Automated Visual Surveillance Using Hidden Markov Models

Vinod Nair James J. Clark

Centre for Intelligent Machines
McGill University
Montreal, PQ H3A 2A7
{vnair, clark}@cim.mcgill.ca

Abstract

This paper describes an automated visual surveillance system that detects suspicious human activity in a scene. The system is designed to: 1) detect and track people in the scene, 2) recognize the "normal" activities in the scene, and 3) detect anomalous activity by finding sufficiently large deviations from the normal activity patterns. The stochastic time-sequence recognition framework of the Hidden Markov Model (HMM) forms the basis of activity recognition and anomaly detection. We have implemented the system to monitor an office corridor in real-time using a Pentium III machine running Windows 2000. The results show that the system correctly classifies examples of normal activities in the corridor and identifies a mock break-in attempt as suspicious activity.

1. Introduction

Automated visual surveillance is becoming an increasingly important area of research in computer vision. CMU's Video Surveillance and Monitoring (VSAM) project [2] and MIT AI Lab's Forest of Sensors project [7] are examples of recent research efforts in the field. Interest has been motivated by commercial applications such as surveillance of airports and office buildings, as well as military ones, such as monitoring the battlefield to automatically collect strategic information. Conventional visual surveillance systems have limitations that make them less than ideal for many applications. For instance, recording the surveillance video on tapes can provide evidence only after a security breach has occurred. The alternative of dedicating a security worker to watch the live video is expensive and prone to human error. Automated visual surveillance overcomes these limitations by detecting suspicious activity as it happens, without human effort.

Our approach to automated visual surveillance is to classify the normal activities using a set of discrete Hidden Markov Models (HMMs), each trained to recognize one activity, and label the unrecognized activities as unusual. In recent years, HMMs have become popular in computer vision as an activity recognition algorithm. They have been used to recognize hand gestures in sign language [6], facial expressions [4], and different tennis strokes [8]. They have also been used in visual surveillance systems for classifying activities in an office room [1], and in a parking lot [1, 3]. A common feature of these applications is the use of HMMs to generate high-level inferences with only relatively coarse, low-level sensory data, such as blob features. This illustrates an important advantage of HMMs – combining coarse, low-level sensory data with the prior knowledge of the data's statistical characteristics learned by HMMs, avoids the need to compute high-level representations of the data using expensive image processing algorithms.

The paper is organized as follows: section 2 describes the architecture of the surveillance system. Section 3 gives the results of the system's performance, and section 4 presents the conclusions of the paper.

2. System's Architecture

2.1. Hardware Component

The surveillance system uses an ordinary netcam to obtain video of the office corridor under surveillance. The netcam has a built-in HTTP server from which the video can be downloaded as hardware-compressed JPEG frames via the internet, as shown in figure 1. The use of netcams makes the system simple and inexpensive, and allows great

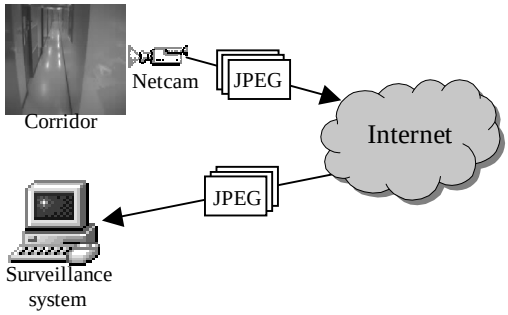


Figure 1. The netcam transmits JPEG frames to the surveillance system via the internet.

flexibility for adding or removing cameras from the system. More cameras can be added to the system by simply connecting them to the internet. The physical distance between the camera and the computer running the system can be very large, since they can be connected at any two arbitrary points on the network. In addition, it is possible to distribute the computer vision processing performed by the system over multiple, cooperating computers communicating through the internet. The principal drawback of a netcam is its low frame rate, which is limited to a maximum of about 2 frames per second due to the delay in compressing and uncompressing JPEG frames, and the latency of the internet.

2.2. Detecting and Tracking People

The first step in recognizing human activities is to detect the people in the scene. Since the background of the office corridor is relatively constant in time, background subtraction is used for detecting people. The background model $B_n(x,y)$ is manually initialized to the average of five consecutive video frames taken without anyone present in the corridor. The threshold function $T_n(x,y)$ is initialized to 50 at all pixel locations. Then, a pixel in the frame $I_n(x,y)$ is labelled as foreground if it satisfies the condition

$$\left|I_n(x,y) - B_n(x,y)\right| > T_n(x,y)\,. \tag{1}$$

The background model and the threshold function are updated every frame using temporal averaging:

$$B_{n+1}(x,y) = \begin{cases} B_n(x,y) \text{ if } (x,y) \text{ is foreground}, \\ \alpha B_n(x,y) + (1-\alpha) I_n(x,y) \text{ otherwise} \end{cases} \tag{2}$$

$$T_{n+1}(x,y) = \begin{cases} T_n(x,y) \text{ if } (x,y) \text{ is foreground}, \\ \alpha T_n(x,y) + 2(1-\alpha)\,|\, I_n(x,y) - B_n(x,y)\,| \text{ otherwise} \end{cases} \tag{3}$$

where α is a constant that determines how fast $B_n(x,y)$ and $T_n(x,y)$ adapt to changes in the scene. The output of the background subtraction is represented as a binary mask that labels background pixels as black and foreground pixels as white. Morphological dilation is applied on the mask to connect together fragmented foreground regions. The foreground pixels are then clustered together using a connected components algorithm to generate a list of blobs detected in the current frame. Blobs that are either too small or do not have approximately the same aspect ratio as the human body are ignored.

The coordinates of the blob's center of mass, average colour, and height are the features used for blob correspondence across frames. At every frame, the match score is computed between all possible pairs of existing blob tracks and newly detected blobs. Blobs are assigned to the tracks in increasing order of match score (a lower score means a better match), with a maximum score constraint enforced to avoid making arbitrarily bad matches. Newly detected blobs that are not matched with any existing track are assigned new tracks. Existing tracks that remain unmatched with any blob for a fixed number of seconds are deleted.

2.3. Activity Recognition Using HMMs

The detection and tracking modules provide time sequence data on the blob features as the blobs are tracked in the scene. The recognition module then uses this data to classify each blob's activity with discrete HMMs. The blob feature vector consists of the center of mass coordinates of the blob, its direction of motion (i.e. moving towards the camera, away from the camera or stationary), and height. Since the HMMs require a discrete, symbolic representation of the feature vector data, each feature vector is converted into a discrete symbol using vector quantization. Codebook vectors of the quantizer are computed from a set of training vectors using a *k*-means algorithm. Once trained, the quantizer maps a vector to the index of the nearest (in a Euclidean distance sense) codebook vector.

HMMs model discrete-time sequences as the output of a process involving stochastic transitions among hidden states at discrete time steps (figure 2). The probability of transiting to any state depends only on the preceding state. In the case of a discrete HMM, an observable discrete symbol is stochastically output by the current hidden state at each time step, forming a time series of symbols. Suppose a discrete HMM has N states $Q = \{q_1, q_2, \ldots, q_N\}$ and M output symbols $V = \{v_1, v_2, \ldots, v_M\}$. Denote the

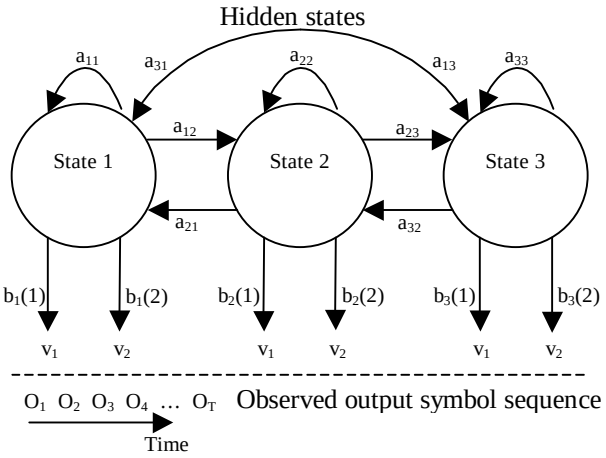


Figure 2. An example of a fully-connected 3-state, 2-output discrete Hidden Markov Model.

state at time step t as s_t. Then the parameters of the HMM are fully specified by the triplet $\lambda = \{A, B, \pi\}$, where

$$A \equiv \{a_{ij} \mid a_{ij} = P(s_{t+1} = q_j \mid s_t = q_i)\} \quad (4)$$

is the N x N state transition probability matrix,

$$B \equiv \{b_i(k) \mid b_i(k) = P(v_k \mid s_t = q_i)\} \quad (5)$$

is the M x N state output probability matrix, and

$$\pi \equiv \{\pi_i \mid \pi_i = P(s_1 = q_i)\} \quad (6)$$

is the initial state probability vector.

To recognize an observed symbol sequence $O = O_1 O_2 \ldots O_T$ of length T time steps, the probability of the sequence for a given HMM λ is computed using Bayes' rule as $P(O|\lambda)$. This probability is evaluated using the forward algorithm [5], given as

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_t(i)\big|_{t=T}, \quad (7)$$

where the forward variable $\alpha_t(i)$ is defined as

$$\alpha_t(i) \equiv P(O_1, O_2, \ldots, O_t, s_t = q_i \mid \lambda). \quad (8)$$

To train a HMM to recognize the observation sequence O, the parameter set λ that maximizes $P(O|\lambda)$ must be estimated from the training data. The Baum-Welch algorithm [5] is used to iteratively obtain an estimate of λ that is guaranteed to locally maximize $P(O|\lambda)$. Define the backward variable $\beta_t(i)$ as

$$\beta_t(i) \equiv P(O_{t+1}, \ldots, O_T, s_t = q_i \mid \lambda). \quad (9)$$

Also define

$$\gamma_t(i) \equiv P(s_t = q_i \mid O_1, \ldots, O_T, \lambda) = \frac{\alpha_t(i)\beta_t(i)}{P(O \mid \lambda)}, \quad (10)$$

$$\xi_t(i, j) \equiv P(s_t = q_i, s_{t+1} = q_j \mid O_1, \ldots, O_T, \lambda) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O \mid \lambda)}. \quad (11)$$

Then, given an estimate of λ, a better estimate λ' can be obtained as follows:

$$a'_{ij} \equiv \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad b'_i(k) \equiv \frac{\sum_{t \in \{t \mid O_t = v_k\}} \gamma_t(i)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad \pi'_i = \gamma_1(i). \quad (12)$$

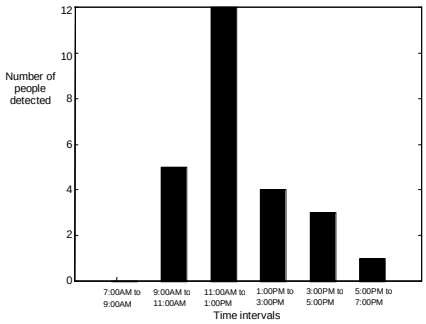
Learning converges to a local maximum of $P(O|\lambda)$ when $\lambda' = \lambda$.

The activities in the office corridor are recognized using a set of HMMs, each trained to recognize one activity. The probability of the observation sequence is computed for each HMM, and the activity is recognized as the one represented by the HMM with the highest log likelihood. However, if the log likelihood is below a minimum threshold for all the HMMs, then the activity is classified as suspicious.

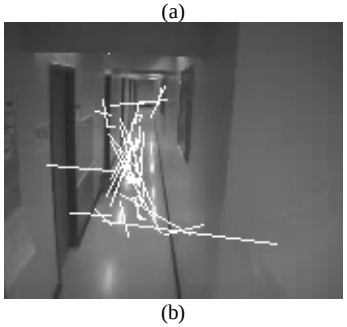
3. Results

3.1. Detection and Tracking Results

Figure 5 shows an example of the surveillance system detecting and tracking a person in the office corridor. Background subtraction properly segments the people in the scene most of the time. However, strong shadows are often misclassified as foreground, creating erroneous blobs and distorting the appearance of legitimate blobs. The histogram in figure 3(a) shows the number of people tracked in the corridor at various times during one day. The paths taken by the people are shown in figure 3(b).



(a)



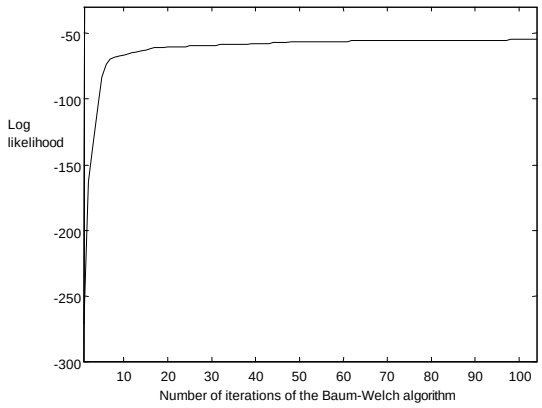
(b)

Figure 3. (a) Histogram of the number of people in the corridor during various times of the day, (b) the paths taken by the people.

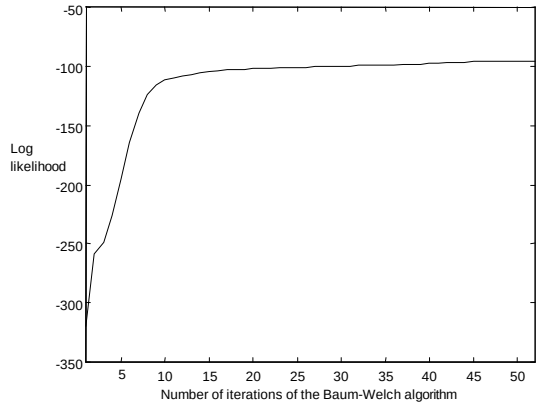
These results suggest that the activities in the scene have spatial and temporal patterns that can be characterized statistically, and qualitatively justify the use of a statistical pattern recognition technique such as the HMM to learn those patterns.

3.2 Activity Recognition Results

The most common activities in the corridor are entering and exiting a room. Therefore, we trained two HMMs to recognize people entering and exiting one of the rooms in the corridor (the procedure could be repeated for the other rooms also, although it was not done here). The training was done offline using the Baum-Welch algorithm with 15 training sequences for each activity. 10 states were used for each HMM, and a quantizer with 7 codebook vectors was computed from the training data. The learning curves for the two HMMs (figure 4) show how the log likelihood of each activity improves for the corresponding HMM as training progresses. In both cases, the Baum-Welch algorithm reaches near convergence within about 10 iterations, and then slowly approaches the final convergence point.



(a)



(b)

Figure 4. Learning curves show how the log likelihood of a sequence for (a) entering the room and (b) exiting the room improves as the corresponding HMM's training progresses.

After training, the surveillance system was used to recognize real-life examples of a person entering and exiting a room, as well as a mock break-in attempt. Sequences showing a person entering and exiting the room, and the break-in attempt are given in figures 6-8. Table 1 shows the likelihood of each sequence for the two HMMs.

Table 1. Log likelihood results for entry, exit and break-in sequences

Sequence	Log likelihood for entry HMM	Log likelihood for exit HMM
Entering room	-63.05	$-\infty$
Exiting room	$-\infty$	-763.96
Break-in	$-\infty$	$-\infty$

In all three cases, the system correctly classified the observed activity. For the sequences showing a person entering and exiting the room, the HMM corresponding to the activity produced a finite likelihood value while the other one gave zero likelihood. For the break-in sequence, both HMMs produced zero likelihood, thus indicating that the activity is unrecognizable and possibly suspicious. As a result, a security alert was displayed on the screen of the computer running the surveillance system (see figure 8).

4. Conclusions

This paper has described an automated visual surveillance system that classifies human activities and detects suspicious events in a scene. We have implemented the system to monitor the activities in an office corridor in real-time. The system detects people in the corridor using background subtraction, and tracks them to obtain time-sequence data on each person's motion, which is then used for activity recognition. Normal activities are classified using a set of discrete HMMs, each trained to recognize one of the activities. Outlier activities with low likelihoods for all the HMMs are classified as anomalous.

The results demonstrate that the HMM-based approach to activity recognition can be effective. Our system correctly classified normal activities such as a person entering a room and exiting a room, and identified a mock break-in attempt as suspicious. However, the system's false alarm rate is high since it can recognize only two of the many possible normal activities. This problem can be solved partially by expanding the set of recognized activities. As our future direction of work, we are considering the use of unsupervised learning techniques to automatically discover the normal activities.

Acknowledgements

Vinod Nair would like to thank Mathieu Lamarre for his great help throughout this work. The research was supported by a grant from the IRIS Network of Centres of Excellence, and by an NSERC USRA.

References

[1] M. Brand and V. Kettnaker, "Discovery and Segmentation of Activities in Video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 844-851, August 2000.

[2] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L. Wixson, "A System for Video Surveillance and Monitoring," Technical Report CMU-RI-TR-00-12, Carnegie Mellon University, 2000.

[3] Y. A. Ivanov and A. F. Bobick, "Recognition of Visual Activities and Interactions by Stochastic Parsing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 852-872, August 2000.

[4] N. Oliver, A. Pentland, and F. Berard, "LAFTER: A Real-time Face and Lips Tracker with Facial Expression Recognition," *Pattern Recognition*, vol. 33, pp. 1369-1382, 2000.

[5] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-286, 1989.

[6] T. E. Starner and A. Pentland, "Visual Recognition of American Sign Language Using Hidden Markov Models," *Proceedings of the International Workshop on Automatic Face- and Gesture-Recognition*, pp. 189-194, 1995.

[7] C. Stauffer and W. E. L. Grimson, "Learning Patterns of Activity Using Real-Time Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747-757, August 2000.

[8] J. Yamato, J. Ohya, and K. Ishii, "Recognizing Human Action in Time Sequential Images Using Hidden Markov Model," *1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 379-385, 1992.

Figure 5. A sequence showing the detection and tracking of a person in the corridor.

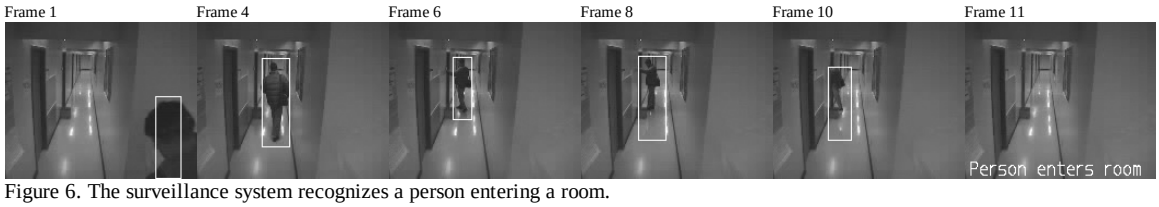


Figure 6. The surveillance system recognizes a person entering a room.

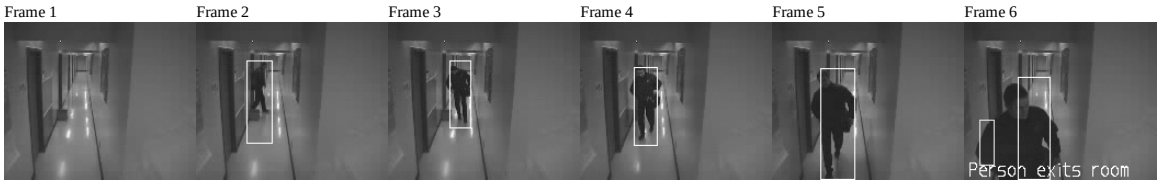


Figure 7. The surveillance system recognizes a person exiting a room.

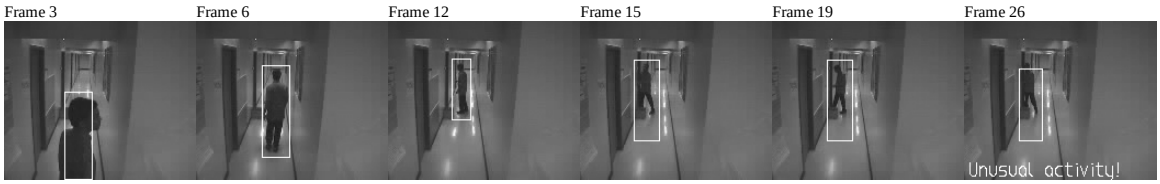


Figure 8. A mock break-in attempt results in a security alert message.