5

Detection of Moving Violations

Wencheng Wu[1], Orhan Bulan[2], Edgar A. Bernal[3] and Robert P. Loce[1]

[1] *Conduent Labs, Webster, NY, USA*
[2] *General Motors Technical Center, Warren, MI, USA*
[3] *United Technologies Research Center, East Hartford, CT, USA*

5.1 Introduction

Law enforcement agencies and municipalities are increasing the deployment of camera-based roadway monitoring systems with the goal of reducing unsafe driving behavior. The most common applications are detection of violations for speeding, running red lights or stop signs, wrong-way driving, and making illegal turns. Other applications are also being pursued, such as detection of tailgating, blocking the box, and reckless driving. While some camera-based systems use the acquired images solely for evidentiary purposes, there is increasing use of computer vision techniques for automating the detection of violations.

Most applications in roadway computer vision systems involve analyzing well-defined and acceptable trajectories and speeds, which leads to clearly defined rules and detections. In some cases, the detections are binary, such as in red light enforcement or RLE (stopped or not), or divided highway driving (wrong way or correct way). Other applications require increased accuracy and precision, such as detecting speed violations and applying a fine according to the estimated vehicle speed. There are other deployed applications where the violation involves less definitive criteria, such as reckless driving.

The following sections present various applications, giving the motivation, system requirements, methodology, and effectiveness. The more common applications of speed and stop light will be described in detail, while the less common will be briefly noted.

5.2 Detection of Speed Violations

Studies have shown that there is a strong relationship between excessive speed and traffic accidents. In the United States in 2012, speeding was a contributing factor in 30% of all fatal crashes (10,219 lives) [1]. The economic cost of speeding-related crashes was estimated at $52 billion for 2010 [2]. In extensive review of international studies, automated speed enforcement is estimated to reduce injury-related crashes by 20–25% [3]. Hence, there is significant motivation to evolve and deploy speed enforcement systems.

Computer Vision and Imaging in Intelligent Transportation Systems, First Edition.
Edited by Robert P. Loce, Raja Bala and Mohan Trivedi.

Companion website: www.wiley.com/go/loce/ComputerVisionandImaginginITS

A typical photo enforcement system for speed violations consists of (i) an imaging module, which provides visual confirmation of the violation; (ii) a speed measurement module; and (iii) a citation issuing module, which issues the citation to a violator based on the information collected from the imaging and speed measurement modules. Many technologies have been developed and deployed to real-world environments in all three modules. In this section, we focus the discussion on technologies applied to speed measurement using computer vision.

Common methods for speed measurement in transportation include the use of in-ground inductive loops, radar, lidar, and video cameras. There are several advantages to the use of a vision system over the use of inductive loops or radar/lidar, while presenting new challenges that need to be addressed. The disruption and expense of installing in-ground induction loops can be avoided. A vision system is needed to recognize the vehicle, so extending its capabilities to measure speed eliminates the cost and complexity of additional system components associated with radar and lidar. A high-quality vision system can include intelligence to perform additional functions, such as recognition of tailgating, reckless driving, and accidents; gather usage statistics; and serve as a general surveillance device. Conceptually, it is fairly simple for a vision system to provide some measure of speed of an object once the object of interest is properly detected, identified, and tracked. The issue is the accuracy and precision of the measurement. Although a significant body of research exists on applying computer vision technologies to traffic and traffic flow measurements, only a very small fraction of published research evaluates accuracy and precision of speed measurement of an individual vehicle, which is critical for speed enforcement applications.

5.2.1 Speed Estimation from Monocular Cameras

In order to yield accurate speed measurement of individual vehicles via computer vision, a first requirement is good performance of the vehicle detection and tracking methods. Much research has been conducted in object detection and tracking as fundamental building blocks for video processing technologies. An excellent survey can be found in Ref. [4]. Although many of these techniques are readily applicable to vehicle tracking for speed measurement, there is a distinct aspect that needs to be considered here. More specifically, common methods focus on a coarser concept of tracking. The objective of most trackers is to track the object "as a whole." The operation of a tracker is considered effective as long as it can track the object as it appears or reappears in the scene over time under various practical noises. For speed measurement, the tracking objective needs to be more refined: it is necessary to track a specific portion(s) of the object. Consider a simple example. If a tracker starts with a reference point located about the front of a vehicle, and as the vehicle moves leading to a change in perspective, ends with a reference point located about the rear of a vehicle, the tracked trajectory alone is not adequate to estimate speed with sufficient accuracy for most speed law enforcement applications. Consequently, a suitable tracker for an accurate speed measurement system adheres to one of the following: (i) directly tracking a specific portion(s) of the object to determine its trajectory or (ii) coarsely tracking the object as a common practice while applying additional processing to infer the trajectory of a specific portion(s) of the object indirectly.

Two common tracking approaches, cross-correlation tracking and motion-blob proximity association tracking, are presented here to illustrate the fine differences between tracking a specific portion of the object and coarsely tracking the object. The cross-correlation method tracks a specific portion of the object, while motion-blob proximity association only tracks the object coarsely.

Let $I(i,j;t)$ be the pixel value of an image frame at position (i,j) and time t and $v(i_t,j_t)$, $t = t_0 \sim t_1$ be the resulting trajectory of a tracker on a vehicle. Here, t_0 and t_1 are the start and end of the tracking

of the vehicle, respectively. The first common tracking method to be described is cross-correlation matching approach. In this approach, first a region, $I_v(t_0) = I(i + i_{t_0}, j + j_{t_0}; t_0) \forall (i,j) \in R_v$, is identified and used as initial tracking template. In a typical setting, R_v is chosen as a rectangular region $(2m+1)\times(2n+1)$ centered around the centroid of the template (i_{t_0}, j_{t_0}). In the subsequent frames, the location that best matches the template $I_v(t_0)$ in the current frame $I(x, y; t)$ is then found using the following optimization criterion:

$$v(i_t, j_t) = \underset{(i_t, j_t)}{\operatorname{argmax}} \frac{\sum_{i=-m}^{m}\sum_{j=-n}^{n}\left(I(i+i_t, j+j_t; t)\cdot I(i+i_{t_0}, j+j_{t_0}; t_0)\right)}{\sqrt{\sum_{i=-m}^{m}\sum_{j=-n}^{n}\left(I(i+i_t, j+j_t; t)\right)^2}\sqrt{\sum_{i=-m}^{m}\sum_{j=-n}^{n}\left(I(i+i_{t_0}, j+j_{t_0}; t_0)\right)^2}} \tag{5.1}$$

Conceptually, Equation 5.1 is equivalent to finding the location in the current image frame where its appearance is closest to the tracking template. The resulting $v(i_t, j_t)$ is the new center position, where the image content at time t is most similar to the template at time t_0 measured by cross-correlation. The tracker would repeat this process for the time duration where the tracked vehicle is in the scene and eventually yields the full trajectory $v(i_t, j_t)$, $t = t_0 \sim t_1$ of the vehicle. Since this approach utilizes appearance matching, an appropriate selection of template would warrant the tracker to track a specific portion of the vehicle to determine its trajectory. An excellent example of such a template is a portion of the license plate of the vehicle.

The second common tracking method to be discussed uses foreground or motion detection followed by a proximity association. In this approach, potential pixels with motion are identified through frame differencing.

$$M(x,y;t) = \begin{cases} 1 & \text{if } \left|I(x,y;t) - I(x,y;t-\delta)\right| > \eta \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

The resulting binary image is then postprocessed with morphological filtering and connectivity analysis to extract motion blob(s), regions of potential moving objects. These candidate motion blobs may be further processed with thresholding on size, shape, aspect ratio, etc., to better detect blobs that are indeed from moving vehicles. Once the motion blob(s) are detected for the current frame, the tracker associates these blobs with those detected from past frame(s) based on measures such as proximity, smoothness of the overall trajectory, and coherence of the motions. The trajectory of the tracked vehicle would then be the collection of some reference points such as centroids of these associated blobs over time. Since the tracking of this approach is mainly based on motion blob rather than the appearance of object, it only tracks the object coarsely.

The schematic illustrations of the two tracking methods discussed earlier are shown in Figure 5.1 to help readers comprehend the fine differences. Figure 5.1a shows the first frame where the tracking starts. The dashed square is the template used by cross-correlation tracking to start the tracking, that is, $I_v(t_0)$. The triangle mark is the centroid of the template. The black outlined blob is the detected motion blob by motion-blob proximity association tracking. The solid-circle mark is the centroid of the motion blob. We set solid-circle and solid-triangle marks at the same location for illustration purpose. Note that the shadow of the vehicle was detected as part of the vehicle. Figure 5.1b shows the later frame where the tracking ends. The location of dashed square on this frame is found using Equation 5.1, and the solid-triangle mark is the centroid. As can be seen in Figure 5.1b, it is on the same location of the vehicle as before. The solid-circle mark in Figure 5.1b is the centroid of the

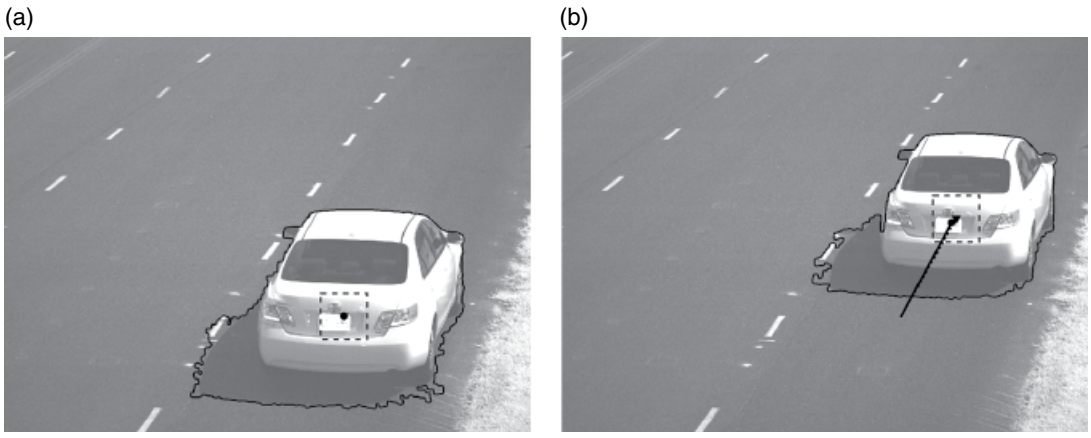

Figure 5.1 Schematic illustrations of cross-correlation and motion-blob proximity association tracking methods.

new black outlined blob, which is NOT on the same location of the vehicle as before. This is because the motion blob changes its shape a little bit due to noise and projective distortion. As a result, the pixel/distance traveled (solid and dash lines in Figure 5.1b) are slightly different by the two tracking methods (~5% difference). This difference is significant for the purpose of speed enforcement. The cross-correlation tracking method is preferred as discussed earlier.

In addition to accurate detection and tracking of vehicles, a computer vision–based speed measurement system requires (i) an accurate camera calibration strategy that produces a geometric mapping for translating image pixel positions to real-world coordinates [5–14], (ii) an understanding of the impact of tracked-feature height to speed accuracy [8, 14–16], and (iii) an accurate reference measurement system [17]. The geometric mapping is typically performed using a projective matrix transformation. More detailed discussions on these three requirements are presented in the following text.

Consider the work presented in Ref. [13], which introduces both the approach and potential pitfalls associated with *manual calibration methods*. In this work, the calibration is achieved by manually placing marks on the roadway, identifying image pixel locations that contain the marks, and then using the pixel location and mark location data to construct the camera calibration mapping. A couple of issues can arise with this approach. One consideration is that manually placing marks on the road may be impractical or costly, especially in high traffic areas. Second, both the placement and the identification of the location of the marks on the road need to be quite accurate. A systematic 10 cm combined error in the mark placement and pixel location for a 10 m spacing between marks would translate to a 1% bias error in subsequent speed measurements. Finally, the camera may move or change field of view over time (intentionally or unintentionally). Hence, camera recalibration may be needed periodically. Given these issues and constraints in *manual calibration methods*, it is preferred to use *model-based techniques*, which decompose entries of a projective matrix into functions of a set of parameters such as focal length, camera pose, etc., and estimate the parameters via scene analyses rather than the matrix entries directly from manually placed marks.

In camera calibration, the geometric mapping between pixel coordinates (i,j) to real-world planar coordinates $(x,y;z=z_0)$ can be characterized by a projective matrix as follows:

$$k\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix}. \tag{5.3}$$

The 3×3 $\boldsymbol{H}(z_0)$ matrix is known as the camera projective matrix for $z = z_0$. Different z would have a different projective matrix, but they are related across z's. If real-world coordinates are chosen such that its z-axis aligns with the camera optical axis, then the projective matrices for all z's can be described as follows:

$$k\begin{bmatrix} x/z \\ y/z \\ 1 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix}\begin{bmatrix} i \\ j \\ 1 \end{bmatrix}. \tag{5.4}$$

Here, the projective matrix is the same for all z's. This is often not the case for speed enforcement applications since the plane of interest is the road surface, which is roughly on a 2D plane, but its normal direction rarely aligns with the camera optical axis. There is a conversion that one can use to relate the two coordinate systems. For a monocular camera, it is necessary to comprehend the z-position or depth position to have unique conversion between pixel coordinate and real-world coordinate. The task of camera calibration/characterization is now reduced to identifying $\boldsymbol{H}(z_0)$ for a given camera. A straightforward approach is to manually place reference markers on the plane $z = z_0$ whose (x, y) are known and whose (i, j) can be identified from the images. Due to noise and potential errors in (x, y) or (i, j) of the reference markers, a common practice is to use many more reference markers than the minimal number needed (=4) and the random sample consensus (RANSAC) process to derive a robust and accurate $\boldsymbol{H}(z_0)$. This type of approach is referred to as *manual calibration*, which directly estimates the projective matrix based on reference data without setting any constraint on the relationship among entries in the projective matrix.

A *model-based camera calibration method*, on the other hand, imposes meaningful structure/constraint to the projective matrix and derives the corresponding parameters via scene analysis. Figure 5.2 shows an example camera–road configuration discussed in Ref. [5] for deriving a model-based camera calibration method. It utilizes the identification of vanishing points along and perpendicular to the road travel direction and one additional piece of information, such as the height of the camera above the road. This method is briefly summarized here. Let $f, h, \phi,$ and θ be the camera focal length, camera height above the road, camera tilt angle, and camera pan angle, respectively. Assuming

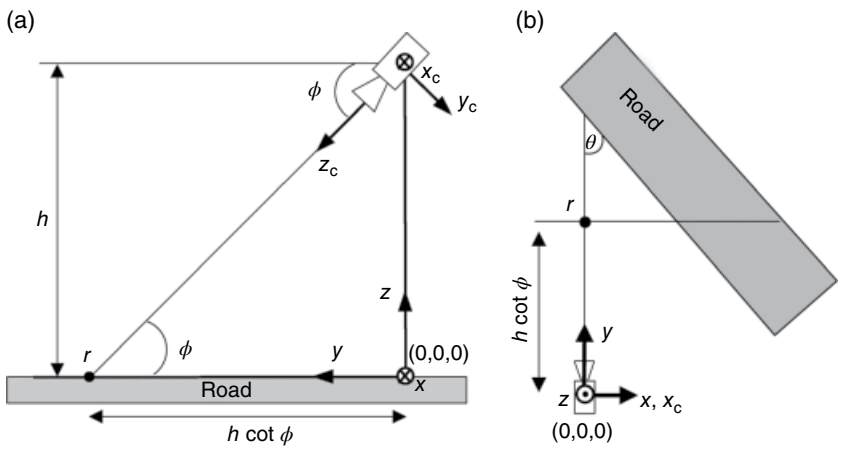

Figure 5.2 Illustration example for *model-based camera calibration method*: (a) left-side view of the scene and (b) top view of the scene. *Source*: Kanhere and Birchfield [5]. Reproduced with permission of IEEE.

that the camera has zero roll and has square pixels, it can be shown that the geometric mapping between pixel coordinates (i,j) to real-world coordinates on the road plane is

$$k\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h\sec\phi & 0 & 0 \\ 0 & -h\tan\phi & fh \\ 0 & 1 & f\tan\phi \end{bmatrix}\begin{bmatrix} i \\ j \\ 1 \end{bmatrix}. \tag{5.5}$$

The form is the same as Equation 5.3, but the entries of the projective matrix are more constrained. Although the matrix can be determined from actual measurements of f, h, ϕ, in practice it is difficult and inconvenient to measure these parameters directly. In Ref. [5], various known methods that utilize scene analysis to estimate these parameters indirectly are discussed and summarized. For example, in roadway transportation applications, it is feasible to determine a vanishing point (i_0, j_0) along the road travel direction and another vanishing point (i_1, j_1) perpendicular to the road travel direction from analyzing the image frame(s). The physical parameters f, ϕ, θ can then be determined by

$$f = \sqrt{-\left(j_0^2 + i_0 i_1\right)}, \phi = \tan^{-1}\left(\frac{-j_0}{f}\right), \theta = \tan^{-1}\left(\frac{-i_0\cos\phi}{f}\right) \tag{5.6}$$

If the height of the camera is also known, then one can recover the projective matrix Equation 5.5 using the detected pixel locations of the two vanishing points from scene analysis and Equation 5.6. The approach briefly reviewed here is only one of the many known model-based calibration methods. Note that there are several advantages to using model-based calibration methods over using manual approaches. First, model-based methods are more robust since fewer parameters need to be estimated. There is no need for any placement of reference marks on the road. The calibration can be easily updated and refined as the camera setting is changed or drifted over time since the parameters are derived through the scene analysis. In many road-side settings, the camera height is fixed but may be paned, zoomed, or tilted over time. The method of Equations 5.5 and 5.6 fits well for these scenarios.

Next, we review a few examples of model-based calibration methods from the perspective of impact on aspects of speed measurement, including accuracy. First, we discuss traffic-flow vision applications, where the goal is measurement of *average speed* and vehicle counting rather than law enforcement. In Refs. [6–12], the approaches taken focus on the use of vanishing points and/or heuristic knowledge for deriving the projective matrix transform. The vanishing point(s) are identified directly from the scene. Hence, they can be automatically updated as the scene changes, for example after pan, zoom, or tilt (PZT) operations. Furthermore, scene changes can be detected by analyzing the motion activity within the scene [7], which makes the calibration steps fully automated and dynamic. More specifically, in Ref. [6] the heuristic knowledge used includes a scale factor that varies linearly as a function of the traveling direction, which reduces the problem to a single dimension with known vehicle length distributions. The use of a known vehicle length distribution yields reasonable accuracy for average speeds over 20 s intervals (4% difference from inductive loop methods); however, the accuracy of individual vehicle speed estimates is quite poor. In Ref. [6], it is noted that the effect of shadows on centroid tracking is the main contributor for inaccuracies larger than 10%. In Ref. [8], lane boundaries and vanishing points are detected using a motion activity map. The histogram of average speed over 20 s intervals shows a bias of 4–8 mph compared to inductive loop measurements.

Note that unlike Ref. [6], where blob centroids are used for speed estimation, the pixel at the lowest row of the vehicle blob is used for speed calculation in Ref. [8]. Recently, Ref. [9] proposes a practical traffic camera calibration method from moving objects. This approach also follows the general concept of model-based camera calibration through the identification of vanishing points from the scene. The key difference of their approach comes from how the vanishing points are identified. None of the prior methods use the expected general orientations of moving vehicles or humans for identification of these vanishing points. The clear advantage of Ref. [9] is the sheer volume of available data since there are many more vehicles moving on the road than lane marks. However, there are also more challenges in order to get good quality data. Overall, Ref. [9] achieves good accuracy in a variety of traffic scenarios but no reported accuracy on how it translates to speed measurement accuracy.

Law enforcement is primarily concerned with the *speed of individual vehicles*, and here accuracy of the measurement becomes a critical concern. Accuracy requirements can be as tight as ±1 mph or ±1%. In Ref. [11], vanishing points and the assumption that the mean vehicle width is 14 ft are used to construct a camera calibration and resulting projective matrix transform. The reported inaccuracy of the estimated speed of an individual vehicle is below 10%, a figure somewhat below that achieved when lane boundaries are used for camera calibration [8]. Note that the improvement in speed estimation accuracy may not necessarily be due to differences in the calibration procedure; rather, it may be due to the use of a vehicle tracking method that is insensitive to shadows. In Ref. [12], the vanishing point is first detected from the road edges of the scene. The camera calibration mapping is then derived in a manner similar to the methods discussed earlier. The reported inaccuracy of the average speed of three test vehicles with 10 runs each is 4%. In Refs. [13–15, 18], the camera calibrations are all performed based on the known real-world coordinates of some form of landmarks. The reported inaccuracy of the speed estimates for individual vehicles ranges from 1.7, ±3, to ±5 km/h for five tested cars with speeds ranging from 13 to 25 km/h.

Consider an accuracy issue related to the height of a vehicle image feature being tracked and the dimensionality of the image acquisition scenario. As shown in Figure 5.3, a camera views a vehicle from an angle, and a tracking algorithm tracks one or more features (e.g., feature *A* and *B*) in the

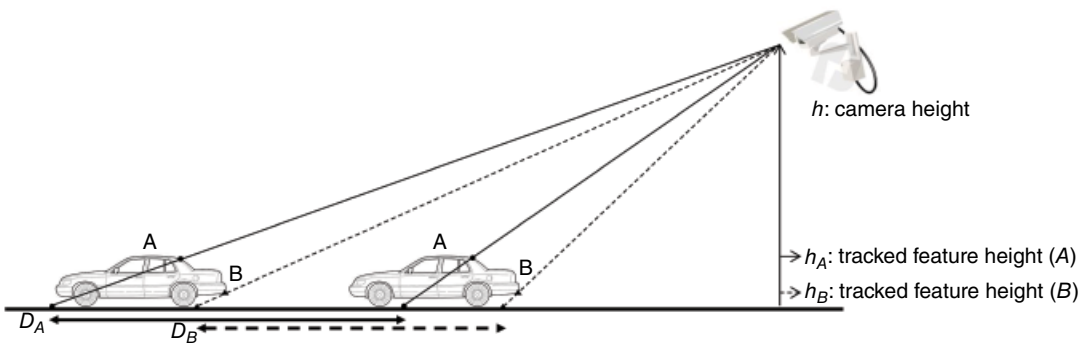

Figure 5.3 Illustration of an accuracy issue related to tracked vehicle image feature height and the dimensionality of image acquisition.

acquired images. Speed on the road surface is the desired measure, while the feature being tracked is generally above the road at an unknown height. It is usually not possible to determine the height of the tracked feature because a single camera image is a 2D representation of a 3D phenomenon, which introduces mapping ambiguities. The calibration of 2D pixel locations to road locations assumes a given feature height, such as the road surface. Speed measurement based on tracked features at other heights will be inaccurate due to the discrepancy between assumed and real feature heights. The issue is less severe if vehicle speeds are calculated based on features that are the lowest edges or points of a motion blob (closest to the ground plane) [8, 14–16] rather than the centroids of a motion blob. Alternatively, if the height of the tracked feature is known or can be estimated, then the measured speed can be height corrected. Using the example in Figure 5.3, it can be shown that $D_A = \frac{h-h_B}{h-h_A} D_B$. This implies that the actual ground speed can be corrected from the tracked speed of the feature (A) with the factor $\left(1-\frac{h_A}{h}\right)$. The higher the tracked feature, the more the reduction needed. The higher the camera height, the less the tracked height impacts the accuracy of the speed estimation without correction. As discussed in Section 5.2.2, the height of the tracked feature can be estimated through stereo imaging, which solves the dimensionality problem described here.

Finally, a typical accuracy requirement for speed enforcement systems can be as tight as ±1 mph or ±1%. It is thus necessary to have an accurate reference measurement system that is at least an order of magnitude more accurate and precise. An example of research on this topic is found in Ref. [17].

5.2.2 Speed Estimation from Stereo Cameras

Speed estimation from monocular imagery presents specific challenges that may be difficult to overcome. As discussed, vehicle detection is a key step in the process; methods for moving object detection from monocular video include foreground detection via background estimation and subtraction, temporal frame differencing, and flow analysis. Shortcomings associated with all three approaches include the "black hole" problem whereby it becomes difficult to detect an object in motion whose appearance closely matches that of the background; additionally, when the moving object casts a shadow, the shadow can be detected as part of the moving object itself, which may pose difficulties for subsequent tracking tasks. Even when object detection issues are overcome, ambiguities arising from the projection of the 3D world onto a 2D plane result in increased speed estimate uncertainties as described before.

Speed estimation based on binocular, depth-capable systems has been developed and deployed to overcome these limitations. Depth-capable systems rely on the use of two or more calibrated cameras with overlapping fields of view acquiring simultaneous images of the monitored scene. As illustrated in Figure 5.4, the speed estimation process from a stereo camera is similar to that from a monocular

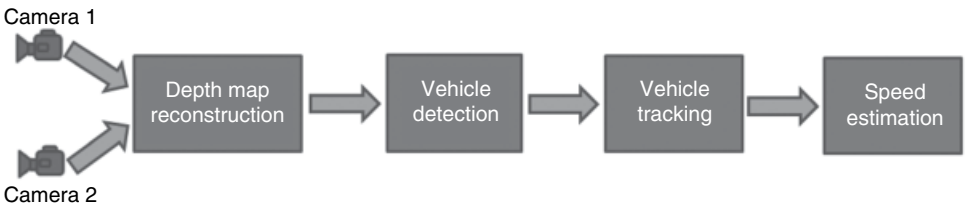

Figure 5.4 Speed estimation process in a stereo vision system.

camera, except that imagery from two cameras is processed jointly to obtain information of the scene including, but not limited to, the distance or range between the system and a feature in the scene, as well as real-world coordinates of objects in the scene. With the forthcoming discussion, it will become clear that the advantages brought about by depth-capable systems based on binocular imaging come at the cost of tighter design requirements. In particular, the accuracy of depth estimates depends on, among other factors, the placement of the system relative to the road being monitored, the resolution of the sensors in each of the monocular cameras in the system, and on the relative positioning between the cameras and the disparity between the assumed and real relative positioning. For instance, the monocular cameras comprising the system in Ref. [19] are located 100 cm from each other, have a 1.4 megapixel sensor each, and are housed in a structure made of carbon fiber, alloy, and polycarbonate to ensure high rigidity (torsion is rated at under 6.4 μrad) ensuring the relative positioning of the cameras is maintained under the most extreme environmental forces. The system is carefully calibrated at the factory, and error-checking routines are performed continuously in order to verify that the mutual geometric constraints between the monocular cameras are maintained so as to guarantee high precision depth (depth estimation errors are at most 3 cm) and speed (average speed estimation errors are under 1% in the 20–240 km/h speed range) estimates. The system is placed on a vertical pole mount at a slight angle from traffic, 15–25 m behind or ahead of the highway area being monitored.

It will also be appreciated with Sections 5.2.2.1–5.2.2.4 that the computational requirements of a depth-capable system are more demanding than those imposed by a monocular system, not only because two video streams are processed simultaneously but also because joint processing of the streams is required in order to extract depth estimates. This means that there is an additional processing overhead on top of the already doubled computational requirements, which places stricter constraints on the computational resources of the system, in particular when real-time monitoring of potentially busy multilane highways is desired. Ultimately, the ability to estimate the depth of objects in the scene leads to improved robustness to environmental and traffic conditions as well as increased capabilities to extract accurate vehicle trajectories, which in turn translates into added extensibility to other traffic enforcement-related applications such as vehicle classification, RLE, and forbidden turn monitoring.

5.2.2.1 Depth Estimation in Binocular Camera Systems

In the simplest scenario, the depth capable systems consist of two cameras, each defining coordinate systems x_1, y_1, z_1 and x_2, y_2, z_2 with origins $\boldsymbol{O}_1$ and $\boldsymbol{O}_2$, respectively [20]. Let us (approximately) model the cameras as pinhole cameras with each optical axis coinciding with their respective z-axis. Also, let the cameras have focal points located at $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ at distances f_1 and f_2 from their respective origins along their respective z-axis, as illustrated in Figure 5.5. Image planes π_1 and π_2 lie on planes x_1y_1 and x_2z_2, respectively.

Coordinates between the two coordinate systems are related by

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \boldsymbol{T}, \tag{5.7}$$

where $\boldsymbol{R}$ is a 3×3 orthonormal matrix describing a rotation and $\boldsymbol{T}$ is a 3×1 vector representing translation. Since the cameras have overlapping fields of view, assume both cameras are imaging an object point $\boldsymbol{P}_0$ located at $[x_{p0}, y_{p0}, z_{p0}]^{\mathrm{T}}$ relative to $\boldsymbol{O}_1$ and $[x'_{p0}, y'_{p0}, z'_{p0}]^{\mathrm{T}}$ relative to $\boldsymbol{O}_2$ in the 3D space that projects onto point $\boldsymbol{P}_1$ on image plane π_1 and onto point $\boldsymbol{P}_2$ on image plane π_2. Note that $[x_{p0}, y_{p0}, z_{p0}]^{\mathrm{T}}$

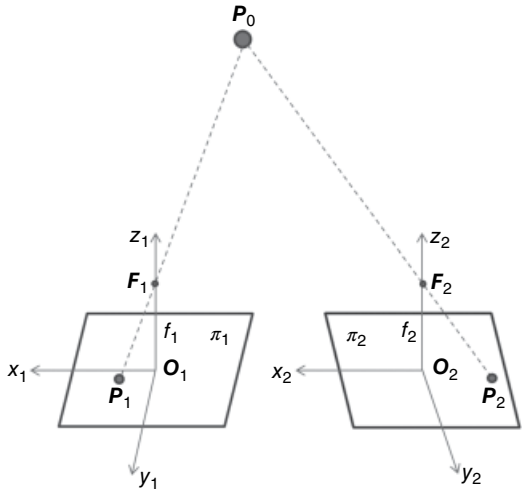

Figure 5.5 Depth capable system comprising two cameras.

and $[x'_{p0}, y'_{p0}, z'_{p0}]^T$ are related by Equation 5.7. The coordinates of $\boldsymbol{P}_1$ relative to $\boldsymbol{O}_1$ are $\boldsymbol{P}_1 = [x_{p1}, y_{p1}, 0]^T$, and the coordinates of $\boldsymbol{P}_2$ relative to $\boldsymbol{O}_2$ are $\boldsymbol{P}_2 = [x_{p2}, y_{p2}, 0]^T$ given that both $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are on their respective image planes. Since $\boldsymbol{P}_0$, $\boldsymbol{F}_1$, and $\boldsymbol{P}_1$ are collinear, and $\boldsymbol{P}_0$, $\boldsymbol{F}_2$, and $\boldsymbol{P}_2$ are collinear as well, the relationships $\boldsymbol{P}_1 = \boldsymbol{F}_1 + \alpha_1(\boldsymbol{P}_0 - \boldsymbol{F}_1)$ and $\boldsymbol{P}_2 = \boldsymbol{F}_2 + \alpha_2(\boldsymbol{P}_0 - \boldsymbol{F}_2)$ hold for some scalars α_1 and α_2. Expanding each of these equations into their scalar forms, we get

$$\frac{x_{p_0}}{x_{p_1}} = \frac{y_{p_0}}{y_{p_1}} = \frac{f_1 - z_{p_0}}{f_1} \tag{5.8}$$

$$\frac{x'_{p_0}}{x_{p_2}} = \frac{y'_{p_0}}{y_{p_2}} = \frac{f_2 - z'_{p_0}}{f_2} \tag{5.9}$$

Assuming the focal lengths of the cameras, the rotation matrix $\boldsymbol{R}$, and the translation vector $\boldsymbol{T}$, $\boldsymbol{P}_1$, and $\boldsymbol{P}_2$ are known, Equations 5.7, 5.8, and 5.9 can be used to compute $\boldsymbol{P}_0$ in the 3D world. Note that this assumes a correspondence between $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ has been established. A depth map of the scene can be reconstructed by combining 3D coordinates of points for which correspondences are found. At other locations, depth information can be inferred via interpolation techniques.

Given two or more images acquired with two or more cameras with at least partially overlapping fields of view, the correspondence problem refers to the task of finding pairs of points (one in each image) in the images that can be identified as being projections of the same points in the scene. Correspondences can be found via local correlations, or via feature extraction and matching. Assume that the location of an image point $\boldsymbol{P}_1$ on image plane π_1 is known and that we need to find the location of the corresponding point $\boldsymbol{P}_2$. Instead of searching across the whole image plane π_2 for corresponding point $\boldsymbol{P}_2$, the size of the search space can be significantly reduced by noting that $\boldsymbol{P}_2$ must be located along the line segment defined by the intersection between π_2 and the plane defined by the points $\boldsymbol{P}_1$, $\boldsymbol{F}_1$, and $\boldsymbol{F}_2$. This is referred to as the epipolar constraint.

5.2.2.2 Vehicle Detection from Sequences of Depth Maps

Once a sequence of depth maps of the scene is available, vehicles traversing the scene can be detected with a high degree of confidence. As mentioned earlier, object detection in monocular video sequences is usually addressed by detecting differences in appearance (e.g., intensity or color) between the foreground

object and a background model, which may lead to erroneous detections when the object's appearance closely matches that of the background. In the case of a stereo vision system, in addition to scene appearance, the depth of various points in the scene relative to the camera system is known. Since foreground objects are usually located between the background and the camera system, more robust foreground object segmentation can be achieved by augmenting background appearance models with range data.

More specifically, consider modeling the attributes of the background via a statistical model such as a distribution comprising a mixture of K Gaussian components, as proposed in Ref. [21]. According to this approach, a historical statistical model for each pixel is constructed and updated continuously with each incoming frame at a predetermined learning rate. Foreground detection is performed by determining a measure of fit of each pixel value in the incoming frame relative to its constructed statistical model: pixels that do not fit their corresponding background model are considered foreground pixels. Instead of modeling attributes related to the appearance of the background only, the statistical models can be seamlessly extended to capture the behavior of the background in terms of its depth relative to the scene, as proposed in Ref. [22]. This process is illustrated in Figure 5.6. Let $F_{i,j}$ denote the ith video frame, acquired by camera j, where i represents a discrete temporal index, and $j \in \{1, 2\}$. Let F_i denote the augmented ith video frame including color and depth information obtained by fusing frames $F_{i,1}$ and $F_{i,2}$. F_i will generally be smaller in spatial support, but possibly larger in terms of number of channels/bits per channel than $F_{i,1}$ and $F_{i,2}$ since depth information can only be extracted from scene locations that fall in the common field of view of the cameras. Background estimation is achieved by estimating the parameters of the distributions that describe the historical behavior in terms of color and depth values for every pixel in the scene as represented by the augmented video frames F_i. Specifically, at frame i, what is known about a particular pixel located at (x,y) in F_i is the history of its values$\{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_i\} = \{F_m(x,y),\ 1 \le m \le i\}$, where $\boldsymbol{X}_m$ is a vector containing color and depth values, that is $\boldsymbol{X}_m = [R_{Xm}, G_{Xm}, B_{Xm}, d_{Xm}]$ in the case of RGB cameras. The value of

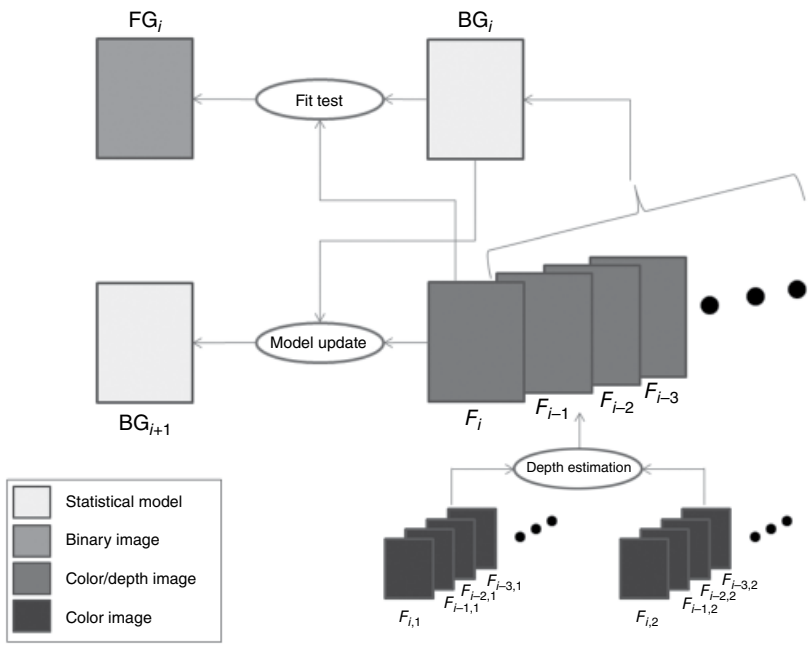

Figure 5.6 Foreground detection based on augmented background modeling.

d_{Xm} can be estimated via Equations 5.7, 5.8, and 5.9, while the values of R_{Xm}, G_{Xm}, and B_{Xm} can be estimated, for example, by averaging the RGB values of the corresponding pixels used to estimate d_{Xm}. For pixels for which a correspondence wasn't found, but which are still within the reduced field of view associated with F_i, RGB and depth values can be estimated via interpolation.

The recent history of behavior of values of each pixel can be modeled as a distribution consisting of a mixture of K Gaussian components, so that the probability of observing the current value is

$$P(\boldsymbol{X}_m) = \sum_{k=1}^{K} w_{km} \Phi(\boldsymbol{X}_m, \boldsymbol{\mu}_{km}, \boldsymbol{\Sigma}_{km}), \tag{5.10}$$

where w_{km} is an estimate of the weight of the kth Gaussian component in the mixture at time m, $\boldsymbol{\mu}_{km}$ is the mean value of the kth Gaussian component in the mixture at time m, $\boldsymbol{\Sigma}_{km}$ is the covariance matrix of the kth Gaussian component in the mixture at time m, and $\Phi(\cdot)$ is the Gaussian probability density function. Sometimes, a reasonable assumption is for the different pixel attributes to be uncorrelated, in which case $\boldsymbol{\Sigma}_{km} = \sigma_{km}\boldsymbol{I}$, for a set of scalars σ_{km}, $1 \le k \le K$, where $\boldsymbol{I}$ is the $K \times K$ identity matrix. Let BG_i denote the ith augmented background model, that is, an array of pixel-wise statistical models of the form of Equation 5.10 including color and depth information. Then FG_i, the ith foreground binary mask indicating the pixel locations associated with a detected foreground object can be obtained via comparison between BG_i and F_i. Foreground detection is performed by determining a measure of fit of each pixel value in the incoming augmented frame F_i relative to its constructed statistical model which is stored in BG_i and has the form of Equation 5.10. In the simplest implementation, as a new frame comes in, every pixel value in the augmented frame F_i is checked against its respective mixture model so that a pixel is deemed to be a background pixel if it is located within T_1 standard deviations of the mean of any of the $K-1$ color components, and its estimated depth is within T_2 standard deviations of the depth component with the largest mean, where T_1 and T_2 are predetermined thresholds. The former condition checks for consistency in the color appearance between the current pixel and the background model, and the latter verifies that the objects in the current frame are as far as possible from the camera system, relative to what has been observed in the past.

Since scenes are usually dynamic and in order to maintain accurate foreground detection, the background model BG_i can be updated to obtain a background model BG_{i+1} after incoming augmented frame F_i is processed. Note that FG_{i+1} will subsequently be determined via comparison between BG_{i+1} and augmented frame F_{i+1}. If the current pixel value is found not to match any of the K components according to the fit test described before, the pixel is identified as a foreground pixel; furthermore, the least probable component in the mixture is replaced with a new component with mean equal to the incoming pixel value, some arbitrarily high variance, and a small weighting factor. If, on the other hand, at least one of the distribution components matches the incoming pixel value, the model is updated using the value of the incoming pixel value. Updating of the model is achieved by adjusting the weight parameters for all components (i.e., for all k, $1 \le k \le K$) in Equation 5.10 according to

$$w_{k(i+1)} = (1-\alpha) w_{ki} + \alpha M_{ki}, \tag{5.11}$$

where α is the learning rate, M_{ki} is an indicator variable equaling 0 for every component except the matching one, in which case $M_{ki} = 1$. After applying Equation 5.11 to update the mixture weights, the weights are renormalized to sum up to 1. Once the weights are updated, the parameters of the distribution k_0 that was found to match the new observation are updated according to

$$\mu_{k_0(i+1)} = (1-\rho)\mu_{k_0 i} + \rho \boldsymbol{X}_i \tag{5.12}$$

$$\sigma^2_{k_0(i+1)} = (1-\rho)\sigma^2_{k_0 i} + \rho\left(\boldsymbol{X}_i - \mu_{k_0(i+1)}\right)^T \left(\boldsymbol{X}_i - \mu_{k_0(i+1)}\right) \tag{5.13}$$

where $\boldsymbol{X}_i$ is the value of the incoming pixel and $\rho = \alpha\Phi\left(\boldsymbol{X}_i \mid \mu_{k_0 i}, \sigma^2_{k_0 i}\right)$. The parameters of the components that didn't match are left unchanged.

5.2.2.3 Vehicle Tracking from Sequences of Depth Maps

Tracking can be performed once the vehicle in motion is detected. Most of the existing video-based object trackers are optimized for monocular imagery. Recall that one of the requirements for the output of a tracking algorithm to provide robust speed-related information is for it to convey positional data of highly localized portions of the vehicle being tracked. In view of this requirement, and given the fact that salient scene points are extracted when solving the correspondence problem, the authors of Ref. [23] proposed a method for efficiently tracking features extracted for stereo reconstruction purposes. Assuming the stereo system cameras satisfy the epipolar constraint, the displacement of corresponding features across frames can be represented via a 3D vector. For example, in the case where the cameras are vertically aligned, feature displacement across video frames can be uniquely represented by two horizontal offsets, one for each camera, and a shared vertical offset, $\boldsymbol{D} = (d_1, d_2, d_y)$. This assumption achieves a reduction of one degree of freedom relative to the more general approach of independent tracking of the features in their respective video feeds. Let us assume features $T_1(\boldsymbol{x}_1)$ and $T_2(\boldsymbol{x}_2)$ from windows centered at locations $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ of the point being tracked are extracted from frames acquired by cameras 1 and 2, respectively, for frame i, as illustrated in Figure 5.7. For simplicity, assume that the extracted features are pixel values within the window.

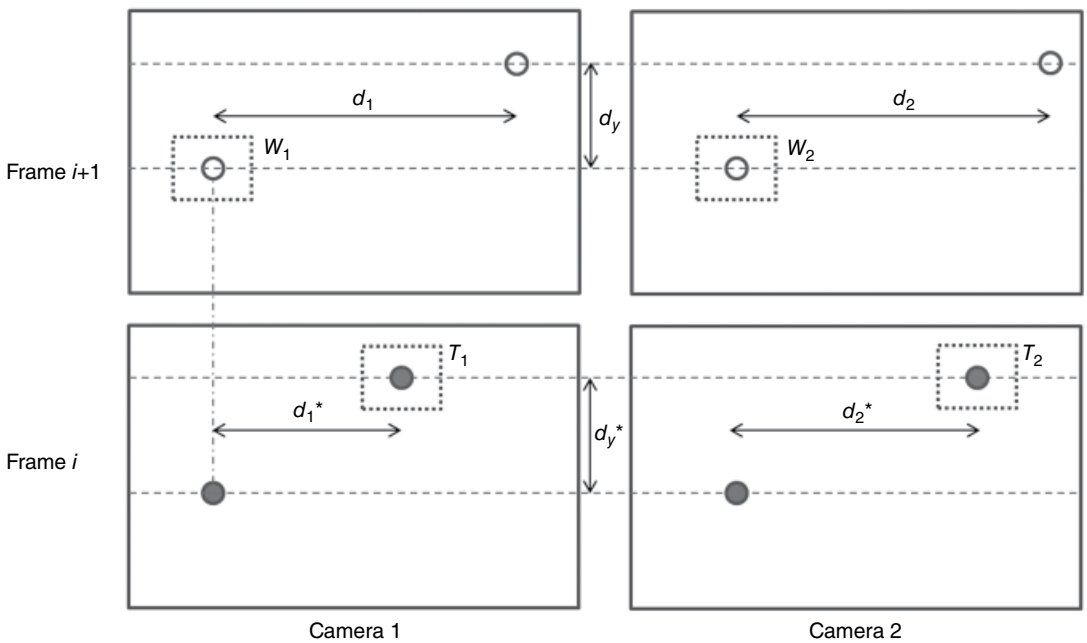

Figure 5.7 Point tracking in the stereo domain.

In the scenario considered by Lucas and Kanade [24], the task of tracking the point consists of finding its displacement between frames i and $i+1$, which can be expressed as finding warping parameters $\boldsymbol{p}^*$ that satisfy

$$\boldsymbol{p}^* = \text{argmin}_{\boldsymbol{p}} \sum_{j=1}^{2} \sum_{x} \left[I\left(W_j\left(\boldsymbol{x}_j;\boldsymbol{p}\right)\right) - T_j\left(\boldsymbol{x}_j\right)\right]^2, \tag{5.14}$$

where $I(W_j(\boldsymbol{x};\boldsymbol{p}))$ is an image resulting from warping frame $i+1$ acquired by camera j at location $\boldsymbol{x}$ according to warping parameters $\boldsymbol{p}$. For simplicity, assume that the frame rate of the acquired video is large enough relative to the apparent motion of the object so that $\boldsymbol{p}$ can be well approximated by displacements (d_j, d_y) for $j = 1, 2$. Equation 5.14 then becomes

$$\left(d_1^*, d_2^*, d_y^*\right) = \text{argmin}_{(d_1, d_2, d_y)} \sum_{j=1}^{2} \sum_{x} \left[I\left(W_j\left(\boldsymbol{x}_j;\left(d_j, d_y\right)\right)\right) - T_j\left(\boldsymbol{x}_j\right)\right]^2, \tag{5.15}$$

where $I(W_j(\boldsymbol{x}_j;(d_j, d_y)))$ is the image window displaced by (d_j, d_y) from pixel location $\boldsymbol{x}_j$ in frame $i+1$ acquired by camera j, for $j = 1, 2$. The expression in Equation 5.15 can be solved iteratively by assuming a current estimate of (d_1, d_2, d_y) is available and solving for increments $(\Delta d_1, \Delta d_2, \Delta d_y)$ as demonstrated in Ref. [23].

5.2.2.4 Speed Estimation from Tracking Data

Tracking information for features extracted in and around the identified vehicle consists of a temporal sequence of 3D coordinates $\boldsymbol{P}_s^{(1)}, \boldsymbol{P}_s^{(2)}, \ldots, \boldsymbol{P}_s^{(i)}$, where $\boldsymbol{P}_s^{(k)}$ denotes the coordinates at frame k of feature s being tracked. Pair-wise instantaneous speed estimation computation for frame k is then performed by computing

$$v_s^{(k)} = r\left\|\boldsymbol{P}_s^{(k)} - \boldsymbol{P}_s^{(k-1)}\right\|, \tag{5.16}$$

where r is the frame rate of the image capture system.

There are multiple sources of error that affect the instantaneous speed estimation process described before. For one, the location of corresponding features is established relative to a discrete set of coordinates determined by the discrete camera sensors, which gives rise to quantization errors intrinsic to digital systems. It can be shown that quantization errors result in uncertainties on the estimated depth or range of the features for which a correspondence has been found, and that the magnitude of the depth estimation error increases as the depth of the feature relative to the sensor increases. Additionally, for fixed focal length and sensor resolution, the larger the separation between the cameras, the larger the range of depths supported, but the smaller the accuracy in the feature matching; this conflict gives rise to a trade-off between how accurately correspondences can be established and how accurately range estimation can be performed [25]. Clearly, the parameters of a stereo imaging system affect depth estimation capabilities, and consequently, the accuracy of the estimated speed. Careful selection of the system parameters is then required to achieve the required accuracy specifications.

In order to address errors associated with instantaneous speed measurements, Ref. [26] proposes acquiring a set of measurements at different times and performing regression on the multiple measurements. According to the proposed approach, linear regression is performed on a set of range or distance measurements; the slope of the resulting linear model indicates whether the vehicle is

approaching or receding from the camera system. A mean speed estimate can be obtained from the slope estimate. Additionally, a confidence level represented by the R^2 value of the regression process can be computed, which in turn reflects the confidence of the estimated speed. Estimates with confidence level below certain predetermined thresholds may be discarded.

5.2.3 Discussion

In summary, although individual vehicle speed is a straightforward output from most computer vision systems, there is an accuracy gap for single camera systems. While stereo cameras [19, 27] for photo enforcement are becoming widely available, there are very few scientific publications on calibration and practical accuracy of 3D systems. There are also potential issues with a lack of accurate reference measurements. In addition to 3D solutions, another common approach to photo enforcement of speed has been through use of radar/lidar for speed and a camera for vehicle identification and evidence recording [28].

5.3 Stop Violations

Driver violations at intersections are the major cause of fatal accidents in urban traffic. According to the US Department of Transportation (USDOT), nearly half of all traffic accidents and 20% of all fatal crashes occur in close proximity to an intersection [29]. This statistic has remained substantially unchanged since the past decade despite significant efforts by transportation agencies for improved intersection designs and sophisticated applications of transportation engineering [29]. To combat this persistent trend, municipalities and law enforcement agencies are employing automated stop enforcement technologies at red lights and stop signs, as well as for stopped school buses.

5.3.1 Red Light Cameras

The most common violation at intersections is running a red light. Several studies found in the literature show the prevalence and severity of stop light violations [30–32]. One study conducted in Arlington, Virginia, reported a red light violation occurs on an average of every 20 min at each intersection [33]. The rate of violations significantly increases at peak hours, and rises as high as one per 5 min, which in turn causes a high number of traffic accidents, with associated damaged property and lost lives [33].

Prevention of intersection-related crashes to improve public safety and reduce property damage has led to the adoption of two primary approaches [34], namely optimizing signal light timing and stop light enforcement via red light cameras (RLCs). Several studies claim that longer yellow light duration reduces red light running violations [35, 36], and increased yellow light duration with an all-red interval can reduce the number of accidents [37, 38]. The Institute of Transportation Engineers (ITE) has provided a standard for minimum yellow light duration Y, given by Equation 5.17 and computed in Table 5.1.

$$Y = t + \frac{1.47v}{2(a + Gg)}, \tag{5.17}$$

where

Y = length of yellow interval (s)
t = perception-reaction time (use 1 s)

Table 5.1 Minimum yellow light intervals given approach speeds for straight road and 0 grade.

Approach speed (mph)	Yellow interval (s)
25	3.0 (rounded up)
30	3.2
35	3.6
40	4.0
45	4.3
50	4.7
55	5.0
60	5.4
65	5.8

v = approach speed (mph)
a = deceleration rate in response to the onset of a yellow indication (use 10 ft/s^2)
g = acceleration due to gravity (32.2 ft/s^2)
G = fractional grade, with uphill positive and downhill negative

Most studies examined increasing Y by 0.5–1.5 s over the ITE minimum.

RLC systems, also known as RLE systems, are deployed extensively throughout the world. The efficiency of RLCs on reducing number of red light running has been reported in several studies [30, 32, 39, 40]. In Ref. [39], the efficiency of RLCs was evaluated in New York City, Polk County, Florida, Howard County, Maryland, and it is reported that enforcement cameras yielded 20% reduction in violations in New York City while showing promising results for the other municipalities. Similarly, Ref. [40] reports that RLCs reduce stop light violations on average up to 50% based on a study performed on international RLCs.

5.3.1.1 RLCs, Evidentiary Systems

Common RLC systems consist of the following three modules: (i) violation detection, (ii) evidentiary photo/video capturing, and (iii) control unit [41, 42]. Figure 5.8 shows an illustration of a typical RLC system. The violation detection module uses in-ground induction loops to estimate the speed of an approaching vehicle before the stop bar at the intersection. The speed estimate is used to conjecture whether the vehicle will run through the red light. The cameras are solely utilized to capture evidentiary images and videos, and vehicle identification information (i.e., license plate number) of violators is extracted from the evidentiary images by human operators [42].

The violation detection module relies on two magnetic induction loops buried under the pavement and in communication with the controller unit to estimate vehicle speed as it approaches the stop bar. The speed estimation can be as simple as dividing the distance between induction loops by the time difference that vehicle is detected by each loop. More complicated regression models can be also employed based on the road geometry and historical data. When a vehicle activates the first and second induction loops within a time threshold (e.g., estimated speed above threshold) when the light is red, then a violation signal is sent to trigger the evidentiary cameras. The cameras and control system record information about the violation event, such as date, time, estimated speed, license plate, and lane of violation. This auxiliary information can be combined with an image or images and used in a citation document.

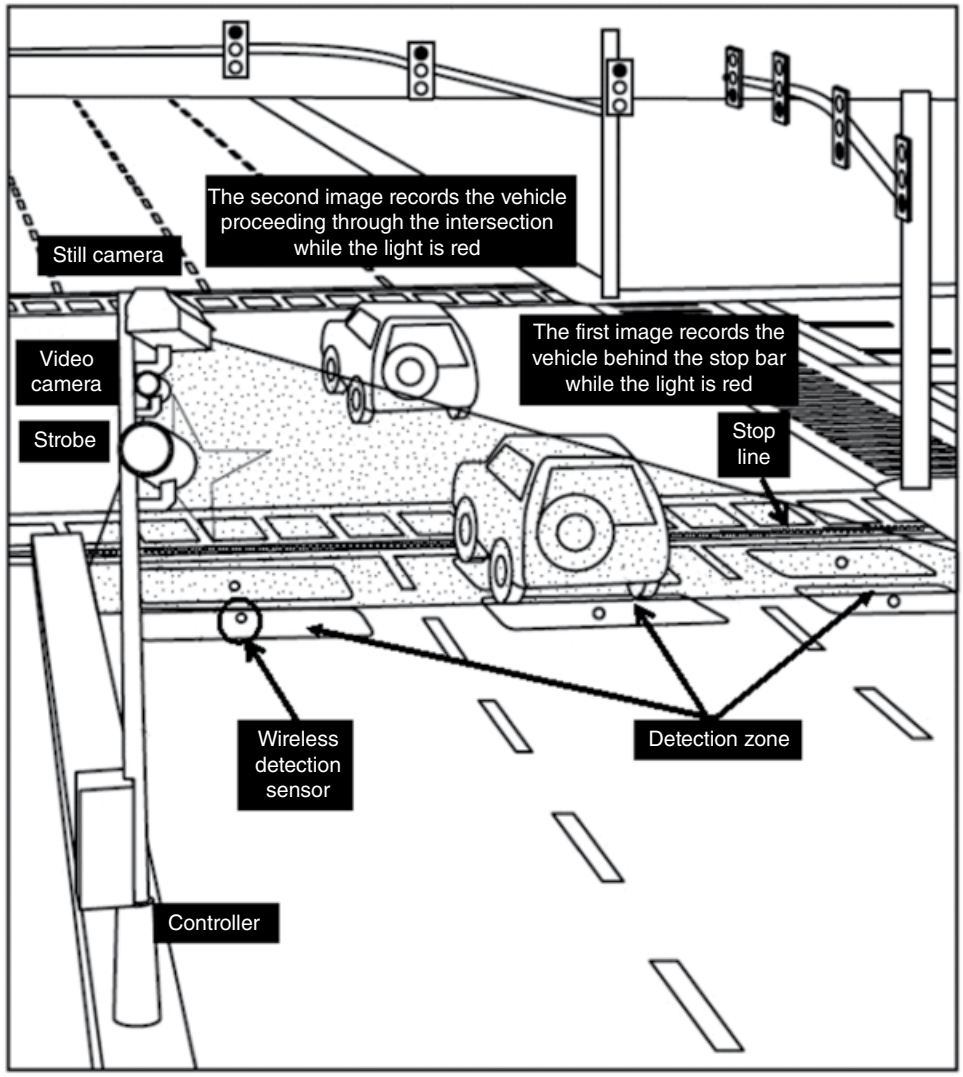

Figure 5.8 Red light enforcement system as described in Ref. [43]. *Source*: App. No 14/278196.

As Figure 5.8 shows, current RLE systems typically include two digital cameras. The first (i.e., upper camera in Figure 5.8) is a relatively high-resolution (e.g., 4080×2720 pixels and above) still image camera that takes two evidentiary pictures (Shot A and Shot B) intended to show a violator before the stop bar and at the intersection. The license plate of the violator is acquired from high-resolution evidentiary images captured by the first camera. The high-resolution cameras might also have NIR capabilities to enable night-time functionality with an external IR illuminator. NIR cameras, however, capture images in grayscale, which can be restrictive in some evidentiary settings. Therefore, in most RLC systems, RGB cameras are used with a visible light illuminator to provide night-time functionality. The illuminator is flashed behind the violator to view the rear license plate and to prevent disrupting the driver. The second camera (i.e., the lower camera in Figure 5.8) operates at lower resolution (e.g., 896×596 pixels) and captures a video of the incident given a trigger

from the violation detection module. The captured video along with the high-resolution evidentiary photos are transferred to a processing center for manual review. A citation ticket is mailed to the registered owner of the vehicle once a violation is manually reviewed.

5.3.1.2 RLC, Computer Vision Systems

Induction loops in RLC systems are typically installed at a distance from the intersection for safety reasons, so that traffic lights can be automatically switched based on the speed of approaching vehicles to prevent T-bone crashes [44]. The performance of the violation detection module is adversely impacted as the speed estimation is performed further away from the traffic lights, especially for vehicles abruptly stopping before reaching the intersection. In these cases, the violation detection module identifies the vehicle as a violator based on the measured speed of the vehicle before the stop bar, and triggers the enforcement camera. In one operating RLC system, for example, it is reported that only 20% of the detected cases were actual violators and the remainders were false positives detected by the violation detection module [43]. In current RLC implementations, the evidentiary photos and videos of potential violators are manually reviewed to avoid issuing improper citations to the false positives caused by vehicles making a sudden stop at the traffic light.

Most of the computer vision and video analytics algorithms in the literature focus on the violation detection module [43, 45–50]. The proposed computer vision and video analytics algorithms can be divided into two main groups. The methods in the first group propose techniques to complement and support existing RLC systems to reduce the need for the costly manual review process [43, 45]. These methods propose the use of postprocessing techniques on the evidentiary images/videos to reduce number of false positive that go to human review. Reference [43], for example, proposes an algorithm to reduce false detections of nonviolators using the two evidentiary images captured before the vehicle reaches the stop bar (e.g., Shot A) and when the vehicle is at the intersection (e.g., Shot B) as shown in Figure 5.9. A feature matching algorithm is applied between the Shot A and Shot B images using speeded up robust feature (SURF) points and a violation/nonviolation decision is made based on the attributes of the matched features. The decision is made based on a criterion of finding "coherent clusters of matched features" that comply with the attributes of the matched features on a violating vehicle as shown in Figure 5.10. More specifically, a violation is detected from the

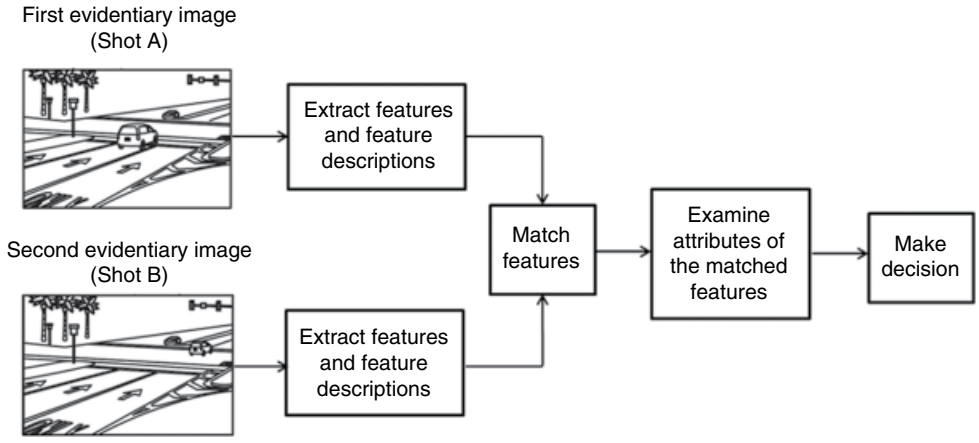

Figure 5.9 High-level overview of the red light violation detection system proposed in Ref. [43] from evidentiary photos. *Source*: App. No 14/278196.

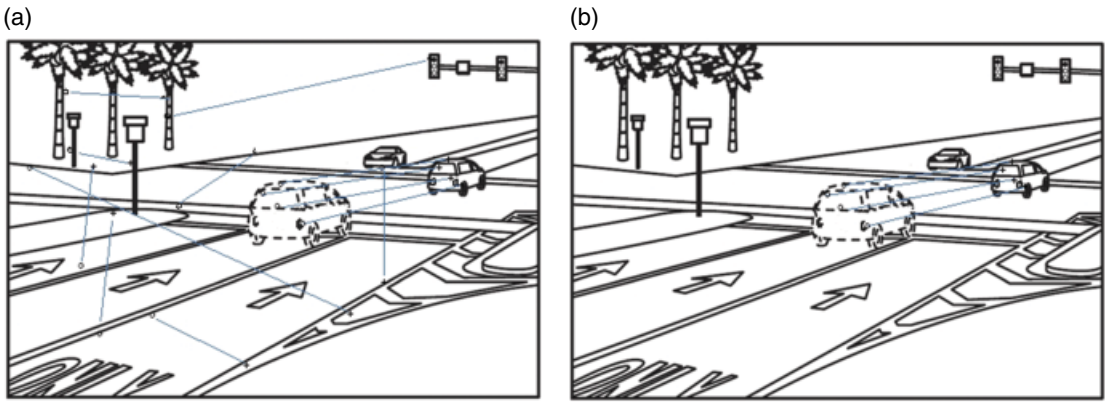

Figure 5.10 Feature matching in evidentiary red light images from Ref. [43]. Matched SURF features in Shot A and Shot B images (a), and coherent cluster of matched features after eliminating feature pairs irrelevant to red light violation (b). *Source*: App. No 14/278196.

evidentiary images if a coherent cluster of matched features is detected satisfying the following conditions:

- Length of lines connecting matched feature pairs within the cluster are longer than a specific threshold T.
- Angle lines connecting matched feature pairs within the cluster are within a specified angular interval around the road direction.
- Matched feature pairs within the coherent cluster should all start before the stop bar and end in the intersection.

Reference [45] proposes another computer vision method to support existing RLC systems by reducing false positives. The method analyses the evidentiary video clips to automatically identify and distinguish violators from nonviolators. The following steps are included in the method: (i) defining ROIs in the video (i.e., virtual detection zones before traffic lights for vehicle detection); (ii) detecting the state of traffic lights; (iii) when a vehicle is detected in the ROI and the traffic light is red, extracting a set of attributes/features from the vehicle and tracking it across frames; and (iv) processing and analyzing the trajectories of the tracked vehicles to identify a violation. When coupled with the method in Ref. [43], it is reported that a significant reduction can be achieved in automatically reducing false positives of current RLC systems [45].

The second group of computer vision methods eliminate the need for induction loops by using video cameras to identify violations in real time [46–52]. Video-based violation detection provides a nonobtrusive approach that can reduce construction and maintenance costs associated with the installation of induction loops in the pavement. Figure 5.11 illustrates an overall block diagram of the video-based red light violation detection system proposed in Ref. [48]. The system consists of two main modules: (i) traffic light sequence detection (TLSD) module that determines the spatial coordinates of the lights and estimates light sequence from video, and (ii) vehicle motion estimation (VME) module to detect a red light running violation. The position of traffic lights in a video can be determined by performing edge detection and finding the closed loops corresponding to the shape of the traffic signals within the edge map. This, however, is a challenging task given that outdoor scenes are uncontrolled and can be quite complex. A better way to find the position of traffic lights is based on detection of colored regions in video. This detection requires multiple frames as different

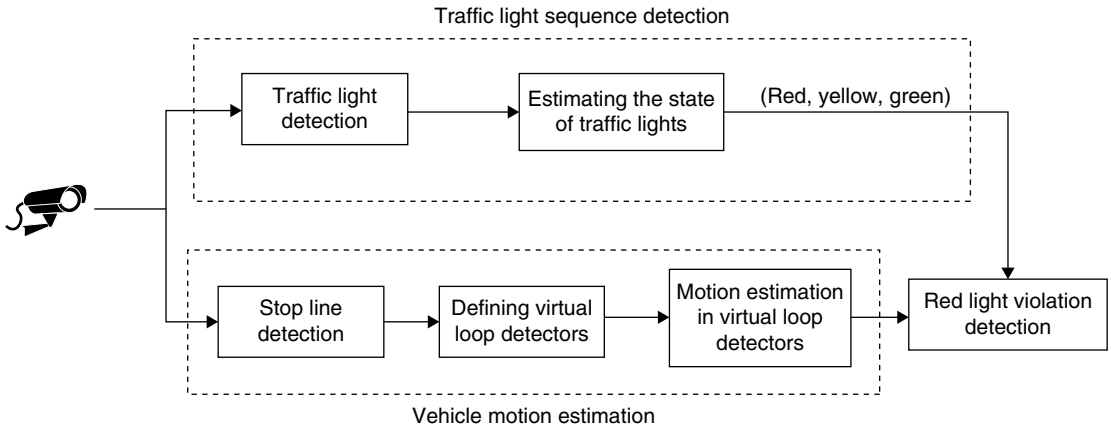

Figure 5.11 Block diagram for the video-based red light violation detection system based on Ref. [48]. *Source*: Yung and Lai [48]. Reproduced with permission of IEEE.

lights turn on at different times. Colored regions can be detected in various color spaces. Reference [48] performs color segmentation for red, yellow, and green regions in the Hue-Saturation-Value (HSV) color space based on the following criteria:

$$\text{Red region} = \left\{ I(x,y) : \left| I_H(x,y) \right| < t_h \quad \text{and} \quad \left| I_V(x,y) \right| > t_v \right\}$$

$$\text{Yellow region} = \left\{ I(x,y) : \left| I_H(x,y) - \frac{\pi}{3} \right| < t_h \quad \text{and} \quad \left| I_V(x,y) \right| > t_v \right\}$$

$$\text{Green region} = \left\{ I(x,y) : \left| I_H(x,y) - \frac{2\pi}{3} \right| < t_h \quad \text{and} \quad \left| I_V(x,y) \right| > t_v \right\},$$

where $I(x,y)$ represents the current frame, $I_{\mathrm{H}}(x,y)$ and $I_{\mathrm{V}}(x,y)$ represent the hue and value channels of the current frame. t_h and t_v are threshold values to account for color variations in outdoor environments. Reference [48] reports typical values for the thresholds t_h and t_v as $\pi/6$ and 0.7, respectively. The location of a traffic light in the video is then determined by finding red, yellow, and green regions that are spatially related (i.e., in sequence from top to bottom) and having similar sizes. After finding the location of traffic lights in the video, they are continuously monitored, as well as tracked for possible movement, to determine the sequence of the lights as the video is streamed. In the VME module, the first step is detecting the stop line in the video. This detection is performed from a single background frame where the stop line is visible. The background frame can be constructed at the initialization using conventional background subtraction methods [48]. The stop line is then detected by applying a Hough transform on the constructed background image where the a priori information about the orientation of the stop line is assumed to be precalculated based on the traffic flow along the street. The stop line is expected to be orthogonal to the traffic flow, which can be estimated by the predominant motion vectors calculated in the video over a period of time.

After detecting the stop line, a prohibited zone is defined beyond the stop line toward intersection. When the lights are red, no moving car is expected to be seen in the prohibited zone moving along

Figure 5.12 Three situations in the cross junction. The VLDs cannot distinguish right turns (situation 3) from the red light violations (situations 1 and 2). *Source*: Luo and Qin [51]. Reproduced with permission of IEEE.

the road direction. In order to detect moving vehicles, a number of virtual loop detectors (VLDs) are defined in the prohibited zone. The number of VLDs depends on both the size of the prohibited zone and the size of the VLDs. Following assignment of VLDs, a reduction process is performed to eliminate redundant VLDs based on the mean and standard deviation of motion vectors across a small set of frames with vehicle motions. The loop detectors that fall into any of the following categories are eliminated:

- VLD with average motion vector not aligned with the road orientation
- VLD with average motion vector magnitude lower than the mean magnitude over all VLDs
- VLD with standard deviation of the motion vector magnitude higher than the average standard deviation of motion vector magnitude over all VLDs

Given a reduced number of VLDs and a signal from the TLSD, the motion of each VLD is estimated by using a block matching algorithm between two consecutive frames. The block matching algorithm tries to find the best match in the target frame (i.e., the first frame) corresponding to the block in the reference frame (i.e., the second frame) in terms of mean absolute difference (MAD). A violation decision is made based on the calculated motion vectors for all VLDs and the signal from the TLSD. If motion along the road direction is detected in more than half of the VLDs with magnitudes larger than half of the mean magnitudes of motion vectors for a given lane, the algorithm makes a decision that the vehicle is in violation.

While VLDs detect red light runners by mimicking the function of induction loops, alone they are not able to distinguish legal right-turning vehicles from the red light violators, which in turn causes false positives for right-turning vehicles. Figure 5.12 illustrates three trajectories that a vehicle can follow at a junction. The first two trajectories correspond to red light runners, while the third trajectory is for the vehicles making a legal right turn. Tracking-based approaches have been proposed in the literature to address this issue [45, 49–52]. Similar to the VLD method, these methods also use video cameras to detect the traffic light sequence without a direct connection to traffic light controller. Apart from traffic light and stop line detection, these methods typically include the following three main steps to detect red light runners: vehicle detection, vehicle tracking, and trajectory analysis. Reference [51], for example, proposes a tracking-based method where vehicle detection is performed using motion analysis. Another common way to detect vehicles is using

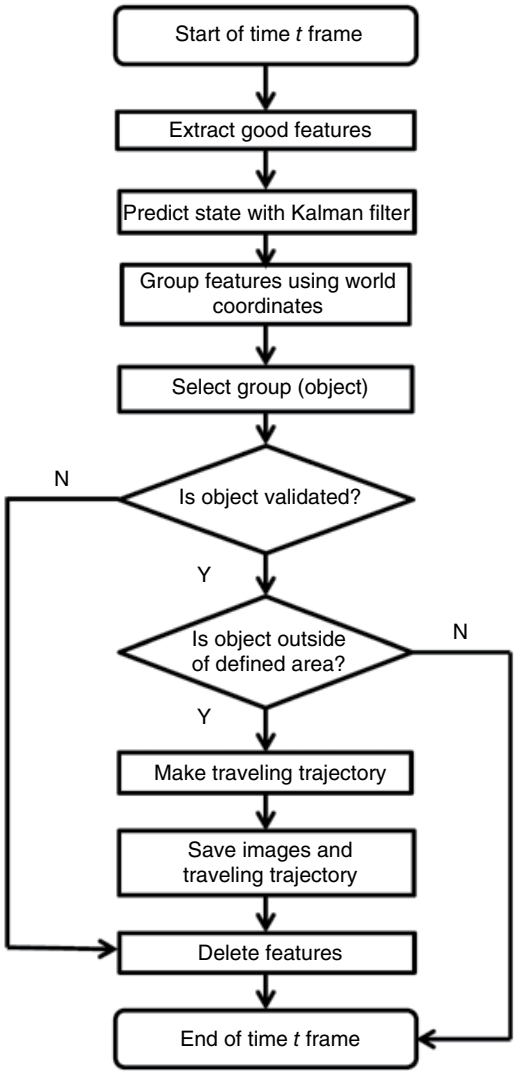

Figure 5.13 Flowchart of the tracking algorithm proposed in Ref. [49]. *Source*: Lim et al. [49]. Reproduced with permission of IEEE.

background subtraction where the background image is adaptively updated using the current frame and background image [49]. Background subtraction can be performed in conventional red, green, and blue channels [49] or in the hue channel after converting RGB image to HSV color space [52]. Once a vehicle is detected in the prohibited region using either motion analysis or background subtraction, the detected vehicle is first checked for correspondence with a vehicle already being tracked. This check ensures that only one tracker is assigned to a vehicle to prevent additional computation burden in tracking. If the detected vehicle is not in the list of vehicles already being tracked, a set of attributes/features are extracted from the detected vehicle. The extracted attributes/features depend on the type of the tracker used. After extracting a set of features/attributes from the

detected region, they are tracked across the video. The tracking is performed as long as the vehicle stays within the field of view of the camera. Tracking techniques such as mean shift tracking, contour tracking, Kalman filtering, KLT tracking, and particle filtering can be employed. Figure 5.13, for example, shows the flowchart of the tracking algorithm used in Ref. [49]. The tracking algorithm specifically uses Kalman filtering where the state $p(t)$ at time t is defined as the 4D vector including the center and size of the vehicle. The state transition between time t and $t+1$ is expressed as follows:

$$\boldsymbol{p}(t+1) = \boldsymbol{p}(t) * \varphi(t, t+1) + \boldsymbol{w}(t).$$

Here, $\varphi(t, t+1)$ and $\boldsymbol{w}(t)$ are a 4×4 identity state transition matrix and a noise term, respectively.

The output of the tracking algorithm is a sequence of x–y coordinates $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ that shows the pixel position of the vehicle in the image plane at each time instant where n denotes the number of moving points. Ideally, it is desired that the sequence is continuous along the y-axis in a way that for every y_i there is a corresponding x_i so that violation detection can be determined using the continuity of the trajectory. This continuity in the sequence, however, is not always possible due to variations in vehicle speed, occlusion, and camera geometry. References [50] and [51] perform a cubic spline interpolation in order to complete the missing points in the sequence and provide continuity in the trajectory. For a viewing geometry similar to Figure 5.12, the violation decision is made based on the following criteria:

- As y_i values in the sequence increase, the extent of the change in the corresponding x_i values is less than a predefined threshold T; the vehicle is concluded as going straight, which in turn indicates a red light violation.
- As y_i values in the sequence increase, the corresponding x_i values also increase and the extent of the increase is larger than a predefined threshold T; the trajectory is determined to be that of a left-turning vehicle and indicates a red light violation.
- As y_i values in the sequence increase, the corresponding x_i values reduce and the extent of the reduce is larger than a predefined threshold T; the trajectory is determined to be that of a right-turning vehicle.

Other alternative methods for trajectory analysis have been also considered in the literature for red light violation detection [45, 49]. Reference [49], for example, determines a red light violation using template matching techniques from the calculated trajectories. Reference [45] proposes a technique that investigates start and end points of the calculated trajectory, where a violation is detected if any of the calculated trajectories has a component in an ROI defined after the traffic lights. In the case of a point tracker, a coherent cluster of trajectories, all having a component in the ROI, can be required to declare a violation.

5.3.2 Stop Sign Enforcement Systems

Nearly 700 thousand annually police-reported motor vehicle crashes and one-third of all intersection crashes in the United States occur at stop signs. Approximately one-third of intersection crashes involve injuries, and more than 40 percent of all fatal crashes occur at stop sign–controlled intersections [53]. Stop sign cameras are a relatively new tool applied to reducing stop sign violations, and resulting accidents [54–56]. The deployed stop sign camera systems are based roughly on RLC platforms, but they differ in the violation–detection mechanism. Stop sign compliance requires a complete stop (zero velocity), which cannot be detected reliably with induction loops. One proposed

system uses a laser speed detector to measure velocity operating within a particular distance range for a given lane [57]. Reference [55] describes a system using a combination of high-resolution still images with full-motion video for irrefutable evidence.

Stop sign violation detection based solely on computer vision has been proposed in the literature. Reference [58] proposes a video-based method using motion vectors. The motion vectors can be read from the live, incoming video stream, or it can be read from a partially decompressed video file that may have been sent to a server for processing. Key steps in the method include the following: (i) capture video using the camera directed at the target stop area; (ii) determine motion vectors from the incoming video stream; (iii) detect the presence of a vehicle that moves across the target area by using a cluster of the motion vectors that are above a threshold in length; (iv) determine whether the detected vehicle stopped within the target area by using a cluster of motion vectors and a motion vector length threshold; (v) if a violation is detected, provide a signal of violation; and (vi) optionally, frames that capture the start, middle, or end of the violation event can be encoded as reference frames to facilitate future searches or rapid retrieval of evidentiary imagery.

In one version of the method proposed in Ref. [58], a block-based approach is used to generate the motion vectors. Motion vectors in block-based approaches describe motion between matching blocks across adjacent frames. The current frame is divided into a number of blocks of a predetermined size, for example, 16×16 pixels. For each reference block of $m \times n$ pixels in a reference frame, a current frame is searched for a target block that is most similar to the reference block. A search window can be defined around the location of the reference block in the reference frame. The search for the best-matching block in a corresponding region of the search window can be conducted using a process, such as, a full extensive search, a binary search, a three-step search, spiral search algorithms, and a combination of these.

The search is achieved by computing a similarity measure between each reference block and potential target blocks in the current frame. Displacements are computed between the select reference block and the target blocks. The displacement can computed using a mean-squared error (MSE) or MAD as follows:

$$\mathrm{MSE}\left(d_1, d_2\right) = \frac{1}{mn}\sum\left(B\left(k,l,j\right) - B\left(k + d_1, l + d_2, j - 1\right)\right)^2, \tag{5.18}$$

$$\mathrm{MAD}\left(d_1, d_2\right) = \frac{1}{mn}\sum\left|B\left(k,l,j\right) - B\left(k + d_1, l + d_2, j - 1\right)\right|, \tag{5.19}$$

where (d_1, d_2) is the vector that describes the relative displacement between reference and target blocks, and $B(k,l,j)$ denotes the pixel located on the kth row and lth column of the $m \times n$ block of pixels in the jth frame; $(j-1)$th frame is the reference frame, and jth frame is the current frame. Because both MSE and MAD measure how dissimilar two blocks are, the similarity measure can be defined as the reciprocal or the negative MSE or MAD. As mentioned earlier, this vector-based method can be used on partial uncompressed video that may have been sent to a server for off-line processing. Compression/decompression algorithms, such as the H264 and MPEG4 algorithms, can be used for extracting the motion vectors. If the standard compression block size and motion vectors are not sufficiently accurate, at minimum they can be used to screen for potential violators that are more deeply analyzed.

While the previous violations involve stopping at an intersection, school buses present a stop sign scenario that can occur in various locations along a roadway. The potential for injury to a child is quite high when vehicles drive past a school bus that is loading and unloading children. Although the

buses activate flashing red lights and deploy a stop sign, it is estimated that 50,000 vehicles illegally pass a school bus each day in New York State alone [59]. Cameras are being deployed to aid in the enforcement of school bus signs. While some enforcement systems are using manual review of video to detect violators, others are proposing computer vision solutions with significantly reduced labor and cost [60]. In Ref. [60], the video is partitioned into segments each corresponding to a single bus stop, moving vehicles are detected using techniques such as frame differencing or analysis of motion vectors, frames are tagged where moving vehicles are detected, license plates are localized, and automatic license plate recognition (ALPR) is performed.

5.4 Other Violations

Driver violations at intersections are the major cause of fatal accidents in urban traffic. According to the USDOT, nearly half of all traffic accidents and 20% of all fatal crashes occur in close proximity to an intersection [29]. This statistic has remained substantially unchanged since the past decade despite significant efforts by transportation agencies for improved intersection designs and sophisticated applications of transportation engineering [29]. To combat this persistent trend, municipalities and law enforcement agencies are employing automated stop enforcement technologies at red lights and stop signs, as well as for stopped school buses.

5.4.1 Wrong-Way Driver Detection

According to the US National Transportation Safety Board [61], wrong-way driving is "vehicular movement along a travel lane in a direction opposing the legal flow of traffic." Although wrong-way driving events occur relatively infrequently (accounting for only 3% of accidents on high-speed divided highways), they can result some of the most serious types of accidents that occur on highways. Consequently, accidents resulting from wrong-way driving are much more likely to result in fatal and serious injuries than other types of accidents. As a result, considerable effort has been put into developing technologies that enable prevention and detection of wrong-way drivers. While detection systems based on microwave and magnetic sensors, as well as on Doppler radar, have been proposed [62], we focus on systems relying on video acquisition and processing.

In one instance, the authors of Ref. [63] exploited the fact that a wrong-way driving event is an anomalous event and, consequently, detectable by a system that is aware of expected patterns of motion associated with traffic on a highway. Based on this assumption, they propose a system that undergoes a learning stage in which optical flow is computed for video sequences in which vehicles are traversing the highway in a lawful manner. The normal direction of motion for each lane in the scene is represented by a mixture of Gaussians aimed at modeling the statistical behavior of the orientation parameter in the optical flow field. Once a model is constructed, optical flow is computed from frames in the incoming video feed. A determination is made that an object is traveling in the wrong direction when the difference between the direction of the flow associated with the object and that learned by the model is larger than 2.57 times the standard deviation of the closest matching Gaussian component in the model. In order to minimize false alarms, a verification stage ensures that a wrong-way notification is only issued when an anomalous motion pattern is detected more than n times across m consecutive frames, for two positive integers n and m. In Ref. [62], the feasibility of a thermal imaging system aimed at detecting entry of wrong-way vehicles onto the highway system was tested. The proposed system relied on a long-range thermal camera with dual detection zones that estimated the direction of travel of vehicles entering and exiting the highway based on the

sequence of and the temporal intervals between the triggered detections. The performance of the proposed system was shown to outperform that of competing systems based on regular RGB imaging, which failed to detect wrong-way vehicles at night when the vehicle headlights were off.

5.4.2 Crossing Solid Lines

Unsafe lane changes account for a high percentage of the total accidents that occur on the road, second only to speeding. There are several scenarios for unsafe lane changes, a common one being overtaking a vehicle by crossing a solid line. The corresponding accidents are often quite severe due to the head-on nature of the collisions. Crossing a solid line is a moving violation in many jurisdictions worldwide. One photo enforcement system described in the literature is known as *Police Eyes* [64]. Police Eyes operates by detecting moving blobs and their intersection with a violation region. Key steps in the Police Eyes systems are the following:

1) *System Initialization*: An operator specifies a violation region and the processing area on an initial image indicated by manually clicking on image points.
2) *Image Acquisition from IP Cameras*: Images are acquired from two IP cameras continuously. A low-resolution image from one camera is used to detect blobs and to identify violations. A high-resolution image from the second camera is used to identify the vehicle. The images and video clips and can be later used for evidentiary purposes and to reject nontrivial cases such as avoidance of hazards.
3) *Update a Background Model and Background Subtraction*: The background model is initialized using a single frame and then updated for every new frame. A Gaussian mixture model is used for every pixel in the image. The number of Gaussian components is constantly adapted per pixel. Frame differencing from the background model produces a foreground image.
4) *Shadow Detection*: Shadow pixels must be removed from the foreground image to avoid false violation detections. Shadow pixels are identified using a combination of normalized cross-correlation between the foreground region and the corresponding background pixels, along with RGB vector distances between the foreground pixels and underlying background pixels.
5) *Blob Extraction*: Foreground blobs are extracted from the foreground image through connected component analysis after performing morphological operations on the foreground image to remove noisy blobs. The base profile is extracted for each remaining blob. The base of a blob is identified as the set of lowermost pixels of the external contour of the blob.
6) *Violation Analysis*: Analysis of the region of intersection of the base profile of each blob with the violation area is used to detect violations.

References

1 Traffic Safety Facts, 2012 Data, Speeding, National Highway Traffic Safety Administration, NHTSA's National Center for Statistical Analysis, Washington, DC, DOT HS 812 021 (2014).
2 The Economic and Societal Impact of Motor Vehicle Crashes, NHTSA's National Highway Traffic Safety Administration, National Center for Statistical Analysis, Washington, DC, DOT HS 812 013, (2010) (Revised).
3 L. Thomas, R. Srinivasan, L. Decina, and L. Staplin, Safety effects of automated speed enforcement programs: critical review of international literature, *Transportation Research Record: Journal of the Transportation Research Board*, 2078, 117–126, 2008, Transportation Research Board of the National Academies, Washington, DC.

4 A. Yilmaz, O. Javed, and M. Shah, Object tracking: a survey, *ACM Computing Surveys*, 38(4), 1–45, 2006.

5 N. K. Kanhere and S. T. Birchfield, A taxonomy and analysis of camera calibration methods for traffic monitoring applications, *IEEE Transactions on Intelligent Transportation Systems*, 11(2), 441–452, 2010.

6 D. J. Dailey, F. W. Cathey, and S. Pumrin, An algorithm to estimate mean traffic speed using uncalibrated cameras, *IEEE Transactions on Intelligent Transportation Systems*, 1(2), 98–107, 2000.

7 S. Pumrin and D. J. Dailey, Roadside camera motion detection for automated speed measurement, in *Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems*, Singapore, September 3–6, 2002, pp. 147–151.

8 T. N. Schoepflin and D. J. Dailey, Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation, *IEEE Transactions on Intelligent Transportation Systems*, 4(2), 90–98, 2003.

9 Z. Zhang, T. Tan, K. Huang, and Y. Wang, Practical camera calibration from moving objects for traffic scene surveillance, *IEEE Transactions on Circuits and Systems for Video Technology*, 36(5), 1091–1103, 2013.

10 F. W. Cathey and D. J. Dailey, A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras, in *Proceedings of IEEE Intelligent Vehicles Symposium*, Las Vegas, NV, June 6–8, 2005, pp. 777–782.

11 N. K. Kanhere, S. T. Birchfield, and W. A. Sarasua, Automatic camera calibration using pattern detection for vision-based speed sensing, in *Transportation Research Board Annual Meeting*, 87th Annual Meeting, Washington, DC, January 13–17, 2008, pp. 30–39.

12 J. Wu, Z. Liu, J. Li, C. Gu, M. Si, and F. Tan, An algorithm for automatic vehicle speed detection using video camera, in *Proceedings of 2009 4th International Conference on Computer Science and Education*, Nanning, July 25–28, 2009.

13 L. G. C. Wimalaratna and D. U. J. Sonnadara, Estimation of the speeds of moving vehicles from video sequences, in *Proceedings of the Technical Sessions*, 24, March 2008, pp. 6–12, Institute of Physics, Colombo, Sri Lanka. http://www.ipsl.lk/index.php/technical-sessions/18-publications/technical-sessions/58-volume-24-2008 (accessed on October 14, 2016).

14 L. Grammatikopoulos, G. Karras, and E. Petsa (GR), Automatic estimation of vehicle speed from uncalibrated video sequences, in *Proceedings of the International Symposium on Modern Technologies, Education and Professional Practice in Geodesy and Related Fields*, Sofia, November 2005, pp. 332–338.

15 A. G. Rad, A. Dehghani, and M. R. Karim, Vehicle speed detection in video image sequences using CVS method, *International Journal of the Physical Sciences*, 5(17), 2555–2563, 2010.

16 E. A. Bernal,W. Wu, O. Bulan, and R. P. Loce, Monocular vision-based vehicular speed estimation from compressed video streams, in *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems for All Transport Modes*, The Hague, October 6–9, 2013.

17 P. Bellucci, E. Cipriani, M. Gagliarducci, and C. Riccucci, The SMART project: speed measurement validation in real traffic condition, in *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, Vienna, September 16–16, 2005.

18 Z. Tian, M. Kyte, and H. Liu, Vehicle tracking and speed measurement at intersections using video-detection systems, *ITE Journal*, 79(1), 42–46, 2009.

19 Available: http://project-asset.com/data/presentations/P_011-4.pdf (accessed on September 19, 2016).

20 J. Battle, E. Mouaddib, and J. Salvi, Recent progress in coded structured light as a technique to solve the correspondence problem: a survey, *Pattern Recognition*, 31(7), 963–982, 1998.

21 C. Stauffer and W. Grimson, Adaptive background mixture models for real-time tracking, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Fort Collins, CO, June 23–25, 1999.

22 G. Gordon, T. Darrell, M. Harville, and J. Woodfill, Background estimation and removal based on range and color, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Fort Collins, CO, June 23–25, 1999.

23 K. Ni and F. Dellaert, Stereo tracking and three-point/one-point algorithms: a robust approach in visual odometry, in *Proceedings of the IEEE International Conference on Image Processing*, Atlanta, October 8–11, 2006.

24 S. Baker and I. Matthews, Lucas-Kanade 20 years on a unifying framework, *International Journal of Computer Vision*, 56(3), 221–255, 2004.

25 R. Balasubramanian, S. Das, S. Udayabaskaran, and K. Swaminathan, Quantization error in stereo imaging systems, *International Journal of Computer Math*, 79(6), 671–691, 2002.

26 L. Hardin and L. Nash, Optical range and speed detection system. U.S. Patent No. 5,586,063, December 17, 1996.

27 Available: http://www.imagsa.com/main/images/datasheet/Atalaya3D_Speed.pdf (accessed on September 19, 2016).

28 Available: http://www.foxnews.com/leisure/2011/11/11/russian-super-speed-camera-can-issue-thousands-tickets-per-hour/ (accessed on September 19, 2016).

29 The National Intersection Safety Problem, Brief Issues #2, U. S. Department of Transportation, Federal Highway Administration, FHWA-SA-10-005.

30 R. A. Retting, A. F. Feldman, C. M. Farmer, and A. F. Williams, Evaluation of red light camera enforcement in Fairfax, Va., USA, *ITE Journal*, 69, 30–35, 1999.

31 R. A. Retting, R. G. Ulmer, and A. F. Williams, Prevalence and characteristics of red light running crashes in the United States. *Accident Analysis and Prevention*, 31, 687–694, 1999.

32 T. Supriyasilp, D. S. Turner, and J. K. Lindly, *Pilot Study of Automated Red Light Enforcement*. University Transportation Center for Alabama (*UTCA*) *Report* 470, September 30, 2003.

33 S. E. Hill and J. K. Lindly, *Red light running prediction and analysis*. Ph.D. dissertation, University of Alabama, 2002.

34 R. A. Retting, S. A. Ferguson, and C. M. Farmer, Reducing red light running through longer yellow signal timing and red light camera enforcement: results of a field investigation, *Accident Analysis and Prevention*, 40, 327–333, 2008.

35 J. A. Bonneson and H.J. Son, Prediction of expected red-light running frequency at urban intersections. *Transportation Research Record: Journal of the Transportation Research Board*, 1830, 38–47, 2003.

36 J. A. Bonneson and K. H. Zimmerman, Effect of yellow-interval timing on the frequency of red light violations at urban intersections. *Transportation Research Record: Journal of the Transportation Research Board*, 1865, 20–27, 2004.

37 P. Guerin, *City of Albuquerque Yellow Light Timing Change and All-Red Clearance Interval Timing Change Effectiveness Study Final Report*. Prepared for: The City of Albuquerque, Department of Municipal Development and the Office of the Mayor, September 2012. http://isr.unm.edu/reports/2012/city-of-albuquerque-yellow-light-timing-change-and-all-red-clearance-interval-time-change-effectiveness-study-final-report..pdf (accessed on October 14, 2016).

38 Q. Yang, L. D. Han, and C. R. Cherry, Some measures for sustaining red-light camera programs and their negative impacts, *Transport Policy*, 29, 192–198, 2013.

39 D. M. Smith, J. McFadden, and K. A. Passetti, Automated enforcement of red light running technology and programs: a review, *Transportation Research Record: Journal of the Transportation Research Board*, 1734, 29–37, 2000.

40 R. A. Retting, S. A. Ferguson, and A. S. Hakkert, Effects of red light cameras on violations and crashes: a review of the international literature, *Traffic Injury Prevention*, 4, 17–23, 2003.

41 K. A. Passetti and T. H. Hicks, *Use of automated enforcement for red light violations*. Texas A&M University, College Station (1997).

42 R. W. Lock, Image recording apparatus and method. U.S. Patent No. 8,390,476. March 5, 2013.

43 O. Bulan, A. Burry, and R. P. Loce, Short-time stopping detection from red light camera evidentiary photos. U.S. Patent App. No 14/278196, filed May 15, 2014

44 M. Glier, D. Reilly, M. Tinnemeier, S. Small, S. Hsieh, R. Sybel, and M. Laird, Method and apparatus for traffic light violation prediction and control. U.S. Patent App. No. 09/852487, published May 9, 2002.

45 O. Bulan, A. Burry, and R. P. Loce, Short-time stopping detection from red light camera videos. U.S. Patent App. No 14/278218, filed May 15, 2014.

46 O. Fucik, P. Zemcik, P. Tupec, L. Crha, and A. Herout, The networked photo-enforcement and traffic monitoring system Unicam, in *Proceedings of the 11th IEEE International Conference and Workshop on Engineering of Computer-Based Systems*, Brno, May 27, 2004, pp. 423–428.

47 Camea, Red light monitoring, http://www.camea.cz/en/traffic-applications/enforcement-systems/red-light-violation-detection-unicamredlight-2/ (accessed December 3, 2014).

48 N. H. C. Yung and H. S. Lai, An effective video analysis method for detecting red light runners, *IEEE Transactions on Vehicular Technology*, 50(4), 1074–1084, 2001.

49 D. W. Lim, S. H. Choi, and J. S. Jun, Automated detection of all kinds of violations at a street intersection using real time individual vehicle tracking, in *Proceedings of the Fifth IEEE Southwest Symposium on Image Analysis and Interpretation*, Sante Fe, NM, April 9, 2002, pp. 126–129.

50 M. Heidari and S. A. Monadjemi, Effective video analysis for red light violation detection, *Journal of Basic and Applied Scientific Research*, 3(1), 642–646, 2013.

51 D. Luo, X. Huang, and L. Qin, The research of red light runners video detection based on analysis of tracks of vehicles, in *Proceedings of the IEEE International Conference on Computer Science and Information Technology*, Singapore, August 29 to September 2, 2008, pp. 734–738.

52 K. Klubsuwan, W. Koodtalang, and S. Mungsing, Traffic violation detection using multiple trajectories evaluation of vehicles, in *Proceedings of the 4th International Conference on Intelligent Systems Modelling and Simulation (ISMS)*, Bangkok, January 29–31, 2013. IEEE.

53 Insurance Institute for Highway Safety, 37, No 9, October 26, 2000. http://safety.fhwa.dot.gov/intersection/conventional/unsignalized/case_studies/fhwasa09010/ (accessed on October 14, 2016).

54 TheNewspaper.com: Driving Politics. Stop Sign Ticket Cameras Developed: New Stop Sign Cameras Will Issue Automated Tickets for Boulevard Stops. http://www.thenewspaper.com/news/17/1742.asp, September 5, 2007.

55 http://www.bypass.redflex.com/application/files/5814/4902/9070/redflexstop.pdf (accessed October 14, 2015).

56 M. Trajkovic and S. Gutta, Vision-based method and apparatus for monitoring vehicular traffic events. U.S. Patent 6,442,474, August 27, 2002.

57 M. Phippen and D. W. Williams, Speed measurement system with onsite digital image capture and processing for use in stop sign enforcement. U.S. Patent No. 6,985,827. January 10, 2006.

58 E. Bernal, O. Bulan, and R. Loce, Method for stop sign law enforcement using motion vectors in video streams. U.S. Patent App. No. 13/613174, March 13, 2014.

59 OPERATION SAFE STOP, An Educational Campaign by the New York Association for Pupil Transportation. Prepared by NYAPT through the National Transportation Safety Administration under a grant from the Governor's Traffic Safety Committee. http://slideplayer.com/slide/1508781/ (accessed on October 14, 2016).

60 A. Burry, R. Bala, and Z. Fan, *Automated processing method for bus crossing enforcement*. U.S. Patent App. No. 13/210447, published February 21, 2013.

61 US National Transportation Safety Board, Wrong-Way Driving. Special Investigative Report 12/01. US National Highway Traffic Safety Administration, 2012. pp. 1–77.

62 S. Simpson, Wrong-way Vehicle Detection: Proof of Concept, Report to the Arizona Department of Transportation Research Center, FHWA-AZ-13-697, March 2013.

63 G. Monteiro, M. Ribeiro, J. Marcos, and J. Batista, A framework for wrong-way driver detection using optical flow, *Proceedings of the 4th International Conference on Image Analysis and Recognition*, Montreal, August 22–24, 2007.

64 R. Marikhu, J. Moonrinta, M. Ekpanyapong, M. Dailey, and S. Siddhichai, Police eyes: real world automated detection of traffic violations, in *Proceedings of the 2013 10th International Conference on Electrical Engineering/Electronics, Computer Telecommunications and Information Technology (ECTI-CON)*, Krabi, May 15–17, 2013.