A Dynamic Hierarchical Clustering Method for Trajectory-Based Suspected Video Event Detection

***Abstract* — The proposed suspected video event detection method is based on unsupervised clustering of object trajectories, which are modeled by hidden Markov models (HMM). The clustering-based approach for detecting abnormalities in surveillance video requires the appropriate definition of similarity between events. Motion segmentation is based on an adaptive back-ground subtraction method that models each pixel as a mixture of Gaussians. The Gaussian distributions are then evaluated to determine which are most likely to result from a background process. This yields a stable, real-time outdoor tracker that reliably deals with lighting changes, repetitive motions from clutter, and long-term scene changes. The HMM-based similarity method falls short in handling the overfitting problem. In this paper a multi-sample-based similarity measure is proposed, where HMM training and distance measuring are based on multiple samples. These multiple training data are acquired by a novel dynamic hierarchical clustering (DHC) method. By iteratively reclassifying and retraining the data groups at different clustering levels, the initial training and clustering errors due to overfitting will be sequentially corrected in later steps.**

***Keywords*- real-time visual tracking, adaptive background estimation, hidden Markov model, event detection, unsupervised clustering, Overfitting.**

I. INTRODUCTION

Video surveillance systems are widely used in many important sites such as supermarket, bank, hotel, etc. However, the captured video data are commonly stored or previewed by operators for finding abnormal moving objects or events. The value in use is very low.

Many surveillance applications require analysis of the events taking place in video streams recorded in specific situations, in order to find suspicious or abnormal actions, which might present a threat and should be signaled to a human operator. Typically, the video camera is fixed and the site being monitored is mainly static. Object trajectories are extracted from the video and the video events can be represented by time sequence of the various features of the objects. In many cases, no *a priori* knowledge is given for patterns of unusual video events. Thus, we aim to analyze all the trajectories extracted from existing videos, and differentiate unusual trajectories from normal ones automatically.

To address this problem, the approach is based on the fact that a normal event is associated with the commonality of the behavior and an unusual event indicates its distinctness. For instance, people running represent an unusual event if most people in the crowd are walking, and a car moving against traffic also represents an unusual event. Clearly, what characterizes normality is the high recurrence of some similar events. Typically, there are only a few such normal patterns in a specific surveillance scenario. Therefore, unsupervised clustering can be performed on all video events, so that those events clustered into dominant (e.g., large) groups can be identified as normal, while those that cannot be explained by the dominant groups (e.g., distant from all cluster centers) are defined as unusual.

In real videos, the suspicious events are rare, difficult to describe, hard to predict and can be subtle. Some researchers [1-6] define events as either clusters of parameter space components (normal events) or outliers (abnormal events). In order to perform this clustering-based approach, a similarity measure between two events, probably with different time lengths, needs to be specified. Some recent results [1-4] define the distance of two HMM-represented sequences based on the likelihood of observing one sequence given the HMM trained from another sequence. To be exact, the larger their likelihood of being generated from each other's model will be, the more similar these two sequences are. However, this cross likelihood measurement has the problem of model over-fitting due to data shortage, as the HMM is trained on only one sample. Therefore data clustering using this single sample-based similarity is quite unreliable, especially for the popular spectral clustering algorithm [2,4-6], which is extremely sensitive to the construction of the similarity matrix (whose Eigen values are utilized).

In this paper first we discuss about building a robust motion tracker, which is required to track the objects and their positions to get their trajectories. Second a multi-sample-based similarity measure to suppress the overfitting problem is proposed, where HMM representation is based on several similar samples. The acquisition of these multiple training data is by hierarchically clustering and iteratively retraining the whole dataset, which is summarized as dynamic hierarchical clustering (DHC) algorithm. This algorithm can dynamically correct initial overfitting errors as the numbers of training samples increase (i.e. data clusters become larger).

II. BUILDING A ROBUST MOTION TRACKER

A robust video surveillance and monitoring system should not depend on careful placement of cameras. It should also be robust to whatever is in its visual field or whatever lighting effects occur. It should be capable of dealing with movement through cluttered areas, objects overlapping in the visual field, shadows, lighting changes, and effects of moving elements of the scene (e.g. swaying trees), slow-moving objects, and objects being introduced or removed from the scene. Thus, to monitor activities in real outdoor settings, we need robust motion detection and tracking that can account for such a wide range of effects. Traditional approaches based on backgrounding methods typically fail in these general situations. The goal is to create a robust, adaptive tracking system that is flexible enough to handle variations in lighting, moving scene clutter, multiple moving objects and other arbitrary changes to the observed scene. The resulting tracker is primarily geared towards scene-level video surveillance applications.

A. Adaptive approach to motion tracking

Rather than explicitly modeling the values of all the pixels as one particular type of distribution, simply model the values of a particular pixel as a mixture of Gaussians. Based on the persistence and the variance of each of the Gaussians of the mixture, and then determine which Gaussians may correspond to back ground colors. Pixel values that do not fit the back ground distributions are considered foreground until there is a Gaussian that includes them with sufficient, consistent evidence supporting it to convert it to a new background mixture. The system adapts to deal robustly with lighting changes, repetitive motions of scene elements, tracking through cluttered regions, slow-moving objects, and introducing or removing objects from the scene. Slowly moving objects take longer to be incorporated into the background, because their color has a larger variance than the background. Also, repetitive variations are learned, and a model for the background distribution is generally maintained even if it is temporarily replaced by another distribution which leads to faster recovery when objects are removed. The adaptive back grounding method contains two significant parameters α, the learning constant and T, the proportion of the data that should be accounted for by the background.

If each pixel resulted from a single surface under fixed lighting, a single Gaussian would be sufficient to model the pixel value while accounting for acquisition noise. If only lighting changed over time, a single, adaptive Gaussian per pixel would be sufficient. In practice, multiple surfaces often appear in the view frustum of a particular pixel and the lighting conditions change. Thus, multiple, adaptive Gaussians are required. Use an adaptive mixture of Gaussians to approximate this process. [12]

The figure 1 shows the results of the approach. Fig 1 (a) is the back ground image frame, Fig (b) is foreground image, Fig 1 (c) is red hued foreground image where object is tracked, and Fig 1 (d) is the image frame where object's position is marked with Yellow 'plus' and all the noise is removed from foreground frame. So the proposed method of tracking object is robust and overcomes lighting effects.

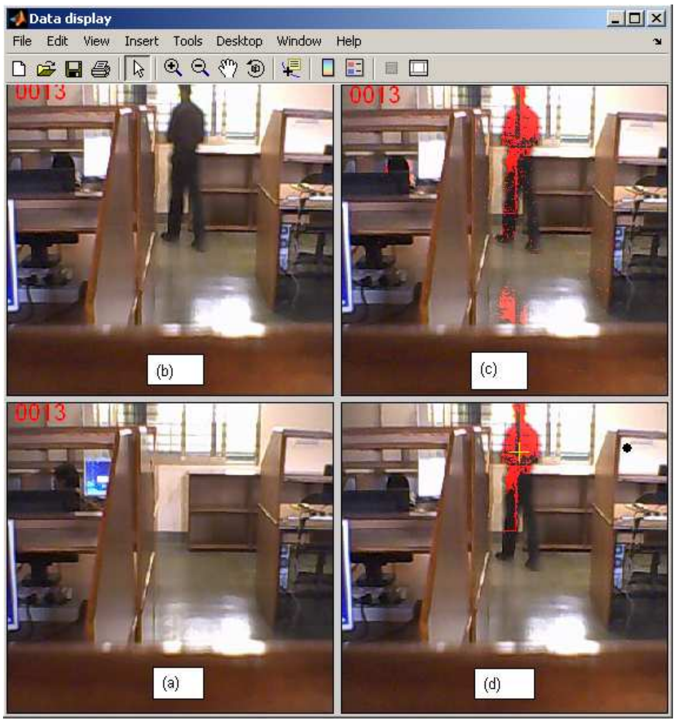


Fig. 1 Adaptive Tracker Result.

III. CLUSTERING-BASED APPROACH FOR ABNORMAL EVENT DETECTION

A. HMM representation of video events

In many existing work on surveillance video analysis [2, 4, 7, 8], video events are represented as object trajectories or time evolutions of certain frame features, which can be further modeled by HMM. For example, Fig. 1(d) shows a human in motion is tracked and whose position can be extracted from a surveillance video monitoring a room door in research lab. A HMM with Gaussian emission probability is fitted to the 2-D trajectory feature vector $\{(x_1, y_1), (x_2, y_2), \ldots (x_T, y_T)\}$, where $\{x, y\}$ denotes the coordinate of object center at every frame and T is the length of the trajectory.

B. Modeling of normal events

The clustering-based approach detects abnormal events by first modeling normal events. After training data that are acquired from the history videos are represented /parameterized by HMMs as described in Sec. A, unsupervised clustering is performed on them based on a certain similarity measure (will be described later in Sec. III). The clustering process ends up with a few data groups. Those groups containing large number of samples (e.g., more than the average number) are chosen as normal pattern groups, and then HMMs are learned for every normal group. These HMMs, denoted by $\{m_k\}$ (k = 1, 2, …), are models of normal events.

C. *Detection of abnormal events*

Based on the models of normal groups, detection of abnormal events can be performed to new video data. Specifically, given an unseen object trajectory *i*, the likelihood of observing *i* given any HMM of normal events m_k is denoted by $L\ (i|m_k)$. If the maximum likelihood is less than a threshold, i.e.,

$$\max\{L(i\mid m_k)\} < Th_A \qquad (1)$$

where T_{hA} is a threshold, this query trajectory *i* is detected as abnormal.

IV. CLUSTERING ALGORITHM

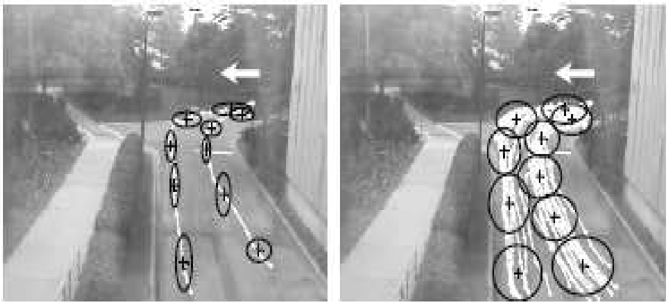
A. *Multi-sample-based similarity measure*

In some recent work [2, 4], the distance d_{ij} between two events/trajectories *i* and *j*, modeled by two HMMs m_i and m_j respectively, is defined as:

$$d_{ij} = L(i\mid m_i) + L(j\mid m_j) - L(j\mid m_i) \qquad (2)$$

where $L(i|m_j)$ denotes the log-likelihood of trajectory *i* utilizing the model m_j for trajectory *j*, normalized by trajectory length *T*, that is,

$$L(i\mid m_j) = \frac{1}{T_i}\log P(i\mid m_j) \qquad (3)$$



(a) (b)

Fig. 2. HMM modeling of object trajectories.

If the trajectories *i* and *j* are different, their likelihood of being generated by each other's model, $L(i\mid m_j)$ and $L(j\mid m_i)$, will be smaller than the likelihood of being generated by itself's model, $L(i\mid m_i)$ and $L(j\mid m_j)$, thus the distance d_{ij} will be large. If the two trajectories are similar, the difference between the cross likelihood and likelihood of self modeling will be small, thus the distance is small. The extreme case is that distance of two identical trajectories will be equal to zero. However, this HMM-based distance measure has the problem of overfitting with trajectory data extracted from real videos. Note that the variances of the fitted Gaussian distributions indicated by black ellipses in Fig. 2(a) are very small. This is because HMM is trained on only one sample thus it fits the data too closely. This overfitted model will generate very different parameters for similar trajectories in the same direction (e.g., the two trajectories in Fig. 2(a)). As a result, the distance defined in Eq. 2 becomes too large to group similar trajectories into one cluster. One solution to this problem is to use more similar data to train a model that allows for larger variation as illustrated in Fig. 2(b). In terms of this multi-sample-based modeling, the distance between two *groups* of trajectories (groups *i* and *j*) can be defined similarly to Eq. 2, except for a modification of the likelihood term. That is, we propose the following definition

$$L(i\mid m_j) = \frac{1}{N_i}\sum_r \frac{1}{T_r}\log P(i_r\mid m_j) \qquad (4)$$

where *ir* denotes the *r*-th trajectory in group *i* (with its length equal to T_r) and N_i is the number of trajectories in group *i.* In other words, $L(i\mid m_j)$ is refined as the average of the likelihood of all trajectories in group *i*, generated by the HMM trained on group *j*. The multi-sample-based distance measure is more reliable than the one based on a single sample. For example, the distance between the two trajectories in Fig. 2(a) calculated by Eqs. 2 and 3 is equal to 263.72, while the distance between the two groups containing 20 trajectories each in Fig. 2(b) calculated by Eqs. 2 and 4 is equal to 22.16[10]. As the trajectories shown in Figs. 2(a) and (b) are all on the same road and in the same direction, thus they need to be clustered into one group. This can be accomplished much easier with a smaller distance calculated using Eq. 4.

V. DYNAMIC HIERARCHICAL CLUSTERING (DHC)

HMM modeling based on multiple samples provides a better representation of the trajectory data. However, this is a "chicken-and-egg" problem. On one hand, models are acquired by training on samples in one group; while on the other hand, groups are acquired by model-based clustering.

1) Initialization: each trajectory in the dataset forms a group and is fitted with a HMM. There are *N* groups and *N* HMMs
2) Distance measurements: calculate distances {*dij*} between two groups *i* and *j* in the dataset by Eqs. 2 and 4.
3) Merging: the two groups *i* and *j* with smallest *dij* are merged into one if the following criterion is satisfied

$$\frac{L(i\mid m_i).L(i\mid m_j)}{L(i\cup j\mid m_{i\cup j})} < 1$$

where $L(i\mid m_i)$ and $L(j\mid m_j)$ are likelihood of group *i* and *j* generated by HMMs trained on the two groups respectively, $L(i\ U\ j\mid m_{iUj})$ the likelihood of samples of both groups generated by HMM trained on all these samples, as defined in Eq. 4; otherwise no groups can be merged and the clustering process ends;
4). Reclassifying: m_i and m_j are replaced by $\mathrm{m}_{\mathrm{iUj}}$; then based on the *N-1* HMMs, all the data are classified into *N*-1 groups by the maximum likelihood (ML) criterion;
5) Retraining the N-1 HMMs are retrained based on the updated N-1 data groups;
6) N= N-1; go back to step 1).

Fig. 3 Proposed dynamic hierarchical clustering algorithm.

The common approach to solve such an interlocked problem is to use an iterative approach. For instance, the EM algorithm is an iterative way to solve the embedded problem of data segmentation and model parameters estimation. To allow for an iterative solution, trajectory clustering cannot be accomplished in one-step but in a hierarchical fashion instead. In fact, our proposed dynamic hierarchical clustering (DHC) algorithm is based on classic hierarchical clustering [9], incorporated with an iteration process of data reclassifying and model retraining, as described in Fig. 3. At the beginning of this clustering algorithm (step 0), data samples are possibly over fitted as each HMM is trained on just one trajectory. However, when samples are clustered into larger groups, the number of training data increases as retraining is performed on groups of samples instead of on a single sample at step 4. Therefore, the over fitted HMMs at the beginning can be sequentially refined/updated. Meanwhile, the first few samples that are probably clustered incorrectly due to overfitting will be gradually corrected at step 3 of reclassifying during the iteration process. In other words, the proposed DHC algorithm has the ability of self adjustment in both model training and data clustering. Another advantage of this algorithm is that it is not sensitive to the absolute similarity/distance values, as at step 2 only the comparison of distance is required to find two group candidates for merging, compared to the complex. Eigen value decomposition used in spectral clustering [2, 4]. In addition, testing is used at step 2 to automatically decide at which level the clustering process stops.

VI. CONCLUSION

Motion segmentation is based on an adaptive back-ground subtraction method that models each pixel as a mixture of Gaussians. The adaptive tracking method overcomes lighting effects and object trajectories are effectively obtained.The HMM representation of object trajectories enables the measure of similarity between video events by cross likelihood but suffers from the overfitting problem due to data shortage. A novel dynamic hierarchical clustering (DHC) approach proposed in this paper, where the HMMs are trained on multiple samples and the initial clustering errors caused by over fit are corrected in the iterative process. The proposed method is not sensitive to the absolute similarity values and calculates the number of clusters automatically.

REFERENCES

[1] J. Ajmera and C. Wooters, “A Robust Speaker Clustering Algorithm,” in IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 411-416, December 2003.

[2] F. Porikli and T. Haga, “Event Detection by Eigenvector Decomposition Using Object and Frame Features,” in IEEE Conference on Computer Vision and Pattern Recognition.

[3] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan, “Semi-supervised Adapted HMMs for Unusual Event Detection,” in IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 611-618, June 2005.

[4] T. Xiang and S. Gong, “Video Behaviour Profiling and Abnormality Detection without Manual Labelling,” in IEEE International Conference on Computer Vision, vol. 2, pp. 1238-1245, October 2005.

[5] L. Zelnik-Manor and M. Irani, “Event-Based Analysis of Video,” in IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 123-130, 2001.

[6] H. Zhong, J. Shi, and M. Visontai, “Detecting Unusual Activity in Video,” in IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 819-826, July 2004.

[7] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, “Traffic Monitoring and Accident Detection at Intersections,” in IEEE Transactions on Intelligent Transportation Systems, vol. 1, pp. 108-118, June 2000.

[8] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia, “Event Detection and Analysis from Video Streams,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, pp. 873-889, August 2001.

[9] R. Duda, P. Hart, and D. Stork, “Pattern Classification,” by John Wiley & Sons, Inc. pp. 550-556, 2001. Workshop, pp. 114-114, June 2004.

[10] Fan Jiang, Ying Wu, Aggelos K. Katsaggelos, “Abnormal Event Detection From Surveillance Video By Dynamic Hierarchical clustering “.

[11] Ke-Xue Dai, Guo-Hui Li, Ya-Li Gan **“**A Probabilistic Model For Surveillance Video Mining”, Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian, 13-16 August 2006.

[12] Chris Stau_er W. Eric L. Grimson “Learning patterns of activity using real-time tracking” Artificial Intelligence Laboratory Massachusetts Institute of Technology Cambridge.