Proceedings of the 13th
Mediterranean Conference on Control and Automation
Limassol, Cyprus, June 27-29, 2005

Switching Kalman Filter-Based Approach for Tracking and Event Detection at Traffic Intersections

Harini Veeraraghavan Paul Schrater Nikolaos Papanikolopoulos *
Department of Computer Science and Engineering
University of Minnesota
{harini,schrater,npapas}@cs.umn.edu

Abstract– **Automatic event detection from video sequences has applications in several areas such as automatic visual surveillance, traffic monitoring for Intelligent Transportation Systems, key frame detection for video compression, and virtual reality applications. In this work, we present a computer vision-based approach for event detection and data collection at traffic intersections. Specifically, we make the following two contributions: (i) a robust tracking algorithm for targets through combination of multiple cues and multiple motion models, and (ii) a simple event detection system using the results of a switching Kalman filter in combination with some simple rules. We show the results of tracking and event detection, such as, turning, stopped or stalling vehicles, as well as motion statistics (average speeds and accelerations), collected for some outdoor traffic scenes.**

Keywords– **Event detection, switching Kalman filter, multiple cue-based tracking, cue combination.**

1 Introduction

Trajectory-based event recognition is the basis of most automated surveillance and monitoring applications. The events of interest for a given application domain could be a collection of pre-specified events or unusual or rarely occurring events. While the former just requires specification of the events of interest as domain specific knowledge or models, the latter requires identification of unique events not conforming to the standard pattern of learned activities in the scene.

The events of interest are generally learned or specified using supervised or unsupervised learning methods. Detection and learning methods vary in the degree of complexity ranging from just specification of particular events directly in the form of constraints in the scene [1], specific motion models, patterns of motion represented as hidden Markov models, or Markov random fields [2, 3], methods that obtain a low dimensional representation of the pattern of events in a scene [4, 5], etc.

In this work, we use a simple representation of the events of interest in the scene. The events of interest are based on the trajectory of the targets such as turn directions, lane changes, and the motion characteristics such as slow moving or stopped vehicles, speeding vehicles, etc. The motion characteristics are detected based on the motion models, specifically employing a switching filter used for estimating the target's motion, while the events such as turns are detected using simple rules as described in a later section of the paper.

The basic requirement for reliably detecting events using trajectory data is the accurate estimation of the target trajectories. Using vision-based methods for tracking in outdoor environments incurs problems in good target registration owing to ambiguities primarily introduced by (i) occlusions, and (ii) illumination variations. While occlusions result in data association ambiguity in the case of blobs, illumination variations such as shadows may alter the color of the target with respect to the viewing camera. Typical solutions to addressing such problems include, probabilistic data association [6], interacting multiple models, sampling-based approaches [7], etc. To address these issues, we make use of two different cues, namely, foreground segmented blobs obtained from an adaptive background segmentation method, and target color represented using a color histogram with probabilistic data association.

1.1 Multiple Motion Models for Tracking

Uncertain motion of the targets especially near intersections, makes a single model-based estimator unsuitable for accurate tracking. Methods for tracking maneuvering targets include, multiple hypothesis approaches, interacting multiple models, switching state space models [8, 9], mixture of filters [10], etc. In this work, we use a mixture of Kalman filters for modeling the different motions exhibited by the targets such as slow moving or stopping, turning, ac-

*author to whom all correspondence should be sent.

0-7803-8936-0/05/$20.00 ©2005 IEEE

celerating, and uniform velocity motion. The approach for tracking is depicted in Figure 1.

One difficulty in applying a switching Kalman filter to each cue is the high computational requirement of a switching Kalman filter, with the problem compounded by the requirement for data association. Hence, we use a two-step approach. In the first step, the combined position estimates obtained from the blob and color tracker are incorporated one after the other into an extended Kalman filter. The state estimate obtained from the extended Kalman filter is then incorporated into a switching Kalman filter to obtain the state estimate, its covariance, and the motion characteristics.

This paper makes the following two specific contributions: (i) we make use of a collection of simple, easy to track cues for obtaining reliable registration of the targets in real-time, and (ii) we show that the various events of interest can be detected employing the probabilities of the different filter models used for estimating the target trajectories.

The paper is organized as follows. Section 2 discusses the details of the tracking method, namely, the different tracking modalities used, the switching Kalman filter and the method for cue combination. Section 3 presents the details of event detection. The results of the tracking method and event detection are presented in Section 4, while discussion of the results and future work are in Section 5. Section 6 concludes the paper.

2 Tracking Approach

The basic tracking approach is outlined in the Figure 1. The system makes use of two different cues, namely, foreground blobs obtained through an adaptive region segmentation [11], and the target color represented as histogram, tracked using [12]. The results of the blob tracking are used for initializing new targets as well as a position measurement for the targets. The target color distribution is initialized from the blob associated to the initialized target. The measurement from the two different tracking modalities are combined sequentially in the estimator using two different filter settings, namely, (i) an extended joint probabilistic data association filter for blob measurements to deal with the ambiguities in measurements due to occlusions, and (ii) an extended Kalman filter in the case of color-based position measurements (since color retains more information peculiar to a target, thereby reducing ambiguities).

The result of the position estimates obtained after combining the two measurements is then passed to a switching Kalman filter. The switching filter makes use of three different motion models of different degrees (approximating

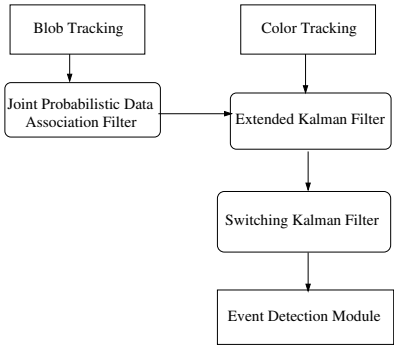

Figure 1: Tracking and event detection flow chart.

the behavior of vehicles in the scene), namely, a zeroth order or constant position, first-order or constant velocity, and a second-order or constant acceleration motion models as shown in Equation (3).

$$x_t = x_{t-1} + \omega_1 \quad (1)$$

$$x_t = x_{t-1} + \dot{x}_{t-1}\delta t + \omega_2 \quad (2)$$

$$x_t = x_{t-1} + \dot{x}_{t-1}\delta t + \ddot{x}_{t-1}\delta t^2/2 + \omega_3 \quad (3)$$

where x_t and x_{t-1} are the state estimates at times t and $t-1$, δt is the time interval, and $\omega_1, \omega_2, \omega_3$ denote the noise associated with each of the models. The advantage of using a switching Kalman filter framework is that at any given time, the motion ascribed to a vehicle is determined as a weighted combination of the three models, thereby providing more flexibility to describe their motion. The results of the target trajectory are then used by the event detection module, the details of which are in Section 3.

2.1 Measurements and Measurement Error Covariances

The measurements consist of the position of the target in the image obtained from: (i) the centroid of the blobs associated to a target, and (ii) the position obtained from the color-based mean shift tracker.

2.2 Blob Measurements

Since a blob is a very abstract representation of a target, very little information is retained in the blob to compute the error covariance in the measurement quantitatively. Two major sources of error in a blob measurement are: (i) noise or error in segmentation, and (ii) occlusions. Hence, the error is defined by two constant covariances Q_1 and $Q_2, Q_2 > Q_1$ depending on whether a blob measurement is obtained as a result of occlusion or not.

$$\begin{array}{ll} Q_1 & if\ no\ occlusion \\ Q_2 & otherwise. \end{array} \quad (4)$$

While occlusions resulting from tracked targets can be detected trivially using blob associations, occlusions resulting due to background or uninitialized targets cannot be detected trivially. For this purpose, we make use of a Mahalanobis distance-based gating scheme such that only measurements that truly arise from an unoccluded blob are incorporated while others are discarded.

2.3 Color-based Measurements

The procedure for locating targets using color is a mean shift tracking method proposed by [12], which uses a gradient descent procedure for locating the target. The procedure searches for the location in the image which minimizes the difference between the original target density q and the target density p computed at a given location. Interested readers can find details of the method in [12].

The error in a measurement is computed using the Sum-of-Squared Differences (SSD) metric as in [13]. The error in measurement is computed as the standard deviation of a scaled Gaussian centered at the measured target location. This error corresponds directly to the certainty of the target location since the more peaked the distribution is, the higher the confidence and lower the error are. The error e along the x and y coordinate measurements is computed as,

$$e = \sum_{i=-N:\{i\in\Delta w\times M \,\wedge\, i\neq z\}}^{N} [SSD(i)-SSD(z)][SSD(i)-SSD(z)]' \tag{5}$$

where $(-N,N)$ is the region along which the SSD error is computed, Δw is the window size or interval used for computing the error, which in our case is $\Delta w = 5$ (since a 5x5 region is used for computing the SSD match). M is an integer corresponding to the maximum number of pixels distant from the mean position z where the feature is detected in the current frame.

2.4 Switching Kalman Filter

Switching Kalman filters are basically switching state space models which maintain a discrete set of (hidden) states (filter models) and switch between or take a linear combination of them. The main difference from a hidden Markov model is a real-valued switch variable that allows to use a weighted combination of the different state space models. This allows to model a wider variety of operating conditions of the system. To overcome the problem of exponential belief growth with time, operations such as collapsing, selection, and variational methods are frequently used. For details, interested readers can refer to [14]. The standard Kalman filter recursions are performed in each of the filters with the exception of the computation of an innovation likelihood term. This term differs from the standard update in

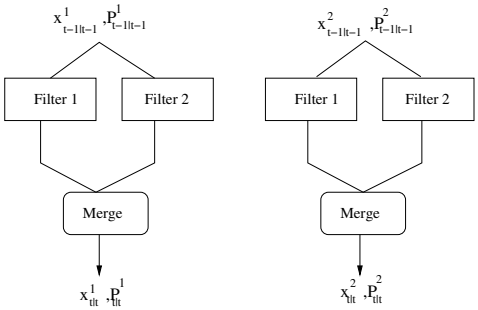

Figure 2: Moment matching of filters.

a Kalman filter in that the residual in the standard Kalman filter corresponds to the difference between the measurement and the predicted state estimate, while in a switching Kalman filter, the residual corresponds to the difference between the predicted state estimates of the models i and j. In other words, the likelihood term corresponds to the likelihood that the current model at time t is j, given that the model at time $t-1$ was i. It is computed as,

$$L_t = N(\Delta_t; 0, C_t) \tag{6}$$

$$\Delta_t = x^t_j - x^t_i \tag{7}$$

where Δ_t is the filter residual between models i and j and C_t is the inverse of the covariance in the filter j combined with the measurement error covariance. The switch parameter is updated as,

$$\begin{aligned} E^{t-1,t}_t(i,j) &= P(S_{t-1}=i, S_t=j|Y_{1:t}) \\ &= \frac{L_t(i,j)\Phi(i,j)E^{t-1}_{t-1}(i)}{\sum_i\sum_j L_t(i,j)\Phi(i,j)E^{t-1}_{t-1}(i)} \end{aligned} \tag{8}$$

$$E^t_t(j) = \sum_i E^{t-1,t}_{t-1}(i,j) \tag{9}$$

$$\begin{aligned} W^{i|j}_t &= P(S_{t-1}=i|S_t=j, 1:Y_t) \\ &= \frac{E^{t-1,t}_t(i,j)}{E^t_t(j)} \end{aligned} \tag{10}$$

where i, j correspond to filter models. $\Phi(i,j)$ corresponds to the conditional probability of the switch variable being j at time t, given that the switch variable at time $t-1$ is i with the data from $1:t-1$.

The collapse operation is the same as the moment matching of Gaussian distributions. The equations for collapse operation are as follows:

$$x^t_t(j) = \sum_i x^t_t i(j) P^t_t i(j) \tag{11}$$

$$P^t_t(j) = \sum_i P^t_t i(j) W^{i|j}_t \tag{12}$$

where the value of the switch node inside the parenthesis corresponds to the switch node at time t and the one outside to time $t-1$. The collapsing operation is pictorially depicted in Figure 2.

The targets are tracked in the world coordinates. The outputs from the single model Kalman filters used for incorporating the two measurements are used as inputs to the filters in the switching filter. The estimated state covariance from the single model Kalman filter is used as a measurement error for these filters. The filters employed in the switching framework consist of:

- Constant position $[x\,y]$ where x and y correspond to the position of the target in the scene,
- Constant velocity $[x\,y\,\dot{x}\,\dot{y}]$ where $\dot{x}$ and $\dot{y}$ correspond to the velocity of the target in the scene, and,
- Constant acceleration $[x\,y\,\dot{x}\,\dot{y}\,\ddot{x}\,\ddot{y}]$ where $\ddot{x}$ and $\ddot{y}$ correspond to the target acceleration in the scene.

3 Event Detection

The input to the event detection module consists of the target's trajectory and the filter model probabilities obtained from the switching Kalman filter. As mentioned earlier, all the events of interest are specified using simple domain rules and discussed in Section 3.1 and Section 3.2. The events of interest consist of the following:

- Target trajectory based: turning, lane changes, and
- Direction and speed based: slow moving or stopped; speeding vehicles.

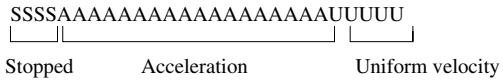

Figure 3: Event representation using runs. The different letters marked U, A, and S correspond to different events such as, uniform velocity, uniform acceleration, and stopped or slow moving, respectively.

A target's trajectory along a path is characterized into any of the following modes, namely, (a) right turn, (b) left turn, (c) uniform motion (or motion with constant velocity), (d) accelerating or decelerating, (e) velocity over the speeding threshold, and (f) slow moving or stopped. At any instant, the estimates of a target's motion are used to classify its motion into any of the above mentioned modes.

These modes are then collected as runs along the entire sequence of a target's trajectory. An example run is shown in Figure 3. From these runs, the motion of the target during a certain time duration can be extracted. For instance, in Figure 3, the most significant motion during the time interval (0 to 30 frames) is acceleration motion indicated by **A**. In order to discount noise in the position estimates, we take samples of a target's motion separated by certain time period. Further, it is necessary that the samples of the position trajectory be separated by at-least a minimum time interval δt for turn detection. The details of the different events detection are discussed in Section 3.1 and Section 3.2.

3.1 Trajectory-Based Event Detection

The events detected based on the trajectory of the target are turns (right and left), and lane changes. Lane changes are treated similarly to turn detection with the difference that lane changes are shorter turns compared to actual turns. Hence, the main difference between a lane change and turn is the length of a turn run. The basic method for computing the turn direction is depicted in Figure 4. As shown in the

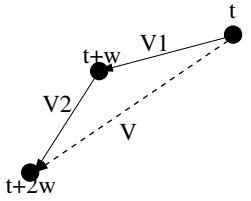

Figure 4: Turn computation.

figure, the turn direction is computed from the resultant of the vector connecting the positions sampled at the time intervals, t, $t+w$, and $t+2w$. The direction of the resultant vector corresponds to the direction of motion. The angle of the resultant vector has a specific relation to the turn direction as shown in Figure 5. This can be expressed as follows:

$$\begin{aligned} if\ |a| > \lambda 1 \wedge |a| < \lambda 2, left \\ if\ |a| > \lambda 2 \wedge |a| < \pi - \lambda 1, right \\ otherwise, straight \end{aligned} \tag{13}$$

where $\lambda 1$ is the threshold angle below which the motion is along a straight path and $\lambda 2$ is the threshold for separating the motions in the right and left directions. This is illustrated in Figure 5.

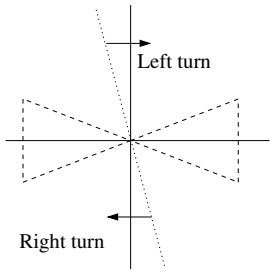

Figure 5: Direction of motion related to resultant vector direction.

(a) Frame 2730 (b) Frame 2814 (c) Frame 2847

Figure 6: Tracking in a crowded scene.

3.2 Motion-Based Event Detection

These events are detected based on the motion parameters such as the velocity and acceleration of the target. The detected events include, slow moving or stopped, moving at speeds over the pre-specified speed limit, acceleration, and uniform motion. Vehicles moving at speeds over the speed limits are simply detected by comparing their mean speed during a fixed interval with the speed threshold. Slow moving or stopped motion is obtained whenever the filter corresponding to constant position gets the highest probability. Similarly, acceleration and uniform motions are obtained from the corresponding filters for vehicles detected to be moving along a straight path.

4 Results

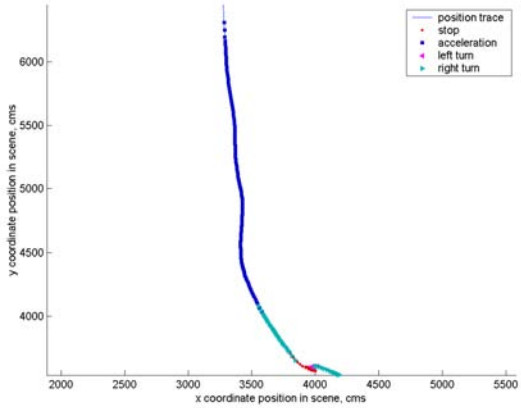

Figure 7: Right turning sequence.

Figure 6 shows an example of tracking in a fairly crowded scene. As shown, most of the targets continue to be tracked reliably despite the occlusions and absence of any information from the blob tracker. The contours around the vehicles correspond to the blobs returned by the blob tracker. Figure 7 shows the result for a right turning vehicle. The stopped or slow moving portions correspond to the vehicle waiting for pedestrians before completing its right turn. The results are in world coordinates, expressed in cm. The events are overlaid on the target's trajectory. Figure 8

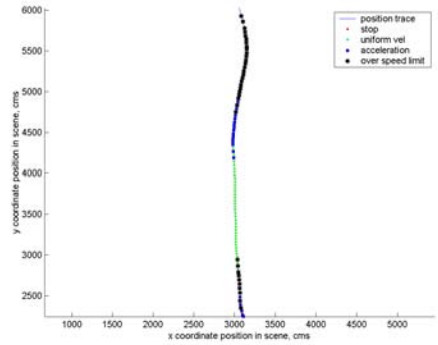

Figure 8: Speeding event detection.

shows the trajectory with the detected events for a vehicle moving at speeds higher than the user defined speed limits (34 mph in our experiments), with its speed and acceleration profile depicted in Figure 9. The constant acceleration in Figure 9 is the result of the constant velocity model having a higher probability as a result of which the acceleration is no longer updated.

5 Discussion and Future Work

As indicated by the results, the use of multiple cues coupled with multiple motion models helps increase the accuracy and robustness of the tracking. All of our experiments were carried out using video sequences obtained from two different outdoor traffic intersections, each about 30 min long. The system showed an accuracy of 88% for tracking. Tracking failures result predominantly owing to poor resolution of distant targets, persistent occlusions, as well as false segmentation (true targets classified as background

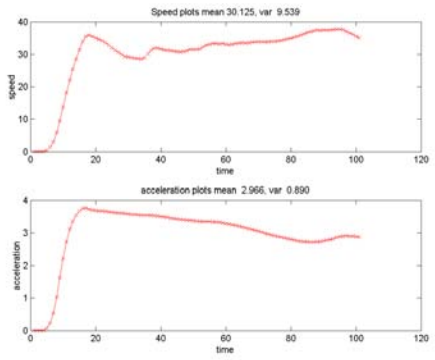

Figure 9: Speeding and acceleration plots.

or background portion classified as foreground). While the mean shift tracker provides good results for targets with distinct color, the results deteriorate as the similarity of the target color distribution with the background or other nearby targets increases.

Since the event detection is handled through simple rules and the results of the switching filter, it can be performed in real-time with very little computational overhead. While most of the events are classified correctly when using the entire string for classification, there are occasional misclassifications in the substrings due to (i) incorrect estimation of target motion by the filters, and (ii) thresholds used for event detection (e.g., turns). Overall, the misclassification rate was about 3%.

Currently, the detected events are very simple based on simple rules for detection. For example, the system makes no distinction between a target stopped at an intersection versus a target stopped at non-stopping portions of the scene, although the latter is a more interesting event compared to the former. Similarly, while the system can detect turns, it cannot detect more complex events such as U-turns. Future work in the event detection involves developing more complex rules or learning mechanisms for detection of such events.

6 Conclusions

This paper presented a method for automatic event detection for outdoor traffic applications. Event detection is based on the results of a switching Kalman filter for modeling different target motions and simple rules for deployment in a real-time application such as traffic monitoring. The paper also presented a method for robust target tracking and estimation based on combining multiple cues such as motion segmented blobs and target color in association with a switching Kalman filter. We also showed results obtained for a data collection system on two different traffic scenes.

7 Acknowledgements

This work has been supported in part by the National Science Foundation through grant IIS-0219863, Architecture Technology Corporation, the Minnesota Department of Transportation, and the ITS Institute at the University of Minnesota.

References

[1] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia. Event detection and analysis from video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):873–889, August 2001.

[2] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi. Traffic monitoring and accident detection at intersections. *IEEE Transactions on Intelligent Transportation Systems*, 1(2):108–118, June 2000.

[3] M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):844–851, August 2000.

[4] L. Lee, R. Romano, and G. Stein. Monitoring activities from multiple video streams: establishing a common coordinate frame. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):758–767, August 2000.

[5] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activities in video. In *Proc. Computer Vision and Pattern Recognition Conf.*, volume 2, pages 819–826, June 2004.

[6] Y. Bar-Shalom and T. E. Fortmann. *Tracking and data association*. Academic Press, 1987.

[7] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[8] V. Pavlović, J.M. Rehg, and J. MacCormick. Learning switching linear models of human motion. In *Neural Information Processing Systems*, pages 981–987, 2000.

[9] Z. Ghahramani and G.E. Hinton. Switching state-space models. *Neural Computation*, 12(4):831–864, April 2000.

[10] R. Chen and J.S. Liu. Mixture Kalman filters. *Journal of Royal Statistical Society Series-B Statistical Methodology*, 62(Part 3):493–508, 2000.

[11] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. Computer Vision and Pattern Recognition Conf.*, June 1999.

[12] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. In *IEEE Transactions Pattern Analysis and Machine Intelligence*, volume 25, 2003.

[13] K. Nickels and S. Hutchinson. Estimating uncertainty in SSD-based feature tracking. *Image and Vision Computing*, 20(1):47–58, January 2002.

[14] K.P. Murphy. Switching Kalman filters. Technical report, U.C. Berkeley, 1998.