A Tracking-Based Traffic Performance Measurement System for Roundabouts and Intersections

Final Report

Prepared by:

Hua Tang
Hai Dinh

Department of Electrical and Computer Engineering
University of Minnesota Duluth

Northland Advanced Transportation Systems Research Laboratories
University of Minnesota Duluth

CTS 12-10

Technical Report Documentation Page

1. Report No.	2.	3. Recipients Accession No.	
CTS 12-10			
4. Title and Subtitle A Tracking-Based Traffic Performance Measurement System for Roundabouts and Intersections		5. Report Date May 2012 6.	
7. Author(s) Hua Tang, Hai Dinh		8. Performing Organization Report No.	
9. Performing Organization Name and Address Department of Electrical and Computer Engineering University of Minnesota Duluth 1023 University Drive Duluth MN 55812		10. Project/Task/Work Unit No. CTS Project #2011013 11. Contract (C) or Grant (G) No.	
12. Sponsoring Organization Name and Address Intelligent Transportation Systems Institute Center for Transportation Studies 200 Transportation and Safety Building 511 Washington Ave. SE Minneapolis, MN 55455		13. Type of Report and Period Covered Final Report, 07/01/2010 to 12/31/2011 14. Sponsoring Agency Code	
15. Supplementary Notes http://www.its.umn.edu/Publications/ResearchReports/			

16. Abstract (Limit: 250 words)

Automatic traffic data collection can significantly save labor work and cost compared to manual data collection. The collected traffic data is necessary for traffic simulation and modeling, performance evaluation of the traffic scene, and eventually (re)design of the traffic scene. However, automatic traffic data collection has been one of the challenges in Intelligent Transportation Systems (ITS). This project presents the development of a single camera-based video system for automatic traffic data collection for roundabouts and intersections. The system targets roundabouts and intersections because no mature data collection systems exist for these traffic scenes yet in contrast to highway scenes. The developed system has mainly processing modules. First, the camera is calibrated for the traffic scene of interest and a novel circle-based calibration algorithm is proposed for roundabouts. Second, the system tracks vehicles from the video by incorporating powerful imaging processing techniques and tracking algorithms. Finally, the resulting vehicle trajectories from vehicle tracking are analyzed to extract the interested traffic data, which includes vehicle volume, vehicle speed (including acceleration/de-acceleration behavior), travel time, rejected gaps, accepted gaps, follow-up time and lane use. Practical tests of the developed system show that it can reliably track vehicles and provide reasonably accurate traffic data in most cases.

17. Document Analysis/Descriptors	18. Availability Statement		
Intelligent transportation systems, Cameras, Traffic surveillance, Automatic data collection systems, Traffic data, Implementation	No restrictions. Document available from: National Technical Information Services, Alexandria, Virginia 22312		
19. Security Class (this report) Unclassified	20. Security Class (this page) Unclassified	21. No. of Pages 62	22. Price

A Tracking-Based Traffic Performance Measurement System for Roundabouts and Intersections

Final Report

Prepared by:

Hua Tang
Hai Dinh

Department of Electrical and Computer Engineering
University of Minnesota Duluth

Northland Advanced Transportation Systems Research Laboratories
University of Minnesota Duluth

May 2012

Published by:

Intelligent Transportation Systems Institute
Center for Transportation Studies
University of Minnesota
200 Transportation and Safety Building
511 Washington Avenue, S.E.
Minneapolis, Minnesota 55455

Acknowledgments

The authors wish to acknowledge those who made this research possible. The study was funded by the Intelligent Transportation Systems (ITS) Institute, a program of the University of Minnesota's Center for Transportation Studies (CTS). Financial support was provided by the United States Department of Transportation's Research and Innovative Technologies Administration (RITA).

The project was also supported by the Northland Advanced Transportation Systems Research Laboratories (NATSRL), a cooperative research program of the Minnesota Department of Transportation, the ITS Institute, and the University of Minnesota Duluth College of Science and Engineering.

The authors would like to thank Dr. Eil Kwon, director of NATSRL at University of Minnesota Duluth, for very valuable discussions and comments.

Authors would also like to thank the Minnesota Department of Transportation (MnDOT) for providing all traffic videos tested in this project.

Table of Contents

List of Figures

List of Tables

Executive Summary

In Intelligent Transportation Systems (ITS), traffic data collection has been a challenge. The collected traffic data is necessary for traffic simulation and modeling, performance evaluation of the traffic scene, and eventually (re)design of the traffic scene. Traditionally, manual data collection is the only approach either using handheld devices for vehicle counting or post-counting or post-processing of pre-recorded videos. However, the manual approach is very laborious and costly. Therefore, in past years, automatic traffic data collection is an important research topic in ITS. Automatic traffic data collection for highways has become available now and even commercial tools have been developed. However, due to much more complicated traffic scene and traffic behavior, automatic traffic data collection for intersections and roundabouts has been left behind.

The proposed project is to design a tracking based traffic performance measurement system that can obtain all types of performance measurements for roundabouts and intersections. For this purpose, a video-based tracking approach is most competent as it allows for the most comprehensive traffic data collection, such as vehicle volume, vehicle speed (including acceleration/de-acceleration behavior), travel time, rejected gaps, accepted gaps, follow-up time, lane change and so on. In spite of this significant advantage for the video-based tracking system, it is known that the video-based approach suffers from a few problems, such as vehicle occlusion, camera shaking (due to wind), light and weather change.

To allow video-based traffic performance measurements, the first step is to perform calibration of cameras. In literature, most works target camera calibration for highways and few methods are reported for roundabouts. In this project, we proposed a simple method to estimate intrinsic and extrinsic parameters of the camera for roundabout traffic scenes. The proposed method can estimate focal length, pan angle, tilt angle, and camera height by matching the ellipse equation extracted from an image with the perspective-transformed equation of the corresponding real-world circle. The proposed method requires only one image for camera calibration. The method is validated with real-world roundabout traffic scenes and the calibration results are reasonably accurate compared to ground truth measurements.

Once the camera is calibrated, the next step in the developed traffic data collections system is to process the videos to allow vehicle tracking. Traditional video-based vehicle tracking has two steps of processing, which are vehicle segmentation and vehicle association (or tracking). This conceptual two-step processing flow is retained in our proposed system, but we have re-designed or significantly revised each processing step to deal with the aforementioned issues and improve tracking accuracy. First, in this project, the vehicle segmentation algorithm is based on the Mixture-of-Gaussian (MoG) algorithm. In MoG, each pixel is modeled probabilistically with a mixture of several Gaussian distributions (typically 3 to 5) and each background sample is used to update the Gaussian distributions, so that the history of pixel variations can be stored in the MoG model. Latest light and weather changes can be reflected by updating the mean and variance of MoG. Hence, the sensitivity and accuracy of vehicle detection is highly improved compared to traditional techniques. Another very practical advantage of MoG is its robustness against camera shaking, which is one very practical problem that affects tracking accuracy. As the obtained images are not stable, large regions of noisy strips are segmented from the image, which severely affects the accuracy of object extraction. MoG can fight with camera shaking

very well, as the pixel changes caused by camera shaking are recorded in the background distributions and therefore not identified as false segmentation regions any more. In addition, we add one post-processing step after object segmentation to further suppress remaining false detections from camera shaking. This is based on the observation that the false detection regions will have a high probability of being part of the background distributions at their original locations. Therefore, we can decide whether a detected pixel is caused by a real foreground object or a background object by considering the background distributions in a small neighborhood of the detection. If the detection pixel matches with the background distributions of pixels in the neighborhood, it is highly possible that this detection is falsely caused by camera shaking. When considering color images with the RGB (Red, Green, Blue) space, this technique proves to be very effective as the probability of mismatching is very small.

The next step of processing after object segmentation is vehicle tracking (or association), that is to relate the objects extracted from the current image frame to the identified vehicles from the previous image frame. The proposed approach in the project is region-based tracking with combined Kalman filtering, Kernel-based tracking and overlap-based optimization techniques for handling vehicle occlusions. Kalman filtering could help when vehicles are occluded for a short time, but if vehicles are occluded for a long time and have dramatic velocity changes while occluded, then errors could happen. To deal with this problem, we apply Kernel-based tracking in joint feature-spatial spaces and overlap-based optimization when occlusion is detected. To further improve tracking accuracy in case of occlusion, we also employ overlap-based optimization. The purpose is to maximize covering the area of the objects by occluding vehicles.

The outputs from vehicle tracking are moving trajectories of all vehicles. The last step of the developed traffic performance measure system is to mine the vehicle trajectories to automatically extract interested traffic data. From these moving trajectories, we can obtain readily source-destination pairs and vehicle volume. Also, note that the moving trajectory is composed of positions of the vehicles at different time moments, therefore we can compute the vehicle speed at any time, which would help us study the acceleration/de-acceleration behavior of the vehicles. The projection of the vehicle in real-world coordinates to image coordinates is needed, which has been obtained from camera calibration in the first place. Note that from vehicle moving trajectories, we can also detect whether the vehicle make lane changes. This can be done by using the position information of the lanes from camera calibration. As for waiting time, this can be obtained by counting the number of image frames that it takes for the vehicle to exit the scene after its entrance. In many cases, traffic engineers may be more interested in the waiting time from a certain position in real-world to another position, for instance, when the vehicle starts to queue at the intersection until it makes the turn. This can be accommodated in our system, since the complete moving trajectory with attached time moments are available from vehicle tracking. Finally, we can derive gap size. In the case of roundabouts, we can first identify when each vehicle enters the roundabout from the ramp entrance, then record this time moment and the positions of other vehicles in the roundabout at this time moment. By computing the real-world distances of other vehicles from the entering vehicle, we can compute gap size of the entering vehicle. Except the real-world distances, we can compute gap size in terms of time duration by counting how many image frames or how long it takes for the other vehicles to reach the position where the vehicle enters the roundabout. The approach can be similarly applied to compute gap size for intersections with slight modifications. Similarly, we can derive rejected gap size and follow-up time between two consecutive vehicles entering the traffic scene.

The developed traffic performance measurement system is currently implemented in combined C programming languages and MATLAB software running on personal computers. We have performed extensive testing of the system using 72 practical videos of a roundabout entrance traffic scene from existing Minnesota Department of Transportation (Mn/DOT) surveillance cameras. The automatically extracted traffic data have been compared to ground truth data that were manually collected by inspecting the video and it was found that the automatically extracted traffic data are reasonably accurate with 70% to 90% accuracy. It was also found that the main factors that affect the accuracy are severe vehicle occlusions (mainly due to the limited view coverage of the single camera and long queue of vehicles at the roundabout entrance).

1 Introduction

1.1 Traffic surveillance

The increasing traffic demand in conjunction with the limited construction of new roads has caused recurring congestion in the U.S. While building new facilities is still needed, a much less expensive additional measure is utilization of existing roads, which is one of the important goals in Intelligent Transportation System (ITS). Using current facilities wisely could help reduce travel time and pollutant emissions, ease delay and congestion, and improve safety. To enhance efficient usage and capacity of current transportation networks, detailed traffic data are needed. However, manual collection of traffic data is very laborious and costly. Therefore, automatic traffic data collection has been one of the active research areas in ITs. In the past, various approaches to automatic traffic data collection based on inductive loops, microwave radar, cameras, etc. were proposed.

Each method has its strength and weakness. For example, whereas the inductive loop technique is mature and well understood, its installation requires pavement cut and also the accuracy decreases when a large variety of vehicle classes are to be detected. Microwave radar, whereas it is easier to install and could measure speed directly, has difficulty when detecting stopped vehicles. Another emerging technology is vehicle-based sensor networks, which collects data by locating vehicles via mobile phones or Global Positioning System (GPS) devices over the entire road network. But this technology has raised many concerns about drivers' privacy and it also requires devices installed in all vehicles.

The type of data that can be collected also varies between these methods. Whereas most methods could produce vehicle count and speed, not all of them could classify vehicle type or be employed for multiple lane detection. Table 1.1 shows traffic output data and communication bandwidth of typical technologies. Their equipment costs and life time can be found in Table 1.2. Those data gives us a glimpse of how to choose appropriate technology to collect desired traffic data. As it is seen from Table 1.1, regarding the most variety of traffic data that could be collected, video image processor is the best approach. This is a very important advantage to traffic data collection for some complex traffic scenes such as a roundabout or an intersection for which there are many traffic data specifications that are interesting to the traffic engineers in order fully assess the performance of the traffic scene. For example, traffic performance measurements for highway target mostly vehicle volume, vehicle speed and lane use, but for roundabouts or intersections, other measurements such as origin-destination pairs, waiting time and gap size are needed as well.

1.2 Video-based system

Video-based systems have been applied for traffic data collection since 1970s. Figure 1.1 shows an Autoscope camera by Image Sensing System Inc., a video-based system which was initiated at the University of Minnesota in 1984 [3] and has been installed in more than 55 countries [4]. However, the main application of the Autoscope video system is vehicle presence detection or monitoring while producing limited performance measurements such as vehicle count. Also, the system uses running average based background estimation for segmentation. Therefore, there is no explicit global threshold for extracting vehicles. Instead, the empirical database is used to

adjust the vehicle presence detection by the selection of particular algorithms or merely parameter values. Furthermore, the system was developed mainly for highways and intersections, not roundabouts.

In this project, we present the development of a video-based traffic data collection system for roundabouts and intersections. The system targets roundabouts and intersections because no mature data collection systems exist for these traffic scenes yet in contrast to highway scenes. The developed system has mainly three steps of processing. First, the camera is calibrated for the traffic scene of interest and a novel circle-based calibration algorithm is proposed for roundabouts. Second, the system tracks vehicles from the video by incorporating powerful imaging processing techniques and tracking algorithms. The proposed approach in the project is region-based tracking with combined Kalman filtering, Kernel-based tracking and overlap-based optimization techniques for handling vehicle occlusions. Finally, the resulting vehicle trajectories from vehicle tracking are analyzed to extract the interested traffic data, which includes vehicle volume, vehicle speed (including acceleration/de-acceleration behavior), travel time, rejected gaps, accepted gaps, follow-up time and lane use.

Table 1.1. Traffic output data and communication bandwidth of available sensors [1].

Sensor technology	Count	Presence	Speed	Output data	Classification	Multiple lane, multiple detection zone data	Communication bandwidth
Inductive loop	✓	✓	✓[b]	✓	✓[c]		Low to moderate
Magnetometer (two axis fluxgate)	✓	✓	✓[b]	✓			Low
Magnetic induction coil	✓	✓[d]	✓[b]	✓			Low
Microwave radar	✓	✓[e]	✓	✓[e]	✓[e]	✓[e]	Moderate
Active infrared	✓	✓	✓[f]	✓	✓	✓	Low to moderate
Passive infrared	✓	✓	✓[f]	✓			Low to moderate
Ultrasonic	✓	✓		✓			Low
Acoustic array	✓	✓	✓	✓		✓[g]	Low to moderate
Video image processor	✓	✓	✓	✓	✓	✓	Low to high[h]

Table 1.2. Equipment cost of some detectors [2].

Unit Cost Element	Lifetime (years)	Capital Cost ($1000)	Cost Date	O&M Cost ($1000)	Cost Date
Inductive Loop Surveillance on Corridor	5	3-8	2001	0.4-0.6	2005
Inductive Loop Surveillance at Intersection	5	8.6-15.3	2005	0.9-1.4	2005
Machine Vision Sensor on Corridor	10	21.7-29	2003	0.2-0.4	2003
Machine Vision Sensor at Intersection	10	16-25.5	2005	0.2-1	2005
Passive Acoustic Sensor on Corridor		3.7-8	2002	0.2-0.4	1998
Passive Acoustic Sensor at Intersection		5-15	2001	0.2-0.4	2002
Remote Traffic Microwave Sensor on Corridor	10	9-13	2005	0.1-0.58	2005
Remote Traffic Microwave Sensor at Intersection	10	18	2001	0.1	2001
Infrared Sensor Active		6-7.5	2000		
Infrared Sensor Passive		0.7-12	2002		
CCTV Video Camera	10	9-19	2005	1-2.3	2004
CCTV Video Camera Tower	20	4-12	2005		

Figure 1.1. Autoscope Solo Terra video detection system [4].

1.3 Organization of the report

In the following chapters, we describe in detail the developed video-based system for traffic data collection of roundabouts and intersections.

- In Chapter two, we present a novel approach to calibrate the camera for roundabout traffic scenes.
- In Chapter three, the vehicle segmentation and tracking algorithms are presented in detail.
- In Chapter four, algorithms for traffic data collection are reported.
- Finally, conclusions are given in chapter five.

2 Camera Calibration

As introduced and motivated in Chapter 1, this project targets a video-based system for automatic traffic performance measurement. For any camera-based video system in ITS, camera calibration is the pre-requisite. The purpose of camera calibration is to establish the projection relationship between the vehicle in the real world and those in the images so that vehicle movements in the 2D image space can be associated to the real-world distances in the 3D world. In this project, we present a simple method to estimate intrinsic and extrinsic parameters of the camera for roundabout traffic scenes. Unlike many previous works on camera calibration for roadway scenes by using parallel lines, our work addresses camera calibration for roundabout scenes by using a circle. The proposed method can estimate focal length, pan angle, tilt angle, and camera height by matching the ellipse equation extracted from an image with the perspective-transformed equation of the corresponding real-world circle. Our method requires only one image for camera calibration. The method is validated with real-world roundabout traffic scenes and the calibration results are reasonably accurate compared to ground truth measurements. This rest of the Chapter is organized as follows. Section 2.1 gives an introduction of previous methods for camera calibration and the motivation behind this work. Section 2.2 describes the proposed method for camera calibration of roundabout scenes in detail. Section 2.3 presents experimental results of the proposed method tested on both lab scenes and real-world roundabout scenes. Finally, conclusions are drawn in Section 2.4.

2.1 Introduction and previous work

Camera calibration is to estimate the camera's perspective information from images of real-world objects. It is a well-grown field in computer vision and many techniques for camera calibration have been proposed [5]. According to the dimension of calibration objects, they can be classified into four categories, 3D reference object based calibration (using orthogonal planes) [6], 2D-plane based calibration (using planar objects) [7][8], 1D-line based calibration (using a set of collinear points) [9], and 0D calibration (or self-calibration) that does not use any calibration object [10].

Due to widespread use of camera-based vision systems in ITS for video recording, traffic monitoring and surveillance, camera calibration has also become an important topic in ITS. The extracted camera parameters are typically necessary for estimation of vehicle speed [11], object classification and tracking [12], etc. In [13], camera calibration is also used for resolving vehicle occlusions. However, camera calibration in ITS is different from that in computer vision. In computer vision, accurate camera calibration, which is to fully extract both the intrinsic and extrinsic parameters, mostly requires the use of well-designed patterns (such as a 2D check board [7-8]) to be placed in the field of view of the cameras and multiple images of the pattern from different orientations [14]. However, such patterns are seldom available for camera calibration of traffic scenes in ITS. Therefore, camera calibration in ITS applications usually will not (and can not) calibrate all intrinsic parameters such as focal length, principal point, skew, aspect ratio and radial distortion [14]. In ITS applications, camera calibration of practical traffic scenes mostly considers only one intrinsic parameter, focal length. As shown in [14], with the restriction to consider only one intrinsic parameter (focal length) and to have constant aspect ratio, known principal point, no skew and distortion, the re-projection error is still acceptable. If

an elaborate intrinsic model of all intrinsic parameters is indeed necessary, it can be calibrated a priori as in computer vision applications [5-10][14].

On the other hand, in recent years Pan-Tilt-Zoom (PTZ) cameras [11] have been widely used in ITS applications for monitoring and surveillance purposes and these cameras are usually mounted high above the ground to overlook the traffic scene so that no rotation angle is necessary. As shown in [11], the reduced two-angle model (with pan and tilt angle) has less than 10% bias in distance measurements compared to the full three-angle model even for road grades of 2% .

From the discussion in the above two paragraphs, it can be seen that camera calibration of practical traffic scenes in ITS applications has its own restrictions on available scene patterns, accordingly on intrinsic and extrinsic parameters. Therefore, most camera calibration methods in computer vision are not suitable to practical traffic scenes in ITS applications. Besides, most of those methods involve complex calculations and optimizations to fully extract intrinsic and/or extrinsic parameters. As a result, in the past years simplified methods to perform camera calibration in ITS applications have been proposed. In the following, we briefly review some previous work for camera calibration of traffic scenes in ITS applications and then motivate our work.

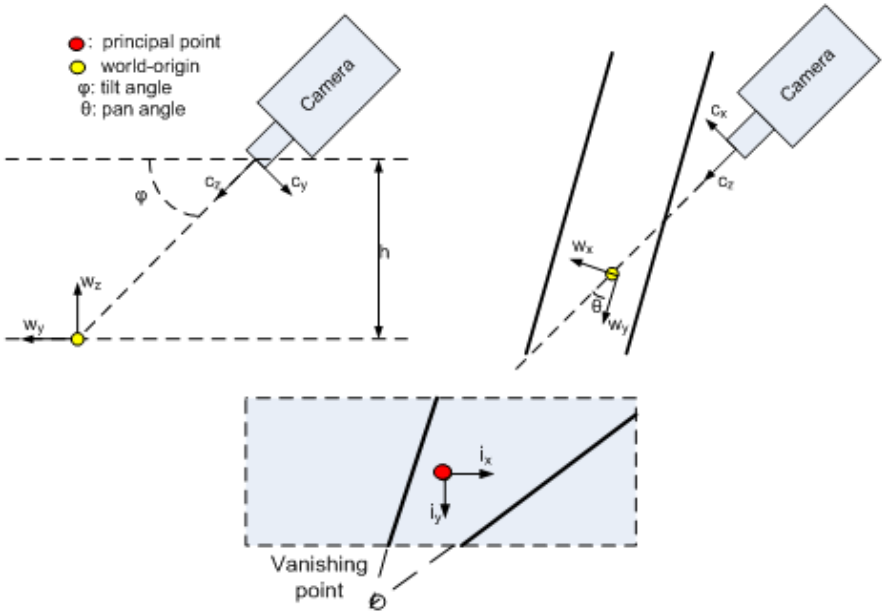


Figure 2.1. Side view and top view of the camera setup for roadway scenes and projection of real-world traffic lanes in the image coordinate.

2.1.1 Camera calibration for roadways

Most previous works on camera calibration in ITS applications have targeted roadway scenes such as highways, in which vehicle moving directions are known. For roadway scenes, the

geometry setup of the camera for calibration is typically described as shown in Figure 2.1 [15-20]. The camera height and the lane width are typically known a priori for camera calibration. The main idea underlying these methods is to use one or two vanishing points from parallel lines to derive closed-form solutions of one intrinsic parameter (focal length) and two or three extrinsic parameters (pan angle, tilt angle and camera height). Some works obtain parallel lines from landmarks like points, lines, poles, or objects with known shape, while other works from moving vehicle trajectories.

The landmarks used in [16-17] are traffic stripe lanes, which are supposed to be parallel in the real world but generally do not appear parallel in the image frames. In some cases, these traffic lanes can be automatically extracted from the images by various image processing algorithms such as the Hough-transform technique [16-17]. Once the traffic lanes are extracted, their intersection coordinates in the image are computed. This intersection point is called a vanishing point, and can be used to estimate intrinsic and extrinsic parameters of the camera. This vanishing point is often computed along the vehicle moving direction.

Extracting traffic lanes to estimate the vanishing point in images may require expensive computational processing. Besides, in practice, traffic lanes may not appear clearly in images depending on the specific traffic scene or road condition. To address this concern, some works propose to track vehicles to find vehicle trajectories [11][18-19], which can be regarded as the equivalents of traffic lanes, and then estimate the vanishing point from the intersection of multiple vehicle trajectories. In [18], focal length, pan angle, tilt angle, and camera height can be found, but mean dimensions of vehicles (width, length) to calculate scale factors on vertical and horizontal axes are required. These techniques can be automated but need to collect data from a large number of image frames.

In some works, two vanishing points are used for camera calibration. The second vanishing point is the intersection of lines perpendicular to the traffic lanes. There are also various techniques to estimate the second vanishing point. In [11], bottom edges of vehicles are used with the assumption that the camera is oriented properly. This technique may not work well if bottom edges of vehicles are distorted or of abnormal shape in case of shadow. In [19], edges of vehicle windows are used, which seem to be more homogenous in vehicles.

In [20-21], the authors suggest a simplified form of camera calibration using a scale factor. This camera model with a reduced number of calibration parameters is claimed to be adequate for making accurate mean speed estimations. But one of the critical underlying assumptions is that the vehicle's moving direction is known.

All the methods discussed above calibrate one intrinsic parameter (focal length) and two to three extrinsic parameters (pan, tilt, and height). Using the vanishing points, these camera parameters can be efficiently derived from closed-form formulas, which is one very important advantage.

In [14], the authors also proposed a very similar method for camera calibration as those discussed above. That method also uses vanishing points to provide an initial estimate of the camera parameters. One improvement of that method is that more geometric primitives (such as normal to plane lines from traffic poles, building edges, and point-to-point distances) are used to allow post-optimization of the camera parameters in contrast to those methods that only rely on

closed-form equations. Also, one more extrinsic parameter (rotation angle) is considered in [14] compared to those methods discussed above.

In summary, previous works on camera calibration of roadway scenes are mostly based on the parallel line method in which vanishing points are estimated. However, these methods are not suitable to some traffic scenes such as roundabouts. On one hand, parallel lines in roundabouts are seldom available, as traffic lanes in roundabouts are not parallel. On the other hand, it is rarely possible to derive parallel lines from vehicle trajectories as vehicles move in a circular fashion. Therefore, a different technique for camera calibration of roundabout traffic scenes is needed.

2.1.2 Camera calibration for roundabouts

Roundabouts are important traffic scenes in ITS. In the case of camera calibration for roundabouts, the most relevant method seems to be the calibration method in [14]. However, the method requires at least two concentric circles for calibration, which is not always possible in roundabout traffic scenes. Also, the method in [14] is very complex especially when deriving the initial solution by first finding a rectifying homography, then estimating focal length and three angles by solving an over-determined linear homogenous system and finally estimate camera height by fitting a model of concentric circles.

We then looked into the literature of computer vision to see whether there are methods that can be suitable to camera calibration of roundabouts in ITS applications. [22-27] propose methods to calibrate the camera using circles. A circle in the real world becomes an ellipse in the image after projection. However, in general the projected circle-center is not the ellipse-center in the image. If the coordinate of the projected circle-center in the image is found, it can then be used for camera calibration. For example, [24] applies a technique that asymptotically locates the projected circle-center in the image using two concentric circles in the real world. This method requires a well-arranged scene (i.e., multiple pairs of coplanar concentric circles) to extract full intrinsic and extrinsic parameters of the camera. However, such a scene is not available in ITS applications. Another disadvantage of this method is that it requires the complete circles in the scene (not just a set of perimeter points). [22] uses both a circle and a few lines that go through the circle-center in the real world to compute the vanishing line, and then estimates the camera's intrinsic parameters. This method also requires an arranged scene and three or more images of the scene for calibration. Also, the lines that go through the circle-center are seldom available in practical roundabout traffic scenes. Besides, it only calibrates intrinsic parameters. Other methods proposed in [23][26] use two arbitrary coplanar non-concentric circles to calibrate one intrinsic parameter (focal length) and extrinsic parameters. Though they require only one image frame, the arranged scene is rarely found in roundabout traffic scenes. The method proposed in [25] uses only one circle, but it requires multiple image frames and some right angles to calibrate intrinsic parameters only. The method in [27] calibrates focal length and extrinsic parameters, but requires two concentric circles and one additional marker.

In summary, one common disadvantage of the above methods is that they mostly require well-arranged scenes. The other drawback is that computation complexity of these methods is usually very high in spite of good accuracy. Hence, these methods are not suitable to real-world roundabout traffic scenes in ITS applications.

2.1.3 Motivation for this work

Recently, much attention has been focused on traffic studies of roundabouts [28-30], which are widely used in rural areas. Some traffic data, such as origin-destination pairs, vehicle waiting time, follow-up time and accepted/rejected gaps for entering roundabouts, are in great need to assess performance, improve safety and design roads at roundabouts. In [28-30], video cameras are used to record traffic. Then the videos are manually analyzed to collect traffic data, which is very laborious, inefficient and costly. To improve efficiency and reduce cost, videos can be analyzed in computers to allow automated traffic data collection, which has been done for highway traffic scenes [12]. For automated traffic data collection using camera-based video systems, camera calibration is always the pre-requisite [11-12].

In this project, we are motivated to propose a new and simple method for camera calibration of roundabout traffic scenes. Our main motivations for this work are two-fold. On one hand, as summarized in the above two Sections, those camera calibration methods discussed in Section 2.1.1 apply only to roadways (including intersections) but not to roundabouts, and those in Section 2.1.2 are not suitable to roundabouts due to the arranged scene requirement and its complexity. On the other hand, a calibration method as simple and efficient for roundabouts as those methods in Section 2.1.1 for roadway scenes is very desirable.

Similar to those methods in Section 2.1.1, the camera parameters to be calibrated are one intrinsic parameter, focal length, and three extrinsic parameters, tilt angle, pan angle and camera height. Calibrating a simplified camera model consisting of these parameters (without roll angle) is usually acceptable for ITS applications in terms of its effect on accuracy of collected traffic data [11]. In fact, the main effect is usually associated with speed estimations, whose accuracy is proportional to the distance measurements (given a frame rate of traffic videos), which is then proportional to accuracy of the adopted camera model. On the other hand, the estimated speeds are then used to facilitate vehicle tracking (for example using a Kalman filter [14]) to derive vehicle trajectories, from which most other traffic data such as vehicle volume, origin-destination pairs, vehicle waiting time, follow-up time and accepted/rejected gaps can then be extracted. However, note that the accuracy of these traffic data depends mainly on the tracking algorithm and the simplified camera model is not likely to introduce significant error.

The proposed method uses only one common landmark, a circle, which is usually available and can be identified, to calibrate the camera for roundabout traffic scenes. We first extract the equation of the ellipse in the image, and then match it with the projected equation of the corresponding real-world circle to derive closed-form solutions of focal length, pan angle, tilt angle and camera height. The method requires only a single circle and one image frame for camera calibration.

2.2 The proposed method for camera calibration

In this Section, we first describe geometry setup of the camera and the projection matrix between the image coordinate and the world coordinate. Then, we discuss the projection of a circle from the world coordinate to the image coordinate. Subsequently, we present fitting of the ellipse in the image. Finally, the method to calibrate the camera is described.

2.2.1 Geometry setup of the camera

Cameras used for video recording and traffic surveillance are generally PTZ cameras and mounted nearby but high above the traffic scene [11][17]. Figure 2.2 shows the side view and top view of the camera setup used in our calibration method. Let (w_x, w_y, w_z) denote the unit vector of the world coordinate, (c_x, c_y, c_z) the unit vector of the camera coordinate, (i_x, i_y, i_z) the unit vector of the image coordinate, φ the tilt angle, θ the pan angle, (a,b) the coordinates of the circle-center in the world coordinate, and finally P the origin of the world coordinate that is projected to the center of the image (or the principal point). Here the pan angle θ is defined between the w_y axis and the line connecting the camera and the circle-center (in the top view).

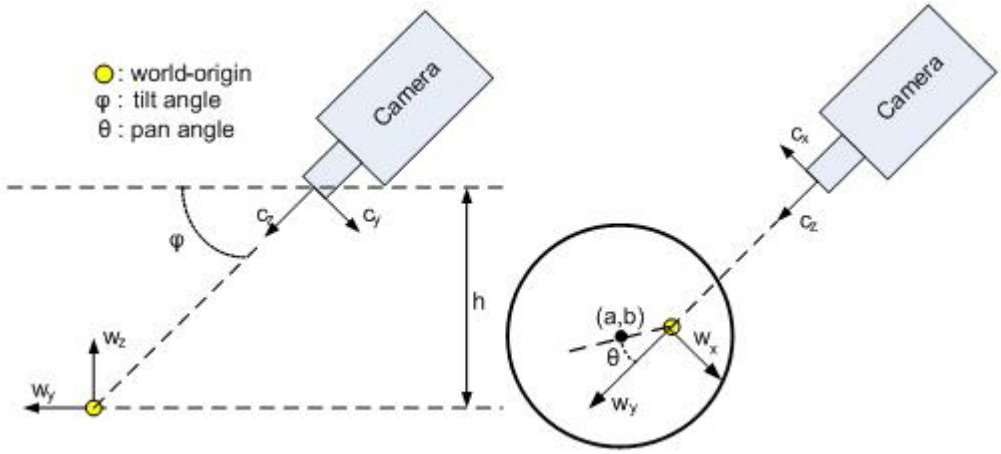


Figure 2.2. Side view and top view of the camera setup used in our calibration method.

The camera's intrinsic parameters in general can be represented by the following matrix K [31-32]

$$K = \begin{bmatrix} f & s & p_x \\ 0 & \tau f & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where parameter f refers to the camera's focal length with unit in pixels. The point (p_x, p_y) is the principal point in the image coordinate, τ the aspect ratio and s the skew factor. In most cases in traffic surveillance applications, we can assume that $(p_x, p_y) = (0,0)$ and $\tau=1$ and the error incurred is typically small [11][17]. The skew factor s is the amount by which the angle between the horizontal axis and vertical axis differs from 90 degrees. This parameter is often admitted to zero because true CCD cameras have x and y axes perpendicular [31].

The rotation matrix, representing the orientation of the camera coordinate with respect to the world coordinate, can be written as

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & sin\varphi & cos\varphi \\ 0 & -cos\varphi & sin\varphi \end{bmatrix} \quad (2)$$

Note that in (2), pan angle θ does not appear as we did not make w_y axis point to the circle-center. But it will be shown later that pan angle is easily found.

With $C = \begin{bmatrix} w_{cx0} \\ w_{cy0} \\ w_{cz0} \end{bmatrix} = \begin{bmatrix} 0 \\ -h/tan\varphi \\ h \end{bmatrix}$ representing the camera's optical center in the world coordinate and the transformation vector $t = -RC$, perspective transformation from the world coordinate to the image planar coordinate can be described as

$$\begin{bmatrix} u \\ v \\ g \end{bmatrix} = K[R|t] \begin{bmatrix} w_x \\ w_y \\ w_z \\ 1 \end{bmatrix}; \quad \begin{bmatrix} i_x \\ i_y \end{bmatrix} = \frac{1}{g} \begin{bmatrix} u \\ v \end{bmatrix} \tag{3}$$

where $[u\ v\ g]$ is the homogenous vector. Substituting *K*, *R* and *t*, (3) can be rewritten as:

$$\begin{bmatrix} u \\ v \\ g \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f.sin\varphi & f.cos\varphi & 0 \\ 0 & -cos\varphi & sin\varphi & -h/sin\varphi \end{bmatrix} \begin{bmatrix} w_x \\ w_y \\ w_z \\ 1 \end{bmatrix} \tag{4}$$

In typical traffic scenes, the depth of images is large because the camera is often mounted high above the ground to have a wide view. For that reason, the height of objects (vehicles or some landmarks) can be neglected to simplify the calculation. With $w_z = 0$, from (3) and (4) we can obtain the transformation equations between the image coordinate and the world coordinate as follows

$$i_x = -\frac{f w_x}{w_y.cos\varphi + \frac{h}{sin\varphi}} \tag{5}$$

$$i_y = -\frac{f sin\varphi.w_y}{w_y.cos\varphi + \frac{h}{sin\varphi}} \tag{6}$$

Solving (w_x, w_y) from (5) and (6), the image coordinate can be mapped back to the world coordinate as follows

$$w_x = -\frac{h.i_x}{f.sin\varphi + i_y.cos\varphi} \tag{7}$$

$$w_y = -\frac{h.i_y}{\sin\varphi.(f.sin\varphi + i_y.cos\varphi)} \tag{8}$$

Equation (7) and (8) reveal three important parameters for camera calibration, which are focal length *f*, tilt angle φ and camera height *h*. Once the circle-center *(a,b)* are found as well, pan angle θ can be derived.

2.2.2 Projection of a circle

General equations of a circle and an ellipse can be written as follows

$$(w_x - a)^2 + \left(w_y - b\right)^2 = R^2 \tag{9}$$

$$i_x{}^2 + 2.H.i_x.i_y + B.i_y{}^2 + 2.G.i_x + 2.F.i_y + E = 0 \tag{10}$$

The point *(a,b)* is the circle-center in the world coordinate and *R* refers to the radius of the circle. The ellipse is characterized by a set of parameters *(H, B, G, F, E)* in the equation. When a circle in the real world is projected to the image plane with perspective projection, it becomes an ellipse in the image [31-32]. As discussed, the projection depends on camera parameters such as focal length, camera height, pan angle and tilt angle. The key is that after projection, a circle originally characterized by (9) in the world coordinate should match the ellipse characterized by (10) in the image coordinate. Based on this principle, substituting (7) and (8) in (9), (9) can be re-written as in (11)

$$h^2.i_x{}^2 + 2h.a.cos\varphi.i_x.i_y + 2.h.a.f.sin\varphi.i_x - f^2sin^2\varphi(R^2 - a^2 - b^2) +$$

$$[2.h.b.f - 2.f.sin\varphi.\cos\varphi.(R^2 - a^2 - b^2)].i_y +$$

$$\left[\frac{h^2}{sin^2\varphi} + 2.h.b.cot\varphi - cos^2\varphi.(R^2 - a^2 - b^2)\right].i_y{}^2 = 0 \tag{11}$$

Then, by matching (10) and (11) we can compute the camera parameters.

2.2.3 Ellipse fitting

In this subsection, we briefly describe how to obtain equation (10). The ellipse in the image can be mathematically characterized by a set of five parameters *(H, B, G, F, E)*. We apply ellipse fitting techniques based on the least-square algorithm to find these parameters [33]. The least-square method chooses the best fit for which the sum of squared residuals has the least value. Let us denote $F(i_x,i_y)$ the distance of a point (i_x,i_y) to the ellipse in the image, which is calculated by substituting (i_x,i_y) into equation (10), the fitting can be approached by minimizing the sum of squared distances from *N* known points on the ellipse as follows:

$$Sum = \sum_{k=1}^{N} F(i_{x_k}, i_{y_k})^2$$

The ellipse equation has five degrees of freedom, hence *N* should be at least equal to five.

2.2.4 Overall camera calibration

With what is described in the above three sub-sections, we can now outline the proposed method for camera calibration of roundabouts. However, we consider two options for the proposed method.

In option one, which is named as the direct solving method, we first fit the ellipse equation from selected points on the ellipse and then derive closed-form equations of the camera parameters based on matching of (10) and (11). In this case, the tilt angle and focal length can be computed as follows

$$\varphi = \sin^{-1}\left(\sqrt{\frac{1}{B + \frac{E.H^2}{G^2} - \frac{2.H.F}{G}}}\right) \tag{12}$$

$$f = \frac{G}{H}\cot(\varphi) \tag{13}$$

If the radius of the circle, denoted as *R,* is known, which is usually the case, then camera height can be solved as

$$h = \sqrt{\frac{R^2.f^2.G^2.cos^2\varphi}{H^2.f^2.(G^2 - E) + (F.G - H.E)^2.cos^2\varphi}} \tag{14}$$

It should be noted that the coordinates *(a,b)* of the circle-center in the world coordinate does not appear in the above equations, and in fact they can be solved as follows:

$$a = \frac{H.h}{cos\varphi} \tag{15}$$

$$b = \left(F - \frac{H.E}{G}\right)\frac{h}{f} \tag{16}$$

Once *(a,b)* are known, the pan angle is

$$\theta = \begin{cases} \tan^{-1}\left|\frac{a}{b}\right| & if\ b > 0 \\ \pi - \tan^{-1}\left|\frac{a}{b}\right| & if\ b < 0 \end{cases} \tag{17}$$

From (12)-(17), we can derive intrinsic and extrinsic camera parameters defined in our camera model.

One possible concern for the direct solving method is that point selections from the ellipse is subject to human error, which statistically gives different solutions among different trials. We will evaluate the reliability of the direct solving method in the next Section. To address this concern, we propose a more robust method based on non-linear optimization, which is the second option. With *N* selected points from the ellipse in the image, we first transform them from the image coordinate (i_{x_k}, i_{y_k}) back to the world coordinate (w_{x_k}, w_{y_k}) by applying

equations (7) and (8). Denoting $D(w_{x_k}, w_{y_k})$ the algebraic distance of a point (w_{x_k}, w_{y_k}) in the world coordinate to the circle as follows

$$D(w_{x_k}, w_{y_k}) = (w_{x_k} - a)^2 + (w_{y_k} - b)^2 - R^2$$

Then the cost function for the optimization problem, which is an error term to be minimized, with *N* points is

$$Error = \sum_{k=1}^{N} D(w_{x_k}, w_{y_k})^2$$

If more geometric primitives such as point-to-point distances are available [14], their re-projection errors can be incorporated in the cost function as well. Note that in either case, the initial solution for the optimization problem could be derived from a trial of the direct solving method to make convergence fast.

2.3 Experiment results

We applied the proposed method to many artificial lab scenes and real-world roundabout traffic scenes and the results for a few of them are shown below.

2.3.1 Lab scene

Our first experiment targets an artificial lab scene as shown in Figure 2.3. The image has a very high resolution of 3000×2300. For comparison purpose, the lab scene has two parallel lines, a check board and a dish plate to represent a circle (denoted in red) as in Figure 2.3. The radius of the plate is 13.5cm. The camera is set up 77cm above ground. The calibration results from eight trials are shown in Table 2.1. In each trial, 15 to 20 points are manually and randomly selected from the ellipse in the image, and then fitted to obtain the ellipse equation which is matched against the circle equation to derive the calibration results. Due to manual selection, there is inevitable error. It was evaluated that the maximum error was 15 pixels (that is, a selected point is at most 15 pixels away from the nearest true ellipse point). Table 2.1 also shows the ratio of standard deviation over mean for each ellipse parameter (*H*, *B*, *G*, *F*, *E*) and camera parameter (*tilt*, *pan, f*, *h*, *a*, *b*). It can be seen that all are very stable thanks to the high resolution of the image in spite of some error incurred in point selections. In particular, note that the distance estimation (for the segment marked *X1* in Figure 2.3) in the last column is very stable too.

We then used the optimization method to calibrate the lab scene. Each camera parameter (*tilt*, *f*, *h*, *a*, *b*) is allowed vary by ±50%. Using the calibration results from any trial as the initial solution, the optimization method gives the same optimal solution as shown in the last row of Table 2.1. This optimal solution is very close to that from the direct solving method. Similarly, the distance estimated from the optimal solution is also very close to that from the direct solving method (as shown in the last Column of Table 2.1). Using this optimal solution, we then estimated the distances of other marked segments in Figure 2.3 and compared them to the ground truth measurements. The comparison results are shown in Table 2.2. The average accuracy of distance estimations is close to 99%.

For the proposed method, it should be noted that in each trial the points are manually selected in the original image using the data cursor in MATLAB [34]. Given the very high resolution of the image, less error would be incurred in point selections and therefore more stable calibration results would be obtained if the ellipse was zoomed in for more careful point selections.

Table 2.3 compares the calibration results between the proposed method, Bouguet's method [34], and the Parallel line method [16]. Bouguet's method gives most accurate calibration results which were almost ground truth measurements. The proposed method has better accuracy than the Parallel line method in terms of not only the camera parameters, but also the distance estimations (for the four marked segments in Figure 2.3).

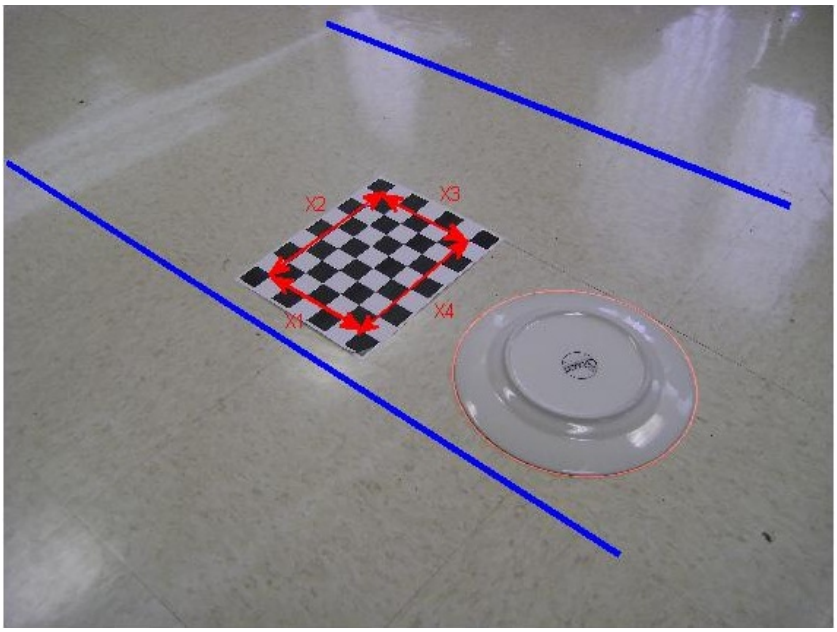


Figure 2.3. An image of a lab scene (resolution 3000×2300).

Table 2.1. Calibration results for the lab scene in Figure 3 (Radius *R=13.5cm*): *h*, *a*, *b* and *Distance X1* are in units of *R*.

Trial	*H*	*B*	*G*	*F*	*E*	*tilt (rad)*	*pan (rad)*	*f (pixel)*	*h (×R)*	*a (×R)*	*b (×R)*	*Dis. X1 / Accu. (×R / %)*
1	-0.16	1.71	-538.7	-337.7	200190	0.81	2.06	3315	5.77	-1.30	-0.69	1.13 / 98.3
2	-0.16	1.70	-536.1	-332.1	196650	0.81	2.06	3188	5.58	-1.30	-0.68	1.13 / 98.3
3	-0.15	1.70	-541.9	-339.9	202120	0.81	2.06	3510	6.11	-1.30	-0.69	1.11 / 99.9
4	-0.16	1.71	-539.7	-335.5	199690	0.81	2.06	3293	5.72	-1.30	-0.68	1.12 / 99.2
5	-0.16	1.70	-536.0	-328.0	195140	0.81	2.05	3127	5.47	-1.30	-0.68	1.13 / 98.3
6	-0.15	1.72	-541.4	-343.4	201500	0.81	2.06	3556	6.15	-1.30	-0.69	1.11 / 99.9
7	-0.15	1.68	-541.5	-331.4	201510	0.82	2.05	3320	5.83	-1.31	-0.68	1.12 / 99.2
8	-0.15	1.70	-539.3	-337.4	199970	0.81	2.06	3344	5.82	-1.30	-0.69	1.12 / 99.2
Std./Me.	**0.04**	**0.006**	**0.004**	**0.015**	**0.012**	**0.004**	**0.005**	**0.04**	**0.04**	**0.002**	**0.005**	**0.007**
Opt.	**-**	**-**	**-**	**-**	**-**	**0.81**	**2.06**	**3464**	**6.0**	**-1.30**	**-0.69**	**1.11 / 99.9**

Table 2.2. Comparison of estimated distances from the calibration results to ground truth distances.

	Figure 3			Figure 4			Figure 5		
Distance	*Est. (cm)*	*Gr. Tru. (cm)*	*Accu.*	*Est. (ft)*	*Gr. Tru. (ft)*	*Accu.*	*Est. (ft)*	*Gr. Tru. (ft)*	*Accu.*
X1	14.98	15	99.9%	22.9	22	96%	25.6	29	88%
X2	20.73	21	98.7%	20.7	22	94%	34.6	36	96%
X3	14.73	15	98.2%	24.6	22	88%	33.5	36	93%
X4	20.76	21	98.8%	25.5	22	84%	8.1	9	90%

Table 2.3. A Comparison of the calibration results from three methods.

	f (pixel)	*tilt (rad)*	*pan (rad)*	*h (cm)*	*Average accuracy of dis. estimations (%)*
Bouguet's method [30]	3200	0.81	2.06	77	100
Parallel line method [12]	3632	0.80	2.07	-	82.6
Proposed method (manual)	3464	0.81	2.06	81	98.9
Proposed method (automatic)	3075	0.78	2.05	72.4	96.8

2.3.2 Roundabout traffic scene

A real-world roundabout is shown in Figure 2.4 and the landmark of the circle is visible. This roundabout is located in Cordata Parkway in Bellingham, Washington. The images were captured by a surveillance camera nearby the scene and the resolution of the image is 400×300. The radius of the roundabout is 44 feet and camera height is not known. To calibrate the camera, we first experimented with the direct solving method by manually selecting 10 to 25 different ellipse points each time for many trials. It was also evaluated that the maximum error for the selected points was 4 pixels. Table 2.4 lists the results of ellipse parameters (*H*, *B*, *G*, *F*, *E*) and camera parameters (*tilt*, *pan, f*, *h*, *a*, *b*) for eight trials. In Table 2.4, we also show the ratio of standard deviation over mean for each parameter. It can be seen that some ellipse parameters (such as *H*) and camera parameters (such as *f* and *h*) can be subject to relatively large variation up to 20%. However, the distance estimation (for the lane width marked *X1* in Figure 2.4) using the camera parameters is very stable across the trials. In fact, it is not only stable, but also very close to the actual distance of 22 feet with at least 92% accuracy.

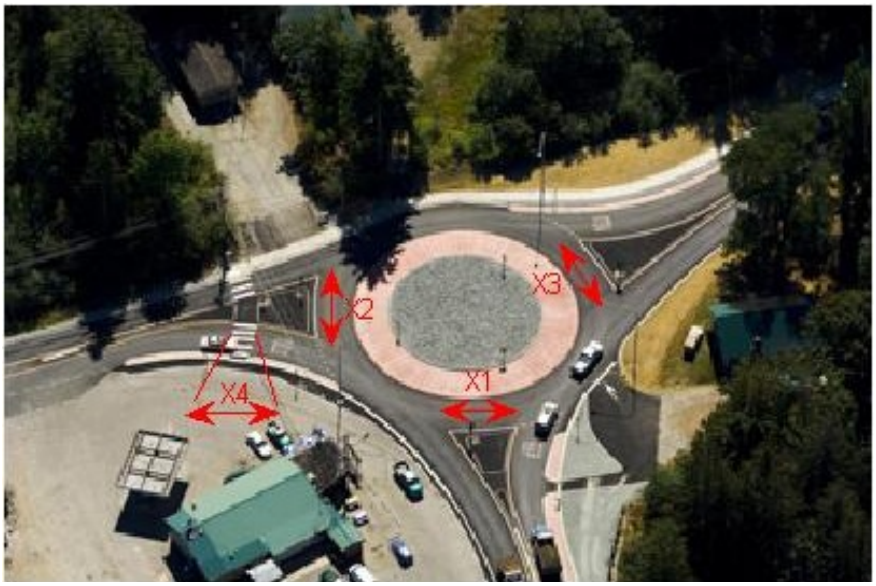
We then used the optimization method for camera calibration. We used calibration results from the direct solving method in each trial as the initial solution and allowed *f*, *h*, *a*, *b* and *tilt* to vary by ±50%, and the optimal solutions ended up with the same one as shown in Table 2.4 (each optimization run takes less than a quarter minute). Note that this optimal solution gives the estimated distance very close to those obtained from the direct solving method and has 96% accuracy compared to the ground truth. With this optimal solution, we also estimated the distances of other lane widths as marked in Figure 2.4 and compared them to ground truth measurements and the average accuracy is 90.5% as shown in Table 2.2.

Figure 2.5 shows another roundabout at the intersection of SR9 and SR538 located in Skagit County, Washington. The radius of the roundabout is 50 feet. The ellipse parameters and camera parameters from eight trials using the direct solving method is show in Table 2.5 (the maximum error for the selected points was 4 pixels). It can be seen that some parameters (such as *H*, *f*, *h*, and *b*) are subject to relatively large variations. However, the distance estimation (for

the road width marked *X1* in Figure 2.5) is very stable, similar to the previous case. We then tried the optimization method for optimal camera parameters. Again, we used calibration results from the direct solving method in each trial as the initial solution and allowed *tilt*, *f*, *h*, *a*, and *b* to vary by ±50%, and the optimal solutions ended up with the same one as shown in Table 2.5 (each optimization run takes less than a minute). With this optimal solution, we estimated the distances of four road segments marked in Figure 2.5 and compared them to ground truth measurements. The average accuracy is 92% as shown in Table 2.2.

Google Maps - ©2012 Google

Figure 2.4. An image of the roundabout (resolution 400×300).



Google Maps - ©2012 Google

Figure 2.5. An image of the roundabout (resolution 350×232).

Table 2.4. Calibration results for the roundabouts in Figure 4 (Radius *R=44feet*): *h, a, b* and *Distance X1* are in units of *R*

Trial	*H*	*B*	*G*	*F*	*E*	*Tilt (rad)*	*pan (rad)*	*f (pixel)*	*h (×R)*	*a (×R)*	*b (×R)*	*Dis. X1 / Accu. (×R / %)*
1	0.17	3.49	33.94	63.95	-1914.8	0.64	0.66	274.60	2.32	0.48	0.62	0.46 / 92
2	0.14	3.43	34.18	63.17	-1852.7	0.63	0.69	334.00	2.85	0.50	0.60	0.49 / 98
3	0.13	3.44	33.57	62.70	-1929.6	0.63	0.69	343.73	2.92	0.49	0.60	0.50 / 100
4	014	3.39	33.59	61.37	-1955.5	0.63	0.68	330.24	2.83	0.49	0.60	0.49 / 98
5	0.16	3.35	34.29	62.70	-1818.5	0.65	0.67	287.17	2.50	0.49	0.62	0.47 / 94
6	0.12	3.43	33.35	62.18	-1952.8	0.62	0.69	391.70	2.33	0.49	0.59	0.52 / 96
7	0.14	3.42	33.54	62.14	-1903.1	0.63	0.68	333.27	2.86	0.49	0.60	0.49 / 98
8	0.15	3.38	33.87	61.81	-1887.2	0.64	0.68	307.49	2.65	0.49	0.61	0.48 / 96
Std./Me.	**0.1**	**0.012**	**0.01**	**0.013**	**0.025**	**0.014**	**0.01**	**0.12**	**0.11**	**0.009**	**0.018**	**0.037**
Opt.	**-**	**-**	**-**	**-**	**-**	**0.64**	**0.68**	**308.60**	**2.64**	**0.49**	**0.61**	**0.48 / 96**
Auto.	**0.12**	**3.38**	**33.44**	**61.27**	**-1993.0**	**0.63**	**0.69**	**378.7**	**3.23**	**0.49**	**0.59**	**0.51 / 98**

Table 2.5. Calibration results for the roundabouts in Figure 5 (Radius *R=50feet*): *h, a, b* and *Distance X1* are in units of *R*

Trial	*H*	*B*	*G*	*F*	*E*	*tilt (rad)*	*pan (rad)*	*f (pixel)*	*h (×R)*	*a (×R)*	*b (×R)*	*Dist. X1 / Accu. (×R / %))*
1	-0.052	1.81	-11.49	-15.37	-1852.9	0.81	1.99	211.36	3.41	-0.26	-0.11	0.50 / 88
2	-0.054	1.85	-11.35	-15.87	-1897.6	0.80	1.98	207.89	3.28	-0.25	-0.11	0.49 / 86
3	-0.060	1.76	-10.93	-14.53	-1873.7	0.82	1.85	168.56	2.76	-0.24	-0.07	0.48 / 84
4	-0.041	1.80	-11.27	-15.17	-1871.3	0.82	2.07	258.83	4.18	-0.25	-0.14	0.52 / 91
5	-0.051	1.82	-11.14	-15.11	-1904.6	0.81	1.96	208.19	3.32	-0.25	-0.10	0.49 / 86
6	-0.040	1.78	-11.34	-15.37	-1848.9	0.82	2.09	265.14	4.32	-0.25	-0.14	0.52 / 91
7	-0.048	1.75	-11.10	-14.44	-1857.3	0.83	1.97	212.67	3.51	-0.25	-0.11	0.51 / 89
8	-0.060	1.80	-11.13	-14.95	-1892.8	0.80	1.87	175.39	2.82	-0.25	-0.08	0.48 / 84
Std./Me.	**0.15**	**0.019**	**0.016**	**0.031**	**0.011**	**0.014**	**0.08**	**0.16**	**0.16**	**0.015**	**0.24**	**0.016**
Opt.	**-**	**-**	**-**	**-**	**-**	**0.81**	**1.98**	**220.02**	**3.54**	**-0.25**	**-0.11**	**0.50 / 88**
Auto.	**-0.05**	**1.80**	**-11.32**	**-15.10**	**-1882.3**	**0.81**	**1.98**	**215.71**	**3.47**	**-0.25**	**-0.11**	**0.50 / 88**

Another real-world roundabout is shown in Figure 2.6 and the landmark of the circle is shown in red. The images were captured by a surveillance camera nearby the scene. This roundabout is located in Cottage Grove, Washington County in Minnesota. Traffic speed limit, radius of the roundabout, and camera height are respectively 25 miles per hour, 110 feet, and 51 feet. Using the proposed method for camera calibration, the estimated tilt angle, focal length, and camera height are shown in Table 2.6.

The estimated camera height (52.3 feet) is very close to the measured camera height (51 feet) with an error less than 2.5%. With these parameters and the known frame rate of the video, we estimate the speed of each vehicle (see Table 2.7) in this roundabout to be in the range of 10-35mph, which agrees with the speed limit well, though in this case no ground speed measurements are available for comparison. Also, we measured several marked distances in the real world and compared them with the distances estimated using the calibration results. The comparison results are shown in Table 2.8 and the accuracy is about 93%.

From the above three experiments, it can be concluded that (1) the direct solving method can give stable and relatively accurate distance estimations (thus vehicle speed estimations) in spite of large variations of ellipse parameters and camera parameters; (2) the optimization method is effective to produce an optimal solution, but the direct solving method is still useful as it provides a reasonably good initial solution for optimization which makes convergence fast.

Figure 2.6. An image of the roundabout (Figure is provided from MnDOT video cameras).

Table 2.6. Calibration results of the roundabouts in Figure 2.6.

	φ (rad)	*f (pixel)*	*h (feet)*
Roundabout in Figure 2.6	0.318	240	52.3

Table 2.7. Estimation of vehicles' speed in Figure 2.6.

Vehicle #	1	2	3	4	5	6	7	8
Speed (mph)	35	12.1	24.6	27.8	29.8	22.3	15.2	15.2

Table 2.8. Estimation of distances in Figure 2.6.

Distance	AB	CD	EF
Ground truth	32 ft.	14 ft.	8 ft.
Estimated	31.6 ft.	13 ft.	7.7 ft.

Based on our experience, we recommend the following empirical rules to achieve accurate calibration results: (1) the ellipse is clearly visible for point selections; (2) tilt angle of the camera is preferred to be relatively large so that the ellipse contour has enough depth; (3) the selected ellipse points are preferred to cover uniformly the ellipse. In other words, selected points concentrating on a segment of the ellipse may cause larger error to the fitted ellipse equation; (4) the resolution of the picture is preferred to be large, because rounding of the pixels of the selected points would statistically lead to less accuracy loss to the fitted ellipse equation and marked distance. This partially explains why the lab scene in Figure 3 has ellipse parameters, camera parameters and distance estimations all very stable compared to Figure 2.4 and Figure 2.5.

The proposed calibration method based on a single circle can be considered as one of the circle-based calibration methods discussed in Section 2.1.2. The circle-based calibration methods that are closest to the proposed one are those from [14] and [27]. However, both methods in [14] and [27] require at least two concentric circles and that in [27] requires one additional marker. While two concentric circles may be available in some roundabout traffic scenes (such as in Figure 2.4), they are not always available or reliable (for example in Figure 2.5, the inside circle in white is not reliable). In our method, we require only a single circle, which relaxes the scene requirement at the cost of calibrating one extrinsic parameter less (which is the roll angle) than those methods. However, this reduced camera model which is widely used in camera calibration of roadways as discussed in Section 2.1.1, has also shown to be practically acceptable in terms of accuracy of distance estimations in our experiments of roundabouts. With this reduced camera model, the proposed method takes full advantage of the fitted ellipse equation to drive closed-form solutions of calibration parameters, which is much simpler than those methods in [14] and [27].

2.3.3 Automatic camera calibration

In the above experiments, the points have been manually selected for ellipse fitting. As in previous works for traffic lane extraction [16-17], ellipse can also be automatically extracted. First, we convert the color image to grayscale and perform Sobel edge detection. The result is a binary image with edge points. With these edge points which may not all belong to the ellipse, we used the Chord-Tangent method in [35] for automatic ellipse fitting. Once the ellipse

equation is obtained, the direct solving method can be applied for camera calibration and the optimization method can follow as well.

For comparison, the calibration results in the automatic mode for the experiments on the lab scene in Figure 2.3 and the real-world roundabouts in Figure 2.4 and 2.5 are shown in Table 2.3, 2.4 and 2.5 respectively. It can be seen the calibration results, in particular the distance estimation, between the manual mode and the automatic model agree well. However, it should be noted that camera calibration in the automatic mode has the disadvantage that unclear (or poorly legible) ellipse may have significant negative effect on the accuracy of ellipse fitting. Currently, we provide both manual mode and automatic mode in our software implementation in MATLAB and let the user decide which option to use.

In case the ellipse is not clear in the image (as the landmark of the circle is not clear in real-world roundabout scenes), we may track the vehicles in the roundabout scene to obtain the vehicle trajectories, which can then be regarded as the equivalents of ellipse . This idea has been also used in previous works on camera calibration for roadway scenes when parallel traffic lanes are not clear in the image [11][18-19], however this is more challenging for roundabouts.

2.4 Summary and conclusion

As this project targets a video-based traffic performance measurements system for roundabouts and intersections, camera calibration is required for roundabout traffic scenes. While camera calibration for intersection is considered as a solved problem using previous methods based on parallel traffic lanes used in roadways, that for roundabouts is not. Therefore, in this project, we propose a new and simple method for camera calibration of roundabout traffic scenes in ITS applications by using a landmark of a circle. The proposed method extracts the ellipse equation from the image and matches it with the projected equation of the corresponding circle in real-world traffic scenes. The proposed method is validated using images of both lab scenes and real-world roundabout scenes. The estimated distances using the obtained calibration results are very comparable to ground truth measurements with more than 90% accuracy on average in all our experiments. Another advantage of the proposed method is that it requires only a single circle and one image for camera calibration, which is very convenient to use for traffic engineers.

The calibration results can be used further to estimate vehicle speed and facilitate vehicle tracking to derive vehicle trajectories for roundabout traffic data collection.

In our algorithm, we assume that the principal point is the image origin (p_x=0, p_y=0). To evaluate the stability of results regarding principal point's position, we recomputed camera parameters given different positions of principal points. For example, when (p_x=10, p_y=10), the result show that tilt angle, focal length, and camera height vary by 1%, 5%, and 4% respectively. Those variations are relatively small and in acceptable ranges.

3 Vision-based Tracking System

Following camera calibration, vehicle tracking is the next main module in the overall traffic performance measurements system in this project. Vision-based vehicle tracking systems have been used widely due to their advantage to provide the most comprehensive information about the vehicles compared to loop-detectors or radars. Though videos captured could be manually inspected for traffic performance measurements, it is very costly and laborious. Recent research work focuses on automatically collecting traffic performance measurements from video processing. There are relatively mature data acquisition technologies available for highways, but automatic data collection from roundabouts and intersections presents unique challenges because of more complex traffic scenes, data specifications and vehicle behavior. In this chapter, we propose a tracking-based automated traffic data collection system dedicated to roundabouts. This system could also be applied to intersections and highways with slight modifications.

The proposed vehicle tracking scheme has three main steps of processing. First, the system uses an enhanced Mixture of Gaussian algorithm with shaking removal for video segmentation, which can tolerate repeated camera displacements and background movements. Next, Kalman filtering, Kernel-based tracking and overlap-based optimization are employed to track the vehicles occluded and to derive the complete vehicle trajectories. The resulting vehicle trajectory of each individual vehicle gives the position, size, shape and speed of the vehicle at each time moment. Finally, a data mining algorithm is used to automatically extract the interested traffic data from the vehicle trajectories.

In the rest of the Chapter, we first give an introduction of previous methods for vehicle tracking. Then, in Section 3.2, we give an overview of the proposed method. In Section 3.3, we discuss the vehicle segmentation method used in vehicle tracking, followed by the vehicle tracking algorithm in Section 3.4. Finally, some experimental results are shown in Section 3.5.

3.1 Introduction

Traffic data collection is a very important task in transportation applications as it provides data necessary for traffic simulation, modeling and performance evaluation. While traffic data can be manually collected, automating this task is vital in reducing cost and improving efficiency. In literature, there has been a significant amount of work on automated traffic data collection for highways. Those systems developed for highway scenes are relatively mature and successful thanks to relatively simple vehicle behavior and simple traffic data specifications. Some representative work can be found in references [36-41]. We will not go into detail for the developed systems for highways. Instead, we focus on systems that are developed for roundabouts, as these traffic scenes are more challenging due to complex scene characteristics, data specifications and vehicle behavior. For example, camera calibration techniques have not been well developed for roundabout traffic scenes while mature for highway scenes. Also, existing traffic data collection for highways target mostly vehicle volume, vehicle speed and lane use, but for roundabouts other measurements such as origin-destination pairs, waiting time and gap size are needed as well. Besides, vehicle behavior at roundabouts tends to be more complicated, as it involves frequent acceleration/de-acceleration, waiting and turning.

A variety of methods have been proposed to use sensors, loop inductors, radars or cameras for traffic data collection. Recently, in [42] an approach using wireless sensor networks is proposed for traffic data estimation for intersections. For each way of the intersection, a sensor is placed in each lane. When a vehicle drives through the sensor, its timing is simply recorded in the sensor's log. Then, all the logs of the sensors are analyzed offline to estimate the vehicle volumes of each turning direction. While the detection part could be accurate using sensors, however, a number of issues exist with this approach. First, there is ambiguity in the process of analyzing the logs. As shown in [42], depending on the specific timings of the vehicles, some vehicle trajectories could not be correctly classified. Second, the major limitation of the system is that it can only estimate traffic data like vehicle turning volumes and waiting time. To estimate the vehicle speed, more sensors are needed in each lane. However, even in that case, it is not possible to track the acceleration/deceleration behavior of the vehicle. Other traffic data, such as gap size and lane use, cannot be handled either. A very similar project and developed system based on detection using sensors is also reported in [43]. Another commercial system, Miovision [44], is available today for traffic data collection for intersections; however it is restricted to vehicle count. It should be noted that the Miovision system uses cameras instead of sensors.

While various sensor technologies are available, visual information provided by cameras can potentially provide comprehensive and accurate traffic data with non-intrusiveness, easier management, lower cost and higher efficiency. In this chapter, we present a tracking-based data collection system that can address the above issues with the existing data acquisition technologies.

3.2 System overview

The proposed traffic data collection system is implemented as software that runs on a regular PC. It incorporates powerful image processing algorithms to allow accurate traffic data collection. The data collection system has mainly three processing steps, segmentation to identify vehicles, tracking vehicles to derive the trajectory of each vehicle, and the final step of data-mining to extract traffic data. This Chapter focus on the first two steps of the system.

Vehicle segmentation is the first processing step in the data collection system. There are many existing methods for vehicle segmentation, such as background subtraction, running average, texture-based method and Eigen-background method [45-51]. However, one main disadvantage of these methods is that they do not cope well with camera shaking and background movements, which happen quite often in practice. Another method to build background models is to model each pixel with a Gaussian distribution. To cope with repeated camera shaking, the proposed system applies a mixture of Gaussian distributions for each pixel [52-53]. To further remove false detections from camera shaking, a shaking-removal algorithm is applied to refine the segmented objects.

The resulting outputs from vehicle segmentation are binary objects that are potentially vehicles. After segmentation, tracking is the next step to detect and track vehicles in each time frame. The goal is to obtain the complete vehicle trajectory, as it gives comprehensive information about the state of the vehicle at each time moment, which is more accurate for traffic data collection than without trajectories. A number of methods exist for tracking. Some popular approaches are

region-based tracking [36], contour-based tracking [54], model-based tracking [55], and feature-based tracking [56]. Our approach is region-based tracking with combined Kalman filtering, Kernel-based tracking and overlap-based optimization techniques for handling vehicle occlusions.

Finally, in the last step a data mining algorithm is applied to extract all interested traffic data from the vehicle trajectories. This step does not incur any accuracy loss on traffic data given vehicle trajectories. Such a traffic data collection system relieves traffic engineers from the laborious job of manual data collection as in [29-30], thus can significantly reduce cost and improve efficiency. The data mining algorithm will be covered in Chapter 4.

The next section describes the implementation of the proposed data collection system in detail, followed by results and conclusions.

3.3 Vehicle segmentation

Segmentation is the first step for traffic surveillance. The better the result of this step is, the easier and more accurate vehicle tracking is. We first briefly review some previous methods and then introduce a segmentation method based on background modeling in which each pixel is consider as a mixture of Gaussian.

3.3.1 Existing approaches

There are many methods for background modeling. The beginning idea is calculating background based on the history of pixel values. The average background simply takes the average of N previous frame, and when a new frame comes, it will replace the oldest frame. Other similar approaches take median or mode value instead of average. Background of those approaches can be defined as

$$B_N = \text{average, median or mode } \{\mathrm{I_k}|\ \mathrm{k} = 0, 1 \ldots \mathrm{N}\}$$

where I_k is the intensity at frame k. Those methods run very fast, but they also require a lot of memories: N * size (frame). To reduce memory consumption, the running average method simply adapts background based on incoming frame with a certain learning rate:

$$B_N = (1 - \alpha) * B_{N-1} + \alpha * I_N$$

where B_{N-1} is the background at frame N-1, I_N is pixel intensity value at frame N, and α a learning rate. When a new frame comes, we take α percent of its intensity and (1-α) percent of existing background as the new background. A large value of α means that the model will update faster, typically α= 5% (0.05). An example of pseudo code for this algorithm could be found in Table 3.1.

There is also a combined method like in [50], choosing between running mode and running average base on a scoreboard algorithm. The mechanism to update background can be modified from blind update to selective update:

$$B_N = (1-\alpha) * B_{N-1} + \alpha * I_N * M$$

with M=1 if the pixel is classified as background, and 0 if foreground.

[51] uses another advanced approach, eigenspace, to form the background. M eigenvectors corresponding to M largest eigenvalues are kept when applying Principle Component Analysis (PCA) on a sequence of N images. Background pixels, which appear frequently in frames, will significantly contribute to this model while moving objects do not. [57] is an modified version of eigenbackground modeling, which run faster because the decomposition step is eliminated in updating procedure.

Table 3.1. Pseudo code for the running average algorithm.

```
%initialize
read the first frame
create a matrix with the same size as the frame to represent background
for all pixels in the frame
    background = current frame;
end for

%processing
for all frames in the video
    read new frame
    for all pixels in the frame
        if |current frame - background| > threshold
            set the pixel as foreground;
        end if
        background = (1-alpha)*background + alpha*I_N;%update
    end for
end for
```

Texture property of the image also could be used for segmentation. [49] uses autocorrelation difference between two image blocks to compare their similarities. This method also requires an empirical threshold which is not explicit and varies through videos. Moreover, the computational workload for autocorrelation is high.

The above-discussed methods have some common disadvantages. They do not provide an explicit way to choose the threshold for segmentation. Also, they do not cope well with camera shaking (repeated or occasional shaking due to wind) and background movements (such as trees, grass).

3.3.2 The Mixture-of-Gaussian approach

In our implementation, the variation of each pixel across time is modeled by a mixture of 5 Gaussian distributions (MoG) and each distribution has its own mean, standard deviation, and weight [52-53]. The MoG actually models both foreground and background. The first *B* (*B*<=5) distributions are chosen as the background model. A threshold *T* is defined to represent the portion of the data that should be accounted for the background model:

$$Background = min_B \left(\sum_{k=1}^{B} w_k > T \right)$$

where w_k is the weight of distribution k (k=1,2,3,4,5). We normalize the weights such that:

$$\sum_{k=1}^{5} w_k = 1$$

For a new image frame, each incoming pixel will be subject to a matching test to see whether it belongs to any existing distribution of that pixel or not. In our project, we apply confidence intervals for matching. We choose the confidence interval to be 98%, so an incoming pixel X will belong to the distribution D(μ,σ) if:

$$p = \frac{|X - \mu|}{\sigma} \leq 2.5$$

Background model update is an important step to keep the background models up-to-date with environment changes such as illumination and scene changes. If none of 5 Gaussian distributions matches the incoming pixel, the update will simply replace the distribution with the lowest weight by a new distribution with the same weight, a large standard deviation and the mean equal to the incoming pixel. If there is one distribution matching the pixel, its weight, mean and standard deviation will be updated as follows [52]:

$$w_k = w_{k-1} + \alpha * (1 - w_{k-1})$$

$$\mu_t = (1 - \alpha)\mu_{t-1} + \alpha * X_t$$

$$\sigma_t^2 = (1 - \alpha)\sigma_{t-1}^2 + \alpha * (X_t - \mu_t)^2$$

whereas the mean and standard deviation for other distributions are kept the same except their weights which are reduced as follows:

$$w_k = w_{k-1} + \alpha * w_{k-1}$$

where α is a learning rate. In case there are more than one distributions that match the pixel, the distribution which has smallest p will be chosen for update.

MoG can model fast illumination changes in the scene. Another advantage of MoG is its robustness against repeated camera shaking, which is one very practical problem that affects tracking accuracy. As the obtained images are not stable, large regions of noisy strips are segmented from the image, which severely affects the accuracy of object extraction. MoG can fight with repeated camera shaking very well, as the pixel changes caused by repeated camera

shaking are recorded in the background distributions and are therefore not identified as false segmentation regions any more.

On the other hand, when the camera shakes to specific positions which have not yet been observed before, many background pixels may get pixel values not modeled before and be falsely classified as foreground while they are actually from nearby background. Those false detections could also happen if the camera shakes only occasionally. In that case, even the pixel values are modeled in some distributions before, but their rare observations make their weights too small to be considered in background models. Therefore, to suppress these false detections and further refine the detection results, we employ a shaking-removal step. This is based on the observation that the false detection regions will have a high probability to be a part of the background distributions at its original locations. Hence, we can decide whether a detected region is caused by a real foreground object or the background by considering the background distributions in a small neighborhood of the detection region. If the detection pixel matches with the background distributions of pixels in the neighborhood, it is highly possible that this detection is false caused by camera shaking. When considering color images with the RGB (Red, Green, Blue) space, this technique proves to be very effective as the probability of mismatching is very small [53]. In our implementation, we choose a square 5x5 window for shaking removal.

The combined MoG modeling of the background and the shaking-removal algorithm turn out to be very valuable as video cameras are subject to shakings in practice, which would significantly affect segmentation quality if not treated. The proposed method is very general and does not involve manual intervention or rely on any prior scene knowledge. The pseudo code for proposed method using MoG could be found in Table 3.2.

Table 3.2. Pseudo code for the MoG algorithm for vehicle segmentation.

```
%Initialize
read the first frame from the video to get the size of images
create a matrix M with the same size as the frame
for all elements in M
    for all distributions
        mean=initial mean;
        variance=initial variance;
    end for
end for
%Processing
for all frames of the video
    %segmenting
    for all pixel in the frame
        for all distributions
            t=sum((mean-newcoming_pixel)^2-(2.5*var)^2);%matching
            if t<0
                the new pixel matches one or more distributions;
            end if
        end for
        if there is no match
            the pixel is recognize as foreground;
        end if
        if there is one or more matches
            find the match that has smallest t value;
            check whether that match belongs to the background model;
            if no
                set pixel as foreground;
            end if
        end if
    end for
    %shaking remove
    for all pixel in the frame
        if foreground
            for neighbor pixels
                take the distribution that has largest weight;
                t=sum((mean-newcoming_pixel)^2-(2.5*var)^2);%matching
                if t<0
                    match found, reset pixel as background;
                end if
            end for
        end if
    end for
    %tracking
    tracking %see more detail in tracking pseudo code
    %updating
    for all pixel in the frame
        if not belong to a vehicle
            update mean, variance, weight;
        end if
    end for
end for
```

3.4 Vehicle tracking

3.4.1 Object and vehicle model

After segmentation, results are binary objects in the image. Objects are then extracted by the connected component algorithm [36], and classified as vehicles if their sizes are large enough. To well represent objects, each object is characterized by both a rectangular box with width and length to bind the object, and a contour with the detailed shape of the object. An object with above descriptions would be associated with one or more vehicles. A vehicle is detailed by states, predicted states, Kalman parameters, a start frame, the frame when the vehicle is first detected, and an end frame, the last frame when vehicle still could be detected. A state vector, denoted as [x y v_x v_y] is associated with each claimed vehicle, where (x,y) gives the center position of the vehicle and (v_x v_y) the velocity along the image coordinates (x,y). Figure 3.1 shows how an object and a vehicle are represented in our system.

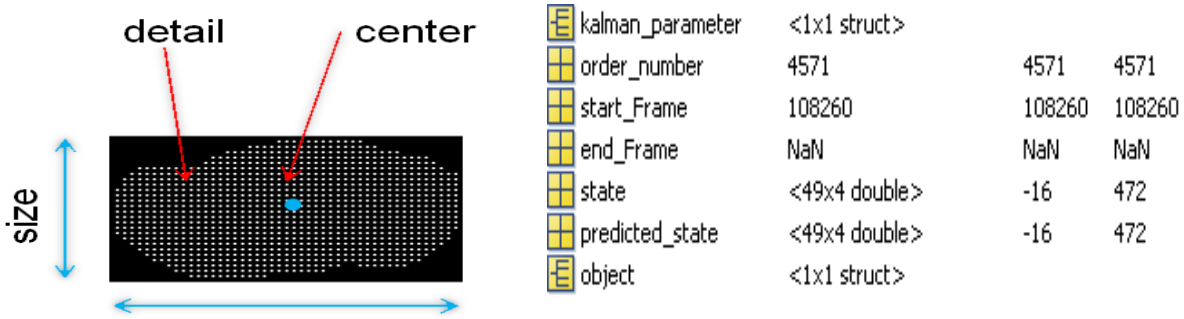


Figure 3.1. Left: object's description; right: vehicle's hierarchy.

To relate objects to vehicles, we compare the areas of overlap between the objects and vehicles in the image frame. Vehicles' speed or camera's frame-rate affects the areas of overlap. For example, a fast-moving vehicle or low frame-rate camera could cause segmented objects in the current and previous frame of the same vehicle to have no overlap at all. And then the system will create new vehicle description for the current segmented object while report the previous vehicle missing. To address this problem, we use Kalman filter [58] to estimate the state of vehicles in the next frame based on their current states and Kalman parameters, and then compare the areas of overlap between the segmented objects and predicted vehicles in the current image frame.

3.4.2 Kalman filter

The Kalman filter is a set of mathematical equations that provides a recursive solution of estimations of past, present, and future states. We use the Kalman filter to predict the state of a vehicle in the next time frame and save it in predicted state vector. The state vector is then updated and tracking is repeated in the next image frame. The state vectors at all times are recorded to derive the complete vehicle trajectory since its detection until exit.

Due to prediction errors, objects and predicted vehicles could slightly overlap even though they are just close to each other. In this case, one object would be associated with multiple vehicles or vice versa. To accurately track vehicles, we break those associations by considering the percent of overlap. If an object and a vehicle overlap by more than a certain threshold, say 80%, this association is recognized as a "strong match". Otherwise, they have a "loose match". The priority is given to strong matches. If an object and a vehicle has a strong match, all loose matches associated with this object and this vehicle will be cut off. Then, we try to break all multiple-object to multiple-vehicle associations by breaking the association that has the lowest overlap percentage. In this way, object-to-vehicle associations are either one-to-one, multiple-to-one or one-to-multiple, which simplify the tracking process.

Moreover, when vehicles have a long stop, the system will gradually model the vehicle itself in the background. This leads to the case that many scattered segmented objects are associated with one vehicle. A simple solution is grouping all scattered objects to create a new object associated with that vehicle. A pseudo code for overall tracking system can be found in Table 3.3.

Table 3.3. Pseudo code for the tracking algorithm.

```
%Initialize
create an empty list of vehicle and object

for all frame in the video
    segment foreground using MoG %see more details in MoG pseudo code
    apply median filter, holes filter;
    label and group connected components to objects;
    %check overlap
    for all objects
    for all vehicles
        check overlap between objects and vehicles;
        if percent of overlap > 0.8
            set a strong match and a loose match;
        else set only a loose match
        end if
    end for
    end for
    %break some associations which have multiple matches
    for all objects
    for all vehicles
        if there is a strong match
            break all other matches for the current vehicle and object;
        else
            if there are multiple matches
                if multiple objects match with one vehicle K
                    for all those objects
                        if there is another match with other vehicles
                            break the match with vehicle K;
                        end if
                    end for
                end if
                if multiple objects match with multiple vehicles
                    break the match with the least overlap percentage;
                end if
            end if
```

```
        end if
    end for
    end for
    %update vehicles' state
    for all objects
    for all vehicles
        if the match is one-one
            update the vehicle with the object;
        end if
        if multiple objects match with one vehicle k
            group all objects together to create a new object;
            update the vehicle with the new object created;
        end if
        if multiple vehicles match with one object
            apply kernel-based tracking to locate each vehicle;
            create a new object for vehicle;
            update vehicles with corresponding new objects;
        end if
    end for
    end for
    %update vehicles/objects that have no associations
    for all objects
    for all vehicles
        if there is no loose match
            compute potential match;
            if potential match > threshold
                   update vehicles with corresponding objects;
            else
                   update vehicle as missing;
                   create new vehicles for objects;
            end if
        end if
    end for
    end for
    create a current vehicle list;
end for
```

3.4.3 Kernel-based tracking in joint feature-spatial spaces

In a complex or crowded traffic scene, vehicles can partially or fully occlude each other. The occlusion makes it difficult to track the individual vehicle. Kalman filtering could help when vehicles are occluded for a short time, but if vehicles are occluded for a long time and have dramatic velocity changes while occluded, then errors could happen. To deal with this problem, we apply Kernel-based tracking in joint feature-spatial spaces [59-60] and overlap-based optimization when occlusion is detected. Given sample points $\{x_i, u_i\}_{i=1}^{N}$centered at $\hat{x}$ in the model image, and $\{y_j, v_j\}_{j=1}^{N}$ centered at $\hat{y}_0$ in the current target image, the kernel-based tracking with the Gaussian kernel is [59]:

$$\hat{y} = \frac{\sum y_i f(y_i)}{\sum f(y_i)}$$

where

$$f(y_i) = \sum e^{-(x_i-\hat{x})^2/\theta^2} e^{-(u_i-v_j)^2/h^2} e^{-(y_j-\hat{y}_0)^2/\theta^2}$$

where σ and h are the bandwidths in the spatial and feature spaces.

The model of the vehicle is taken before occlusion. The initial center to be applied in Kernel-based tracking is the predicted position of the vehicle provided by Kalman filtering.

To further improve tracking accuracy in case of occlusion, we also employ overlap-based optimization. The purpose of this optimization is to maximize covering the area of the objects by occluding vehicles. For example, considering a simple case of two occluding vehicles A and B associated with an object C, we perform the following operations:

$$F = (A \cap C) \cup (B \cap C) \; and \; D = F \oplus C$$

where D represents the area of the object that is not covered by the vehicles. We compute the center of D, and then iteratively move the vehicles toward that center to cover the overall area of the object as much as possible until no improvement is achieved. Finally, the pseudo code for the overall tracking system can be found in Table 3.3.

3.5 Experiment results

In our work so far, there are totally seventy-two videos processed by our system. These videos were captured from July 2009 to October 2010 for the same roundabout. However, main traffic scene of interest in all these videos is an entrance to the roundabout, not the roundabout itself. An image of the traffic scene can be found in Figure 3.3. In addition, it can be seen that the entrance way has a slope, therefore does not satisfy the ground-plane constraints. Therefore, the camera calibration algorithm proposed in Chapter 2 was not used for these videos of this specific traffic scene. Instead, the camera calibration algorithm would be used to process videos that target the whole roundabout to derive origin-destinations for example. Without camera calibration, the proposed tracking system derives vehicle states in 2-dimensional (2D) image, instead of the real 3D world. In spite of that, the proposed system is still valuable and can measure a broad variety of traffic data such as vehicle volume, travel time, rejected gaps, accepted gaps and follow-up time. What cannot be found out is only vehicle speed without camera calibration. The overall traffic data collection system has been implemented in software.

We first show the results from vehicle segmentation. Figure 3.2 (a) shows the image frame that has encountered significant camera shaking with respect to the previous image frame, and Figure 3.2 (b) shows segmentation results in traditional background subtraction methods, and finally Figure 3.2 (c) gives segmentation results with the MoG method. It is noted the MoG background modeling and shaking removal algorithm are robust to false detections from camera shaking in comparison to other methods.

(a)

(b)

(c)

(a): original image (provided from MnDOT video cameras).

(b): segmentation results using traditional background subtraction

(c): segmentation results using the MoG method

Figure 3.2. Segmentation results in case of camera shaking.

Figure 3.3 shows the overlay of some sample vehicle trajectories for a 2-hour video (4-6pm) recorded on July 17th 2009 for a roundabout entrance. All tracked vehicle trajectories have been checked for correctness and those that do not have complete vehicle trajectories from ramp entrance to exit are excluded.

The Kernel-based tracking in joint feature-spatial spaces and overlap-based optimization are applied when two or more vehicles encounter occlusions. Figure 3.4 gives an example which shows how the system identifies the positions of individual vehicles involved in occlusions, when one vehicle (#2) is passing the other two (#3 and #4) leading to a merging of three of them. Also notice the two waiting vehicles (#5 and #6) with significant occlusions on the left-hand side of Figure 3.4, due to incoming vehicles (#7 and #8) from the other ramp entrance that has right-of-way. Note that in the figure, objects are represented in red while vehicles in other various colors.

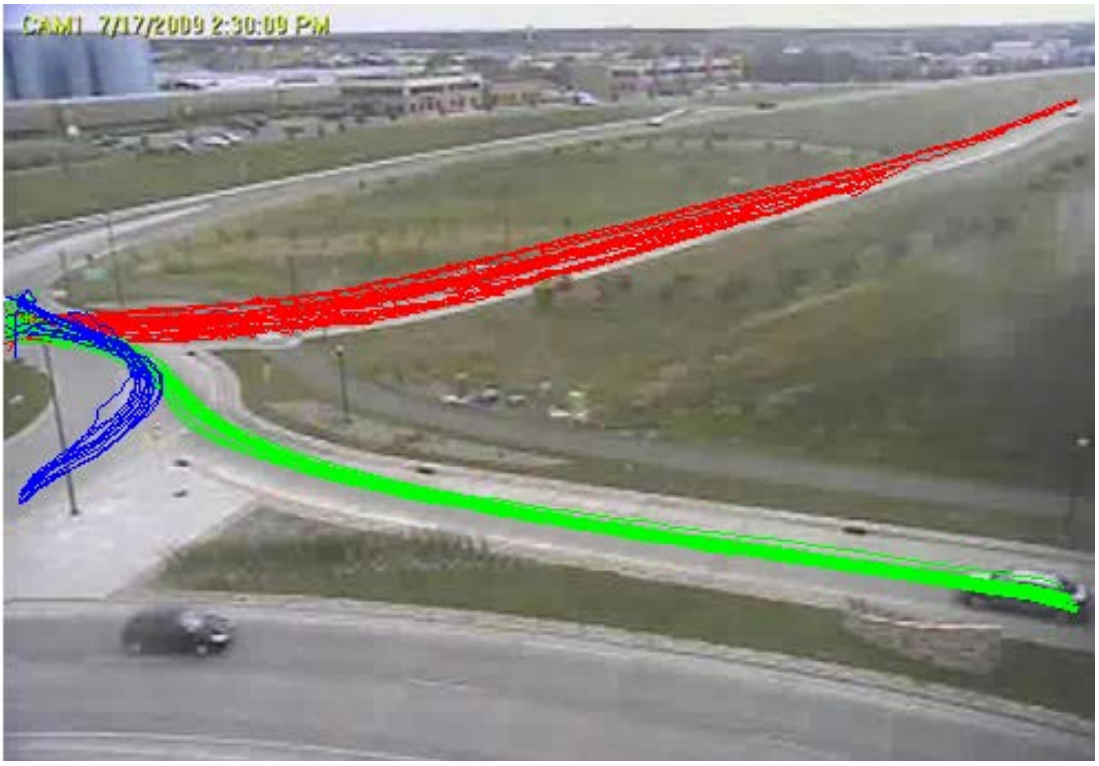
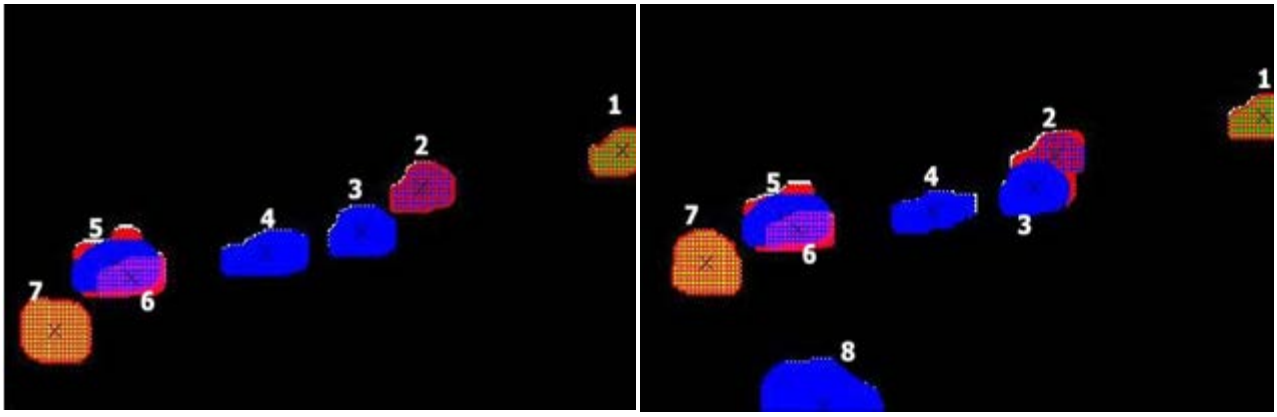


Figure 3.3. Overlay of vehicle trajectories (one line represents one vehicle trajectory). (Image provided from MnDOT video cameras.)

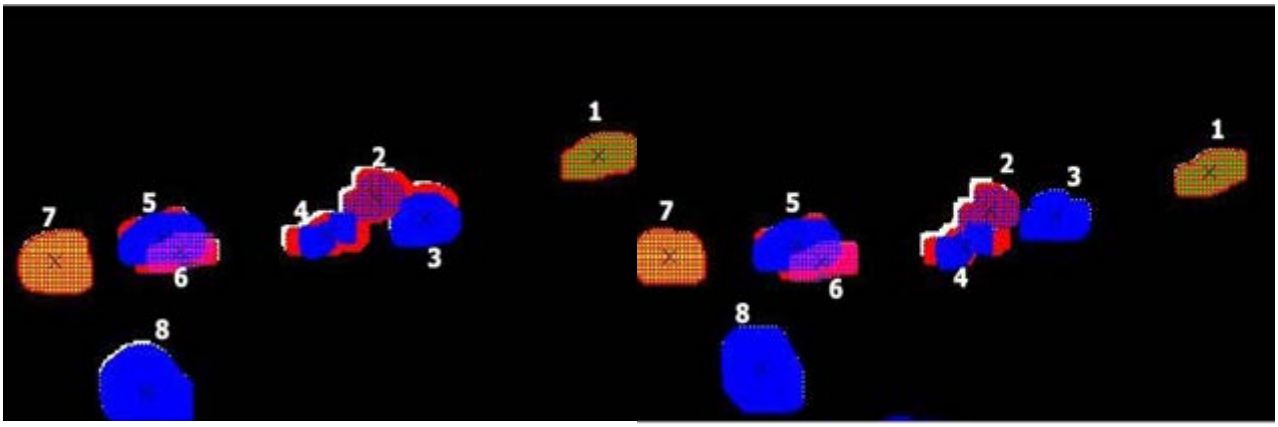
3.6 Summary and conclusion

In this Chapter, we present in detail the vehicle tracking module in the overall traffic performance measurement system. Vehicle tracking is the main core of the traffic performance measurement system, as eventually the accuracy of collected traffic data depends on how well vehicle tracking are performed or the accuracy of the outputs in this step, vehicle trajectories.

Vehicle tracking include two major steps, vehicle segmentation and the actual tracking. Vehicle segmentation is very critical and in this project we propose to use the MoG method for vehicle segmentation to cope with practical camera shaking issues. In the tracking part, we propose to use combined Kalman filtering, Kernel-based tracking and overlap-based optimization to cope with vehicle occlusions. The experiments results have shown that the vehicle tracking modules are capable of dealing with these issues.



Frame 250 Frame 255



Frame 260 Frame 265

Figure 3.4. Tracking with vehicle occlusions.

4 Traffic Data Collection

The proposed traffic performance measurement system has mainly three modules, camera calibration, vehicle tracking and traffic data collection. After camera calibration, vehicle tracking is the second main module of the overall system, followed by data collection module to be discussed in the next Chapter. As the outputs from the vehicle tracking modules are vehicle trajectories of all vehicles (provided they are correctly detected and tracked), and from those the data collection modules needs to mine the vast amount of data for useful traffic data, we also call this step data-mining. The output of the vehicle tracking module is a list of vehicle trajectories with their details such as the start and end frame, and their position at each specific frame while they're still in the scene. Figure 4.1 shows the hierarchy of the output. In Figure 4.1, the list of vehicles shows the total number of detected and tracked vehicles; and then each vehicle has a number of parameters as shown and particularly important is the "state", which has recorded the vehicle's position and speed (in the 2D or 3D domain depending on whether camera calibration are used) at each time moment. In our example shown in Figure 4.1, the vehicle's position and speed are represented in the 2D image domain as explained in Chapter 3 we did not use camera calibration for the specific traffic scene recorded in the videos.

In the rest of the Chapter, we discuss how to automatically extract traffic data from the resulting vehicle trajectories provided by the vehicle tracking module. In Section 4.1, we discuss how to extract vehicle count and travel time. In Section 4.2, we focus on accepted, rejected gaps and follow-up time. Then, some experiment results are shown in Section 4.3. Finally, a summary of this Chapter and conclusions are offered in Section 4.4.

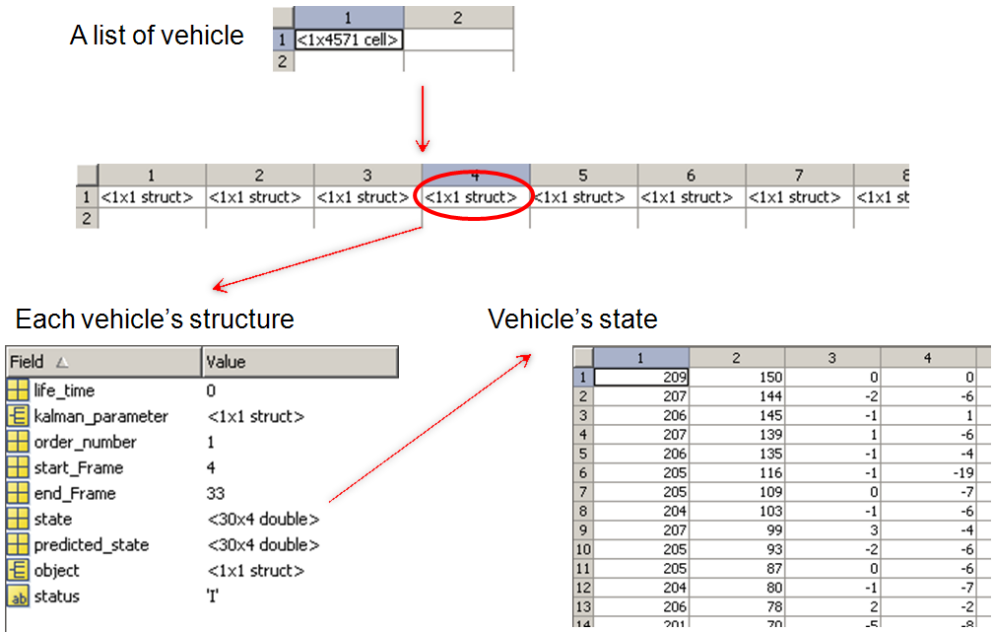


Figure 4.1. The hierarchy of the result from tracking.

4.1 Vehicle count and travel time

In traffic data collection, the number of vehicles is a basic type of data to acquire. In our system, every new vehicle detected will be given an incremental Identification Number (ID), from which we then know the total number of vehicle in the scene.

On the contrary, travel time is not available in all surveillance systems. Many of them use vehicles' average speed to indirectly estimate the travel time. In the proposed system, travel time could be measured directly from trajectory. For example, as in Figure 4.2, we specify two lines (line1 and line2) for the roundabout entrance and the time it takes for each vehicle to travel between the two lines is derived as travel time. Those customized lines ensure that the travel time is consistently measured for all vehicles. Currently, the user needs to select a number of image points to define the set of lines for data collection purpose. Future work will eliminate this requirement of manual input.

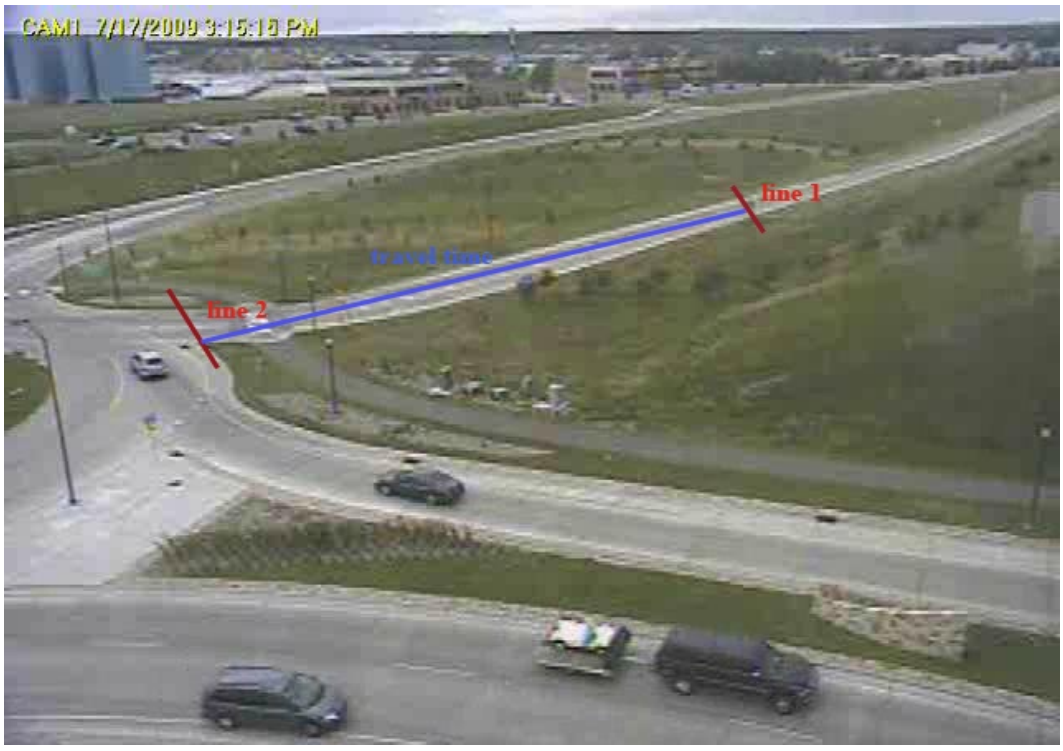


Figure 4.2. Travel time definition. (Image provided from MnDOT video cameras.)

4.2 Accepted, rejected gaps and follow-up time

We collect accepted and rejected gaps by using manually added lines as in Figure 4.3. We consider vehicle A from the roundabout entrance has entered the roundabout at the moment it passes line 2. If this happens while there is a vehicle B in the other roundabout entrance or the roundabout itself, we will group them as an accepted-gap pair (A, B). And the time duration from the moment at which A passes line 2 to the moment at which B passes line 4 is computed as the accepted gap.

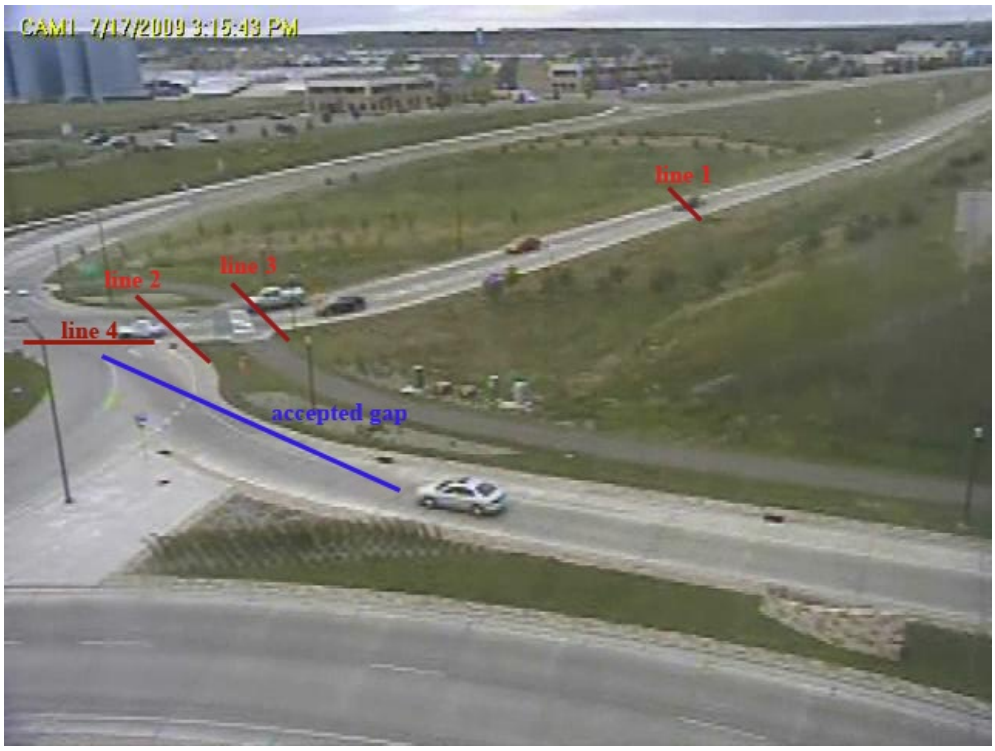


Figure 4.3. Illustration of accepted and rejected gaps. (Image provided from MnDOT video cameras.)

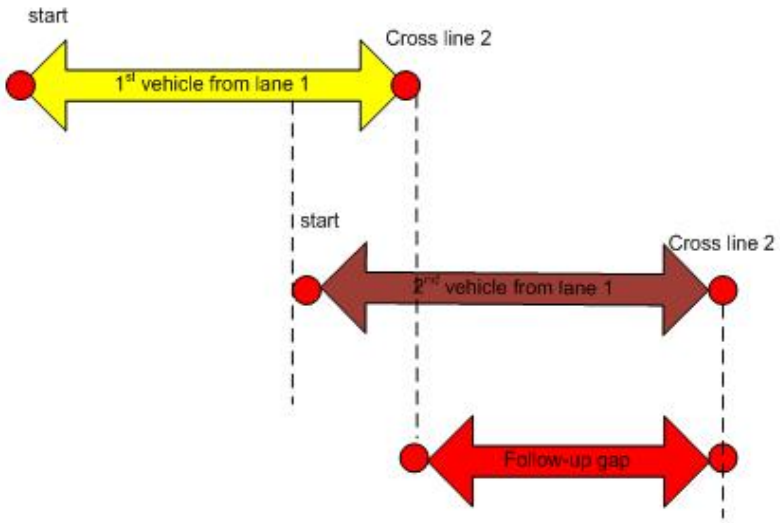


Figure 4.4. Follow-up time definition.

To collect rejected gaps, we consider a vehicle A from the roundabout entrance in waiting mode when it crosses line 3 but has not entered roundabout yet (i.e., it has not passed line 2). If there is a vehicle B in the other entrance or in the roundabout while A is in the waiting mode, we group them as a rejected-gap pair. And the time duration from the moment at which vehicle A

crosses line 3 (but not line 2 yet) to the moment at which vehicle B in passes line 4 is defined as the rejected gap.

The follow-up time is more complex to define. We consider two consecutive vehicles from the roundabout entrance to form a follow-up time if they both freely enter the roundabout. This means there should not be vehicles from the other entrance or the roundabout itself during the time duration when two consecutive vehicles cross line 2. Figure 4.4 shows how we define the follow-up time. In practice, the follow-up time in the case of roundabouts is more meaningful when the two consecutive vehicles enter the roundabout after long waiting and the roundabout frees up. Future work will take this into consideration.

4.3 Experiment results

The overall data collection system has been implemented in C and Matlab and runs on a 2.33GHz Xeon PC. We tested the system on a total of 72 videos captured from typical surveillance cameras installed by MnDoT on different days (mostly in the afternoon) and the average video length is 3 hours. The processing time for a 3-hour video is currently 6 hours in the PC. All the 72 videos are for the same traffic scene, a roundabout entrance (see the above Figures 4.2 or 4.3). The roundabout is located in Cottage Grove, Washington County in Minnesota.

In this Section, we show results from two videos collected on July 17th 2009 (one video for 2:30-3:55pm and the other 4-6pm). Figure 4.5 shows the number of vehicles that enters and exits the ramp 1 (denoted with the red line in Figure 4.2) every 60 seconds. Note that exiting the ramp 1 means the vehicle has entered the roundabout. In addition, Figure 4.5 shows the waiting time it takes for the vehicles to turn into the roundabout in ramp 1 every 60 seconds.

Figure 4.6 shows the accepted gaps for the two videos. The main error for gaps is incurred when there are significant occlusions. First, the vehicle did not turn into the roundabout but was taken so by the system due to poor detections sometimes, which caused false gap sizes. Second, the vehicle turned into the roundabout but was not taken so, causing some gap misses. Considering the number of false gaps and gap misses, the accuracy of gaps in terms of the number of correct entries is 95%. In other words, out of 100 collected gaps, on average only 5 of them are either invalid gaps or gap misses. However, it should be noted that the accuracy of gap size in terms of actual time value is close to 100% when averaged over the number of entries.

We also collected rejected and follow-up time for the video captured by the same camera on September 7th 2010. Data are shown in Figure 4.7.

4.4 Summary and conclusion

In summary, the proposed system is capable of producing a variety of traffic data. When compared to ground-truth measurements obtained from manual inspection of the videos, it is found that the accuracy on the traffic data is in the range of 70% to 90%, which is encouraging for a single-camera-based video system. The main sources of error are large vehicle shadow (especially in the afternoon time) and heavy vehicle occlusions (due to either limited camera view from a single camera or large vehicle queue in the entrance). As previous mentioned, the data collection module discussed in this Chapter follows the vehicle tracking module in Chapter

3 and does not incur any accuracy loss given vehicle trajectories from the vehicle tracking module. Therefore, the error is strictly from vehicle tracking, for example, poor detection and vehicle occlusion.

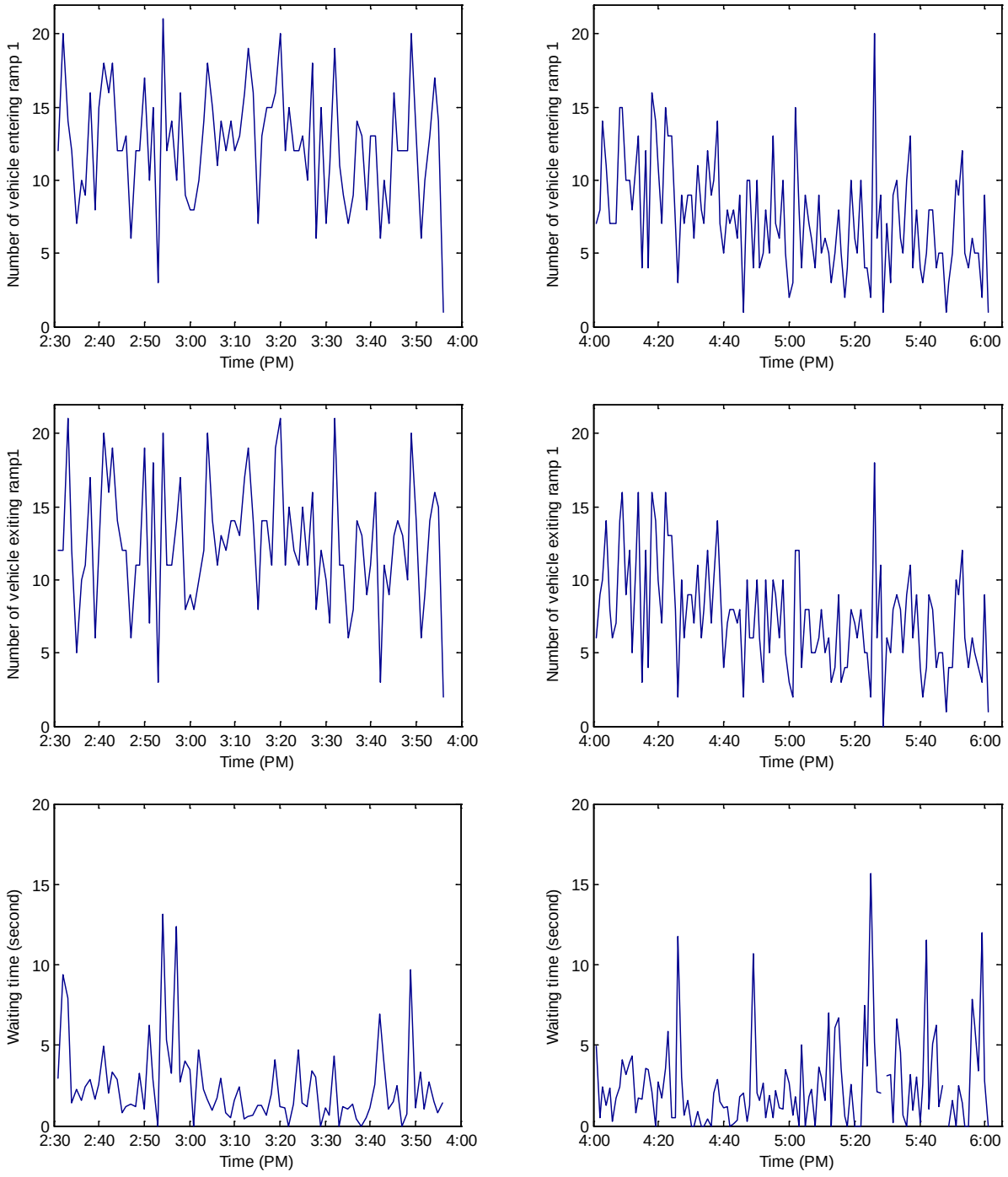


Figure 4.5. Vehicle count and waiting time for video 1 (2:30-3:55pm) on left-hand side and video 2 (4-6pm) on right-hand side.

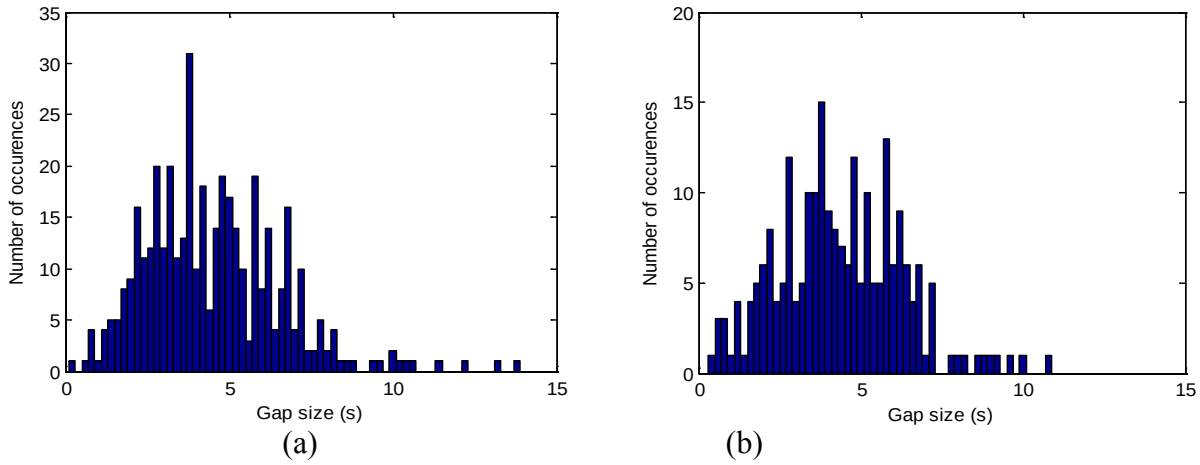


Figure 4.6. Histograms of accepted gap sizes for (a) video 1 (2:30-3:55pm) and (b) video 2 (4-6pm).

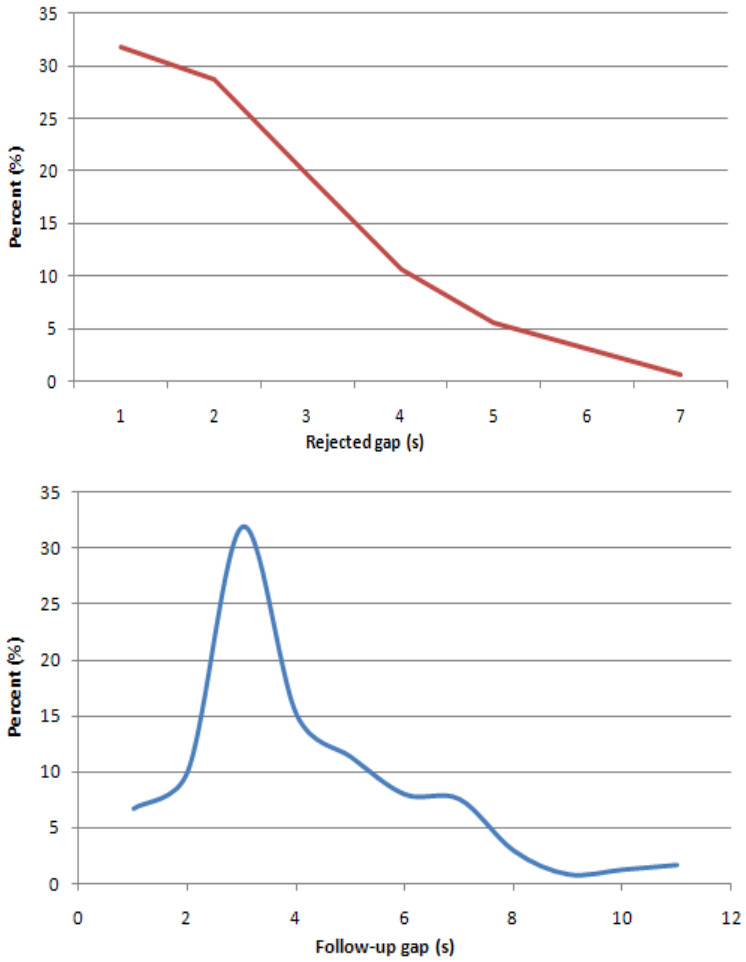


Figure 4.7. Rejected gaps and follow-up time for video on September 7th 2010 (horizontal axis refers to gaps in seconds and vertical axis percentages of all occurrences).

5 Summary and Conclusions

In this project, we have developed a tracking-based traffic data collection system to extract traffic data from a single-camera-based vision system. We first propose a novel circle-based method for camera calibration of roundabout traffic scenes in Chapter 2. Then, the system takes pre-recorded videos as inputs, applies the mixture-of-Gaussian background modeling method and a shaking-removal algorithm to segment vehicles, subsequently Kalman filtering, Kernel-based tracking and overlap-based optimization to derive complete vehicle trajectories as discussed in Chapter 3, and finally a data mining algorithm to extract interested traffic data as in Chapter 4.

Extensive experiments on many videos of a real-world roundabout traffic scene have shown that the proposed data collection system can provide traffic data, such as vehicle count, waiting time (or travel time), rejected gaps, accepted gaps and follow-up gaps with up to 90% accuracy. Compared to previous methods using sensors or loop inductors for traffic data collection of intersections [42-43], the proposed data collection system derives the complete vehicle trajectories, giving enough details to collect all types of interested traffic data. The main benefit of the proposed system is that it can automatically collect traffic data with minimum requirements of manual inputs so that significant labor work and cost can be saved.

A few specific innovations achieved in this research project are as follows:

- First of all, to the best of our knowledge, this project reports the first work on traffic data collection for roundabouts. Also, the proposed system can be used for traffic data collection for intersections, as traffic behavior at intersections is similar to that at roundabouts.
- This project also develops a novel and simple method for camera calibration of roundabouts. However, we did not use the calibration results in our experiments due to the sloping roundabout entrances in the videos and absence of the complete roundabouts.
- Third, we propose to apply Kernel-based tracking and overlap-based optimization to address vehicle occlusions, which has not been explored before. The proposed system also employs a shaking-removal algorithm to address false vehicle segmentations resulting from camera shaking.

The proposed system is similar to Miovision in concept [44]. However, as mentioned before Miovision is restricted to vehicle count for intersections so far [44], whereas the proposed system can collect all types of interested traffic data for roundabouts and can be readily extended for intersections.

At the same time, we are also aware of the limitations of the developed system, which will be addressed in future work. First, the proposed system has not addressed nighttime vehicle tracking, as vehicles are more difficult to detect at night. Though there are methods proposed to use the headlights or taillights for vehicle tracking, they are not found to be very reliable in our experiments of roundabouts (especially with the winter weather in Minnesota). Second, vehicle shadow is also a difficult issue. Vehicle shadows could be recognized as part of the objects, which lead to tracking errors. While in some cases, vehicle shadow can be tolerated, it is too large in the afternoon. In particular, when the sun moves to some specific angles, the sunlight

significantly blocks the view of the camera in the videos used in our experiments. One way to deal with vehicles' shadows is to take advantage of the color information, which is not very reliable in our experiments, though. Third, a single camera in the developed system may not provide adequate view coverage of the whole roundabout and this prevents deriving original-destination pairs of the roundabout. If the single camera is zoomed out to cover the whole roundabout, the vehicle may be too small to be differentiated from environmental noise during image processing. The other main challenge with a single camera is vehicle occlusion, which is a notorious problem again due to limited view of one camera. For example, vehicle #2 and #3 have significant occlusions in Figure 2.6 due to the single-camera view. Future work plans to explore the use of multiple cameras to improve tracking accuracy.

References

Traffic Detector Handbook, 3rd edition, vol. 1, pp 1-10, 2006, US Department of Transportation, Washington, D.C.

Intelligent Transportation System Unit Costs Database, 2007, US Department of Transportation, Washington, D.C.

P. G. Michalopoulos, "Vehicle detection video through image processing: the Autoscope system," *IEEE Transactions on Vehicular Technology*, vol. 40, no.1, pp. 21-29, 1991.

Image Sensing Systems, Inc., "Autoscope," http://autoscope.com/, accessed Jan. 2011.

Z. Zhang, "Camera Calibration," Chapter 2 in *Emerging Topics in Computer Vision*, edited by G. Medioni and S.B. Kang, pp. 4-43, 2004, Prentice Hall, Upper Saddle River, NJ.

O. Faugeras, "Three-Dimensional Computer Vision: a Geometric Viewpoint," 1993, MIT Press, Boston, MA.

P. Sturm, S. Maybank, "On plane-based camera calibration: A general algorithm, singularities, applications," *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 432-437, 1999, Ft. Collins, CO.

Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, 2000.

Z. Zhang, "Camera calibration with one-dimensional objects," *Proceedings of European Conference on Computer Vision*, vol. IV, pp. 161–174, May 2002, Copenhagen, Denmark.

R. I. Hartley, "An algorithm for self-calibration from several views," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 908-912, 1994, Seattle, WA.

T. N. Schoepflin, D. J. Dailey, "Dynamic Camera Calibration of Roadside Traffic Management Cameras for Vehicle Speed Estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 2, pp. 90-98, 2003.

S. Gupte, O. Masoud, R. F. K. Martin, N. P. Papanikolopoulos, "Detection and Classification of Vehicles," *IEEE Trans. Intelligent Transportation Systems*, vol. 3, no. 1, pp. 37-47, 2002.

C. C. C. Pang, W. W. L. Lam, N. H. C. Yung, "A Novel Method for Resolving Vehicle Occlusion in a Monocular Traffic-Image Sequence," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 3, pp. 129-141, 2004.

O. Masoud, N. P. Papanikolopoulos, "Using Geometric Primitives to Calibrate Traffic Scenes," *Transportation Research Part C*, vol. 15, pp. 361-379, 2007.

E. K. Bas, J. D. Crisman, "An Easy to Install Camera Calibration for Traffic Monitoring," *Proceedings of IEEE Intelligent Transportation Systems Conference*, 1997, Boston, MA.

Y. Li, F. Zhu, Y. Ai, F. Wang, "On Automatic and Dynamic Camera Calibration based on Traffic Visual Surveillance," Proceedings of IEEE Intelligent Vehicle Symposium, 2007, Istanbul, Turkey.

K. Song, J. Tai, "Dynamic Calibration of Pan-Tilt-Zoom Cameras for Traffic Monitoring," *IEEE Transportation System, Man, and Cybernetics*, vol. 36, no. 5, pp. 1091-1103, 2006.

T. H. Thi, S. Lu, J. Zhang, "Self-Calibration of Traffic Surveillance Camera using Motion Tracking," *Proceedings of IEEE Intelligent Transportation Systems Conference*, 2008, Beijing China.

N. K. Kanhere, S. T. Birchfield, W. A. Sarasua, "Automatic Camera Calibration Using Pattern Detection for Vision-Based Speed Sensing," *Transportation Research Record*, no. 2086, pp. 30-39, 2008.

S. Pumrin, D. J. Dailey, "Dynamic Camera Calibration in Support of Intelligent Transportation Systems," *Transportation Research Record*, no. 1804, pp. 77-84, 2002.

F. W. Cathey, D. J. Dailey, "One-Parameter Camera Calibration for Traffic Management Cameras," *Proceedings of IEEE Intelligent Transportation Systems Conference*, 2004, Washington, D.C.

X. Meng, H. Li, Z. Hu, "A New Easy Camera Calibration Technique Based on Circular Points," *British Machine Vision Conference*, 2000, Bristol, United Kingdom.

Q. Chen, H. Wu, T. Wada, "Camera Calibration with Two Arbitrary Coplanar Circles," *Proceedings of European Conference on Computer Vision*, 2004, Prague, Czech Republic.

G. Jiang, L. Quan, "Detection of Concentric Circles for Camera Calibration," *Proceedings of IEEE Conference on Computer Vision*, 2005, Beijing, China.

H. Zhong, F. Mai, and Y. S. Hung, "Camera Calibration Using Circle and Right Angles," *Proceedings of IEEE Conference on Pattern Recognition*, 2006, New York, NY.

Y. Zheng, Y. Liu, "Camera Calibration using One Perspective View of Two Arbitrary Coplanar Circles," *Optical Engineering*, vol. 47, no. 6, 2008.

F. Abad, E. Camahort, R. Vivo, "Camera Calibration Using Two Concentric Circles," *Proceedings of the International Conference on Image Analysis and Recognition*, 2004, Proto, Portugal.

S. Mensah, S. Eshragh, A. Faghri, "A Critical Gap Analysis of Modern Roundabouts," *Transportation Research Board 2010 Annual Meeting*, 2010, Washington, D.C.

S. Mandavilli, "Study of Operational Performance and Environmental Impacts of Modern Roundabouts in Kansas," *Transportation Research Board National Roundabout Conference*, 2005, Vail, CO.

K. Parma, “Going around the Neighborhood – a Roundabout Case Study in Texas,” http://www.leeengineering.com/roundabouts/, accessed January 2011.

R. Hartley, A. Zisserman, “Multiple View Geometry in Computer Vision,” 2nd edition, Chapter 6, 2003, Cambridge University Press, Cambridge, UK.

E. R. Davies, “Machine Vision: Theory, Algorithm, Practicalities,” 3rd edition, Chapter 12, 2005, Morgan Kaufmann, Waltham, MA.

A. Fitzgibbon, M. Pilu, R. B. Fisher, “Direct Least Square Fitting of Ellipses,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 476-480, 1999.

J.-Y. Bouguet, “Complete Camera Calibration Toolbox for Matlab,” California Institute of Technology, Pasadena, CA, http://www.vision.caltech.edu/bouguetj/, accessed January 2011.

H. K. Yuen, J. Illingworth, J. Kittler, “Ellipse Detection using Hough Transform,” *Proceedings of 4th Alvey Vision Conference*, pp. 265-271, 1988, Manchester, UK.

C. Smith, C. Richards, S. Brandt, N. P. Papanikolopoulos, “Visual Tracking for Intelligent Vehicle Highway Systems,” *IEEE Transactions on Vehicular Technology*, vol. 45, no. 4, pp. 744-759, 1996.

S. Gupte, N. P. Papanikolopoulos, *Algorithms for Vehicle Classification*, MN/RC-2000-27, 2000, Minnesota Department of Transportation, St. Paul, MN.

D. Dailey, *CCTV Technical Report: Phase III*, Technical report, WA/RD-2006-635.2, 2006, Washington Department of Transportation, Olympia, WA.

Y. Wu, F. Lian, C. Huang, T. Chang, “Traffic Monitoring and Vehicle Tracking using Roadside Camera,” *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 2006, Taipei, Taiwan.

R, Ervin, C. MacAdam, J. Walker, S. Bogard, M. Hagan, A. Vayda, E. Anderson, *System for Assessment of the Vehicle Motion Environment (SAVME)*, UMTRI-2000-21-1, 2000, US Department of Transportation, Washington, D.C.

“Next Generation of Simulation Program (NGSIM)”, www.ngsim-community.org, accessed January 2011.

T. Kwon, *Portable Cellular Wireless Mesh Sensor Network for Vehicle Tracking in an Intersection*, CTS 08-29, 2008, Minnesota Department of Transportation, St. Paul, MN.

A. Abdel-Rahim, B. Johnson, *An Intersection Traffic Data Collection Device Utilizing Logging Capabilities of Traffic Controllers and Current Traffic Sensors*, 2008, National Institute for Advanced Transportation Technology, University of Idaho, Moscow, ID.

“Automated Video Traffic Studies,” 2011, Miovision Technologies, Inc., Kitchener, Ontario.

S. Gupte, O. Masound, R. F. K. Martin, N. P. Papanikolopoulos, "Detection and Classification of Vehicles," *IEEE Transaction on Intelligent Transportation Systems*, vol. 3 no. 1, pp. 37-47, 2002.

L. Wang, N. H. C. Yung, "Extraction of Moving Objects From Their Background Based on Multiple Adaptive Thresholds and Boundary Evaluation," *IEEE Transaction on Intelligent Transportation System*s, vol. 11, no. 1, pp. 40-51, 2010.

J. W. Hsieh, S. H. Yu, Y. S. Chen, W. F. Hu, "Automatic Traffic Surveillance System for Vehicle Tracking and Classification," *IEEE Transaction on Intelligent Transportation System*, vol. 7, no. 2, pp. 175-187, 2006.

W. W. L. Lam, N. H. C. Hung, "Highly accurate texture-based vehicle segmentation method," *Optical Engineering*, vol. 43, no. 3, pp. 591-603, 2004.

M. Heikkila, M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 657-662, 2006.

H. S. Lai, N. H. C. Yung, "A fast and Accurate Scoreboard Algorithm for Estimating Stationary Backgrounds in an Image Sequence," *Proceedings of the IEEE International Symposium Circuit System*, vol. 4, pp. 241-244, 1998, Monterey, CA.

N. M. Oliver, B. Rosario, A. P. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 831-843, 2000.

C. Stauffer, W.E.L Grimson, "Adaptive background mixture models for real-time tracking," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 246-252, 1999, Ft. Collins, CO.

I. Pavlidis, V. Morellas, P. Tsiamyrtzis, S. Harp, "Urban Surveillance Systems: From the Laboratory to the Commercial World," *Proceedings of IEEE*, vol. 89, no. 10, pp. 1478-1496, 2001.

B. Yiu, K Wong, F. Chin, R. Chung, "Explicit Contour Model for Vehicle Tracking with Automatic Hypothesis Validation," *IEEE International Conference on Image Processing*, 2005, Genoa, Italy.

T. N. Tan, G. D. Sullivan, K. D. Baker, "Model-based Localization and Recognition of Road Vehicles," *International Journal of Computer Vision*, vol. 27, no. 1, pp. 5-25, 1998.

D. Beymer, P. McLauchlan, B. Coifman, J. Malik, "A real-time computer vision system for measuring traffic parameters," *IEEE Conference on Computer Vision and Pattern Recognition*, 1997, San Juan, Puerto Rico.

R. Li, Y. Chen, X. Zhang, "Fast Robust Eigen-Background Updating for Foreground Detection," *Proceedings of the International Conference on Image Processing*, 2006, Atlanta, GA.

G. Welch, G. Bishop, “An Introduction to the Kalman filter” University of North Carolina, Chapel Hill, NC, http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html, accessed Feb. 2010.

D. Comaniciu, V. Ramesh, P. Meer, “Kernel-based object tracking”, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564-577, 2003.

C. Yang, R. Duraiswami, A. Elgammal, L. Davis, “Real-time Kernel-Based Tracking in Joint Feature-Spatial Spaces,” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005, San Diego, CA.