Human Activity Detection and Recognition for Video Surveillance

Wei Niu, Jiao Long, Dan Han, and Yuan-Fang Wang
Department of Computer Science
University of California
Santa Barbara, CA 93106

Abstract

We present a framework for detecting and recognizing human activities for outdoor video surveillance applications. Our research makes the following contributions: For activity detection and tracking, we improve robustness by providing intelligent control and fail-over mechanisms, built on top of low-level motion detection algorithms such as frame differencing and feature correlation. For activity recognition, we propose an efficient representation of human activities that enables recognition of different interaction patterns among a group of people based on simple statistics computed on the tracked trajectories, without building complicated Markov chain, hidden Markov models (HMM), or coupled hidden Markov models (CHMM). We demonstrate our techniques using real-world video data to automatically distinguish normal behaviors from suspicious ones in a parking lot setting, which can aid security surveillance.

1. Introduction

Video surveillance can be an effective tool for today's businesses—large and small—in security surveillance, production monitoring, and deterring predatory and purloining behaviors. Since the introduction of analog video surveillance systems back in the 1970s, tremendous strides have been made in sensing, storage, networking, and communication technologies. The consequence is that, instead of employing video surveillance mainly as an "after-effect" forensic tool, it is now feasible to deploy digital, network-based surveillance systems to provide interactive, real-time monitoring and surveillance. This research proposes a software framework for video analysis to enable robust and real-time human activity detection and recognition. Our research makes the following contributions:

1.) For activity detection and tracking, we propose a real-time, robust algorithm that is well suited for analyzing outdoor, far-field activities. In particular, we improve the robustness in activity analysis by providing intelligent control and fail-over mechanisms, built on top of low-level motion detection algorithms such as frame differencing and feature correlation. These mechanisms improve overall robustness and accuracy by maintaining tracking and recovering from both non-catastrophic errors (such as occasional, short periods of occlusion and silhouette merging) and catastrophic errors (such as long periods of disappearance of activities from a camera's field of view). These fail-over mechanisms include efficient multi-hypothesis tracking to disambiguate figures from cluttered background, and a two-level, hierarchical activity representation scheme for bottom-up data fusion and top-down information guidance.

2.) For activity recognition, we propose an efficient representation of human activities based on tracked trajectories. We have developed a scheme that distinguishes different interaction patterns among a group of people by identifying the unique signatures in the relative position and velocity of the participants' trajectories. These analyses can be performed efficiently, without building complicated Markov chain, hidden Markov models (HMM) [2], or coupled hidden Markov models (CHMM) [1] to describe individual activities and the interaction among them.

We demonstrate our techniques using real-world video data to detect and recognize behaviors in a parking lot setting. In particular, we are able to distinguish normal following behaviors from suspicious, and potentially dangerous, stalking behaviors, which can aid security surveillance.

2. Methods

The validation scenario is an outdoor environment, such as a mall and a parking lot. We assume that there are multiple surveillance cameras positioned strategically around the place of interest. These cameras can be stationary, mounted on a PTZ (pan-tilt-zoom) platform and executing a fixed sweep pattern, or under the interactive control of a human operator. The view volumes of different cameras can be disjoint or partially overlap. Under this scenario, we would like to automatically detect and track the activities and interaction of people in the scene, represent these activities efficiently, and classify the activities into benign (normal) and potentially dangerous (suspicious) categories.

2.1 Activity Detection and Tracking

Outdoor surveillance by and large falls in the far-field scenario, which is also assumed for this research. When people in the scene are sufficiently far away, we can approximately describe each person as a "blob" and use a single object state to describe the trajectory as a function of time. As shown in Fig. 1, the goal of activity detection and tracking is then to optimally infer the state vectors for people in the scene from multiple cameras $\mathbf{x}^{(i)} = [\mathbf{p}^{(i)}(t), \dot{\mathbf{p}}^{(i)}(t), \ddot{\mathbf{p}}^{(i)}(t)]^T, 0 \leq i < m$, where m is the number of cameras used, and fuse such 2D state estimates from multiple sensors to derive a consistent, global 3D estimate $\mathbf{X} = [\mathbf{P}(t), \dot{\mathbf{P}}(t), \ddot{\mathbf{P}}(t)]^T$. [1] [2]

Our contribution here is to propose a robust, yet still real-time, control and fail-over mechanism—on top of low-level frame differencing- and correlation-based tracking—to deal with noise, scene clutter, short periods of absence and merging of silhouettes, and long periods of occlusion of activities from a camera's field of view. Our formulation is based on the powerful hypothesis-and-verification paradigm, which has been alternatively christened as Monte Carlo filtering [7], particle filtering [9], genetic algorithms [4], condensation (conditional density propagation) [5], and Icondensation (importance-based condensation) [6].

Mathematically, all state estimation algorithms are geared toward estimating the following conditional probability in an iterative manner:

$$\begin{array}{ll} & p(\mathbf{x}_n^+|\mathbf{x}_0, \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3 \cdots, \mathbf{z}_n) \\ = & p(\mathbf{x}_n^+|\mathbf{x}_1, \mathbf{z}_2, \mathbf{z}_3 \cdots, \mathbf{z}_n) \\ = & p(\mathbf{x}_n^+|\mathbf{x}_2, \mathbf{z}_3, \cdots, \mathbf{z}_n) \\ = & \cdots \\ = & p(\mathbf{x}_n^+|\mathbf{x}_{n-1}, \mathbf{z}_n) \\ \propto & p(\mathbf{z}_n|\mathbf{x}_n^-)p(\mathbf{x}_n^-|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}) \end{array} \qquad (1)$$

where sensor data are denoted as $\mathbf{z}$ and states as $\mathbf{x}$ ($\mathbf{x}^-$ and $\mathbf{x}^+$ denote, respectively, the state before and after the sensor measurement made at a particular time is incorporated.) The difference is in the forms the state prior and the various noise processes inherent in sensor measurement can assume, and in the complexity of the state propagation process. The utility of a general hypothesis-verification formulation, over traditional linear state estimation algorithms such as Kalman filtering, is that the noise processes do not have to be Gaussian and state propagation does not have to be unimodal. This allows multiple competing hypotheses to be maintained and contribute to the state estimation. In this sense, hypothesis-verification is akin to a Bayesian estimator instead of an maximum likelihood estimator [2].

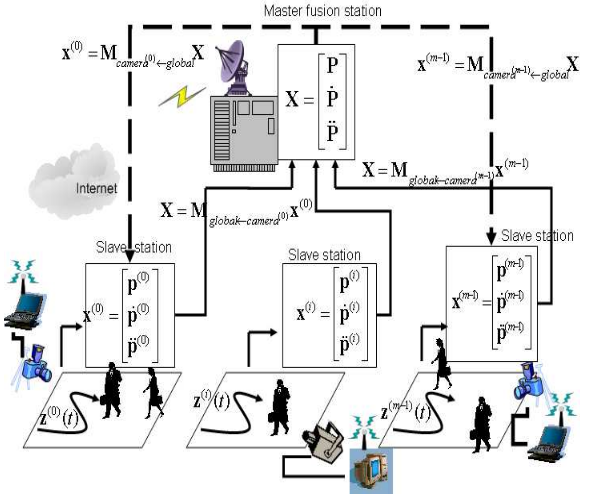
Master fusion station
$\mathbf{x}^{(0)} = \mathbf{M}_{camera^{(0)} \leftarrow global}\mathbf{X}$
$\mathbf{x}^{(m-1)} = \mathbf{M}_{camera^{(m-1)} \leftarrow global}\mathbf{X}$
$\mathbf{X} = \begin{bmatrix} \mathbf{P} \\ \dot{\mathbf{P}} \\ \ddot{\mathbf{P}} \end{bmatrix}$
Internet
$\mathbf{X} = \mathbf{M}_{global \leftarrow camera^{(m-1)}}\mathbf{x}^{(m-1)}$
$\mathbf{X} = \mathbf{M}_{global \leftarrow camera^{(0)}}\mathbf{x}^{(0)}$
Slave station
Slave station
Slave station
$\mathbf{x}^{(0)} = \begin{bmatrix} \mathbf{p}^{(0)} \\ \dot{\mathbf{p}}^{(0)} \\ \ddot{\mathbf{p}}^{(0)} \end{bmatrix}$
$\mathbf{x}^{(i)} = \begin{bmatrix} \mathbf{p}^{(i)} \\ \dot{\mathbf{p}}^{(i)} \\ \ddot{\mathbf{p}}^{(i)} \end{bmatrix}$
$\mathbf{x}^{(m-1)} = \begin{bmatrix} \mathbf{p}^{(m-1)} \\ \dot{\mathbf{p}}^{(m-1)} \\ \ddot{\mathbf{p}}^{(m-1)} \end{bmatrix}$
$\mathbf{z}^{(0)}(t)$
$\mathbf{z}^{(i)}(t)$
$\mathbf{z}^{(m-1)}(t)$

Figure 1: System configuration.

Different incarnations of the same hypothesis-verification principle all comprise the following essential components:[3] (1) A collection of candidate states and their likelihood estimates ($p(\mathbf{x}_{n-1})$ in Eq. 1) that are initialized, propagated, and updated over time, (2) a state propagation mechanism ($p(\mathbf{x}_n^-|\mathbf{x}_{n-1})$ in Eq. 1), (3) a state verification mechanism using sensor feedback ($p(\mathbf{z}_n|\mathbf{x}_n^-)$ in Eq. 1), and (4) a re-normalization process to refresh and/or regenerate the collection of candidate states and their associated likelihood ($p(\mathbf{x}_n^+|\mathbf{x}_o, \mathbf{z}_1, \cdots, \mathbf{z_n})$).

We employ frame differencing and region growing for the initial detection of presence of activities. When a moving region is identified in a camera's image, multiple hypotheses (up to the number allowed for real-time execution) are postulated. A single-person hypothesis is to represent such a region as a single moving person, characterized by a state comprising the estimated position, velocity, and acceleration of the moving region. Alternatively, the region might represent a group of up to n people, with their silhouettes merged into a single composite image region. Hence, multiple states are maintained to track these hypotheses.

The object position is propagated in time using the instantaneous velocity and acceleration recorded in the state vector. The predictions are validated against the image observation (i.e., consistent color and texture signatures over time). If tracking is lost due to scene clutter or the tracked object moving out of the camera's field-of-view, alternate hypotheses are formulated. These alternate hypotheses can be based on the object's signature discovered from a search in the reduced-resolution image over a wider image area, or

[1] To avoid unnecessarily complicating the notation, we do not use subscripts for different moving regions unless the discussion calls for it (e.g., interaction of two persons i and j). The discussion here applies to a single moving region. Multiple moving regions can be handled similarly.

[2] To fuse measurements from multiple sensors into one global estimate, two registration processes are needed: *spatial* registration to establish the transformation among different coordinate systems, and *temporal* registration to synchronize multiple local clocks. These techniques are well established in the literature, and we have developed algorithms in the past to accomplish both spatial and temporal registration [8, 3]. Due to space limit, these will not be repeated here.

[3] The difference is in the exact mechanism for realizing these. For example, standard condensation algorithms generate and update candidate states based strictly on the Bayesian formula, while importance-based condensation allows auxiliary information (e.g., from sensors) to be brought in to better position candidate states for efficiently exploring the state space.

formulated from the query result of the master fusion station in Fig. 1 (i.e., information from other cameras).

2.2 Activity Representation and Recognition

A significant amount of research has been done in the *structural* representation and recognition of human activities and interactions using Markov chains, hidden Markov models, and coupled hidden Markov models. Here, we strive to recognize individual and group activities and interactions based directly on statistical properties computed using the recovered trajectories, without elaborate parsing of the state vectors against pre-established Markov models. For example, loitering behaviors are characterized by large variance in the direction, but small variance in the position.

For multiple trajectories, we examine the relative position and velocity $[\mathbf{P}_{ij}, \dot{\mathbf{P}}_{ij}]^T = [\mathbf{P}_i - \mathbf{P}_j, \dot{\mathbf{P}}_i - \dot{\mathbf{P}}_j]^T$.[4] We slide a suitable windowing function (e.g, a box or Gaussian filter) over the trajectories as $w(t - t_n)[\mathbf{P}_{ij}(t), \dot{\mathbf{P}}_{ij}(t)]^T$. We then compute a best linear fit to these localized trajectory curves and the fitting error. Many activities can be recognized solely based on the fitting and error residue. Some examples of two-person interactions that can be distinguished based on the above statistics are:

Following behaviors: characterized by an almost constant relative position and a nearly zero relative velocity.

Following-and-gaining behaviors: characterized by a linearly-shrinking change in the relative position and a nearly constant, but nonzero relative velocity.

Stalking behaviors: similar to the above, but with much large variance in both relative position and velocity, and hence, large error residues in linear curve fitting.

In Fig. 2, we show sample trajectories detected from real video that depict following, following-and-gaining, and stalking behaviors (from top to bottom). The x-axis represents the time, and the solid (green) curve is the relative position and the dotted (red) curve is the relative velocity. To simply the illustration, only the x components of the relative position and velocity are shown. However, the distinction in the trajectory signatures is quite apparent.

3. Experimental Results

The results are to illustrate that our algorithm can automatically detect and track motion events, and classify them into benign and potentially dangerous categories. This ability can be useful in alerting parking lot attendants of abnormal events and behaviors that warrant investigation.

We collected about an-hour worth of real video in a parking lot. Three cameras were used to provide overlapping spatial coverage. We acted out many different sequences of following, following-and-gaining, and stalking behaviors. The actions were performed in such a way that occlusion, merging and splitting of silhouettes, entering and exiting a camera's field-of-view happened frequently to validate our robust tracking algorithm.

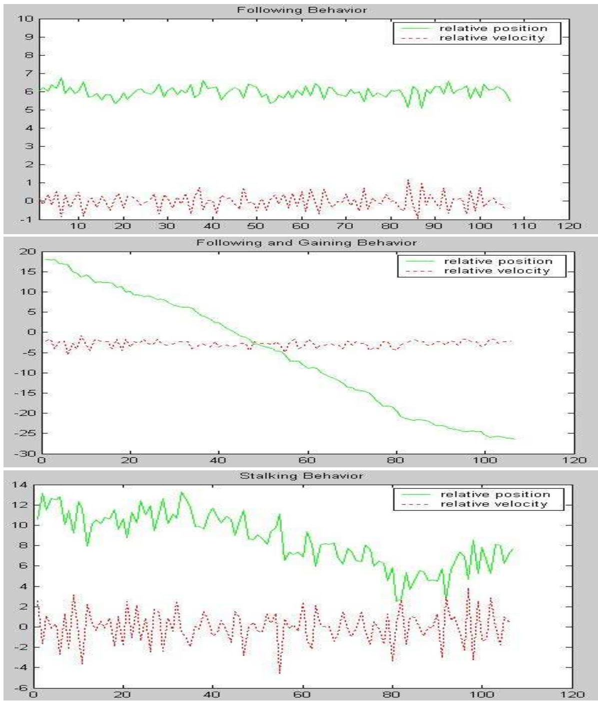

Figure 2: From top to bottom: following, following-and-gaining, and stalking behaviors. The green (solid) curves represent relative position and the red (dotted) curves represent relative velocity as a function of time.

Fig. 3(a) shows the trajectories of a stalker and the potential victim. Notice that the views of both cameras 2 and 3 were partially blocked by parked cars in the scene, and the scene shown in Fig. 3(a) was that of camera 1.[5] Fig. 3(b) shows the schematic drawing of the parking lot, the trajectories detected by the three cameras, and the fused trajectory. As can be seen that even individual trajectories show big gaps because of occlusion and missing data, the fused trajectory is complete. Similar results for following behaviors are shown in Fig. 4.

By comparing tracking results against those obtained using manual tracking (the ground truth), we estimated that the average error of the proposed tracking algorithm was less than 4 pixels. The error came from at least three sources: 1) Shadow of a person often resulted in extraneous foreground regions, 2) error in camera calibration caused mis-alignment of trajectories from multiple cameras both spatially and temporally, and 3) we did not employ sub-pixel processing to gain high accuracy. For multi-state tracking, we kept about 150 candidate states for a single moving region. The number of states kept was influenced

[4] Here the subscripts i and j denote different persons in the scene.

[5] The camera positions in the figure only indicate the general directions of camera placement. The actual cameras were placed much further away and pointed toward the parking lot.

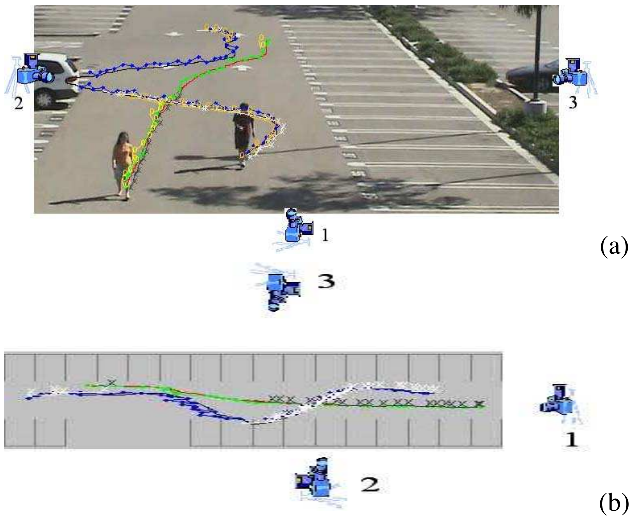

Figure 3: (a) A simulated stalking behavior in a parking lot and (b) trajectories of the sample stalking behavior. The "-" is the fused trajectory; "." is the tracked trajectory from camera 1; "x" is the tracked trajectory from camera 2; and "o" is the tracked trajectory from camera 3.

by the desired tracking accuracy and the real-time processing requirement. When resizing the video to 340×240, the tracking algorithm as configured ran at 20 fps on a Pentium IV 2.66G machine, and the multi-state tracking added only a 7% overhead in running time.

For recognition, we generated training data (i.e., trajectories of following, following-and-gaining, and stalking behaviors) using two different means: (1) by adding noise to real video data, and (2) by synthesizing trajectories drawn using a computer mouse. The feature we extracted from the data was a 6-by-1 vector $[S_p, I_p, R_p, S_v, I_v, R_v]$, where S and I are the slope and intercept of the best-fit linear line segment to a motion trajectory, and R is the residue error in linear fitting. Subscripts p and v denote the relative position and velocity curves, respectively.

We have about 30 instances of testing data for each of the following, following-and-gaining, and stalking behaviors. We use SVM with the Gaussian kernel function for training and classification, and the result is presented in Table 1.

Table 1: The confusion Matrix. F is following, FG is following-and-gaining, and S is stalking behavior.

Behaviors	*# of Instance*	F	FG	S
F	30	30	0	0
FG	28	1	25	2
S	26	3	2	21

The recognition rate was as high as 100% for the following and about 80% for the stalking behavior, which was better than algorithms using HMM and CHMM. Moreover, the complexity of a statistical approach is much lower than a structural approach which performs the additional task of learning complicated Markov models. Hence, a statistical approach might be more suitable here to distinguish between benign and suspicious behaviors.

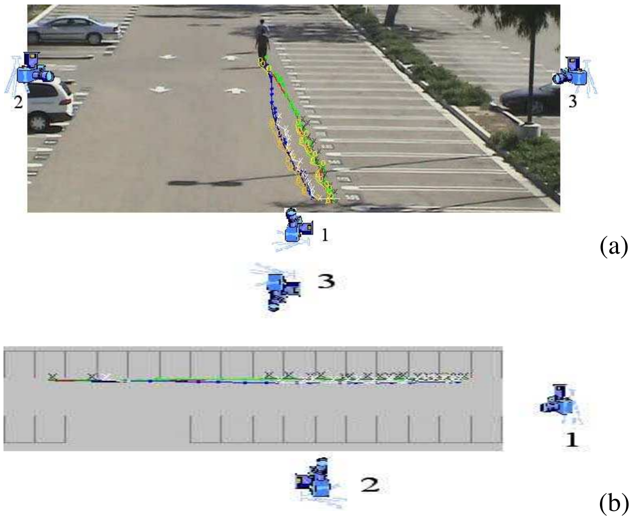

Figure 4: (a) A simulated following behavior in a parking lot and (b) trajectories of the sample following behavior. The "-" is the fused trajectory; "." is the tracked trajectory from camera 1; "x" is the tracked trajectory from camera 2; and "o" is the tracked trajectory from camera 3.

4. Concluding Remarks

This paper presents a framework for analyzing human activities. In particular, we address the issues of real-time detection and tracking of activities in a robust manner and an efficient activity representation and recognition scheme.

References

[1] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit.*, 1996.

[2] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification, 2nd Ed.* Wiley, New York, 2001.

[3] G. Wu and Y. Wu and L. Jiao and Y. F. Wang and E. Chang. Multicamera Spatial-temporal Fusion and Biased Sequence-data Learning for Security Surveillance. In *Proceedings of ACM Multimedia Conference*, pages 528–538, 2003.

[4] S. Haykin. *Neural Networks, 2nd ed.* Prentice Hall, Englewood Cliffs, NJ, 1999.

[5] M. Isard and A. Blake. Condensation–Conditional Density Propagation for Visual Tracking. *Int. J. Comput. Vision*, 29:5–28, 1998.

[6] M. Isard and A. Blake. ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework. *Lecture Notes in Computer Science*, 1406:893–908, 1998.

[7] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, 5:1–25, 1996.

[8] W. Niu, L. Jiao, D. Han, and Y. F. Wang. Real-time Multi-person Tracking in Video Surveillance. In *Proceedings of Pacific Rim Multimedia Conference*, 2003.

[9] M. K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94:590–??, 1999.