameter p varies from (a) $p = 3$ to (f) $p = 8$.

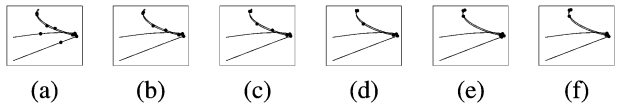

Fig. 4. Events detected at different scales for trajectory of opening the cabinet door. The scale parameter p varies from (a) $p = 3$ to (f) $p = 8$.

HMM notation [39], [41]. $A = [a_{ij}]$ is the probability transition matrix of size $N \times N$, where $a_{ij} = P(q_t = j | q_{t-1} = i)$. The probability of observing a given data vector conditioned on the current state is described by B. In our experiments, the (stationary) output distribution $b_j(o_t) = P(o_t | q_t = j)$ is assumed to be Gaussian. Initial probability of states is given by Π. Parameters of the HMM are computed using the Baum–Welch algorithm [39]. Using the HMM, an event probability sequence is computed for every motion trajectory as follows.

Rewriting (3), we have

$$\eta_t^{(2)}(i,j) = \frac{P(q_{t-1} = i, q_t = i, q_{t+1} = j, q_{t+2} = j, O|\lambda)}{P(O|\lambda)} = \frac{1}{P(O|\lambda)} P\left(q_{t-1} = i, o_1^{t-1}, q_t = i, q_{t+1} = j \; q_{t+2} = j, o_t^T | \lambda\right). \tag{7}$$

It can be shown that (7) simplifies as follows: (proof in Appendix I)

$$\eta_t^{(2)}(i,j) = \frac{\alpha_{t-1}(i) a_{ii} b_i(o_t) a_{ij} b_j(o_{t+1}) a_{jj} b_j(o_{t+2}) \beta_{t+2}(j)}{P(O|\lambda)} \tag{8}$$

where $\alpha_t(i)$ and $\beta_t(j)$ are forward and backward variables [39]

$$\alpha_t(i) = P(o_1, o_2, \ldots, o_t, q_t = i | \lambda) \tag{9}$$

$$\beta_t(j) = P(o_{t+1}, o_{t+2}, \ldots, o_T | q_t = j, \lambda). \tag{10}$$

Similarly, for $p \geq 2$ (proof in Appendix I), see (11), shown at the bottom of the page.

Given a new trajectory (not used for training), candidate event probability sequences are computed using each of the learned HMMs so that there are as many candidate event probability sequences as the number of trained HMMs. Every candidate event probability sequence is compared with those computed during training phase using DTW (Section V-B) to handle slight changes in time alignment [42].

The number of events detected in the event probability sequence depends on the scale p that is used in modeling. For example, if $p = 3$, *picking up* an object has more number of events [Fig. 3(a)]; for $p = 8$, it has one event when the object is picked [Fig. 3(f)]. (Events are represented by dots). Similarly, for *opening* the cabinet door [e.g., Fig. 1(a)], the number of events is related to the choice of scale. Fig. 4 depicts events detected at various scales ($p = 3$-8).

A. Parameter Selection

In this section, we discuss the choice of HMM model order (i.e., number of states) N and scale parameter p. Researchers have proposed several criteria ([39], [43]) to estimate the optimal model order N^* for HMMs. We use the Bayesian information criterion (BIC) in our experiments since it has been shown to be a strongly consistent Markov order estimator [43]. The optimal value N^* is given by

$$N^* = \arg\min_{N \in \mathcal{N}} \left(\mathrm{Log}l(N) + \frac{k_N}{2} \log T \right) \tag{12}$$

where $Logl$ is the negative log likelihood of the observed motion trajectory of length T and k_N is the degrees of freedom associated with the Nth-order model. We describe two choices of scale parameter p: conditionally optimal p^* (conditioned on optimal N^*) and jointly optimal $(N,p)^*$.

1) Conditionally Optimal Scale Parameter p^*: Using BIC, the scale parameter can be chosen as follows:

$$p^* = \arg\min_{p \in \mathcal{P}} \left(\Sigma_{t=1}^{t=T} e_t^p(k,l) + \frac{p}{2} \log T \right). \tag{13}$$

These two criteria for optimality, i.e., $N = N^*$ and $p = p^*$ imply that both the representation of trajectory and the sequence of events are optimal. This is a stronger requirement than finding the optimal scale of event representation. Instead of the optimal pair (N^*, p^*). there may be a pair of values $(N,p)^* = (N_1, p_1)$ that gives an optimal representation of event probabilities, where N_1 may be a suboptimal order. This suggests a way to modify the optimality criterion to focus on the event probability sequences.

2) Jointly Optimal Parameters $(N,p)^*$: Events at certain key frames are said to represent the activity. This does not require the entire trajectory to be modeled optimally. So, we can confine the penalized likelihood calculations to the key frames, or equivalently to the sequence of event probabilities $\{e_t^p, t \in [1,T]\}$. If the underlying HMM is ergodic, each event can be one of $N^2 - N$ possible types as given by (6). If a left-to-right model is assumed for the HMM, then the degrees of freedom for the possible types of events is halved. The overall penalty is $\alpha(N^2 - N) + p$, where $0 < \alpha \leq 1$ with equality for ergodic HMM. For a left-to-right model, $\alpha = (1/2)$. The event likelihood term associated with the pair (N,p) is $\sum_t e_t^p$. We discuss the behavior of event probability sequences for different values of N and p.

3) Case 1: Low value of N: Since only a few events are allowed to occur, we may not be able to obtain a sufficiently rich sequence of events to represent the activity. For example, with

$$\eta_t^{(p)}(i,j) = \frac{\alpha_{t-p+1}(i) a_{ii}^{p-1} b_i(o_{t-p+2}) b_i(o_{t-p+3}) \ldots b_i(o_t) a_{ij} b_j(o_{t+1}) b_j(o_{t+2}) \ldots b_j(o_{t+p}) a_{jj}^{p-1} \beta_{t+p}(j)}{P(O|\lambda)} \tag{11}$$

a two-state HMM, only two types of events are allowed: (1, 2) and (2, 1), where 1 and 2 are state indices. Even at moderate p values, many activities may resemble each other leading to poor recognition performance.

4) Case 2: High value of N: This allows for several types of events. Reliable estimation of event probabilities requires more number of trajectories of the activity. The maximum value of scale P has to be smaller than that of *Case 1* because the transitions in (6) may not be computable for large p.

B. Matching Event Probability Sequences

Given a test sequence O_0, we extract the motion trajectory as before. Suppose there are R trajectories in the training set. Then there are R sets of event probability sequences that are computed using the respective HMMs for different values of p. In general, the number of distinct HMMs is less than the number of training trajectories. Multiple trajectories may be associated with the same HMM, but each of the trajectories is associated with an event probability sequence. We compute R candidate event probability sequences using O_0 as the observation sequence in (5) and (6).

The candidate event probability sequences are compared with the event probability sequences that were obtained during the training phase (at the same scale) to compute a similarity score. If there are P candidate event probability sequences computed for different scale parameters, we obtain P similarity scores. A direct frame-to-frame matching of event probability sequences is not realistic since the events need not occur at exactly the same time instants during different realizations of an activity. There have to be allowances for missed or spurious events as well. We use DTW for computing the similarity score since it allows for nonlinear time normalization [42].

A brief outline of the DTW algorithm is given below; a detailed explanation is available in [42]. The objective of the algorithm is to align a test sequence indexed by $\{x(t), t = 1, \ldots, T_1\}$ with a reference sequence indexed by $\{y(t), t = 1, \ldots, T_2\}$ so that the distance between the two sequences along the warping path $\{C(t), t = 1, \ldots, T\}$ is minimized, subject to the following constraints.

- *Endpoint constraint:* The first and last points of the two sequences are matched, i.e., $x(1) = 1$, $y(1) = 1$, $x(T) = T_1$, $y(T) = T_2$.
- *Monotonicity constraint:* To ensure temporal ordering during time normalization, $x(t-1) \leq x(t)$ and $y(t-1) \leq y(t)$.
- *Local continuity constraint:* To ensure that every point along the test trajectory is used when comparing the sequences, $x(t) - x(t-1) = 1$ and $y(t) - y(t-1) \leq 2$.
- *Global path constraint:* A band of size $2W$ is defined along the diagonal, which constrains the warping region. Instead of a diagonal band, a parallelogram band can also be used.

For every point within the warping region, the distance between the test and reference vectors is computed using a suitable norm (Euclidean, in our case). The cumulative distance is computed using the above constraints. And the warping path is found by backtracking such that the distance between sequences is minimized.

VI. View Invariance in Representation

In this section, we describe view invariance of event probability sequences and develop constraints on the HMMs to allow for view invariance. Motion trajectories on the image plane can change because of relative motion between the camera and the person performing an activity; or because of differences in style of execution. The differences, however, are minimal at the time of occurrence of events. For example, changes near the *pick up* event is similar in different instances of the activity irrespective of initial position, speed and person performing the activity. This is observed in surveillance scenarios such as an airport tarmac. A luggage cart that is meeting a plane on arrival, can enter scene in a wide area. The luggage transfer event occurs when it stops near the plane. This has little variation though the rest of the trajectory can vary drastically. So, event probability sequences can be expected to resemble each other in spite of differences in motion trajectories.

We describe a sufficient condition on the structure of HMMs corresponding to different viewing directions to ensure that similar events are observed across viewing directions. Assume that an affine camera model is used to relate the 3-D trajectories to the 2-D trajectories viewed from two different viewing points. The state descriptions represent the spatial context in the image plane. The transition matrix encodes the temporal evolution of the activity. The key idea is that equivalence relations between models corresponding to two different views can be captured by event probability sequences.

Definition 1: A pair of HMMs is said to be *conforming* if there exists a homeomorphism between the set of states of the HMMs.

1) Proposition 1 (Sufficient Condition for View Invariance of Event Probability Sequences): For event probability sequences to be invariant under changing viewing conditions, the generating HMMs must be conforming. (Proof is given in Appendix II).

To illustrate the above proposition, we consider a set of straight-line trajectories generated by a person walking from one part of the scene to some other fixed part. Assume that the camera axis (z-axis) is perpendicular to the 3-D trajectories. Let the mean-subtracted trajectories be modeled using a 2-state HMM in which each state emits instantaneous positions according to a Gaussian distribution. From (5), an event is detected at the instant of switching from state 1 to 2. Then we analyze the effect of changes in viewing direction.

Rotation by an angle θ about the z-axis causes the 2-D trajectory to rotate by θ and the HMM parameters change correspondingly. In particular, the mean values of the two states also rotate by the same angle. The homeomorphism between the two trajectory models across viewing directions ensures that there exists a corresponding pairs of states (k, l) and $(\tilde{k}, \tilde{l})$ such that $\eta_t^{(p)}(i, j)$ and $\eta_t^{(p)}(\tilde{i}, \tilde{j})$ respectively are maximized; and the event probability sequence is preserved. As the angle changes beyond 90°, the roles of the two states are reversed. Rotation with respect to x-axis (tilt) does not change the 2-D trajectory. Camera rotation with respect to y-axis (pan) by an angle α causes a shortening of the 2-D trajectory by a factor $\cos\alpha$. And the mean values of the two

states move closer. As long as the two states are physically separated in the 2-D image, the states remain distinct. This ensures that O and $\tilde{O}$, as described in the proposition, are of the same dimension. In practice, this means that straight-line trajectories in one viewing direction do not collapse to a point when viewed from most directions (except certain viewpoints).

A. Conforming HMMs

The above example is elaborated using synthetic data. The effect of changing viewing direction can be analyzed by rotating the observed (straight-line) trajectory in the image plane. It has been shown that the apparent slope of the line varies directly with the slope of the line in the 3-D world, assuming that the trajectory is approximately parallel to the ground plane [44], i.e., $\tan\theta = (1/K)\tan\alpha$, where θ is the change in angle in the 3-D world and α is the corresponding value in the image plane; K depends on focal length and can be obtained by camera calibration. The effect of changes in α is described next.

Two HMMs are trained using trajectories captured from multiple viewing directions. From proposition 1, we require a one-to-one correspondence between the states of the two HMMs. A natural measure of similarity between the states is the KL-divergence $D(p\|q)$ [39] or the symmetric form $D(p\|q) + D(q\|p)$, where p and q represent the output distributions corresponding to the two states. The comparison yields a real-valued matrix S of size $N \times N$, where N is the number of states in the two HMMs. The condition on conforming HMMs can then be expressed as follows.

Definition 2: For every $i \in \{1,2,\ldots,N\}$, if $\exists j = j(i) = \arg\min_{\{1,\ldots,N\}} S_{ij}$ such that $j \neq j(i_-)$, where $i_- = \{1,\ldots,i-1\}$, the HMMs that are compared in S are conforming.

Viewing changes within a 126° hemisphere are simulated so that the slope of the line in the image plane varies between $\tan\alpha = -2$ to $\tan\alpha = +2$. Straight line trajectories are simulated with several values of $\tan\alpha$, and contaminated by additive Gaussian noise. Trajectories are divided into three classes for analysis. In the first case, trajectories are divided into two classes; the spread of slopes in each class is 2, i.e., $\tan\alpha_1 \in [0,2]$, $\tan\alpha_2 \in (0,-2]$. In the second case, there are three classes; the slope spread within each class is 1.5. In the third case, there are four classes; the slope spread in each class is 1. In each of these cases, an HMM was trained using trajectories from the respective classes. Let λ_{mn} denote the HMM, where $m = \{1,2,3\}$ corresponds to the three cases, and n is the number of classes in the three cases. So, when the trajectories are divided into two sets ($m = 1$), there are two HMMs $\{\lambda_{11}, \lambda_{12}\}$. Similarly, for $m = 3$, there are four classes and, therefore, four HMMs: $\{\lambda_{31}, \lambda_{32}, \lambda_{33}, \lambda_{34}\}$. For proposition 1 to hold, we need to check that for each $m = \{1,2,3\}$, HMMs $\lambda_{m\cdot}$, taken pairwise, are conforming. This is verified using definition 2. In all the cases, for $i = 1,\ldots,5$, we found that $j = j(i) = i$.

The two sets of event probability sequences $\{m = 3, n = 1\}$ and $\{m = 3, n = 4\}$ were used for training and the sets $\{m = 3, n = 2\}$ and $\{m = 3, n = 3\}$ were used for testing (Fig. 5). The event probability sequences of $\{m = 3, n = 1\}$ set matched closer to $\{m = 3, n = 2\}$ than $\{m = 3, n = 4\}$. Similarly, those of $\{m = 3, n = 3\}$ matched closer to those of $\{m = 3, n = 4\}$. These are plotted in Fig. 5 and superimposed on event probability sequences of the training set to illustrate the location of events remains invariant to changes in viewing direction. Each plot in Fig. 5 depicts multiple event probability sequences computed for four viewing directions (−65°, −30°, 65° and 30°) respectively. In Fig. 5(a) and (b), the HMM trained using the −65° viewing direction is used to compute event probabilities. In Fig. 5(c) and (d), the HMM corresponding to 65° is used. So, the test directions in Fig. 5(b) and (c) are previously unobserved. That the strength of event probability sequences changes with viewing directions is not crucial since only the locations of events (local maxima in event probability sequences) are matched during recognition.

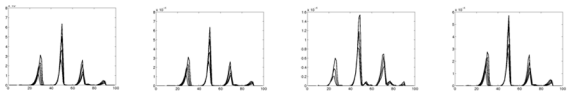

Fig. 5. View invariance of event probability sequences using synthetic trajectories. Four viewing directions ranging from -65° and $+65^\circ$ are used to demonstrate the invariance of location of events (i.e., local maxima in event probability sequences). Event probabilities are plotted as a function of time.

VII. Experiments

The utility of event probability sequences is demonstrated using indoor and outdoor video sequences. The following datasets are used: the UCF human action dataset, the CMU/Credo Intelligence Inc. MOCAP dataset and the TSA airport tarmac surveillance dataset.

A. Activity Recognition

1) UCF Human Action Dataset: The UCF dataset (Fig. 1) consists of common activities performed in an office environment. We divide the dataset into the following seven classes (the number of samples per class is given in paranthesis): *open door* (18), *pick up* (21), *put down* (17), *close door* (4), *erase board* (4), *pour water into cup* (3), and *pick up object and put down elsewhere* (8). The hand trajectories are obtained after initialization using a skin detection technique. The resulting trajectories are smoothed out using anisotropic diffusion [45]. Detailed description of the dataset and preprocessing are available in [12]. Most of the activities last for a few seconds.

Training Conforming HMMs: Given trajectories of an activity, we train an HMM using the Baum-Welch algorithm [39]. Each state is modeled using a single Gaussian distribution. A fully connected state transition matrix is initialized, which has equal probability of transition to any state. The output distributions for each state are initialized using the segmental k-means algorithm. After reestimation, the matrix exhibited a left-to-right structure.

If variations in viewing direction are small, multiple HMMs are not required since the distribution of the states of the HMM provides sufficient generalization across these views (except in

case of accidental alignment). Variance of the output distribution tends to increase with increasing variations in viewing direction. During training, we train multiple HMMs for an activity if viewing conditions change significantly. By inspection, we found that each of the following activities needed two HMMs: *open the cabinet door*, *pick up an object*, *put down an object*, whereas *close the cabinet door*, *erase white board*, *pour water into a cup*, *pick up an object and put it down elsewhere* all had one HMM per activity.

Event Probability Sequences for training data: Using the learned HMM, we compute an event probability sequence for every trajectory using (5) and (6). The event probability sequence along with the HMM forms a signature of the activity. Events detected at different values of scale parameter are illustrated in Figs. 3 and 4.

Matching: Given an new trajectory x, we compute a set of candidate event probability sequences $e_t^{x,g}(k,l)$ for every learned HMM using HMM λ_g for $g = 1,\ldots,G$, where G is the number of HMMs obtained during training. The DTW algorithm is used to compare the candidate event sequences with those obtained during training (Section V-B). We compute candidate event probability sequences at 6 scales for scale parameter values $p = 3$ to $p = 8$. They are matched with training event probability sequences, at the corresponding scale and 6 similarity scores are obtained. Coarse-to-fine matching is used to compute the overall similarity score (Section V-B). Fig. 11 shows recognition rates and comparison to those in [12].

Analysis of results: We illustrate properties of the recognition scheme using a few examples.

- *Context insensitivity:* Fig. 7 shows trajectories and event probability sequences of two *pick up* activities that differ in context. Though HMMs for the two cases—picking up object from a desk and picking up an umbrella from a cabinet—are different, both their event sequences show peaks surrounding the instant when the *pick up* event occurs.
- *Quasi view invariance:* As long as viewing directions do not create singularities, events detected remain insensitive to the viewing direction. Fig. 8 shows differences in appearance due to differing position of the person opening the cabinet. All the four trajectories were correctly recognized.
- *Spurious events:* Stylistic variations or errors in tracking may cause spurious events (Fig. 6). Event sequences are correctly recognized because of the DTW algorithm used in matching.
- *Subactivities:* Composite activities may have subactivities embedded within them. Depending on the application, we may wish to recognize subactivities separately; or the composite activity. Fig. 9 illustrates the former. The top three matches were subactivities, and the fourth match was the correct composite activity. In Fig. 10, the composite activity is recognized correctly in the top first and second matches. The next three matches are the subactivities.

Comparison with the UCF method [12]: Rao *et al.* treat activities as a sequence of *dynamic instants* [12], which are points of maximum curvature along the trajectory. Unlike [12], events in e_t are robust to changes in curvature. For instance, *pick up umbrella while twisting the hand* is not recognized in [12] because of excessive peaks in the curvature. The event probability sequences are correctly matched. Trajectories without finite curvature cannot be recognized by dynamic instants. This is not a limitation in the proposed method. A comparison of recognition

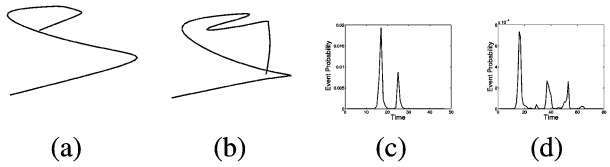

Fig. 6. Invariance to spurious events: (a) hand trajectory for closing the door, (c) its event probability sequence, (b) closing the door along with random motion. This generates a spurious peak in the event probability sequence, shown in (d). The activity is recognized correctly in spite of the spurious peak.

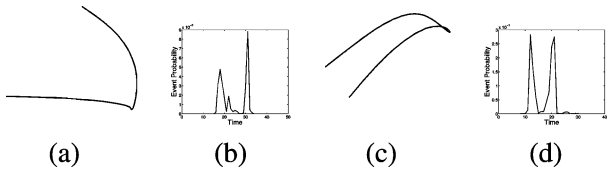

Fig. 7. (a) Hand trajectory and (b) its event probability sequences for picking up object from desk with $p = 5$; (c) and (d) for pick up an umbrella from cabinet. Both instances of pick up activity have two dominant events.

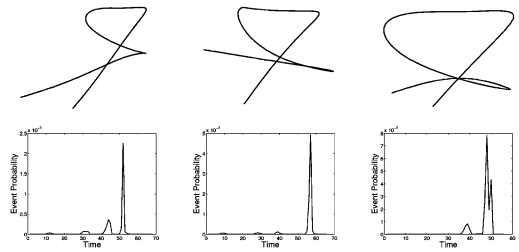

Fig. 8. Quasi-view Invariance: Different samples of opening the cabinet door. The appearance of the trajectory depends on the location of the person performing the activity. (a)–(c) Hand trajectories; (d)–(f) corresponding event probabilities.

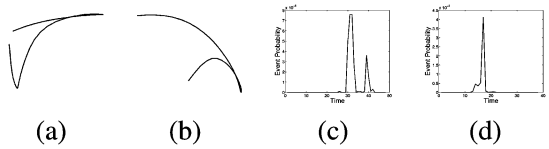

Fig. 9. Recognizing subactivities: (a) hand trajectory for pick up object from desk and put it on desk; (c) its event probability. (b) Pick up object trajectory and (d) the event sequence for the top match using (a) as test sequence.

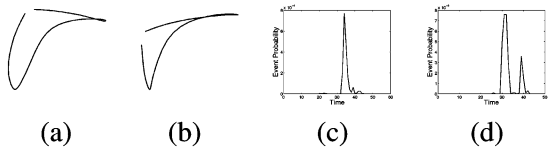

Fig. 10. Recognizing composite activities: (a) hand trajectory for picking up an object from the desk and putting it back on the desk; (c) corresponding event probability. (b) Another sample of the same composite activity. It was the top match when (a) as test sequence. (d) Event probability sequence computed during the testing phase.

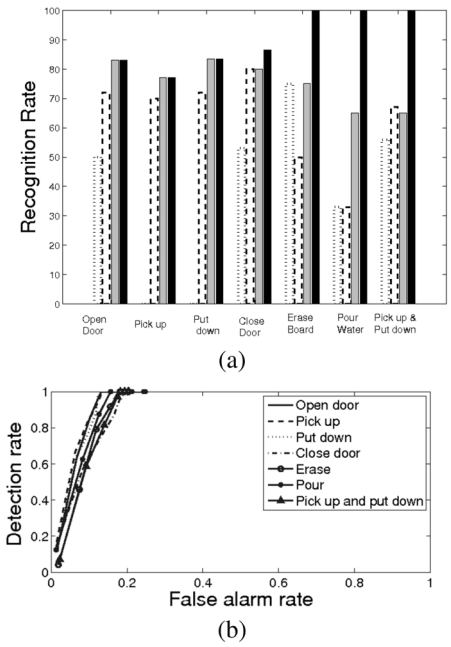

Fig. 11. (a) Recognition rate for UCF database. Dotted line: UCF results [12]. Dashed line: Overall similarity obtained by integrating P similarity scores from coarse-to-fine scales. Gray bar: Conditionally optimal p^*. Black bar: Jointly optimal $(N, p)^*$. (b) ROC curves for different activities in the conditionally optimal case.

rates is given in Fig. 11. It demonstrates that a jointly optimal choice of parameters (N, p) improves recognition performance.

2) Motion Capture (MOCAP) Dataset: The MOCAP dataset available from Credo Interactive Inc. and CMU consists of motion capture data of subjects performing different activities including walking, jogging, sitting, and crawling. The system tracks 53 joint locations and stores the trajectories in *bvh* format. Since not all the 53 points are relevant to the types of activity that we are interested in, we use only a few of the trajectories. For example, trajectories of fingers and toes may not be as informative as the location of the arms, legs or hip for activities such as walking or sitting. We choose the following five such regions to demonstrate activity classification: head, neck, shoulders, hands and feet. Within each region, multiple points are tracked. The trajectory of each region is computed using the mean of points in the region. Averaging over a local region also helps in dealing with limited availability of data.

The dataset contains nine activities and 75 sets of trajectories. Some trajectories (e.g., walking) consist of multiple cycles of the activity. The sequence is divided into walking cycles and each half-cycle is treated as an observation. Half-cycle refers to the part of the walking cycle starting from the standing pose, right (or left) leg forward, reaching the swing pose, and withdrawing the right (or left) leg to the standing pose.

Training event probability sequences: The dataset is divided into two halves for training and testing. Using the training set, one HMM is learned per activity. A four-state HMM with single Gaussian output distribution is used. Event probability sequences are computed for every trajectory (Section IV) using the learned HMM for the activity. This forms the training set of event probability sequences. Fig. 12 shows event probability sequences for *sit, blind walk, and normal walk.*

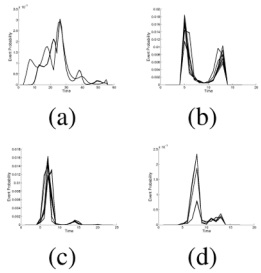

Fig. 12. MOCAP dataset: Examples of event probability sequences. Each figure has multiple event sequences corresponding to multiple observations. (a) Sit, (b) blind walk, and (c), (d) two instances of normal walk.

TABLE I
MOCAP DATASET: CLOSEST-MATCHING ACTIVITIES BASED ON COMPARING EVENT PROBABILITY SEQUENCES. ALL ACTIVITIES WERE CORRECTLY RECOGNIZED. THE SECOND BEST, THIRD BEST, AND FOURTH BEST MATCHES, RESPECTIVELY

Test activity	2^{nd} best match	3^{rd} best match	4^{th} best match
Blind-walk (BW)	EW	NW	Jog
Prowl-walk (PW)	NW	EW	Jog
Broom	Broom2	EW	NW
Crawl	PW	Jog	NW
Exaggerated walk (EW)	NW	Jog	BW
Jog	NW	PW	EW
Sit	Sit1	Jog	PW
Normal walk (NW)	SW	Jog	PW
Sad walk (SW)	NW	Jog	Jog

Comparing test data: Given a test sequence, we compute the same number of candidate event probability sequences as the number of learned HMMs. The scale parameter is fixed at $p = 3$. Increasing the value of p from $p = 4$ to $p = 8$ did not produce significant changes in the events detected. Event probability sequences are compared with those in the training set using the DTW algorithm. All the activities were correctly recognized. Table I summarizes activities that were the closest matches following the top match. We observe that the different types of walking resemble each other whereas the similarity scores corresponding to *sitting*, *sweeping with a broom* are significantly larger.

View invariance: We tested view invariance of event probability sequences for two activities across four viewing directions within a 120° hemisphere (60° on either side of reference direction). We chose *walking* and *sweeping with a broom* as the two activities. The trajectories for the different test conditions were synthetically generated from the 3-D motion capture data. The classification rates are summarized in Table II. The training data in all four test viewing directions was the reference $\alpha = \beta = \gamma = 0$. The event probability sequences for the test sequences are generated using the trained HMM for the reference direction. The classification rates (Table II) demonstrate robustness of event probability sequences to viewing direction.

B. Anomalous Trajectory Detection

The TSA airport tarmac surveillance dataset is used to demonstrate the usefulness of event probability sequences for anomalous activity detection.

1) TSA Airport Tarmac Surveillance Dataset: The TSA dataset (Fig. 13) consists of surveillance videos of an airport

TABLE II
MOCAP DATASET: EFFECT OF CHANGING VIEWING DIRECTION: CLASSIFICATION RATE FOR DISTINGUISHING BETWEEN TWO ACTIVITIES—WALKING, SWEEPING. T.V.D. = TEST DATA VIEWING DIRECTION. ALL FOUR CASES ARE COMPARED TO THE REFERENCE (0, 0, 0) VIEWING DIRECTION

	T.V.D. #1	T.V.D. #2	T.V.D. #3	T.V.D. #4
Walk	100	94	91	80
Sweep	100	98	95	95

(a) (b)

Fig. 13. (a) Snapshot of TSA dataset. (b) Simulated anomaly—person deviates from virtual path between plane and gate, and walks toward fuel truck.

tarmac captured using a stationary camera that operates at approximately 30 frames per second. The image size is 320×240. It contains approximately 230 000 frames or 120 minutes of data. Activities include movement of ground crew personnel, vehicles, planes and passengers. The trajectories of vehicles, passengers and luggage carts are extracted as follows.

The background is modeled using a Gaussian distribution at every pixel. Ten consecutive frames are used to compute the background model and moving objects are detected. The KLT tracker, which was originally developed as an image registration method, was used to obtain motion trajectories [46]. The background model is updated every 100 frames.

Event probability sequences for normal activities: We trained HMMs for three activities that occur around the plane: passengers embarking, passengers disembarking and luggage cart leaving the plane. During the training phase, HMM parameters are estimated using the Baum-Welch algorithm and event probability sequences are computed using the learned HMMs. Fig. 15 shows event probability sequences at different scale parameters for passengers disembarking from the airplane and proceeding to the gate. We observe that fewer number of events are detected at coarser scales as expected. At a particular scale, we may think of the events as reflecting the progress of a passenger as he/she walks from the airplane to the gate. Events partition the path into regions. If we use a left-to-right HMM model, these regions roughly correspond to the states of the HMM at a sufficiently fine scale. In other words, at the appropriate scale $\tilde{p}$, we may expect $N-1$ events, where N is the number of states at regularly spaced intervals. Given a test trajectory, we use each of the three HMMs (for the three activities) and obtain three candidate event probability sequences. We compare these candidates with the trained event sequences using DTW. All activities were correctly recognized. Not surprisingly, in this limited setting, the HMM itself (without event probability sequences) is able to correctly classify activities. The utility of event probability sequences lies in anomaly detection.

Anomaly detection using event probability sequences: Consider normal trajectories of passengers disembarking from the plane and walking toward the gate [Fig. 13(a)]. Deviations from the normal path taken by passengers may be considered anomalies. We simulate a spatial anomaly because there are no anomalies in the dataset [Fig. 13(b)]. The person violates the normal path, and walks toward the fuel truck [lower left corner of Fig. 13(b)]. The extent of deviation is controlled by a parameter σ. As the value of σ increases, the V-shape of the trajectory in Fig. 13(b) deepens, and the deviation increases.

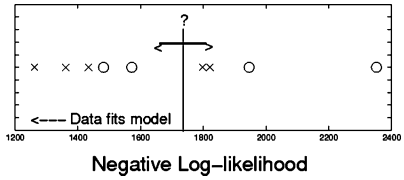

Fig. 14. Negative log likelihood for normal ("x") and anomalous ("o") instances of people walking from plane to gate in the TSA airport surveillance dataset. Values on the left end of the scale are more likely to be generated by the learned HMM.

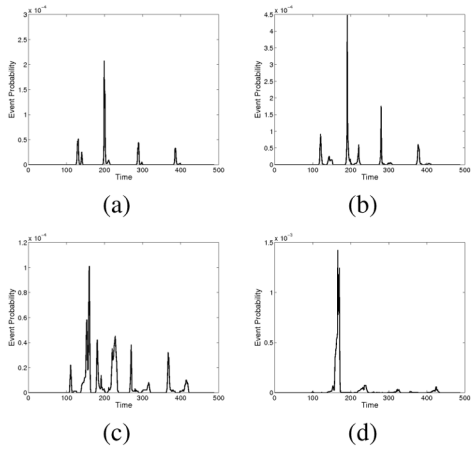

Fig. 15. Event probability sequence at different scales for normal trajectories of people deplaning. At coarse scales, we see fewer events. The scale parameter increases from (a)–(d).

Using the set of normal trajectories, a five-state HMM is trained. Fig. 14 shows a plot of negative log-likelihood for different normal (marked with a cross) and anomalous (marked with a circle) trajectories. As the value increases along the x-axis, the trajectories are less likely to be generated by the trained HMM. Using this, it may be tempting to find a threshold above which trajectories are declared anomalous. This is not feasible as illustrated in Fig. 14. Some anomalous trajectories have higher likelihood (more left on the scale) than normal trajectories. The HMM likelihood ignores anomalies that can be subtle deviations from normal trajectories. Moreover, the deviations can be confined to a part of the activity. These deviations that cause anomalies can be detected using event probability sequences.

Given a new trajectory, event probabilities are computed using the learned HMM. Anomalies are reflected in the sequence of events detected, even if they are present in a part of the trajectory. This is so because the method does not accumulate errors at all time instants, but only based on the times when events occur or when an event was expected to

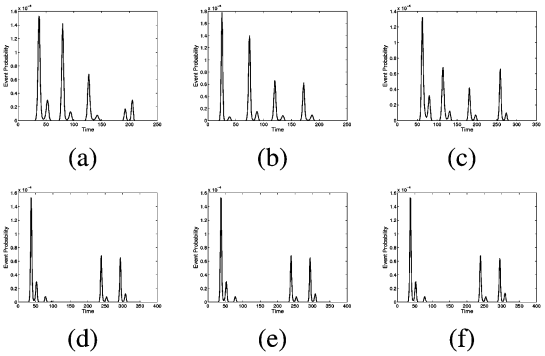

Fig. 16. (a)–(c) Event probabilities for three normal trajectories of people deplaning and walking toward the terminal. There are four dominant events in the normal trajectories, irrespective of the exact paths that people follow. (d)–(f) Event probability sequences for deviating trajectories with increasing extents of deviation $\sigma^2 = 2, 4, 16$ respectively.

occur as seen in the training data. To declare an anomaly, we use both the number of events detected (spurious and missing events are both anomalies) as well the location of the detected events. Fig. 16(d)–(f) shows the event probability sequences for a person who deviates from the normal path and later rejoins the virtual path. Fig. 16(a)–(c) shows three such sequences. We observe that the latter two dominant event probability peaks resemble the latter half of the normal event sequences, whereas a missing event in the first half indicates an anomaly.

Measuring relative speed using event probability sequence: The relative strength of events is a measure of walking speed of the passenger. Consider the event probability sequence at scale $\tilde{p}$. The event variable $\eta_t^{\tilde{p}}(i,j)$ measures the probability of persisting in state i for $\tilde{p}$ frames, transitioning to state j at time t and persisting in state j for $\tilde{p}$ frames conditioned on the observed data. A faster moving person is likely to persist to a lesser extent in either state i or j. Consequently, the strength of the event is likely to be less compared to a passenger walking at a slower speed. Whereas a passenger who stops at an intermediate point will not produce an event since the necessary state transition does not occur. The relative strength of the events in the event probability sequence bears an inverse relation to speed. In Fig. 15(a) and (b), we see that the second event is larger than the rest. This means that initially, as the passenger started walking from the plane, he/she was walking slowly. Gradually, as the passenger approached the terminal, he/she picks up speed as the latter event probabilities show.

VIII. Summary and Future Work

A sequence of instantaneous events is said to represent an activity. It provides a compact model for recognizing activities and detecting anomalies. We formulate the event detection problem within the HMM framework and compute event probability sequences that contain the probability of event occurrence as a function of time. Event probability sequences were demonstrated to be robust to changes in viewing direction.

The event probability is based on a simple step edge with a certain support region, i.e., for $p = 3$ we have an event if the posterior probability of the state sequence, say $Q = \{2,2,2,1,1,1\}$, attains a local maximum. We categorize events based on the pair of states (2,1) involved in the transition, the region of support and the probability of the transition. This can be modified to include more complex patterns so that the events of various types can be detected; for example, an event of the type $Q = \{2,1,2,1,2,1\}$. Instead of assuming a family of events, discovering event patterns using training data is a challenging problem.

Appendix I

The calculations to derive event probability variables in (8) and (11) are shown here. For the $p = 2$ case, we show the algebraic manipulations needed to derive (8). The proof for (11) is by induction. The calculations use Bayes rule, Markov property of states and conditional independence of data given the current state of the HMM. For compactness, let $q_{t_1}^{t_2} = i$ represent the state sequence $\{q_{t_1} = i, q_{t_1+1} = i, \ldots, q_{t_2} = i\}$. Let $o_{t_1}^{t_2}$ denote the observation sequence $\{o_{t_1}, o_{t_1+1}, \ldots, o_{t_2}\}$. Let $b_i\left(o_{t_1}^{t_2}\right)$ denote the product of output probabilities $b_i(o_{t_1})b_i(o_{t_1+1})\ldots b_i(o_{t_2})$. Let $V^{(p)} = P(O|\lambda)\eta_t^{(p)}(i,j)$.

For the base case $p = 2$, using (7), we have

$$V^{(2)} = P\left(q_{t-1}^{t} = i, q_{t+1}^{t+2} = j, o_1^{t-1}, o_t^T|\lambda\right) \tag{14}$$

$$\begin{aligned}
&= P\left(q_{t-1} = i, o_1^{t-1}|\lambda\right) \\
&\quad \times P\left(q_t = i, q_{t+1}^{t+2} = j, o_t^T|q_{t-1} = i, \lambda\right) \\
&= \alpha_{t-1}(i)P(q_t = i|q_{t-1} = i|\lambda) \\
&\quad \times P\left(q_{t+1}^{t+2} = j, o_t^T|q_t = i, \lambda\right) \\
&= \alpha_{t-1}(i)a_{ii}P(o_t|q_t = i, \lambda) \\
&\quad \times P\left(q_{t+1}^{t+2} = j, o_{t+1}^T|q_t = i, \lambda\right) \\
&= \alpha_{t-1}(i)a_{ii}b_i(o_t)P\left(q_{t+1}^{t+2} = j, o_{t+1}^T|q_t = i, \lambda\right) \\
&= \alpha_{t-1}(i)a_{ii}b_i(o_t)P(q_{t+1} = j|q_t = i, \lambda) \\
&\quad \times P\left(q_{t+2} = j, o_{t+1}^T|q_{t+1} = j, \lambda\right) \\
&= \alpha_{t-1}(i)a_{ii}b_i(o_t)a_{ij}P(o_{t+1}|q_{t+1} = j, \lambda) \\
&\quad \times P\left(q_{t+2} = j, o_{t+2}^T|q_{t+1} = j, \lambda\right) \\
&= \alpha_{t-1}(i)a_{ii}b_i(o_t)a_{ij}b_j(o_{t+1}) \\
&\quad \times P(q_{t+2} = j|q_{t+1} = j, \lambda)P\left(o_{t+2}^T|q_{t+2} = j, \lambda\right) \\
&= \alpha_{t-1}(i)a_{ii}b_i(o_t)a_{ij}b_j(o_{t+1})a_{jj} \\
&\quad \times P(o_{t+2}|q_{t+2} = j, \lambda)P\left(o_{t+3}^T|q_{t+2} = j, \lambda\right) \\
&= \alpha_{t-1}(i)a_{ii}b_i(o_t)a_{ij}b_j(o_{t+1})a_{jj}b_j(o_{t+2}) \\
&\quad \times \beta_{t+2}(j).
\end{aligned} \tag{15}$$

Rearranging (15), we get (8). To extend the above calculation for $\eta_t^{(p)}$, $p > 2$, (14) is written as follows:

$$\begin{aligned}
V^{(2)} &= P\left(q_{t-1} = i, o_1^{t-1}|\lambda\right) \\
&\quad \times P\left(q_t = i, q_{t+1}^{t+2} = j, o_t^{t+2}|q_{t-1} = i, \lambda\right) \\
&\quad \times P\left(o_{t+3}^T|q_{t+2} = j, \lambda\right)
\end{aligned} \tag{16}$$

$$= \alpha_{t-1}(i)N^{(2)}\beta_{t+2}(j) \tag{17}$$

where $N^{(2)} = P\left(q_t = i, q_{t+1}^{t+2} = j, o_t^{t+2}|q_{t-1} = i, \lambda\right)$.

Comparing (15) and (17), $N^{(2)} = a_{ii}b_i(o_t)a_{ij}b_j(o_{t+1})a_{jj}b_j(o_{t+2})$.

Assume that (11) is true for some $p = k$, i.e.,

$$\begin{aligned} V^{(k)} &= P\left(q_{t-k+1}^{t} = i, q_{t+1}^{t+k} = j, o_1^T|\lambda\right) \\ &= P\left(q_{t-k+1} = i, o_1^{t-k+1}|\lambda\right) N^{(k)} \\ &\quad \times P\left(o_{t+k+1}^{T}|q_{t+k} = j, \lambda\right) \\ &= \alpha_{t-k+1}(i) N^{(k)} \beta_{t+k}(j) \end{aligned} \tag{18}$$

$$\begin{aligned} \text{where } N^{(k)} &= P\Bigg(q_{t-k+2}^{t} = i, q_{t+1}^{t+k} = j \\ &\qquad o_{t-k+2}^{t+k}|q_{t-k+1} = i, \lambda\Bigg) \\ &= a_{ii}^{k-1} b_i\left(o_{t-k+2}^{t}\right) a_{ij} b_j\left(o_{t+1}^{t+k}\right) a_{jj}^{k-1}. \end{aligned} \tag{19}$$

To prove (11), we have to show that $V^{(k)} \Longrightarrow V^{(k+1)}$. Consider the case $p = k + 1$

$$\begin{aligned} V^{(k+1)} &= P\left(q_{t-k}^{t} = i, q_{t+1}^{t+k+1} = j, o_1^T|\lambda\right) \\ &= P\left(q_{t-k} = i, o_1^{t-k}|\lambda\right) \\ &\quad \times P\left(q_{t-k+1}^{t} = i, q_{t+1}^{t+k+1} = j, o_{t-k+1}^{t+k+1}|q_{t-k} = i, \lambda\right) \\ &\quad \times P\left(o_{t+k+2}^{T}|q_{t+k+1} = j, \lambda\right) \\ &= \alpha_{t-k}(i) N^{(k+1)} \beta_{t+k+1}(j) \\ N^{(k+1)} &= P(q_{t-k+1} = i, o_{t-k+1}|q_{t-k} = i, \lambda) \\ &\quad \times P\left(q_{t-k+2}^{t} = i, q_{t+1}^{t+k} = j, o_{t-k+2}^{t+k}|q_{t-k+1} = i, \lambda\right) \\ &\quad \times P\left(q_{t+k+1} = j, o_{t+k+1}|q_{t+k} = j, \lambda\right). \end{aligned} \tag{20}$$

Using (19) to substitute for the second term in (20) and expanding the first and third terms of (20), we get

$$\begin{aligned} N^{(k+1)} &= P(q_{t-k+1} = i|q_{t-k} = i, \lambda) \\ &\quad \times P(o_{t-k+1}|q_{t-k+1} = i, \lambda) N^{(k)} \\ &\quad \times P(q_{t+k+1} = j|q_{t+k} = j, \lambda) \\ &\quad \times P(o_{t+k+1}|q_{t+k+1} = j, \lambda) \\ &= a_{ii} b_i(o_{t-k+1}) N^{(k)} a_{jj} b_j(o_{t+k+1}) \\ &= a_{ii}^{k} b_i\left(o_{t-k+1}^{t}\right) a_{ij} b_j\left(o_{t+1}^{t+k+1}\right) a_{jj}^{k}. \end{aligned} \tag{21}$$

From (21), $V^{(k)} \Longrightarrow V^{(k+1)}$ and the proof is complete.

APPENDIX II

Proposition 1 (Sufficient Condition for Event Sequence to be View Invariant): For the event probability sequences to be invariant under changing viewing conditions, the associated HMMs must be conforming.

Proof: The basic idea of the proof is that for every event of type (k_i, l_i) detected for a viewing direction, there must be a corresponding event of type $(\tilde{k}_i, \tilde{l}_i)$ when viewed from a different direction.

Let $\lambda = (A, B, \Pi)$ be an HMM that generates output symbols O, and $X(\lambda)$ the associated topological space. The structure of the probability transition matrix A and the state distributions B reflect the topology of the HMM. Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be an affine transformation of O. We can construct another HMM $\tilde{\lambda}$, with the same number of states, that generates $\tilde{O} = f(O)$ such that $F : X(\lambda) \to X(\tilde{\lambda})$ is a homeomorphism.

1) By construction, the two HMMs have the same topology (number of states, model structure etc). Since the two sets of observation O and $\tilde{O}$ are related by a nontrivial affine map, we may write $P(O \in S/\lambda) = P(f(O) \in \tilde{S}/\tilde{\lambda})$ for some open S. So, the map F is surjective. If $O_1, O_2 \in X(\lambda)$ are both mapped to $\tilde{O} \in X(\tilde{\lambda})$ under F, choose the observation sequence that is generated by the optimal state sequence. So, the mapping F is made injective. 2) Every open set U in $X(\tilde{\lambda})$ has an open set $F^{-1}(U) \in X(\lambda)$ as long as the dimension of O and $\tilde{O}$ are the same. Every open ball in $X(\lambda)$ is transformed to an open ball in $X(\tilde{\lambda})$ as long as the dimension of O and $\tilde{O} = f(O)$ are the same. It follows that F is continuous. 3) By interchanging the position of (O, λ) and $(\tilde{O}, \tilde{\lambda})$, we see that F^{-1} is continuous. From 1)–3), F is a homeomorphism. We see that these assumptions may break down at orthogonal viewing directions.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the associate editor for valuable suggestions and A. Veeraraghavan for helpful discussions.

REFERENCES

[1] A. Kojima, T. Tamura, and K. Fukunaga, "Natural language description of human activities from video images based on concept of hierarchy of actions," *Int. J. Comput. Vis.*, vol. 50, no. 2, pp. 171–184, Feb. 2002.

[2] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 852–872, Aug. 2000.

[3] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia, "Event detection and analysis from video streams," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 844–851, Aug. 2001.

[4] A. R. J. Francois, R. Nevatia, J. Hobbs, and R. C. Bolles, "VERL: An ontology framework for representing and annotating video events," *IEEE Trans. Multimedia*, vol. 12, no. 4, pp. 76–86, Oct. 2005.

[5] V. T. Vu, F. Bremond, and M. Thonnat, "Temporal constraints for video interpretation," presented at the Eur. Conf. Artificial Intelligence, 2002.

[6] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden Markov model," in *Proc. IEEE CVPR*, 1992, pp. 994–999.

[7] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden Markov models for complex action recognition," in *Proc. IEEE CVPR*, 1992, pp. 379–385.

[8] T. Izo and W. E. L. Grimson, "Simultaneous pose estimation and camera calibration from multiple views," in *Proc. IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, 2004, vol. 1, pp. 14–21.

[9] T. Starner and A. Pentland, "Real-time american sign language recognition from video using hidden Markov models," in *Proc. Symp. Computer Vision*, 1995, pp. 265–270.

[10] D. Koller and U. Lerner, "Sampling in factored dynamic systems," in *Sequential Monte Carlo Methods in Practice*. New York: Springer, 2001, pp. 445–464.

[11] Y. Shi and A. F. Bobick, "P-net: A representation for partially-sequenced, multi-stream activity," in *Proc. IEEE Workshop on Event Mining*, Madison, WI, 2003, vol. 4, pp. 40–47.

[12] C. Rao, A. Yilmaz, and M. Shah, "View-invariant representation and recognition of actions," *Int. J. Comput. Vis.*, vol. 50, no. 2, pp. 203–226, 2002.

[13] D. D. Vecchio, R. M. Murray, and P. Perona, "Decomposition of human motion into dynamics based primitives with application to drawing tasks," *Automatica*, vol. 39, no. 12, pp. 2085–2098, 2003.

[14] A. Kale, A. N. Rajagopalan, A. Sundaresan, N. P. Cuntoor, A. K. Roy-Chowdhury, V. Kruger, and R. Chellappa, "Identification of humans using gait," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1163–1173, Sep. 2004.

[15] A. F. Bobick and J. Davis, "Real-time recognition of activity using temporal templates," in *Proc. IEEE Workshop Appl. Comput. Vis.*, 1996, pp. 39–42.

[16] H. Zhong, J. Shi, and M. Visontai, "Detecting unusual activity in video," in *Proc. IEEE CVPR*, 2004, pp. 819–826.

[17] N. Vaswani, A. R. Chowdhury, and R. Chellappa, "Activity recognition using the dynamics of the configuration of interacting objects," in *Proc. IEEE CVPR*, 2003, vol. 2, pp. 633–639.
[18] T. Xiang and S. Gong, "Video behavior profiling and abnormality detection without manual labeling," in *Proc. ICCV*, 2005, pp. 1238–1245.
[19] W. Chen and S. F. Chang, "Motion trajectory matching of video objects," in *Proc. SPIE Storage and Retrieval for Media Databases*, 2000, pp. 544–553.
[20] T. Syeda-Mahmood, "Segmenting actions in velocity curve space," in *Proc. IEEE ICPR*, 2002, vol. 4, pp. 170–175.
[21] R. Chellappa, N. P. Cuntoor, S. W. Joo, V. Subrahmanian, and P. Turaga, "Computational vision approaches for event modeling," presented at the Understanding Events: How Humans See, Represent, and Act on Events.
[22] S. Tsuji, A. Morizono, and S. Kuroda, "Understanding a simple cartoon film by a computer vision system," in *Proc. IJCAI*, 1977, pp. 609–610.
[23] L. Zelnik-Manor and M. Irani, "Event based analysis of video," in *Proc. IEEE CVPR*, 2001, vol. 2, pp. 123–130.
[24] V. Parameswaran and R. Chellappa, "View invariants for human action recognition," in *Proc. IEEE CVPR*, 2003, vol. 2, pp. 613–619.
[25] A. K. R. Chowdhury and R. Chellappa, "A factorization approach for activity recognition," in *Proc. IEEE Workshop on Event Mining*, 2003, vol. 4, pp. 41–46.
[26] D. Meyer, "Human gait classification based on hidden Markov models," *3D Image Anal. Synthesis*, pp. 139–146, 1997.
[27] J. Aggarwal and Q. Cai, "Human motion analysis: A review," *Comput. Vis. Image Understand.*, vol. 73, no. 3, pp. 428–440, 1999.
[28] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, "Object trajectory-based activity classification and recognition using hidden Markov models," *IEEE Trans. Image Process.*, vol. 16, no. 7, pp. 1912–1919, Jul. 2007.
[29] F. M. Porikli, "Clustering variable length sequences by eigenvector decompostion using HMM," in *Proc. Int. Workshop on Structural and Syntactic Pattern Recognition*, 2004, pp. 352–360.
[30] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, "A hybrid system for affine-invariant trajectory retrieval," in *Proc. ACM SIGMM Int. Workshop on Multimedia Information Retrieval*, 2004, pp. 235–242.
[31] J.-F. Mari, J.-P. Haton, and A. Kriouile, "Automatic word recognition based on second-order hidden Markov models," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 1, pp. 22–25, Jan. 1997.
[32] Y. Wang, L. Zhou, J. Feng, J. Wang, and Z.-Q. Liu, "Mining complex time-series data by learning Markovian models," in *Proc. IEEE Int. Conf. Data Mining*, 2006, pp. 1136–1140.
[33] Z. Ghahramani and M. I. Jordan, "Factorial hidden Markov models," *Mach. Learn.*, vol. 29, no. 2-3, pp. 245–273, 1997.
[34] M. Brand and V. Kettnaker, "Discovery and segmentation of activities in video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 873–899, Aug. 2000.
[35] N. Oliver, E. Horvitz, and A. Garg, "Layered representations for human activity recognition," in *Proc. IEEE Int. Conf. Mulitmodal Interfaces*, Pittsburgh, PA, 2002, pp. 3–7.
[36] R. Hamid, Y. Huang, and I. Essa, "ARGmode—Activity recognition using graphical models," in *Proc. IEEE CVPR*, 2003, vol. 4, pp. 38–43.
[37] T. Xiang and S. Gong, "Modeling activity and understanding behavior," *Int. J. Comput. Vis.*, vol. 67, no. 1, pp. 21–51, 2006.
[38] C. Stauffer and E. Grimson, "Learning patterns of activity using real-time tracing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747–757, Aug. 2000.
[39] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech processing," *Proc. IEEE*, vol. 63, no. 2, pp. 257–285, Feb. 1989.
[40] N. P. Cuntoor, B. Yegnanarayana, and R. Chellappa, "Interpretation of state sequences in HMM for activity representation," in *Proc. IEEE Conf. Acoustic Speech and Signal Processing*, Philadelphia, PA, 2005, vol. 2, pp. 709–712.
[41] J. Bilmes, "A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," Tech. Rep. ICSI-TR-97–021, 1997.
[42] B. H. Juang, "On the hidden Markov model and dynamic time warping for speech recognition—A unified view," *Tech. J.*, vol. 63, pp. 1213–1243, 1984.
[43] I. Csiszar and P. Shields, "Consistency of the BIC Markov order estimator," *Ann. Statist.*, vol. 28, pp. 1601–1619, 2000.
[44] A. R. Chowdhury, A. Kale, and R. Chellappa, "Towards a view invariant gait recognition algorithm," in *Proc. IEEE Int. Conf. Advanced Video and Signal Based Surveillance*, 2003, pp. 143–150.
[45] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990.
[46] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Int. J. CAI*, pp. 674–679, 1981.

Naresh P. Cuntoor (S'99–M'07) received the B.E. degree in electronics and communication engineering from the Karnataka Regional Engineering College (KREC), Surathkal, India, in 2000, and the M.S. and the Ph.D. degrees in Electrical Engineering from the University of Maryland, College Park, in 2003 and 2007, respectively. His M.S. thesis was on human identification using gait as a biometric. His Ph.D. dissertation was on statistical approaches to activity modeling in video sequences.

He is a Research Engineer with the Signal Innovations Group, Research Triangle Park, NC. His research interests are signal, image and video processing, pattern recognition, and topology and commercial applications of video processing and analytics.

B. Yegnanarayana (M'78–SM'84) received the B.Sc. degree from Andhra University, Waltair, India, in 1961, and the B.E., M.E., and Ph.D. degrees in electrical communication engineering from the Indian Institute of Science (IISc) Bangalore, India, in 1964, 1966, and 1974, respectively.

He is a Professor and Microsoft Chair at the International Institute of Information Technology (IIIT), Hyderabad. Prior to joining IIIT, he was a Professor in the Department of Computer Science and Engineering at the Indian Institute of Technology (IIT), Madras, India, from 1980 to 2006. He was the Chairman of the Department from 1985 to 1989. He was a Visiting Associate Professor of computer science at Carnegie-Mellon University, Pittsburgh, PA, from 1977 to 1980. He was a member of the faculty at the Indian Institute of Science (IISc), Bangalore, from 1966 to 1978. He has supervised 31 M.S. theses and 21 Ph.D. dissertations. His research interests are in signal processing, speech, image processing, and neural networks. He has published over 300 papers in these areas in IEEE journals and other international journals, and in the proceedings of national and international conferences. He is also the author of the book *Artificial Neural Networks* (Prentice-Hall of India, 1999).

Dr. Yegnanarayana was an Associate Editor for the IEEE TRASACTIONS ON SPEECH AND AUDIO PROCESSING from 2003 to 2006. He is a Fellow of the Indian National Academy of Engineering, a Fellow of the Indian National Science Academy, and a Fellow of the Indian Academy of Sciences. He was the recipient of the Third IETE Prof. S. V. C. Aiya Memorial Award in 1996. He received the Prof. S. N. Mitra memorial Award for the year 2006 from the Indian National Academy of Engineering.

Rama Chellappa (S'78–M'79–SM'83–F'92) received the B.E. (Hons.) degree from the University of Madras, Madras, India, in 1975 and the M.E. (Distinction) degree from the Indian Institute of Science, Bangalore, in 1977. He received the M.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1978 and 1981, respectively.

Since 1991, he has been a Professor of electrical engineering and an affiliate Professor of Computer Science at the University of Maryland, College Park. Recently, he was named the Minta Martin Professor of Engineering. He is also affiliated with the Center for Automation Research (Director) and the Institute for Advanced Computer Studies (permanent member). Prior to joining the University of Maryland, he was an Assistant Professor (1981 to 1986) and an Associate Professor (1986 to 1991) and Director of the Signal and Image Processing Institute (1988 to 1990) with the University of Southern California (USC), Los Angeles. Over the last 25 years, he has published numerous book chapters and peer-reviewed journal and conference papers. He has also coedited and coauthored many research monographs. His current research interests are face and gait analysis, 3-D modeling from video, automatic target recognition from stationary and moving platforms, surveillance and monitoring, hyperspectral processing, image understanding, and commercial applications of image processing and understanding.

Dr. Chellappa has served as an Associate Editor of four IEEE journals. He was a co-Editor-in-Chief of *Graphical Models and Image Processing*. He also served as the Editor-in-Chief of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He served as a member of the IEEE Signal Processing Society Board of Governors and as its Vice President of Awards and Membership. He has received several awards, including the National Science Foundation Presidential Young Investigator Award in 1985, three IBM Faculty Development Awards, the 1990 Excellence in Teaching Award from the School of Engineering at USC, the 1992 Best Industry Related Paper Award from the International Association of Pattern Recognition (with Q. Zheng), and the 2000 Technical Achievement Award from the IEEE Signal Processing Society. He was elected as a Distinguished Faculty Research Fellow (1996 to 1998), and as a Distinguished Scholar-Teacher (2003) at University of Maryland. He co-authored a paper that received the Best Student Paper in the Computer Vision Track at the International Association of Pattern Recognition in 2006. He is a co-recipient of the 2007 Outstanding Innovator of the Year Award (with A. Sundaresan) from the Office of Technology Commercialization at University of Maryland and received the A. J. Clark School of Engineering Faculty Outstanding Research Award. He was recently elected to serve as a Distinguished Lecturer of the Signal Processing Society. He is a Fellow of the International Association for Pattern Recognition. He has served as a General the Technical Program Chair for several IEEE international and national conferences and workshops. He is a Golden Core Member of IEEE Computer Society and received its Meritorious Service Award in 2004.