A Bayesian Computer Vision System for Modeling Human Interactions

Nuria Oliver[1], Barbara Rosario[1] and Alex Pentland[1]

[1] Vision and Modeling. Media Laboratory, MIT,
Cambridge, MA 02139, USA
{nuria,rosario,sandy}@media.mit.edu
http://nuria.www.media.mit.edu/~nuria/humanBehavior/humanBehavior.html

Abstract. We describe a real-time computer vision and machine learning system for modeling and recognizing human behaviors in a visual surveillance task. 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. 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.

1 Introduction

We describe a real-time computer vision and machine learning system for modeling and recognizing human behaviors in a visual surveillance task. 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 ([10],[3],[21], [8], [17], [14],[9], [11]). 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 multi-person pedestrian and highway scenes.

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 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 multi-person 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 such new behaviors, and to build models of the behavior from as 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 [6], such as Hidden Markov Models (HMMs) [22] and Coupled Hidden Markov Models (CHMMs) [5, 4, 19], 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 behaviors of interest using *synthetic agents*. 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

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

5.1) 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. Section 5 contains experimental results with both synthetic agent data and real video data, and section 6 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 [1] 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.

For each moving object an appearance-based description is generated, allowing it to be tracked though temporary occlusions and multi-object 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.

Figure 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 are continuously 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 2-D *blob features* for modeling each pedestrian. The notion of "blobs" as a representation for image features has a long history in computer vision [20, 15, 2, 26, 18], and has had many different mathematical

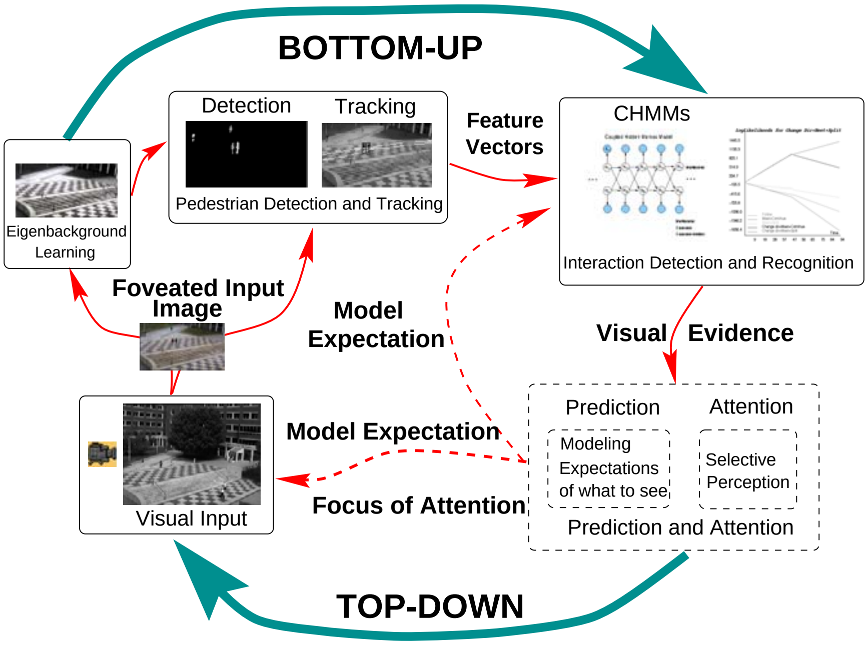


Fig. 1. Top-down and bottom-up processing loop

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 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 [16]) 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 Pfinder [27]. 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

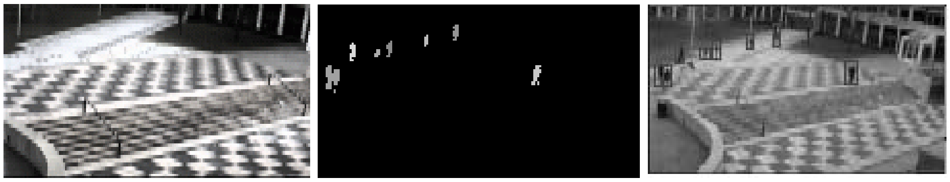


Fig. 2. Background mean image, blob segmentation image and input image with blob bounding boxes

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) [7, 13] 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 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. Figure 3 shows a typical image for our pedestrian scenario.

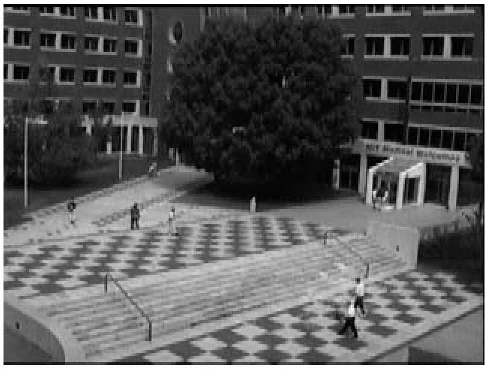


Fig. 3. A typical image of a pedestrian plaza

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 figure 4.

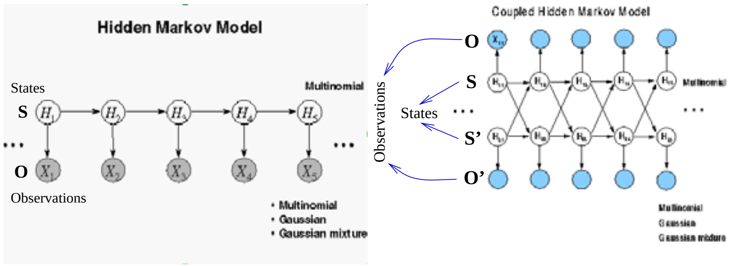


Fig. 4. Graphical representation of HMM and CHMM rolled-out in time

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. With a single state variable, Markov models are ill-suited to these problems. In order to model these interactions a more complex architecture is needed.

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. It is well known that the exact solution of extensions of the basic HMM to 3 or more chains is intractable. In those cases approximation techniques are needed ([23, 12, 24, 25]). However, it is also known that there exists an exact solution for the case of 2 interacting chains, as it is our case [23, 4].

We therefore use two Coupled Hidden Markov Models (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 figure 4. 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 $O = \{o_1, \ldots, o_n\}$. This most likely sequence is obtained by $\hat{S} = argmax_S P(S|O)$.

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

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

where $S = \{a_1, \ldots, a_N\}$ 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 need to 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

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

where $s_t, s'_t; o_t, o'_t$ denote states and observations for each of the Markov chains that compose the CHMMs. We direct the reader to [4] for a more detailed description of the MAP estimation in CHMMs.

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 is possible.

5 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. To address this problem we have created a synthetic agents modeling package which allows us to build flexible prior behavior models.

5.1 Synthetic Agents Behaviors

We have developed a framework for creating synthetic agents that mimic human behavior in a virtual environment. The agents can be assigned different behaviors and they can interact with each other as well. Currently they can generate 5 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 (by hand) 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.

In the experiments reported here, we considered five different interacting behaviors: (1) Follow, reach and walk together (inter1), (2) Approach, meet and go on separately (inter2), (3) Approach, meet and go on together (inter3), (4) Change direction in order to meet, approach, meet and continue together (inter4), and (5) Change direction in order to meet, approach, meet and go on separately (inter5).

Note that we assume that these interactions can happen at any moment in time and at any location, provided only that the preconditition s for the interactions are satisfied.

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; $\alpha_{1,2} = sign(< v_1, v_2 >)$, 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 feature vector is invariant to the absolute position and direction of the agents and the particular environment they are in.

Figure 5 illustrates the agents trajectories and associated feature vector for an example of interaction 2, i.e. an 'approach, meet and continue separately' behavior.

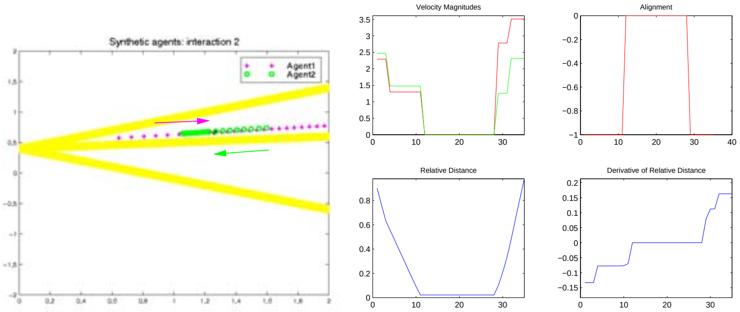


Fig. 5. Example trajectories and feature vector for interaction 2, or approach, meet and continue separately behavior.

Comparison of CHMM and HMM Architectures We built models of the previously described interactions with both CHMMs and HMMs. We used 2 or 3 states per chain in the case of CHMMs, and 3 to 5 states in the case of HMMs (accordingly to the complexity of the various interactions). Each of these architectures corresponds to a different physical hypothesis: CHMMs encode a spatial coupling in time between two agents (e.g., a non-stationary 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% 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 6-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 3-dimensional feature vector.

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.

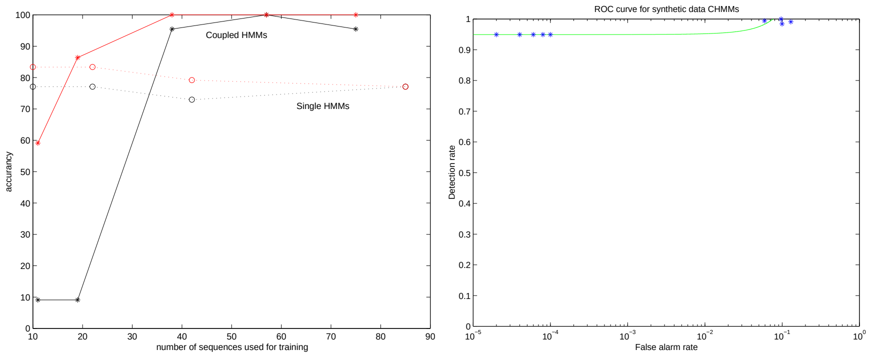
Table 5.1 illustrates the accuracy for each of the two different architectures and interactions. Note the superiority of CHMMs versus HMMs for classifying the different interactions and, more significantly, identifying the case in which there are no interactions present in the testing data.

Table 1. Accuracy for HMMs and CHMMs on synthetic data. 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

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

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 respect to the number of sequences used for training. Figure 5.1 illustrates the accuracies in the case of interaction 4 (change direction for meeting, stop and continue together). Efficiency in terms of training data is specially important in the case of on-line real-time learning systems -such as ours would ultimately be-, specially in domains in which collecting clean labeled data is 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 a accuracy ceiling of around 80% for any amount of training



Accuracies of CHMMs (solid line) and HMMs (dotted line).

ROC curve on synthetic data.

Fig. 6. *First figure:* Accuracies of CHMMs (solid line) and HMMs (dotted line) for one particular interaction. The dashed line is the accuracy on testing without considering the case of no interaction, while the dark line includes this case. *Second figure:* ROC curve on synthetic data.

that was evaluated, whereas for CHMMs we were able to reach approximately 100% 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 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. Figure 5.1 shows that it is quite possible to achieve very low false alarm rates while still maintaining good classification accuracy.

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

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 $(\hat{x}, \dot{\hat{x}}, \hat{y}, \dot{\hat{y}})$, where $\hat{.}$ 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: $\dot{d}_{12}$, derivative of the relative distance between two persons, $|v_i|, i = 1, 2$, norm of the velocity vector for each person, $\alpha = sign(< v_1, v_2 >)$, 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 figure 7. This is the same type of behavior as the one displayed in figure 5 for the synthetic agents. Note the similarity of the feature vectors in both cases.

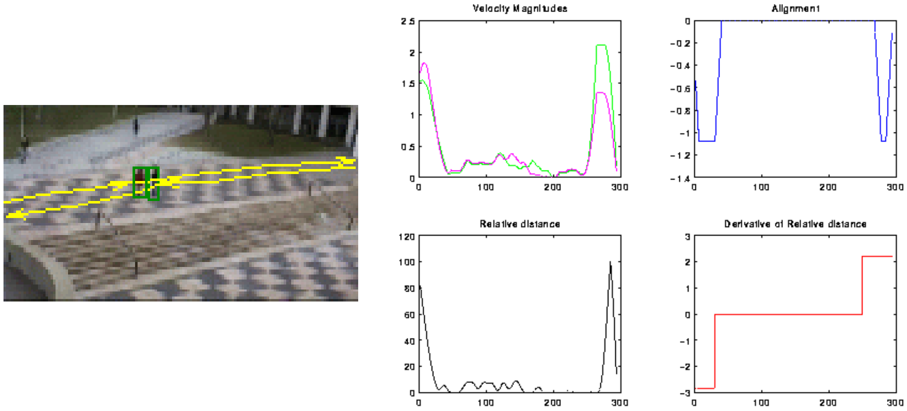


Fig. 7. Example trajectories and feature vector for interaction 2, or approach, meet and continue separately behavior.

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 8 examples of each interaction data from the specific site.

Recognition accuracies for both these 'prior' and 'posterior' CHMMs are summarized in table 5.2. It is noteworthy that with only 8 training examples, the recognition accuracy on the real data could be raised to 100%. This results demonstrates the ability to accomplish extremely rapid refinement of our behavior models from the initial prior models.

Table 2. Accuracy for both untuned, a priori models and site-specific CHMMs tested on real pedestrian data. The first entry in each row is the interaction vs no-interaction detection 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.

Testing on real pedestrian data				
	No-inter	Inter1	Inter2	Inter3
Prior CHMMs	90.9	93.7	100	100
Posterior CHMMs	100	100	100	100

Finally the ROC curve for the posterior CHMMs is displayed in figure 8.

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 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 — avoiding thus the problem of limited amount of training data — and apply these models to real human behaviors without additional parameter tuning or training.

Parameters 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 5. 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 relative rates of movement.

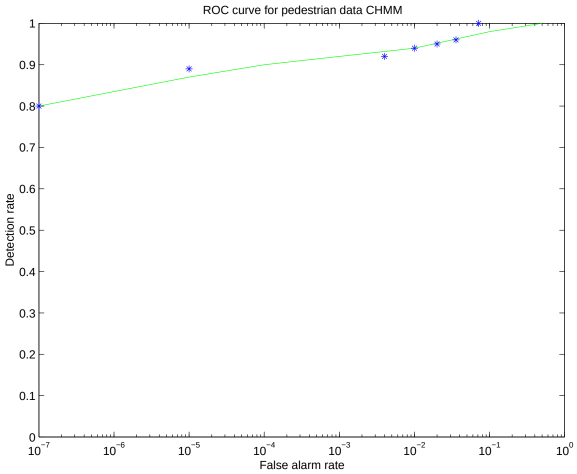


Fig. 8. ROC curve for real pedestrian data

6 Summary, Conclusions and Future Work

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 specially important, given the limited amount of training data available.

Acknowledgments

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

References

1. R.K. Bajcsy. Active perception vs. passive perception. In *CVWS85*, pages 55–62, 1985.

2. A. Bobick and R. Bolles. The representation space paradigm of concurrent evolving object descriptions. *PAMI*, pages 146–156, February 1992.
3. A.F. Bobick. Computers seeing action. In *Proceedings of BMVC*, volume 1, pages 13–22, 1996.
4. Matthew Brand. Coupled hidden markov models for modeling interacting processes. *Submitted to Neural Computation*, November 1996.
5. Matthew Brand, Nuria Oliver, and Alex Pentland. Coupled hidden markov models for complex action recognition. In *In Proceedings of IEEE CVPR97*, 1996.
6. W.L. Buntine. Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, 1994.
7. W.L. Buntine. A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering*, 1996.
8. Hilary Buxton and Shaogang Gong. Advanced visual surveillance using bayesian networks. In *International Conference on Computer Vision*, Cambridge, Massachusetts, June 1995.
9. C. Castel, L. Chaudron, and C. Tessier. What is going on? a high level interpretation of sequences of images. In *Proceedings of the workshop on conceptual descriptions from images, ECCV*, pages 13–27, 1996.
10. T. Darrell and A. Pentland. Active gesture recognition using partially observable markov decision processes. In *ICPR96*, page C9E.5, 1996.
11. J.H. Fernyhough, A.G. Cohn, and D.C. Hogg. Building qualitative event models automatically from visual input. In *ICCV98*, pages 350–355, 1998.
12. Zoubin Ghahramani and Michael I. Jordan. Factorial hidden Markov models. In David S. Touretzky, Michael C. Mozer, and M.E. Hasselmo, editors, *NIPS*, volume 8, Cambridge, MA, 1996. MITP.
13. David Heckerman. A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Redmond, Washington, 1995. Revised June 96.
14. T. Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S. Russel, and J. Weber. Automatic symbolic traffic scene analysis using belief networks. pages 966–972. Proceedings 12th National Conference in AI, 1994.
15. R. Kauth, A. Pentland, and G. Thomas. Blob: An unsupervised clustering approach to spatial preprocessing of mss imagery. In *11th Int'l Symp. on Remote Sensing of the Environment*, Ann Harbor MI, 1977.
16. B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *ICCV95*, pages 786–793, 1995.
17. H.H. Nagel. From image sequences towards conceptual descriptions. *IVC*, 6(2):59–74, May 1988.
18. N. Oliver, F. Berard, and A. Pentland. Lafter: Lips and face tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR97)*, S.Juan, Puerto Rico, June 1997.
19. N. Oliver, B. Rosario, and A. Pentland. Graphical models for recognizing human interactions. In *To appear in Proceedings of NIPS98, Denver, Colorado,USA*, November 1998.
20. A. Pentland. Classification by clustering. In *IEEE Symp. on Machine Processing and Remotely Sensed Data*, Purdue, IN, 1976.
21. A. Pentland and A. Liu. Modeling and prediction of human behavior. In *DARPA97*, page 201 206, 1997.
22. Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *PIEEE*, 77(2):257–285, 1989.

23. Lawrence K. Saul and Michael I. Jordan. Boltzmann chains and hidden Markov models. In Gary Tesauro, David S. Touretzky, and T.K. Leen, editors, *NIPS*, volume 7, Cambridge, MA, 1995. MITP.
24. Padhraic Smyth, David Heckerman, and Michael Jordan. Probabilistic independence networks for hidden Markov probability models. AI memo 1565, MIT, Cambridge, MA, Feb 1996.
25. C. Williams and G. E. Hinton. Mean field networks that learn to discriminate temporally distorted strings. In *Proceedings, connectionist models summer school*, pages 18–22, San Mateo, CA, 1990. Morgan Kaufmann.
26. C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *In* Photonics East, SPIE, volume 2615, 1995. Bellingham, WA.
27. C.R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.

Appendix: Forward (α) and Backward (β) Expressions for CHMMs

In [4] 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:

$$\alpha_{j,t+1} = [\sum_{i=1}^{N} \alpha_{i,t} P_{i|j}] p_i(o_t) \tag{3}$$

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

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

$$\alpha_{i,t} = p_i(o_t) \sum_j P_{i|h_{j,t-1}} P_{i|k_{j',t-1}} \sum_g p_{k_{g',t}}(o_t) P_{k_{g',t}|h_{j,t-1}} P_{k_{g',t}|k_{j',t-1}} \alpha^*_{j,t-1} \quad (5)$$

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 re-estimated according to the conventional HMM expression. The coupling matrices are given by:

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

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

This article was processed using the LaTeX macro package with LLNCS style