Hierarchical monitoring of people's behaviors in complex environments using multiple cameras

Nam T. Nguyen Svetha Venkatesh Geoff West Hung H. Bui
School of Computing, Curtin University of Technology
GPO Box U1987 Perth, 6845 Western Australia
{nguyentn, svetha, geoff, buihh}@cs.curtin.edu.au

Abstract

We present a distributed, surveillance system that works in large and complex indoor environments. To track and recognize behaviors of people, we propose the use of the Abstract Hidden Markov Model (AHMM), which can be considered as an extension of the Hidden Markov Model (HMM), where the single Markov chain in the HMM is replaced by a hierarchy of Markov policies. In this policy hierarchy, each behavior can be represented as a policy at the corresponding level of abstraction. The noisy observations are handled in the same way as an HMM and an efficient Rao-Blackwellised particle filter method is used to compute the probabilities of the current policy at different levels of the hierarchy. The novelty of the paper lies in the implementation of a scalable framework in the context of both the scale of behaviors and the size of the environment, making it ideal for distributed surveillance. The results of the system demonstrate the ability to answer queries about people's behaviors at different levels of details using multiple cameras in a large and complex indoor environment.

1 Introduction

The goal of surveillance has changed from building surveillance systems using only a single, powerful camera to building systems deploying many cheap cameras. These systems often require tracking multiple people and recognizing their behaviors. Such problems have found no robust solutions with earlier techniques. The scenario we consider is the use of many static cameras tracking people's behaviors in a large and complex environment.

Understanding people's behaviors in the context of visual surveillance is of current interest. Much work has been done in this field. Rota and Thonnat [10] propose a video understanding framework based on prior knowledge to infer human behaviors. Castel *et al* [4] describe the objects and their possible behaviors using conceptual knowledge. Galata *et al* [5] use variable length Markov models to learn and recognize human behaviors. Ivanov *et al* [6] represent object behaviors as a set of rules incorporating uncertainty, which allows the surveillance system to monitor people activities in a car park. To recognize human behaviors in noisy and uncertain domains, Buxton and Gong [3] construct a high level representation of object activities based on a Bayesian network. This model is applied in a traffic surveillance system to determine behaviors of the objects.

In this paper, we present a distributed surveillance system that works in large and complex indoor environments. To effectively track and recognize behaviors of people, we propose the use of the Abstract Hidden Markov model (AHMM) [1], which is an extension of the Hidden Markov Model (HMM). This model is a computationally effective structure for dealing with a hierarchy of behaviors within a probabilistic framework and thus more scalable in terms of surveillance in large environments. A distributed framework involving a Kalman filter at the lowest level to track people is integrated with the AHMM to recognize behaviors. We demonstrate the results of the system and show how it can answer queries about people's behaviors. The novelty of the method lies in the use of a hierarchical, probabilistic framework to model and track people's behaviors at different levels of details. This allows for a scalable framework to be developed in the context of both the scale of behaviors and the size of the environment.

The paper is organized as follows. First, we describe how the system tracks multiple people using multiple cameras. Then, we briefly describe the AHMM, which we use as the underlying framework for representing and inferring people's behaviors. Finally, the experimental results show how the system works in a real scenario.

2 Tracking people

Our goal is to use multiple static cameras to track people moving in large and complex indoor environments. The system needs to track people well as they move from one camera's field of view (FOV) to another and deal with occlusion. To meet these requirements, we have constructed a distributed tracking system (Nguyen *et al* [8]). The system has a Camera Processing Module (CPM) for each camera and a Central Module (CM). Most processing is done in the CPMs while the CM coordinates the operations of the cam-

1051-4651/02 $17.00 © 2002 IEEE

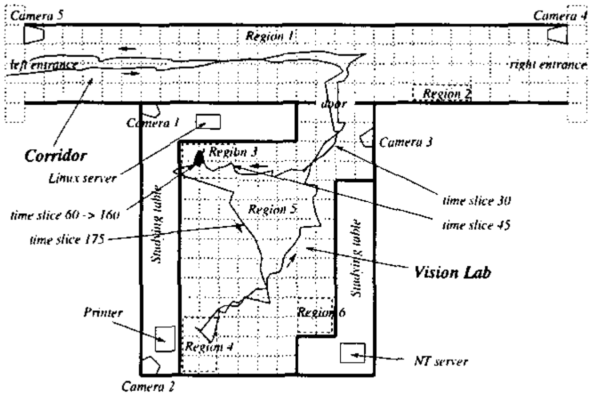

Figure 1. *The environment and the trajectory of person 1*

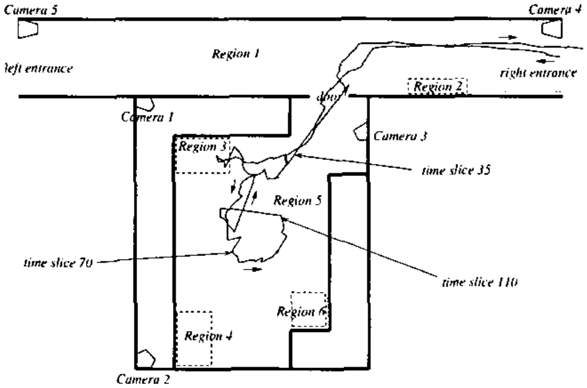

Figure 2. *The trajectory of person 2*

eras and maintains a database of properties and trajectories of all people in the scene.

Each CPM tracks a group of people assigned by the CM. It initializes a Kalman filter [7] for each person to deal with camera noise. The Kalman filter state is updated regularly and sent to the CM for further processing. Observations for the Kalman filter are blobs of motion, which are extracted from images captured by the corresponding camera.

The CM assigns each person to a suitable camera. This assignment is necessary, because each person should be tracked by one camera to reduce the computational cost. Usually, in the case of no occlusion, the nearest camera to a person will capture the largest view. Therefore the CM assigns this person to the nearest camera. If there is occlusion, the person is assigned to the nearest camera that can see that person clearly.

The CM also maintains a database of all people in the scene. It uses the Kalman filter states sent from the CPMs to update the database. The database maintains the positions of people and cameras that are tracking them. This information is passed to an inference module to recognize the people's behaviors. Figures 1 and 2 show examples of using the overlapping cameras to observe and track people.

3 Behavior recognition

3.1 Abstract Hidden Markov Models

To recognize and monitor the people's behaviors over time, we need a framework for representing behaviors at different levels of abstraction. Furthermore, there is a need to deal with the issue of uncertainty and inaccuracy in our observations due to the camera noise and the limitations of the low-level image processing modules. To meet these requirements, we employ the AHMM [1], which can be considered as an extension of the basic HMM [9], where the single Markov chain in the HMM is replaced by a hierarchy of Markov *policies*. In this policy hierarchy, each behavior can be represented by a policy at the corresponding level of abstraction. Starting from a single top-level policy, each policy is refined into a sequence of policies at the lower level and so on until a sequence of states is obtained. The noisy observations of the states are then handled in the same way as in the HMM. Given an AHMM and a sequence of observations, an efficient Rao-Blackwellised particle filter method exists for computing the probabilities of the current policy at different levels in the hierarchy [1].

3.2 Constructing the behavioral hierarchy

The method that we use to construct a policy hierarchy in a spatial environment is similar to the one described in [2]. Firstly, a region hierarchy is constructed. Then, we define a set of policies for each region. Depending on which behaviors of a person in a region we are interested in, we define the corresponding policies to represent these behaviors. For example, for a person in the Corridor of a building, we define policies representing the person's plan of leaving the Corridor via the entrances. For a person in the printer region of the Vision lab, we define policies representing the person's using the printer or simply passing through the region. The policies at the bottom-levels are defined first. Then, we define the policies at the higher levels based on the lower level policies. The parameters of all policies are learned off-line by observing activities of people in the scene. The observation models of the AHMM are also learned off-line separately for each camera by comparing the people's positions returned by the system with the corresponding true positions of people which is specified manually.

4 Experimental results

We have implemented the surveillance system in a scene that consists of a corridor and the Vision lab (see Figure 1). People enter/exit the scene via the left or the right entrance of the Corridor. The system has five static cameras, in

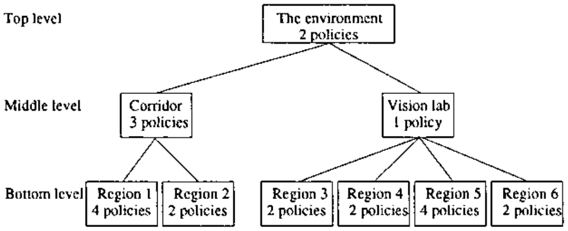

Figure 3. *The region and policy hierarchies*

which three (cameras 1, 2 and 3) are mounted inside the Vision lab and the other two (cameras 4 and 5) are mounted in the Corridor. The FOVs of these cameras overlap each other and cover most of the whole ground plane of the scene. The cameras are calibrated to get the correspondence between points on the floor (ground plane) and points on the image plane. The environment is modeled by a grid of cells or states.. After each time slice, a person can move from its current state to the neighboring states by one of eight directions.

The region hierarchy is shown in Figure 3. It has three levels. The top level is the whole environment. The middle level has two regions: the Corridor and the Vision lab. The bottom level consists of six regions indexed from 1 to 6 (see Figure 1). The Corridor is divided into 2 regions: the news reading region (region 2) and the remaining area of the Corridor (region 1). The Vision lab is divided into four regions: the Linux server region (region 3), printer region (region 4), NT server region (region 6) and the remaining area of the Vision lab (region 5).

The policy hierarchy representing people's behaviors has three levels (see Figure 3) corresponding to the three levels of the region hierarchy. It is defined as follows. Firstly, we define the policies of the bottom level regions (region 1, ..., region 6). In region 1, four policies are defined to represent a person's plan of exiting this region by the left/right entrance, entering the Vision lab or entering region 2. In region 5, four policies are defined to represent a person's plan of visiting region 3, 4 and 6 or walking around in this region. In each of the remaining bottom level regions (regions 2, 3, 4 and 6), we define two policies: one represents a person doing a "special action" in the region and the other represents a person simply passing through the region. The special actions in regions 2, 3, 4 and 6 are: reading news, using the Linux server, using the printer and using the NT server respectively. Next, we define policies for the middle level regions (the Corridor and the Vision lab). Three policies are defined for the Corridor representing a person's plan of exiting the Corridor by the left/right entrance or entering the Vision lab. We define only one policy for the Vision lab, because a person in the lab eventually must take the only lab door to get out. Finally, we define two policies for the top level region (the whole environment) representing a person's leaving the scene via the left/right entrance.

Consider two people entering the scene. Person 1 enters the scene from the left entrance (see Figure 4(a)). Person 2 enters the scene from the right entrance (see Figure 4(b)). Then, both enter the Vision lab (see Figure 4(c)). Figures 1 and 2 show the trajectories of these two people returned by the multiple camera tracking system.

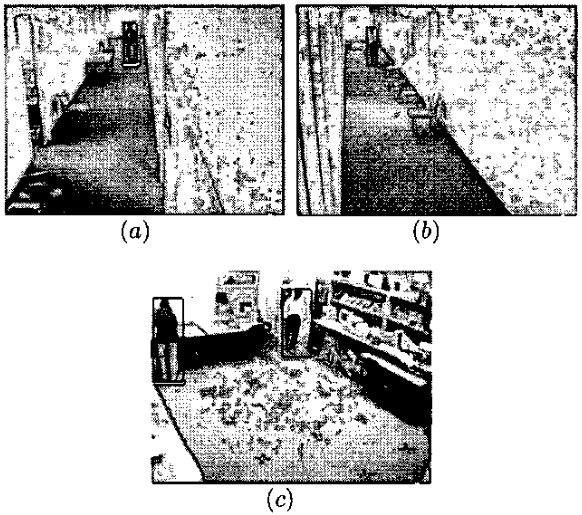

Figure 4. *(a) Person 1 is entering the scene, (b) Person 2 is entering the scene and (c) Both are in Vision lab.*

With the AHMM model defined above and the sequence of observations of the two people, the AHMM inference algorithm [1] is run to infer the policies used by this person at various levels.

We can recognize the behaviors of the two people at various levels of details (Figures 5-7). In each figure, each time slice lasts approximately one third of a second. Figure 5 shows the probabilities that person 1 exits the environment by the left or right entrance (denoted by p_{left_e} and p_{right_e} respectively). At the beginning, p_{left_e} decreases when person 1 goes further away from the left entrance (see the trajectory in Figure 1). Then, p_{left_e} is constant from time slice 30 when person 1 is inside in the Vision lab. This is because only one middle level policy is defined for the Vision lab. When person 1 is in the lab, he must take this policy and his movement inside the lab is independent of his final exit/entrance. p_{left_e} increases and approaches 1 when person 1 leaves the lab and goes towards the left entrance. In contrast, p_{right_e} falls to zero during this time.

We now look at the results of querying the bottom level policies. Figure 6 shows the distribution of the possible destinations of person 2 from time slice 35 to time slice 110, when he is in region 5 (see the trajectory in Figure 2). We denote the probabilities that person 2 is visiting the Linux server region, the printer region, the NT server region or just walking around by p_{v_Linux}, $p_{v\_printer}$, p_{v_NT} and p_{w_around} respectively. As in Figure 6, at time slice 35, p_{w_around} is quite high while other probabilities are low

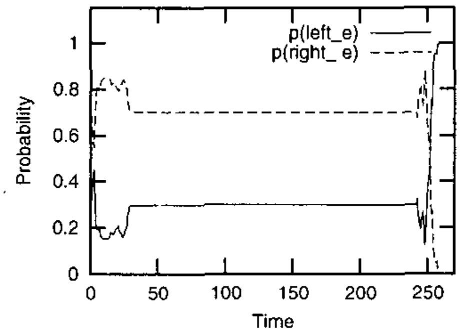

Figure 5. *The probabilities that person 1 leaves the scene via the entrances*

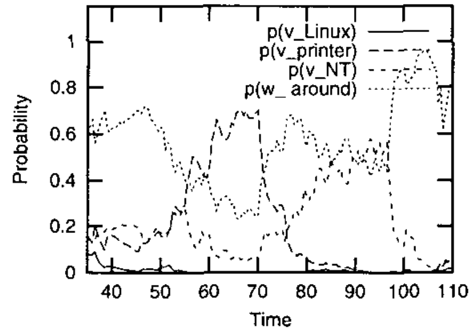

Figure 6. *The probabilities that person 2 will visit region 3, 4, 6 or just walk around in the lab.*

because person 2 does not clearly head to the Linux server, printer or NT server region. After this time, $p_{v\_printer}$ increases because person 2 heads to the printer region (region 4). At time slice 70, he changes the direction and heads to the NT server region (region 6), so p_{v_NT} increases while $p_{v\_printer}$ decreases gradually.

We are interested in the two people's actions in the scene. For example, we want to know whether person 1 is using the Linux server or simply passing through the Linux server region (see the trajectory in Figure 1). We denote the probabilities that person 1 is doing these actions by p_{u_Linux} and p_{pass} respectively. Figure 7 shows p_{u_Linux} and p_{pass} from time slice 30 to time slice 175. At time slice 45, p_{pass} is quite high compared with p_{u_Linux} showing that the system is not sure about the behavior of person 1 at this time. From time slice 60 to time slice 160, p_{u_Linux} is nearly 1 showing that the system is almost certain that person 1 is using the Linux server. This is because person 1 is in the Linux server region and the position is nearly unchanged during this time (see Figure 1).

5 Conclusion

Using the framework of AHMM, we have constructed a surveillance system using multiple cameras to recognize people's behaviors in large and complex indoor scenes. We have demonstrated how both the environment and people's

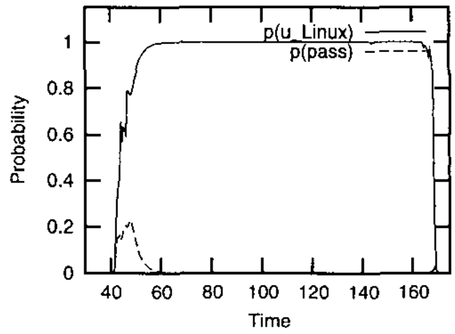

Figure 7. *The probabilities that person 2 is using the Linux server or passing through the Linux server region*

behaviors can be modeled and queried at different levels of details. The distributed nature of the multiple camera tracking system together with the efficiency of the AHMM inference algorithm ensures that the system is scalable to larger/more complex spatial domains.

References

[1] H. H. Bui, S. Venkatesh, and G. West. Policy recognition in the Abstract Hidden Markov Model. *Journal of Artficial Intelligence Research.* to appear.

[2] H. H. Bui, S. Venkatesh, and G. West. Tracking and surveillance in wide-area spatial environments using the Abstract Hidden Markov Model. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):177–195, 2000.

[3] H. Buxton and S. Gong. Advanced visual surveillance using Bayesian networks. In *Proceedings of the IEEE International Conference on Computer Vision*, Cambridge, Massachusetts, June 1995.

[4] C. Castel, L. Chaudron, and C. Tessier. What is going on? A high level interpretation of sequences of images. In *4th European Conference on Computer Vision*, Cambridge UK, April 1996.

[5] A. Galata, N. Johnson, and D. Hogg. Learning variable length markov models of behaviour. *Intenational Journal of Computer Vision and Image Understanding*, 81(3):398–413, March 2001.

[6] Y. Ivanov and A. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 22(8):852–872, August 2000.

[7] P. S. Maybeck. *Stochastic Models, Estimation and Control.* Academic Press, New York, 1979.

[8] N. T. Nguyen, S. Venkatesh, G. West, and H. H. Bui. Coordination of multiple cameras to track multiple people. In *Asian Conference on Computer Vision*, pages 302-307, 2002.

[9] L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[10] N. Rota and M. Thonnat. Video sequence interpretation for video surveillance. In *Proceedings of the Third IEEE International Workshop on Visual Surveillance*, 2000.