

A Bayesian Computer Vision System for Modeling Human Interactions

Nuria M. Oliver, Barbara Rosario, and Alex P. Pentland, *Senior Member, IEEE*

Abstract—We describe a real-time computer vision and machine learning system for modeling and recognizing human behaviors in a visual surveillance task [1]. The system is particularly concerned with detecting when interactions between people occur and classifying the type of interaction. Examples of interesting interaction behaviors include following another person, altering one's path to meet another, and so forth. Our system combines top-down with bottom-up information in a closed feedback loop, with both components employing a statistical Bayesian approach [2]. We propose and compare two different state-based learning architectures, namely, HMMs and CHMMs for modeling behaviors and interactions. The CHMM model is shown to work much more efficiently and accurately. Finally, to deal with the problem of limited training data, a synthetic “Alife-style” training system is used to develop flexible prior models for recognizing human interactions. We demonstrate the ability to use these a priori models to accurately classify real human behaviors and interactions with no additional tuning or training.

Index Terms—Visual surveillance, people detection, tracking, human behavior recognition, Hidden Markov Models.

1 INTRODUCTION

WE describe a real-time computer vision and machine learning system for modeling and recognizing human behaviors in a visual surveillance task [1]. The system is particularly concerned with detecting when interactions between people occur and classifying the type of interaction.

Over the last decade there has been growing interest within the computer vision and machine learning communities in the problem of analyzing human behavior in video ([3], [4], [5], [6], [7], [8], [9], [10]). Such systems typically consist of a low- or mid-level computer vision system to detect and segment a moving object—human or car, for example—and a higher level interpretation module that classifies the motion into “atomic” behaviors such as, for example, a pointing gesture or a car turning left.

However, there have been relatively few efforts to understand human behaviors that have substantial extent in time, particularly when they involve interactions between people. This level of interpretation is the goal of this paper, with the intention of building systems that can deal with the complexity of multiperson pedestrian and highway scenes [2].

This computational task combines elements of AI/machine learning and computer vision and presents challenging problems in both domains: from a *Computer Vision* viewpoint, it requires real-time, accurate, and robust detection and tracking of the objects of interest in an unconstrained environment; from a *Machine Learning and Artificial Intelligence* perspective behavior models for interacting agents are needed to interpret the set of perceived actions and detect eventual anomalous behaviors or

potentially dangerous situations. Moreover, all the processing modules need to be integrated in a consistent way.

Our approach to modeling person-to-person interactions is to use supervised statistical machine learning techniques to teach the system to recognize normal single-person behaviors and common person-to-person interactions. A major problem with a data-driven statistical approach, especially when modeling rare or anomalous behaviors, is the limited number of examples of those behaviors for training the models. A major emphasis of our work, therefore, is on efficient Bayesian integration of both prior knowledge (by the use of synthetic prior models) with evidence from data (by situation-specific parameter tuning). Our goal is to be able to successfully apply the system to any normal multiperson interaction situation without additional training.

Another potential problem arises when a completely new pattern of behavior is presented to the system. After the system has been trained at a few different sites, previously unobserved behaviors will be (by definition) rare and unusual. To account for such novel behaviors, the system should be able to recognize new behaviors and to build models of them from as little as a single example.

We have pursued a Bayesian approach to modeling that includes both *prior* knowledge and *evidence* from data, believing that the Bayesian approach provides the best framework for coping with small data sets and novel behaviors. Graphical models [11], such as Hidden Markov Models (HMMs) [12] and Coupled Hidden Markov Models (CHMMs) [13], [14], [15], seem most appropriate for modeling and classifying human behaviors because they offer dynamic time warping, a well-understood training algorithm, and a clear Bayesian semantics for both individual (HMMs) and interacting or coupled (CHMMs) generative processes.

To specify the priors in our system, we have developed a framework for building and training models of the

• The authors are with the Vision and Modeling Media Laboratory MIT, Cambridge, MA 02139. E-mail: {nuria, rosario, sandy}@media.mit.edu.

Manuscript received 21 Apr. 1999; revised 10 Feb. 2000; accepted 28 Mar. 2000.

Recommended for acceptance by R. Collins.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 109636.

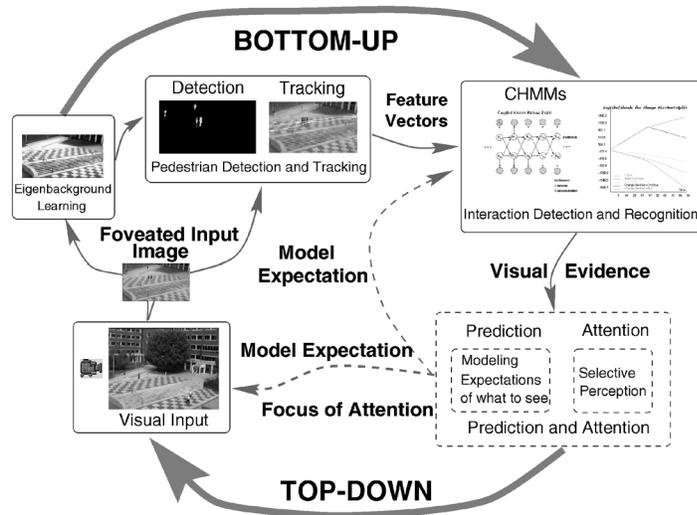


Fig. 1. Top-down and bottom-up processing loop.

behaviors of interest using *synthetic agents* [16], [17]. Simulation with the agents yields synthetic data that is used to train *prior models*. These prior models are then used recursively in a Bayesian framework to fit real behavioral data. This approach provides a rather straightforward and flexible technique to the design of priors, one that does not require strong analytical assumptions to be made about the form of the priors.¹ In our experiments, we have found that by combining such synthetic priors with limited real data we can easily achieve very high accuracies of recognition of different human-to-human interactions. Thus, our system is robust to cases in which there are only a few examples of a certain behavior (such as in interaction type 2 described in Section 5) or even no examples except synthetically-generated ones.

The paper is structured as follows: Section 2 presents an overview of the system, Section 3 describes the computer vision techniques used for segmentation and tracking of the pedestrians and the statistical models used for behavior modeling and recognition are described in Section 4. A brief description of the synthetic agent environment that we have created is described in Section 5. Section 6 contains experimental results with both synthetic agent data and real video data and Section 7 summarizes the main conclusions and sketches our future directions of research. Finally, a summary of the CHMM formulation is presented in the Appendix.

2 SYSTEM OVERVIEW

Our system employs a static camera with wide field-of-view watching a dynamic outdoor scene (the extension to an active camera [18] is straightforward and planned for the next version). A real-time computer vision system segments moving objects from the learned scene. The scene description method allows variations in lighting, weather, etc., to be learned and accurately discounted.

1. Note that our priors have the same form as our posteriors, namely they are Markov models.

For each moving object an appearance-based description is generated, allowing it to be tracked through temporary occlusions and multiobject meetings. A Kalman filter tracks the objects' location, coarse shape, color pattern, and velocity. This temporally ordered stream of data is then used to obtain a behavioral description of each object and to detect interactions between objects.

Fig. 1 depicts the processing loop and main functional units of our ultimate system.

1. The real-time computer vision input module detects and tracks moving objects in the scene, and for each moving object outputs a feature vector describing its motion and heading, and its spatial relationship to all nearby moving objects.
2. These feature vectors constitute the input to stochastic state-based behavior models. Both HMMs and CHMMs, with varying structures depending on the complexity of the behavior, are then used for classifying the perceived behaviors.

Note that both *top-down* and *bottom-up* streams of information would continuously be managed and combined for each moving object within the scene. Consequently, our Bayesian approach offers a mathematical framework for both combining the observations (bottom-up) with complex behavioral priors (top-down) to provide expectations that will be fed back to the perceptual system.

3 SEGMENTATION AND TRACKING

The first step in the system is to reliably and robustly detect and track the pedestrians in the scene. We use 2D *blob features* for modeling each pedestrian. The notion of "blobs" as a representation for image features has a long history in computer vision [19], [20], [21], [22], [23] and has had many different mathematical definitions. In our usage, it is a compact set of pixels that share some visual properties that are not shared by the surrounding pixels. These properties could be color, texture, brightness, motion, shading, a



Fig. 2. Background mean image, blob segmentation image, and input image with blob bounding boxes.

combination of these, or any other salient spatio-temporal property derived from the signal (the image sequence).

3.1 Segmentation by Eigenbackground Subtraction

In our system, the main cue for clustering the pixels into blobs is motion, because we have a static background with moving objects. To detect these moving objects, we adaptively build an eigenspace that models the background. This eigenspace model describes the range of appearances (e.g., lighting variations over the day, weather variations, etc.) that have been observed. The eigenspace can also be generated from a site model using standard computer graphics techniques.

The eigenspace model is formed by taking a sample of N images and computing both the mean μ_b background image and its covariance matrix C_b . This covariance matrix can be diagonalized via an eigenvalue decomposition $L_b = \Phi_b C_b \Phi_b^T$, where Φ_b is the eigenvector matrix of the covariance of the data and L_b is the corresponding diagonal matrix of its eigenvalues. In order to reduce the dimensionality of the space, in principal component analysis (PCA) only M eigenvectors (eigenbackgrounds) are kept, corresponding to the M largest eigenvalues to give a Φ_M matrix. A principal component feature vector $I_i - \Phi_{M_b}^T X_i$ is then formed, where $X_i = I_i - \mu_b$ is the mean normalized image vector.

Note that moving objects, because they don't appear in the same location in the N sample images and they are typically small, do not have a significant contribution to this model. Consequently, the portions of an image containing a moving object cannot be well-described by this eigenspace model (except in very unusual cases), whereas the static portions of the image can be accurately described as a sum of the the various eigenbasis vectors. That is, the eigenspace provides a robust model of the probability distribution function of the background, but not for the moving objects.

Once the eigenbackground images (stored in a matrix called Φ_{M_b} hereafter) are obtained, as well as their mean μ_b , we can project each input image I_i onto the space expanded by the eigenbackground images $B_i = \Phi_{M_b} X_i$ to model the static parts of the scene, pertaining to the background. Therefore, by computing and thresholding the Euclidean distance (distance from feature space DFFS [24]) between the input image and the projected image, we can detect the moving objects present in the scene: $D_i = |I_i - B_i| > t$, where t is a given threshold. Note that it is easy to *adaptively* perform the eigenbackground subtraction in order to compensate for changes such as big shadows. This motion mask is the input to a connected component algorithm that produces blob descriptions that characterize each person's shape. We have also experimented with modeling the background by using a mixture of Gaussian distributions at

each pixel, as in Pfister [25]. However, we finally opted for the eigenbackground method because it offered good results and less computational load.

3.2 Tracking

The trajectories of each blob are computed and saved into a *dynamic track memory*. Each trajectory has associated a first order Kalman filter that predicts the blob's position and velocity in the next frame. Recall that the Kalman Filter is the "best linear unbiased estimator" in a mean squared sense and that for Gaussian processes, the Kalman filter equations corresponds to the optimal Bayes' estimate.

In order to handle occlusions as well as to solve the correspondence between blobs over time, the appearance of each blob is also modeled by a Gaussian PDF in RGB color space. When a new blob appears in the scene, a new trajectory is associated to it. Thus for each blob, the Kalman-filter-generated spatial PDF and the Gaussian color PDF are combined to form a joint (x, y) image space and color space PDF. In subsequent frames, the Mahalanobis distance is used to determine the blob that is most likely to have the same identity.

4 BEHAVIOR MODELS

In this section, we develop our framework for building and applying models of individual behaviors and person-to-person interactions. In order to build effective computer models of human behaviors, we need to address the question of how knowledge can be mapped onto computation to dynamically deliver consistent interpretations.

From a strict computational viewpoint there are two key problems when processing the continuous flow of feature data coming from a stream of input video: 1) Managing the computational load imposed by frame-by-frame examination of all of the agents and their interactions. For example, the number of possible interactions between any two agents of a set of N agents is $N * (N - 1)/2$. If naively managed, this load can easily become large for even moderate N . 2) Even when the frame-by-frame load is small and the representation of each agent's instantaneous behavior is compact, there is still the problem of managing all this information over time.

Statistical directed acyclic graphs (DAGs) or probabilistic inference networks (PINs) [26], [27] can provide a computationally efficient solution to these problems. HMMs and their extensions, such as CHMMs, can be viewed as a particular, simple case of temporal PIN or DAG. PINs consist of a set of random variables represented as nodes as well as directed edges or links between them. They define a mathematical form of the joint or conditional PDF between



Fig. 3. A typical image of a pedestrian plaza.

the random variables. They constitute a simple graphical way of representing causal dependencies between variables. The absence of directed links between nodes implies a conditional independence. Moreover, there is a family of transformations performed on the graphical structure that has a direct translation in terms of mathematical operations applied to the underlying PDF. Finally, they are modular, i.e., one can express the joint global PDF as the product of local conditional PDFs.

PINs present several important advantages that are relevant to our problem: They can handle incomplete data as well as uncertainty; they are trainable and easy to avoid overfitting; they encode causality in a natural way; there are algorithms for both doing prediction and probabilistic inference; they offer a framework for combining prior knowledge and data; and, finally, they are modular and parallelizable.

In this paper, the behaviors we examine are generated by pedestrians walking in an open outdoor environment. Our goal is to develop a generic, compositional analysis of the observed behaviors in terms of states and transitions between states over time in such a manner that 1) the states correspond to our common sense notions of human behaviors and 2) they are immediately applicable to a wide range of sites and viewing situations. Fig. 3 shows a typical image for our pedestrian scenario.

4.1 Visual Understanding via Graphical Models: HMMs and CHMMs

Hidden Markov models (HMMs) are a popular probabilistic framework for modeling processes that have structure in time. They have a clear Bayesian semantics, efficient

algorithms for state and parameter estimation, and they automatically perform dynamic time warping. An HMM is essentially a quantization of a system's configuration space into a small number of discrete states, together with probabilities for transitions between states. A single finite discrete variable indexes the current state of the system. Any information about the history of the process needed for future inferences must be reflected in the current value of this state variable. Graphically, HMMs are often depicted "rolled-out in time" as PINs, such as in Fig. 4.

However, many interesting systems are composed of multiple interacting processes and, thus, merit a compositional representation of two or more variables. This is typically the case for systems that have structure both in time and space. Even with the correct number of states and vast amounts of data, large HMMs generally train poorly because the data is partitioned among states early (and incorrectly) during training; the Markov independence structure then ensures that the data is not shared by states, thus reinforcing any mistakes in the initial partitioning. Systems with multiple processes have states that share properties and, thus, emit similar signals. With a single state variable, Markov models are ill-suited to these problems. Even though an HMM can model any system in principle, in practice, the simple independence structure is a liability for large systems and for systems with compositional state. In order to model these interactions, a more complex architecture is needed.

4.1.1 Varieties of Couplings

Extensions to the basic Markov model generally increase the memory of the system (durational modeling), providing it with compositional state in time. We are interested in systems that have compositional state in *space*, e.g., more than one simultaneous state variable. Models with compositional state would offer conceptual advantages of parsimony and clarity, with consequent computational benefits in efficiency and accuracy. Using graphical models notation, we can construct various architectures for multi-HMM couplings offering compositional state under various assumptions of independence. It is well-known that the exact solution of extensions of the basic HMM to three or more chains is intractable. In those cases, approximation techniques are needed ([28], [29], [30], [31]). However, it is also known that there exists an exact solution for the case of two interacting chains, as it is in our case [28], [14].

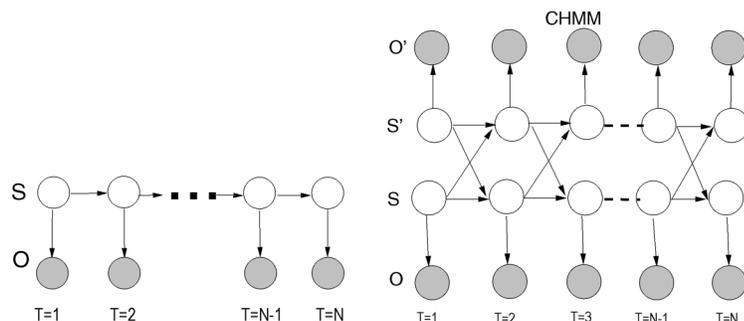


Fig. 4. Graphical representation of HMM and CHMM rolled-out in time.

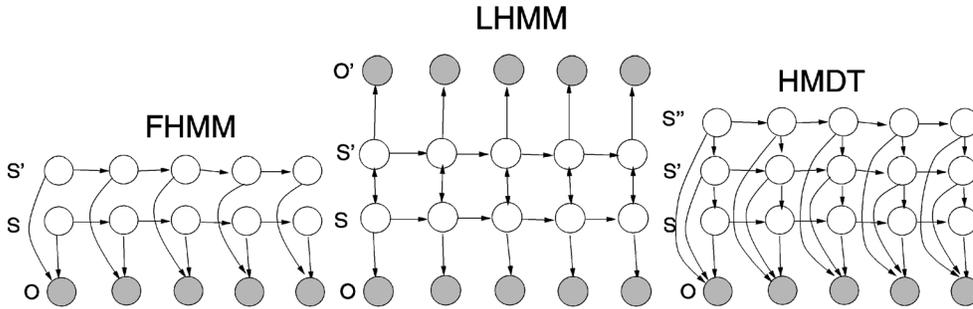


Fig. 5. Graphical representation of FHMM, LHMM, and HMDT rolled-out in time.

In particular, one can think of extending the basic HMM framework at two different levels:

1. **Coupling the outputs.** The weakest coupling is when two independent processes are coupled at the output, superimposing their outputs in a single observed signal (Fig. 5). This is known as a source separation problem: signals with zero mutual information are overlaid in a single channel. In true couplings, however, the processes are dependent and interact by influencing each other's states. One example is the sensor fusion problem: Multiple channels carry complementary information about different components of a system, e.g., acoustical signals from speech and visual features from lip tracking [32]. In [29], a generalization of HMMs with coupling at the outputs is presented. These are Factorial HMMs (FHMMs) where the state variable is factored into multiple state variables. They have a clear representational advantage over HMMs: to model C processes, each with N states, each would require an HMM with N^C joint states, typically intractable in both space and time. FHMMs are tractable in space, taking NC states, but present an inference problem equivalent to that of a combinatoric HMM. Therefore, exact solutions are intractable in time. The authors present tractable approximations using Gibbs sampling, mean field theory, or structured mean field.
2. **Coupling the states.** In [28], a statistical mechanical framework for modeling discrete time series is presented. The authors couple two HMMs to exploit the correlation between feature sets. Two parallel Boltzmann chains are coupled by weights that connect their hidden units—shown in Fig. 5 as Linked HMMs (LHMMs). Like the transition and emission weights within each chain, the coupling weights are tied across the length of the network. The independence structure of such an architecture is suitable for expressing symmetrical synchronous constraints, long-term dependencies between hidden states or processes that are coupled at different time scales. Their algorithm is based on *decimation*, a method from statistical mechanics in which the marginal distributions of singly or doubly connected nodes are integrated out. A limited class of graphs can be recursively decimated, obtaining correlations for any connected pair of nodes.

Finally, Hidden Markov Decision Trees (HMDTs) [33] are a decision tree with Markov temporal structure (see Fig. 5). The model is intractable for exact calculations. Thus, the authors use variational approximations. They consider three distributions for the approximation: one in which the Markov calculations are performed exactly and the layers of the decision tree are decoupled, one in which the decision tree calculations are performed exactly and the time steps of the Markov chain are decoupled, and one in which a Viterbi-like assumption is made to pick out a single most likely state sequence. The underlying independence structure is suitable for representing hierarchical structure in a signal, for example, the baseline of a song constrains the melody and both constrain the harmony.

We use two CHMMs for modeling two interacting processes, in our case, they correspond to individual humans. In this architecture state, chains are coupled via matrices of conditional probabilities modeling causal (temporal) influences between their hidden state variables. The graphical representation of CHMMs is shown in Fig. 4. Exact maximum a posteriori (MAP) inference is an $O(TN^4)$ computation [34], [30]. We have developed a deterministic $O(TN^2)$ algorithm for maximum entropy approximations to state and parameter values in CHMMs. From the graph it can be seen that for each chain, the state at time t depends on the state at time $t - 1$ in both chains. The influence of one chain on the other is through a causal link. The Appendix contains a summary of the CHMM formulation.

In this paper, we compare performance of HMMs and CHMMs for maximum a posteriori (MAP) state estimation. We compute the most likely sequence of states \hat{S} within a model given the observation sequence $O = \{o_1, \dots, o_n\}$. This most likely sequence is obtained by $\hat{S} = \arg \max_S P(S|O)$.

In the case of HMMs, the posterior state sequence probability $P(S|O)$ is given by

$$P(S|O) = \frac{P_{s_1} p_{s_1}(o_1) \prod_{t=2}^T P_{s_t|s_{t-1}} P_{s_t}(o_t)}{P(O)}, \quad (1)$$

where $S = \{a_1, \dots, a_N\}$ is the set of discrete states, $s_t \in S$ corresponds to the state at time t . $P_{i|j} \doteq P_{s_t=a_i|s_{t-1}=a_j}$ is the state-to-state transition probability (i.e., probability of being in state a_i at time t given that the system was in state a_j at time $t - 1$). In the following, we will write them as $P_{s_t|s_{t-1}}$. The prior probabilities for the initial state are $P_i \doteq P_{s_1=a_i} = P_{s_1}$. And, finally, $p_i(o_t) \doteq p_{s_t=a_i}(o_t) = p_{s_t}(o_t)$ are the output probabilities

for each state, (i.e., the probability of observing o_t given state a_i at time t).

In the case of CHMMs, we introduce another set of probabilities, $P_{s_t|s'_{t-1}}$, which correspond to the probability of state s_t at time t in one chain given that the other chain—denoted hereafter by superscript ' $'$ —was in state s'_{t-1} at time $t - 1$. These new probabilities express the causal influence (coupling) of one chain to the other. The posterior state probability for CHMMs is given by

$$P(S|O) = \frac{P_{s_1} p_{s_1}(o_1) P_{s'_1} p_{s'_1}(o'_1)}{P(O)} \times \prod_{t=2}^T P_{s_t|s_{t-1}} P_{s'_t|s'_{t-1}} P_{s_t|s'_{t-1}} P_{s'_t|s_{t-1}} p_{s_t}(o_t) p_{s'_t}(o'_t), \quad (2)$$

where $s_t, s'_t; o_t, o'_t$ denote states and observations for each of the Markov chains that compose the CHMMs. A coupled HMM of C chains has a joint state trellis that is in principle N^C states wide; the associated dynamic programming problem is $O(TN^2C)$. In [14], an approximation is developed using N-heads dynamic programming such that an $O(T(CN)^2)$ algorithm is obtained that closely approximates the full combinatoric result.

Coming back to our problem of modeling human behaviors, two persons (each modeled as a generative process) may interact without wholly determining each others' behavior. Instead, each of them has its own internal dynamics and is influenced (either weakly or strongly) by others. The probabilities $P_{s_t|s'_{t-1}}$ and $P_{s'_t|s_{t-1}}$ describe this kind of interactions and CHMMs are intended to model them in as efficient a manner as possible.

5 SYNTHETIC BEHAVIORAL AGENTS

We have developed a framework for creating synthetic agents that mimic human behavior in a virtual environment [16], [17]. The agents can be assigned different behaviors and they can interact with each other as well. Currently, they can generate five different interacting behaviors and various kinds of individual behaviors (with no interaction). The parameters of this virtual environment are modeled on the basis of a real pedestrian scene from which we obtained measurements of typical pedestrian movement.

One of the main motivations for constructing such synthetic agents is the ability to generate synthetic data which allows us to determine which Markov model architecture will be best for recognizing a new behavior (since it is difficult to collect real examples of rare behaviors). By designing the synthetic agents models such that they have the best generalization and invariance properties possible, we can obtain flexible prior models that are transferable to real human behaviors with little or no need of additional training. The use of synthetic agents to generate robust behavior models from very few real behavior examples is of special importance in a visual surveillance task, where typically the behaviors of greatest interest are also the most rare.

5.1 Agent Architecture

Our dynamic multiagent system consists of some number of agents that perform some specific behavior from a set of possible behaviors. The system starts at time zero, moving discretely forward to time T or until the agents disappear from the scene.

The agents can follow three different paths with two possible directions, as illustrated in Figs. 6 and 7 by the yellow paths.² They walk with random speeds within an interval; they appear at random instances of time. They can slow down, speed up, stop, or change direction independently from the other agents on the scene. Their velocity is normally distributed around a mean that increases or decreases when they slow down or speed up. When certain preconditions are satisfied a specific interaction between two agents takes place. Each agent has perfect knowledge of the world, including the position of the other agents.

In the following, we will describe without loss of generality, the two-agent system that we used for generating prior models and synthetic data of agents interactions. Each agent makes its own decisions depending on the type of interaction, its location, and the location of the other agent on the scene. There is no scripted behavior or a priori knowledge of what kind of interaction, if any, is going to take place. The agents' behavior is determined by the perceived contextual information: current position, relative position of the other agent, speeds, paths they are in, directions of walk, etc., as well as by its own repertoire of possible behaviors and triggering events. For example, if one agent decides to "follow" the other agent, it will proceed on its own path increasing its speed progressively until reaching the other agent, that will also be walking on the same path. Once the agent has been reached, they will adapt their mutual speeds in order to keep together and continue advancing together until exiting the scene.

For each agent the position, orientation, and velocity is measured, and from this data a feature vector is constructed which consists of: \dot{d}_{12} , the derivative of the relative distance between two agents; $\alpha_{1,2} = \text{sign}(\langle v_1, v_2 \rangle)$, or degree of alignment of the agents, and $v_i = \sqrt{\dot{x}^2 + \dot{y}^2}$, $i = 1, 2$, the magnitude of their velocities. Note that such a feature vector is invariant to the absolute position and direction of the agents and the particular environment they are in.

5.2 Agent Behaviors

The agent behavioral system is structured in a hierarchical way. There are *primitive or simple behaviors* and *complex interactive behaviors* to simulate the human interactions.

In the experiments reported in Section 4, we considered five different interacting behaviors that appear illustrated in Figs. 6 and 7:

1. Follow, reach, and walk together (inter1): The two agents happen to be on the same path walking in the

2. The three paths were obtained by statistical analysis of the most frequent paths that the pedestrians in the observed plaza followed. Note, however, that the performance of neither the computer vision nor the tracking modules is limited to these three paths.

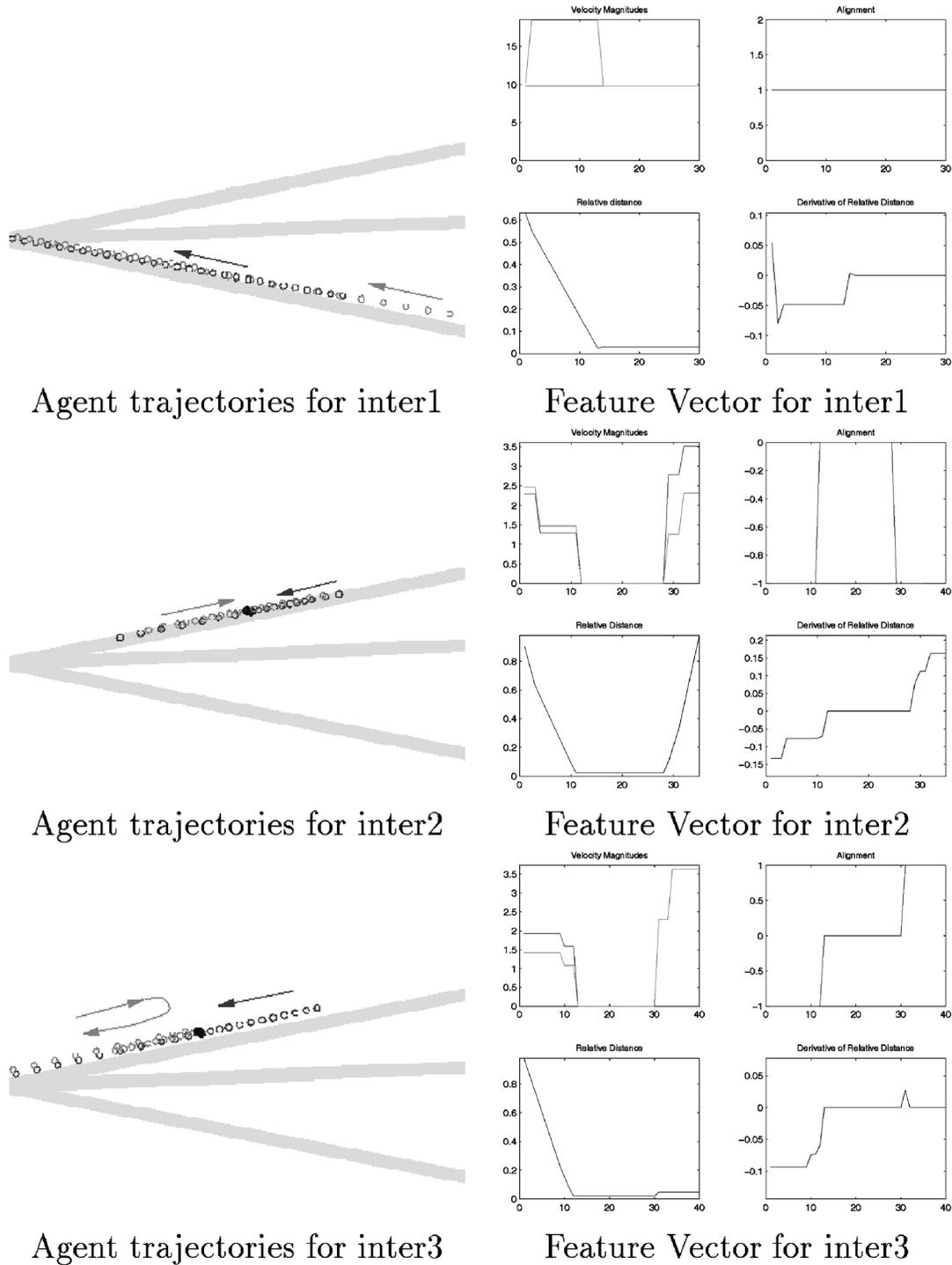


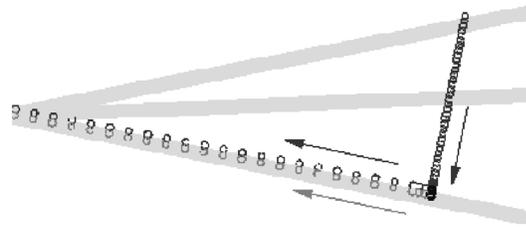
Fig. 6. Example trajectories and feature vector for the interactions: follow, approach+meet+continue separately, and approach+meet+continue together.

same direction. The agent behind decides that it wants to reach the other. Therefore, it speeds up in order to reach the other agent. When this happens, it slows down such that they keep walking together with the same speed.

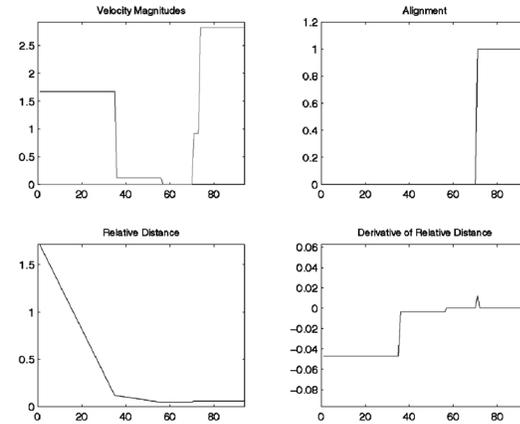
2. Approach, meet, and go on separately (inter2): The agents are on the same path, but in the opposite direction. When they are close enough, if they realize that they “know” each other, they slow down and

finally stop to chat. After talking they go on separately, becoming independent again.

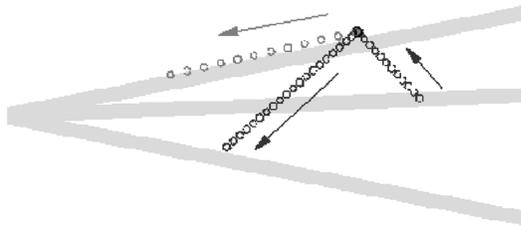
3. Approach, meet, and go on together (inter3): In this case, the agents behave like in “inter2,” but now after talking they decide to continue together. One agent therefore, changes, its direction to follow the other.
4. Change direction in order to meet, approach, meet, and continue together (inter4): The agents start on different paths. When they are close enough they can



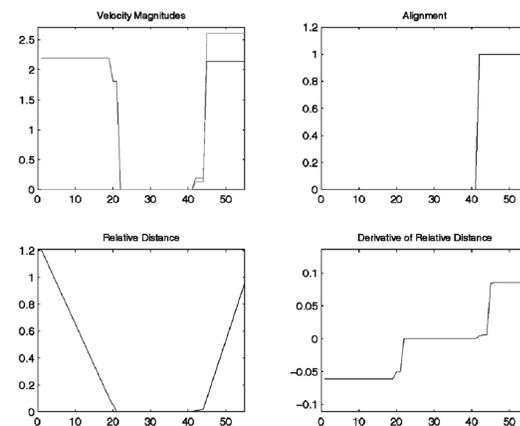
Agent trajectories for inter4



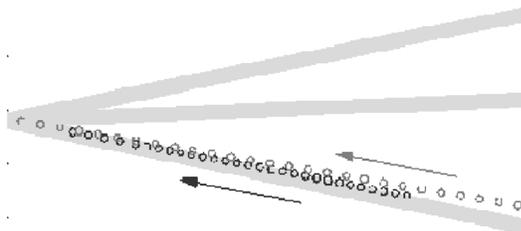
Feature Vector for inter4



Agent trajectories for inter5



Feature Vector for inter5



Agent trajectories for no interaction Feature Vector for no interaction

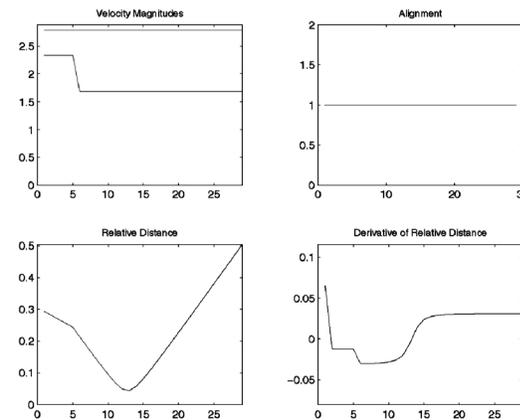


Fig. 7. Example trajectories and feature vector for the interactions: change direction+approach+meet+continue separately, change direction+approach+meet+continue together, and no interacting behavior.

see each other and decide to interact. One agent waits for the other to reach it. The other changes direction in order to go toward the waiting agent. Then they meet, chat for some time, and decide to go on together.

5. Change direction in order to meet, approach, meet, and go on separately (inter5): This interaction is the same as "inter4" except that when they decide to go on after talking, they separate, becoming independent.

Proper design of the interactive behaviors requires the agents to have knowledge about the position of each other as well as synchronization between the successive individual behaviors activated in each of the agents. Fig. 8 illustrates the timeline and synchronization of the simple behaviors and events that constitute the interactions.

These interactions can happen at any moment in time and at any location, provided only that the preconditions for the interactions are satisfied. The speeds they walk at,

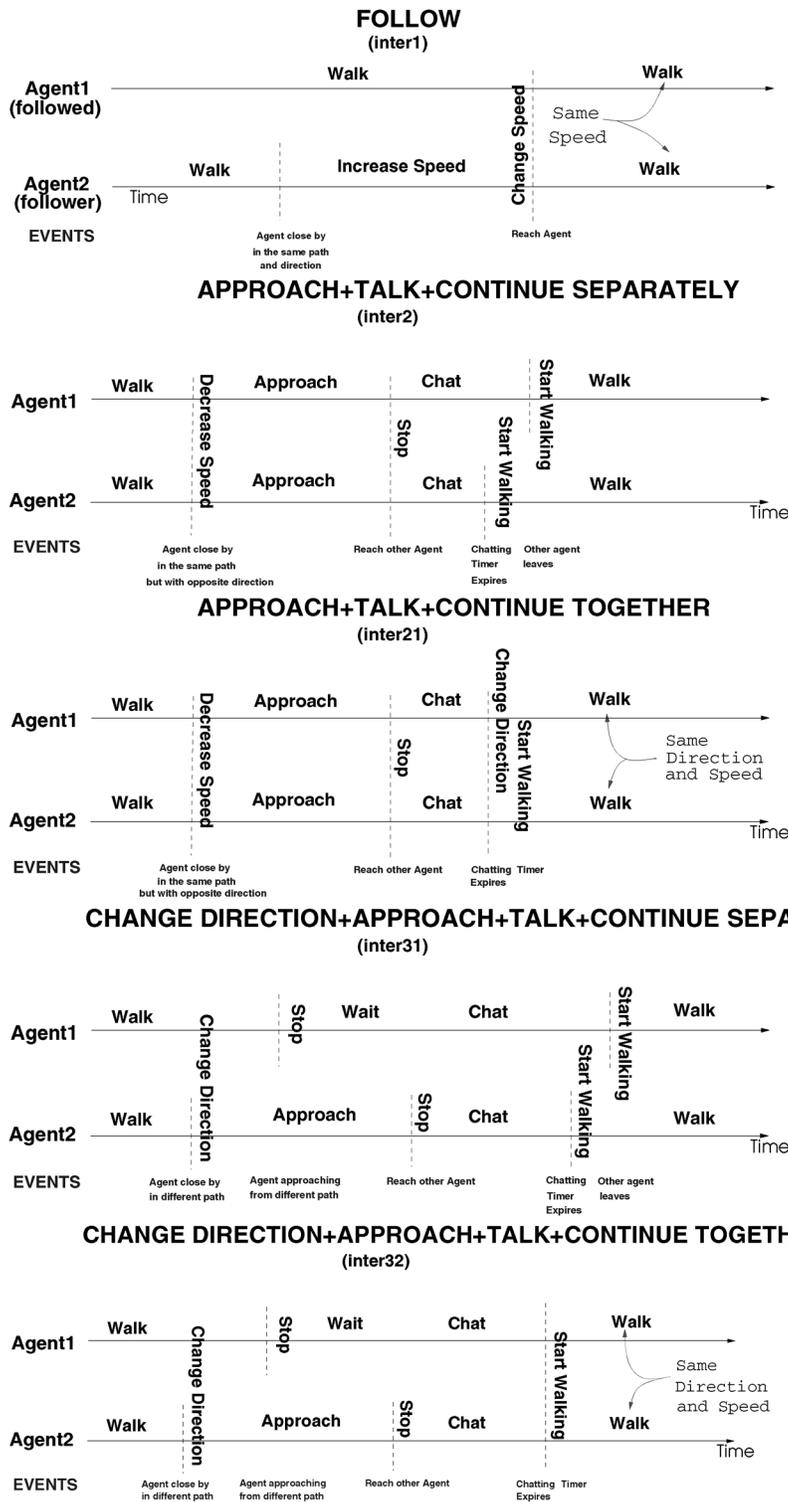


Fig. 8. Timeline of the five complex behaviors in terms of events and simple behaviors.

the duration of their chats, the changes of direction, the starting and ending of the actions vary highly. This high variance in the quantitative aspects of the interactions confers robustness to the learned models that tend to capture only the invariant parts of the interactions. The invariance reflects the nature of their interactions and the environment.

6 EXPERIMENTAL RESULTS

Our goal is to have a system that will accurately interpret behaviors and interactions within almost any pedestrian scene with little or no training. One critical problem, therefore, is generation of models that capture our prior knowledge about human behavior. The selection of priors is one of the most controversial and open issues in Bayesian inference. As we have already described, we solve this

problem by using a synthetic agents modeling package, which allows us to build flexible prior behavior models.

6.1 Comparison of CHMM and HMM Architectures with Synthetic Agent Data

We built models of the five previously described synthetic agent interactions with both CHMMs and HMMs. We used two or three states per chain in the case of CHMMs and three to five states in the case of HMMs (accordingly to the complexity of the various interactions). The particular number of states for each architecture was determined using 10 percent cross validation. Because we used the same amount of data for training both architectures, we tried keeping the number of parameters to estimate roughly the same. For example, a three state ($N = 3$) per chain CHMM with three-dimensional ($d = 3$) Gaussian observations has $(CN)^2 + N * (d + d!) = (2 * 3)^2 + 3 * (3 + 6) = 36 + 27 = 63$ parameters. A five state ($N = 5$) HMM with six-dimensional ($d = 6$) Gaussian observations has $N^2 + N * (d + d!) = 5^2 + 5 * (3 + 6) = 25 + 45 = 70$ parameters to estimate.

Each of these architectures corresponds to a different physical hypothesis: CHMMs encode a spatial coupling in time between two agents (e.g., a nonstationary process) whereas HMMs model the data as an isolated, stationary process. We used from 11 to 75 sequences for training each of the models, depending on their complexity, such that we avoided overfitting. The optimal number of training examples, of states for each interaction, as well as the optimal model parameters were obtained by a 10 percent cross-validation process. In all cases, the models were set up with a full state-to-state connection topology, so that the training algorithm was responsible for determining an appropriate state structure for the training data. The feature vector was six-dimensional in the case of HMMs, whereas in the case of CHMMs, each agent was modeled by a different chain, each of them with a three-dimensional feature vector. The feature vector was the same as the one described for the synthetic agents, namely \dot{d}_{12} , the derivative of the relative distance between two persons; $\alpha_{1,2} = \text{sign}(\langle v_1, v_2 \rangle)$, or degree of alignment of the people, and $v_i = \sqrt{\dot{x}^2 + \dot{y}^2}$, $i = 1, 2$, the magnitude of their velocities.

To compare the performance of the two previously described architectures, we used the best trained models to classify 20 unseen new sequences. In order to find the most likely model, the Viterbi algorithm was used for HMMs and the N-heads dynamic programming forward-backward propagation algorithm for CHMMs.

Table 1 illustrates the accuracy for each of the two different architectures and interactions. Note the superiority of CHMMs versus HMMs for classifying the different interactions and, more significantly, identifying the case in which there were no interactions present in the testing data.

Complexity in time and space is an important issue when modeling dynamic time series. The number of degrees of freedom (state-to-state probabilities+output means+output covariances) in the largest best-scoring model was 85 for HMMs and 54 for CHMMs. We also performed an analysis of the accuracies of the models and architectures with

TABLE 1
Accuracy for HMMs and CHMMs on Synthetic Data

Accuracy on synthetic test data (%)		
	HMMs	CHMMs
No inter	68.7	90.9
Inter1	87.5	100
Inter2	85.4	100
Inter3	91.6	100
Inter4	77	100
Inter5	97.9	100

Accuracy at recognizing when no interaction occurs ("No inter"), and accuracy at classifying each type of interaction: "Inter1" is follow, reach, and walk together; "Inter2" is approach, meet, and go on; "Inter3" is approach, meet, and continue together; "Inter4" is change direction to meet, approach, meet, and go together and "Inter5" is change direction to meet, approach, meet, and go on separately.

respect to the number of sequences used for training. Efficiency in terms of training data is especially important in the case of online real-time learning systems—such as ours would ultimately be—and/or in domains in which collecting clean labeled data may be difficult.

The cross-product HMMs that result from incorporating both generative processes into the same joint-product state space usually requires many more sequences for training because of the larger number of parameters. In our case, this appears to result in an accuracy ceiling of around 80 percent for any amount of training that was evaluated, whereas for CHMMs we were able to reach approximately 100 percent accuracy with only a small amount of training. From this result, it seems that the CHMMs architecture, with two coupled generative processes, is more suited to the problem of modeling the behavior of interacting agents than a generative process encoded by a single HMM.

In a visual surveillance system, the *false alarm* rate is often as important as the classification accuracy. In an ideal automatic surveillance system, all the targeted behaviors should be detected with a close-to-zero false alarm rate, so that we can reasonably alert a human operator to examine them further. To analyze this aspect of our system's performance, we calculated the system's ROC curve. Fig. 9 shows that it is quite possible to achieve very low false alarm rates while still maintaining good classification accuracy.

6.2 Pedestrian Behaviors

Our goal is to develop a framework for detecting, classifying, and learning generic models of behavior in a visual surveillance situation. It is important that the models be generic, applicable to many different situations, rather than being tuned to the particular viewing or site. This was one of our main motivations for developing a virtual agent environment for modeling behaviors. If the synthetic agents are "similar" enough in their behavior to humans, then the same models that were trained with synthetic data should be directly applicable to human data. This section describes the experiments we have performed analyzing real pedes-

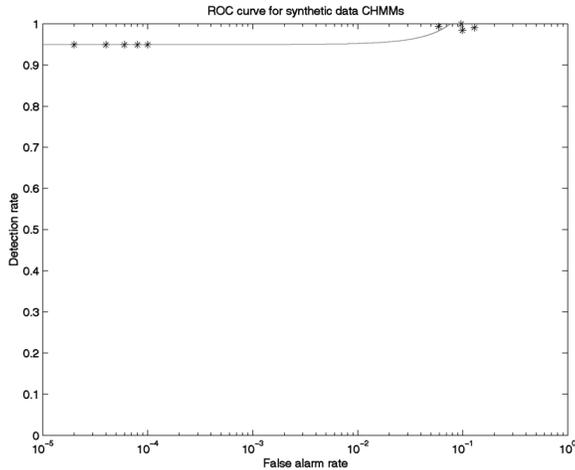


Fig. 9. ROC curve on synthetic data.

trian data using both synthetic and site-specific models (models trained on data from the site being monitored).

6.2.1 Data Collection and Preprocessing

Using the person detection and tracking system described in Section 3, we obtained 2D blob features for each person in several hours of video. Up to 20 examples of *following* and various types of *meeting* behaviors were detected and processed.

The feature vector \bar{x} coming from the computer vision processing module consisted of the 2D (x, y) centroid (mean position) of each person’s blob, the Kalman Filter state for each instant of time, consisting of $(\hat{x}, \hat{x}, \hat{y}, \hat{y})$, where $\hat{\cdot}$ represents the filter estimation, and the (r, g, b) components of the mean of the Gaussian fitted to each blob in color space. The frame-rate of the vision system was of about 20-30 Hz on an SGI R10000 O2 computer. We low-pass filtered the data with a 3Hz cutoff filter and computed for every pair of nearby persons a feature vector consisting of: d_{12} , derivative of the relative distance between two persons, $|v_i|, i = 1, 2$, norm of the velocity vector for each person, $\alpha = \text{sign}(\langle v_1, v_2 \rangle)$, or degree of alignment of the trajectories of each person. Typical

trajectories and feature vectors for an “approach, meet, and continue separately” behavior (interaction 2) are shown in Fig. 10. This is the same type of behavior as “inter2” displayed in Fig. 6 for the synthetic agents. Note the similarity of the feature vectors in both cases.

Even though multiple pairwise interactions could potentially be detected and recognized, we only had examples of one interaction taking place at a time. Therefore, all our results refer to single pairwise interaction detection.

6.2.2 Behavior Models and Results

CHMMs were used for modeling three different behaviors: meet and continue together (interaction 3), meet and split (interaction 2), and follow (interaction 1). In addition, an *interaction* versus *no interaction* detection test was also performed. HMMs performed much worse than CHMMs and, therefore, we omit reporting their results.

We used models trained with two types of data:

1. Prior-only (synthetic data) models: that is, the behavior models learned in our synthetic agent environment and then directly applied to the real data with *no additional training or tuning of the parameters*.
2. Posterior (synthetic-plus-real data) models: new behavior models trained by using as starting points the synthetic best models. We used eight examples of each interaction data from the specific site.

Recognition accuracies for both these “prior” and “posterior” CHMMs are summarized in Table 2. It is noteworthy that with only eight training examples, the recognition accuracy on the real data could be raised to 100 percent. This result demonstrates the ability to accomplish extremely rapid refinement of our behavior models from the initial prior models.

Finally, the ROC curve for the posterior CHMMs is displayed in Fig. 11.

One of the most interesting results from these experiments is the high accuracy obtained when testing the a priori models obtained from synthetic agent simulations. The fact that a priori models transfer so well to real data

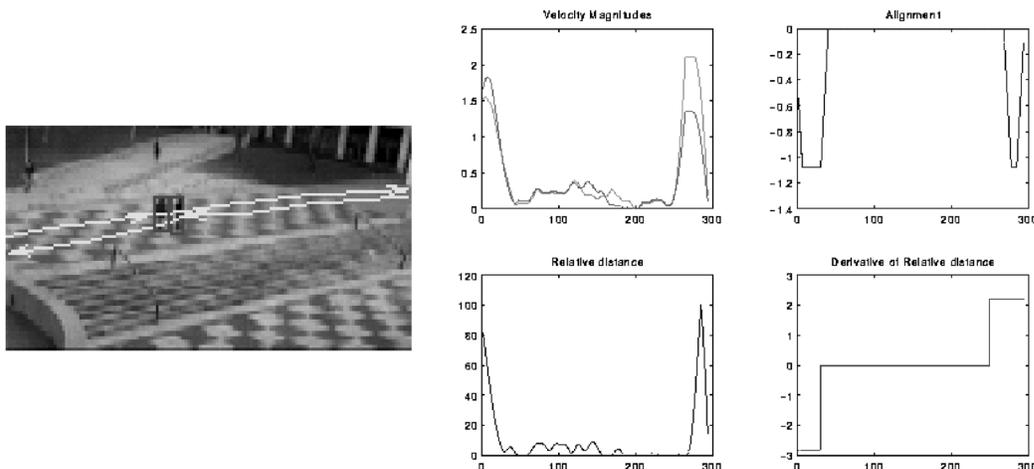


Fig. 10. Example trajectories and feature vector for interaction 2, or approach, meet, and continue separately behavior.

TABLE 2
Accuracy for Both Untuned, a Priori Models, and Site-Specific CHMMs Tested on Real Pedestrian Data

Accuracy on real pedestrian test data (%)		
	Prior CHMMs	Posterior CHMMs
No-inter	90.9	100
Inter1	93.7	100
Inter2	100	100
Inter3	100	100

The first entry in each column is the interaction versus no-interaction accuracy, the remaining entries are classification accuracies between the different interacting behaviors. Interactions are: "Inter1" follow, reach, and walk together; "Inter2" approach, meet, and go on; "Inter3" approach, meet, and continue together.

demonstrates the robustness of the approach. It shows that with our synthetic agent training system, we can develop models of many different types of behavior—thus avoiding the problem of limited amount of training data—and apply these models to real human behaviors without additional parameter tuning or training.

6.2.3 Parameter Sensitivity

In order to evaluate the sensitivity of our classification accuracy to variations in the model parameters, we trained a set of models where we changed different parameters of the agents' dynamics by factors of 2.5 and five. The performance of these altered models turned out to be virtually the same in every case except for the "inter1" (follow) interaction, which seems to be sensitive to people's velocities. Only when the agents' speeds were within the range of normal (average) pedestrian walking speeds "inter1" (follow) was correctly recognized.

7 SUMMARY AND CONCLUSIONS

In this paper, we have described a computer vision system and a mathematical modeling framework for recognizing different human behaviors and interactions in a visual surveillance task. Our system combines top-down with bottom-up information in a closed feedback loop, with both components employing a statistical Bayesian approach.

Two different state-based statistical learning architectures, namely, HMMs and CHMMs have been proposed and compared for modeling behaviors and interactions. The superiority of the CHMM formulation has been demonstrated in terms of both training efficiency and classification accuracy. A synthetic agent training system has been created in order to develop flexible and interpretable prior behavior models and we have demonstrated the ability to use these a priori models to accurately classify real behaviors with no additional tuning or training. This fact is especially important, given the limited amount of training data available.

The presented CHMM framework is not limited to only two interacting processes. Interactions between more than two people could potentially be modeled and recognized.

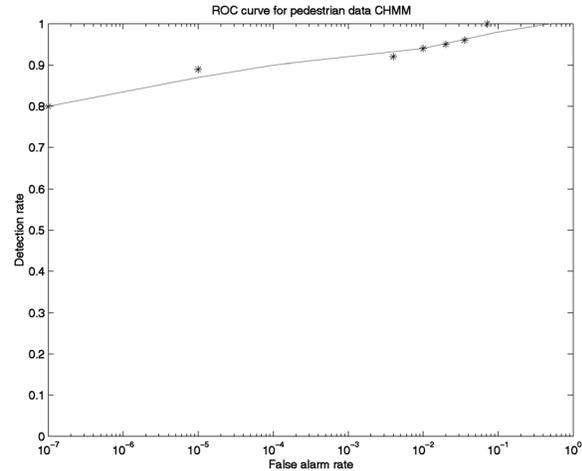


Fig. 11. ROC curve for real pedestrian data.

APPENDIX

FORWARD (α) AND BACKWARD (β) EXPRESSIONS FOR CHMMs

In [14], a deterministic approximation for maximum a posterior (MAP) state estimation is introduced. It enables fast classification and parameter estimation via expectation maximization and also obtains an upper bound on the cross entropy with the full (combinatoric) posterior, which can be minimized using a subspace that is linear in the number of state variables. An "N-heads" dynamic programming algorithm samples from the $O(N)$ highest probability paths through a compacted state trellis, with complexity $O(T(CN)^2)$ for C chains of N states apiece observing T data points. For interesting cases with limited couplings, the complexity falls further to $O(TCN^2)$.

For HMMs, the forward-backward or Baum-Welch algorithm provides expressions for the α and β variables, whose product leads to the *likelihood* of a sequence at each instant of time. In the case of CHMMs, two state-paths have to be followed over time for each chain: one path corresponds to the "head" (represented with subscript "h") and another corresponds to the "sidekick" (indicated with subscript "k") of this head. Therefore, in the new forward-backward algorithm the expressions for computing the α and β variables will incorporate the probabilities of the head and sidekick for each chain (the second chain is indicated with '). As an illustration of the effect of maintaining multiple paths per chain, the traditional expression for the α variable in a single HMM:

$$\alpha_{j,t+1} = \left[\sum_{i=1}^N \alpha_{i,t} P_{ij} \right] p_i(o_t) \quad (3)$$

will be transformed into a pair of equations, one for the full posterior α^* and another for the marginalized posterior α :

$$\alpha_{i,t}^* = p_i(o_t) p_{k',t}(o_t) \sum_j P_{i|h_{j,t-1}} P_{i|k',t-1} P_{k',t|h_{j,t-1}} P_{k',t|k_{j,t-1}} \alpha_{j,t-1}^* \quad (4)$$

$$\alpha_{i,t} = p_i(o_t) \sum_j P_{i|h_{j,t-1}} P_{i|k_{j,t-1}} \sum_g p_{k_{g,t}}(o_t) P_{k_{g,t}|h_{j,t-1}} P_{k_{g,t}|k_{j,t-1}} \alpha_{j,t-1}^* \quad (5)$$

The β variable can be computed in a similar way by tracing back through the paths selected by the forward analysis. After collecting statistics using N-heads dynamic programming, transition matrices within chains are reestimated according to the conventional HMM expression. The coupling matrices are given by:

$$P_{s'_t=i, s_{t-1}=j|O} = \frac{\alpha_{j,t-1} P_{i'|j} P_{s'_t=i}(o'_t) \beta_{i',t}}{P(O)} \quad (6)$$

$$\hat{P}_{i'|j} = \frac{\sum_{t=2}^T P_{s'_t=i, s_{t-1}=j|O}}{\sum_{t=2}^T \alpha_{j,t-1} \beta_{j,t-1}} \quad (7)$$

ACKNOWLEDGMENTS

The authors would like to thank Michael Jordan, Tony Jebara, and Matthew Brand for their inestimable help and insightful comments.

REFERENCES

- [1] N. Oliver, B. Rosario, and A. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions," *Proc. Int'l Conf. Vision Systems*, 1999.
- [2] N. Oliver, "Towards Perceptual Intelligence: Statistical Modeling of Human Individual and Interactive Behaviors," PhD thesis, Massachusetts Institute of Technology (MIT), Media Lab, Cambridge, Mass., 2000.
- [3] T. Darrell and A. Pentland, "Active Gesture Recognition Using Partially Observable Markov Decision Processes," *Int'l Conf. Pattern Recognition*, vol. 5, p. C9E, 1996.
- [4] A.F. Bobick, "Computers Seeing Action," *Proc. British Machine Vision Conf.*, vol. 1, pp. 13-22, 1996.
- [5] A. Pentland and A. Liu, "Modeling and Prediction of Human Behavior," *Defense Advanced Research Projects Agency*, pp. 201-206, 1997.
- [6] H. Buxton and S. Gong, "Advanced Visual Surveillance Using Bayesian Networks," *Int'l Conf. Computer Vision*, June 1995.
- [7] H.H. Nagel, "From Image Sequences Toward Conceptual Descriptions," *IVC*, vol. 6, no. 2, pp. 59-74, May 1988.
- [8] T. Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S. Russel, and J. Weber, "Automatic Symbolic Traffic Scene Analysis Using Belief Networks," *Proc. 12th Nat'l Conf. Artificial Intelligence*, pp. 966-972, 1994.
- [9] C. Castel, L. Chaudron, and C. Tessier, "What is Going On? A High Level Interpretation of Sequences of Images," *Proc. Workshop on Conceptual Descriptions from Images, European Conf. Computer Vision*, pp. 13-27, 1996.
- [10] J.H. Fernyhough, A.G. Cohn, and D.C. Hogg, "Building Qualitative Event Models Automatically from Visual Input," *Proc. Int'l Conf. Computer Vision*, pp. 350-355, 1998.
- [11] W.L. Buntine, "Operations for Learning with Graphical Models," *J. Artificial Intelligence Research*, 1994.
- [12] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257-285, 1989.
- [13] M. Brand, N. Oliver, and A. Pentland, "Coupled Hidden Markov Models for Complex Action Recognition," *Proc. IEEE Computer Vision and Pattern Recognition*, 1996.
- [14] M. Brand, "Coupled Hidden Markov Models for Modeling Interacting Processes," *Neural Computation*, Nov. 1996.
- [15] N. Oliver, B. Rosario, and A. Pentland, "Graphical Models for Recognizing Human Interactions," *Proc. Neural Information Processing Systems*, Nov. 1998.
- [16] N. Oliver, B. Rosario, and A. Pentland, "A Synthetic Agent System for Modeling Human Interactions," Technical Report, Vision and Modeling Media Lab, MIT, Cambridge, Mass., 1998. <http://whitechapel.media.mit.edu/pub/tech-reports>.
- [17] B. Rosario, N. Oliver, and A. Pentland, "A Synthetic Agent System for Modeling Human Interactions," *Proc. AA*, 1999.
- [18] R.K. Bajcsy, "Active Perception vs. Passive Perception," *Proc. CASE Vendor's Workshop*, pp. 55-62, 1985.
- [19] A. Pentland, "Classification by Clustering," *Proc. IEEE Symp. Machine Processing and Remotely Sensed Data*, 1976.
- [20] R. Kauth, A. Pentland, and G. Thomas, "Blob: An Unsupervised Clustering Approach to Spatial Preprocessing of MSS Imagery," *11th Int'l Symp. Remote Sensing of the Environment*, 1977.
- [21] A. Bobick and R. Bolles, "The Representation Space Paradigm of Concurrent Evolving Object Descriptions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 146-156, Feb. 1992.
- [22] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time Tracking of the Human Body," *Photonics East, SPIE*, vol. 2,615, 1995.
- [23] N. Oliver, F. Berard, and A. Pentland, "Lafter: Lips and Face Tracking," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition (CVPR '97)*, June 1997.
- [24] B. Moghaddam and A. Pentland, "Probabilistic Visual Learning for Object Detection," *Proc. Int'l Conf. Computer Vision*, pp. 786-793, 1995.
- [25] C.R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780-785, July 1997.
- [26] W.L. Buntine, "A Guide to the Literature on Learning Probabilistic Networks from Data," *IEEE Trans. Knowledge and Data Engineering*, 1996.
- [27] D. Heckerman, "A Tutorial on Learning with Bayesian Networks," Technical Report MSR-TR-95-06, Microsoft Research, Redmond, Wash., 1995, revised June 1996.
- [28] L.K. Saul and M.I. Jordan, "Boltzmann Chains and Hidden Markov Models," *Proc. Neural Information Processing Systems*, G. Tesauro, D.S. Touretzky, and T.K. Leen, eds., vol. 7, 1995.
- [29] Z. Ghahramani and M.I. Jordan, "Factorial Hidden Markov Models," *Proc. Neural Information Processing Systems*, D.S. Touretzky, M.C. Mozer, and M.E. Hasselmo, eds., vol. 8, 1996.
- [30] P. Smyth, D. Heckerman, and M. Jordan, "Probabilistic Independence Networks for Hidden Markov Probability Models," AI memo 1565, MIT, Cambridge, Mass., Feb. 1996.
- [31] C. Williams and G.E. Hinton, "Mean Field Networks That Learn to Discriminate Temporally Distorted Strings," *Proc. Connectionist Models Summer School*, pp. 18-22, 1990.
- [32] D. Stork and M. Hennecke, "Speechreading: An Overview of Image Procassing, Feature Extraction, Sensory Integration and Pattern Recognition Techniques," *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, 1996.
- [33] M.I. Jordan, Z. Ghahramani, and L.K. Saul, "Hidden Markov Decision Trees," *Proc. Neural Information Processing Systems*, D.S. Touretzky, M.C. Mozer, and M.E. Hasselmo, eds., vol. 8, 1996.
- [34] F.V. Jensen, S.L. Lauritzen, and K.G. Olesen, "Bayesian Updating in Recursive Graphical Models by Local Computations," *Computational Statistical Quarterly*, vol. 4, pp. 269-282, 1990.



Nuria M. Oliver received the BSc (honors) and MSc degrees in electrical engineering and computer science from ETSIT at the Universidad Politecnica of Madrid (UPM), Spain, 1994. She received the PhD degree in media arts and sciences from Massachusetts Institute of Technology (MIT), Cambridge, in June 2000. She is a researcher in the Vision and Modeling Group at the Media Laboratory of MIT, where she works with professor Alex Pentland. Before starting her

PhD at MIT, she worked as a research engineer at Telefonica I+D. Her research interests are computer vision, statistical machine learning, artificial intelligence, and human computer interaction. Currently, she is working on the previous disciplines in order build computational models of human behavior via perceptually intelligent systems.

Barbara Rosario was a visiting researcher in the Vision and Modeling Group at the Media Laboratory of the Massachusetts Institute of Technology. Currently, she is a computer science graduate student at the University of California, Berkeley.



Alex P. Pentland is the academic head of the MIT Media Laboratory. He is also the Toshiba professor of media arts and sciences, an endowed chair last held by Marvin Minsky. His recent research focus includes understanding human behavior in video, including face, expression, gesture, and intention recognition, as described in the April 1996 issue of *Scientific American*. He is also one of the pioneers of wearable computing, a founder of the IEEE wearable computer technical area, and general chair of the upcoming IEEE International Symposium on Wearable Computing. He has won awards from the AAAI, IEEE, and Ars Electronica. He is a founder of the IEEE wearable computer technical area and general chair of the upcoming IEEE International Symposium on Wearable Computing. He is a senior member of the IEEE.