

Visual Surveillance of Vehicles for the Detection of Anomalies

Satyam Srivastava, *Student Member, IEEE* and Edward J. Delp, *Fellow, IEEE*

Abstract—The large volume of vehicles on the road has created new challenges for agencies responsible for traffic management, law enforcement, and public safety. Such agencies frequently utilize visual surveillance technology to assist monitoring of vehicles from a remote location. These surveillance systems typically require trained human operators. Consequently, they are prone to human errors due to fatigue or diverted attention caused by excess information. Thus a need exists for an automated system that can analyze the surveillance videos and extract important information. This information would be used to detect occurrence of “anomalous” events in which case the human operator would be alerted.

In this paper we propose a visual surveillance system designed to function in the above-mentioned manner. More precisely, the system observes vehicular traffic from a standoff range and extracts information about the vehicles. This information includes vehicle type, make, tire size, and its dynamic trajectory. Based on this information the system checks for anomalies in the appearance and/or motion of the vehicle. We describe image analysis methods for obtaining the vehicle information from two cameras placed in an orthogonal configuration and for classifying the vehicles using these observations. We present the results of applying these methods on traffic videos. Our proposed system can be deployed for traffic monitoring (on highways/intersections) or infrastructure protection (at check points).

I. INTRODUCTION

THE advances in digital imaging, availability of inexpensive sensors, and the growing need for preventive technology have made surveillance a powerful tool for the law enforcement and security personnel. In particular, video surveillance is ubiquitous in office buildings, roads and intersections, and public transport systems. Image processing techniques are used to enhance the storage and visualization in many video surveillance systems but most systems simply record the video data in a passive fashion for potential forensic applications. In some scenarios this video data could also be monitored by a human supervisor.

While human-operated surveillance systems have the advantage of utilizing the expertise and judgement of a human being such systems have significant issues related to human error. The problems associated with these surveillance systems have been researched and well-documented [1], [2]. Two issues for which image analysis could provide solutions are loss of

efficiency due to fatigue and monotony, and lack of attention due to distraction from less important information. The task of monitoring surveillance footage is monotonous primarily because most of the time there is no “significant” activity occurring. Similarly an operator trying to study every subject in a video sequence may be unnecessarily fatigued. Once the task is extended to multiple video feeds there is simply too much information for the operator. It is widely accepted that the ability to monitor multiple video feeds drops significantly beyond 3 – 4 feeds [3]. Thus a trespass may go undetected while the operator is randomly scanning a parking lot or an empty hallway.

Traditionally, research in computer vision and image analysis has been aimed at developing methods and systems that mimic human vision [4]. However, even with state-of-the-art techniques a completely autonomous surveillance system that can work in any deployment scenario has not been described [5], [6]. Dee and Velastin [5] discuss situations where image analysis based methods can exceed human performance. Therefore an “intelligent” surveillance system would not replace human operators but would function in a complementary manner to maximize the overall effectiveness of the system. In this paper we examine the problem of visual surveillance of vehicles for the purpose of detecting anomalies. The goal is to perform the analyses without obstructing the traffic flow and to involve a human operator for making only the highest level decisions (for example, stopping a suspicious vehicle for manual inspection).

Due to the increase in volume of vehicles on the road traffic monitoring has become both important and difficult. A reliable traffic monitoring system together with an adequate response process would not only make the roads safer but can also potentially disrupt criminal and/or terrorist activities. Again, most technology solutions for vehicular monitoring involve significant human supervision or are passive tools for forensic application. During the last ten years several efforts have been made to develop image analysis based vehicle surveillance systems.

Coifman *et al.* proposed a vision-based surveillance system with a goal of assisting traffic flow management [7]. Their system would detect and track the vehicles, and determine certain parameters deemed useful for the purpose of traffic management. As part of the Video Surveillance and Monitoring (VSAM) project [8], researchers at Carnegie Mellon University developed a system to detect, track, and classify objects as vehicles and humans. They also designed methods for detecting simple activities and interactions between human subjects. The European AVITRACK project was aimed at

Satyam Srivastava and Edward Delp are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, USA. E-mail: (ace@ecn.purdue.edu).

Manuscript received . . .

This work was partially funded by a grant from the US Department of Homeland Security and the US Naval Research Laboratory. Any opinions, findings, and conclusions or recommendations in this paper are those of the authors and do not necessarily reflect the views of the sponsors.

developing methods for monitoring activities of service vehicles and personnel on airport aprons [9]. This would enable safer and more efficient servicing of aircrafts. Li and Porikli described a method for estimating traffic conditions using Gaussian Mixture Hidden Markov Models [10]. The IBM Smart Surveillance System (S3) detects and tracks objects, and has the capability for vehicle license plate recognition and detection of user-defined events [11]. In an alternative deployment scenario, Gutchess *et al.* propose a surveillance system to monitor vehicles and persons in a parking lot [12]. Their system can detect and track objects, and study the interaction of humans and vehicles. The system also offers nighttime operation capability by utilizing visible and near-infrared imagery. Some other interesting surveillance systems and technologies have been reported; these will be referenced in later sections.

Our work is different from these previous (or ongoing) projects because we are mainly interested in detecting anomalous events in vehicular traffic that pose potential risk to public and infrastructure safety. Therefore we do not estimate parameters related to general traffic conditions. Furthermore we do not consider human-vehicle interaction with the exception of the vehicles' occupants. This is justified because a core requirement for our system is the ability to observe the traffic without affecting the flow. Since the observed vehicles are almost always in motion, the chances of human-vehicle interaction are reduced to a minimum. Unlike most other cases where one or more cameras are used to obtain similar views (eg. stereo) of the subjects, we use a novel two camera orthogonal configuration to obtain the front and side views of the vehicles simultaneously. This allows us to extract certain information in more efficient ways, and other information (such as tire size) which would otherwise not be possible. The capabilities of this system can be further extended by addition of other non-video sensors in a synergic manner.

In this paper we describe techniques which can be used in a system to obtain different types of information. However, the set of information produced and used for decision-making is flexible and determined by the application and resource constraints. In Section II, we describe the overall function of our proposed system and the types of information we extract using image analysis techniques. These analysis techniques are presented under two categories – the methods used for vehicle identification/characterization are discussed in Section III while those for behavioral analysis from motion are described in Section IV. We provide results from the testing of these methods on traffic videos in Section V. Conclusions and future extensions to the system are provided in Section VI.

II. OVERVIEW OF THE SYSTEM

We begin this section by stating the working hypotheses of our approach. For most situations there is an accepted range of responses which can be termed as “normal” and anything outside of this range would be anomalous. Furthermore, such anomalous behaviors can often be associated with serious consequences. For example, a vehicle repeatedly ignoring the lane markings may be indicative of drunk/impaired driving

or road rage. Secondly, there are many traits of anomalous behavior which can be ignored if they occur in isolation. However occurrence of multiple anomalies is more likely to signal a potentially risky situation. We select certain observations associated with an oncoming vehicle and determine normal and abnormal responses for them. These are later used to identify potential anomalies in the traffic.

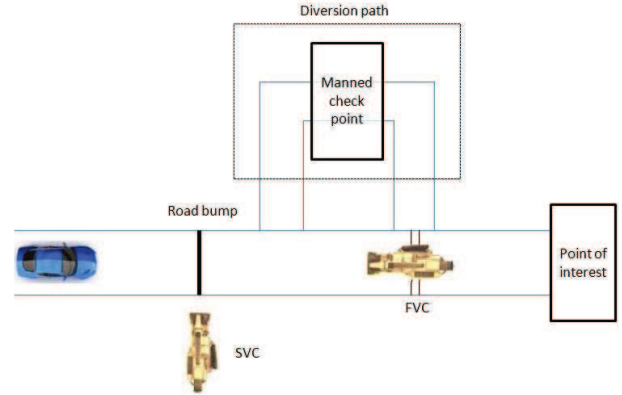


Fig. 1. An overview of our proposed surveillance system. The distances are not drawn to scale.

Consider a situation where video surveillance is used to monitor the vehicles approaching a point of interest (we also referred to as a “target” in this paper). Such targets would include airports, sporting venues, business districts, and other public establishments. In such places an accident or an organized criminal/terrorist activity would cause significant harm but the number of vehicles is too large to allow individual monitoring or a stop-and-go manual screening. Our proposed system can be used to automate the screening process without obstructing the flow of traffic. This deployment scenario is illustrated in Figure 1. We use two video cameras denoted by FVC (front view camera) and SVC (side view camera) in the figure. The video from these cameras would be available at a manned observation station along with the output generated by our video-analysis methods. For any vehicle, the system would produce one of three action labels – **Pass**, **Monitor**, and **Stop**. A vehicle designated for ‘Stop’ would be diverted for secondary screening while the ‘Pass’ vehicles will be allowed passage unobstructed. The treatment of ‘Monitor’ labeled vehicles would be application dependent, but such vehicles would typically require the judgement of a human operator (facilitated by the threat analysis provided by our system). The exact mechanism for vehicle diversion and secondary screening is not discussed in this paper.

The selection of two cameras and the particular configuration is described as follows. In our system, the purpose of multiple sensors is to generate complementary information (rather than redundant information). Thus for a cost-effective design, the number of cameras should be small. It can be argued that a two camera system provides considerably more information than a one camera system. However the incremental gain with a three camera system is small. Furthermore, the front and side views provide more important information than alternative configurations of a two camera system. For example, a rear-

view (instead of front-view) allows license plate detection but eliminates the scope for driver behavior monitoring.

Using the two cameras, we collect the information about the oncoming vehicles under two categories.

Physical Information: The data obtained under this category is used to characterize and/or identify a vehicle. This information includes vehicle body type, tire size, and make.

Behavioral Information: This set of information includes high-level analysis of the vehicle's appearance and motion. It includes analysis of the velocity and trajectory of the vehicle.

We can organize the various information collected and/or inferred from the video data into an approximate hierarchy. This is shown in Figure 2. This figure also illustrates the importance of using the two cameras in a synergic relationship. A typical sequence of operations when a vehicle approaches would be as follows. The name of the camera which is primarily used for making a particular detection is provided inside parentheses.

- 1) An oncoming vehicle is detected. (FVC)
- 2) The vehicle tracking and trajectory analysis systems are activated. An unexpected slowing/stopping may lead to the vehicle being marked for 'Monitor' while a sudden acceleration towards the target may result in a 'Stop' label. These decisions are based on the assumption that the driver may be searching for ways to evade the security systems or may be intending to crash through it resulting in patterns in the vehicle's approach. (FVC)
- 3) When the vehicle enters the SVC's field of view its body type is determined. (SVC)
- 4) The vehicle's tires are segmented and their size is estimated. (SVC)
- 5) The make of the vehicle is determined. (FVC)
- 6) The gap between the tires and the vehicle body (wheel well) is estimated and the front and rear values are compared. If the gap above the rear tire is significantly smaller, it can indicate a heavy load in the trunk which may lead to a 'Monitor' label. (SVC)

We next describe the various image and video analysis methods used to achieve the above tasks. The criteria for selecting these methods include functional effectiveness, low-complexity (suitability for real-time operation), and robustness to changes in operating conditions.

III. IMAGE AND VIDEO ANALYSIS FOR VEHICLE INFORMATION

In the previous section, we described the measurement/inference capabilities that our system should have in order to characterize/identify a vehicle. We now provide details of the techniques that can be used to achieve these capabilities.

A. Vehicle Detection

The initial steps in most intelligent surveillance systems involve detection of a subject of interest. Background subtraction is the most common approach to object detection and several elegant methods have been proposed [13]. A background subtraction operation consists of comparing the

current video frame with a background model with respect to some properties such as pixel intensity and color.

Let the current frame be represented by

$$C = \{c(i, j) : i = 0, 1, \dots, W - 1; j = 0, 1, \dots, H - 1\}. \quad (1)$$

Here, $c(i, j)$ represents the value of a chosen image property at the pixel located at (i, j) . The frame is considered to be of width W and height H . Similarly, we can represent the background model as an image $B = \{b(i, j)\}$. Then, the output of background subtraction is a spatial mask $F = \{f(i, j)\}$ defined as follows:

$$f(i, j) = \begin{cases} 1.0, & \text{if } c(i, j) \text{ is in the foreground,} \\ 0.0, & \text{if } c(i, j) \text{ is in the background.} \end{cases} \quad (2)$$

The decision of classifying a pixel as foreground is based on a metric that measures the difference and a decision threshold corresponding to the metric. Let the features used to specify each pixel belong to a space Ψ , and consider a function $\delta : \Psi^2 \rightarrow \mathbb{R}$ where \mathbb{R} is the real number line. The value of the function $\delta(c(i, j), b(i, j))$ quantifies the difference between the current frame pixel and the background model pixel at the location (i, j) . While the space Ψ can contain velocity vectors, acceleration vectors, intensity levels, edge orientation and trichromatic digital values (such as R, G, B), we consider a simple case of grayscale images with the pixel intensity specified by a real number between 0.0 and 1.0 where 1.0 represents the maximum gray level. We selected gray level because many background subtraction methods are developed for gray levels. While color may be useful in high quality and well-lit situations, typical vehicle surveillance systems would not benefit greatly. For gray level features the corresponding difference function would be $\delta(x, y) = |x - y|$. We can redefine the mask F as:

$$f(i, j) = \begin{cases} 0.0, & \text{if } |c(i, j) - b(i, j)| \leq \tau, \\ 1.0, & \text{otherwise.} \end{cases} \quad (3)$$

Note that $0.0 \leq \delta(x, y) \leq 1.0$. Therefore, the decision threshold τ would be a real number in $[0.0, 1.0]$.

In the basic background subtraction method, the threshold τ and the model B may be empirically chosen and remain fixed throughout the operation. Most of the improvements in the basic method affect the selection and updating of these two parameters. Some methods allow gradual modification of the background model after each frame based on the latest classification. This gives the system some flexibility to perform correct classification even when the background is not truly static. These tasks may be performed statistically [14], [15], [16] or adaptively [17], [18]. One of the key considerations for a real-time surveillance system is the complexity of the method. We next describe our method for background subtraction which does not require a model update every frame and still accounts for background perturbations.

Most commonly used background subtraction methods update the background model using the results from the current frame classification. Let C_n and B_n represent the current frame and the background model at time n . Similarly, let F_n be the foreground mask corresponding to C_n . Then, the new

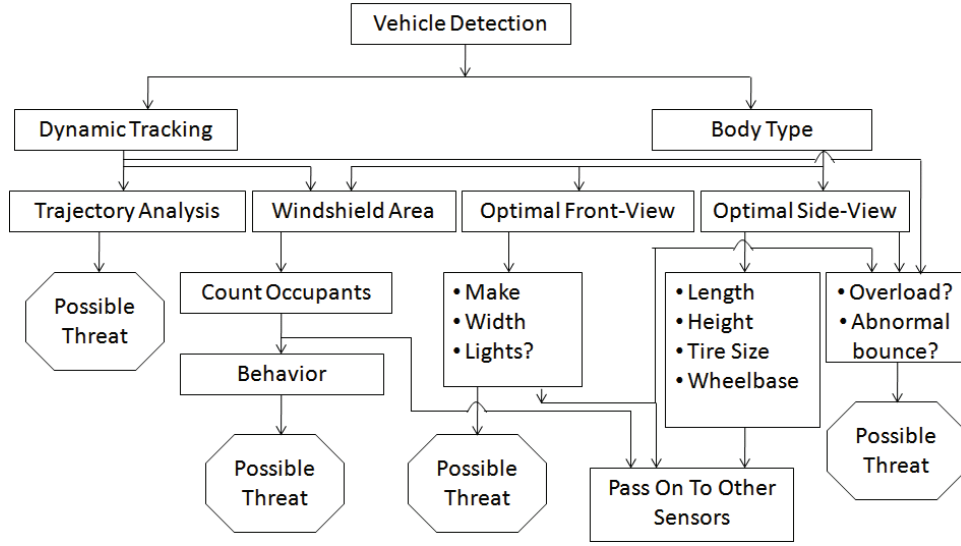


Fig. 2. Various information collected by our proposed system and the inter-dependencies of the decision processes.

background model can be constructed as:

$$B_{n+1} = \alpha \times C_n + (1 - \alpha) \times B_n, \quad (4)$$

where α represents the *learning rate* of the method and $0.0 \leq \alpha \leq 1.0$. This approach has the limitation that if a foreground object stays in the scene for an extended amount of time, it may alter the background model unnecessarily. Another issue is that for certain types of repetitive motion (tree leaves or shaky camera) the model might never be able to “catch up” and always result in incorrect classification. This might occur even in methods that allow multiple modes for the background model (such as GMM [14]). Moreover, updating the model for every frame is computationally expensive.

We observe that whenever the background model is changed due to small (and temporary) displacement of a background object we can account for the change simply by comparing the current pixel with the correct background pixel. However, since the correct position of such a pixel is not known, we compare the current pixel with a small neighborhood of pixels in the background model. Let (i, j) be the pixel location being investigated. Then we construct a neighborhood of $c(i, j)$ as:

$$N(i, j) = \{b(p, q) \in B : i - \epsilon \leq p \leq i + \epsilon, j - \epsilon \leq q \leq j + \epsilon\}, \quad (5)$$

where ϵ is the search window size. Next, the foreground mask

as generated by our method is given by:

$$f(i, j) = \begin{cases} 0.0, & \text{if } \exists b \in N(i, j) \text{ such that } |c(i, j) - b| \leq \tau_1 \\ 0.5, & \text{if } \tau_1 < |c(i, j) - b| \forall b \in N(i, j) \text{ and} \\ & \exists b \in N(i, j) \text{ such that } |c(i, j) - b| \leq \tau_2 \\ 1.0, & \text{otherwise.} \end{cases} \quad (6)$$

The method is known as motion-assisted background subtraction (MABS).

In this formulation an extra state of low confidence is added, represented by $f(i, j) = 0.5$ and computed using a second threshold $\tau_2 > \tau_1$. For the purpose of classification, every pixel $c(i, j)$ is considered a background pixel if $f(i, j) \in \{0.0, 0.5\}$. This extra state of low confidence classification is used in background model update which occurs only when a global illumination change is detected. The new threshold τ_2 is chosen as a fixed fraction of the dynamic range of the frame.

A global illumination change is inferred when the average gray level of the current frame differs from the background model by a pre-defined amount τ_i . This constant is empirically chosen by considering the effects of multiple foreground objects, the global gray levels, and the tolerance provided by the selection of the threshold used in Equation 6. In the event of a global illumination change, the background model is updated as follows:

$$b_{n+1}(i, j) = \begin{cases} c_n(i, j), & \text{if } f_n(i, j) = 0.0 \\ (1 - \alpha) \times b_n(i, j) + \\ \alpha \times c_n(i, j), & \text{if } f_n(i, j) = 0.5 \\ b_n(i, j), & \text{if } f_n(i, j) = 1.0. \end{cases} \quad (7)$$

Thus, we select the learning rates for model update on the basis of confidence in the current classification.

B. Vehicle Body-Type Determination

The knowledge of an oncoming vehicle’s body type is useful in several ways. It narrows the expected range corresponding various others measurements. For example, we describe the tire size estimation in the next section. This task would be much harder if the estimation method is ignorant of the body type. This information also helps in determining the region of interest for make recognition. Therefore, the primary role of this information is to aid the functioning of other modules. We consider vehicles belonging to four classes – sedan, light truck, sport/utility vehicle, and hatchback.

Determination of a vehicle’s type is a part of many traffic monitoring systems where it may be used to generate the temporal composition of traffic on a busy route. Some traffic monitoring systems are based on inductive loops embedded in the roadway and other non-video methods [19]. Some video-based methods have also been proposed for this problem. In [20], a statistical model-based approach is used to detect and classify vehicles using a two stage process. Avery [21] presents a simple method to distinguish cars from trucks based solely on the vehicle length. Other techniques extract shape and motion information to identify the type [22], [23]. Lai [24] describes a traffic analysis system using a video-based virtual loop technique which also classifies vehicles.

We now describe a shape matching technique to determine the most likely class of a vehicle using an image taken by the side view camera (SVC). This approach is suitable because it can use the output of the background subtraction method with only a small amount of processing. Recall that the output of the vehicle detection routines is a binary image where the pixels are marked as a part of the vehicle or a part of the background. Note that the third state of low confidence described in Section III-A is treated the same as the background for classification. Thus, the foreground mask F produced by the SVC is a silhouette of the vehicle. It may be interesting to point out that the FVC assists this analysis by providing the position of the vehicle. In other words, a particular SVC frame is selected for this analysis when the vehicle is known to be in the SVC field of view.

Unlike [22], we do not extract features from the silhouette. Instead, we directly perform silhouette-matching against the templates corresponding to each vehicle class. Let $S = \{s(i, j)\}$ be the silhouette of an oncoming vehicle as obtained by inverting the foreground mask F . Also let us represent the silhouettes of the template vehicles as T_1, T_2, T_3, T_4 where the class labels correspond to sedan, truck, SUV, and hatchback (respectively). These templates are obtained by thresholding manually selected images in which the vehicles differ considerably from the background. An example of such a vehicle and the resulting template is shown in Figure 3. After thresholding, the binary image is cropped so that the vehicle silhouette is immediately surrounded by the image boundaries. At this instant, the aspect ratio of the vehicle and the size of the cropped image are not altered in any way. Thus the templates may mutually differ in the image dimensions.

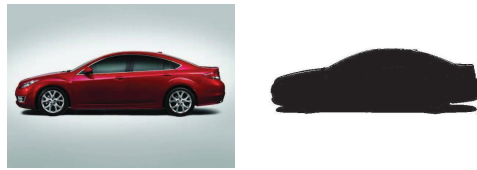


Fig. 3. An example of the template extracted from a vehicle belonging to the “sedan” class.

While the templates are obtained from images with favorable backgrounds, the test vehicle may not be viewed under such conditions. Therefore the silhouette S may contain regions where the background subtraction method made erroneous classification. This would be a problem if the silhouette image was used to extract, for instance, edge features since the erroneous regions would give rise to false edges. However, we use a direct difference based approach and the erroneous regions only appear as an additive constant in each vehicle class. The differencing operation is described next:

- 1) As in the case of the templates, the test vehicle silhouette is obtained by cropping the inverted foreground mask to the smallest size that still contains the whole vehicle. Let the silhouette S have dimensions $w \times h$. It should be noted that cropping eliminates the issues related to registration.
- 2) Each class template is scaled to a length of the test vehicle silhouette w such that its original aspect ratio is preserved. These scaled templates \tilde{T}_k are used to compute the sum of absolute difference (SAD) $\Delta_k = \sum |s(i, j) - \tilde{t}_k(i, j)|$ where the summation is taken over all the pixels.
- 3) The class corresponding to the smallest SAD value would be the most likely type of the test vehicle.

It can be argued that scaling the templates destroys the length information about the class representative vehicles. We explain scaling as follows. In the absence of scaling a simple difference cannot be obtained. Thus, one would need to perform feature matching (such as linear discriminant analysis) which would be affected by the patches in the silhouettes. Secondly, the templates themselves are obtained from images taken at different scales implying that scaling does not cause significant loss of information. Finally, we propose to use length as an independent dimension for separating the vehicle classes which are otherwise not well resolved with shape alone. We also take a note of this in Section V.

The primary purpose of obtaining the vehicle’s silhouette was to compute the SAD against the templates but in the process, we have also determined the vehicle’s height and length in image co-ordinates. More specifically, we note that the values w and h in the cropped templates are exactly the length and height of the vehicle in pixels. By mapping these to ground co-ordinates, the true dimensions of the vehicle can be estimated. There are two popular methods to perform this mapping – camera calibration and use of fiducial markers.

Camera calibration techniques use spatial transformations along with some analytical models to develop the pixel-inches mapping. These can also be realized using simpler approximate functions or even look-up tables. However, in

the circumstances when such calibration is not possible the systems utilize the presence of objects of known dimensions in the scene for the purpose. These objects are known as fiducial marks and are typically simple geometric shapes such as lines and points. In cases when such marks are not intrinsically present in the scene, they may be explicitly added. We discuss more about the image co-ordinate to world co-ordinate mapping in Section IV.

C. Tire Size Estimation

Visual inspection of a vehicle's tires sometimes provides useful information such as over/under inflation or the presence of oversized tires for better towing and off-road capability. Tire size can be used to determine if the tires are overinflated which indicate that a heavy payload is being concealed in the vehicle. However, much of this information is not directly of value to traffic monitoring and analysis systems. Thus most traffic surveillance systems lack the capability to accurately segment the tires and estimate their size. Although many of the advanced object recognition systems may be trained to detect tires, they may not be suitable for a real-time and real life application because of the high complexity and because of the variability in the tire profiles. Some examples of tires with differing hubs are shown in Figure 4.

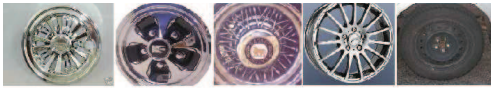


Fig. 4. Examples of tire hubs including bare wheels, plastic hubcaps and alloy wheels.

Accurate location of an oncoming vehicle's tires is important to us for two main reasons. This information along with the make (described in Section III-D) and the exterior dimensions can help identify the vehicle's model. Secondly, a direct analysis of the tire flatness and bounce (described in Section IV-D) may be used to detect overloading. The tire size estimate can also be used to determine other tire parameters, such as remote tire pressure measurement [25]. We now describe a method to correctly locate the tires and determine their size. This analysis is performed using frames from the SVC's output. Note again that the frame is selected using the vehicle position determined by the FVC.

There are two major challenges in accurate segmentation of a moving vehicle's tires – the lack of information about both the position and size of the tire, and the different shapes of tire hubs. We note that despite the variation in the shapes of the hub, most tires have one common feature which is the prominent circular edge between the wheel and the rubber. Since most vehicles on the road today have either hub caps or alloy wheels this observation holds true for them. The notable exceptions include vehicles with older wheels and missing hubcaps. Since a single approach is not universally applicable in all tires, we describe the two cases separately.

Tires with bright hub: In the cases when a significant level of contrast exists between the tire's wheel and rubber it is possible to detect the wheel-rubber edge. This situation is illustrated in Figure 5. Our goal is to accurately locate the circular edge because it provides us the knowledge of the tire's center and the radius of the (visible part of the) wheel. For the present discussion consider only the front tire of the vehicle as seen during a particular video frame V_{SVC} . Let (x_t, y_t) be the true location of the front tire center in the frame and the true tire radius be r_t pixels. Further, let r_w be the true radius of the wheel as visible from a side view.

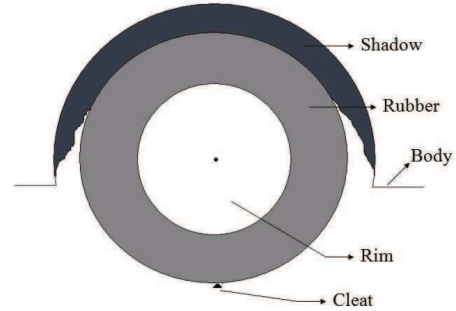


Fig. 5. Graphical illustration of the system's view of a tire with bright hub.

We model the wheel-rubber edge as concentric white and black circles on a neutral background as illustrated in Figure 6. The thickness of the circles t is kept constant (usually 2 pixels) whereas the radius can be changed during the estimation process. Let r be the (outer) radius of the white circle. Then, at the end of the processing, r would be the method's estimate of r_w . The estimation is performed by placing the edge model of a chosen radius at different positions in the frame.



Fig. 6. In the case of a tire with a bright hub, the wheel-rubber edge is modeled as concentric circles of chosen radii.

Let the tire model be represented by $T_{r,p,q}$ where r is the radius of the wheel and (p, q) is the hypothesized position of the wheel center. In order to superpose the edge model on the frame, we can define the model for the same width and height

as the frame:

$$T_{r,p,q}(i,j) = \begin{cases} 0.0, & \text{if } r - t < \|(p,q) - (i,j)\| \leq r \\ 1.0, & \text{if } r < \|(p,q) - (i,j)\| \leq r + t \\ 0.5, & \text{otherwise.} \end{cases} \quad (8)$$

Then, the error associated with this set of values for (r,p,q) is computed by

$$\tilde{\Delta}_{r,p,q} = \Sigma |V_{SVC} - T_{r,p,q}|, \quad (9)$$

where the summation is carried out over all pixels in the frame for which the edge model is not neutral. That is, the difference is computed only if $t_{r,p,q}(i,j) \neq 0.5$. Finally, the best estimate of wheel parameters is given by,

$$(r,p,q)_{optimal} = \arg \min \{ \tilde{\delta}_{r,p,q} \}. \quad (10)$$

It should be noted that the ranges of (r,p,q) to be searched over can be intelligently limited by using information about the vehicle's body type and the vehicle's position with respect to the speed bump shown in Figure 1 (and labeled as a cleat in Figure 5).

Once the position of the tire center is determined, the tire radius can be determined by dropping straight lines away from the center as illustrated in Figure 7. Multiple estimates of the radius are made by measuring the length of each such line between the center and the rubber-background edge. We drop lines only in the lower half of the tire and consider only the lines which are longer than r_w . Then, the best estimate of the tire radius is the shortest such length.

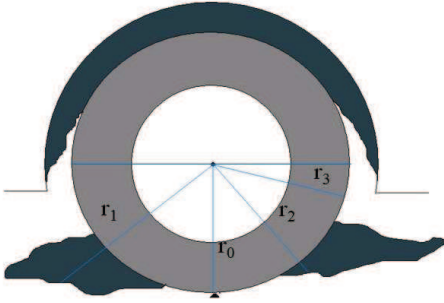


Fig. 7. Multiple lines are dropped from the known tire center towards the rubber-background edge to estimate the tire radius.

Tires with dark hub: If there is insufficient contrast between the tire rubber and the wheel (as in Figure 8), the strategy to look for a distinct circular edge fails. In such cases, additional information may be needed by the system to estimate the size of the tire. Therefore, we assume that the horizontal position of the tire's center can be determined by non-video methods. One way to accomplish this is to place pressure or switching sensors near (or inside) the speed bump to determine the exact instant when the tire crosses the bump (marked as a cleat in the figure). Since the position of the

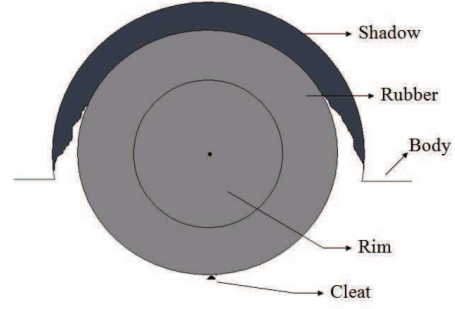


Fig. 8. Graphical illustration of the system's view of a tire with dark hub.

speed bump in the frame is known, the position of the lower-most point of the tire is determined.

Assuming that the tires are roughly circular in shape, we model the tire T_r as a black disk on a neutral background such that the disk has a radius r and its lower-most point coincides with the speed bump. As in the case of tires with bright hubs, we compute the error associated with a particular choice of r using,

$$\tilde{\Delta}_r = \Sigma |V_{SVC} - T_r| \quad (11)$$

where the summation is performed over all pixels for which the tire model is not neutral. It can be argued that if the radius of the disk is taken small enough, the error will vanish but this value could be much smaller than the actual tire radius. Thus, we define our estimate of the tire radius as:

$$r_{optimal} = \max \{ r : \tilde{\Delta}_r < \epsilon \} \quad (12)$$

In other words, the largest disk which can be superposed on the frame and still produce no more than ϵ error will be the best estimate of the tire. Note that, unlike in the previous method we cannot determine the wheel radius separately.

D. Make Recognition

We can broadly classify the various make and model recognition (MMR) techniques into two categories – the ones based on feature matching and the ones based on appearance matching. In [26], Petrovic and Cootes describe the use of gradient features for vehicle identification. Clady proposes a two stage method based on oriented contours which uses multiple frontal images and is claimed to be robust to partial occlusion [27]. Methods based on scale invariant feature transform (SIFT) proposed by Lowe [28] have also been applied to the MMR problem [29], [30]. Appearance based methods use pixel intensities and provide more global features. Zafar discusses the use of appearance matching using principle component analysis (PCA) and two-dimensional linear discriminant analysis (2D-LDA) [31]. This approach is inspired by the application of these techniques in face recognition systems.

The above-mentioned methods are promising solutions for the MMR problem. They can be used in our system with the following modification. The selection of a region-of-interest

(ROI) is critical to the success of any recognition technique. Many of the techniques above use existing license plate recognition tools to locate the license plate. This provides an adequate reference to demarcate the ROI in the image. While all vehicles have a rear license plate, in our sensor deployment scheme (Figure 1) the rear plate and the rear view of the vehicle cannot be used. Since many areas in the United States do not require a front license plate there is practically no reference for a make recognition method to select the ROI. We select a large ROI containing the entire front grille using the vehicle body type information.

We construct a make recognition system which uses the front view of a vehicle (and the body type information provided by the SVC) with no other spatial reference. We use feature matching based on histograms of edge orientations from the front grille. Let the video frame in which make recognition is performed be represented by V_{FVC} and the foreground mask generated by the vehicle detection system be F for this frame. Recall that $f(i, j) = 1$ implies that the pixel located at (i, j) belongs to the foreground object (the vehicle). Assuming that the object detection can reliably classify the pixels, we can define a bounding box around the vehicle. Let (i_1, j_1) and (i_2, j_2) be the defining vertices of a rectangular box ($i_1 < i_2$ and $j_1 < j_2$) such that,

$$f(i, j) = 1 \implies i_1 \leq i \leq i_2 \text{ and } j_1 \leq j \leq j_2$$

Then, the geometric center of the vehicle in the image is given by $(c_i, c_j) = (\frac{i_1+i_2}{2}, \frac{j_1+j_2}{2})$. The region of interest is specified as a rectangular region whose defining vertices are given by (i_1, c_j) and $(i_2, c_j + \epsilon)$. Note that this definition covers the entire width of the vehicle's front and considers the area up to ϵ pixels below the geometric center of the vehicle. Different values of ϵ can be used with a larger value selected when the vehicle is a truck or an SUV than when it is a sedan or a hatchback. Thus, we define the ROI as the pixels in V_{FVC} such that $i_1 \leq i \leq i_2$ and $c_j \leq j \leq c_j + \epsilon$. This method tends to over-select the region where desired features are located but we prefer it to under-selection where the key features may lie outside the ROI.

Having decided on the ROI, we construct a histogram of edge features using the Sobel operator [32]. More specifically, we evaluate the gradient at each pixel inside the ROI in terms of magnitude and orientation. Consider a pixel in the video frame at a location (p, q) inside the ROI. We represent the Sobel operator as a 2D array or a spatial filter,

$$M = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (13)$$

Then, the operation of pointwise multiplication and summation can be seen as a 2D convolution after shifting the filter to the desired location. Let $C_x = \{c_x(i, j)\}$ such that $c_x(i, j) = V_{FVC} * M$ be the two dimensional array containing the filter output when M is shifted so that its center is at the position (i, j) . Note that C_x contains the edge information in the horizontal direction. Similarly, we can compute the edge information in the vertical direction $c_y(i, j) = V_{FVC} * M^T$.

The magnitude of the edge at any point (i, j) is given by

$$\sigma = \sqrt{c_x^2(i, j) + c_y^2(i, j)}, \quad (14)$$

and the orientation is given by

$$\theta = \arctan c_y(i, j)/c_x(i, j), \quad (15)$$

where $-\pi/2 \leq \theta \leq \pi/2$. We obtain the edge orientation at each point (p, q) inside the ROI and construct an N -point histogram of these values. Let h_α^N be the N -point histogram of edge orientations for an arbitrary vehicle α . We select a front view image for vehicles from each car maker and generate the histograms which are used as the ‘signature’ of the particular make, $h_1^N, h_2^N, \dots, h_k^N$. We further process the histograms by normalizing them with respect to the size of the ROI in each of the make. Thus, $\tilde{h}_1 = \frac{1}{w \times t} h_1^N$ where w and t are the width and height of the ROI in the image of car maker 1. The other histograms are similarly normalized. We then determine the histogram for an oncoming vehicle h_{test}^N and normalize it with respect to its own ROI to obtain \tilde{h}_{test} . Finally, we estimate the correlation coefficient as the measure of similarity between the test vehicle and the r^{th} make template. This is defined as

$$\rho_r = \frac{\tilde{h}_r^T \tilde{h}_{test}}{\sqrt{\tilde{h}_r^T \tilde{h}_r} \sqrt{\tilde{h}_{test}^T \tilde{h}_{test}}} \quad (16)$$

Then, the most likely make is given by,

$$\alpha = \arg \max_r (\rho_r), \quad (17)$$

where α is the index number associated with each car maker.

IV. IMAGE AND VIDEO ANALYSIS FOR ANOMALY DETECTION

The methods described in the previous section enable us to determine the ‘normal’ ranges of behavior for vehicles of the observed type. We now describe methods for characterizing the vehicle motion and detecting abnormal events. The range of behavioral analysis for vehicles is limited when compared with human subjects [33] due to rigid shapes and regularity of motion. Still, timely detection of anomalies can potentially prevent mishaps and damage to life/property. Traditionally such anomaly detection is achieved using trajectory clustering and identifying deviations from the clusters [34], [35], [36]. Although the trajectory given by a tracker includes both position and velocity estimates, many approaches end up discarding the velocity changes which can be highly informative. Furthermore, path learning via clustering is affected by issues such as unequal trajectory lengths, choice of path models, and effects of distance metric.

We approach the task by hypothesizing that the velocity and shape of the trajectory can be analyzed separately and that in most situations patterns in the velocity provide enough information to detect anomalies. Basharat *et al.* describe modeling of motion patterns which does not require clustering [37]. We illustrate this with an example of vehicles approaching a check point. Medioni *et al.* use a clustering-based method to detect if a vehicle tries to evade the check point [35]. However, their approach does not detect if the vehicle slows down or stops

and then proceeds toward the check point or picks up speed as it approaches indicating an intention of crashing through it. In contrast, a simple velocity differential [38] can detect these patterns.

We propose to form path models of velocities as functions of position rather than time. We also estimate the velocity models for finite sub-regions of the field of view. This is different from identifying sub-paths after clustering. In each such division, we represent the velocity with a Gaussian distribution. Note that the use of finite spatial divisions prevents the issue of unequal trajectory lengths. This allows an anomaly detection system to utilize the patterns in the velocity and also allows better modeling of the decision function than the simple thresholding used in [38]. We further observe that such velocity clusters would intrinsically have high variance because people drive at different speeds, often near the speed limits. Thus, we first scale the velocities with the average speed estimated when the vehicle is relatively far from the observation point. These methods are described in Section IV-B.

Next we observe that detecting deviations from “normal” velocity would be easier if there is only one set of normal driving patterns. This is not true in general. For instance, a vehicle would slow down from its original speeds when making a turn or taking a highway exit. Thus, if a vehicle’s velocity matches a cluster corresponding to vehicles that made a turn, this vehicle should also be making a turn. We use template matching to identify significant maneuvers from the trajectory shape in Section IV-C. By separating the temporal and spatial analyses of the trajectory, we obtain better flexibility to define the rules of anomalous behavior.

The above analyses assume that vehicles are point objects moving in a 2D plane. In Section IV-A we define a co-ordinate transformation in which the motion of the vehicle consists of two components – motion along the direction of the road and motion perpendicular to it. This transformation allows us to represent the velocity with a 1D descriptor (by ignoring the perpendicular component). It also improves the shape analysis by discarding the curvature caused by naturally curved roads.

We describe the above analyses with a goal of achieving real-time operation. More specifically, we would like to process the state of the vehicle (as given by a tracker) as soon as it is received. Our methods are compared with other techniques which do not offer such operation capability. The results of testing our methods are provided in Section V. The methods described in this section are extensions of methods presented in [39].

A. Trajectory Estimation and Co-Ordinate Mapping

The tasks of robust object detection and tracking are central to many image analysis problems. Over the years many interesting methods have been proposed for detect moving objects and tracking them. Haritaoglu *et al.* [40] and Stauffer and Grimson [14] describe popular methods for object detection and tracking. A comprehensive survey of tracking is provided by Yilmaz [41]. Our system does object detection and tracking in a continuous manner. Foreground objects are detected by background subtraction using the method described in

Section III-A. The output of the detection (foreground blobs) is characterized by connected components analysis and is used to create targets which are tracked using a particle filter framework as described in [42]. Particle filter is a Bayesian framework which is used to estimate the “state” of a target being tracked. In our system, the object’s state includes its position and size. The filter generates several hypotheses (particles) of the new state and computes the weighted average of these hypotheses as the estimate of the new state. The weights are computed by determining the similarity of the hypotheses with respect to some chosen object features. We use color and edge orientation as the objects features. Arulampalam *et al.* provide a detailed tutorial on the application of particle filters in object tracking [43].

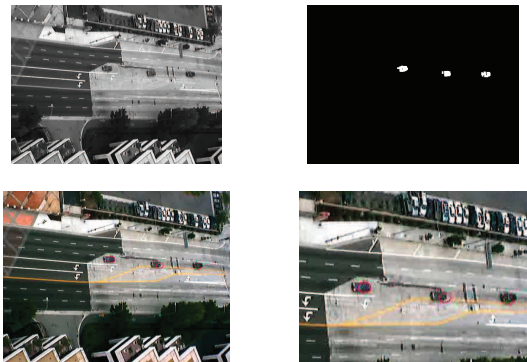


Fig. 9. An example of results produced by our object detection and tracking systems – from left to right: original frame and the foreground mask (top), three tracked vehicles and a close-up of the scene (bottom).

Figure 9 shows a snapshot from one of our test sequences. The corresponding outputs from background subtraction and object tracking are also provided. For the rest of this paper, we assume that the vehicle’s position in the video frame is available. More precisely, a vehicle’s “image trajectory” is defined as its position in the frame as a function of time (or frame number), and is provided by the tracker. Let this image trajectory be represented by

$$\Phi_c^i = \{(x_t^i, y_t^i) : 1 \leq t \leq T_c\}, \quad (18)$$

where the subscript c denotes a particular object. For the sake of brevity, this subscript is not used in further discussion. Similarly, T_c is the lifetime of the object in number of frames, t is the time index, and the superscript i denotes image co-ordinates. We follow the convention that x increases from left to right and y from top to bottom.

For any object, one would like to transform the trajectory into actual ground co-ordinates in order to remove the distortion due to camera perspective. Let this transformed trajectory be represent by Φ^g with definition similar to Φ^i . In our application scenario where an object can be modeled as a point, an ideal transformation would result in an aerial view of the trajectory. This can be achieved with a planar homography [44] or an approximation of the same. Our preferred method of perspective correction uses look-up tables because it is fast and requires minimal calibration. However, from our experiments we have observed that ground co-ordinates are often

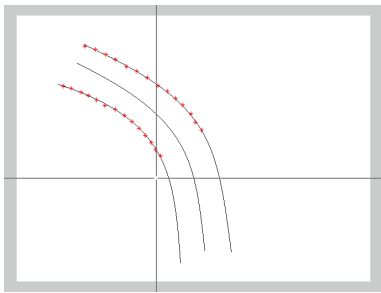


Fig. 10. An illustration of user specified co-ordinate transformation. The points marked with red crosses divide the “road” into equal distance segments.

not the most efficient representation for trajectory analysis. For example, vehicles moving along a naturally curved road traverse a curved trajectory in ground co-ordinates whereas their motion is a “straight line” with respect to the road.

We propose analyzing vehicle trajectories in a hypothetical co-ordinate system in which the distances are arbitrarily close to the distance in ground co-ordinates but the axes are defined with respect to the shape of the roads. Thus, a vehicle moving parallel to the center median is considered as moving along a straight line. This transformation results in two immediate benefits – (1) the vehicle’s velocity can be specified with a single component (in the direction of the road) and (2) false curve/turn detection along gently curved roads can be suppressed. Again, we utilize 2D look-up tables to achieve this transformation from ground to the hypothetical co-ordinate system. This table is constructed interactively by user input. This is explained next with an example.

Consider the situation in Figure 10 where the parallel curves represent the boundaries of a curved road. This image is assumed to be perspective corrected. We next assume that this corresponds to a road in the hypothetical space whose length and width matches the actual section of the road but which is not curved. We establish an approximate point correspondence by asking a user to divide the two curves into equal intervals (equal in the ground co-ordinates). Depending on the number of divisions, corresponding points are located on the hypothetical road. This completes the look-up table where the input is a position in the ground co-ordinates and the output is the corresponding position in the hypothetical system. For points not aligned with the input grid, 2D Shepard interpolation [45] is used. This look-up table is later used to transform the vehicle’s trajectory into the desired co-ordinate system.

Note that a single look-up table can be created to map the object position from the image co-ordinates directly to the hypothetical co-ordinates. That is, no explicit perspective correction step is required. This is done by taking the user input (or an equivalent method) to divide the road boundaries in the image co-ordinates. This is our recommended approach and has been used in our experiments. Let the final trajectory be represented as

$$\Phi^h = \{(x_t^h, y_t^h) : 1 \leq t \leq T\}, \quad (19)$$

where the position x^h is along the road and y^h is perpendicular

to it.

B. Velocity Analysis

We first introduce our proposed representation of the velocity as a function of the vehicle’s position. Note first that we only consider the axial component of the velocity (parallel to the road) in this section. Let $x(t)$ and $v(t)$ represent the (axial) position and velocity estimates of a vehicle at time t . At this point we make the following observations:

- Even in the absence of obstacles there is natural variation in “normal” driving speeds due to curves, turns, and inclinations/declinations. However there will not be many variations when looking at a small portion of the roadway. Therefore, patterns of normal velocities should be estimated for different portions of the road.
- A fast moving vehicle will have fewer samples in its trajectory while a slow moving vehicle will have more samples. Thus, the k^{th} sample of these vehicles’ trajectories would not correspond to the same portion of the road.
- Resampling of trajectories to get the same number of samples may solve this problem of mis-registration. For instance, if a vehicle A drives twice as fast as another vehicle B we can upsample A’s trajectory by a factor of 2. If the sampling time (inverse of the frame rate) is τ then the k^{th} sample in A’s resampled trajectory corresponds to time $k\tau/2$ (since A’s entry) while in B’s trajectory it represents time $k\tau$. However, their spatial position would be very similar due to the inverse effect of velocities.

The third observation motivated us to represent the velocities as function of position $u(x)$. This representation removes the need for resampling because the velocities of two vehicles at same position *are* comparable. To allow easier indexing we quantize the position x by dividing the road into M smaller segments. The velocity can be specified using the index of the segment in which x lies. Let the length of the road in the field of view be L (feet/meter) where we follow the convention that x increases from 0 to L as the vehicle gets closer to the observation point. We represent the segments by a set of points $\{s_0, s_1, \dots, s_M\}$ such that $s_0 = 0$, $s_M = L$, and

$$s_i - s_{i-1} = \frac{L}{M}, \quad \forall i \in \{1, 2, \dots, M\}. \quad (20)$$

With the length of the segments $\Delta = \frac{L}{M}$, the velocity $u(x)$ can be represented using $\bar{u}(i)$ where

$$i = \lfloor x/\Delta \rfloor \quad (21)$$

indicates the i^{th} segment and $\lfloor \cdot \rfloor$ denotes integer floor operation. The length Δ (or equivalently, M) would be chosen small (large) enough to prevent velocity behavior changes within a segment while not over-segmenting because that would discard spatial averaging.

In the final pre-processing stage, we propose to scale the velocities with respect to each vehicle’s average speed. This allows us to emphasize the *changes* in the velocity rather than the absolute speeds. Further, it also leads to smaller variances in the normal behavior distributions which we describe next.

Note that our goal is to achieve real-time operation. Thus, the average speed must be estimated with only part of the observations. We estimate the average speed u_{avg} over the velocities observed when $x < L/2$. This was motivated by the assumption that a visible change in behavior is more likely to occur when a vehicle is near the observation point (e.g. a check point). Hence it will exhibit more “typical” behavior when in the far field of view. The final form of the velocity is given by

$$c(i) = \bar{u}(i)/u_{avg}. \quad (22)$$

The normal velocity behavior for each segment i is modeled with a mixture of Gaussian distributions specified by the mean $\mu_{i,k}$ and variance $\sigma_{i,k}^2$ of vehicle velocities for the k^{th} Gaussian component. One can use clustering to obtain these parameters. We use a simpler context-based approach in which the number of Gaussian components is related to the number of maneuvers observed in the segment. For instance, a “straight-only” lane with no curvature and/or grade will require only one component while other sections may require components for turns and straight line course.

In the testing phase, the scaled velocity $c(i)$ of an oncoming vehicle is compared with the components in the i^{th} segment. An “anomaly” is detected if

$$|c(i) - \mu_{i,k}| > \alpha \cdot \sigma_{i,k} \quad \forall k. \quad (23)$$

Here α is a scalar that determines the threshold. Under Gaussian assumption, a value of $\alpha = 2$ would imply approximately 5% outliers. In the case when there exists some component such that the velocity difference is within the threshold, we validate the selection of this component by estimating the maneuver with shape analysis (described next). This ensures, for example, that if a vehicle’s velocity matches the component obtained for turning vehicles then that vehicle should also turn. A mismatch in the anticipated and observed maneuver would also result in an anomaly detection.

C. Shape Analysis

We now describe our methods to characterize the shape of a vehicle’s trajectory. In this exercise our goal is to develop techniques to identify significant maneuvers such as turns and lane changes. We construct a library of shapes associated with different maneuvers and match the vehicle’s trajectory to these templates. Note that we perform template matching in the hypothetical co-ordinates defined earlier and both x^h and y^h are used to define the vehicle position. In order to simplify discussion (and implementation) we use a complex number representation of the trajectory in this section.

Let a spatial trajectory be defined as:

$$P = \{p_t = x_t^h + jy_t^h : 1 \leq t \leq N\}, \quad (24)$$

where N is the number of samples in the trajectory and j denotes the square root of -1 . We obtain a trajectory template for each significant maneuver. Thus for k maneuvers we obtain the templates P_1, P_2, \dots, P_k .

1) *Procrustes Analysis*: According to Dryden [46], “shape is all the geometrical information that remains when location, scale, and rotational effects are filtered out from an object.” For analysis purposes we represent the shape of an object by a finite number of pixels on the object’s surface. There are different methods to estimate the similarity of object shapes. Procrustes shape analysis is one such method which is invariant to scale, translation, and rotation [47]. Our use of Procrustes analysis for turn detection is inspired by a similar work by Harguess and Aggarwal [48]. Our approach is different because we do not pre-segment the trajectories resulting in a real-time analysis.

Let each object be represented by a $N \times 1$ vector where N is the number of 2D points on the object’s surface (in our case, the number of points on the trajectory). We represent the two trajectories as $U = (U_1, U_2, \dots, U_N)^T$ and $V = (V_1, V_2, \dots, V_N)^T$ where $U, V \in \mathbb{C}^N$. That is, each 2D point is represented by a complex number as in Equation 24. The Procrustes distance is a minimum sum of squared distances shape metric that requires shapes with one-to-one corresponding point pairs. After filtering the location, scale, and rotational effects the minimum Procrustes distance configurations that is used in this paper is defined by:

$$d_F(U, V) = (1 - \frac{V^*UU^*V}{U^*UV^*V})^{1/2}. \quad (25)$$

2) *Shape Matching with Sliding Windows*: Our motivation for using a sliding window based method arises from the need for real-time operation. We observe that the approach used by Harguess [48] creates significant delay in the analysis because it requires the complete trajectory (or at least a large part of it) in order to detect steps, ramps, and impulses. Thus we devise a method to use template matching without a need for explicit segmentation of the trajectory.

Let us represent a window W as an index set defined by the two end points τ_1 and τ_2 :

$$W = \{\tau_1, \tau_1 + 1, \dots, \tau_2 : 1 \leq \tau_1 < \tau_2 \leq N\}. \quad (26)$$

We ensure sufficient samples in the windows by the constraint that $\tau_2 - \tau_1 > n_{min}$ ($n_{min} = 20$ in our experiments). The shape to be matched is constructed from the trajectory points located in the window. Thus,

$$\beta = \{p_t \in P : t \in W\}. \quad (27)$$

The shape descriptor thus obtained is a complex vector of length $\tau_2 - \tau_1 + 1$. In order to compute the Procrustes distance with respect to the k templates described above, two operations are needed. First, the vector is centered by subtracting the median value. Next, the centered vector must be resampled so that it contains exactly the same number of elements as the l^{th} template for each $1 \leq l \leq k$. The resampling is achieved using linear interpolation. Let $\hat{\beta}$ represent the centered and length-adjusted shape vector corresponding to a given window W .

Let Δ represent the Procrustes distances for $\hat{\beta}$ for each of the template. That is,

$$\Delta = \{\delta_l : 1 \leq l \leq k\}, \quad (28)$$

where δ_l is the distance from the l^{th} template shape. We declare the occurrence of a particular maneuver when the distance of its template is significantly smaller than the other distances. From our experiments, we observed that a 25–35% difference works well. We use 30% as the decision threshold in our experiments. An event is declared by marking p_{τ_2} as the position of occurrence.

Since the only two parameters determining a window are the end points, we now explain how these points get updated during the process.

- If a turn is detected, all but the very latest parts of the current window are discarded, and a new window is created with the smallest permissible size (n_{min}). Thus,

$$\tau_1^{new} = \tau_2^{old} - \frac{n_{min}}{2} \text{ and } \tau_2^{new} = \tau_2^{old} + \frac{n_{min}}{2}, \quad (29)$$

where the superscripts *new* and *old* have been inserted for illustration purpose.

- If a turn is detected and is the same type as the last detected turn, a u-turn is inferred if the two events occurred within a chosen distance from each other. We declare a u-turn if two identical turns occur within 50 samples of each other. The end points are updated as earlier.
- If the event detected is a straight line course or no event is detected, only τ_2 is modified – increased by one if the end of the trajectory has not been reached. The analysis terminates when the end is reached.
- If the event detected is a straight line course or no event is detected, one can optionally choose to discard the oldest points in the window if the window exceeds a certain size threshold. We discard the older half of the samples if the window becomes larger than 100 samples.

D. Overload Detection

In this task our goal is to detect from the exterior appearance and motion if a vehicle is carrying excess load which could potentially include illegal goods or persons. Overloading is also often associated with compromised handling and braking [49] thus requiring attention and monitoring. We assume in this analysis that the vehicles are not modified for extra load-bearing in which case the visible signs of loading would be absent.

Addition of a large load into a vehicle’s trunk causes two visible effects. The rear tires might appear more flat than the front tires, and the gap between the rear tire and the body (the wheel well) may appear smaller than that between the front tire and the body. While the flatness of the tires can be easily compensated by over-inflation, the lowering of the body cannot be avoided without modifying the suspension. Therefore we focus only on measuring the gaps.

Similar to the problem of accurate tire extraction, the problem of visual determination of rear-heaviness has not been explored sufficiently. Recall Figure 5 where the side view of a vehicle’s tire is illustrated. The region above the tire, marked as shadow is the gap we are interested in characterizing. More specifically, we need an accurate estimate of the thickness of this gap at a point which is vertically above the center of the

tire. Let us denote this value by δ . Ideally, one would like to compare this value with a predefined normal δ_{norm} for a vehicle of the same make and model. If the measured gap is significantly smaller, $\delta_{norm} - \delta > \tau$, it would be deemed anomalous. On the other hand, one can also compare the δ values for the front and rear tires. This approach is more appealing because it does not depend on the knowledge of the make and model (and the normal values associated with it), and the decision threshold can be specified directly in pixels, thereby avoiding additional errors that may arise from the image co-ordinate to world co-ordinate mapping.

Our approach to estimating the thickness of the gaps is by using the information about the tire positions. Consider the rear tire of a vehicle which has been analyzed by our tire size estimation technique. Let (x_t, y_t) be the location of the tire center and r_t be the best estimate of its radius. Our goal is to find a point (x^*, y^*) such that $x^* = x_t$ and $y_t - y^*$ is the smallest distance from the tire center to the vehicle body directly above it. Then, the gap thickness δ is given by $\delta = y_t - (y^* + r_t)$ (recall that y increases from top to bottom). Note that this analysis could be made more robust by using color values which would help identify the shadow region which is invariably achromatic.

An identical set of operations would be performed on the other (front) tire and the gap value recorded as δ_{front} . Then, the decision about rear-heaviness is made if $\delta_{front} - \delta_{rear} > \tau$ where τ is an experimentally determined constant. Note that the difference is not absolute. This is because under normal conditions many front engine cars may have smaller front gap than the rear. Thus the method may detect a false positive if absolute difference is taken.

Note that it is also possible to characterize the vertical oscillations of a vehicle in order to detect overloading. Such an approach was discussed by Dockstader in [50]. Our system can optionally use the cleat/road bump to excite the impulse response of the vehicle’s oscillation. Then the method in [50] or [51] may be used to analyze the bounce.

V. EXPERIMENTAL RESULTS

We used the methods described in the previous sections to analyze various traffic video sequences for our experiments. There are two important points about the experiments. First, our primary goal in this paper is to demonstrate the feasibility of our system using off-the-shelf equipment and easily obtained video data. While a thorough testing of the system would require running the system in real time using live video feed, such experimentation is not reported in this paper. Secondly, we conducted many of our experiments using publicly available datasets. However, due to the unique camera configuration proposed for our system, all the methods cannot be tested on these datasets. Therefore, we also describe results for our methods using a 20 minute front and side view video sequence collected for this purpose.

Vehicle detection: Our method of motion-assisted background subtraction (MABS, as described in Section III-A) was used for surveillance sequences containing vehicles against an uncontrolled outdoor background. Four such sequences were

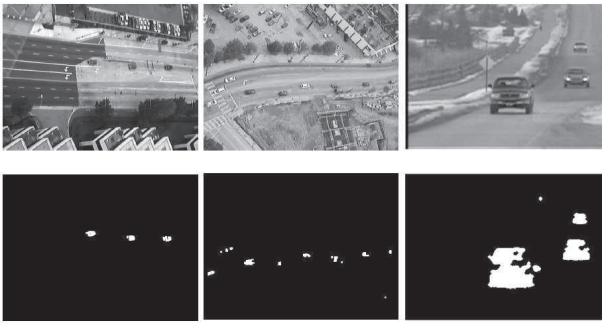


Fig. 11. Examples of vehicle detection from different video sets (A: Lankershim, B: Peachtree, and C: Authors).

used to test the method – (A) the Lankershim Boulevard dataset [52] (camera 2, footage starting at 8.30), (B) the Peachtree Street dataset [52] (camera 4, footage starting at 4.00), and (C) a traffic video recorded by the authors. This video was taken from a much lower height (about 6 feet (1.8m) above the road) and with overcast weather conditions. Thus, unlike the first two datasets the size of vehicle “blobs” changes with its position in the camera’s field of view.

A detection was defined as a successful hand-off from object detection method to the particle filter-based tracker. In other words, we count a true positive when a new “target” is created in the tracker following its entry into the field of view. This was depicted by the red ellipses in the Figure 9. Missed detections and false positives have corresponding definitions. However, note that we do not focus on the results of the tracker in this paper. Instead the significant visual output from the object detection stage is the foreground-background mask. Some representative frames from the different datasets and their foreground masks are provided in Figure 11.

During the nearly fifteen (combined) minutes of analysis, approximately one thousand vehicles appeared in the video. Approximately five percent of detection failures occurred. Most of the failures were missed detections when the vehicles were highly similar to the background (the road). Failures were also encountered when the system counted two vehicles moving very close to each other as a single object. In the case of the SVC, no tracking is performed. Thus the results are evaluated visually and by the success of other methods which use this output. An example of background subtraction on the SVC output is presented in Figure 12.



Fig. 12. An example of vehicle detection from the SVC video.

In order to estimate the time complexity of the method, we created five video sequences with dynamic background, large foreground objects, and sudden illumination changes. Owing to the difficulty of creating such test cases with vehicles, we used human subjects for this experiments. The

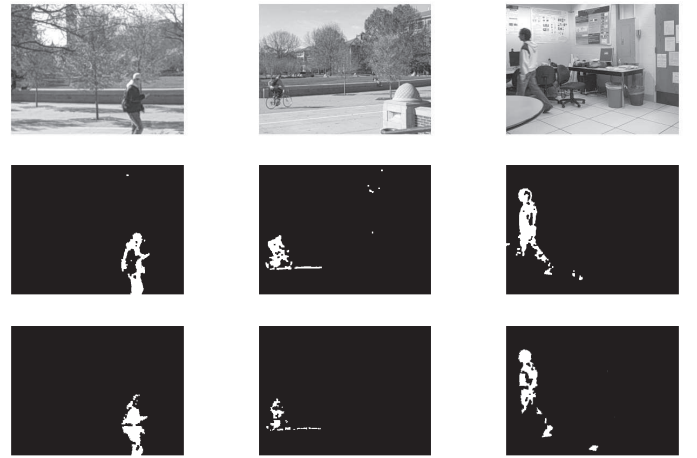


Fig. 13. Comparison of MABS (bottom) with Gaussian mixture models (GMM, center) for object detection on three test sequences. Columns 1 and 2 show processed frames from sequences with considerable background clutter while column 3 shows a frame shortly after a sudden change of illumination.

TABLE I
RUNNING TIME FOR THE TWO OBJECT DETECTION METHODS (IN SECONDS). THE EXPERIMENTS WERE CONDUCTED IN MATLAB RUNNING ON A MAC OS-X MACHINE.

Sequence	Frames	GMM	MABS
Seq 1	100	1020	102
Seq 2	100	994	104
Seq 3	180	1910	225
Seq 4	270	2150	224
Seq 5	210	2092	193

proposed method for background subtraction was used on these sequences. We also used a background subtraction method based on Gaussian mixture models (GMM) with each pixel represented by a mixture of three Gaussian components. The parameters were selected such that both methods provided comparable outputs (visually determined). Figure 13 shows some examples from the video sequences and the outputs of the two methods. The execution time of the two methods are listed in Table I. It can be observed that the proposed method is much faster than the GMM method and hence, is more suitable for our application.

Vehicle type determination: Our template-based vehicle body type detector was designed to use the side view of vehicles. Since such a dataset is not readily available, we generated a testing dataset with about 20 minutes of traffic video. The camera was placed about one foot above the road surface and positioned such that the field of view was large enough to contain 3 – 4 typical length vehicles. Some frames from the dataset can be seen in the top row of Figure 14. Most vehicles were moving between 30 and 45 miles per hour. It should be noted that the presence of a speed bump would result in lower speeds in the SVC’s field of view but we have maintained in this paper that the speed bump is an optional component. Thus our methods are tested for a more general (and difficult) situation. As stated earlier, we restrict the vehicle type to one of the four classes – sedan, light truck, sport/utility, and hatchback. Hence, the frames used in this

experiment were manually selected.

The method was used for 85 vehicles roughly equally distributed over the four classes. While all the sedans and trucks were correctly identified, the SUV and hatchback classes were not so well separated. Nine vehicles were assigned to the wrong class, and some other (correct) classifications were achieved with a small margin. Figure 14 shows examples of classification with a good margin, another with a small margin, and a failure case. Note that, the y -axis represents the normalized absolute difference. Thus, the class with the smallest difference is chosen. It should also be noted that the proximity of the SUV and hatchback classes is expected since both types have similar shapes. These classes can be easily separated by using a simple threshold on the vehicle length because most utility vehicles are bigger than hatchbacks.

Tire size estimation: As in the case of body type determination the methods for tire size estimation require a side view of the vehicles. Thus the above-mentioned dataset (containing 20 minutes of footage) was also used in these experiments. The ground truth regarding the tire sizes (measured in pixels) was obtained manually. Therefore, these methods were used with manually selected frames and on a smaller testing set with 45 vehicles. In all but one cases, the tire diameter was correctly estimated to within one pixel tolerance. The failure case reported an incorrect position for the tire center and hence, the size estimate was irrelevant. Figure 15 shows some examples of the vehicles and the tire region as extracted by the system. The failure case is also shown.

Note that all 45 vehicles in these experiments had a bright hub compared to the rubber. In the absence of a suitable test case, the method for estimating the size of a tire with dark hub was used on an synthetic test case. This was done by manually coloring the tire area of a vehicle’s image and is illustrated in Figure 16. The position of the lower-most point of the tire was provided as input to the system (corresponding to the position of the cleat as explained in Section III-C). The diameter of the tire was correctly estimated. We would like to emphasize that this particular experiment (with dark tires) was only conducted as a proof of concept and does not capture rigorous evaluation of the method. On the contrary, the absence of vehicles with such tires in the 20 minutes of traffic footage itself demonstrates the rarity of such a situation.



Fig. 16. Result of tire extraction on a vehicle with artificially colored dark tires. The position of the lowermost point of the tire was provided as input.

Make recognition: Many experiments on vehicle make and model recognition are carried out using “clean” images of vehicles. Such images have high resolution and use vehicles which are stationary or near-stationary (for example, near the entrance of a parking lot or at a toll booth). Instead our methods were used on video frames taken in outdoor conditions and on vehicles moving as fast as 45 miles per

TABLE II

THE GAP ABOVE TIRES MEASURED IN PIXELS FOR THE TEST VEHICLES.

Vehicle	Unloaded		Loaded	
	Front	Rear	Front	Rear
Vehicle 1	56	57	55	49
Vehicle 2	53	53	54	48
Vehicle 3	70	72	71	67

hour. In fact, the traffic video used in this experiment was recorded at the same time and location as the video used in the previous two experiments. For some vehicles, the front grille area used to determine the make was as small as 30×15 pixels. Some sample frames from the dataset are shown in the top row of Figure 17. One can observe that the camera position does not produce strictly front view images (as would be produced by an overpass camera). This does not significantly affect the results because the edge signatures of the selected makes are also obtained from vehicles in the same video sequence. Naturally, these prototype vehicles are not used in the testing set.

Vehicles from five car makers were used to test our methods. These five makes were determined by viewing the video and identifying the most frequently occurring makes. Then a total of 65 vehicles were selected as the testing set. Of these, the system correctly recognized 56 makes. As in the case of vehicle type experiment some decisions had a large margin (or confidence) while others (both correct and incorrect) were close decisions. Figure 17 shows some examples of the make recognition output. These include both the well-resolved and the marginal cases. Note that the y -axis represents a normalized similarity metric. Thus the class with the highest value is selected. It can be observed that the classification accuracy is lower than some of the state-of-the-art make recognition methods. Even the edge orientation method of Petrovic and Cootes [26] results in about a 90 percent accuracy. This can be explained by two factors – the use of more realistic images and the automatic selection of the front region of interest. While these two factors reduce the accuracy our experiments prove that the method is robust enough to be deployed in real-life scenarios.

Overload Analysis: The purpose of this particular experiment was to investigate our methods for detecting overloaded vehicles. Due to the nature of the problem we tested our method on a small number of vehicles. The vehicles were loaded by placing 500 pounds of sand bags in the trunk. We imaged the lowering of the vehicle due to loading, as well as drove the vehicle over a bump to observe the pattern of oscillations. A checkerboard shaped fiducial marker was attached to each test vehicle to allow easier tracking.

Figure 18 shows 3-tuples of images corresponding to the test vehicles under unloaded and loaded conditions, and the difference image. It should be noted that these images has been significantly downsized. It can be observed that for all the three test vehicles, the vehicle depresses (captured by the thick white patches above the tire) by a visually significant amount in the rear while the front end is unaffected. In a standard resolution video frame, the lowering was between 5

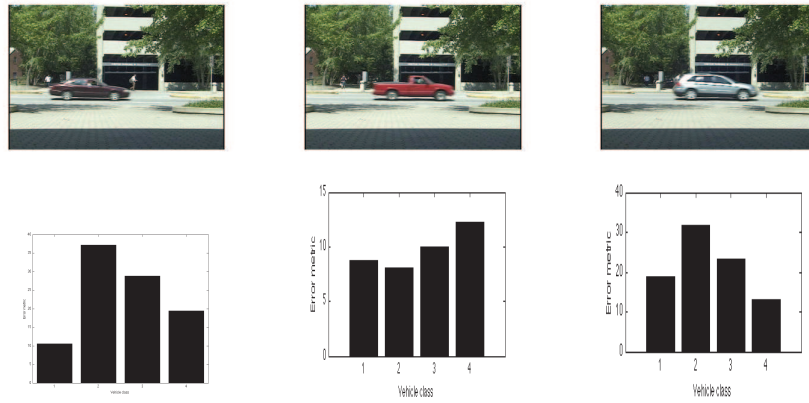


Fig. 14. Examples of vehicle type determination using shape matching. The bars in the bottom row represent error metric (SAD), thus lowest error class is selected. The three examples represent (left to right) cases of classification with high confidence, classification with low confidence, and a mis-classification.

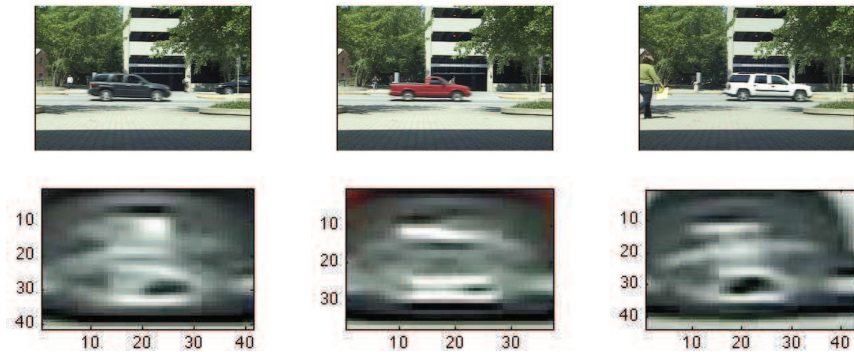


Fig. 15. Examples of vehicle tire extraction using the circular edge search method. In each case the rear tire was segmented and is shown in the bottom row. Example 3 shows the failure case.

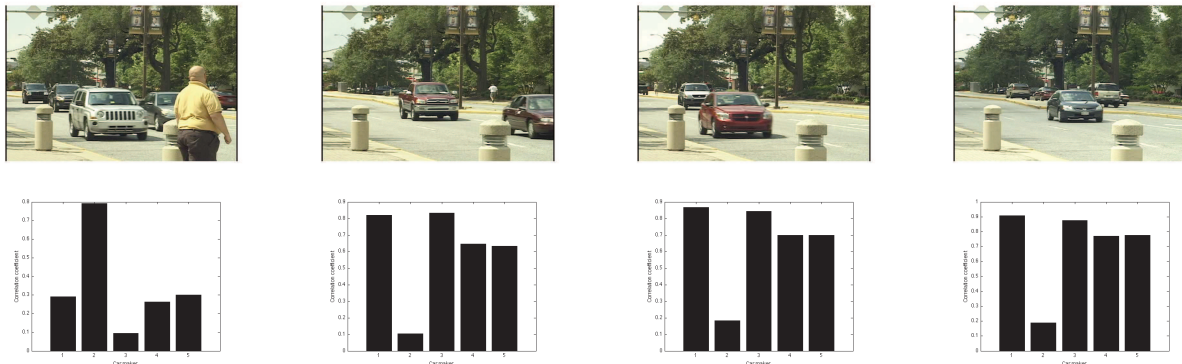


Fig. 17. Examples of vehicle make recognition using edge orientation histograms and automatic region selection. The bars in the bottom row represent the similarity metric, thus highest similarity make is selected. Example 4 shows a failure case.

and 8 pixels. The gap above the front and rear tires under loaded and unloaded conditions is listed in Table II.

Trajectory Analysis: These results are provided for three categories – evaluation of the co-ordinate transformation technique, evaluation of the velocity analysis method, and the evaluation of turn identification using shape analysis. Three video sequences were used in our experiments. These included the Lankershim Boulevard, California surveillance dataset generated for the NGSIM project [52] (denoted by LANK in this section) and two videos recorded by the authors (denoted by ANON1 and ANON2). A sample of the video sets is shown in Figure 19. In order to illustrate the role of the hypothetical

co-ordinate system, we also provide a satellite image of the road in the field of view, using Google Maps [53]. For the datasets collected by the authors, the labels have been erased.

We present two cases in which the proposed look-up table based method has been used on curved trajectories. These are shown in Figure 20. The first case is a synthetic trajectory created for the curved road used to illustrate the training process in Section IV-A. The second case is a trajectory from the ANON2 dataset. The linearized trajectories (in the hypothetical co-ordinate system) are shown in the second row. Note that in the second case, a single 2D LUT is used to transform from the image co-ordinates to the desired co-

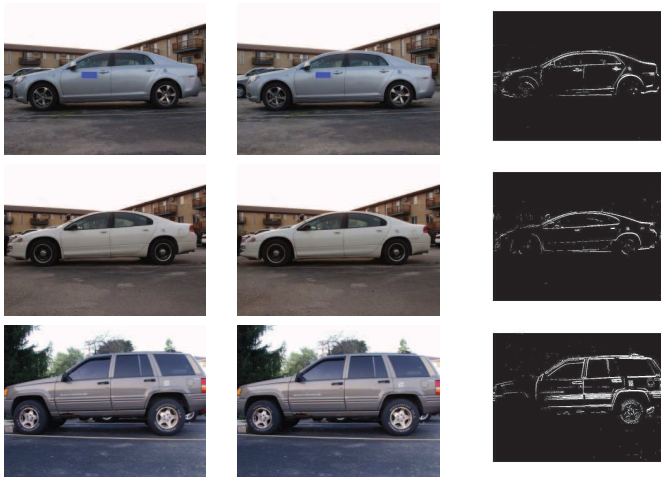


Fig. 18. Illustration of the visual impact of loading on test vehicles. The difference image (column 3) captures the displacement caused by loading. Columns 1 and 2 show the unloaded and loaded vehicles.



Fig. 19. Sample images from our test videos (a-c) and the corresponding map (d-f) (courtesy: Google Maps [53]). The field of view is highlighted in yellow and the labels have been erased.

ordinates. The perturbation seen in the upper part of the transformed trajectory (in case 2) is due to the sensitivity of the transformation very deep in the field of view. This error can be improved by better user input or with alternative camera calibration methods although the non-linearity will always cause greater errors in the far field of view.

Our method for analyzing the approach velocity of a vehicle was used with the ANON1 dataset. Different scenarios were realized by driving a designated vehicle in various manners. In Figure 21, we present the decision vectors corresponding to four cases – driving at nearly uniform velocity, unexpected slowing, unexpected sudden acceleration, and making a u-turn. Recall that we treat velocity as a one-dimensional signal (in the direction of the road) as a function of the position. Thus, the vertical axis represents the scaled velocity at different points in the roadway. The blue circles indicate normal approach while the red crosses denote (atomic) anomalies. The dashed black line represents the mean (scaled) velocity estimated from the “normal” vehicles. It is clear to see that this treatment and the resultant method for visualization of the decisions would enable very fast decision-making by an operator.

Shape analysis for detecting turns was used on both syn-

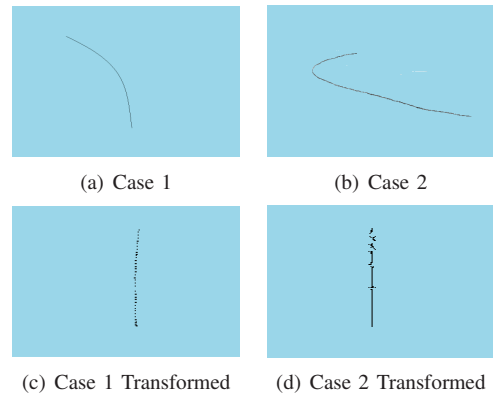


Fig. 20. Results of the LUT-based co-ordinate transformation to curved trajectories. Case 1 pair is a synthetic test case while the Case 2 pair is from the dataset ANON2.

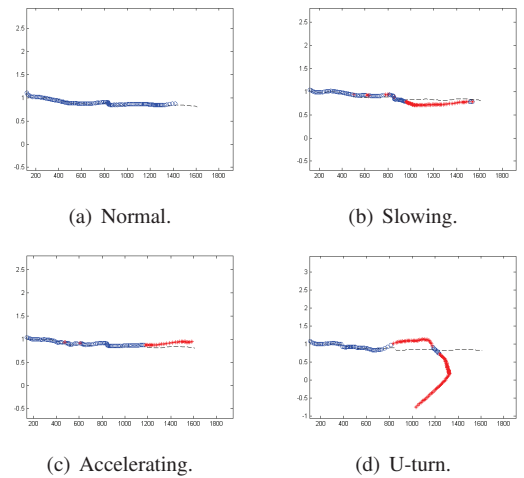


Fig. 21. Plot of the decision vectors corresponding to different driving behaviors. The red crosses indicate occurrence of atomic anomalies.

thetic and true trajectories. Figure 22 shows the examples of template matching output on four types of maneuvers. These trajectories were obtained by tracking individual vehicles in the LANK dataset [52]. The black circle indicates the starting end of the trajectory (since there is no temporal information in these plots). Right turns are denoted by red circles and left turns by blue circles. The color of a u-turn marker is chosen based on the turn being left-handed or right-handed. A total of 46 vehicle trajectories were analyzed and 45 maneuvers were correctly identified. These cases included vehicles from East-West and North-South traffic and roughly equal number of right, left, and no turns. The u-turn case in Figure 22 was the only such maneuver in the dataset. The failure case occurred when a vehicle made a turn after prolonged delay and our shape matching method did not detect the turn.

A synthetic trajectory was generated to simulate complex multi-turn scenarios to test the effectiveness of our methods. Figure 23 shows the result of turn detection on the synthetic trajectory. It can be seen that our methods not only detect all the turns but also declare the occurrence of turns at a logically correct point in the trajectory.

Finally, we demonstrate the usefulness of the hypothetical

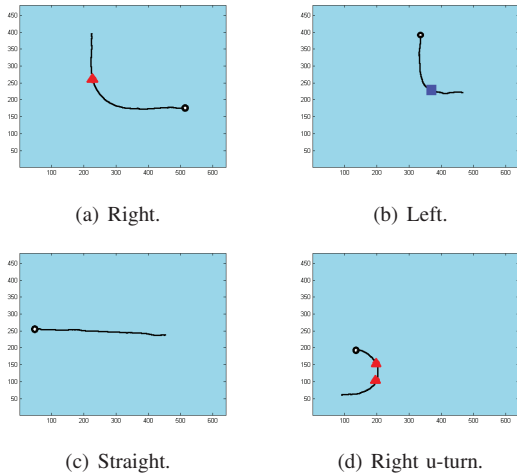


Fig. 22. Results of shape analysis on trajectories of different maneuvers from the LANK dataset. A black circle is used to denote the starting end of the trajectory.

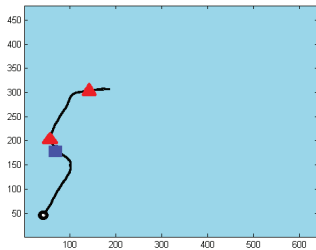


Fig. 23. Result of shape analysis on a synthetic multi-turn trajectory.

co-ordinate system by using shape matching for the two test cases shown in Figure 20. The window method was used to detect turns in the original (ground) and the transformed trajectories (in the hypothetical co-ordinates). The results are shown in Figure 24. Clearly, the linearization enforced by the co-ordinate transformation is effective in suppressing the false turn detections.

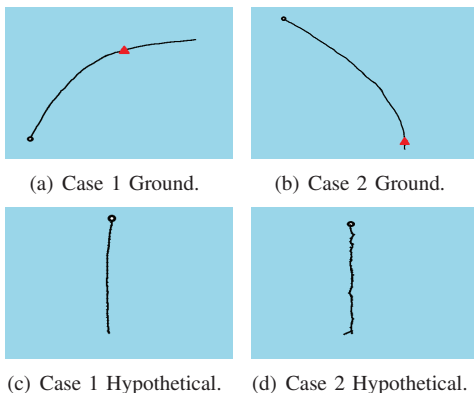


Fig. 24. Window method for turn detection applied to curved roads. The rows represent output of turn detection in ground (a-b) and hypothetical (c-d) co-ordinates. Note that false turns are detected only in the ground co-ordinates.

Discussion of Results: The results described above can

be summarized as follows. Many of our methods were tested with a large number of test cases obtained without requiring simulation of anomalous situations. These results and the accuracies of the methods are provided in Table V. Other methods such as velocity anomaly detection, overload detection, and co-ordinate mapping are discussed with lesser rigor.

TABLE III
SUMMARY OF EXPERIMENTAL RESULTS FOR OUR ANALYSIS
TECHNIQUES.

Task	Testing Set	Success	Accuracy
Vehicle Detection ¹	1012	960	94.9%
Body Type Identification	85	76	89.4%
Make Recognition	65	56	86.1%
Tire Size Estimation ²	45	44	97.7%
Turn Detection ³	46	45	97.8%

¹ Only the front view detection results are reported.

² The artificially colored dark tire is not included.

³ The synthetic multi-turn trajectory is not included.

It can be observed that our methods have high accuracies despite the uncontrolled outdoor setup and without any strong assumptions. The accuracies in object detection, body type identification and make recognition are comparable with many state-of-the-art techniques for these tasks. Visual estimation of tire size is a novel application and we achieve high accuracy. While other techniques exist for shape matching (for turn detection) our sliding window based approach is accurate and more suitable for a real-time operation.

The method for curvature compensation by transforming to a hypothetical co-ordinate system was tested on three videos from three geographic locations. The curvature was successfully corrected in all three cases. The number of test cases was limited by non-availability of many videos with ground truth (for training). However, the three tests were selected to cover the “typical” as well as the difficult cases.

Similarly, the method for detection of anomalous patterns in a vehicle’s approach velocity was only tested on videos where such situations were simulated. Such experiments (by creating anomalous driving situations) were conducted in four geographic locations with different vehicles and/or drivers. Results similar to those reported in Figure 21 were obtained which showed the method to be reliable and accurate.

Finally, the method for characterizing rear-heaviness by observing the gaps above the front and rear tires were tested for a small number of vehicles. As stated earlier, this experiment on estimating rear-heaviness was conducted as a proof-of-concept and encouraging results were obtained. We observed that our working hypothesis that addition of load causes unequal (in the front and the rear) and visible lowering of the vehicle was valid for three different vehicles of different body types and load capacities. Further experiments would be needed to identify cases where the hypothesis does *not* hold true. This analysis could also be combined with dynamic analysis of vehicle oscillation.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed an enhanced video surveillance system which can observe vehicular traffic from a stand-off range without human supervision. Our system searches

for patterns in the appearance and behavior of vehicles which could be associated with abnormalities. We describe a camera deployment scheme which provides simultaneous front and side views of the vehicles. We provide details of low-complexity image analysis methods which can detect and track vehicles and extract information from the videos including type, make, tire size, and velocity changes. We present the results of testing these methods on real traffic videos which show the feasibility and effectiveness of our approach. We also demonstrate that many of our methods are more suitable for real life applications than existing methods.

Future extensions to this work might include additional analyses to obtain more information from video/non-video sensors. Examples of such analysis would be estimation of load distribution along the vehicle's length and monitoring the behavior of the occupants. We would also like to test the robustness of the analysis techniques to changes resulting from traffic, weather, and illumination.

REFERENCES

- [1] G. J. D. Smith, "Behind the screens: Examining constructions of deviance and informal practices among CCTV control room operators in the UK," *Surveillance and Society*, vol. 2, no. 2-3, pp. 376–395, 2004.
- [2] J. Tullio *et al.*, "Experience, adjustment, and engagement: The role of video in law enforcement," *Proceedings of the ACM Conference on Human Factors in Computing Systems*, Atlanta, Georgia, April 2010, pp. 1505–1514.
- [3] E. Wallace and C. Diffley, "CCTV control room ergonomics," Technical Report 14/98, Police Scientific Development Branch, UK Home Office, 1998.
- [4] P. Meer, *Emerging Topics in Computer Vision*. Englewood Cliffs, New Jersey: Prentice Hall, 2004.
- [5] H. M. Dee and S. A. Velastin, "How close are we to solving the problem of automated visual surveillance?" *Machine Vision and Applications*, vol. 19, no. 5-6, pp. 329–343, September 2008.
- [6] N. Haering, P. L. Venetianer, and A. Lipton, "The evolution of video surveillance: an overview," *Machine Vision and Applications*, vol. 19, no. 5-6, pp. 279–290, September 2008.
- [7] B. Coifman, D. Beymer, P. McLauchlan, and J. Malik, "A real-time computer vision system for vehicle tracking and traffic surveillance," *Transportation Research: Part C*, vol. 6, no. 4, pp. 271–288, 1998.
- [8] R. T. Collins *et al.*, "A system for visual surveillance and monitoring," Final Report CMU-RI-TR-00-12, Carnegie Mellon University, 2000.
- [9] J. Aguilera *et al.*, "Visual surveillance for airport monitoring applications," *Proceedings of the Computer Vision Winter Workshop*, Telc, Czech Republic, February 2006.
- [10] X. Li and F. M. Porikli, "A hidden markov model framework for traffic event detection using video features," *Proceedings of the IEEE International Conference on Image Processing*, vol. 5, Singapore, October 2004, pp. 2901–2904.
- [11] Y. Tian *et al.*, "IBM smart surveillance system (s3): event based video surveillance system with an open and extensible framework," *Machine Vision and Applications*, vol. 19, no. 5-6, pp. 315–327, September 2008.
- [12] D. Gutchess *et al.*, "Video surveillance of pedestrians and vehicles," *Proceedings of the SPIE: Acquisition, Tracking, Pointing, and Laser Systems Technologies*, vol. 6569, Orlando, Florida, April 2007, pp. 65690E:1–11.
- [13] A. M. McIvor, "Background subtraction techniques," *Proceedings of Image and Vision Computing*, Hamilton, New Zealand, November 2000, pp. 147–153.
- [14] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, August 2000.
- [15] N. Ohta, "A statistical approach to background suppression for surveillance systems," *Proceedings of the International Conference on Computer Vision*, Vancouver, Canada, July 2001, pp. 481–486.
- [16] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background modeling and subtraction of dynamic scenes," *Proceedings of the International Conference on Computer Vision*, Nice, France, October 2003, pp. 1305–1312.
- [17] S. J. McKenna and S. Gong, "Tracking color objects using adaptive mixture models," *Image and Vision Computing*, vol. 19, pp. 225–231, 1999.
- [18] M. Seki, H. Fujiwara, and K. Sumi, "A robust background subtraction method for changing background," *Proceedings of the Workshop on Applications of Computer Vision*, Palm Springs, California, December 2000, pp. 207–213.
- [19] S. Y. Cheung, S. Coleri, B. Dundar, S. Ganesh, C. Tan, and P. Varaiya, "Traffic measurement and vehicle classification with single magnetic sensor," *Transportation Research Record*, vol. 1917, no. 19, pp. 173–181, 2005.
- [20] X. Song and R. Nevatia, "A model-based vehicle segmentation method for tracking," *Proceedings of the International Conference on Computer Vision*, vol. 2, Beijing, China, October 2005, pp. 1124–1131.
- [21] R. P. Avery, Y. Wang, and G. S. Rutherford, "Length-based vehicle classification using images from uncalibrated video cameras," *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, Washington, D.C., October 2004, pp. 737–742.
- [22] C. Huang and W. Liao, "A vision-based vehicle identification system," *Proceedings of the International Conference on Pattern Recognition*, vol. 4, Cambridge, UK, August 2004, pp. 364–367.
- [23] B. Morris and M. Trivedi, "Robust classification and tracking of vehicles in traffic video streams," *Proceedings of the IEEE Intelligent Transportation Systems Conference*, Toronto, Canada, September 2006, pp. 1078–1083.
- [24] A. H. S. Lai and N. H. C. Yung, "Vehicle type identification through automated virtual loop assignment and block-based direction-biased motion estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 2, pp. 86–97, June 2000.
- [25] R. J. Barnett, "Wireless remote tire parameter measurement method and apparatus," United States Patent 6,448,891, 2002.
- [26] V. S. Petrovic and T. F. Cootes, "Vehicle type recognition with match refinement," *Proceedings of the International Conference on Pattern Recognition*, vol. 3, Cambridge, UK, August 2004, pp. 95–98.
- [27] X. Clady, P. Negri, M. Milgram, and R. Poulernard, "Multi-class vehicle type recognition system," *Proceedings of the International Workshop on Artificial Neural Networks in Pattern Recognition*, Paris, France, July 2008, pp. 228–239.
- [28] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–100, 2004.
- [29] L. Dlagnevov, "Video-based car surveillance: License plate, make and model recognition," Masters Thesis, University of California, San Diego, 2005.
- [30] D. A. Torres, "More local structure information for make-model recognition," Project report, CSE252, University of California, San Diego, 2005.
- [31] I. Zafar, E. A. Erdisinghe, S. Acar, and H. E. Bez, "Two dimensional statistical linear discriminant analysis for real-time robust vehicle type recognition," *Proceedings of IS&T/SPIE Electronic Imaging: Real Time Image Processing*, vol. 6496, San Jose, California, January-February 2007.
- [32] I. Sobel and G. Feldman, "A 3×3 isotropic gradient operator for image processing," Unpublished, presented at Stanford Artificial Project, 1968.
- [33] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, vol. 34, no. 3, pp. 334–352, August 2004.
- [34] B. T. Morris and M. M. Trivedi, "A survey of vision-based trajectory learning and analysis for surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1114–1127, August 2008.
- [35] 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*, vol. 23, no. 8, pp. 873–889, August 2001.
- [36] T. Zhang, H. Lu, and S. Li, "Learning semantic scene models by object classification and trajectory clustering," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Miami, Florida, June 2009, pp. 1940–1947.
- [37] A. Basharat, A. Gritai, and M. Shah, "Learning object motion patterns for anomaly detection and improved object detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008, pp. 1–8.
- [38] S. Srivastava and E. J. Delp, "Standoff video analysis for the detection of security anomalies in vehicles," *Proceedings of the IEEE Applied*

Imagery Pattern Recognition Workshop, Washington, DC, October 2010, pp. 5759685:1–8.

- [39] S. Srivastava, K. K. Ng, and E. J. Delp, “Co-ordinate mapping and analysis of vehicle trajectory for anomaly detection,” *Proceedings of the IEEE International Conference on Multimedia and Expo*, Barcelona, Spain, July 2011.
- [40] I. Haritaoglu, D. Harwood, and L. S. Davis, “W4: Real-time surveillance of people and their activities,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, August 2000.
- [41] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *ACM Computing Surveys*, vol. 38, no. 4, pp. 13:1–45, December 2006.
- [42] K. K. Ng and E. J. Delp, “Object tracking initialization using automatic moving object detection,” *Proceedings of SPIE/IS&T Electronic Imaging: Visual Information Processing and Communication*, vol. 7543, San Jose, California, January 2010, pp. 75430M:1–12.
- [43] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, February 2002.
- [44] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. New York: Cambridge University Press, 2004.
- [45] D. Shepard, “A two-dimensional interpolation function for irregularly-shaped data,” *Proceedings of the ACM National Conference*, Princeton, New Jersey, January 1968, pp. 517–524.
- [46] I. L. Dryden and K. V. Mardia, *Statistical Shape Analysis*. New Jersey: Wiley, 1998.
- [47] J. C. Gower and G. B. Dijksterhuis, *Procrustes Problems*. Oxford, UK: Oxford University Press, 2004.
- [48] J. Harguess and J. K. Aggarwal, “Semantic labeling of track events using time series segmentation and shape analysis,” *Proceedings of the IEEE International Conference on Image Processing*, Cairo, Egypt, November 2009, pp. 4317 – 4320.
- [49] Vehicle and U. Operator Safety Agency, “Vehicle safety: the dangers of overloading,” Guidance sheet: http://www.dvtani.gov.uk/uploads/compliance/VOSA_VehicleSafety_DangersofOverloading.pdf, Accessed: July 2010.
- [50] S. H. Dockstader, “Motion trajectory classification for visual surveillance and tracking,” *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, Sydney, Australia, November 2006.
- [51] C. Liu, A. Torralba, W. T. Freeman, F. Durand, and E. H. Adelson, “Motion magnification,” *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 519–526, July 2005.
- [52] The Next Generation Simulation (NGSIM) Community, “Data sets,” Web archive: <http://ngsim-community.org/>, Accessed: May 2010.
- [53] “Google map,” Online: <http://maps.google.com>, Accessed: August 2010.



Edward J. Delp (S.70.M.79.SM.86.F.97) was born in Cincinnati, OH. He received the B.S.E.E. (cum laude) and M.S. degrees from the University of Cincinnati, and the Ph.D. degree from Purdue University, West Lafayette, IN. In May 2002, he received an Honorary Doctor of Technology from the Tampere University of Technology, Tampere, Finland. From 1980-1984, he was with the Department of Electrical and Computer Engineering at The University of Michigan, Ann Arbor. Since August 1984, he has been with the School of Electrical and

Computer Engineering and the School of Biomedical Engineering at Purdue University. From 2002-2008, he was a chaired professor and held the title The Silicon Valley Professor of Electrical and Computer Engineering and Professor of Biomedical Engineering. In 2008, he was named a Distinguished Professor and is currently The Charles William Harrison Distinguished Professor of Electrical and Computer Engineering and Professor of Biomedical Engineering. In 2007, he received a Distinguished Professor appointment from the Academy of Finland as part of the Finland Distinguished Professor Program (FiDiPro). This appointment is at the Tampere International Center for Signal Processing at the Tampere University of Technology. His research interests include image and video compression, multimedia security, medical imaging, multimedia systems, communication, and information theory. He has also consulted for various companies and government agencies in the areas of signal, image, and video processing, pattern recognition, and secure communications. He has published and presented more than 400 papers. Dr. Delp is a Fellow of the SPIE, a Fellow of the Society for Imaging Science and Technology (IS&T), and a Fellow of the American Institute of Medical and Biological Engineering. In 2004, he received the Technical Achievement Award from the IEEE Signal Processing Society (SPS) for his work in image and video compression and multimedia security. In 2008, he received the Society Award from the SPS. This is the highest award given by SPS and it cited his work in multimedia security and image and video compression. In 2009, he received the Purdue College of Engineering Faculty Excellence Award for Research.



Satyam Srivastava IEEE Student Member since 2009. B.E. (Honors) Electrical and Electronics Engineering Birla Institute of Technology and Science, Pilani, India 2006. Doctor of Philosophy Electrical and Computer Engineering Purdue University, West Lafayette, Indiana 2011. Research interests include image and video processing, color science, human and machine vision, and information theory.