An Adaptive Scene Description for Activity Analysis in Surveillance Video

Brendan Morris and Mohan Trivedi
Computer Vision and Robotics Research Laboratory
University of California, San Diego
La Jolla, California 92093-0434
{b1morris,mtrivedi}@ucsd.edu

Abstract

This paper presents an adaptive framework for live video analysis. The activities of surveillance subjects are described using a spatio-temporal vocabulary learned from recurrent motion patterns. The repetitive nature of object trajectories are used to build a topographical map, where nodes are points of interest and the edges correspond to activities, to describe a scene. The graph is learned in an unsupervised manner but is flexible and able to adjust to changes in the environment or other scene variations.

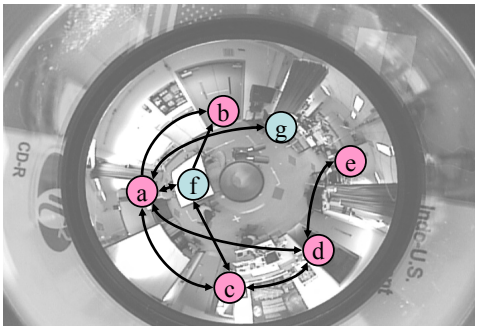


Figure 1. Topographical representation of a scene.

1 Introduction

Widespread use of cameras has generated huge volumes of data to analyze making it an almost impossible task to continually monitor these video sources manually. Methods to recognize certain events and activities of interest automatically are needed to provide a method to compress the video data into a more manageable form. This work develops an unsupervised framework for automatic activity analysis in surveillance video.

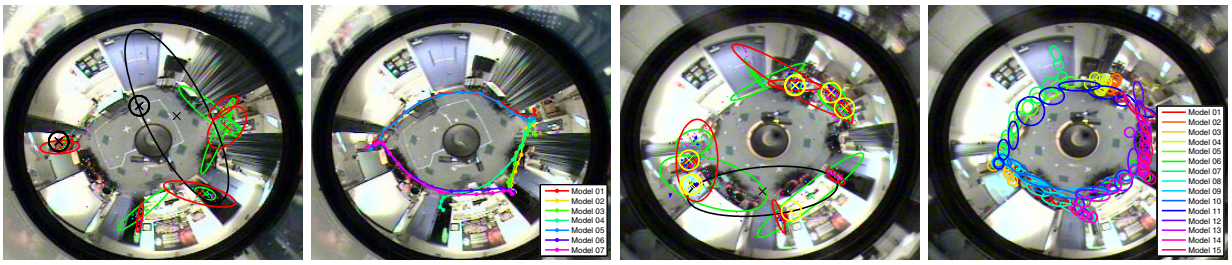
Rather than requiring specific domain knowledge, we restrict ourselves to surveillance applications where events of interest are typically evidenced by motion. Often the observed motion patterns in visual surveillance systems are not completely random but have some underlying structure which dictates the types of activities expected in a scene. Instead of manually configuring a system to a specific location, activity models used for accurate behavior inference can be automatically built through observation of the underlying motion distributions. Pioneering work by Johnson and Hogg [4] described outdoor motions with a flow vector, consisting of position and velocity, and learned paths using a Neural Network. Owens and Hunter [7] extended this work using a Self Organizing Feature Map to learn paths and further detect abnormal behavior. Hu *et al*. [3] sped up the path learning process by using an entire trajectory as the input feature for the path learning algorithms. They also introduced a method to make predictions based on their path models. Makris and Ellis [5] developed a method to learn the interesting regions in a scene as well as build spatial paths in an online fashion allowing adaption to new unseen trajectories.

This paper extends the above work to provide an adaptive framework for automatically analyzing a surveillance scene where activities are updated using an online refinement scheme as well as a batch update to introduce new activities into the scene behavior set.

2 Topographical Scene Description

The topographical scene map shown in Fig. 1 provides the vocabulary to describe object behavior. The nodes localize spatially points of interest (POI) and the motion along the graph edges are encoded in activity paths (AP). The map can be automatically constructed by using measurements obtained during tracking $F = \{f_1, \ldots, f_t, \}$. where the trajectory F consists of object dynamics $f = [x, y, u, v]^T$ for every time t.

978-1-4244-2175-6/08/$25.00 ©2008 IEEE



(a) (b) (c) (d)

Figure 2. Lab Omni experiments: (a) Learned POI, enter in green and exit in red. (b) Routes after merging. (c) Updated zones, notice there are now stop zones (yellow). (d) Set of HMM activity paths.

3 Points of Interest

There are three types of POI nodes, the entry zones (objects enter the scene), exit zones (objects leave the camera view), and stop zones (objects remain idle). These zones are modeled using a 2D Gaussian mixture model (GMM), $Z \sim \sum_{i=1}^{W} w_i N(\mu_i, \Sigma_i)$ with W components which can be learned using expectation maximization (EM). The entry dataset consists of the first tracking point, the exit set includes only the last tracking point, and the stop zone set consists of all tracking points with velocity below a predefined threshold [5] or all points that remain in a circle of radius R for more than τ seconds [1].

The zones are over mixed to completely model all true zones and noise sources, which are separated with a density criterion measuring the compactness of a Gaussian distribution [5]. Tight mixtures indicate true zones while wide mixtures imply tracking noise from broken tracks which are filtered for improved path learning later. Example POI are shown in Figs. 2(a) and 2(c) with green corresponding to entry zones, red to exit zones, yellow to stop zones, and black representing noise mixtures.

4 Activity Paths

The AP describe the typical motion patterns through the scene and are specified in a three step procedure. In the initial learning stage, the spatial configuration of the graph edges, or routes, are learned. Paths are formed by augmenting the routes with dynamic information to describe the spatio-temporal nature of activities. Finally the AP are temporally maintained by adapting to new data in an update phase.

4.1 Route Clustering

The routes, corresponding to xy position between POI, are found by clustering the training set of trajectories. Since tracks are not the same length, due to differing speeds and amount of time spent in the camera field of view, they are spatially resampled to a fixed length L. A resampled trajectory is designed to evenly distribute points along the track length and ensure the distance between consecutive points is equal. A flow vector [4] $F = [x_1, y_1, \ldots, x_L, y_L]$ ignoring velocity information is constructed from each training trajectory and represents a point in the $\mathbb{R}^{2L}$ route space. The space is over partitioned into N_c clusters using fuzzy C means (FCM), to minimize the effect of outliers, which returns cluster prototypes r_k and the membership of each of the N training trajectories u_{ik} to the prototypes.

The FCM route clustering finds prototypes r_k, but because the true number of routes N_p is not known a priori $N_c > N_p$. The true routes are found by merging similar clusters. Routes are compared using dynamic time warping (DTW) [8] to optimally align points and are considered candidates for merging if if all consecutive points are within a small radius or if the total distance between tracks is small. A cluster correspondence list is created from these pairwise similarities, forming similarity groups $\{V_s\}$. Each correspondence group is reduced to a single route

$$r(V_s) = \underset{z \in V_s}{\operatorname{argmin}} \sum_{i=1}^{N} |\hat{u}_{ik}(z) - \tilde{u}_{ik}(z)| \qquad (1)$$

by retaining only the cluster prototype that causes the maximal change in training membership when removed. The membership $\hat{u}_{ik}(z)$ represents the renormalized membership when prototype z is removed from the correspondence set and $\tilde{u}_{ik}(z)$ is the recomputed FCM membership if route z did not exist. Fig. 2(b) shows the routes learned after merging.

4.2 HMM Path Modeling

The object dynamics needed to characterize activities are incorporated into the AP by augmenting routes

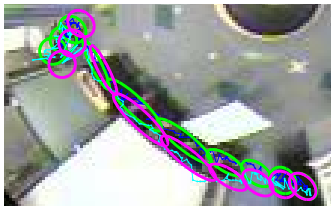


Figure 3. Path adapted using the incremental MLLR update (green = original, magenta = adapted).

(a) (b)

Figure 4. (a) Simulated intersection. (b) Interstate 5 highway.

and encoding spatio-temporal information with hidden Markov models (HMMs). The HMM path representation allows for probabilistic behavior analysis through Bayesian inferencing with simple training, evaluation, and adaption techniques.

An HMM is trained for each path by collecting N_p disjoint training sets, $D = \bigcup_{k=1}^{N_p} D_k$, containing all trajectories with high membership $u_{ik} > 0.9$. Only the high membership tracks are used in training set D_k to improve model precision through removal of outliers or ambiguous examples. Using each path training set D_k, the N_p continuous Gaussian emission HMMs ($\lambda_k = (A, B, \pi)$) can be efficiently learned using standard methods such as the Baum-Welch method or EM [8]. The set of HMM paths, shown in Fig. 2(d), complete the topographical scene representation and describe how objects move.

4.3 HMM Path Update

Since a surveillance scene is not static, the path processes are not guaranteed to be stationary. The AP adapt to changes using two complimentary methods. The first refines a matching model in an online fashion with new trajectories while the second uses a batch update procedure to introduce new activities.

When a new trajectory is generated from a particular AP, it can be used to update the associated HMM in an online fashion using maximum likelihood linear regression (MLLR) [2]. MLLR computes a set of affine transformations that will reduce the mismatch between the initial model set and new adaption data. The adapted mean is given by

$$\hat{\mu} = W\xi, \tag{2}$$

where W is the $d \times (d+1)$ transformation matrix and $\xi = [1, \mu_1, \ldots, \mu_d]^T$ is the extended Gaussian mean vector. W is estimated using EM. Each time a new trajectory is classified into path λ_k, a transformation is learned and applied to each of the HMM states for sequential update

$$\mu_{t+1} = (1-\alpha)\mu_t + \alpha W_t \xi_{kj} \qquad j = 1, \ldots, Q \tag{3}$$

of the mean with $\alpha \in [0, 1]$ a learning rate parameter. The online regression update is demonstrated in Fig. 3, where a path is blocked by a table an people are forced to walk around.

In order to introduce new activities into the AP set, we adopt the batch update procedure of Hu *et al*. [3] for model addition. Trajectories that do not fit any of the HMMs well are collected into a new training database and re-clustered (as done above) periodically allowing assimilation of once atypical motions given enough support.

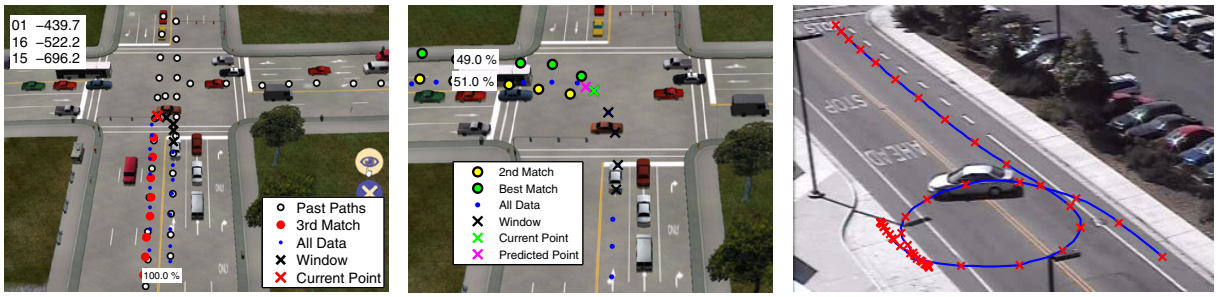
5 Studies and Experimental Analysis

The following section presents performance evaluation using the proposed topographical scene description by examining the accuracy of path classification, prediction, and abnormality detection. Each of these are evaluated by maximum likelihood estimation of the HMM models

$$\lambda^* = \underset{k}{\operatorname{argmax}}\, P(F|\lambda_k). \tag{4}$$

F is chosen appropriately as the full trajectory for classification or as a small time windowed set of points for live prediction and abnormality detection. Example analysis images are shown in Fig. 5. More details of the evaluation scheme can be found in [6]. The experiments consider a simulated traffic intersection (SIM), Fig. 4(a), a view of highway traffic on Interstate 5 (I5), Fig. 4(b), and an indoor laboratory scene from an omnidirectional camera (OMNI), Fig. 2. Table 1 summarizes the study results.

The intersection had 16 acceptable traffic maneuvers which were all accurately discovered with the cluster-merge procedure. In the I5 experiment we overclustered into 25 routes and then merged them into the 8 true lanes with 2 false lanes identified. These false lanes occurred because camera perspective caused localization variance. The lab scene does not have any well defined lanes but paths were mapped between doorways and desks. There were two separate omni experiments,



(a) (b) (c)

Figure 5. Example activity analysis. (a) Suspicious event detected as red X. (b) Left turn prediction. (c) Abnormal trajectory detection.

				live		
	N_p	lane assignment	abnormality	lane assignment	prediction	abnormality
SIM	16	327/327 = 100%	5/5 = 100%	3669/3978 = 92.2%	2871/3978 = 72.2%	40/46 = 87.0%
I5	8	879/923 = 95%	-	14045/14876 = 94.4%	13859/14876 = 93.2 %	-
OMNI1	7	26/26 = 100%	10/14 = 71.4%	1741/3139 = 55.5%	1454/3139 = 46.3%	756/945 = 80.0%
OMNI2	15	12/16 = 75%	15/18 = 83.3%	1457/2693 = 54.1%	1013/2693 = 37.6%	-

Table 1. Experimental Results

OMNI1 and OMNI2. The first experiment only contained 7 paths which were all correctly discovered. But the OMNI2 experiment was more complex, using the nodes shown in Fig. 1 there were 15 unique paths. Although the path learning system found 15 paths, 2 were incorrect noise paths. But, the 2 missing paths had little support in the training set.

Looking at Table 1 we see very high classification and abnormality detection rates through all the tests. The live results which use only a portion of the data (since the full trajectory is not available until the end of a track) are not as high. There is more confusion between paths when less data is available. This is especially apparent for the OMNI tests where there are multiple overlapping paths making them difficult to distinguish. It is apparent that the prediction accuracy is related to how complex the paths are, the straight lanes of I5 doing the best. Even with the difficulty of incomplete data tracking abnormalities are still detected at a high rate making it useful for live warning signals.

6 Conclusion

This paper presents an adaptive framework for live video analysis based on trajectory learning. A surveillance scene is described by a topographical scene map which is learned in unsupervised fashion to indicate interesting image regions and the way objects move between these places. These descriptors provide the vocabulary to categorize past and present activity, predict future behavior, and detect abnormalities.

References

[1] N. Brandle, D. Bauer, and S. Seer. Track-based finding of stopping pedestrians - a practical approach for analyzing a public infrastructure. In *Proc. IEEE Conf. Intell. Transport. Syst.*, pages 115–120, Toronto, Canada, Sept. 2006.

[2] M. Gales, D. Pye, and P. Woodland. Variance compensation within the MLLR framework for robust speech recognition and speaker adaptation. In *Proc. IEEE Intl. Conf. Spoken Language*, pages 1832–1835, Oct. 1996.

[3] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank. A system for learning statistical motion patterns. *IEEE Trans. Pattern Anal. Machine Intell.*, 28(9):1450–1464, Sept. 2006.

[4] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. In *Proc. British Conf. Machine Vision*, volume 2, pages 583–592, Sept. 1995.

[5] D. Makris and T. Ellis. Learning semantic scene models from observing activity in visual surveillance. *IEEE Trans. Syst., Man, Cybern. B*, 35(3):397–408, June 2005.

[6] B. T. Morris and M. M. Trivedi. Learning, modeling, and classification of vehicle track patterns from live video. *IEEE Trans. Intell. Transport. Syst.*, 2008. To be published.

[7] J. Owens and A. Hunter. Application of the self-organising map to trajectory classification. In *Proc. IEEE Visual Surveillance*, pages 77–83, July 2000.

[8] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Englewood Cliffs, New Jersey, 1993.