

# Smart Video Systems in Police Cars

Amirali Jazayeri, Hongyuan Cai, Mihran Tuceryan, Jiang Yu Zheng

Department of Computer & Information Science

Indiana University Purdue University Indianapolis (IUPUI)

Indianapolis, Indiana 46202, USA

[amirali.jazayeri@gmail.com](mailto:amirali.jazayeri@gmail.com), [cai@cs.iupui.edu](mailto:cai@cs.iupui.edu), [tuceryan@cs.iupui.edu](mailto:tuceryan@cs.iupui.edu), [jzheng@cs.iupui.edu](mailto:jzheng@cs.iupui.edu)

## ABSTRACT

The use of video cameras in police cars has been found to have significant value and the number of such installed systems has been increasing. In addition to recording the events in routine traffic stops for later use in legal settings, in-car video cameras can be used to analyze in real-time or near real-time to detect critical events and notify police headquarters for help. This paper presents methods for detecting critical events in such police car videos. The specific critical events are person running out of a stopped car and officer falling down while approaching a stopped car. In the above situations, the aim is to alert the control center immediately for backup forces, especially in the last example when the officer is incapacitated. In order to implement real-time video processing so that a quick response can be generated without employing complex, slow, and brittle video processing algorithms, we use the reduced spatiotemporal representation (1D projection profile) and Hidden Markov Model to detect these events. The methods are tested on many video shots under various environmental and illumination conditions.

## Categories and Subject Descriptors

I.4.8 [Scene Analysis]: Object recognition, Motion, Tracking, Time varying imagery; I.2.10 [Vision and Scene Understanding]: Video Analysis; I.4.9 [Applications].

## General Terms

Algorithms, Performance, Experimentation.

## Keywords

Police video, motion detection, critical event, robustness, flow, data reduction.

## 1. INTRODUCTION

Police cars in the US are typically equipped with video cameras to record the events in routine traffic stops or emergency situations. The use of video in police cars has been found to have significant value and the number of such installed systems is increasing [8]. Most of the stops made by police are simple transactions and no investigative follow-up is needed. However, a few do require an investigation. In the simple cases, a basic video recording of the event is sufficient to assure that nothing untoward occurred. In the more serious cases, it is common for the video recording to be

given to a forensic video analyst for in-depth study. The common  $640 \times 480$  resolution tapes and similar quality digital recordings are generally sufficient for the simple cases. For in-depth study, more quality is expected to result in more conclusive findings. In addition, the in-car video cameras can be used to analyze in real-time or near real-time to detect critical events and notify headquarters for help. This feature adds real-time capability to what is normally an after-the-fact tool.

The traditionally accepted rationale for recording law enforcement related activities on video is to help advance prosecutions and case resolution; but most recently, systems have been designed to monitor the video stream and determine if certain critical events are taking place in real-time. Some examples of critical events include a package thrown out of a pursued car, door of a stopped car opens, passenger runs out of a stopped car, and officer falls down during a stop. By integrating such intelligence into the already existing video equipment, the video can potentially promote police officer's safety and possibly call for reinforcements automatically. In this paper we present an approach for a subset of the critical events listed above: passenger running out of stopped car and officer falling down.

During a routine traffic stop, the door of the stopped car opening or a person running out of it signal that the scenario could evolve into a non-routine and dangerous situation, the officer may have to pursue the passengers of the stopped car, and backup reinforcements may need to be called. By detecting this event the system can initialize the high definition video capture of the suspect for facial recognition and alert the police headquarters for further assistance while the officer on scene can focus on the current situation with the stopped car. Similarly, the detection of an officer down event could potentially represent serious threat to his safety, as it can result from being shot or attacked.

Because of the extremely large variation in the background scenes, illumination, target person, and action, a robust approach giving explicit classification of the events should be developed according to human motion. We utilize 1D projection profiles of the video frames to detect some of the critical events such as person running out of the car and Hidden Markov Model (HMM) to do officer falling down event detection. Prior to this, we have used the 1D projection profile representation for localization of the pursued car in the video frame and verified its effectiveness in car localization and tracking. HMM's have been used for gesture recognition before [12], but we have not seen it used for fall detection.

There are many related works in human pose and event detection [1, 3, 6, 9, 11, 12, 13]. Marker based methods are commonly used in the industry, however they are considered very intrusive and expensive, and not easy to apply in outdoor situations. Shape based methods are a more feasible alternative but their use are limited to clean backgrounds and controlled

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*MM'10*, October 25–29, 2010, Firenze, Italy.

Copyright 2010 ACM 978-1-60558-933-6/10/10...\$10.00.

environments. Because of the lack of robustness, subsequent sophisticated decisions can rarely be made in real time so far.

## 2. REPRESENTATIONS AND FEATURES

Our approach is based on data reduction for temporal analysis of the spatial events in the video frames such as a person running and a Hidden Markov Model (HMM) representation of the person falling down based on 2D features in the video frame.

Once the pursued car is located in the 2D frame using our earlier method [4], certain events such as person running out of the car and door of the car opening will take place in the immediately adjacent regions of the car in the video frame. This information, along with the fact that these events will be reflected in the projection of the video frame onto the  $x$ -axis will give us cues about important changes in the scene.

**1D Vertical Projection Profile:** Our reduced representation of the data is computed by projecting each video frame onto the  $x$ -axis. Because many events near the car in front are horizontal motion in the video field of view, we vertically project the intensity  $I_{R,G,B}(x, y, t)$  in each of the RGB channels in each video frame at time  $t$  to form a 1D profile  $T_{R,G,B}(x, t)$  as shown in Fig. 2a. We will simplify the notation and drop the color channel subscripts in the rest of the paper. The vertical projection of intensities is computed by

$$T(x, t) = \sum_{y=-h/2}^{h/2} K(x, y)I(x, y, t) \quad (1)$$

where  $h$  is the image height, and  $K(x, y)$  is a weight mask (defined in more detail in [4]) around the vehicle which has high values near the doors. The car tracking and position localization have been described elsewhere [4]. The traces in the condensed spatiotemporal image show movement patterns in the video.

Detection of a person falling down is a more difficult task and needs more careful selection of features upon which our algorithms will depend. We have examined a variety of video clips and selected features that can help in the reliable detection of a person falling down.

**Background Subtraction:** We detect the moving blobs in the scene by using a background subtraction method based on motion. We use the Mixture of Gaussian background subtraction algorithm proposed by Kaewtrakulpong and Bowden [5]. Morphological post processing is used to clean up the blobs detected. The blobs are also filtered using a size threshold to remove tiny ones from the scene.

**Center of Gravity (centroid):** We perform a connected component analysis of the detected blobs to identify and separate them. The centroid location of each blob is used in the HMM model described below.

**Movement Gradient Angle:** We compute a movement direction based on the motion of the detected blobs from frame to frame. We keep a history of the detected blobs with the interior filled in with a solid value. This value decays over time for past frames. We compute the gradient direction of the edges between such blobs attenuated by the gradient magnitude. We compute the average gradient direction along the boundaries for a global movement direction of the blob over a small number of past frames. This information is used later in estimating the probability of a falling down event.

Hidden Markov Model (HMM) [12] is used to model the probabilistic formulation of fall detection. A person falling down has time dependency that HMM is well suited for modeling. Using a probabilistic formulation to model the falling down

avoids the use of sensitive thresholds in the task of motion classification as *falling* and *not falling*. In the implementation of HMM, an increase of the minimum number of frames in the sequence results in a better accuracy. The longer we track a target, the higher the probability of a correct identification will be.

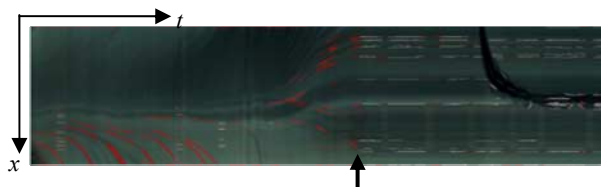
## 3. DETECTION OF CRITICAL EVENTS

We use these features and methods to detect certain critical events in the video. These are the detection of the car stop, detection of the person running out of a car, and detection of an officer falling down.

### 3.1 Detecting the stopping of the car

The point in time in which the pursued car stops can be detected by analyzing the computed 1D projection profile. When the camera is in motion, the traces of points on the static background in the spatiotemporal profile resemble hyperbolas [4]. The slope of the hyperbolas at time  $t$  is determined by the camera speed, the distance to the camera, and the camera parameters. When the camera's speed approaches zero, the speed of the background relative to the camera approaches zero, causing the traces to be straight lines parallel to the time axis. By detecting the moment in time when straight lines start to appear in the spatiotemporal profile, we can accurately detect the moment when the car stops.

We extract high gradient points along the time axis using Canny edge detection [2]. We use standard Hough transform on the edges detected in a small time window in this Canny edge image to detect straight lines. When we detect a significant number of line segments that are parallel to the time axis, we conclude that the vehicle has come to a stop (Fig 1).



Detected point in time when car came to a stop.

**Fig 1: The stopped car detection in the 1D projection profile computed. Detected traces in red indicate car moving and traces in white indicate car stopped.**

### 3.2 Detecting Person Running out of a Car

We use the reduced data representation given by Eq. (1) to detect a person running out of a stopped car ahead. Such an event is reflected in the 1D projection profile as a trace moving toward the edges of the profile in an otherwise smooth and unchanging background (see Fig. 2(a)).

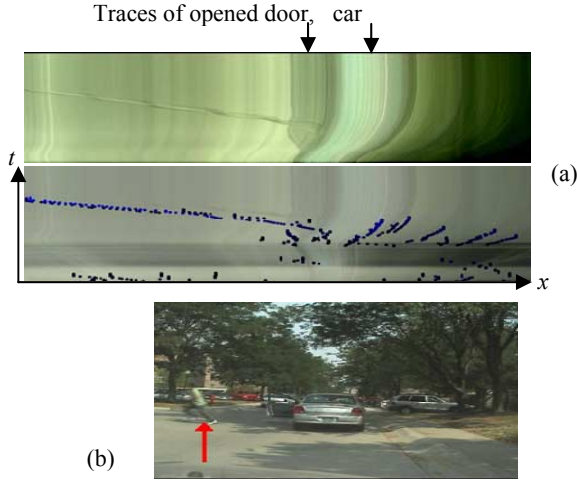
The distinguishable characteristics of this trace are:

- The trace is close to horizontal, leading away from the position of the vehicle.
- Unlike many other traces such as cars, it originates from the detected vehicle.
- The speed of a running person is limited when compared to other moving vehicles in the background scene.

The features represented in this temporal 1D projection profile are the high gradient points in this representation. We compute these points and track them over time using a simple Kalman filter [10]. The Kalman model assumes a first order, constant acceleration model in 1D. The extracted traces are matched against the position of the vehicle obtained from our earlier work. For each time instance  $t$ , we identify the candidate points in the 1D profile,

and select the ones whose distance to the car boundary is less than a threshold,  $\tau$ . The threshold is selected based on the average car door size ( $\tau = 55$  pixels was used in this paper with our camera and distance configuration). These selected points are labeled as candidates and their behavior is monitored using their movement vector calculated from  $P_t - P_0$ , where  $P_t$  denotes the location of the trace at time  $t$  and  $P_0$  denoted the location where the trace started.

If a trace matches the characteristics of a running person, an alarm is raised and the location of the end point of the trace is indicated as the person running out of the vehicle (see Fig. 2(b)).



**Fig. 2: 1D projection profiles vs. time representation: (a) Top: projection data, bottom: detected running person candidates (blue dots). (b) Final detection result of person running out of car (red arrow) against a noisy background.**

### 3.3 Detecting Falling Down Events

A falling down event can be recognized by extracting the centroid of a blob and monitoring its movement over time that are indications for falling down. We separated potential blobs that represent the police officer and other blobs in the background (e.g., cars, trees waving, etc.) by analyzing the point of entry of the blob into the field of view, i.e., from the left or right corner of the video frame. Blobs that do not meet these criteria are

discarded as unrelated.

A tracking algorithm is initialized for each relevant blob; this tracking monitors the  $t + 1$  position of the blob related to  $t$ , and extracts the movement vector  $V_t$ . The trace of the blob centroid over time provides strong evidence for a falling down event with a sharp downwards action; the traces which belong to other moving objects not falling down manifest themselves as mostly linear lines with speed vectors without sudden changes.

We model the motion vectors in a *Falling Down* event with HMM to determine the state of the blobs as falling movement and normal movement. Two hidden states are assigned as Falling  $F_i$  and Normal Movement  $N_i$  to describe every candidate blob  $i$ . The observable features for each track are the movement gradient angle,  $a_i$ , and the magnitude,  $d_i$ , of the vector connecting the centroid of a blob in time  $t$  to  $t + 1$ , which are both described in a continuous space rather than a discrete one. These two features create our movement vector  $V_i(d(t), a(t))$  that represents the blob trace over time.

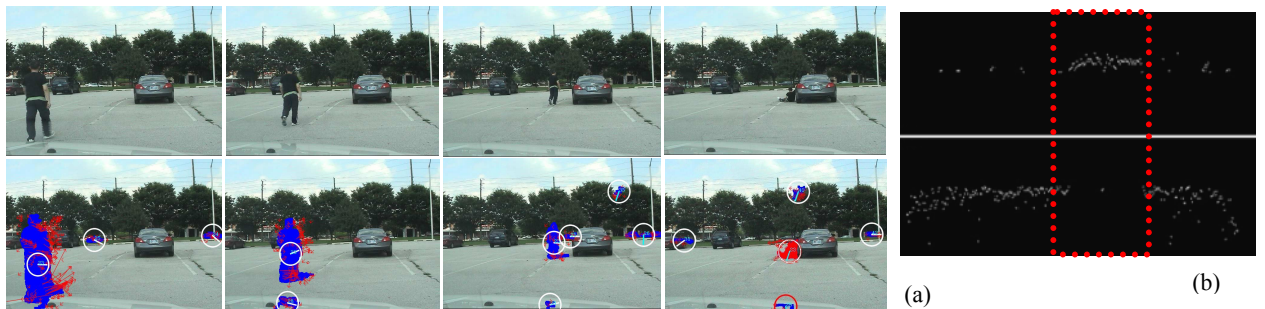
The likelihoods are denoted by  $P(F_t|d(t), a(t))$  and  $P(N_t|d(t), a(t))$  respectively. As parameters of HMM, we keep  $P(F_t) + P(N_t) = 1$  at any time  $t$ . The state transition probabilities from frame  $t - 1$  to frame  $t$  are defined as

$$\begin{aligned} P(F_t|N_{t-1}) &= 0.2 & P(N_t|N_{t-1}) &= 0.8 \\ P(F_t|F_{t-1}) &= 0.6 & P(N_t|F_{t-1}) &= 0.4 \end{aligned}$$

The transition probability values were set empirically.

The Viterbi algorithm was used to estimate the probability of a fall given  $V_t$  [7]. As we track each candidate blob, HMM is applied to the trace to update its state, i.e., Falling or Normal Movement. At every  $t$ , the obtained probabilities of a trace  $P(F|d, a)$  and  $P(N|d, a)$  are normalized in order to avoid a quick decrease of probabilities to zero.

The prior probabilities for the HMM model are based on the direction and magnitude of the motion vector in the image. We model this as a Gaussian density in which downwards motion vectors have a higher prior probability for *Falling* down and other directions have a higher probability for *Not Falling* down. The density is governed by the usual Gaussian density  $G(x, y; x_0, y_0, \sigma_x^2, \sigma_y^2)$ , in which  $(x_0, y_0)$  is where the Gaussian is centered and  $(\sigma_x^2, \sigma_y^2)$  are the variances in the  $x$  and  $y$  directions, respectively. We use one Gaussian  $G_F$  for Falling and one Gaussian  $G_N$  for non-falling. For any location  $(x_0, y_0)$ , the Gaussian is within a window of  $x \in [0, 40]$  and  $y \in [0, 40]$  in pixels. We use  $(\sigma_x, \sigma_y) = (25, 15)$  for  $G_N$ , and  $(\sigma_x, \sigma_y) = (7, 15)$



**Fig. 3: Detecting falling down event. (a) Top row: selected frames from the original video, bottom row: Candidate blobs (blue) for the corresponding frames; in the last frame, the red blob indicates falling down event. Circles and inside arrows indicate the motion vectors of the blobs. (b) The probabilities for Falling (above white line) and Normal motion (below white line) along time (the x-axis). The red rectangle indicates falling down event.**

for  $G_F$ . The use of HMM helps to reduce the effects of noise in the data and results in detection of a *Fall* as evidence is accumulated. This process is in real time.

#### 4. EXPERIMENTAL RESULTS

We have tested our method on video data obtained using a vehicle borne camera. The video samples were captured using different scenarios that might occur during a routine police inspection of a suspect's vehicle. The scenarios were drafted using the help from a veteran police consultant. The total duration of many video clips is more than 40 minutes at a 30fps capture rate. We implemented our method using the C++ language and the OpenCV library due to its efficient use of the Intel CPU.

Fig. 2 shows an example of Running Person detection, whose position is indicated by a vertical red arrow. In our tests, the successful detection rate for person running out of car was around 80%. Fig. 3 shows an example sequence of Fall detection. In Fig. 3(a), a person in the foreground, detected by motion segmentation, is depicted by a blue blobs, and in the last frame of the sequence, the falling down event is indicated by the red blob. The circles and needles indicate the motion vectors of detected blobs. Fig. 3(b) shows the probability values computed over time (horizontal axis) for the Falling down event and the normal motion events.

Table 1 shows the summary of the results of our proposed method in the test video clips. In order to calculate the success rate of the method a human observer inspected the video sequence and counted the correct and incorrect *Fall* detection results.

**Table 1: Fall detection results confusion matrix.**

|              |             | SoftWare Performance |             |
|--------------|-------------|----------------------|-------------|
|              |             | Falling              | Not-falling |
| Ground Truth | Falling     | 83%                  | 17%         |
|              | Not-falling | 43.75%               | 56.25%      |

Although the false-positive rate for fall detection is high, it is necessary for the police safety. In our design of the system, the officers are equipped with radio worn on their uniforms that they can use to inform the headquarters that it is a false alarm. A high true positive rate, however, is important, especially when the officer is incapacitated for the system to perform as intended.

#### 5. CONCLUSION

This paper proposes smart in-car video system that monitors activities of police officers in the field and raises alarms in the event of a detection of a critical event. The three critical events that we address in this paper are Car coming to a stop, Suspect running out of the stopped vehicle, and Police officer down (falling).

Our method uses motion information described by a probability model which is more robust than conventional shape analysis. In the case of running person detection a reduced spatio-temporal representation of the video is created and movement features are extracted. A running person is depicted as a tracked 1D trace in the 1D projection profile starting at the car boundaries and extending towards the edges of the image.

In the case of a Falling police officer, foreground blobs were extracted using background subtraction, by calculating the centroids of the blobs and eliminating irrelevant blobs based on their points of entry in the frame. We use HMM to identify a falling down action based on movement angle and magnitude. The use of HMM follows a probability framework is more robust than

using ad-hoc thresholds. Experimental results show that our method has a success rate of 83% based on our data. The processing is real-time and is feasible to embed such a system into existing hardware inside a police vehicle.

#### 6. ACKNOWLEDGEMENTS

This work was supported by the US National Institute of Justice grant #2007-DE-BX-K007.

#### 7. REFERENCES

- [1] P. Atrney, M. Kankanhalli, and R. Jain. Information assimilation framework for event detection in multimedia surveillance systems. *Multimedia Systems*, 12(3):239–253, Dec. 2006.
- [2] J. Canny. A Computational Approach to Edge Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679–698, Nov. 1986.
- [3] J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel. Optimization and Filtering for Human Motion Capture. *International Journal of Computer Vision*, 87(1):75–92, Mar. 2010.
- [4] A. Jazayeri, H. Cai, J. Y. Zheng, M. Tuceryan, and H. Blitzer. An intelligent video system for vehicle localization and tracking in police cars. In *SAC '09: Proceedings of the 2009 ACM symposium on Applied Computing*, pages 939–940, 2009.
- [5] P. KaewTraKulPong and R. Bowden. An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection. In *Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems, AVBS01*, 2001.
- [6] R. Li, T.-P. Tian, S. Sclaroff, and M.-H. Yang. 3D Human Motion Tracking with a Coordinated Mixture of Factor Analyzers. *International Journal of Computer Vision*, 87(1):170–190, Mar. 2010.
- [7] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989.
- [8] D. N. Rosenblatt, E. R. Cromartie, and J. F. and. The Impact of Video Evidence on Modern Policing: Research and Best Practices from the IACP Study on In-Car Cameras. The International Association of Chiefs of Police Report, 2004.
- [9] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. Procrustes Shape Analysis for Fall Detection. In *The Eighth International Workshop on Visual Surveillance - VS2008*, Marseille, France, 2008.
- [10] G. Welch and G. Bishop. An Introduction to the Kalman Filter. ACM SIGGRAPH 2001 Course Notes, Los Angeles, 2001.
- [11] A. Williams, D. Ganesan, and A. Hanson. Aging in place: fall detection and localization in a distributed smart camera network. In *ACM MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 892–901, Augsburg, Germany, 2007.
- [12] Y. Wu and T. Huang. Vision-Based Gesture Recognition: A Review. In *Lecture Notes in Computer Science*, volume 1739/1999, pages 103–115. Springer Berlin / Heidelberg, 1999.
- [13] W. Yang, T. Lan, and G. Mori. SFU at TRECVID 2009: Event Detection. 2009.