Statistical modeling of human interactions

Nuria Oliver Barbara Rosario Alex Pentland
{nuria,rosario,sandy}@media.mit.edu

Vision and Modeling. Media Laboratory
MIT
Cambridge, MA 02139

Abstract

1 Introduction

In this paper we describe a statistical computer vision and machine learning system for modeling and recognizing different human behaviors, both individual and interactions, in a visual surveillance task. By *behavior* we understand a sequence of simple actions that are performed under a specific situation in order to have some effect upon the world or reach some goal. Note that this definition of behavior is intrinsically teleological.

Over the last decade there has been a growing interest in the computer vision and machine learning communities to analyzing and understanding dynamic scenes ([?], [?], [?], [?],[?], [?]). An implementation of such systems requires both *low* and *intermediate* -level visual processing combined with *higher* level interpretation of the identified objects. Nowadays vision systems are relatively successful in detecting and tracking moving objects in a scene. However, higher-level spatio-temporal reasoning, needed for the computation of behavioral descriptions, is much more rare. This level of understanding is our goal, with the intention of building perceptually more aware computers, able to better understand what is happening in a scene, and to give more human-like interpretations of it. There are several motivations for such interest, such as improve computer vision techniques, acquire a better understanding of human behavior, attain a more natural computer-human interaction or automatize repetitive tasks that are usually performed manually. Visual surveillance is one of such tasks. The goal of a visual surveillance system is the identification, tracking and interpretation of specific objects in a dynamic scene -people and vehicles in our case-.

This computational task lays between AI, machine learning and computer vision, incorporating and combining elements of both and proposing challenging research avenues in both domains: from a *computer vision* viewpoint, they require 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 models of the behaviors are needed to interpret the set of perceived actions and detect eventual anomalous behaviors and potentially dangerous situations, for example.

Any visual surveillance should not only reliably identify anomalous behaviors but also perform a complete behavior analysis. In order to fully analyze behaviors, the system should be able to detect and recognize *interesting* situations. One usual approach to such problem are supervised statistical learning techniques, where the situations considered interesting are previously presented to some learning algorithm. Therefore, during training time, the system learns the optimal (according to some optimization criterion) parameters for each of the interesting situations to be modeled. Posteriorly, during testing time, the system would be able to identify and recognize -up to some accuracy level- similar behaviors to those previously trained with. One drawback to such and approach, specially when modeling rare or anomalous behaviors, is the limited number of examples of those behaviors for training the models. Another drawback arises when a new pattern of behavior is presented to the system. Usually and in the best case, it will not recognize it as any of the known behaviors; in the worst case, it will erroneously recognize it as one of the modeled behaviors. It is of crucial importance, thus, to be able to build our models using *reduced number of samples* as well as let them *adaptively discover and incorporate new behaviors* as needed.

A Bayesian approach that includes both *prior* knowledge and *evidence* from data seems the most appropriate for this purpose. Moreover, we have developed a framework for building and training models of the behaviors of interest using *synthetic agent* data. We have been able to successfully transfer these synthetic agent *behavior models* with no need of additional trainig or tuning to the real situation with accuracies of around 90%. Therefore, even in the cases when there are only a few of examples of a certain behavior, high detection rates can be achieved.

We support the idea that bayesian graphical models are an appropriate framework for modeling and classifying dynamic behaviors. More specifically, Hidden Markov Models (HMMs) and Coupled Hidden Markov Models (CHMMs), because they offer dynamic time warping, a training algorithm and a clear Bayesian semantics for both individual (HMMs) and interacting or coupled (CHMMs) generative processes.

The paper is structured as follows: section **??** compiles the most relevant related previous work; section ?? describes the system's overview in terms of its four principal building blocks. In section ?? we present the computer vision techniques used for segmentation and tracking of the pedestrians. The statistical models used for behavior modeling and recognition are described in section ??. Next section ?? contains experimental results with both synthetic agent data and real video data. Finally section ?? sketches the main conclusions and our future directions of research.

2 Previous Work

According to the overall modeling of the system, one can distinguish different approaches to solving a visual surveillance and interpretation task:

3 System's overview

Our general goal is to build a vision system able to understand what is seeing. Our specific set-up consists of a static camera with wide field of view watching a dynamic outdoor scene. A low and mid-level vision system provides segmentation of moving objects as well as their tracking over time. It returns the locations, coarse shape and color descriptions, and velocity vectors for each of the objects. From this temporally ordered stream of visual data we would like to be able to provide a behavioral description of what is happening in the scene.

Figure 1 depicts the processing loop and main functional units of our system:

1. The computer vision processing module incorporates computer vision techniques for detecting and tracking in real time the moving objects in the scene. As a result, a feature vector is constructed that constitutes the input of next building block.

2. Different stochastic state based behavior models are built. Both HMMs and CHMMs, with varying structures depending on the complexity of the behavior, are used for classifying the perceived behaviors as well as for

3. Discovering possible new behaviors and predicting the most likely (expected) next observation. This prediction directs an

4. Attentional mechanism to direct the visual processing to specific areas of the scene where the most relevant actions are taken place.

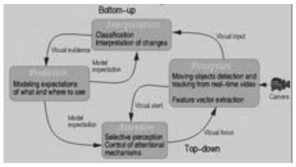
Bottom-up
Classification
Interpretation of changes
Visual input
Visual evidence
Model expectation
Modeling expectations of what and where to see
Moving objects detection and tracking from real-time video
Feature vector extraction
Camera
Visual alert
Model expectation
Selective perception
Control of attentional mechanisms
Visual focus
Top-down

Figure 1: Top-down and bottom-up processing loop

Note that both *top-down* and *bottom-up* streams of information are managed and combined in our system. In this sense, our Bayesian approach offers also a mathematical framework for both combining the observations (bottom-up) with the priors and models (top-down) to provide expectations that will be fed back to the perceptual system.

From a strict computational viewpoint there are two key problems when processing this increasing over time amount of data: (1) the computational load of determining what all the agents/objects are doing. For example, the number of possible interactions between any two agents of a set of N agents is $N*(N-1)/2$; (2) even if the representation provided by the visual processing is compact, there is the problem of managing all this information over time.

4 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 [?, ?, ?, ?], and has had many different mathematical definitions. In our usage it is a compact set of pixels that share a visual property that is not not shared by the surrounding pixels. This property 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). In our usage blobs are, therefore, a coarse, locally-adaptive encoding of the images' spatial and color/texture/motion/etc. properties. A prime motivation for our interest in blob representations is our discovery that they can be reliably detected and tracked even in complex, dynamic scenes, and that they can be extracted in real-time without the need for special purpose hardware. These properties are particularly important in applications that require tracking people, and recently we have used 2-D blob tracking for real-time whole-body human interfaces [?] and real-time face and lips tracking as well as recognition of facial expressions [?].

In modern computer vision processing we seek to group pixels of images together and to "segment" images based on visual coherence, but the "features" obtained from such efforts are usually taken to be the boundaries, or contours, of these regions rather than the regions themselves. In very complex scenes, such as those we are handling with, containing people or natural objects, contour features often prove unreliable and difficult to find and use. Moreover given the low resolution of each pedestrian in our sequences - taken by a wide angle static camera in a top-down view of an open outdoors area- a 2D blob representation offers the right framework for modeling each person.

In this method feature vectors at each pixel are formed by adding (x, y) spatial coordinates to the spectral (or textural) components of the imagery. These are then clustered so that image properties such as color and spatial similarity combine to form coherent connected regions, or "blobs," in which all the pixels have similar image properties. This blob description method is, in fact, a special case of recent Minimum Description Length (MDL) algorithms [?, ?, ?].

We can represent shapes in both 2-D and 3-D by their low-order statistics. Clusters of 2-D points have 2-D spatial means and covariance matrices, which we shall denote $\bar{q}$ and C_q. The blob spatial statistics are described in terms of their second-order properties; for computational convenience we will interpret this as a Gaussian model. The Gaussian interpretation is not terribly significant, because we also keep a pixel-by-pixel *support map* showing the actual occupancy.

Like other representations used in computer vision and signal analysis, including superquadrics, modal analysis, and eigen-representations, blobs represent the global aspects of the shape and can be augmented with higher-order statistics to attain more detail if the data supports it. The reduction of degrees of freedom from individual pixels to blob parameters is a form of regularization which allows the ill-conditioned problem to be solved in a principled and stable way.

For both 2-D and 3-D blobs, there is a useful physical interpretation of the blob parameters in the image space. The mean represents the geometric center of the blob area (2-D) or volume (3-D). The covariance, being symmetric, can be diagonalized via an eigenvalue decomposition: $C = \Phi L \Phi^T$,where Φ is orthonormal and L is diagonal.

The diagonal L matrix represents the size of the blob along independent orthogonal object-centered axes and Φ is a rotation matrix that brings this object-centered basis in alignment with the coordinate basis of C.

This decomposition and physical interpretation is important for estimation, because the shape L can vary at a different rate than the rotation Φ. The parameters must be separated so they can be treated appropriately.

4.0.1 Segmentation by Eigenbackground subtraction

In our system the main cue for clustering the pixels into blobs is motion. We have a *static* background with moving objects on it. An adaptive eigenspace of the background is constructed by taking a certain number N of images and computing both the mean μ_b background image and the 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 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 formed, where $X_i = I_i - \mu_b$ is the mean normalized image vector. 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, the Euclidean distance (distance from feature space DFFS [?]) between the input image and the projected image will correspond to the moving objects present in the scene: $D_i = |I_i - B_i|$. This motion mask is the input of a clustering algorithm in order to obtain the blobs that characterize each person.

Figure 2: Background mean image, difference image and segmentation image

4.0.2 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 in the next frame. The Kalman Filter is the 'best linear unbiased estimator' in a mean squared sense. Moreover, for Gaussian processes, the Kalman filter equations correspond to the optimal Bayes' estimate.

In order to handle occlusions as well as to solve the correspondence between blobs over time, each blob is modeled by a Gaussian pdf in RGB color space as well. When a new blob appears in the scene, a new trajectory is associated to it. This implies attaching a Kalman filter and a Gaussian pdf in both space and color spaces to the blob. In subsequent frames the most likely blob in these two spaces will be assigned to it.

In the next section we present a robust and efficient learning framework for building models of different human behaviors as well as results of applying these models in both a virtual and a real environment.

5 Behavior models

In order to build an effective computer model of behavior we need to address the question of how knowledge can be mapped onto computation to dynamically deliver consistent interpretations. This involves a deep analysis of the spatio-temporal regularities in the image data. Statistical approaches seem to be an adequate framework for modeling such regularities. More specifically, we believe that probabilistic graphical models, such as Hidden Markov Models and more complicated structures, offer the necessary knowledge representation as well as computational efficiency.

Statistical directed graphical models or probabilistic directed graphs (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

Figure 3: Typical highway and pedestrian images

a family of transformations performed on the graphical structure that have a direct translation in terms of mathematical operations applied to the underlying pdf. Finally one can express the joint global pdf as the product of local conditional pdfs.

PINs present several important advantages: they can handle incomplete data as well as uncertainty; they are trainable and easier 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; they are modular and parallelizable.

The behaviors we are interested in correspond to actions carried out in two particular domains: pedestrian walking in an open outdoors environment and car driving in both city and highway situations. In both situations it seems adequate to adopt generic, compositional analysis in terms of states and transitions between states over time or events that underlie our common sense notions of the corresponding behaviors. Figure ?? shows a typical image of each of these scenarios.

5.1 Hidden Markov 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. 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. Graphically Markov Models are oftern depicted 'rolled-out in time' as PINs:

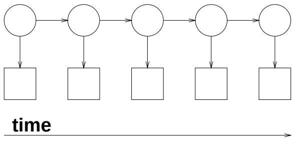

Figure 4: Graphical representation of a 4-state HMM

In order to model these interactions a new architecture is needed, such as Coupled Hidden Markov Models (CHMMs) ([?], [?]). Chains are coupled via matrices of conditional probabilities modeling causal (temporal) influences between their hidden state variables. In [?] we introduce a deterministic approximation for maximum *a posterior* (MAP) state estimation which enables fast classification and parameter estimation via expectation maximization. We obtain 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 $O(TCN^2)$. Naive (Cartesian product), exact (state clustering), and stochastic (Monte Carlo) approaches to CHMMs inference problems involve a combinatoric number of states, typically requiring $O(TN^{2C})$ computations. Much HMM modeling practice is in fact an optimization of the naive strategy, which by definition fails to model interprocess interactions and usually consists of retreats from state counts that overfit the data. We also present a maximum-entropy projection between coupled HMMs and Cartesian product HMMs which also supports estimation and classification. N-heads and projected HMMs compare favorably conventional HMMs and with reduced complexity energy-coupling methods such as mean field approximations.

In the case of modeling human behavior, two persons may interact without wholly determining each other. Each process has its own internal dynamic and is influenced by others, possibly causally. CHMMs are intended to model these kind of interactions.

6 Experimental results

7 Synthetic Agents Behaviors

We have developed a framework for creating synthetic agents that mimic human behavior in a virtual environment. They can be attributed different behaviors and they can interact with each other as well. We generated 5 different kinds of interacting behaviors and various kinds of non interactions. This virtual environment is modelled on the basis of the real scenario from which we obtained the real video data.

There are several motivations for constructing such synthetic agents: Synthetic data let us make a preliminary test of the Markov model architectures chosen for the task of recognizing behavior, Moreover, given the difficulty of collecting examples of rare behaviors from real data, by training and tuning the models of the synthetic agents, combined with good generalization properties and an invariant feature vector, we can obtain synthetic models that are transferable to the real human behavior, with no need of additional training.

This somehow surprising phenomenon is of special importance in a visual surveillance task where most of the behaviors of interest rarely occur and therefore the amount of training data is severely restricted.

Five different interacting behaviors were created:

1. Follow, reach and walk together (inter1),
2. Meet, stop and go on separately (inter2)
3. Meet, stop and go on together (inter3),
4. Change direction in order to meet, stop and continue together (inter4), and
5. Change direction in order to meet, stop and go on separately (inter5).

These interactions could happen at any moment in time and at any location of the virtual environment, provided that the condititions for the different interactions were satisfied. Positions, orientations and velocities for each of the agents were available at each time step. From these data, a feature vector used was constructed, consisting of $\dot{d_{12}}$, the derivatives 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 ?? illustrates some examples for different interacting synthetic agents.

We modeled the interactions with both HMMs and CHMMs with 2 or 3 states per chain for CHMMs, and 3 to 5 states for HMMs accordingly to complexity of the action. Each of these architectures correspond to a different physical hypothesis: CHMMs encode a spatial coupling in time (dynamic balance) whereas HMMs model the data as an arbitrary space-time curve (balance non informative).

We used 90 sequences for training each of the models. The optimal number of states for each interaction as well as the optimal model were obtained by a 10% cross-validation. 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 and sequence for the training data. In the case of HMMs a 6-dimensional feature vector was create; in the case of CHMMs each agent was modeled by a different chain in a 3-dimensional feature vector.

To compare the performance of the two previously described architectures we used the best trained models to classify ***** unseen new sequences. The Viterbi algorithm was used to find the maximum likelihood model for HMMs and the N-heads dynamic programming forward-backward propagation algorithm for CHMMs.

Table 7 illustrates the accuracy for each of the two different architectures and interactions. Both interaction versus no-interaction and inter-interaction classification tests were performed.

Accuracy of Coupled HMMs

No inter	Inter1	Inter2	Inter3	Inter4	Inter5
90.9	100	100	100	100	100
	100	100	100	100	100

Accuracy of Single HMMs

No inter	Inter1	Inter2	Inter3	Inter4	Inter5
68.7	87.5	85.4	91.6	77	97.9
	97.9	100	100	83.3	97.9

Table 1: Accuracy for CHMMs and HMMs. The first row of each table includes testing with no interacting behaviors whereas the second row corresponds to a classification between the different interacting behaviors. Interaction numbers are: Inter0, no interaction; Inter1, follow, reach and walk together; Inter2, meet, stop and go on; Inter3, meet, stop and continue together; Inter4, change direction to meet, stop and go together and Inter5, change direction to meet, stop and go on separately

Note the superiority of CHMMs versus HMMs for classifying the different interactions and, more significantly in the testing case with no interactions.

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.

In Fig ? the accuracies versus the number of sequences used for training in the case of interaction 4.

In systems used for surveillance purposes it's also important having a low rate of false allarms. We therefore calculated the ROC curves for both the coupled and single HMMs. With the coupled HMMs we obtained a minimun false allarm rate in the range of the 20 /

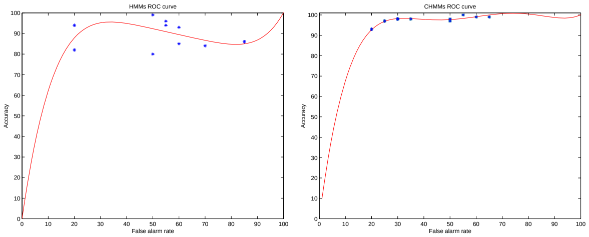

Figure 5:

The amount of training data needed to obtain good results is another important issue, especially for systems designed for working and learning on line, as ours would ultimatelly be, or for domains in which collecting clean data may be difficult.

n Fig ? the plot of the accuracies versus the number of sequences used for training in the case of the interaction 4.

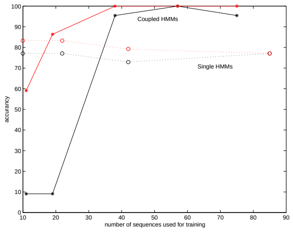

Figure 6: Accuracies of CHMMs (solid line) and HMMs (dotted line) for one particular interaction. The red line is the accuracy without taking into account no interaction while the black consider it.

After these encouraging results obtained for the synthetic agents, we applied the very same architectures to the real data.

8 Pedestrian Behaviors

8.1 Data collection and preprocessing

Using the person detection and tracking system described in section 4 we obtained 2D blob features for each person in several hours of video. Several examples of *following* and different 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 blob that characterizes each person; the Kalman Filter state for each instant of time, consisting of $(\hat{x}, \hat{\dot{x}}, \hat{y}, \hat{\dot{y}})$, 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 close-by 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, $\alpha = sign(< v_1, v_2 >)$, which measures the degree of alignment of the trajectories of each person. Typical feature vectors for a 'follow' and two types of 'meeting' behavior are shown in figure ??.

We used both HMMs and CHMMs for modeling the five following behaviors, illustrated in figure ??: meet and continue together; meet and split; follow; change direction and meet; change direction and meet and continue together. Moreover a *interaction* versus *no interaction*

9 Summary and Conclusions

This template will get you through the minimum article, i.e. no figures or equations. To include those, please refer to your LaTeX manual and the IEEE publications guidelines. Good Luck!

Acknowledgements

This is how to do an unnumbered subsection, which comes out in 11 point bold font.